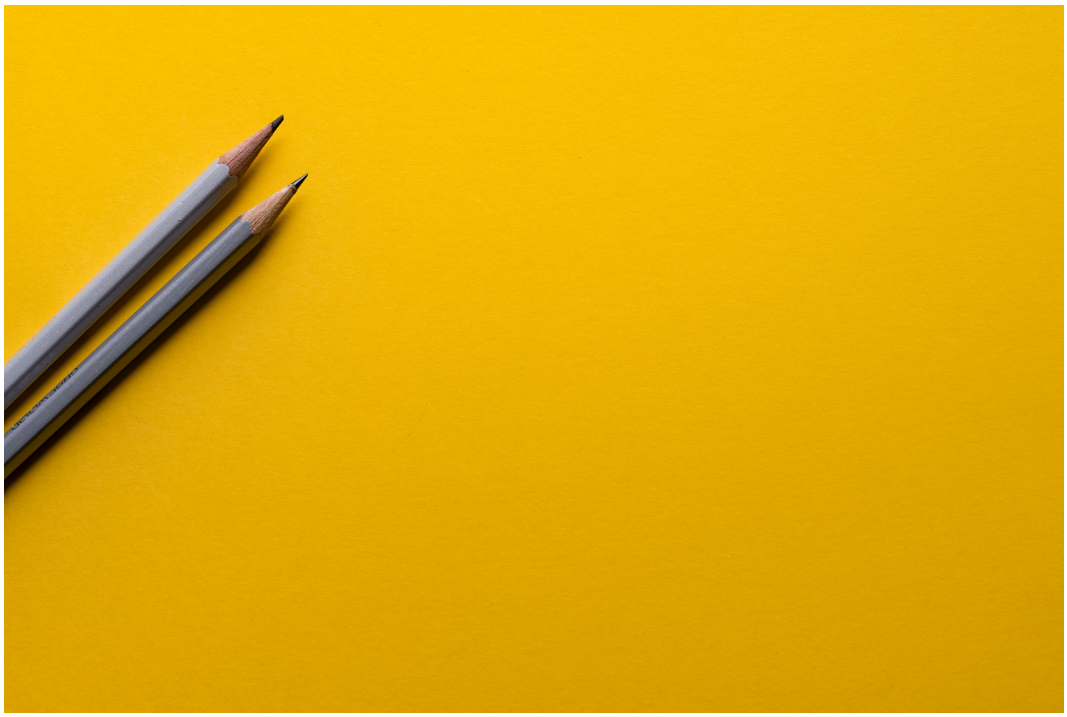
Reducing Human Error in Online Controlled Experiments

A case study at ING



Martijn Steenbergen

Reducing Human Error in Online Controlled Experiments

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

Martijn Steenbergen
born in Woerden, the Netherlands

Software Engineering Research Group
Department of Software Technology
Faculty EEMCS, Delft University of Technology
Delft, the Netherlands
www.ewi.tudelft.nl

ING
Bijlmerdreef 106
Amsterdam, the Netherlands
https://www.ing.com/Home.htm

Reducing Human Error in Online Controlled Experiments

Author: Martijn Steenbergen
Student id: 4311442
Email: martijn.steenbergen@outlook.com

Abstract

Online controlled experimentation (OCE), also called A/B testing, is an often used tool in industry to determine if deploying changes into production is the right decision to make. Running experiments has shown an immense impact to the revenue of companies in industry, however this type of experimentation comes with a lot of pitfalls, of which some that can invalidate the entire experiment.

This thesis describes the impact these pitfalls have on the work of experimenters at ING, a global bank, by performing informal interviews with practitioners and performing a survey with 52 participants. Next, building on existing solutions, a set of solutions is proposed to solve these pitfalls. To determine if these solutions solve the problem and will help the experimenter, these solutions are validated in the same survey.

This thesis shows that experimenters are well informed about the existence of pitfalls and believe that almost all should be resolved, with the exception of competitor safety, which is believed to not be important. There are many promising solutions to these pitfalls which experimenters rate as helpful. The best rated solution was "Enforcing the correct experiment duration". Almost all respondents perceived the solution to (slightly) help the experimenter in performing their experiments. Finally, this thesis creates a roadmap for evaluating these solutions in a real-world scenario.

Thesis Committee:

Chair:	Prof. Dr. A. van Deursen, Faculty EEMCS, TU Delft
University supervisor:	Ass. Prof. Sebastian Proksch, TU Delft
Company supervisor:	Kevin Anderson, ING
Committee Member:	Ass. Prof. Jan S. Rellermeyer, Faculty EEMCS, TU Delft

Preface

Dear wonderful reader,
Thank you for being awesome and reading this thesis. 5 years ago, I started my journey at the TU Delft. Never could I have imagined that I was starting the most amazing adventure of my life. Your kindness and knowledge has made me the person I am today. Thank you for that.

They say that a master thesis is the first thing you really do on your own. Although that is partially true, there are so many people who have supported me and without them, this thesis wouldn't be what you find here today.

Thank you to my supervisors, Arie, for giving me the freedom to do what I like while always pointing me in the right direction. Thank you, Sebastian, for pushing me to see the facts and helping me create something actually achievable. Thank you, Kevin, for bringing me into contact with all of the amazing people at ING and for being there. It was an amazing opportunity to see you work and shine.

Thank you, all the people that proofread my thesis. Thank you, Irene, for always reading my thesis and giving me feedback (even when it was bad) and helping me plan in the beginning days. You are an amazing addition to the team. Thank you Chiel, for the mountain of feedback that I hope I did proud.

Thank you, Team Tetris for welcoming me with open arms, letting me work in your codebase and helping me with all the problems I had. Thank you, survey respondents for making my research possible. Thank you, Huiskool, for helping me grow into the person I am today. You are always there for me. Thank you, 4th floor, for making these last years amazing. Thank you, everyone from the Teaching Team, SERG, Flying Fish and so many others that I met during these years. The greatest honor I could have is that I gave you as much happiness as you gave to me.

Thank you.

Contents

List of Figures

Chapter 1

Introduction

In 2014, Facebook engineers came back after Christmas and found that they had an problem. During Christmas, people had uploaded an enormous amount of Christmas pictures (more than Flickr had gotten in the entire lifetime). This also meant that the amount of content reported as hate speech, sexual abuse, etc. went up so drastically the team responsible for handling reports couldn't keep up anymore. When Facebook engineers started to look at the reports, they realized that 97% of these reports were miscategorized. Weirdly, in many cases the people making the report were in the picture. In fact, a picture flagged as hate speech was usually nothing more than an embarrassing image. To solve this problem, the team made it their goal to reduce the amount of reports by having the person contact the poster of the image to take it down. To achieve this, they added "Its embarassing" to the possible reasons of reporting an image. When a user clicked that it would open a new conversation with that person. The team wanted to run an experiment to test what would work better: an empty chatbox (version A, the control) or a chatbox with a default message (version B, the treatment) [30]. The result was that with an empty chatbox only 20% of the people would start a conversation, but with the default message it was 50%. [43]. This default message was implemented for a long time on the site. This type of test is called an Online Controlled Experiment [30]. This type of experimentation is now used throughout the web and is used extensively by Facebook. For any user of Facebook, there is an almost 100% certainty they have been part of a controlled experiment. In 2015, a user was on average part of 10 experiments at any given time. [43]

1.1 Online Controlled Experiments

Online controlled experiments (OCEs) or in literature also called A/B test, randomized experiment (single-factor or factorial designs), split test, Control/Treatement test and parallel flight are everywhere on the internet [29, 30]. An online controlled experiment is where all visitor to a webpage are split between two different variants to see which performs better on some metric. The idea of a controlled experiments is relatively simple and dates back to Sir Ronald A. Fisher', a founding father of statistics [23]. The idea of systematically running experiments to find the return-on-investment of new features in software seems to

have been first termed by Dan McKinley in 2012 in a presentation about experimentation at Etsy [10]. This idea took off, as the web provides an unprecedented opportunity to evaluate ideas quickly using controlled experiments [30]; deploying changes on the internet is cheap and each experiment can exposes hundreds of thousands —and sometimes even tens of millions— of users to a new feature or change [24, 30]. Furthermore, controlled experiments enables the experimenter to control for external factors, making sure the change is related to a new feature, not to a random fluctuation [36]. Because of this controlled experiments are becoming a standard in data-driven software companies. [19, 18]

Controlled experiments can bring a form of clarity to an organisation. Every experiment needs to have an Overall Evaluation Criterion (OEC), which is a quantitative measure of the experiments objective [13] . These objectives should relate to the overall goal of the organization. This aligns the entire company to focus on one or more (ranked) goals [19, 30]. In the example in the introduction, this goal was number of started conversations. If number of sales is the goal, increasing prices to make more profit, will likely be bad idea, as this will lower the amount of sales.

1.2 Human Error in Controlled Experiments

The number of companies using experimentation and the number of experiments inside these companies are growing [36, 15]. Therefore it is infeasible to have only experts doing experiments. Furthermore, finding an expert in both experimentation *and* the context of the experiment is even less feasible.

To solve this, organizations use in-house tools [18] to help non-experts create and execute experiments. However, this tooling needs to embody the knowledge of an expert. This has proven difficult, as experts are still uncovering and solving problems related to OCEs. In every area, at every stage of running experiments pitfalls exist that can influence the process, at worst invalidating the experiments. Bad data can be actively worse than no data [19]. These pitfalls range from making sure the tooling to collect data are working correctly and making sure the platform is statistically valid, to making sure that the experiment layout is valid. A large amount of literature details these pitfalls and how they have affected the organization reporting it [19, 30, 10, 22].

We can define two main categories of pitfalls, those on a platform level and those on an experimenter level. On a platform level it is a must to make sure that the data is gathered correctly. This problem is, although important, only interesting in a technical perspective. However, pitfalls on an experimenter level combines the technical with a human aspect. That is why it was chosen focus on pitfalls, which contain *human error*.

Many companies are moving to become data-driven organizations and therefore are starting to use experimentation to make decisions. When building their own platform, a list of implementation guidelines is missing. This thesis will provide indications which areas might provide the biggest payoff.

1.3 Industrial Context: ING

ING is a global bank with its main base in the Netherlands and Belgium. It employs around 53,000 employees, which serve around 38.4 million customers, corporate clients and financial institutions in over 40 countries [27]. ING is already running some controlled experiments on the web and wants to expand this in other areas of the company. It can therefore be determined that it is in the midst of adopting this technology. ING has their own experiment platform called the Experiment Office, where tests can be created and results be viewed. ING has recently started a partnership with the TU Delft under the banner of AI4FinTech, under which this research was performed.

1.4 Research Questions

From the main problem statement follows **the main research question** this thesis tries to solve:

How can we reduce human error in online controlled experimentation?

To help experimenters, many areas will need to be resolved. Therefore this main question is divided in 3 sub questions. First, an overview needs to be created of the pitfalls, which are relevant for this research and solutions which can solve these problems. Therefore the first sub research question is:

RQ1 What are key pitfalls and their potential solutions?

After a list of solutions exist, they need to be validated. As one important aspect of supporting the experimenter is that any change we make should be in the benefit of the experimenter. Any change should not decrease the satisfaction the experimenter has of the platform. This requirements is both important, because it is the right thing to do and because organizations have an incentive to keep this in mind. After all, when changing to become a data-driven organization it is counter productive to have tools employees/experimenters do not like. Furthermore, a deeper dive into the pitfalls is required, as the same holds for the pitfalls themselves. Having a great solution does not mean something if the pitfall in and of itself is not believed to be important.

RQ2 How are pitfalls and their solutions experienced?

Once a clear idea exists of which solutions could work, implementing such solutions in a real-world environment should be the next priority. A plan needs to be layed out on how to implement the solution, solve problems on the way to implementation and and if the results after implementation collaborate the results of earlier work. Therefore the last subquestion is:

RQ3 How can one evaluate a solution in a real-world scenario?

1.5 Contributions

This work made many contributions to the scientific knowledge of continuous experimentation. To the authors knowledge, this is the first body of knowledge surveying experimenters with solutions for pitfalls, before implementation. In the rest of this section these contributions will be discussed.

First, **this thesis proposes solutions** to pitfalls in online controlled experimentation by looking at existing solutions.

Second, it **shows a comprehensive overview of the state of online controlled experimentation at ING**. By performing an interview with practisioners in the field, determining workflows and documenting experiences by practisioners and analyzing the current experiment platform, an empirical contribution[6] is added to the knowledge of OCE's.

Finally, another an empirical contribution[6] is created, as this thesis **evaluates how 19 pitfalls are experienced and verifies the 8 proposed solutions**, which can help experimenters avoid those pitfalls when experimenting.

1.6 Thesis outline

To have a good understanding of the industrial context (ING), in chapter 2 a small case study is done, to give the reader an overview of the state and practice of online experimentation at ING. This chapter concludes with informal interviews conducted with ING employees to get an overview of their workflow (2.3).

In chapter 3, the first part of RQ1 is answered; which pitfalls in existing research are key to this thesis. Each of these key pitfalls is then explained to the reader in greater detail. In chapter 4, the second part of RQ1 is answered: which potential solutions exist to solve the pitfalls. In chapter 5, the methodology is described on how a survey is used to determine how pitfalls and their solutions are experienced by experimenters (RQ2). After this survey has been performed, in chapter 6, the results of the survey are analyzed. These results can then inform the implementation of the solutions. In chapter 7, the thesis proposes a roadmap to evaluate the proposed solutions, in order to answer RQ3.

Finally, in chapter 8 the thesis is wrapped up, by drawing conclusions, discussing threats to validity and describing future work.

Chapter 2

The context of ING

Before we can dive deeper into the problem, we need to first have an overview of the context in which this problem resides.

The goal of this chapter is to present the reader a thorough view of the state of continuous experimentation at ING. This is done by first in section 2.1, by providing the history of experimention at ING and the place in the run walk and fly stage. Next, in section 2.2 the experiment platform of ING is discussed. Finally, in section 2.3 the interviews that were performed to gain a deeper understanding of the state of continuous experimentation at the company.

2.1 Experimentation at ING

ING is a global bank with its main base in the Netherlands and Belgium. It employs around 53,000 employees, which serve around 38.4 million customers, corporate clients and financial institutions in over 40 countries [27]. ING, just like many other companies is in the midst of digital transformation [25, 26]. More than 80 percent of customer interactions now go through mobile devices. This switch from offices to digital devices is so profound that the main message in the yearly report of the CEO at the time, Ralph Hamers, was that "The digital customer experience is the key differentiator". He explains that "ING's ambition is to be a leader in terms of the digital banking experience, offering retail and wholesale customers everywhere the same empowering and differentiating experience" [26]. Therefore optimizing this customer experience should be at the heart of the future of ING. One of the ways in which ING has been doing this, is via ING's Analytics Unit. Established in 2018 [26], its goal is to accelerate the bank's analytics capabilities and lead the transformation to become a truly data-driven company [26]. Currently more than a 100 data scientist are working at this unit to make this goal a reality [26].

History of Experimentation at ING From our conversations with employees, we learned that ING has a long history with experimentation. At the PostBank, a predecessor of ING, direct mail was sent to a subset of customers. The direct mail consisted of multiple parts, a header, images, multiple paragraphs, etc. For each of these parts, variants were created and

combinations of these were sent to customers. The most successful variants were combined into the final mail, which was sent to all customers.

2.1.1 Maturity of continuous experimentation at ING

To determine the state of ING testing, we will be using the *crawl walk run and fly* stage by Fabijan et al. [17]. The authors are experts in the field of continuous experimentation, however the framework that is described here does not seem to be referenced in in literature by other authors. Therefore we can consider this framework as highly indicative, but not conclusive to the state of experimentation at ING. According to this framework, ING is in the walk stage, similarly to most other companies trying to perfrom OCEs [18].

ING has made great progress in implementing Continuous experimentation, however there are still improvements that can be made. ING and the research community could use this to their advantage. ING is in the perfect position to document the struggles, roadblocks and solutions, best practices and strategies to lead other organizations moving to a data driven culture.

2.2 The Experiment Office at ING

Experimentation at ING is done using the Experiment Office. The Experiment Office is a tool built in-house to support their own experiments.

The Experiment Office is decoupled from the implementation of the experiment. This is done to support multiple platforms at once. Currently the web is the only supported platform on which experiments can be run. However ING is slowly rolling out support to enable running experiments on the mobile apps as well.

The Experiment Office accomplishes its decoupling, by having a web API, which a developer can call to register feedback. By calling endpoints on the web API the server can determine which variant this user came from and if the variant was successful. The Experiment Offices also allows the creation of experiments in a test or acceptance environment to test if the test is correctly set up.

One interesting thing to note is that the Experiment Office is unable to determine if a variant failed for a user. It's only able to determine that a user succeeded. There is no way for the implementation to register a failed instance and should be inferred from other metrics. This is a result of how the platform was implemented.

When creating an experiment, an experimenter might want to test the experiment in a test or acceptance environment before deploying to production. The Experiment Office offers this functionality. By having each environment on a different subdomain, a clear division is made to experimenters on where they are running their experiments.

The experiment is maintained by the Tetris team at ING. This team took over the Experiment Office from squad Panama in March of 2018. With a small team, they maintain and extend the Experiment Office.

2.2.1 Types of experiments

The Experiment Office has multiple ways of showing users different variants. Although only A/B-tests can be seen as an actual experiment, we will follow the documentation of the Experiment Office:

A/B test An A/B-test is an online controlled experiment, as follows from the definition. Within ING, the term A/B test is preferred over ”online controlled experiment”.

Pilot A pilot differs from an A/B test in the participation strategy. Instead of using a random subset of users to show variant A or B to, the experimenter can either provide a list of users to show the new variant to or users themselves can choose to see the new variant. This therefore introduces bias into the experiment and cannot be used as an actual experiment. However a Pilot can still be very useful to gather feedback from active users. Or to make sure a feature actually works before rolling it out.

Rollout To reduce the possibility of shipping a broken feature to customers, a rollout experiment type exists that enables the experimenter to slowly roll out a new feature to all users. As soon as it is manually determined that the future is not working as expected, if the new feature is breaking functionality or it is throwing exceptions, the feature can be rolled back without having been seen by the entire user base of the platform.

2.2.2 Creating an experiment

To create an experiment, two main steps are required. The first step is to setup the new experiment in the Experiment Office. The second step is to actually build experiments where the content exists (Polymer [21], CMS), commit this to the experimenters own repository and deploy this. To create new experiments in the web interface of the Experiment Office, an experimenter navigates to the dashboard and clicks new experiment. There they can choose the experiment type, as seen in Figure 2.1, which was discussed above.

After this is done, they can move to step 2 (Figure 2.2), where more information about the experiment is requested: the title and a hypothesis. This forces experimenters to think about the experiment and what they actually want to test and what the result of the experiment should be. Next up they choose a success criteria where they are forced down to decide when this experiment should be considered a success.

Finally, in the last step of the process, the experimenter chooses the target group and the tool/tracking method used to collect metrics and track the user through the system.

2.2.3 Experiments on the platform

The Experiment Office launched in March of 2015. The first experiment in production launched a year later, in December of 2016. A total of 1875 experiments have been run on the platform of which 1067 were an online controlled experiment. To get a better view of the experiments being run at ING, an export was made of the platform at June 26th 2020. The rest of this section will discuss insights from this data.

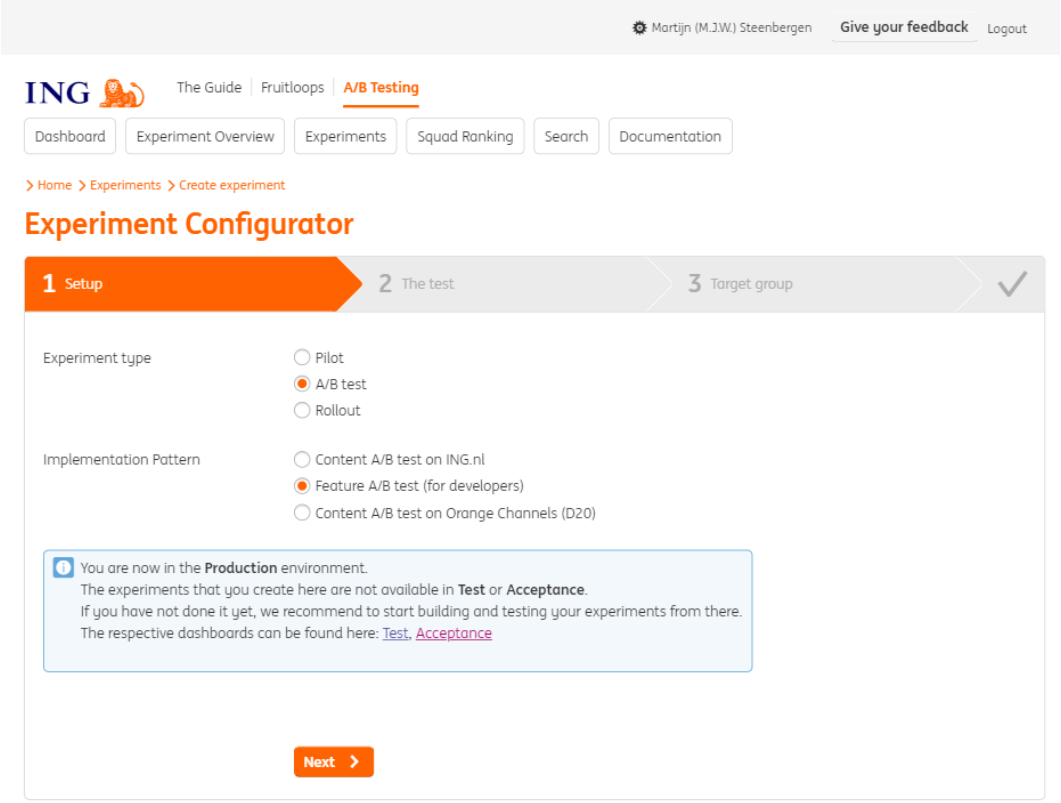


Figure 2.1: Step 1 of the experiment office flow, picking the experiment type and choosing where the experiment will be run

If one looks at the A/B tests performed from June 26th 2018 until June 26th 2020, 339 experiments were started, 43% in production, 37% in testing and 20% in acceptance.

From Figure 2.4 we can see that until March 2019 the number of experiments increased, however after this time, the number of experiments are slowly going down. If we look only at experiments in production (Figure 2.5), this change is even more pronounced. We can also see a sharp decline in the number of A/B experiments in the months of March, April, May and June 2020, which can probably be attributed to the corona outbreak of 2020 [49].

As there are three environments, a single experiment-id might exist on more than one environment. In the Experiment Office, 92 experiments are found in two stages, 148 in all three.

Many experiments do not yield significant results. If we look at A/B experiments run in production, only 26% (37 experiments) of the experiments are significant. In test and acceptance, only a single experiment is significant, which is explained by the low traffic on these pages, making it hard to reach significance. Many teams or squads, as they are called within ING, are using the platform, as can be seen in Figure 2.6, which is both a positive and

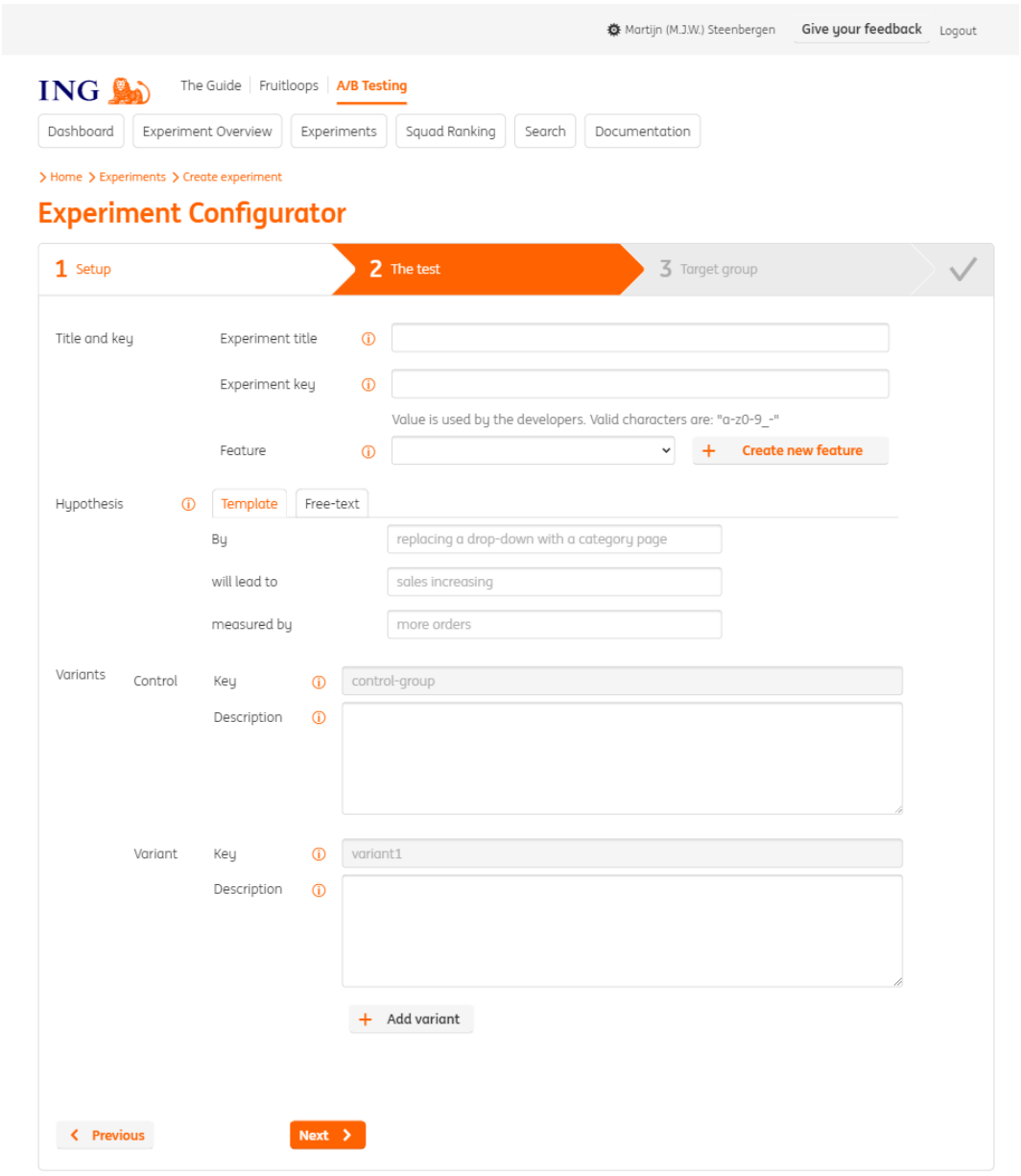


Figure 2.2: Step 2 of the experiment office flow, creating documentation for the experiment (title, hypothesis and variants)

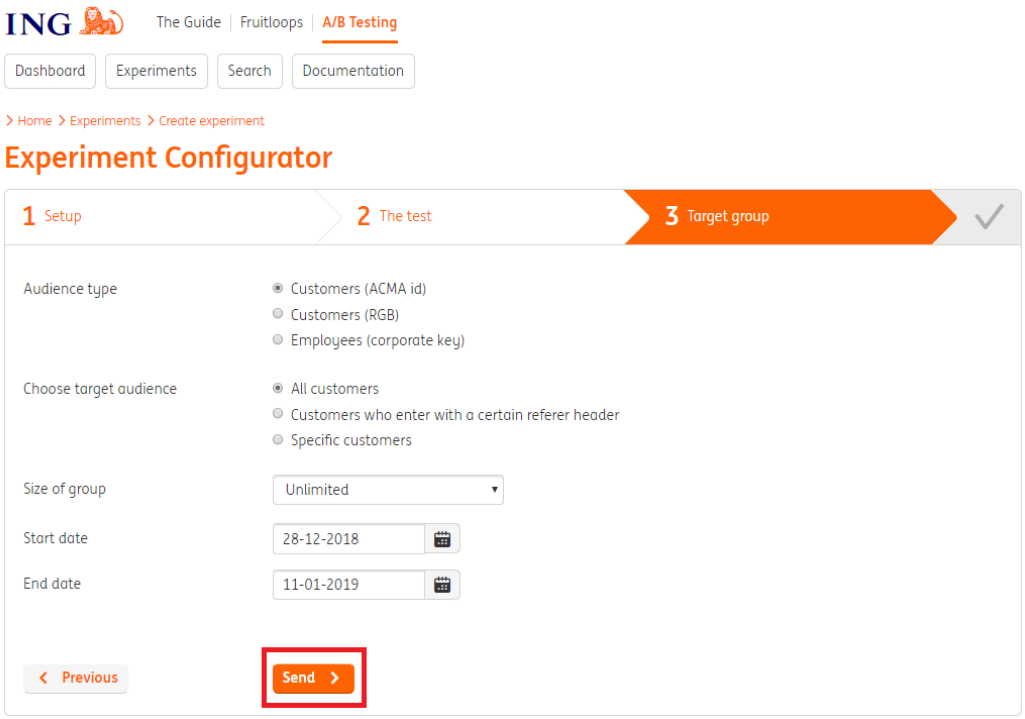


Figure 2.3: Step 3 of the experiment office flow, choosing the participants group and runtime of experiment

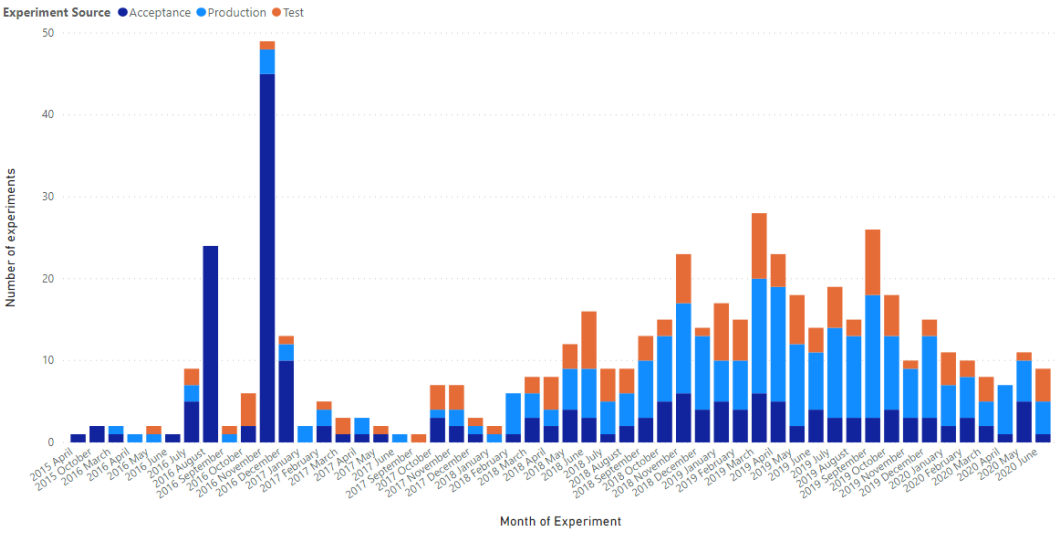


Figure 2.4: Overview of the number of experiments run at ING

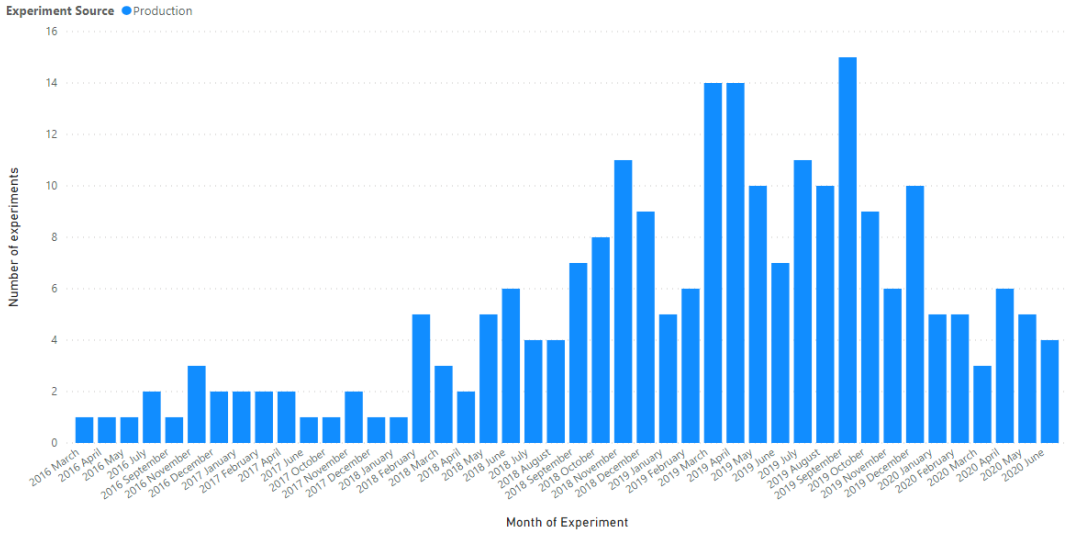


Figure 2.5: Overview of the number of experiments run in production at ING

negative result. It is positive, as this means it is likely many parts of the organisations are learning to become data-driven. However, at the maturity level (2.1.1) ING is in now, this could lead to knowledge being spread out too thin and therefore experimenters not having all the knowledge to run trustworthy experiments and get trustworthy results.

2.3 Interviews

Overview To get an overview of the difficulties experimenters face when creating controlled experiments at ING, we conducted interviews with 4 users (experimenters) (P1-P4) of the Experiment Office. These interviews were unstructured and performed in an informal way with the goal to get an overview of how online controlled experimentation is performed at ING. Questions included how they come up with an experiment, how they use the Experiment Office to run their experiment, what they think of the Experiment Office and what can be improved. Because the interviews were performed by a single person, and in an informal way, this can introduce bias into the results, both in the way the questions were asked and how the results are presented here.

Participants Participants were chosen to have many different viewpoints (experience, platform, etc.). Participants were requested via mail to have an interview of around 45 minutes. Each of the participants at the time of the interview had run experiments in the last month on the Experiment Office. The number of conducted experiments varies per participant. Numbers ranges from 5 (P1), 10 (P4) or 14 (P3). The experience of the participants differs from six months (P1) to 9 years (P2). P1, P2 and P3 all use a a content management system (CMS) where they create two different versions of the same page and hook the Experiment Office into it. P4 is a programmer, who uses code to create experiments and

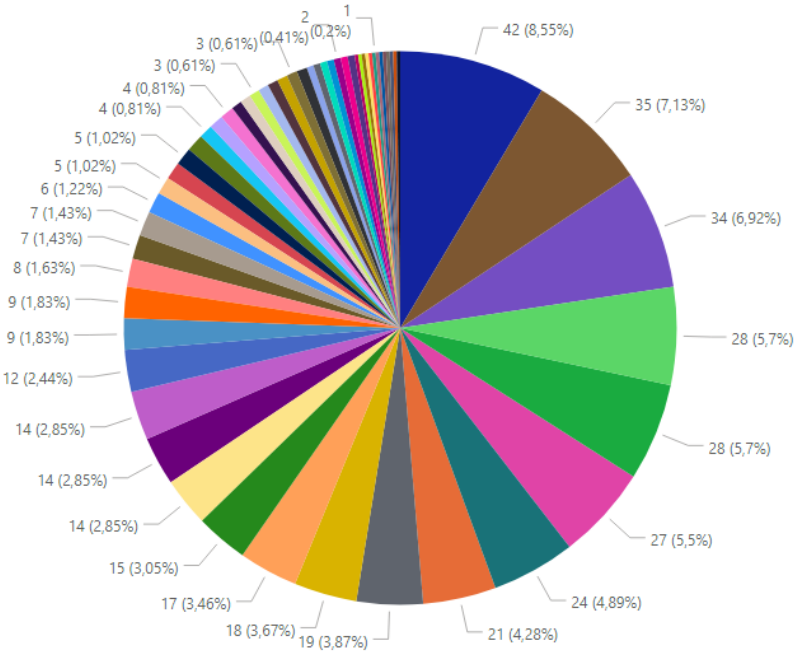


Figure 2.6: Number of experiments ING squads performed in the past two years. Each color is a squad. Names have been left out for non disclosure reasons

finds that creating experiments in this way takes a long time. All see the potential of using experiments to improve customer experience.

Other areas in OCE Interesting to point out is that experiments at ING are not only performed using the Experiment Office, but also using the tools from social media platforms, where ING puts advertisements. For example, Google ads or Facebook ads are used to serve potential customers ads for new products. P2 mentioned that there might be more experiments performed on these platforms than experiments on the ING.nl website. In this case the titles of the ads are changed or it is tested if a countdown timer or static time works best (P2).

Another area where ING performs experiments is via email. If a customer is already an ING customer, ING is allowed to contact the customer to sell a new product. According to P2, the moment this email is sent is crucial. P3 mentions the journey a customer takes when buying a product; See (realize you want something to solve a problem), Think (researching the best solution), Do (buy the product), Care (evaluating if they bought the best solution). Only in the Think and Do phases an email can be relevant and this poses the problem of figuring out in which stage the customer is and how one can best help them.

Origins of OCEs For different participants, inspiration for new experiments comes from different angles. P3 goes through the process the customer goes through and notes down everything that makes P3 think: "hmmm, that can be improved". Then customer feedback is added to add new ideas and determine which ideas have the largest chance of being impactful. P3 states that this order is a must, as you need context to understand what the customer is going through, before reading the feedback. P2 has a different approach and estimates that for them the feedback only accounts for around 20% of new experiments. For P2 the largest origin (˜65%) of the new ideas for experiments come from conversations with other squads. Lessons are learned in one squad, which can be relevant for other parts of the website. The final 15% are found in customer feedback, where they complain that something does not work correctly or have other ideas on improving the flow.

Creating an OCE After ideas have been brainstormed, the experiment needs to be launched. P3 mentions that they calculate the minimum duration of an experiment up front, although both P2 and P3 mention that any experiment will be run for at least two weeks. An observation, made explicit by P2, but also applicable P1 and P3, is that once an experiment is running, another one cannot be launched. This is impossible on the platform with CMS experiments and generally considered bad practice. This creates the situation where if there are 26 small improvements you would like to make to a page, this would take an entire year. This is why P3 mentions that 3 weeks is the maximum length an experiment can run, as otherwise it would simply take too much time, even if the results are not significant yet. This is also recognized in the literature [19].

Dissapointing outcomes In the end, conclusions need to be drawn. Something that is mentioned in literature [17] and here is reiterated, is that a negative outcome can be disappointing. P3 mentions that at ING, the best practice of leaving the current version in production is followed, after which something else might be tried. However the experimenter will always stay defensiveness about the idea. They might think of possible ways that their idea might be right after all. This enforces the idea that trust in the platform is of upmost importance.

Drawing conclusions It is also interesting to see that for drawing the conclusions multiple approaches are taken. P1 and P4 look at the entire sales funnel to draw the final conclusion. A sales funnel is the entire process from seeing the ad to buying the product. After all, more people clicking on an advertisement does not necessarily imply more people buying the product. If more people click on the advertisement, to continue the example, P1 will put it into production and check after two weeks if the number of sales has also increased. P2, however, states that they do not check the sales funnel, as there are too many factors involved to draw good conclusions. Before making a decision, P4 also looks at some guardrail metrics (3.3) to make sure nothing is breaking.

Thoughts of the experiment office Overall, participants are pretty happy with the Experiment Office and praise its ease of use. P1 found getting started with the tool easy. The par-

ticipants were also asked if there could be anything improved about the Experiment Office. Interestingly, all participants mentioned that they would like to keep track of more metrics. P2 and P3 want more metrics to base their conclusion on. Besides this, P4 would also like to have guardrail metrics. P1 would even like to follow the customer through the entire sales funnel and mentions that the integration of the metrics software, WebTrekk (7.2), could be improved upon. P1 and P4 mention that grouping is also lacking, for example P4 would like to group by device (who is on mobile).

Another point of improvement is the sharing of results. Both P1 and P2 miss the possibility to share knowledge about which changes have the biggest effect on the KPI's (key performance indicators). Especially since P2 mentions that such a large part for new ideas for experiments comes from other squads.

When talking about how P3 calculates the minimum duration, they mention they are using an Excel file, where they fill in fields and it gives them the length of an experiment. They would rather see this as a part of the Experiment Office.

P4 mentions that the Experiment Office does not maintain any libraries to integrate with code (in later meetings with the team that maintains the Experiment Office, we learned that they do not have the manpower to do so). P4 continues explaining that because of this, many teams, including his, have created their own libraries to interact with the platform, making the cost and buy-in for experimentation higher and harder than necessary.

Conclusion This sections paints a picture of how experiments are being performed, from the origin of the idea for testing, to drawing the conclusions. Interesting points are that ING is performing experiments, not only on web, but also on other platforms, like Facebook and Google ads. There are still many features that experimenters want, but overall the opinion of the Experiment Office is positive.

Chapter 3

Common pitfalls in Online Controlled Experiments

This chapter will be based on work of Mulders [40], who created a reprehensive literature review of 35 pitfalls found in the field of online controlled experimentation. These pitfalls are often made mistakes, while performing online controlled experiments. Other attempts to find additional pitfalls yielded no new results.

However, not all of these pitfalls are relevant for this research. Some of the pitfalls are already resolved, meaning ING has already implemented solutions to these problems. Trying to figure out the impact or possible solutions is no longer relevant in this case. For example, any experiment started at ING must already have a hypothesis. Asking how often they create an experiment without hypothesis, dives more into the effectiveness of the solution and although interesting, is out of scope of this thesis.

Other pitfalls do not contain a human element. As described by the research questions (1), any pitfall not caused by human error merely requires implementation effort and not are therefore requires a different (less interesting) way to be researched. For example, checking for data quality issues is not something that requires the interaction of an experimenter.

To make sure the filtering of the pitfalls was done correctly, the results were verified by an ING employee, who has extensive experience with OCE's.

Table 3.1 shows that 19 pitfalls are both human error and not resolved in the platform of ING. These we will explain further down below.

Something to mention is that there are three pitfalls, which Ernst [40] does not provide a reference in the original work nor for which references could be found. These pitfalls are "Withholding results", "Not confirming the winning variant" and "Encourage more experiments".

3.1 Falsifiable hypothesis

Issue There is no hypothesis for the experiment or the hypothesis cannot be proven wrong [19]

Id	Shorthand	Is Human Error	Is Implemented
01a	Hypothesis exists	Yes	Yes
01b	Falsifiable hypothesis	Yes	No
02a	Available metrics	No	No
02b	Direction of change in hypothesis	Yes	No
02c	Guardrail metrics	Yes	No
03a	Technical debt	Yes	No
03b	Competitor safety	Yes	No
03c	Churning users	Yes	No
04a	Minimum effect size	Yes	No
04b	Minimum duration	Yes	No
04c	Withholding results	Yes	No
05a	Simultaneous experiments	Yes	No
06a	Failure checks	Yes	No
07a	Needed metrics are present	No	No
07b	Dedicated metric collection system	No	Yes
08a	A/A tests	No	No
08b	P-values in A/A tests	No	No
08c	Significant A/A tests	No	No
09a	Number of changes	Yes	No
10a	Cross experiment contamination	No	Yes
11a	Intermediate conclusive results	Yes	Yes
11b	Day of week effect	Yes	No
11c	Early stopping	No	No
12a	Post experiment health checks	No	No
12b	Confirm winning variant	Yes	No
12c	Post experiment checking of guardrail metrics	No	No
13a	Encourage more experiments	Yes	No
14a	Grouping of experiments	No	No
14b	Higher level question	Yes	No
15a	Share learnings	Yes	No
17a	Data quality issues	No	No
18a	Novelty effects	No	No
19a	Skewed data	No	No
20a	Rerun experiment when results are marginal	Yes	No
20b	Validation of experiment	Yes	No

Table 3.1: table of all the pitfalls . The ids are the ids as they originally appear in the work of Mulders [40]

Cause When creating an experiment, it is of great importance to know what the goal of the experiment is going to be. One important part of determining that goal is the hypothesis that describes what the experiment is trying to prove.

Result If a hypothesis cannot be proven wrong, there will be no experiment necessary, as the result is already known.

3.2 Direction of change in hypothesis

Issue It is unclear if the experimenter is looking for any change (better or worse) or only wants to know if the metric moves in one direction (only better). An example of such a correct hypothesis is: ”When removing the sign up, we expect signups to go down”. In this case the metric measured is the number of signups and the direction is that this number will decrease.

Cause Statistically, the maths between calculating significance and calculating the possibility of a significant change in one direction is different than calculating the possibility of a significant change in one of two directions. Therefore, by forcing the experimenter to choose, we can better manage this. Another advantage of forcing the experimenter to do this, is that they are forced to think about what the result of the experiment should be [28].

Result If the experimenter does not specify what type of statistical test they want to do, this could result in an invalid experiment and therefore wrong results.

3.3 Guardrail metrics

Issue When running experiments, it is best practice to have a set of metrics setup which are made sure to not change significantly negatively [34, 14]

Cause OCEs can be seen as a playground. A safe place to experiment with new ideas, without facing repercussions when things go wrong. However, just like a parent in a playground, you want to keep an eye out, just in case something dangerous happens. A similar thing occurs with OCEs. When running experiments and following the best practice of focussing on only one metric [30, 14], it is best practice to have some metrics being tracked in the background to make sure nothing is going wrong. The most simple thing to look out for is if an experiment crashes the entire website (see section 3.11). However, there might be other metrics the company wants to measure. For example, it might not matter that more people click a link, if that amounts to people overall creating less revenue. Other metrics which could be tracked by the company are long term goals it is striving for [19]. The collection of these metrics is called guardrail metrics.

Problem If no guardrail metrics are setup, some experimenter might inadvertently ruin the reputation of the entire company or damage the company in another way while focussing on that metric

3.4 Technical debt

Issue Technical debt is the implied cost of additional rework in the future by choosing an easy (limited) solution now instead of taking a better approach that would take longer [47].

Cause One thing that is likely is to happen is leftover code of past experiments being in the code base [44].

Problem This should be removed from the code base to reduce technical debt [19], as these 'shortcuts' can hinder future progress. Therefore making it take longer to implement new features.

3.5 Competitor safety

Issue Running OCEs in production might give a competitor an indication of a new product (line) being developed

Cause Testing product ideas in production causes customers and possibly competitors to become aware of these ideas

Problem This might give competitors more time to catch up or beat time to market [19]. This situation is unfavorable and therefore should be thought about when starting an experiment.

3.6 Churning users

Issue Churning in this context explains the situation that the software is degraded in such a way that users leave the product.

Cause Although it is great to test every idea, experimenters need to always keep the final experience for the users of the platform in mind.

Problem If, for example, an OCE gives customers a bad onboarding experience, they might not choose to join after all, making the organization lose a customer [19].

3.7 Minimum effect size

Issue The minimum effect size is part of the calculation of how long the experiment should run.

Cause The minimum effect size enables one to determine, with statistics, how many users are needed to test with to measure a 5%, 10% or 100% change [19]. After all, it is easier to see if a new version is 100% better than 1%, as the cause of a 1% change is more likely randomness if there is a small number of participants.

Problem If this is not calculated, the experiment could run longer than necessary or too short to get meaningful results.

3.8 Minimum duration

Issue The minimum duration is how long an experiment should at least run to gain significance.

Cause A lot of other pitfalls fall under this main pitfall [14]. Novelty effects, which is the effect of the interaction with a new feature being higher, because the feature is new and exciting [19]. After some time, this newness fades and actual results can be drawn. The reverse can also happen. Changing the layout of a webpage, for example, might decrease the speed and interaction at the beginning, but might perform better in the long run. This effect is called the primacy effect. Another pitfall related to the minimum duration is stopping the experiment when significance has been achieved [36]. Running the experiment longer might result in the feature not being significant after all, as the significance can change over time. [19]

Problem If this is not calculated, the experiment could run longer than necessary or too short to get actual results.

3.9 Withholding results

Issue Withholding results from their experiments from experimentersmight will result in the experimenter drawing better conclusions.

Cause If intermediate results are visible for experimenters and variant A starts out as better than variant B, this might lead to ”rooting” for this version to win.

Problem This makes the experimenter biased towards a solution. This should be avoided as much as possible.

3.10 Simultaneous experiments

Issue Experiments which influence each other.

Cause Experiments running at the same time might influence each other. The most extreme example is when one test tests a different color for a button, while another test removes the button.

Problem If the participants are not well distributed, this could lead to unwanted consequences. Furthermore, if users are being exposed to multiple AB-tests at once, this could lead to invalid results [31].

	Color A	Color B	Total
Shown	10	30	40
Hidden	30	10	40
Total	40	40	

Table 3.2: Table showing an example of simulatenous experiments

In Table 3.2, both experiments and variants have an equal number of participants. However, for 75% of the participants the variant with Color A is going to be hidden, therefore not generating any interaction.

3.11 Failure checks

Issue An OCE crashing or otherwise not performing correctly could break the running software.

Cause Sometimes an implementation of a new feature will make the application crash. Failure checks then make sure to shut the experiment down when this happens. [19, 28]

Problem Crashing software brings possible downtime and therefore possible loss of customers and revenue.

3.12 Number of changes

Issue The increasing complexity of experimenting with more than one change at once.

Cause Usually an experiment consists of a version A and B. However sometimes an experimenter might want to experiment with multiple changes at the same time. For example, by giving every button on a page a specific color. However, these tests are usually more difficult and should be double checked if there is the possibility to see if the experiment can be done in the A/B fashion, [13] as this takes less time and effort.

Problem This increased complexity increases the chance that statistical tests or something else in the experiment goes wrong

3.13 Day of week effect

Issue Not running an experiment for a limited amount of time.

Cause The audience of software application can differ on time and day. On weekends different segments might visit the website [30, 53]. To represent the actual population of the website, it is important to run experiments for full weeks. This should even occur when the experiment becomes significant quickly, as running the experiment for the full week might result in the results not being significant.

Problem If one section of the users is not correctly taken into account, this will give an incomplete result and possibly result in the wrong conclusion.

3.14 Not confirming the winning variant

Issue Not checking the version put into production.

Cause After an experiment has concluded, the winning version is usually put in production and the experiment cleaned up.

Problem When an experimenter does not make sure the winning version of the experiment works as expected, this could result in a broken version of the feature being put into production.

3.15 Encourage more experiments

Issue Not asking for new experiment ideas at the right time.

Cause An experiment has the goal to see if a hypothesis is true. However, while examining/discussing the results interesting insights or ideas for new experiments can be discovered

Problem By not providing the space for an experimenter, the business might miss a great new idea.

3.16 Higher level question

Issue Not being able to see the larger picture.

Cause Experiments are not run in a vacuum. Most of the time they are used to answer a higher business level question [19].

Problem If no infrastructure in place to think about these questions, it might be the case that the product ends up in a local optimum. It is therefore good to be able to group these experiments, so that this relation is clear to other users.

3.17 Share learnings

Issue Not being able to share learnings of experiments

Cause One factor important when becoming a data driven organization is to share the learnings from experiments. [19, 17, 35]. These can range from how to setup the best experiment to sharing the actual learnings from the experiment and having a repository of all experiments ever run [28]. From the interviews in chapter section 2.3, we also found that experimenters really feel a need for both.

Problem Experimenters will make the same mistake or miss out on great new product ideas

3.18 Rerun experiment when results are marginal

Issue If the difference (significance/results) between two versions is low, rerunning the experiment could shed more light on which version performs better.

Cause If the difference between two versions is small, i.e. if the metrics do change, but only by a little bit, then the experiment should be rerun again to validate that the results were correct and not the result of chance. [19]

Problem Only barely significant results might be because of chance, not of actual difference.

3.19 Validation of experiment

Issue Not validating the experiment

Cause Reproducibility is one the most important factors of science. If rerunning an experiment results in a different outcome, the setup was incorrect, it might be by chance or there are other influences which were not accounted for. Therefore, experiments which ran on the platform should be reproducible [19] and some experiments should be rerun on a certain timescale.

Problem If experiments are not rerun to check the validity of experiments, this could lead to wrong experiments being performed.

Chapter 4

Solutions to pitfalls

In this chapter, we will analyze the problem and propose solutions to answer the second part of research question 2: "What are key pitfalls and their potential solutions?" (1.4).

4.1 Existing AB-test frameworks

To create solutions for the pitfalls it would be of help to look at other existing frameworks to see if any other platform has already created a solution to this problem and if so how they do this. We mainly look at other commercial products, as they have public documentation and online demos available. For most internal tooling, Microsoft's platform Exp [23] for Example, no screenshots or documentation could be found and therefore are not mentioned here.

4.1.1 Optimizely

Optimizely [53] is a popular AB-testing framework for performing experiments on websites. Optimizely claims that 24 of the top Fortune 100 companies are using Optimizely to perform tests. It uses Javascript to change the content of a page on the fly[42]. This enables it to be deployed quickly to any website, however Javascript has some downsides that should be discussed.

The first downside of using Javascript is that it takes time to load in the code to change it. As the content of the page usually renders earlier than the code that changes it, the user first sees the original page for a split second before it is changed. This is perceived as a flicker and therefore could influence the result of the experiment.

The second downside of using Javascript is that the code still needs to be loaded in, therefore making the page load slower. Third, this approach might not work on some browsers, as some features which this code needs might not be supported.

4.1.2 VWO

VWO [4] is another popular choice in the world of AB-testing, being used by Ubisoft and others. It uses the same Javascript-injecting method of creating variants. It is interesting to

note that none of the pitfalls we research are adressed in this tool.

4.1.3 Adobe Target

Adobe Target [8] is an Adobe Product focussed on online experimentation. Its documentation is extensive and clear about 4 of the pitfalls we are researching (Falsifiable hypothesis, Direction of change in hypothesis, minimum effect size and minimum duration). This gives the impression that the tool has been made with best practices in mind.

4.1.4 Google

From Google, we will look into 2 tools which enable experimenters to experiment: Firebase and Google Optimize.

Firebase

Firebase is a tool to "help mobile and web app teams succeed" [1]. It offers a lot of tools and services to help teams more easily and quickly create apps. One of the services they offer is A/B Testing. Firebase is the only website to offer a demo, which enabled to get a better understanding on how their features can solve our list of pitfalls.

Google Optimize

Google Optimize is part of Google's marketing platform to help engage customers "like never before" [2]. Although it is a separate product, many of the UX/UI elements seems very similar to Firebase.

4.2 Solutions

Given our overview the problems developers face and of the existing solutions, we can start proposing solutions to these problems. As some solutions solve multiple problems at once, each solution starts with a list of pitfalls it solves. After this, we use the existing solutions as inspiration to, finally, propose our solution to the pitfall.

Pitfall	Name of solution
Falsifiable hypothesis (01b, 3.1)	Enforce cross checks
Direction of change in hypothesis (02b, 3.2)	Enforce cross checks
Guardrail metrics (02c, 3.3)	Add guardrail metrics
Technical debt (03a, 3.4)	Add checklist
Competitor safety (03b, 3.5)	Add checklist
Churning users (03c, 3.6)	Add checklist
Minimum effect size (04a, 3.7)	Enforce the correct experiment duration
Minimum duration (04b, 3.8)	Enforce the correct experiment duration

Pitfall	Name of solution
Withholding results (04c, 3.9)	Blur screen of the experiment
Simultaneous experiments (05a, 3.10)	Add checklist
Failure checks (06a, 3.11)	Add guardrail metrics
Number of changes (09a, 3.12)	Enforce cross checks
Day of week effect (11b, 3.13)	Enforce that experiments are run for full weeks
Confirm winning variant (12b, 3.14)	Add review step in experiment office
Encourage more experiments (13a, 3.15)	Add review step in experiment office
Higher level question (14b, 3.16)	Add review step in experiment office
Share learnings (15a, 3.17)	Add review step in experiment office
Rerun experiment when results are marginal (20a, 3.18)	Add button to rerun experiment
Validation of experiment (20b, 3.19)	Add button to rerun experiment

Table 4.1: Table showing the pitfalls and which solution is created to solve the pitfall. The first number behind the pitfall is the id of the pitfall, the second the section in which the pitfall is explained

4.2.1 Enforce cross checks

Resolves:

- Falsifiable hypothesis (1b)
- Direction of change in hypothesis (2b)
- Number of changes (9a)

Making sure the setup of the experiment is correct takes two steps, creating the setup and checking the validity of the setup. The Experiment Office has already taken great steps to help with the first part; By providing a template Figure 4.1 it is much easier to create a good hypothesis. However, hypotheses can still be made better and especially the direction of change is easy to forget, as "the number changing" completes the sentence, but is not valid. To solve this problem, it is proposed to have experimenters review eachothers hypothesis, similarly as is being done at Booking.com [28]. This review process can filter out many wrong or suboptimal hypotheses experimenters create.

Therefore we propose the following solution: We solve this problem by having other people check the experimental setup. Before your experiment can start, it is looked over by a group of experts, who could find mistakes in the experimental setup or tips on how to improve the experiment. The experiment is not allowed to start before the ok is given.

4.2.2 Add guardrail metrics

Resolves:

- Guardrail metrics (2c)

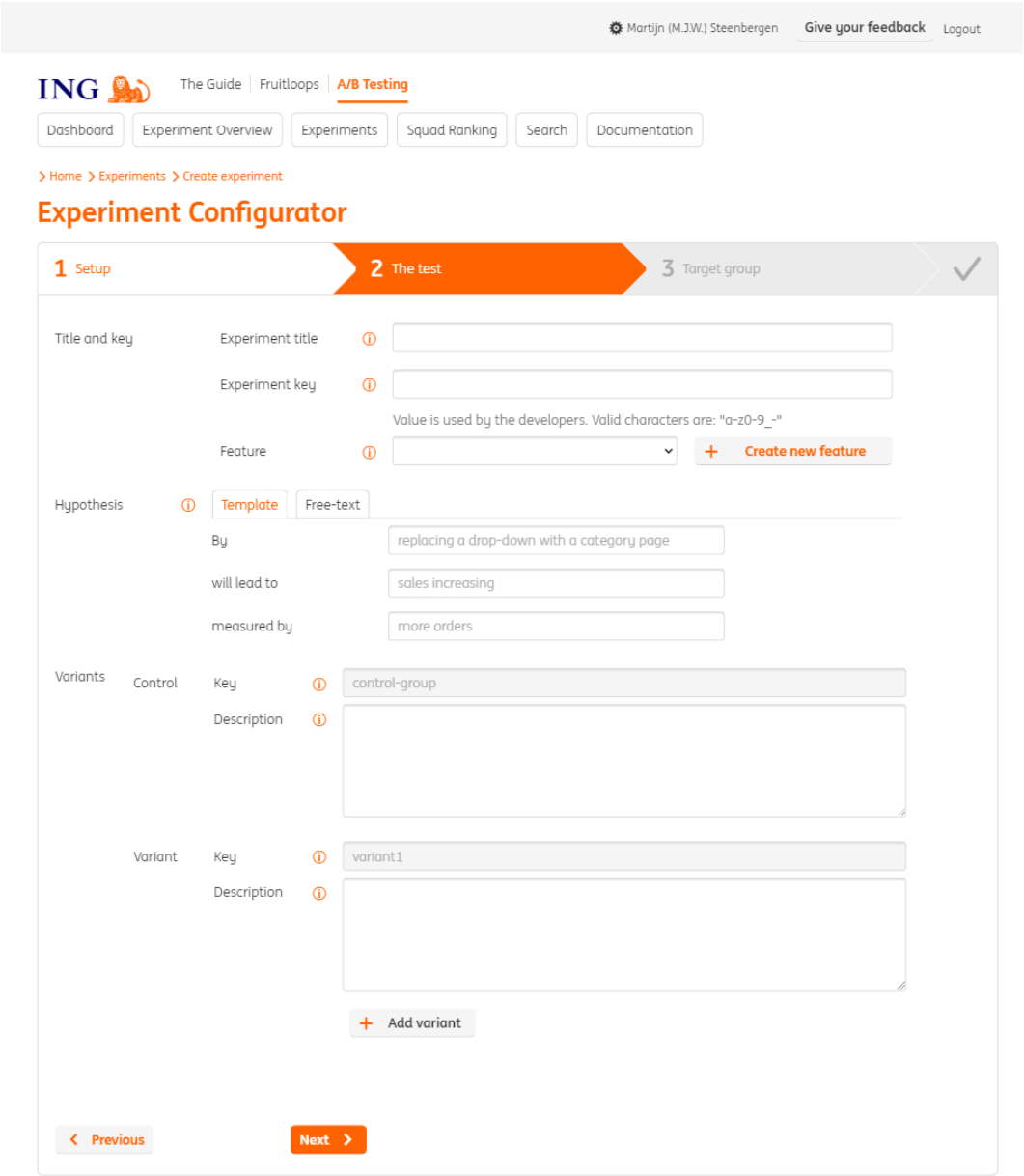


Figure 4.1: Screenshot of the second step of the experiment creation workflow

- Failure checks (6a)

This solution is heavily based on the solution from Firebase (Google) in Figure 4.2, as it offers a very nice way of giving an overview of different metrics at the same time. Similarly, Airbnb [37] has a view (in Figure 4.3) to look at metrics which maybe less clear, but enables the experimenter to dive deeper into the data.

The view from Firebase should be tried first, as it has a lot of similarities to a study Meeuwsen et al. performed at ING [39]. Meeuwsen held a focus group with participants at ING. This resulted in the following sketch, seen in Figure 4.4, made by experimenters at ING on what a possible future version of the Experiment Office could look like.

This results in the following solution in the end:

To resolve this problem, we add the possibility for adding guardrail metrics and have the experimenter choose which metrics they also want to measure besides the default metric.

4.2.3 Add checklist

Resolves:

- Technical debt (3a)
- Competitor safety (3b)
- Churning users (3c)

Solutions to the above problems could not be found in existing solutions. One possible reason for this could be that these problems are not so much a constant problem, but do need attention once in the entire workflow. Therefore it seeems that a simple checklist could be sufficient to resolve these pitfalls.

Therefore we propose the following solution: To solve this, we show a quick list of things that you should think about when running experiments, we remind you of things that can go wrong while experimenting

4.2.4 Enforce the correct experiment duration

Resolves:

- Minimum effect size (4a)
- Calculate minimum duration (4b)

To solve this pitfall, data is needed, like the amount of visiters to a page and the size of the change they want to measure. Many online sample size calculators already exist which can help.

The solution from Optimizely is shown in Figure 4.5. It is able to determine the population size needed. The solution from Adobe Target in Figure 4.6 also calculates the number of days an experiment should be running. Therefore it seems this solution is best. Interestingly, both solutions are standalone and not integrated, something that could be a feature which the Experiment Office could integrate.

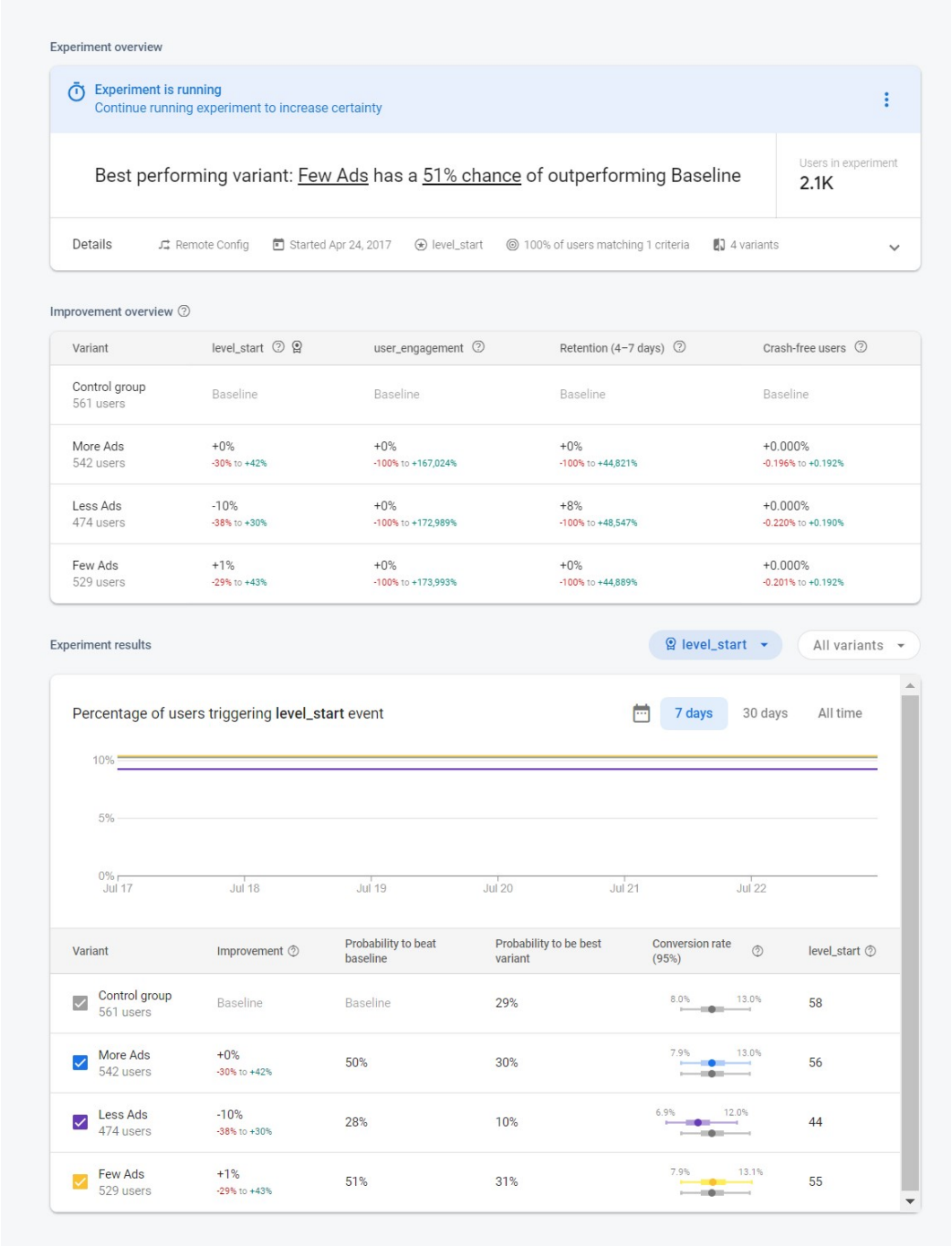


Figure 4.2: Screenshot of the Google Firebase website. Screenshot taken from `https://console.firebase.google.com/u/0/project/fir-demo-project/config/experiment/results/3?hl=en`

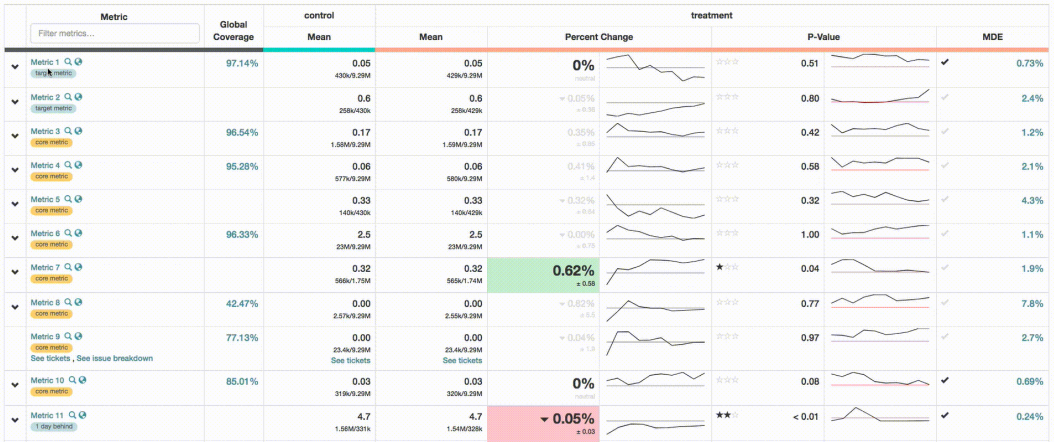


Figure 4.3: Screenshot of the metrics dashboard from Airbnb

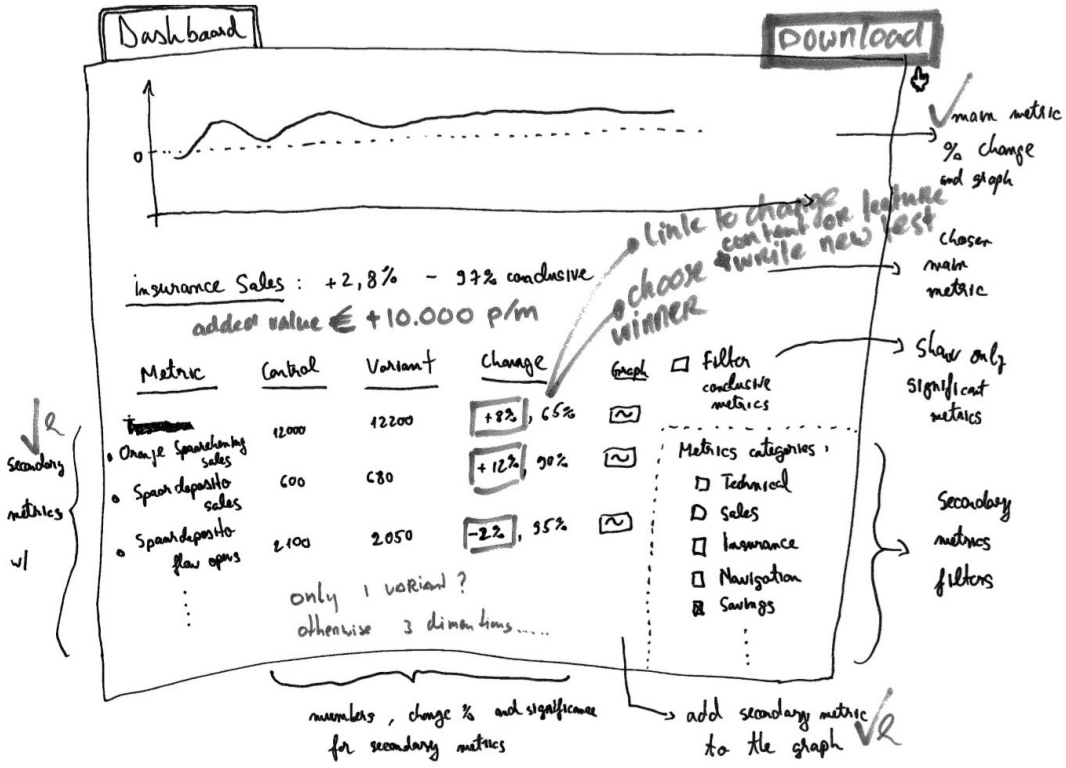


Figure 4.4: Ideal dashboard, brainstormed by ING engineers during a focus-group performed by Meeuwsen et al.

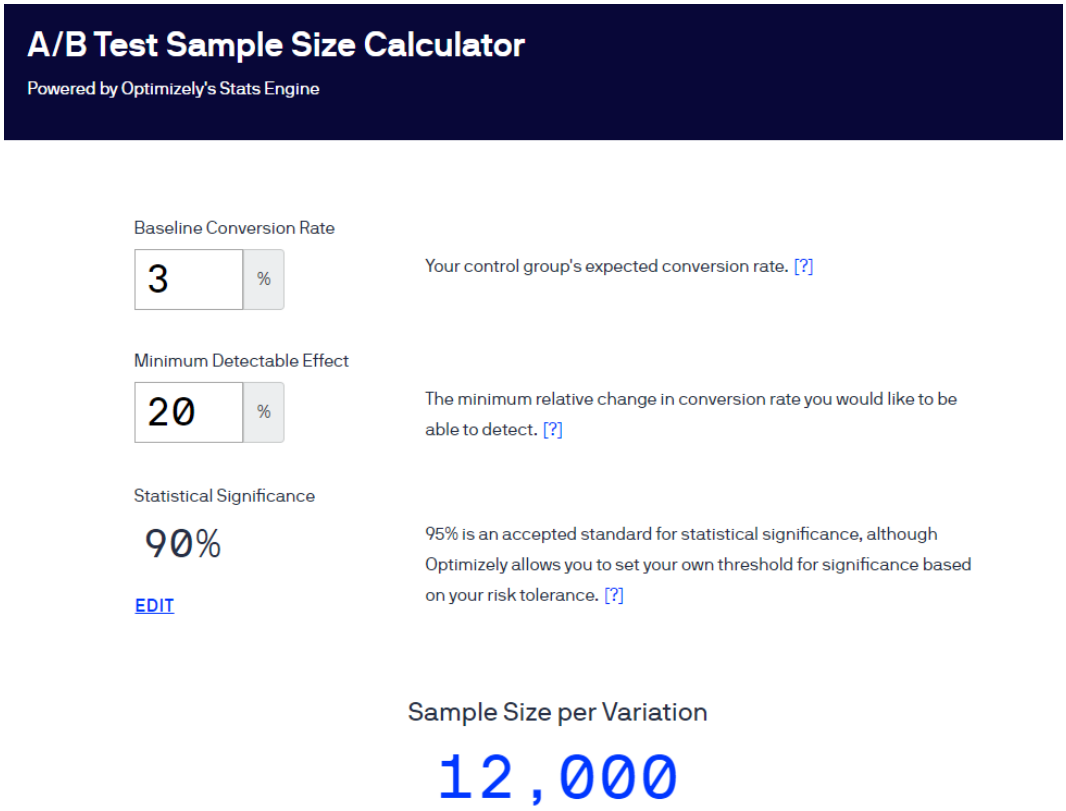


Figure 4.5: Sample size calculator from Optimizely

The solution from Google, seen in Figure 4.7, is different, as they do not require an end date and just wait for the experiment to gain significance. However this generally considered a bad thing, as every experiment will become significant in the end, however the results will be negligible. Furthermore, from the interviews in section 2.3 we learned that currently only one experiment can be done on a page, so this solution is not chosen.

Therefore we propose the following solution: To aid people with setting the correct duration of the experiment, we introduce a screen where users are able to fill in all the information they have. How many people visit the page that is being experimented on, how detailed the result must be, etc. This will then automatically determine how many users need to participate in the experiment and therefore how long the experiment should last.

4.2.5 Blur screen of the experiment

Resolves:

- Witholding results (4c)

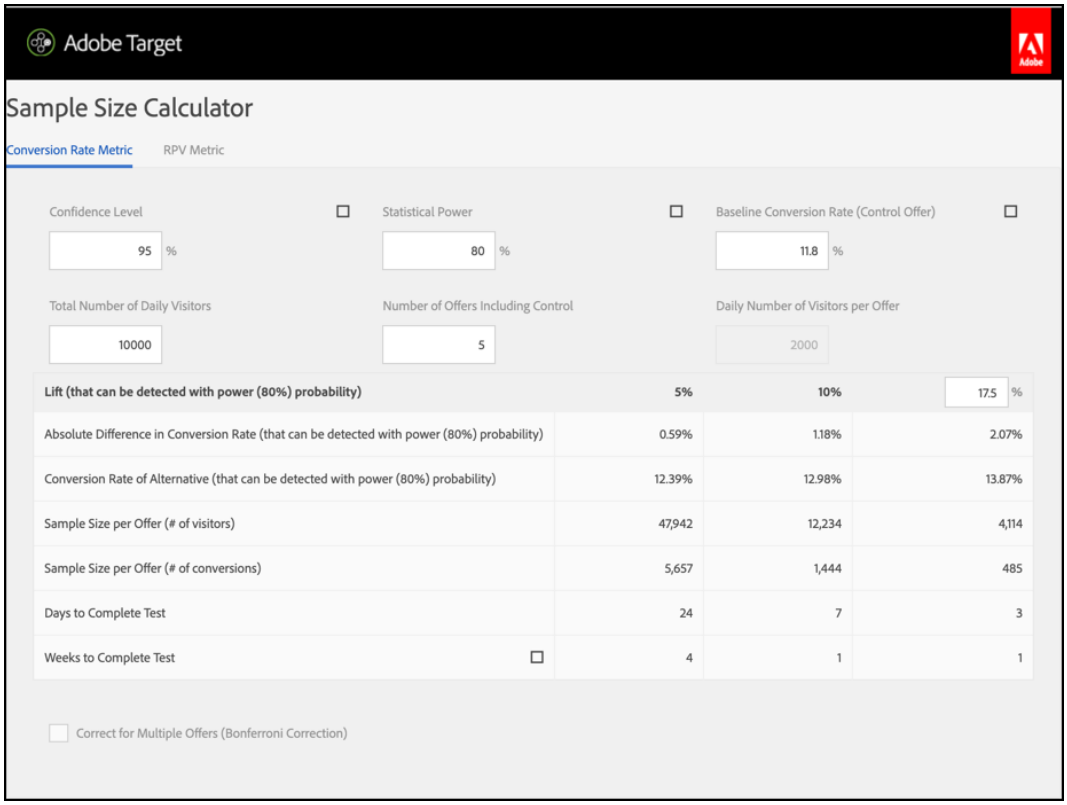


Figure 4.6: Sample size calculator from Adobe Target

To make sure experimenters do not become biased towards a solution, we can blur the results of the experiments. This solution is not found in any examples of other experiment platforms, as this (obviously) also hides the result of the experiment. To give the experimenter the ability to check if the experiment is running as expected, the blur can be removed.

If this solution is not chosen, a less drastic solution, which partially could solve the problem could be derived from Google's Firebase implementation (see Figure 4.2). By always displaying the chance that a variant performs better next to the improvement of the variant, it increases nuance and reduces the chance of people becoming biased to a variant. After all, saying that "a variant has a 100% better performance" gives a different feeling than "a variant has a 100% better performance, but there is a 52% it actually performs this well".

To conclude we propose the following solution: To resolve this problem, we can blur screen until the required number of user and significance has been reached. To aid users, we unblur the screen after users click yes on a pop-up where they are explained why looking at results would be a bad idea

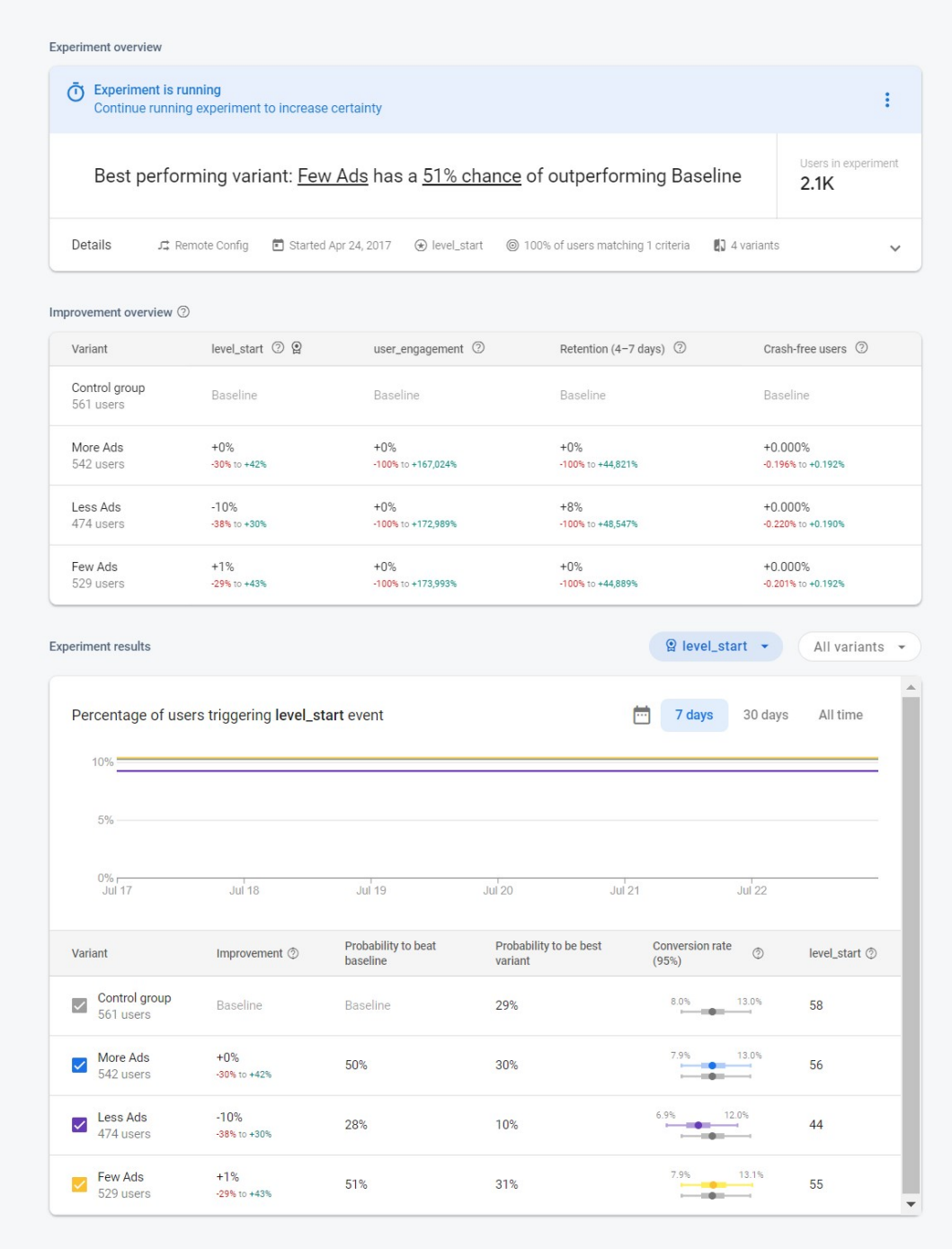


Figure 4.7: Screenshot of the Google Firebase website. Screenshot taken from `https://console.firebase.google.com/u/0/project/fir-demo-project/config/experiment/results/3?hl=en`

4.2.6 Enforce that experiments are run for full weeks

Resolves:

- Day of week effect (11b)

Both Optimizely [53] and Google Optimize [3] mention the "cyclical variations in web traffic"[3]. However, both do not seem to either force experiments to be full weeks or nudge experimenters to run their experiments for full weeks.

Therefore we propose our own solution: To resolve this problem, we present an experimenter with a warning when they want to stop an experiment on another weekday than the weekday the experiment was started. We add the options to let the experiment continue to the original end date, the next full week or stop it immediately.

4.2.7 Add review step in Experiment Office

Resolves:

- Confirm winning variant (12b)
- Encourage more experiments (13a)
- Higher level question (14b)
- Share learning (15a)

Optimizely has an extensive guide on how to convey the results of experiments to other teams mention the following basic items that should be in a report when conveying experiment results, copied directly from the webpage [41] :

- **Purpose**: Provide a brief description of "why" you're running this test, including your experiment hypothesis.
- **Details**: Include the number of variations, a brief description of the differences, the dates when the test was run, the total visitor count, and the visitor count by variation.
- **Results**: Be concrete. Provide the percentage lift or loss, compared to the original, conversion rates by variation, and the statistical significance or difference interval.
- **Lessons Learned**: This is your chance to share your interpretation of what the numbers mean, and key insights generated from the data. The most important part of results sharing is telling a story that influences the decisions your company makes and generating new questions for future testing.
- **Revenue Impact**: Whenever possible, quantify the value of a given percentage lift with year-over-year projected revenue impact.

This gives us a good overview of areas that need effort to keep track of the learnings of experiments (15a) and trigger experimenters to think of new experiments (13a).

To make sure the winning variant is working as expected, we can ask the experimenter to check the page after they cleanup the code to create the experiment. Inspiration to improve this conformation can be found with Optimizely on the page to preview variants in Figure 4.8. When creating experiments in Optimizely, there is an option to preview the

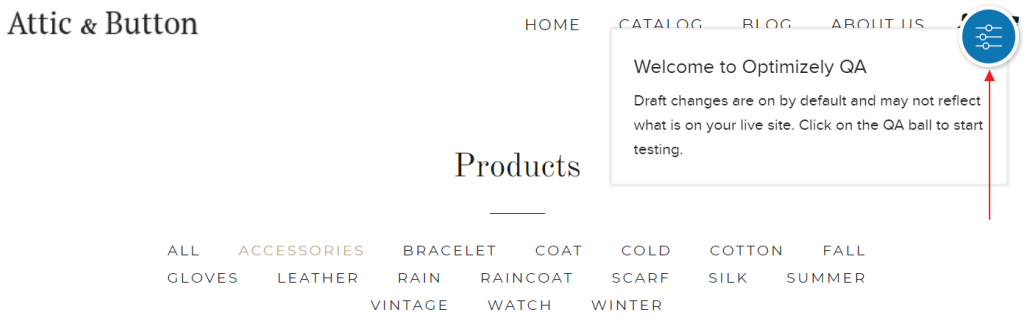


Figure 4.8: Preview of the experiment given by Optimizely

change inside the editor. Being able to preview a webpage in the tool can be very helpful and save the experimenter some time.

To make sure people keep in mind the higher level goals, they can be displayed in the list of metrics, like is proposed with guardrail metrics in subsection 4.2.2. This will help, but by explicitly asking experimenters about it we force them to think about it each time an experiment is wrapped up. It is most likely that when a the best variant to the experiment moves the team towards the goal there is not much to say here (although this should be further researched), but when a chosen version moves the team away from a goal, it might be useful to evaluate on this. (Is this the better solution? Is perhaps the goal wrong?).

Therefore we propose the following solution: After the experiment has concluded, we ask the experimenter to respond to a couple of questions related to the experiment they ran. What was their conclusion? What did they learn about experiments in general? What will be their followup experiments? How does this relate to their overall goal? Did they clean up their experiment and check if it was working as expected?

4.2.8 Add button to rerun experiment

Resolves:

- Rerun experiment when results are marginal (20a)
- Validation of experiments (20b)

No existing solutions for these problems exist in our survey of the tools. Therefore we propose our own solution: A button to rerun the experiment is added, which the experimenter can press if he/she would like to rerun their experiment to double check the results. This button is made more prominent if the significance of the experiment was low or if the experiment is sampled to check if the experiments are still running correctly

Chapter 5

Research methods used

In previous chapters we have determined pitfalls are a problem to experimentation and that 19 out of the 35 known pitfalls are relevant to this research. Next we seek to answer three questions. First, do experimenters indeed experience these pitfalls? Secondly, do they believe the new solution solves the problem? And lastly, are they hindered by the introduction of the solution? To answer these questions, we need to contact the experimenters. By creating a survey and sending this out to experimenters using the platform, we can find this information.

Surveys are chosen over interviews, as this results in a larger population to get more information from. Based on the preliminary interviews, a sufficient overview exists of the workflows at ING. If the results of the survey require follow-up, extra interviews/conversations can always be planned with specific individuals.

To perform the survey we will use Survalyzer[45], an internal ING tool, which respondents will be familiar with.

5.1 Survey contents

5.1.1 Introduction questions

As with any survey, a baseline is needed to understand the participant and get their permission to use the data from the survey for this master thesis.

We asked the participant for their email, in case the response raised questions we wanted to followup on, and the time the participant has been experimenting, to be able to notice any influence this has on the type of answers given.

5.1.2 Pitfall page

There exist 19 pitfalls we want to explore. For each of these pitfalls, a page exists within the survey. Every page contains 2 sections, where each section tries to answer a part of research question 2. Every question has been designed to not bias the participant and be clear to increase the reliability of measures [50]. Furthermore, we tried to make sure each question gives us as much information in as little time for the respondent. The last requirement

is achieved by having only 2 textboxes, one to collect information about the solution we propose and another as an optional catch all for any thoughts the participants might have, for example that they think the pitfall actually is not a problem. Both are optional, so that if the participant does not have enough time, they can skip it, as we would rather have that they give their opinion on more pitfalls, than explain one in great detail.

The first section of the page devoted to a pitfall explores the experimenter's view on the pitfall and tries to answer the first part of RQ2, namely how the experimenter experiences the pitfall. We want to know the prevalence of a pitfall, asking if they experienced it with a Yes/No question. Similarly, we ask about the prevalance of the knowledge of the pitfall. The perceived severity of the pitfall is gathered by using a Likert scale.

The second section details the proposed solution discussed in section 4.2 and tries to answer the second part of RQ2, how the experimenter experiences the solution. By asking participants if the solution solves the pitfall (Yes/No) and if it would help or hinder the participant with experimentation (again with in a Likert scale), we get an indication of the feasibility of the solution, which is then strengthened by the open text question of "Why?". This question is optional.

As discussed before, we wrap up with a catch all textbox, where we ask the participants if they have any other thoughts about the pitfall. If they do not have anything to add, they can skip this question.

Every page concerning pitfalls will look the same, except the description and solution of the pitfall. The contents of these can be found in Appendix C, together with screenshots of the survey.

5.1.3 Incorporating real-world examples

We would like to get as detailed responses as possible. If we can find out instances where an experimenter has fallen into a pitfall or encountered one, we can ask them detailed questions about this instance and their experience.

Only data for the day of week pitfall (3.13) can be extracted from the data of the experiment platform. This is accomplished by simply looking at the start and end date of an experiment and seeing if they are the same weekday. For the rest of the pitfalls, unfortunately, no instances can be extracted. This has a simple reasons. To solve pitfalls solutions need to be made, which do not exist yet. Only then will it become possible to gather which experiments have fallen for the pitfall. Secondly, metrics are currently not integrated in the platform, making tracking anything not already in the platform next to impossible.

Once collected, Survalyzer[45], the survey tool, has the ability to show participants links to only the experiments they performed, to make maximum impact.

5.2 Participation

We aimed for participants to complete at least 5 pitfalls. Each pitfall-part consists of 7 questions with the two initial questions. This totals to 35 questions. This number is large enough to gather information, but small enough that people are more likely to respond to

the survey, as it won't take forever. If, after this, participants want to give feedback on more pitfalls, we give them the opportunity to do so.

The ordering of the questions is as follows. If we have evidence that a participant has fallen for day-of-week pitfall, we start with that, with a personalized survey-section. Once they have completed it, we randomize the remaining pitfalls, as there is no pitfall we deem more important to get feedback on. This means that in the end, all pitfalls should have around the same number of participants.

Creating a list of participants turned out to be more difficult than expected. In the Experiment Office, the focus is put on the experiment and not the user who is executing the experiment. Because of this, there is no way to find out which user is executing a specific experiment (or any experiment). With help from the squad that maintains the Experiment Office, a list of active experimenters, who used the Experiment Office in the last 3 months, was extracted from log files. This list was further expanded with extra information. Each experimenter needs to be a part of a team to run experiments (2.3). These teams can be extracted. For each of these teams, if they created an experiment in the past 2 years, an effort was made to find the corresponding squad in the organization. To be a good citizen of ING, people in squads, for whom it was deemed very unlikely that they would have ever performed an experiment were excluded from participation. All the others were asked to participate in the survey. This resulted in 129 possible participants.

Chapter 6

Results

The survey ran from the 10th of June until the 22nd of June. In total 52 people responded (which is a response rate of 40.3%). Collectively, they gave feedback on 305 pitfalls. This chapter discusses the result of this survey. First in section 6.1, the overview of the results will be discussed, after which in section 6.3 (pitfalls) and section 6.4 (solutions) the results will be more deeply discussed and as a result, research question 2, "How are pitfalls and their solutions experienced?".

6.1 Overview of results

The distribution of answers, which can be seen in Table 6.1 seems to be evenly distributed. Due to the fact that the 'Full-week effects' pitfall was only shown to people who experienced this, this pitfall only received 5 responses. This is a result of both the response rate for the generic pitfalls being higher than expected and the response rate from the people, who experienced this pitfall being lower than expected. The experience with online controlled experimentation can be seen in Figure 6.1. On average participants had 3.67 years of experience.

Before diving deeper into the results, a few generic comments can already be made. For all the pitfalls, there are more people familiar with this pitfall than those who have actually experienced it, indicating that experimenters are proactively thinking about things which can go wrong when experimenting. There are six people, who said they experienced a pitfall, without being aware it existed. In two cases, this could have been as the participants did not think it was an actual problem. Others did not elaborate on this. For only the day of week effect this ratio does not hold, as we showed only people this question, for whom we were sure experienced this pitfall.

People on average think the new features are helpful, giving any feature an average of 3.52 (out of 5). This is because the entire population rates higher, with one outlier saying 9 solutions are 'very helpful' and only one being 'neither hindering, nor helpful'.

The following sections will look further into the results. For each of the questions a diagram was created showing the results to this question:

- Figure 6.2, which percentage has experienced a pitfall

Id	Name	Answers	Awareness	Experience	Solutions fixes problem
01b	Falsifiable hypothesis	17	58.82%	17.65%	58.82%
02b	Direction of change in hypothesis	17	58.82%	35.29%	64.71%
02c	Guardrail metrics	16	68.75%	25.00%	75.00%
03a	Technical debt	13	76.92%	76.92%	53.85%
03b	Competitor safety	16	56.25%	18.75%	50.00%
03c	Churning users	16	81.25%	43.75%	43.75%
04a	Minimum effect size	17	64.71%	35.29%	94.12%
04b	Calculate minimum duration	21	85.71%	52.38%	100.00%
04c	Witholding results	16	50.00%	25.00%	56.25%
05a	Simultaneous experiments	17	88.24%	41.18%	52.94%
06a	Failure checks	21	57.14%	23.81%	76.19%
09a	Number of changes	14	78.57%	57.14%	64.29%
11b	Day of week effect	5	60.00%	100.00%	60.00%
12b	Confirm winning variant	17	58.82%	17.65%	35.29%
13a	Encourage more experiments	18	77.78%	38.89%	61.11%
14b	Higher level question	14	57.14%	35.71%	50.00%
15a	Share learning	16	81.25%	68.75%	62.50%
20a	Rerun experiment when results are marginal	17	70.59%	47.06%	70.59%
20b	Validation of experiments	17	76.47%	29.41%	76.47%
	All	305.00	69.18%	38.69%	64.59%

Table 6.1: Table of all results

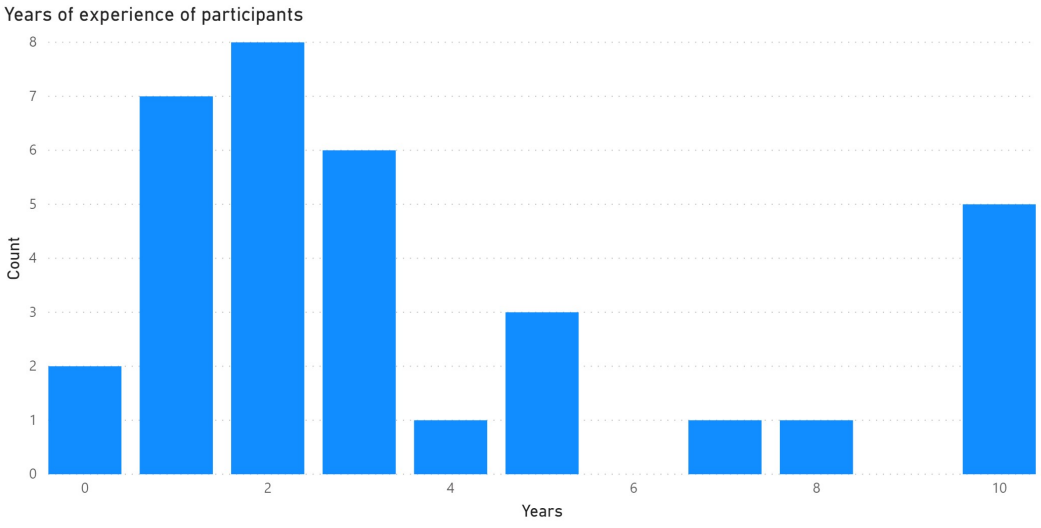


Figure 6.1: Participants years of experience

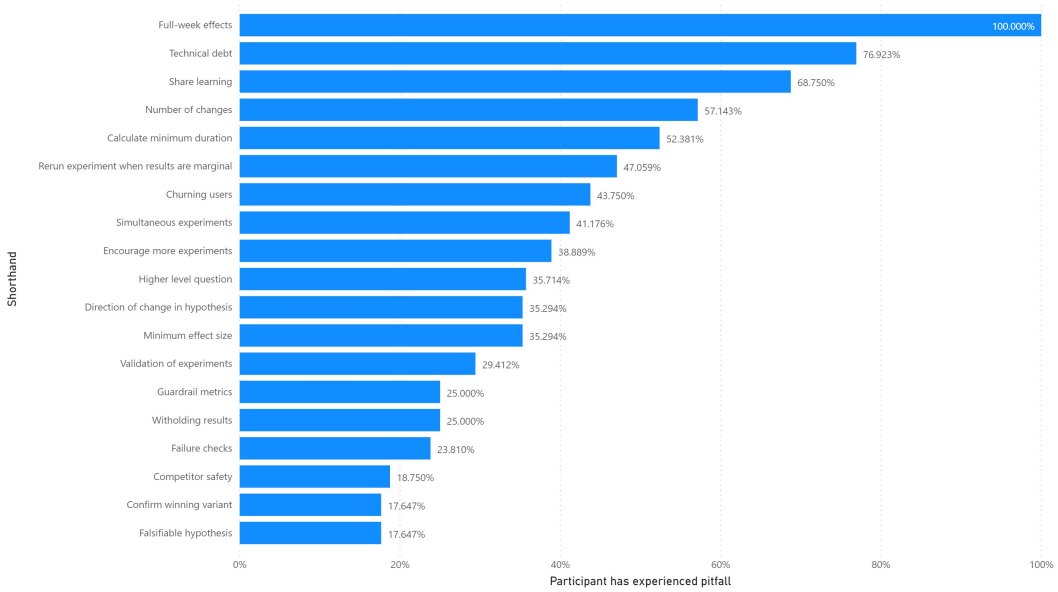


Figure 6.2: Percentage of participants who experienced the pitfall

- Figure 6.3, which percentage is aware the pitfall exists
- Figure 6.4, a divergent stacked bar chart [12] of how severe the participant considers this pitfall
- Figure 6.5, percentage of participants who believes the solution solves the pitfall
- Figure 6.6, if the participant thinks the problem helps or hinders their work

These diagrams will be used to support the rest of this section.

Following the sample size calculation, with a confidence level of 95% and a 5% margin of error, 97 people should have responded. Therefore, the results of this survey should be seen as a starting point for further exploration, but cannot be used for drawing conclusions of statistical significance.

6.2 Differences among pitfalls and their solutions

The differences between the answers to pitfalls and solutions in the survey could be the result of random chance. To determine if this was the case or if the pitfalls are truly experienced differently or if some solutions are truly better than others, we employ statistical tests, which compare the answer from each pitfall/solution to all of the others.

The data collected with the survey is ordinal data [9]. This means that the our values are categories with an order. Therefore simple averaging is not possible.

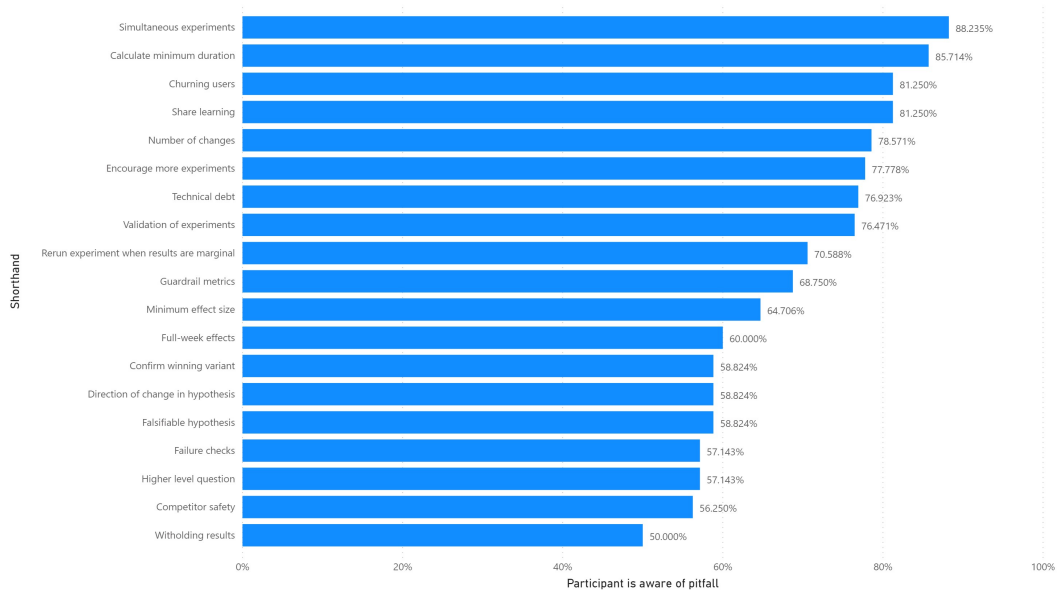


Figure 6.3: Percentage of participants who is aware of the pitfall

Figure 6.4: Divergent stacked bar chart [12] of the perceived severity of the pitfall

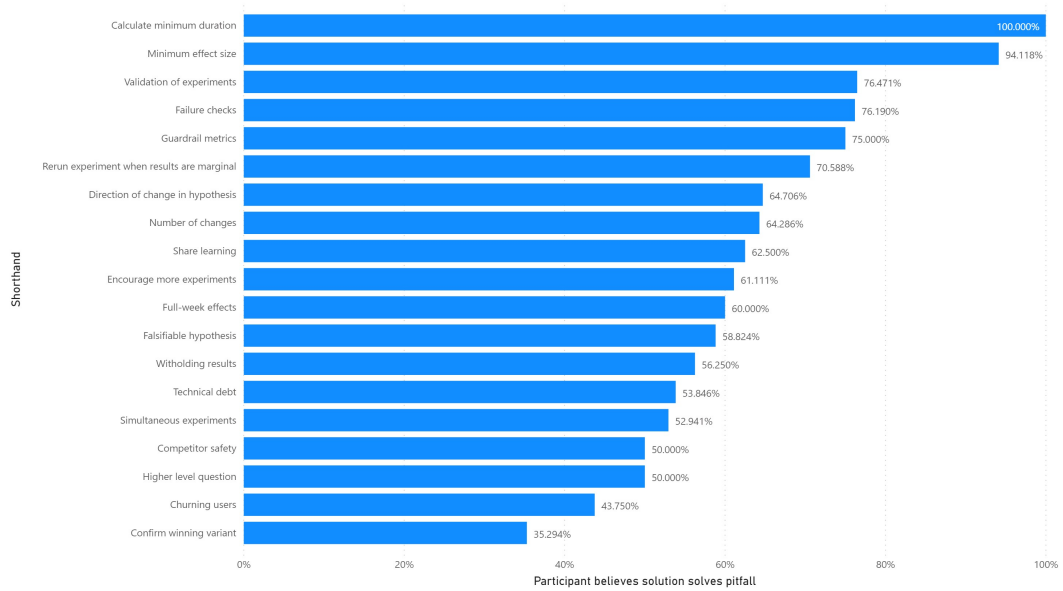


Figure 6.5: Percentage of participants who believes the solution solves the pitfall

Figure 6.6: Divergent stacked bar chart [12] of the perceived usefullness of the solution

6.2.1 Cliffs delta

First, to determine if comparing two samples even makes any sense, the effect size is calculated using Cliff's delta [48]. Four different categories exist, Negligable (< 0.11), Small (0.11 - 0.28), Medium (0.28 - 0.43) and Large (> 0.43). We will only look at results having a large effect size (> 0.43).

6.2.2 Mann-Whitney U

The Mann-Whitney U test is a test developed by Mann, Henry B. and Whitney, Donald R. to determine the probability that two samples are significantly different[32]. For the calculation, the python library `scipy` was used [7].

We will be describing the pitfalls with a large effect size and where the significance of the Mann-Whitney U test is smaller than 0.05.

6.2.3 Results

All of the data used in the calculations for this section can be found in Appendix D.

For the question if the experimenter is aware of the problem, section D.1 shows that no comparison actually results in a large effect size. Therefore none of them have both a significant outcome and large effect size.

For the experience question this is different. As we only asked experimenters the "Full-week effect" if they experienced it, this is significant and has a large effect size compared to other pitfalls (Table D.2). As we we only asked people who experienced it, we will not count it as a result. "Technical debt" is the second most significant result with 7 out of 18 pitfalls.

For severity, from Table D.2 it can be clearly seen that "Competitor Safety" is significantly not important to experimenters, as it shows a significant difference with 17 pitfalls.

For the question about solving the pitfalls, there is no real significant result. "Calculate minimum duration" has the best result and is significant with 7 other pitfalls (Table D.2).

For the HelpHinder question, there is another significant result found in Table D.10. Both "Minimum effect size" and "Calculate minimum duration" are significant with 13 other pitfalls. This is interesting for two reasons. First, this question revolves around the same solution, which makes it likely that they love this solution and secondly that this is a really great solution that will help experimenters.

6.3 Pitfall results highlights

This section will give some highlights on the awareness and perceived severity of the pitfalls. This section will answer the first partion of research question 2, namely how experimenters experience pitfalls.

- From Figure 6.2, it can be determined that "Technical debt", after "the Full weeks effects", is the most experienced pitfall. As many of participants are working with a CMS system, this is surprising. The open text does not explain this. Possible answers

could be that with a CMS technical debt still exists or that more coders than expected filled in the survey.
- It is interesting that "Falsifiable hypothesis", together with "Confirming winning variant", is the lowest experienced pitfall (Figure 6.2). Do participants think every one of their hypotheses has been falsifiable? Future work can investigate this further if this is the case.
- In Figure 6.3, even for the least known pitfall, 50% of the participants are aware of it. This indicates that knowledge about pitfalls is being shared or that they are researching themselves what can go wrong when experimenting.
- The perceived severity of "Competitor safety" is incredibly low (see Figure 6.4). Participants explain that either you should out-experiment your competitor, when running your experiment, or you are already production ready and that experiments can be used as a smoke screen to hide what you're actually working on.
- Unsurprisingly, customers leaving ING is the most perceived severe pitfall (see Figure 6.4)

6.4 Results per solution

This section explains what experimenters think of the proposed solutions and answers the second part of research question 2, how experimenters experience the solutions to the pitfalls.

6.4.1 Enforcing Cross checks

Solution: 4.2.1 - **Pitfalls:** Falsifiable hypothesis (3.1), Direction of change in hypothesis (3.2), Number of changes (3.12)

The solution seems to be very badly received. Although it is not the lowest scoring in the HelpHinder question ("Will this solution help or hinder you when running experiments?"), it is the pitfall where the most people said that the solution would hinder them. Most of the feedback in the "Why?" question focusses around the idea of having the experimenter being in control. Secondly they mention that they don't want anything delaying the running of their experiments. One participant even says there are already too many hurdles to go through. Another makes the point that experiments can be very unique to a business area and that even an expert in experimentation at ING might not understand the nuances of a particular experiment, which leads to delays in the running of an experiment.

6.4.2 Adding Guard rail metrics

Solution: 4.2.2 - **Pitfalls:** Guardrail metrics (3.3), Failure checks (3.11)

The solution seems to be well received. For both the Failure checks and Guardrail metrics pitfall, 76.2% and 75.0% think this solves the problem and most believe it will help. The textual answers are not very relevant in this case. In some cases it seems it was not clearly explained enough and the respondent is misunderstanding the solution. Other

answers agree that this is a good solution or commenting on who should choose which metrics to use; the team, defaults or a combination.

6.4.3 Adding a checklist

Solution: 4.2.3 - **Pitfalls:** Technical debt (3.4), Competitor safety (3.5), Churning users (3.6)

This particular solution is very interesting. The solution seems simple, however implementing seems to have problems. For all of the pitfalls this solution tries to solve, only 50% of the participants think it actually solves the problem. Furthermore, this solution tries to solve the most perceived severe pitfall (Churning Users) and least perceived severe pitfall (Competitor Safety) (Figure 6.4). All this results in the open-text answers to vary wildly. Some participants think that a checklist could prevent often made mistakes. Others believe that experimenters don't read, disagree on the usefulness of a generic checklist or think that they will remember anyways.

6.4.4 Enforcing the correct experiment duration

Solution: 4.2.4 - **Pitfalls:** Minimum effect size (3.7), Calculate minimum duration (3.8)

This solution was *very* well received. For the solution for the Minimum effect size almost all respondents perceived the solution to (slightly) help. Both pitfalls combined, only one participant (out of 38 who gave feedback) did not think this solution solved the pitfall. In the open text fields, many proclaim happiness that they no longer would have to use online calculators or spreadsheets to calculate effect size and experiment duration.

6.4.5 Blurring the screen of the experiment

Solution: 4.2.5 - **Pitfalls:** Witholding results (3.9)

The reactions to this solution are very mixed. Most do think that the solution solves the problem (56.25%), but how much it helps is contested. One participants writes down "Perfect solution", while others disagree, saying that this solution is annoying and unprofessional and that keeping an eye if the experiment is working correctly is necessary. One participant even goes so far to say that looking at progress gamifies the process, making him more enthusiastic to do more experiments.

6.4.6 Enforcing that experiments are run for full weeks

Solution: 4.2.6 - **Pitfalls:** Day of week effect (3.13)

This solution was not reviewed by as many participants as others, as discussed before. The response to this feature cannot easily be summarized. One unfortunate thing is that respondents do not seem to fully understand the problem and why the solution would/could work. One participant mentioning that they run their experiment for one week already, even though best practice is at least two weeks. Another mentioning something about changing the distribution afterwards. Another not understanding how stopping an experiment after

two weeks and one day could influence the results. This could and should be better explained to experimenters. One experimenter does point out that even if you counteract this effect, there will still be others that influence the result of the experiments, like the time of the month when salaries are tranferred.

6.4.7 Adding a review step in experiment office

Solution: 4.2.7 - **Pitfalls:** Not confirming the winning variant (3.14), Encourage more experiments (3.15), Higher level question (3.16), Share learning (3.17)

The reviews for this solution are overall pretty positive, although people think this solution is not very applicable to confirm the winning variant, where only 35% think this solves the problem, making this the worst solution in this area. The open text fields confirm this, as people do not seem to like the extra administration. Again, it seems that some participants did not fully recognize the problem.

6.4.8 Adding a button to rerun experiment

Solution: 4.2.8 - **Pitfalls:** Rerun experiment when results are marginal (3.18), Validation of experiments (3.19)

The response is positive, where 73.5% thinks the solution solves the problem. After "Enforce the correct experiment duration", participants think this solution will help them the most when performing experiments. Some respondents do want to nuance this to only rerun experiments when there are reasons to doubt the results, for example when the power is low. One respondent wants to use this to determine possible seasonality of changes. e.g. determine that variant A works better in the summer and B in the winter.

Chapter 7

Methodology for evaluating solutions in a real world scenario

There now are multiple solutions which can be implemented into the platform. Next answering research question 3, how one can evaluate a solution in a real-world scenario. Due to time constraints, this part could not be fully executed. However, work has been done and there are many lessons learned that are relevant, both for a future party wanting to build on this work and for ING. This chapter will elaborate on these lessons. This chapter will answer research question 3, how a solution can be evaluated in a real-world scenario.

7.1 Research methods used

In chapter 6, it was addressed which solutions experimenters prefer and which they would want to have implemented. This chapter addresses the question how such solutions cna be evaluated in practice. We aim to do this rigorously and in the spirit of continuous experimentation. This can take the form of a before/after study or a controlled experiment, where the difference is measured between the variants. This evaluation should be two pronged, which is discussed in the following sections.

7.1.1 Example

During the thesis project, work on the solution for the "Day of the week effect" (3.13) was already started. A screenshot of the solution can be found in Figure 7.1. When an experimenter fills in a date that is not a full week after the first date, it will pop up a warning to the user with more information and a button to fix it.

7.1.2 Making sure the solution works

When implementing one of the solutions, it is best practice to run an OCE to test if the solution works well. One of the most important parts is conducting the correct experiments.

In the case of the "Day of week effect", this would be how often an experimenter creates an experiment that does not run for full weeks, before and after treatment, but also how the

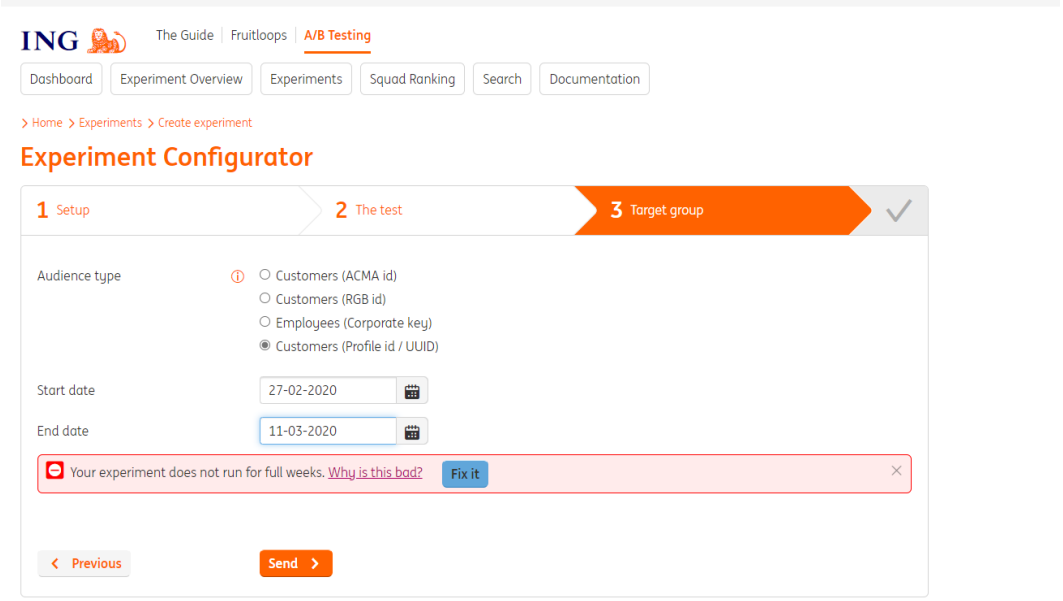


Figure 7.1: Implementation of the full week effect solution

experiment resolves the pitfall. Do they click the button? Do they manually change it? Will they look for more information (click the link)? The full tracking plan can be seen in Appendix B. For each of these pitfalls one of these plans should be made and carefully executed.

Metrics were attempted to be implemented and lessons learned from this attempt are documented in section 7.2.

7.1.3 Making sure the solution is helping the experimenters

To determine if the solution has no unintended side effects, there should be a way for users to report these to improve the solution. Furthermore, if experimenters have improvements for the solutions, they should be able to explain this to the Tetris team (2.2). In section 7.3, lessons learned from using a survey to gather this type of data is explored and it is explained why this type of survey should not be used in this context.

7.2 WebTrekk

To be able to collect metrics, a tool is needed which is able to collect store and analyze this data. As ING is already using WebTrekk, it is the obvious choice to use. WebTrekk is an online analytics platform, the name of which will be changed to Mapp in the future [33]. WebTrekk is one of the main ways ING tracks the behavior of users accessing any of the online software ING offers.

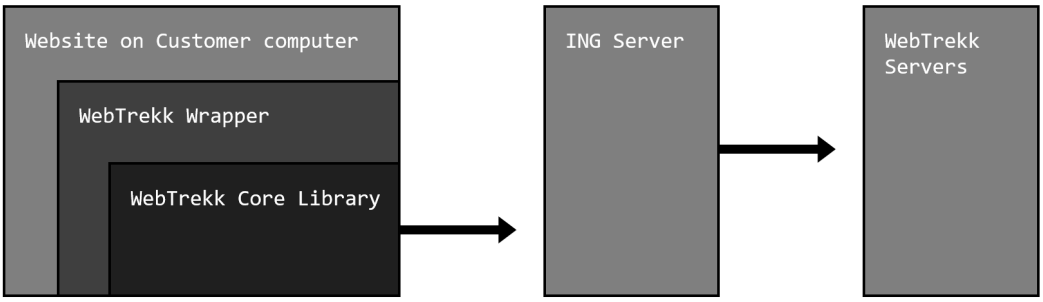


Figure 7.2: Workflow of tracking a user on the platform

7.2.1 Obstacles with implementing WebTrekk

Implementing WebTrekk resulted in two main obstacles. The combination of the web-platform of ING and the infrastructure of ING resulted in the decision to reduce the scope of the thesis.

This highlights a general pain point. To become a data-driven organization, having tracking and metrics is a minimal requirement. To encourage the adoption measurement, the barrier to entry should be as low as possible and waiting for three months is the opposite of this. To encourage tracking, a team supporting and supervising this effort should be created.

Web Platform of ING

ING is currently in the process of moving from AngularJs [20], an old framework to build web applications, to Polymer [21], a framework no longer in development made by Google to use web components on the web.

The library for working with WebTrekk has been created for Polymer and WebComponents, which Angular is not compatible with. The experiment platform, however, is written in Angular. This made it particularly hard to integrate our work. There was an alpha version available, which should integrate with any Javascript based framework. However, the error messages were nondescriptive. This lead to finding the source of the error, the infrastructure of ING, too late for the possibility of finishing this software before the end of the internship.

The infrastructure of ING

The flow of tracking a user can be seen in Figure 7.2. A customer performs an action (clicking a button, loading a page), which has an event handler, which the WebTrekk library hooks into. A WebTrekk wrapper has been incorporated into many of the Polymer elements to make integration easier. As WebTrekk is third party software, the data needs to be sent to their servers. However it cannot be sent by ING directly for two reasons.

The first reason is that WebTrekk is a service which is not hosted on ING servers or at an ING url. Therefore calling this URL within a web page breaks Cross Origin Resource Sharing (CORS) [5] rules. It is a very good security feature and should not be tampared

with. However, this results in the tracking library not being able to send tracking data to the WebTrekk servers.

Another reason not to send data directly to WebTrekk is to have a "circuit breaker" in place when an application breaks and sends a lot of (wrong) information to the WebTrekk servers. Instead of getting a large bill from WebTrekk, they can block the application and wait for it to recover or be fixed.

The solution to overcome this is to use of a reverse proxy. In this case, this results in the website sending data to the ING servers, which in turn sends it to the WebTrekk servers, as ING servers do not have to abide by the CORS rules.

This means that for every application at ING a reverse proxy has to be set up. This requires a change in the configuration of the network of ING, which for security purposes needs to go through a certain workflow. Estimates for the length of this procedure turned out to be around 3 months and this made it impossible to implement in the timeframe left.

7.3 The platform survey

To determine if there are any problems with the proposed solution, a feedback point needs to be created. We could ask experimenters if they are satisfied with the new solutions, however this could have unintended consequences. Asking if experimenters are satisfied with the new solutions will always involve some prejudice. The experimenter might give positive feedback to not disappoint the person asking for it or come up with feedback on the spot, which is not the type of feedback we want. After all, if the experimenter does not care (enough), we resolve the pitfalls without any change in user satisfaction. Furthermore, with the survey we already have a good indication if the solution will be positively received. To counter these possible problems in the feedback cycle, we created a generic survey, which is described below. Because it is a generic survey, the Tetris team (2.2) can determine if there are other aspects of the tool which can be improved and therefore it is more likely that the team allows such a feature to be introduced in the platform.

7.3.1 The survey layout

The survey has two requirements. The first requirement is to determine if the overall satisfaction has not gone down. This is satisfied by the first question on the survey, which is a Likert scale asking how satisfied the developer is with the experiment platform. This question gives us an opportunity to look at the satisfaction over time and therefore give us a reasonable metric of satisfaction over time. The term reasonably here is used as opinions change over time and by definition are subjective. However, if there are enough results, we should be able to average these subjective data points out and get a somewhat objective result.

The second requirement is to determine if the given satisfaction given by the experimenter in the first question has anything to do with the solutions which have been implemented. To do this we present the experimenter with a textbox where they can write detailed feedback on what they like and don't like about the platform. To try and get the experimenter to talk about the solutions the following placeholder text is used: "Write as

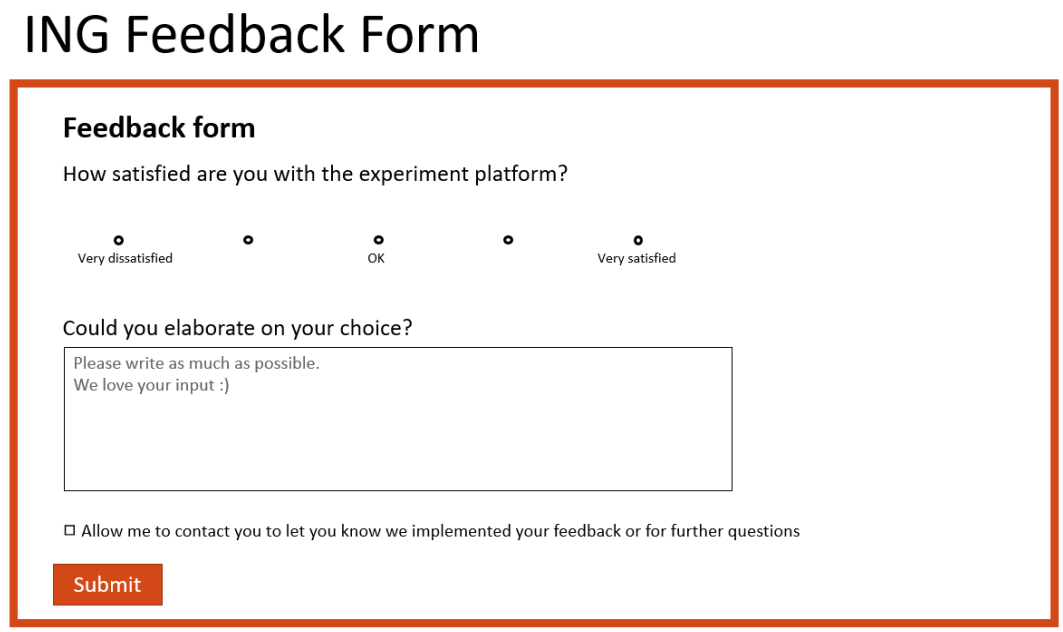


Figure 7.3: Mockup of the survey

much as possible" and "We love your input". The intent is that this will increase the length of the feedback. If the experimenter does not mention the new solutions, this will be either because they deemed it not important enough to talk about, they are happy with it or they do not care. If they do mention the new solutions, we should check if their feedback is valid and actionable. If that is the case, we could improve the solution further. Otherwise we note down this result. If the feedback is positive, we can also note this. Lastly, we ask the experimenter if we can have their email address. This might be of value if we want to follow-up with the experimenter to get a better understanding of why they do not like a solution or if the feedback they gave is not clear. This is not a required field because of course this survey is anonymous to make sure that we do not bias the results by having the experimenter think there might be some negative consequences if they criticize the platform. A mockup of the survey is shown in Figure 7.3 and the final survey is shown in Figure 7.4.

7.3.2 Placing the platform survey

There are three places in the workflow where this survey was placed. The first one is after the experiment-creation workflow and can be seen in Figure 7.5. The creation workflow is when the experimenter has created the experiment in the experiment platform. When creating a new experiment, the experimenter must go through a few steps to set the name, hypothesis and target group. At the end of this workflow we ask experimenters what they think of the platform. This has another benefit, as many solutions are placed in the creation workflow. Therefore, this is a great place to ask for feedback.

The second place a survey will be created is after the experiment has concluded, see

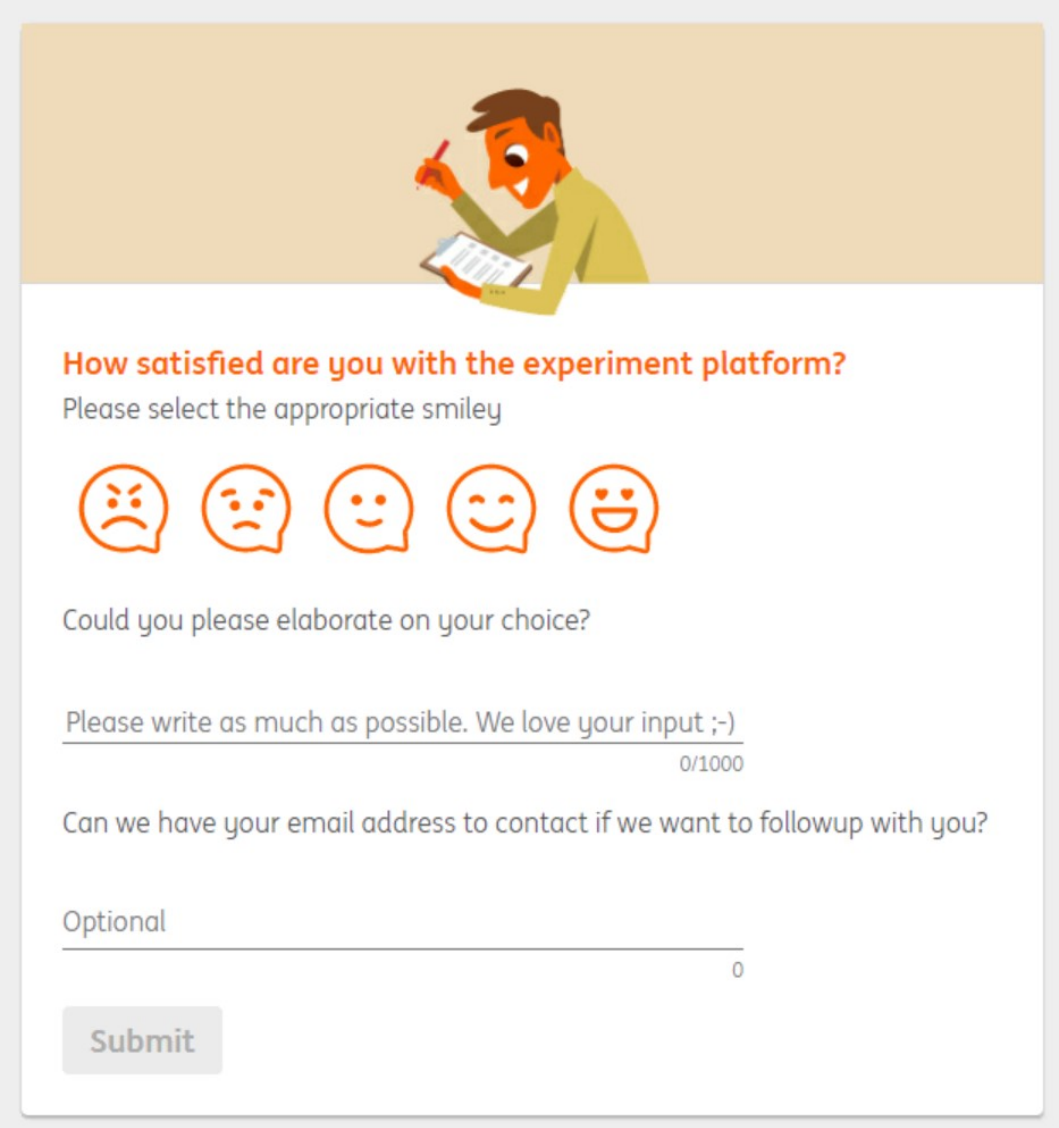


Figure 7.4: Final survey

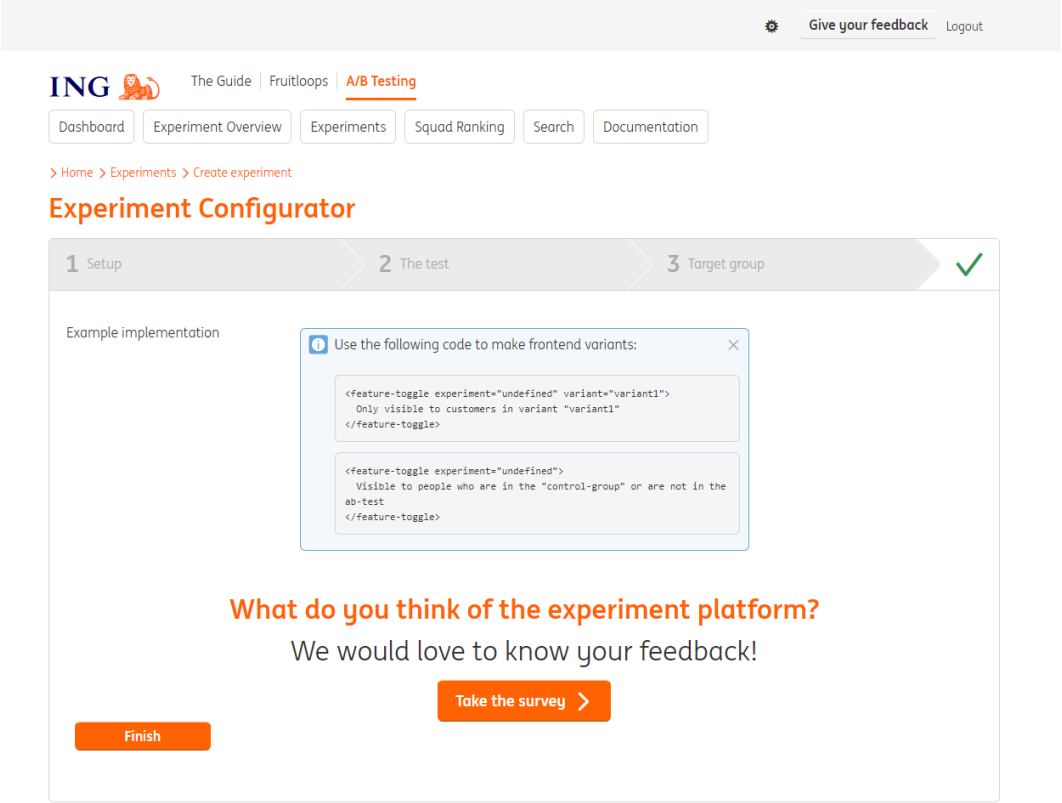


Figure 7.5: Button to the survey

Figure 7.6. At this point the experimenter has possibly checked the website during the runtime of the experiment, looking at the intermediate result, but possibly forgot that experiment was running. To remind the experimenter that the experiment has concluded, we, in collaboration with the Tetris team, created a feature that will send an email to the developer telling them that their experiment has concluded. This opportunity is then also used to ask the experimenter to fill in a survey. Again, this is a great place to ask for feedback, as they have gone through the entire flow of an experiment and therefore can tell exactly where the pain points of the system are and what can be improved.

Lastly, there is a generic feedback button. On every page, as can be seen in Figure 7.5, a user can press the “Give your feedback” button to give feedback on the platform.

Asking at these locations in the process does have some drawbacks. The survey after the experiment-creation-workflow leaves out the demographic of people, which left the workflow halfway through. As we ask experimenters about their opinion after the workflow has finished, this does not include the people who do not finish the workflow (obviously). Unfortunately, it is impossible and impractical to ask a person to fill in a survey when leaving the website and therefore we cannot reach this demographic. The same holds for the survey shown in the reminder email. We might have lost experimenters, who have stopped their

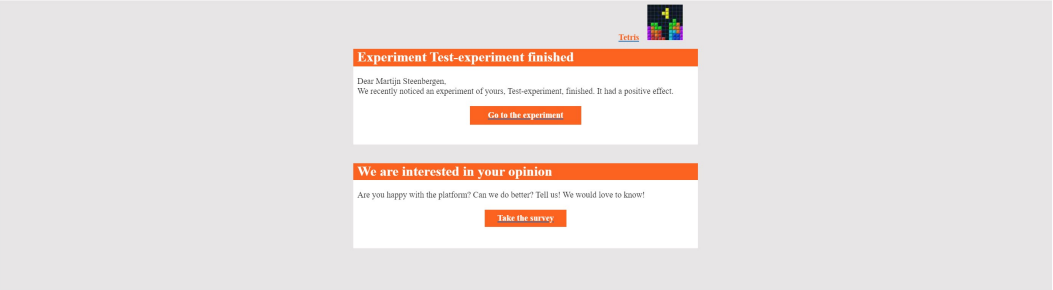


Figure 7.6: Reminder email

experiments midway through.

7.3.3 Preliminary Results

As only few people responded to the survey, we will briefly discuss the results below.

On the 14th of February the first respondent filled in the survey. At the time of writing, July, only 9 experimenters have filled in the survey. 5 of these are just praising the experiment platform, usually saying "great" and rate the system good (2) or great (3). 3 respondents fill in the survey asking for the feature of tracking more than one metric and give a fair (2) or bad (1) rating. The last respondent wanted technical help and gave the platform a bad rating.

7.3.4 Conclusion on the use of a survey

Although setting up this survey was promising, the results are preliminary and should not be used to validate a possible implementation in the end, as they will not have enough results to make definitive conclusions on the solution. These results do indicate that people want more metrics, something that can and should weigh in when deciding the roadmap of the platform.

Chapter 8

Conclusion & Discussion

8.1 Conclusion

Online controlled experimentation is a powerful tool, but can be deceivingly easy, as experimenters can face many pitfalls while performing their experiment. It is vital for experimenters and organisations to perform correct and valuable experiments. To aid them, this thesis tries to answer the question: "*How can we reduce human error in online controlled experimentation?*" To answer this question, exploratory interviews were held to determine common problems experimenters face. By looking at common pitfalls developers face, based on previous work, a list of possible solutions was created, taking existing solution as starting points. A survey, answered by 52 experimenters at ING, was performed to validate the result of the interviews and to evaluate the proposed solutions on the perceived helpfullness to the experimenter.

8.1.1 RQ1 What are key pitfalls and their potential solutions?

By looking at existing literature, we established that a list of 19 pitfalls are relevant for this thesis. By leveraging existing solutions, such as Optimizely [53], VWO [4], Adobe Target [8] and Firebase [1] as inspiration, solutions can be created to solve these problems. This thesis proposes 8 solutions to solve the 19 pitfalls, ranging from invasive changes, such as adding a review step to the workflow, to simple solutions like a checklist.

8.1.2 RQ2 How are pitfalls and their solutions experienced?

Experimenters are well aware of problems that can occur while experimenting. During interviews, we determined that experimenters are already using statistical methods to determine the minimum runtime of experiments. The survey confirmed that for each pitfall more than 50% of the experimenters is aware that this problem could present itself. Less experimenters have actually experienced the pitfalls. Experimenters think each pitfall to be roughly equally severe with the exception of "Competitor safety", which they consider not severe.

There are mixed reactions to the proposed solutions. The solutions were viewed in two different perspectives: whether it solves the problem and whether the solution helps the experimenter. Experimenters are immensely enthusiastic about the solution "Enforcing the correct experiment duration", solving "Calculate the minimum duration" and "Calculate effect size". Respondents believed the solution solved these problems 100% and 94.1% respectively and almost all respondents perceived the solution to (slightly) help the experimenter in performing their experiments. The next best solution (in solving the problem) drops by 20%, making other solutions still relevant to look at, but not as much. The same holds for the perceived usefullness of the solution, where some ("Rerunning experiments" and "Validation of experiments") are highly rated, but not as much. There are not truly badly received results. Worst received is the proposed solution solving "The number of changes" pitfall, where more than half of the respondents thought the solution (slightly) hindered the experimenter.

RQ3 How can one evaluate a solution in a real-world scenario?

From the exploratory work done in this thesis, we determined that evaluating a solution in a real-world scenario runs into technical challenges. To evaluate a solution it must be possible to track and log the interaction an experimenter has with the platform, and to receive feedback from experimenters. Both are not adequately implemented into the platform. Tracking on the platform does not exist and implementing tracking runs into technical and bureaucratic problems. Current feedback methods do exist and work, although there is not enough feedback to be able to draw conclusions. Therefore, at this moment, there is no way of evaluating a solution in a real-world scenario.

Summary

There are a lot of pitfalls experimenters face while performing online controlled experiments. This thesis shows that experimenters are well informed about the existence of pitfalls and believe that almost all should be resolved. There are many promising solutions to these pitfalls which experimenters rate as helpful. However evaluating these solutions at this point in the current context is not possible.

8.2 Implications

8.2.1 Implications for Industry

This work is an example for how an organization can start with implementing continuous experimentation. The case study at ING shows what experiments are being performed and can inspire readers for new avenues of experiments. The main work provides an overview of which pitfalls experimenters deem important and which solutions will help experimenters. This provides any company an indication which pitfalls are most important to resolve first and which solutions could help them in resolving these pitfalls.

8.2.2 Implications for ING

Based on the results of the survey, there now is a list of possible improvements which can help experimenters with experimentation. It should then be no surprise that one of the recommendations is to start implementing the solutions.

In chapter 7, a lot of shortcomings of the process are described and what hurdles still remain in the data driven culture of ING. These should be resolved, so that any employee can improve the interactions ING has with their customers.

Finally, during the time of the internship and research a view on how the experiment office can be improved further was created, which is discussed in Appendix A.

8.2.3 Implications for Research

To the authors' knowledge, this is the first body of knowledge surveying experimenters with solutions for pitfalls, before implementation. This shows interesting insights into which pitfalls are important. Besides points mentioned in 8.2.1, this thesis also tries to provide a jumping off point for any new research being performed under the banner of the AI for Fintech Research (AFR).

8.3 Threats to validity

8.3.1 Internal validity

Internal conditions can affect the independent variable with respect to causality, without the researcher being aware [50]. In regard to this thesis this includes any influences on the survey and interview, which we cannot know.

One possible threat is the threat of experimental mortality, i.e. the loss of participants during the experiment. The survey was sent out to many people. People with limited experience with continuous experimentation might not be motivated to start or continue with the survey if they believe they cannot give relevant answers and/or are not passionate about experimentation. Experienced experimenters might be more motivated to fill in the survey, resulting in biased results.

Exploration into this threat resulted Figure 8.1, which shows the distribution of answers not to be clearly biased towards a particular experience group.

8.3.2 Construct validity

Construct validity concerns generalizing the result of the experiment to the concept or theory behind an experiment [50].

For the survey, we presented the participants with solutions to pitfalls. However, in the eyes of particpants, this might look like a possible feature list. The question if the solution would help or hinder might be also be seen as asking how much they want this feature. Answers to these question might then be biased. This bias might then also increase in areas where the experimenter has more experience. This threat cannot be mitigated, but can be taken and has into account when analyzing the results.

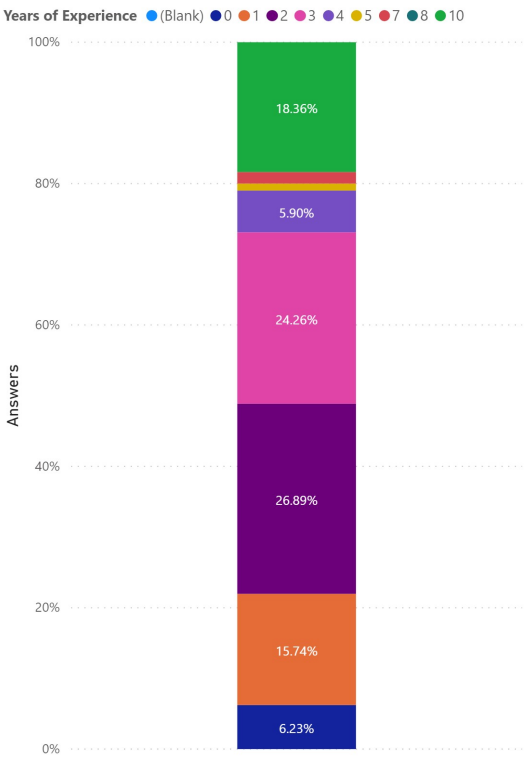


Figure 8.1: Diagram showing the combined number of responses for pitfalls, set to the years of experience

8.3.3 Conclusion validity

Conclusion validity threats can affect the ability to draw correct conclusion [50].

For this research, the biggest threat is low statistical power, as the number of people running experiments at ING is high enough that performing interviews is an insane amount of work, but performing a survey, even with the great response rate seen in this case, is not a large enough sample size to significantly determine the results. However, anything short of making the survey mandatory will not change this.

8.3.4 External validity

External valididity threats are conditions that limit the ability to generalize the results of the experiment industrial practice [50].

As the interviews and survey were performed at a single company, this introduces the possibility that this work is not generalizable over the industry, as views on pitfalls might differ, based on what the needs of the organization are. Furter research is needed to validate these results at other companies.

8.4 Future work

In the overall picture of continuous experimentation research at ING and the AI4Fintech research group, this research can serve as a foundation to a lot of other work.

8.4.1 Evaluate solutions in a real-world scenario

In chapter 7, it was concluded that there were significant hurdles to evaluating the features. These hurdles can be overcome with time. A future researcher can perform this work or wait until a time engineers at ING have performed this research.

8.4.2 Reproduce this research in different industries

This work has shown a first indication of the perceived severity and value of solutions to practitioners. To be truly able to say that experimenters value these solutions, this survey should be repeated at different companies, favourably in different (non FinTech) industries.

8.5 Ethics of online controlled experimentation

Online controlled experiments can bring enormous advantages to companies [53, 17, 30]. It is therefore no surprise that many [46, 15, 38, 36, 24, 16] use them. However, there are some serious ethics questions related to using experiments to try and sway user actions.

OCEs reveal an interesting fact about us, humans. Where one would love to pretend that we act with only logic, example after example in the world of online controlled experimentation shows us that this idea that we have come to believe is wrong. Colours can influence our click behaviour [52] and the change of an image increases the amount of mortgages bought [40].

In 2012, Facebook published a study showing that if a Facebook user was shown more positive posts, their own posts also showed more positive words. The same held for negative words. Another study showed that Facebook could increase voter turnout by displaying badges on users pages [51]. These results show both that there is enormous positive potential in the use of experimentation to increase happyness, but also gives us an insight into the enormous power these companies hold.

Furthermore, it is reasonable to assume that few of the participants of the experiments were aware they were being targeted (until 2012, it was not event mentioned in the terms of service of Facebook that your data could be used for research purposes [11, 51]).

In an academic experimental setting this raises serious questions [54]. While performing research, I could find no company performing experiments on their users actually disclosing this to them. In fact, I was surprised at the breadth of the companies performing OCEs and the scale at which these experiments were performed. It is hard to opt-out of something you do not know is happening. The new cookie law by the European Union seems to have had a positive influence in this case, as it is best practice to use cookies to track users [16] and

decide variants based on a user cookie. Therefore, by not accepting cookies, you also do not give consent being experimented on.

Another large problem in the area of online controlled experimentation seems that there does not seem to be an ethical code of conduct regarding experimentation [11]. There should guidelines to determine areas where OCEs are not ethical or even harmful to the customer of the product. For example, forbidding to perform experiments to increase the amount of children buying V-bucks with their parent's credit card, to tempt broke gamblers to to just play another round of poker or to increase sales of mortgages if this has long term negative effects for the customer.

As with any tool, they can be used for good and bad. As such, OCEs are no exception and therefore time and effort should be put in to make sure they are used correctly and ethical.

Bibliography

[1] Firebase. URL https://firebase.google.com/. Accessed: 2020-05-18 02:03:24.

[2] Website, a/b testing & optimization tools - google optimize, . URL https://marketingplatform.google.com/about/optimize/. Accessed: 2020-05-18 02:11:58.

[3] Vwo, . URL https://vwo.com/. Accessed: 2020-05-18 02:03:24.

[4] Vwo. URL https://vwo.com/. Accessed: 2020-05-18 02:03:24.

[5] Cross-origin resource sharing, 2017. URL https://www.w3.org/TR/2014/REC-cors-20140116/.

[6] Research contributions in human-computer interaction — acm interactions, 2020. URL https://interactions.acm.org/archive/view/may-june-2016/research-contribution-in-human-computer-interaction.

[7] scipy.stats.mannwhitneyu — scipy v1.5.2 reference guide, 2020. URL https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.mannwhitneyu.html.

[8] adobe.com. Personalization - adobe inc. URL https://www.adobe.com/marketing/target.html.

[9] Alan Agresti. *Categorical data analysis*. Wiley-Interscience, Hoboken, N.J, 2013. ISBN 978-0-470-46363-5.

[10] Florian Auer and Michael Felderer. Current state of research on continuous experimentation: A systematic mapping study. In *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, aug 2018. doi: 10.1109/seaa.2018.00062. URL https://doi.org/10.1109%2Fseaa.2018.00062.

[11] Raquel Benbunan-Fich. The ethics of online research with unsuspecting users: From a/b testing to c/d experimentation. *Research Ethics*, 13(3-4):200–218, nov 2016. doi: 10.1177/1747016116680664. URL https://doi.org/10.1177%2F1747016116680664.

[12] Mark Bounthavong. Communicating data effectively with data visualization – part 15 (diverging stacked bar chart for likert scales) — mark bounthavong, 2020. URL https://mbounthavong.com/blog/2019/5/16/communicating-data-effectively-with-data-visualization-part-15-divergent-stacked-bar-chart-for-likert-scales.

[13] Thomas Crook, Brian Frasca, Ron Kohavi, and Roger Longbotham. Seven pitfalls to avoid when running controlled experiments on the web. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09*. ACM Press, 2009. doi: 10.1145/1557019.1557139. URL https://doi.org/10.1145%2F1557019.1557139.

[14] Pavel Dmitriev, Somit Gupta, Dong Woo Kim, and Garnet Vaz. A dirty dozen. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, aug 2017. doi: 10.1145/3097983.3098024. URL https://doi.org/10.1145%2F3097983.3098024.

[15] eng.uber.com. experimentation-platform from eng.uber.com, 2020. URL https://eng.uber.com/experimentation-platform/. (Accessed on Mon Sep 16 2019).

[16] exp platform.com. Documents 2019-first practical online controlled experiments summit sigkdd explorations.pdf from exp-platform.com, 2020. URL https://exp-platform.com/Documents/2019-FirstPracticalOnlineControlledExperimentsSummit_SIGKDDExplorations.pdf. (Accessed on Fri Sep 13 2019).

[17] Aleksander Fabijan, Pavel Dmitriev, Colin McFarland, Lukas Vermeer, Helena Holmström Olsson, and Jan Bosch. Experimentation growth: Evolving trustworthy a/b testing capabilities in online software companies. *Journal of Software: Evolution and Process*, 30(12):e2113, nov 2018. doi: 10.1002/smr.2113. URL https://doi.org/10.1002%2Fsmr.2113.

[18] Aleksander Fabijan, Pavel Dmitriev, Helena Holmstrom Olsson, and Jan Bosch. Online controlled experimentation at scale: An empirical survey on the current state of a/b testing. In *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, aug 2018. doi: 10.1109/seaa.2018.00021. URL https://doi.org/10.1109%2Fseaa.2018.00021.

[19] Aleksander Fabijan, Pavel Dmitriev, Helena Holmstrom Olsson, Jan Bosch, Lukas Vermeer, and Dylan Lewis. Three key checklists and remedies for trustworthy analysis of online controlled experiments at scale. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. IEEE, may 2019. doi: 10.1109/icse-seip.2019.00009. URL https://doi.org/10.1109%2Ficse-seip.2019.00009.

[20] Google. Angularjs — superheroic javascript mvw framework, . URL https://angularjs.org/.

[21] Google. Polymer project, . URL https://www.polymer-project.org/.

[22] Jayant Gupchup, Yasaman Hosseinkashi, Pavel Dmitriev, Daniel Schneider, Ross Cutler, Andrei Jefremov, and Martin Ellis. Trustworthy experimentation under telemetry loss. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, oct 2018. doi: 10.1145/3269206.3271747. URL https://doi.org/10.1145%2F3269206.3271747.

[23] Somit Gupta, Lucy Ulanova, Sumit Bhardwaj, Pavel Dmitriev, Paul Raff, and Aleksander Fabijan. The anatomy of a large-scale experimentation platform. In *2018 IEEE International Conference on Software Architecture (ICSA)*. IEEE, apr 2018. doi: 10.1109/icsa.2018.00009. URL https://doi.org/10.1109%2Ficsa.2018.00009.

[24] hbr.org. 2017 09 the-surprising-power-of-online-experiments from hbr.org, 2020. URL https://hbr.org/2017/09/the-surprising-power-of-online-experiments. (Accessed on Fri Sep 27 2019).

[25] Hennie Huijgens, Davide Spadini, Dick Stevens, Niels Visser, and Arie van Deursen. Software analytics in continuous delivery. In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. ACM, oct 2018. doi: 10.1145/3239235.3240505. URL https://doi.org/10.1145%2F3239235.3240505.

[26] ING.com. 2019 annual report, . URL https://www.ing.com/About-us/Annual-reporting-suite/Annual-Report/2019-Annual-Report.htm.

[27] ING.com. 2018 annual report, . URL https://www.ing.com/About-us/Profile/ING-at-a-glance.htm.

[28] Raphael Kaufman, Jegar Pitchforth, and Lukas Vermeer. Democratizing online controlled experiments at booking.com. 10 2017.

[29] Ron Kohavi, Randal M. Henne, and Dan Sommerfield. Practical guide to controlled experiments on the web. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '07*. ACM Press, 2007. doi: 10.1145/1281192.1281295. URL https://doi.org/10.1145%2F1281192.1281295.

[30] Ron Kohavi, Roger Longbotham, Dan Sommerfield, and Randal M. Henne. Controlled experiments on the web: survey and practical guide. *Data Mining and Knowledge Discovery*, 18(1):140–181, jul 2008. doi: 10.1007/s10618-008-0114-1. URL https://doi.org/10.1007%2Fs10618-008-0114-1.

[31] Ron Kohavi, Diane Tang, Ya Xu, Lars G. Hemkens, and John P. A. Ioannidis. Online randomized controlled experiments at scale: lessons and extensions to medicine. *Trials*, 21(1), feb 2020. doi: 10.1186/s13063-020-4084-y. URL https://doi.org/10.1186%2Fs13063-020-4084-y.

[32] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Ann. Math. Statist.*, 18(1):50–60, 03 1947. doi: 10.1214/aoms/1177730491. URL https://doi.org/10.1214/aoms/1177730491.

[33] Mapp. Webtrekk is now a full-fledged part of mapp. URL https://www.webtrekk.com/.

[34] David Issa Mattos, Pavel Dmitriev, Aleksander Fabijan, Jan Bosch, and Helena Holmström Olsson. An activity and metric model for online controlled experiments. In *Product-Focused Software Process Improvement*, pages 182–198. Springer International Publishing, 2018. doi: 10.1007/978-3-030-03673-7_14. URL https://doi.org/10.1007%2F978-3-030-03673-7_14.

[35] medium.com. Duolingo: 1 URL https://medium.com/googleplaydev/duolingo-1-improvement-every-week-ab7d61689119. (Accessed on 14-Jun-20 12:09:38).

[36] medium.com. airbnb-engineering experiments-at-airbnb-e2db3abf39e7 from medium.com, 2020. URL https://medium.com/airbnb-engineering/experiments-at-airbnb-e2db3abf39e7. (Accessed on Mon Sep 16 2019).

[37] medium.com. airbnb-engineering https-medium-com-jonathan-parks-scaling-erf-23fd17c91166 from medium.com, 2020. URL https://medium.com/airbnb-engineering/https-medium-com-jonathan-parks-scaling-erf-23fd17c91166. (Accessed on Mon Sep 16 2019).

[38] medium.com. netflix-techblog selecting-the-best-artwork-for-videos-through-a-b-testing-f6155c4595f6 from medium.com, 2020. URL https://medium.com/netflix-techblog/selecting-the-best-artwork-for-videos-through-a-b-testing-f6155c4595f6. (Accessed on Mon Sep 16 2019).

[39] Juliette Meeuwsen, Matthijs van Leeuwen, Hennie Huijgens, Kevin Anderson, and Arie van Deursen. Stimulating the adoption of a/b testing in a large-scale agile environment. 5 2019.

[40] Ernst Mulder. Data driven decisions: Validating and supporting a continuous experimentation development environment. Master's thesis, TU Delft Electrical Engineering, Mathematics and Computer Science, 12 2019. https://repository.tudelft.nl/islandora/object/uuid%3A08f2c0b4-2aa8-4e12-9b58-073dcdfb4553?collection=education.

[41] optimizely.com. Share your results with stakeholders, . URL https://help.optimizely.com/Analyze_Results/Share_your_results_with_stakeholders.

[42] optimizely.com. Optimizely x web experimentation, . URL https://www.optimizely.com/platform/experimentation/.

[43] Radiolab. The trust engineers—radiolab—wnyc studios, 2020. URL https://www.wnycstudios.org/podcasts/radiolab/articles/trust-engineers. (Accessed on Tue Mar 24 2020).

[44] Murali Krishna Ramanathan, Lazaro Clapp, Rajkishore Barik, and Manu Sridharan. Piranha: Reducing feature flag debt at uber.

[45] Survalyzer. Professional survey tool and panel management - survalyzer, 2020. URL https://www.survalyzer.com/.

[46] Diane Tang, Ashish Agarwal, Deirdre O'Brien, and Mike Meyer. Overlapping experiment infrastructure. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '10*. ACM Press, 2010. doi: 10.1145/1835804.1835810. URL https://doi.org/10.1145%2F1835804.1835810.

[47] Edith Tom, Aybüke Aurum, and Richard Vidgen. An exploration of technical debt. *Journal of Systems and Software*, 86(6):1498 – 1516, 2013. ISSN 0164-1212. doi: https://doi.org/10.1016/j.jss.2012.12.052. URL http://www.sciencedirect.com/science/article/pii/S0164121213000022.

[48] András Vargha and Harold D. Delaney. A critique and improvement of the cl common language effect size statistics of mcgraw and wong. *Journal of Educational and Behavioral Statistics*, 25(2):101–132, 2000. doi: 10.3102/10769986025002101. URL https://doi.org/10.3102/10769986025002101.

[49] Wikipedia. Covid-19 disease - wikipedia. URL https://en.wikipedia.org/wiki/Coronavirus_disease_2019.

[50] Claes Wohlin. *Experimentation in software engineering*. Springer, Berlin New York, 2012. ISBN 978-3-642-29044-2.

[51] www.nytimes.com. A bright side to facebook's experiments on its users, 2020. URL https://www.nytimes.com/2014/07/03/technology/personaltech/the-bright-side-of-facebooks-social-experiments-on-users.html. (Accessed on 14-Jun-20 12:09:38).

[52] www.nytimes.com. 2009 03 01 business 01marissa.html from www.nytimes.com, 2020. URL https://www.nytimes.com/2009/03/01/business/01marissa.html?pagewanted=print. (Accessed on Fri Sep 27 2019).

[53] www.optimizely.com. customers from www.optimizely.com, 2020. URL https://www.optimizely.com/customers/. (Accessed on Fri Feb 28 2020).

[54] Sezin Yaman, Fabian Fagerholm, Myriam Munezero, Hanna Maenpaa, and Tomi Mannisto. Notifying and involving users in experimentation: Ethical perceptions of software practitioners. In *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE, nov 2017. doi: 10.1109/esem.2017.31. URL https://doi.org/10.1109%2Fesem.2017.31.

Appendix A

Future version of the experiment office

During the months working and thinking about improving the platform, insights was gained into what a future version of the platform could look like. The following is a personal opinion on what this future version could look like and has not been validated.

In Figure A.1, one can find this new version. The overall design is based on an accordion view, inspired by the Firebase UI [1]. When a user opens the page for an experiment, the page opens the step of the process the experimenter is at. If the experimenter wants to look at another step (for example Experiment Setup instead of the Results), they can click on that section and it will expand, to show all the information the experimenter has filled in, while the current section is hidden. This design tries to minimize the use of dialogs, such that any new experimenter can look at a random experimenter, see all the settings used, and be able to rerun it without much trouble.

This version also has better integration with WebTrekk, showing a dashboard, inspired by what Meeuwsen et al. found[39] (see Figure 4.4) and how the dashboard of Airbnb looks Figure A.2. This will be either a dashboard created in-house or the WebTrekk dashboard.

It is very important to have one single place to keep track of experiments. From literature, we have determined that it can be very useful to have a searchable repository of experiments [28]. This approach should be two fold. First, ING should be making an effort to consolidate experimentation to a single platform. Secondly, the platform should integrate with as many products as possible. By importing data, possibly read-only, the platform becomes the go to place if a person wants to know something about a (type of) experiment.

This could lead to the following workflow. John wants to see what effect changing the color of a button on the homepage has on the click-through-rate. John goes to the experiment office to look for anyone who has done experiments with buttons and colors in the last year and finds that his colleague, Hank, has done so on a different platform than John, but it shows up, because it is integrated. After a quick conversation, John learns that the color green usually works best. John creates his new experiment on the platform and creates the variants. Using the platform he determines how long the experiment should run for (6.4.4) and starts the experiment.

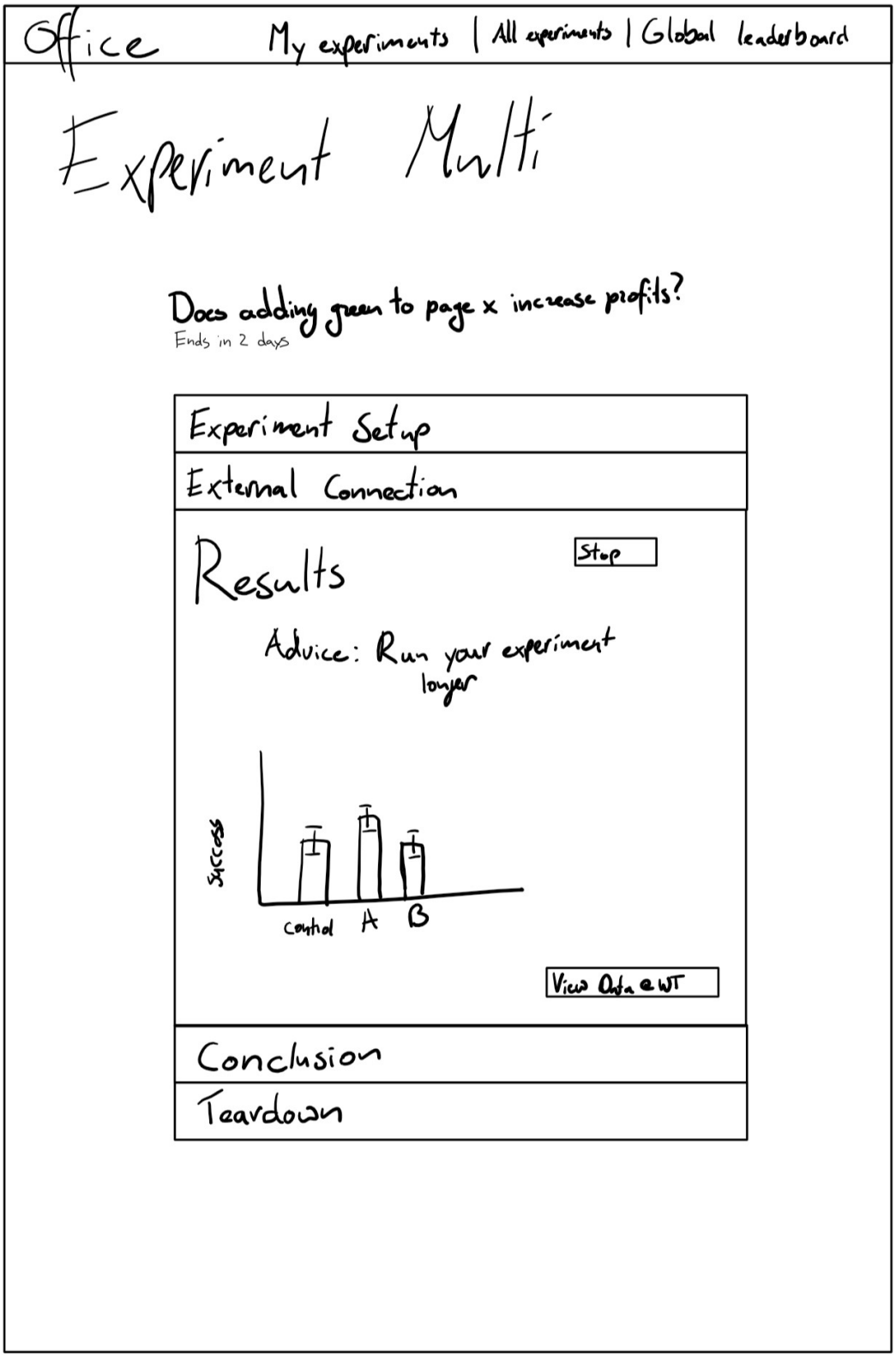


Figure A.1: Mockup of a possible version 2 of the experiment office

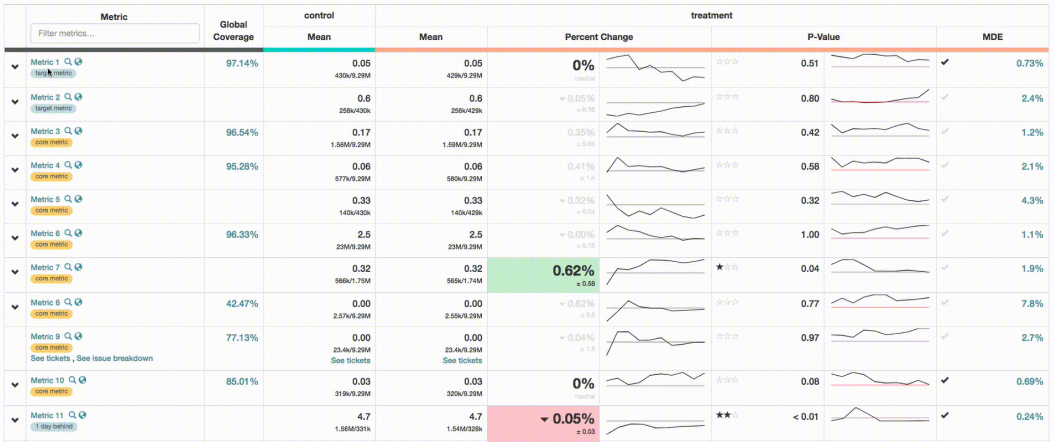
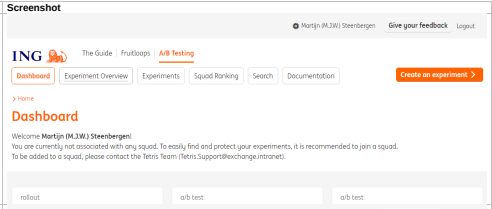
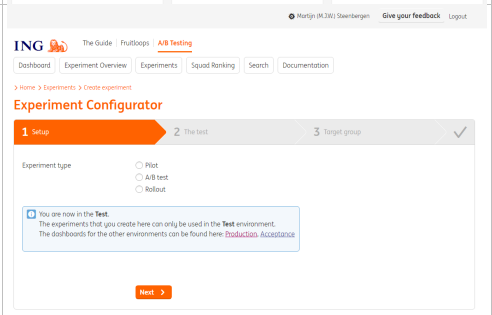
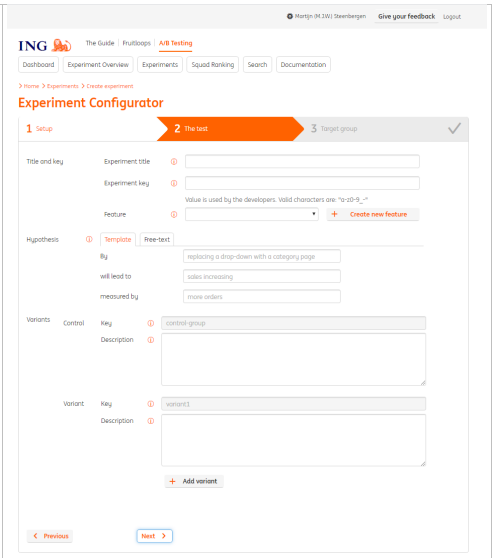


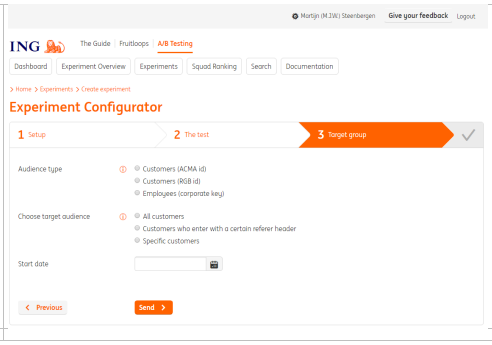
Figure A.2: Screenshot of the metrics dashboard from Airbnb

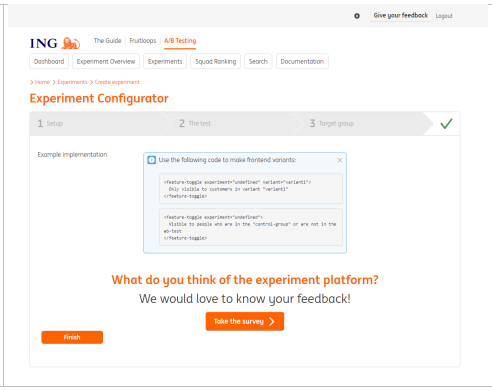
Appendix B

Tracking plan

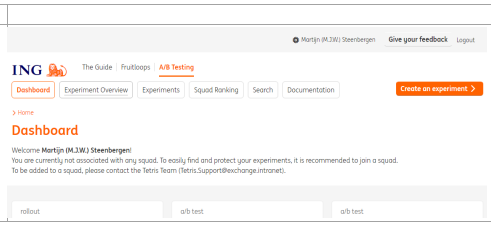
Because there does not exist a prefix for the experiment office, we introduce a new one: expo.

Id	Screenshot	Action	Type	Tag
Home		User logs into the experiment platform	Page	expo:home
		They click the `New Experiment` button	Event	expo:home.createButton -> Create
Create		The new experiment page opens	Page	expo:create
		They fill in the experiment type and click next	FormEvent	FormID: ExperimentCreator FormStep: 1 FormStatus: 'View' -> Create.2
Create.2		They fill in the hypothesis and click next	FormEvent	FormID: ExperimentCreator Formstep: 2 -> Create.3
		They click previous	FormEvent	FormID: ExperimentCreator FormStep: 1 FormStatus: 'View' -> Create
Create.3		They fill in the target group and click next	FormEvent	FormID: ExperimentCreator, Formstep: 3, Formstatus: 'conf' -> Finish

		They click on previous	FormEvent	FormID: ExperimentCreator Formstep: 2 -> Create.2
Finish		They click on fill in the survey	Event	expo:create.inlineFeedbackButton

		They click on Finish	Event	expo:create.FinishButton

An experimenter opens the experiment office to leave feedback using the button at the top

Id		Description	Type	Tag
Home		User logs into the experiment platform	Page	expo:home
		They click on the feedback button	Event	expo:home.genericFeedbackButton

Appendix C

Survey Information

This appendix gives an extensive overview of the survey. In Figure C.1, a screenshot is shown of the first page of the survey. In Figure C.2, one of the survey pages is shown, where questions are asked about a specific pitfall. For each of the pitfalls, everything is the same, except for two items. First, in the text of the first question, an explanation of the pitfall is. Then, secondly, halfway around the page, the explanation for the solution is put. The exact descriptions for each of these pages can be found in this appendix.

C.1 Pitfall descriptions

C.1.1 Falsifiable hypothesis (01b)

Explanation

A falsifiable hypothesis means that a hypothesis can be proven wrong. If a hypothesis cannot be proven wrong, the experiment has less value, as the result is already known. “The views on this page will increase” is a falsifiable hypothesis “The people will be happier with a green button” is not, because there is no way of measuring happiness with metrics from a web pag and therefore impossible to (dis)prove.

Solution

We solve this problem by having other people check the experimental setup. Before your experiment can start, it is looked over by a group of experts, who could find mistakes in the experimental setup or tips on how to improve the experiment. The experiment is not allowed to start before the ok is given.

C.1.2 Direction of change in hypothesis (02b)

Explanation

When creating a hypothesis for an experiment, usually it contains a description of what will happen to the metric that is measured. For example “When removing the sign up, we

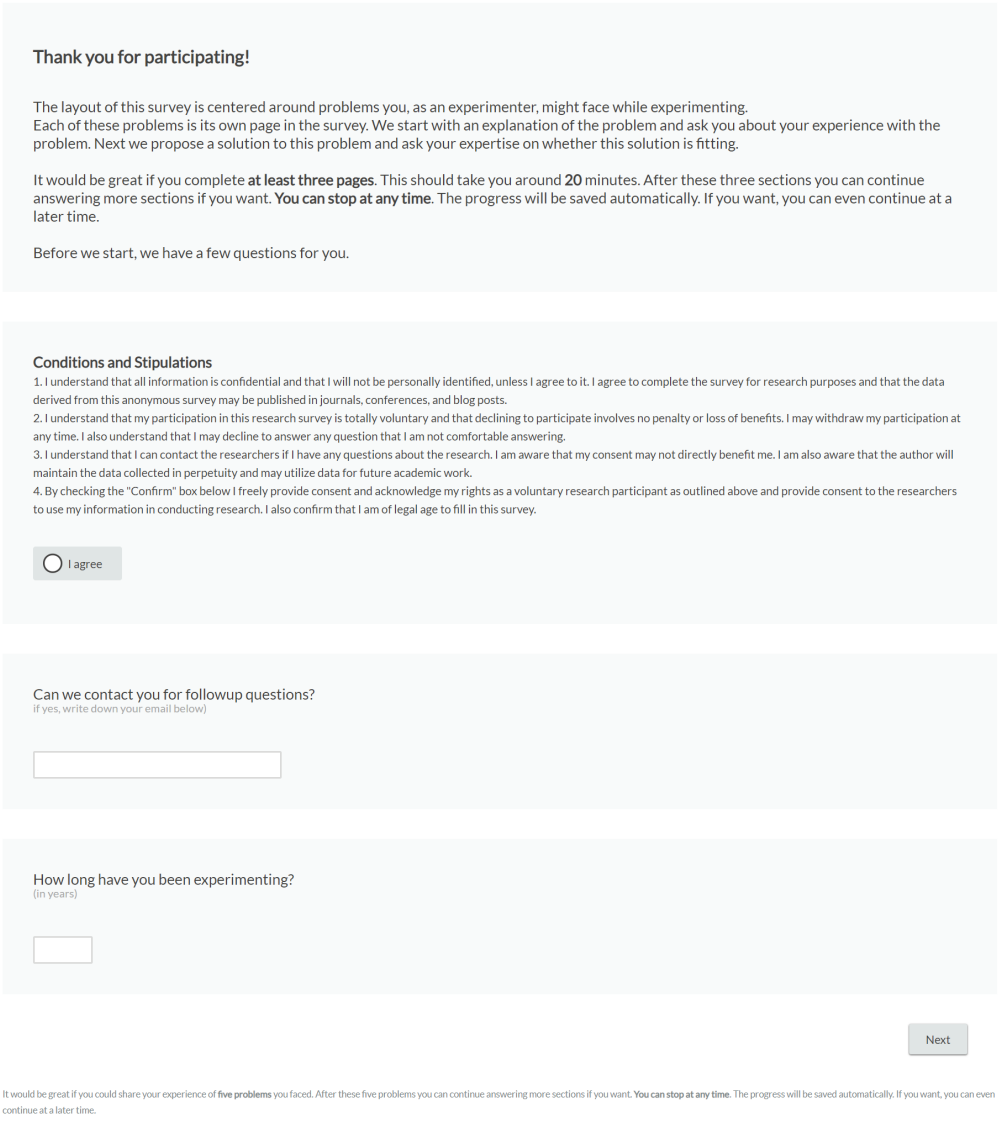
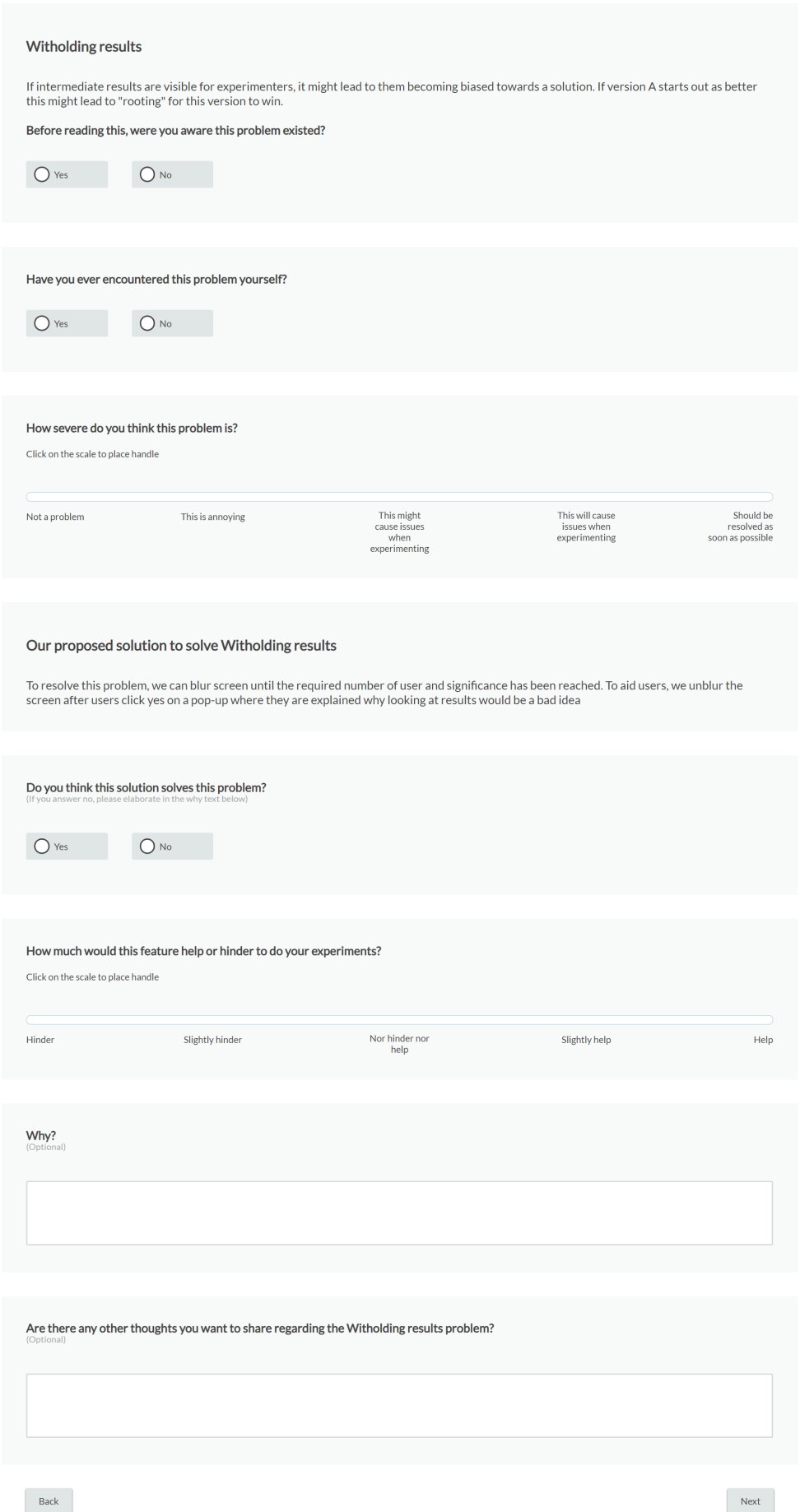


Figure C.1: The first page of the survey


Witholding results
If intermediate results are visible for experimenters, it might lead to them becoming biased towards a solution. If version A starts out as better this might lead to "rooting" for this version to win.
Before reading this, were you aware this problem existed?
Yes
No
Have you ever encountered this problem yourself?
Yes
No
How severe do you think this problem is?
Click on the scale to place handle
Not a problem
This is annoying
This might cause issues when experimenting
This will cause issues when experimenting
Should be resolved as soon as possible
Our proposed solution to solve Witholding results
To resolve this problem, we can blur screen until the required number of user and significance has been reached. To aid users, we unblur the screen after users click yes on a pop-up where they are explained why looking at results would be a bad idea
Do you think this solution solves this problem?
(If you answer no, please elaborate in the why text below)
Yes
No
How much would this feature help or hinder to do your experiments?
Click on the scale to place handle
Hinder
Slightly hinder
Nor hinder nor help
Slightly help
Help
Why?
(Optional)
Are there any other thoughts you want to share regarding the Witholding results problem?
(Optional)
Back
Next
It would be great if you could share your experience of five problems you faced. After these five problems you can continue answering more sections if you want. You can stop at any time. The progress will be saved automatically. If you want, you can even continue at a later time.

expect signups to go down". In this case the metric measured is the number of signups and the direction is that this number will decrease. Writing this down forces the experimenter to explicitly hypothesize what will happen. Also statistically, the maths between calculating significance and calculating the possibility of a significant change in one direction is different than calculating the possibility of a significant change in one of two directions.

Solution

We solve this problem by having other people check the experimental setup. Before your experiment can start, it is looked over by a group of experts, who could find mistakes in the experimental setup or tips on how to improve the experiment. The experiment is not allowed to start before the ok is given.

C.1.3 Guardrail metrics (02c)

Explanation

Guardrail metrics enable the experimenter to keep track of extra metrics to make sure the experiments do not harm the overall goal of the business. An example of this is keeping track of the number of sales when running an experiment in which changes the color of a single button. If guardrail metrics are not in place, this could lead to unwanted results, even though the experiment is a success. To continue the example, the color might lead to more people clicking the button, but the sales going down.

Solution

To resolve this problem, we add the possibility for adding guardrail metrics and have the experimenter choose which metrics they also want to measure besides the default metric.

C.1.4 Technical debt (03a)

Explanation

Technical debt is the implied cost of additional rework in the future by choosing an easy (limited) solution now instead of taking a better approach that would take longer.

Solution

To solve this, we show a quick list of things that you should think about when running experiments, we remind you of things that can go wrong while experimenting

C.1.5 Competitor safety (03b)

Explanation

Running AB tests in production might give a competitor an indication of a new product (line) being developed and give competitors more time to catch up or beat time to market.

Solution

To solve this, we show a quick list of things that you should think about when running experiments, we remind you of things that can go wrong while experimenting

C.1.6 Churning users (03c)

Explanation

If, for example, an AB test gives customers a bad onboarding experience, they might not choose to join after all, making ING lose a customer.

Solution

To solve this, we show a quick list of things that you should think about when running experiments, we remind you of things that can go wrong while experimenting

C.1.7 Minimum effect size (04a)

Explanation

The minimum effect size enables you to, with statistics, determine how many users you need to test with to measure a 5%, 10% or 100% change. After all, it is easier to see if a new version is 100% better than 1%, as the cause of a 1% change is more likely randomness if you have a small number of participants. The minimum effect size is part of the calculation of how long the experiment should run. If this is not calculated, the experiment could run longer than necessary or too short to get actual results.

Solution

To aid people with setting the correct duration of the experiment, we introduce a screen where users are able to fill in all the information they have. How many people visit the page that is being experimented on, how detailed the result must be, etc. This will then automatically determine how many users need to participate in the experiment and therefore how long the experiment should last.

C.1.8 Minimum duration (04b)

Explanation

The minimum duration is how long an experiment should at least run to gain significance. If this is not calculated, the experiment could run longer than neccesary or too short too short to get actual results.

Solution

To aid people with setting the correct duration of the experiment, we introduce a screen where users are able to fill in all the information they have. How many people visit the

page that is being experimented on, how detailed the result must be, etc. This will then automatically determine how many users need to participate in the experiment and therefore how long the experiment should last.

C.1.9 Withholding results (04c)

Explanation

If intermediate results are visible for experimenters, it might lead to them becoming biased towards a solution. If version A starts out as better this might lead to "rooting" for this version to win.

Solution

To resolve this problem, we can blur screen until the required number of user and significance has been reached. To aid users, we unblur the screen after users click yes on a pop-up where they are explained why looking at results would be a bad idea

C.1.10 Simultaneous experiments (05a)

Explanation

If users are being exposed to multiple AB-tests at once, this could lead to invalid results. The most extreme example is when one test tests a different color for a button, while another test removes the button.

Solution

To solve this, we show a quick list of things that you should think about when running experiments, we remind you of things that can go wrong while experimenting

C.1.11 Failure checks (06a)

Explanation

Sometimes an implementation of a new feature will make the application crash. Failure checks then make sure to shut the experiment down when this happens.

Solution

To resolve this problem, we add the possibility for adding guardrail metrics and have the experimenter choose which metrics they also want to measure besides the default metric.

C.1.12 Number of changes (09a)

Explanation

Usually an experiment consists of a version A and B. However sometimes an experimenter might want to measure multiple things at the same time. For example by giving every button on a page a specific color. Test 1: Button A is green, Button B is yellow Test 2: Button A is red, Button B is also red

However, these tests are usually more difficult and should be double checked if there is the possibility to see if the experiment can be done in the A/B fashion, as this takes less time and effort.

Solution

We solve this problem by having other people check the experimental setup. Before your experiment can start, it is looked over by a group of experts, who could find mistakes in the experimental setup or tips on how to improve the experiment. The experiment is not allowed to start before the ok is given.

C.1.13 Day of week effect (11b)

Explanation

Every day a different segment of the users log on to the website. For example, during the week, people who work might not have time to look at a new mortage, while during the weekend they do. Starting and ending the experiment on a different weekday might change the distribution of the visitors of the page so much it is no longer representing the actual distribution of the users visiting the website.

Solution

To resolve this problem, we present an experimenter with a warning when they want to stop an experiment on another weekday than the weekday the experiment was started. We add the options to let the experiment continue to the original end date, the next full week or stop it immediately.

C.1.14 Confirm winning variant (12b)

Explanation

After an experiment has concluded, the winning version is usually put in production and the experiment cleaned up. When an experimenter doesn't make sure the winning version of the experiment works as expected, this could introduce unintentional bugs or changes.

Solution

After the experiment has concluded, we ask the experimenter to respond to a couple of questions related to the experiment they ran. What was their conclusion? What did they

learn about experiments in general? What will be their followup experiments? How does this relate to their overall goal? Did they clean up their experiment and check if it was working as expected?

C.1.15 Encourage more experiments (13a)

Explanation

An experiment has the goal to see if a hypothesis is true. However, while examining/discussing the results interesting insights or ideas for new experiments can be discovered. By not providing the space for an experimenter, the business might miss a great new idea.

Solution

After the experiment has concluded, we ask the experimenter to respond to a couple of questions related to the experiment they ran. What was their conclusion? What did they learn about experiments in general? What will be their followup experiments? How does this relate to their overall goal? Did they clean up their experiment and check if it was working as expected?

C.1.16 Higher level question (14b)

Explanation

Experiments are not run in a vacuum. Most of the time they try to answer a higher business level question. If no infrastructure in place to think about these questions, it might be the case that the product ends up in a local optimum. It is therefore good to be able to group these experiments, so that this relation is clear to other users.

Solution

After the experiment has concluded, we ask the experimenter to respond to a couple of questions related to the experiment they ran. What was their conclusion? What did they learn about experiments in general? What will be their followup experiments? How does this relate to their overall goal? Did they clean up their experiment and check if it was working as expected?

C.1.17 Share learnings (15a)

Explanation

Sharing experiences when experimenting can be very valuable to other experimenters. Think of helping experimenters answer the following questions: What are common problems? How can experimenters test better? If these are not shared, many can make the same mistakes, costing valuable time and effort from the experimenters.

Solution

After the experiment has concluded, we ask the experimenter to respond to a couple of questions related to the experiment they ran. What was their conclusion? What did they learn about experiments in general? What will be their followup experiments? How does this relate to their overall goal? Did they clean up their experiment and check if it was working as expected?

C.1.18 Rerun experiment when results are marginal (20a)

Explanation

If the difference (significance/results) between two versions is low, rerunning the experiment could shed more light on which version performs better.

Solution

A button to rerun the experiment is added, which the experimenter can press if he/she would like to rerun their experiment to double check the results. This button is made more prominent if the significance of the experiment was low or if the experiment is sampled to check if the experiments are still running correctly

C.1.19 Validation of experiment (20b)

Explanation

Reproducibility is one the most important factors of science. If rerunning an experiment results in a different outcome, the setup was incorrect or there are other influences which were not accounted for. Therefore experiments which ran on the platform should be reproducible.

Solution

A button to rerun the experiment is added, which the experimenter can press if he/she would like to rerun their experiment to double check the results. This button is made more prominent if the significance of the experiment was low or if the experiment is sampled to check if the experiments are still running correctly

Appendix D

Statistic tables

The following tables are an n-n comparison of the pitfalls.

D.1 Effect Size

For this work, for each combination of pitfalls the effect size was calculated [48]. See the shorter version in section 6.2. The values in bold have a large effect size (> 0.43).

	01b	02b	02c	03a	03b	03c	04a	04b	04c	05a	06a	09a	11b	12b	13a	14b	15a	20a	20b
01b (Falsifiable hypothesis)	0	0	0.099265	0.180995	0.025735	0.224265	0.058824	0.268908	0.088235	0.294118	0.016807	0.197479	0.011765	0	0.189542	0.016807	0.224265	0.117647	0.176471
02b (Direction of change in hypothesis)	0	0	0.099265	0.180995	0.025735	0.224265	0.058824	0.268908	0.088235	0.294118	0.016807	0.197479	0.011765	0	0.189542	0.016807	0.224265	0.117647	0.176471
02c (Guardrail metrics)	0.099265	0.099265	0	0.081731	0.125	0.125	0.040441	0.169643	0.1875	0.194853	0.116071	0.098214	0.0875	0.099265	0.090278	0.116071	0.125	0.018382	0.077206
03a (Technical debt)	0.180995	0.180995	0.081731	0	0.206731	0.043269	0.122172	0.087912	0.269231	0.113122	0.197802	0.016484	0.169231	0.180995	0.008547	0.197802	0.043269	0.063348	0.004525
03b (Competitor safety)	0.025735	0.025735	0.125	0.206731	0	0.25	0.084559	0.294643	0.0625	0.319853	0.008929	0.223214	0.0375	0.025735	0.215278	0.008929	0.25	0.143382	0.202206
03c (Churning users)	0.224265	0.224265	0.125	0.043269	0.25	0	0.165441	0.044643	0.3125	0.069853	0.241071	0.026786	0.2125	0.224265	0.034722	0.241071	0	0.106618	0.047794
04a (Minimum effect size)	0.058824	0.058824	0.040441	0.122172	0.084559	0.165441	0	0.210084	0.147059	0.235294	0.07563	0.138655	0.047059	0.058824	0.130719	0.07563	0.165441	0.058824	0.117647
04b (Calculate minimum duration)	0.268908	0.268908	0.169643	0.087912	0.294643	0.044643	0.210084	0	0.357143	0.02521	0.285714	0.071429	0.257143	0.268908	0.079365	0.285714	0.044643	0.151261	0.092437
04c (Witholding results)	0.088235	0.088235	0.1875	0.269231	0.0625	0.3125	0.147059	0.357143	0	0.382353	0.071429	0.285714	0.1	0.088235	0.277778	0.071429	0.3125	0.205882	0.264706
05a (Simultaneous experiments)	0.294118	0.294118	0.194853	0.113122	0.319853	0.069853	0.235294	0.02521	0.382353	0	0.310924	0.096639	0.282353	0.294118	0.104575	0.310924	0.069853	0.176471	0.117647
06a (Failure checks)	0.016807	0.016807	0.116071	0.197802	0.008929	0.241071	0.07563	0.285714	0.071429	0.310924	0	0.214286	0.028571	0.016807	0.206349	0	0.241071	0.134454	0.193277
09a (Number of changes)	0.197479	0.197479	0.098214	0.016484	0.223214	0.026786	0.138655	0.071429	0.285714	0.096639	0.214286	0	0.185714	0.197479	0.007937	0.214286	0.026786	0.079832	0.021008
11b (Full-week effects)	0.011765	0.011765	0.0875	0.169231	0.0375	0.2125	0.047059	0.257143	0.1	0.282353	0.028571	0.185714	0	0.011765	0.177778	0.028571	0.2125	0.105882	0.164706
12b (Confirm winning variant)	0	0	0.099265	0.180995	0.025735	0.224265	0.058824	0.268908	0.088235	0.294118	0.016807	0.197479	0.011765	0	0.189542	0.016807	0.224265	0.117647	0.176471
13a (Encourage more experiments)	0.189542	0.189542	0.090278	0.008547	0.215278	0.034722	0.130719	0.079365	0.277778	0.104575	0.206349	0.007937	0.177778	0.189542	0	0.206349	0.034722	0.071895	0.013072
14b (Higher level question)	0.016807	0.016807	0.116071	0.197802	0.008929	0.241071	0.07563	0.285714	0.071429	0.310924	0	0.214286	0.028571	0.016807	0.206349	0	0.241071	0.134454	0.193277
15a (Share learning)	0.224265	0.224265	0.125	0.043269	0.25	0	0.165441	0.044643	0.3125	0.069853	0.241071	0.026786	0.2125	0.224265	0.034722	0.241071	0	0.106618	0.047794
20a (Rerun experiment when results are marginal)	0.117647	0.117647	0.018382	0.063348	0.143382	0.106618	0.058824	0.151261	0.205882	0.176471	0.134454	0.079832	0.105882	0.117647	0.071895	0.134454	0.106618	0	0.058824
20b (Validation of experiments)	0.176471	0.176471	0.077206	0.004525	0.202206	0.047794	0.117647	0.092437	0.264706	0.117647	0.193277	0.021008	0.164706	0.176471	0.013072	0.193277	0.047794	0.058824	0

Table D.1: Before reading this, were you aware this problem existed?

	01b	02b	02c	03a	03b	03c	04a	04b	04c	05a	06a	09a	11b	12b	13a	14b	15a	20a	20b
01b (Falsifiable hypothesis)	0	0.176471	0.073529	**0.59276**	0.011029	0.261029	0.176471	0.347339	0.073529	0.235294	0.061625	0.394958	**0.823529**	0	0.212418	0.180672	**0.511029**	0.294118	0.117647
02b (Direction of change in hypothesis)	0.176471	0	0.102941	0.41629	0.165441	0.084559	0	0.170868	0.102941	0.058824	0.114846	0.218487	**0.647059**	0.176471	0.035948	0.004202	0.334559	0.117647	0.058824
02c (Guardrail metrics)	0.073529	0.102941	0	**0.519231**	0.0625	0.1875	0.102941	0.27381	0	0.161765	0.011905	0.321429	**0.75**	0.073529	0.138889	0.107143	**0.4375**	0.220588	0.044118
03a (Technical debt)	**0.59276**	0.41629	**0.519231**	0	**0.581731**	0.331731	0.41629	0.245421	**0.519231**	0.357466	**0.531136**	0.197802	0.230769	**0.59276**	0.380342	0.412088	0.081731	0.298643	**0.475113**
03b (Competitor safety)	0.011029	0.165441	0.0625	**0.581731**	0	0.25	0.165441	0.33631	0.0625	0.224265	0.050595	0.383929	**0.8125**	0.011029	0.201389	0.169643	**0.5**	0.283088	0.106618
03c (Churning users)	0.261029	0.084559	0.1875	0.331731	0.25	0	0.084559	0.08631	0.1875	0.025735	0.199405	0.133929	**0.5625**	0.261029	0.048611	0.080357	0.25	0.033088	0.143382
04a (Minimum effect size)	0.176471	0	0.102941	0.41629	0.165441	0.084559	0	0.170868	0.102941	0.058824	0.114846	0.218487	**0.647059**	0.176471	0.035948	0.004202	0.334559	0.117647	0.058824
04b (Calculate minimum duration)	0.347339	0.170868	0.27381	0.245421	0.33631	0.08631	0.170868	0	0.27381	0.112045	0.285714	0.047619	**0.47619**	0.347339	0.134921	0.166667	0.16369	0.055221	0.229692
04c (Witholding results)	0.073529	0.102941	0	**0.519231**	0.0625	0.1875	0.102941	0.27381	0	0.161765	0.011905	0.321429	**0.75**	0.073529	0.138889	0.107143	**0.4375**	0.220588	0.044118
05a (Simultaneous experiments)	0.235294	0.058824	0.161765	0.357466	0.224265	0.025735	0.058824	0.112045	0.161765	0	0.173669	0.159664	**0.588235**	0.235294	0.022876	0.054622	0.275735	0.058824	0.117647
06a (Failure checks)	0.061625	0.114846	0.011905	**0.531136**	0.050595	0.199405	0.114846	0.285714	0.011905	0.173669	0	0.333333	**0.761905**	0.061625	0.150794	0.119048	**0.449405**	0.232493	0.056022
09a (Number of changes)	0.394958	0.218487	0.321429	0.197802	0.383929	0.133929	0.218487	0.047619	0.321429	0.159664	0.333333	0	0.428571	0.394958	0.18254	0.214286	0.116071	0.10084	0.277311
11b (Full-week effects)	**0.823529**	**0.647059**	**0.75**	0.230769	**0.8125**	**0.5625**	**0.647059**	**0.47619**	**0.75**	**0.588235**	**0.761905**	0.428571	0	**0.823529**	**0.611111**	**0.642857**	0.3125	**0.529412**	**0.705882**
12b (Confirm winning variant)	0	0.176471	0.073529	**0.59276**	0.011029	0.261029	0.176471	0.347339	0.073529	0.235294	0.061625	0.394958	**0.823529**	0	0.212418	0.180672	**0.511029**	0.294118	0.117647
13a (Encourage more experiments)	0.212418	0.035948	0.138889	0.380342	0.201389	0.048611	0.035948	0.134921	0.138889	0.022876	0.150794	0.18254	**0.611111**	0.212418	0	0.031746	0.298611	0.081699	0.094771
14b (Higher level question)	0.180672	0.004202	0.107143	0.412088	0.169643	0.080357	0.004202	0.166667	0.107143	0.054622	0.119048	0.214286	**0.642857**	0.180672	0.031746	0	0.330357	0.113445	0.063025
15a (Share learning)	**0.511029**	0.334559	**0.4375**	0.081731	**0.5**	0.25	0.334559	0.16369	**0.4375**	0.275735	**0.449405**	0.116071	0.3125	**0.511029**	0.298611	0.330357	0	0.216912	0.393382
20a (Rerun experiment when results are marginal)	0.294118	0.117647	0.220588	0.298643	0.283088	0.033088	0.117647	0.053221	0.220588	0.058824	0.232493	0.10084	**0.529412**	0.294118	0.081699	0.113445	0.216912	0	0.176471
20b (Validation of experiments)	0.117647	0.058824	0.044118	**0.475113**	0.106618	0.143382	0.058824	0.229692	0.044118	0.117647	0.056022	0.277311	**0.705882**	0.117647	0.094771	0.063025	0.393382	0.176471	0

Table D.2: Have you ever encountered this problem yourself?

	01b	02b	02c	03a	03b	03c	04a	04b	04c	05a	06a	09a	11b	12b	13a	14b	15a	20a	20b
01b (Falsifiable hypothesis)	0	0.031142	0.257353	**0.113122**	**0.555147**	0.301471	0.228374	0.142857	0.128676	0.190311	0.022409	0.07563	**0.011765**	0.031142	0.156863	0.063025	**0.0625**	0.128028	0.034602
02b (Direction of change in hypothesis)	0.031142	0	0.283088	0.158371	**0.544118**	0.330882	0.273356	0.182073	0.102941	0.228374	0.002801	0.117647	**0.058824**	0	0.140523	0.096639	0.036765	0.096886	0.079585
02c (Guardrail metrics)	0.257353	0.283088	0	**0.139423**	**0.761719**	0.039063	0.058824	0.127976	0.371094	0.077206	0.261905	0.178571	**0.2625**	0.283088	0.388889	0.191964	**0.316406**	0.386029	0.25
03a (Technical debt)	**0.113122**	0.158371	**0.139423**	0	**0.735577**	0.182692	0.140271	0.021978	**0.331731**	0.113122	**0.168498**	0.06044	0.061538	**0.158371**	0.333333	0	0.235577	0.339367	**0.067873**
03b (Competitor safety)	**0.555147**	**0.544118**	**0.761719**	**0.735577**	0	**0.828125**	**0.827206**	**0.752976**	**0.53125**	**0.720588**	**0.535714**	**0.71875**	**0.6125**	**0.544118**	0.402778	**0.549107**	**0.554688**	**0.588235**	**0.672794**
03c (Churning users)	0.301471	0.330882	0.039063	0.182692	**0.828125**	0	0.095588	0.172619	**0.4375**	0.110294	0.3125	0.227679	**0.3**	0.330882	**0.447917**	0.223214	0.375	**0.455882**	0.294118
04a (Minimum effect size)	0.228374	0.273356	0.058824	0.140271	**0.827206**	0.095588	0	0.103641	**0.430147**	0.010381	0.263305	0.193277	**0.223529**	0.273356	0.411765	0.121849	0.334559	**0.453287**	0.221453
04b (Calculate minimum duration)	0.142857	0.182073	0.127976	0.021978	**0.752976**	0.172619	0.103641	0	0.324405	0.07563	0.179138	0.07483	**0.114286**	0.182073	0.335979	0.047619	0.241071	0.333333	0.109244
04c (Witholding results)	0.128676	0.102941	0.371094	**0.331731**	**0.53125**	**0.4375**	**0.430147**	0.324405	0	0.352941	0.089286	0.28125	**0.2125**	0.102941	0.090278	0.191964	**0.054688**	0.029412	0.231618
05a (Simultaneous experiments)	0.190311	0.228374	0.077206	0.113122	**0.720588**	0.110294	0.010381	0.07563	0.352941	0	0.212885	0.155462	**0.2**	0.228374	0.339869	0.105042	0.272059	0.373702	0.190311
06a (Failure checks)	0.022409	0.002801	0.261905	**0.168498**	**0.535714**	0.3125	0.263305	0.179138	0.089286	0.212885	0	0.129252	**0.057143**	0.002801	0.142857	0.078231	**0.035714**	0.078431	0.078431
09a (Number of changes)	0.07563	0.117647	0.178571	0.06044	**0.71875**	0.227679	0.193277	0.07483	0.28125	0.155462	0.129252	0	0.014286	0.117647	0.301587	0.030612	0.191964	0.281513	0.016807
11b (Full-week effects)	**0.011765**	**0.058824**	**0.2625**	0.061538	**0.6125**	**0.3**	**0.223529**	**0.114286**	**0.2125**	**0.2**	**0.057143**	0.014286	0	**0.058824**	**0.2**	**0.085714**	0.1125	**0.223529**	**0.011765**
12b (Confirm winning variant)	0.031142	0	0.283088	**0.158371**	**0.544118**	0.330882	0.273356	0.182073	0.102941	0.228374	0.002801	0.117647	**0.058824**	0	0.140523	0.096639	**0.036765**	0.096886	0.079585
13a (Encourage more experiments)	0.156863	0.140523	0.388889	0.333333	0.402778	**0.447917**	0.411765	0.335979	0.090278	0.339869	0.142857	0.301587	**0.2**	0.140523	0	0.186508	0.125	0.117647	0.235294
14b (Higher level question)	0.063025	0.096639	0.191964	0	**0.549107**	0.223214	0.121849	0.047619	0.191964	0.105042	0.078231	0.030612	**0.085714**	0.096639	0.186508	0	0.120536	0.205882	0.063025
15a (Share learning)	**0.0625**	0.036765	**0.316406**	0.235577	**0.554688**	0.375	0.334559	0.241071	**0.054688**	0.272059	**0.035714**	0.191964	0.1125	**0.036765**	0.125	0.120536	0	0.036765	0.136029
20a (Rerun experiment when results are marginal)	0.128028	0.096886	0.386029	0.339367	**0.588235**	**0.455882**	**0.453287**	0.333333	0.029412	0.373702	0.078431	0.281513	**0.223529**	0.096886	0.117647	0.205882	0.036765	0	0.238754
20b (Validation of experiments)	0.034602	0.079585	0.25	**0.067873**	**0.672794**	0.294118	0.221453	0.109244	0.231618	0.190311	0.078431	0.016807	**0.011765**	0.079585	0.235294	0.063025	0.136029	0.238754	0

Table D.3: How severe do you think this problem is?

	01b	02b	02c	03a	03b	03c	04a	04b	04c	05a	06a	09a	11b	12b	13a	14b	15a	20a	20b
01b (Falsifiable hypothesis)	0	0.058824	0.161765	**0.049774**	**0.088235**	0.150735	0.352941	0.411765	0.025735	0.058824	0.173669	0.054622	**0.011765**	0.235294	0.022876	0.088235	**0.036765**	0.117647	0.176471
02b (Direction of change in hypothesis)	0.058824	0	0.102941	0.108597	**0.147059**	0.209559	0.294118	0.352941	0.084559	0.117647	0.114846	0.004202	**0.047059**	0.294118	0.035948	0.147059	0.022059	0.058824	0.117647
02c (Guardrail metrics)	0.161765	0.102941	0	**0.211538**	**0.25**	0.3125	0.191176	0.25	0.1875	0.220588	0.011905	0.107143	**0.15**	0.397059	0.138889	0.25	**0.125**	0.044118	0.014706
03a (Technical debt)	**0.049774**	0.108597	**0.211538**	0	**0.038462**	0.100962	0.402715	**0.461538**	**0.024038**	0.00905	**0.223443**	0.104396	0.061538	**0.18552**	0.07265	0.038462	0.086538	0.16742 1	**0.226244**
03b (Competitor safety)	**0.088235**	**0.147059**	**0.25**	**0.038462**	0	**0.0625**	**0.441176**	**0.5**	**0.0625**	**0.029412**	**0.261905**	**0.142857**	**0.1**	**0.147059**	0.111111	**0**	**0.125**	**0.205882**	**0.264706**
03c (Churning users)	0.150735	0.209559	0.3125	0.100962	**0.0625**	0	**0.503676**	**0.5625**	**0.125**	0.091912	0.324405	0.205357	**0.1625**	0.084559	**0.173611**	0.0625	0.1875	**0.268382**	0.327206
04a (Minimum effect size)	0.352941	0.294118	0.191176	0.402715	**0.441176**	**0.503676**	0	0.058824	**0.378676**	0.411765	0.179272	0.298319	**0.341176**	**0.588235**	0.330065	**0.441176**	0.316176	**0.235294**	0.176471
04b (Calculate minimum duration)	0.411765	0.352941	0.25	**0.461538**	**0.5**	**0.5625**	0.058824	0	**0.4375**	**0.470588**	0.238095	0.357143	**0.4**	**0.647059**	0.388889	**0.5**	0.375	0.294118	0.235294
04c (Witholding results)	0.025735	0.084559	0.1875	**0.024038**	**0.0625**	**0.125**	**0.378676**	**0.4375**	0	0.033088	0.199405	0.080357	**0.0375**	0.209559	0.048611	0.0625	**0.0625**	0.143382	0.202206
05a (Simultaneous experiments)	0.058824	0.117647	0.220588	0.00905	**0.029412**	0.091912	0.411765	**0.470588**	0.033088	0	0.232493	0.113445	**0.070588**	0.176471	0.081699	0.029412	0.095588	0.176471	0.235294
06a (Failure checks)	0.173669	0.114846	0.011905	**0.223443**	**0.261905**	0.324405	0.179272	0.238095	0.199405	0.232493	0	0.119048	**0.161905**	0.408964	0.150794	0.261905	**0.136905**	0.056022	0.002801
09a (Number of changes)	0.054622	0.004202	0.107143	0.104396	**0.142857**	0.205357	0.298319	0.357143	0.080357	0.113445	0.119048	0	0.042857	0.289916	0.031746	0.142857	0.017857	0.063025	0.121849
11b (Full-week effects)	**0.011765**	**0.047059**	**0.15**	0.061538	**0.1**	**0.1625**	**0.341176**	**0.4**	**0.0375**	**0.070588**	**0.161905**	0.042857	0	**0.247059**	**0.011111**	**0.1**	0.025	**0.105882**	**0.164706**
12b (Confirm winning variant)	0.235294	0.294118	0.397059	**0.18552**	**0.147059**	0.084559	**0.588235**	**0.647059**	0.209559	0.176471	0.408964	0.289916	**0.247059**	0	0.25817	0.147059	**0.272059**	0.352941	0.411765
13a (Encourage more experiments)	0.022876	0.035948	0.138889	0.07265	0.111111	**0.173611**	0.330065	0.388889	0.048611	0.081699	0.150794	0.031746	**0.011111**	0.25817	0	0.111111	0.013889	0.094771	0.153595
14b (Higher level question)	0.088235	0.147059	0.25	0.038462	**0**	0.0625	**0.441176**	**0.5**	0.0625	0.029412	0.261905	0.142857	**0.1**	0.147059	0.111111	0	0.125	0.205882	0.264706
15a (Share learning)	**0.036765**	0.022059	**0.125**	0.086538	**0.125**	0.1875	0.316176	0.375	**0.0625**	0.095588	**0.136905**	0.017857	0.025	**0.272059**	0.013889	0.125	0	0.080882	0.139706
20a (Rerun experiment when results are marginal)	0.117647	0.058824	0.044118	0.167421	**0.205882**	**0.268382**	**0.235294**	0.294118	0.143382	0.176471	0.056022	0.063025	**0.105882**	0.352941	0.094771	0.205882	0.080882	0	0.058824
20b (Validation of experiments)	0.176471	0.117647	0.014706	**0.226244**	**0.264706**	0.327206	0.176471	0.235294	0.202206	0.235294	0.002801	0.121849	**0.164706**	0.411765	0.153595	0.264706	0.139706	0.058824	0

Table D.4: Do you think this solution solves this problem?

	01b	02b	02c	03a	03b	03c	04a	04b	04c	05a	06a	09a	11b	12b	13a	14b	15a	20a	20b
01b (Falsifiable hypothesis)	0.00000	0.003 46	0.35662	**0.14480**	**0.09926**	0.16912	**0.52249**	**0.53501**	0.01103	0.05536	0.25770	0.00420	**0.02353**	0.00000	0.14052	0.15546	**0.15441**	0.36678	0.41176
02b (Direction of change in hypothesis)	0.00346	0.00000	0.38235	0.19457	**0.12868**	0.19485	**0.55017**	**0.54622**	0.00368	0.03114	0.27731	0.01681	**0.01176**	0.01730	0.18301	0.18908	0.19118	0.39792	0.42907
02c (Guardrail metrics)	0.35662	0.38235	0.00000	**0.28365**	**0.36719**	0.23828	0.23897	0.24107	0.39844	**0.53309**	0.12500	**0.44196**	**0.53750**	**0.45956**	0.26736	0.24554	**0.24609**	0.01838	0.06985
03a (Technical debt)	**0.14480**	0.19457	**0.28365**	0.00000	**0.18750**	0.01923	**0.65611**	**0.64103**	**0.22596**	0.38914	**0.17582**	0.24725	**0.43077**	**0.31674**	0.01282	0.01648	0.06731	0.36199	**0.41176**
03b (Competitor safety)	**0.09926**	**0.12868**	**0.36719**	**0.18750**	0.00000	**0.14063**	**0.65809**	**0.60714**	**0.15625**	**0.19118**	**0.25000**	**0.20982**	**0.17500**	**0.14706**	0.17361	**0.15625**	**0.11328**	**0.43382**	**0.44118**
03c (Churning users)	0.16912	0.19485	0.23828	0.01923	**0.14063**	0.00000	**0.51103**	**0.50298**	**0.21094**	0.32353	0.12202	0.24107	**0.32500**	0.25735	**0.01389**	0.00446	0.03516	**0.27941**	0.32353
04a (Minimum effect size)	**0.52249**	**0.55017**	0.23897	**0.65611**	**0.65809**	**0.51103**	0.00000	0.06162	**0.55515**	**0.82007**	0.36975	**0.57143**	**0.85882**	**0.70934**	**0.60784**	**0.54622**	**0.53309**	**0.25260**	0.14533
04b (Calculate minimum duration)	**0.53501**	**0.54622**	0.24107	**0.64103**	**0.60714**	**0.50298**	0.06162	0.00000	**0.54167**	**0.76471**	0.38322	**0.56803**	**0.77143**	**0.65826**	**0.60053**	**0.53741**	**0.48810**	0.27731	0.17647
04c (Witholding results)	0.01103	0.00368	0.39844	**0.22596**	**0.15625**	**0.21094**	**0.55515**	**0.54167**	0.00000	0.00735	0.28571	0.03125	**0.05000**	0.03676	0.20833	0.20982	**0.22266**	0.41176	**0.43382**
05a (Simultaneous experiments)	0.05536	0.03114	**0.53309**	0.38914	**0.19118**	0.32353	**0.82007**	**0.76471**	0.00735	0.00000	0.42857	0.03361	**0.02353**	0.03466	0.36601	0.34034	0.28309	**0.60900**	**0.60900**
06a (Failure checks)	0.25770	0.27731	0.12500	**0.17582**	**0.25000**	0.12202	0.36975	0.38322	0.28571	0.42857	0.00000	0.31293	**0.42857**	0.35014	0.16138	0.13605	**0.14286**	0.14286	0.20448
09a (Number of changes)	0.00420	0.01681	**0.44196**	0.24725	**0.20982**	0.24107	**0.57143**	**0.56803**	0.03125	0.03361	0.31293	0.00000	0.11429	0.08403	0.22619	0.23469	0.27232	**0.43697**	**0.46639**
11b (Full-week effects)	**0.02353**	**0.01176**	**0.53750**	**0.43077**	**0.17500**	**0.32500**	**0.85882**	**0.77143**	**0.05000**	**0.02353**	**0.42857**	0.11429	0.00000	**0.00000**	**0.40000**	**0.35714**	0.28750	**0.63529**	**0.61176**
12b (Confirm winning variant)	0.00000	0.01730	**0.45956**	**0.31674**	**0.14706**	0.25735	**0.70934**	**0.65826**	0.03676	0.03460	0.35014	0.08403	**0.00000**	0.00000	0.29412	0.27311	**0.23897**	**0.51903**	**0.51903**
13a (Encourage more experiments)	0.14052	0.18301	0.26736	0.01282	0.17361	**0.01389**	**0.60784**	**0.60053**	0.20833	0.36601	0.16138	0.22619	**0.40000**	0.29412	0.00000	0.00794	0.06250	0.33333	0.38562
14b (Higher level question)	0.15546	0.18908	0.24554	0.01648	**0.15625**	0.00446	**0.54622**	**0.53741**	0.20982	0.34034	0.13605	0.23469	**0.35714**	0.27311	0.00794	0.00000	0.04911	0.29832	0.34454
15a (Share learning)	**0.15441**	0.19118	**0.24609**	0.06731	**0.11328**	0.03516	**0.53309**	**0.48810**	**0.22266**	0.28309	**0.14286**	0.27232	0.28750	**0.23897**	0.06250	0.04911	0.00000	0.31250	0.32353
20a (Rerun experiment when results are marginal)	0.36678	0.39792	0.01838	0.36199	**0.43382**	**0.27941**	**0.25260**	0.27731	0.41176	**0.60900**	0.14286	**0.43697**	**0.63529**	**0.51903**	0.33333	0.29832	0.31250	0.00000	0.07612
20b (Validation of experiments)	0.41176	0.42907	0.06985	**0.41176**	**0.44118**	0.32353	0.14533	0.17647	**0.43382**	**0.60900**	0.20448	**0.46639**	**0.61176**	**0.51903**	0.38562	0.34454	0.32353	0.07612	0.00000

Table D.5: Will this solution help or hinder you when running experiments?

D.2 Mann–Whitney U test

For this work, a Mann–Whitney U test was performed. See the shorter version in section 6.2. Each bold value is a significant result (< 0.05) with a large effect size.

	01b (Falsifiable hypothesis)	02b (Direction of change in hypothesis)	02c (Guardrail metrics)	03a (Technical debt)	03b (Competitor safety)	03c (Churning users)	04a (Minimum effect size)	04b (Calculate minimum duration)	04c (Witholding results)	05a (Simultaneous experiments)	06a (Failure checks)	09a (Num
01b (Falsifiable hypothesis)	0.983888	0.983888	0.574268	0.317809	0.899587	0.174783	0.743498	0.067591	0.631153	0.058447	0.931629	0.259746
02b (Direction of change in hypothesis)	0.983888	0.983888	0.574268	0.317809	0.899587	0.174783	0.743498	0.067591	0.631153	0.058447	0.931629	0.259746
02c (Guardrail metrics)	0.574268	0.574268	0.981282	0.650613	0.48622	0.43626	0.825492	0.228805	0.298032	0.186141	0.488205	0.568978
03a (Technical debt)	0.317809	0.317809	0.650613	0.971992	0.263591	0.802695	0.493289	0.537269	0.151546	0.436802	0.256755	0.946308
03b (Competitor safety)	0.899587	0.899587	0.48622	0.263591	0.982511	0.139394	0.639799	0.050914	0.743762	0.044742	0.97151	0.212514
03c (Churning users)	0.174783	0.174783	0.43626	0.802695	0.139394	0.977773	0.304481	0.736863	0.070488	0.601814	0.13057	0.880827
04a (Minimum effect size)	0.743498	0.743498	0.825492	0.493289	0.639799	0.304481	0.983407	0.140316	0.412108	0.116456	0.652248	0.419435
04b (Calculate minimum duration)	0.067591	0.067591	0.228805	0.537269	0.050914	0.736863	0.140316	0.983448	0.021263	0.841089	0.044561	0.606127
04c (Witholding results)	0.631153	0.631153	0.298032	0.151546	0.743762	0.070488	0.412108	0.021263	0.982648	0.019832	0.68306	0.116884
05a (Simultaneous experiments)	0.058447	0.058447	0.186141	0.436802	0.044742	0.601814	0.116456	0.841089	0.019832	0.975391	0.039893	0.493311
06a (Failure checks)	0.931629	0.931629	0.488205	0.256755	0.97151	0.13057	0.652248	0.044561	0.68306	0.039893	0.988295	0.204398
09a (Number of changes)	0.259746	0.259746	0.568978	0.946308	0.212514	0.880827	0.419435	0.606127	0.116884	0.493311	0.204398	0.974229
11b (Full-week effects)	1	1	0.761855	0.525869	0.923342	0.371093	0.887949	0.215681	0.738597	0.177886	0.93944	0.467036
12b (Confirm winning variant)	0.983888	0.983888	0.574268	0.317809	0.899587	0.174783	0.743498	0.067591	0.631153	0.058447	0.931629	0.259746
13a (Encourage more experiments)	0.242255	0.242255	0.572639	0.977961	0.19423	0.824624	0.410968	0.53898	0.099798	0.433411	0.184198	0.978848
14b (Higher level question)	0.94449	0.94449	0.533799	0.298642	0.98069	0.165399	0.689364	0.065013	0.718388	0.056047	0.984335	0.244547
15a (Share learning)	0.174783	0.174783	0.43626	0.802695	0.139394	0.977773	0.304481	0.736863	0.070488	0.601814	0.13057	0.880827
20a (Rerun experiment when results are marginal)	0.492518	0.492518	0.927917	0.722613	0.411613	0.497075	0.733922	0.270732	0.242022	0.219077	0.409112	0.637552
20b (Validation of experiments)	0.288132	0.288132	0.640672	1	0.233799	0.760293	0.471644	0.484341	0.125009	0.389605	0.224625	0.912753

Table D.6: Before reading this, were you aware this problem existed?

	01b (Falsifiable hypothesis)	02b (Direction of change in hypothesis)	02c (Guardrail metrics)	03a (Technical debt)	03b (Competitor safety)	03c (Churning users)	04a (Minimum effect size)	04b (Calculate minimum duration)	04c (Witholding results)	05a (Simultaneous experiments)	06a (Failure checks)	09a (Num
01b (Falsifiable hypothesis)	0.979201	0.260003	0.629047	0.001536	0.957017	0.113385	0.260003	0.030754	0.629047	0.143898	0.66255	0.026008
02b (Direction of change in hypothesis)	0.260003	0.983407	0.541433	0.027628	0.304481	0.639799	0.983407	0.306633	0.541433	0.743498	0.454931	0.240519
02c (Guardrail metrics)	0.629047	0.541433	0.979964	0.006747	0.693154	0.281882	0.541433	0.101066	0.979964	0.343082	0.950675	0.082171
03a (Technical debt)	0.001536	0.027628	0.006747	0.971992	0.002267	0.080673	0.027628	0.164895	0.006747	0.057348	0.003021	0.298642
03b (Competitor safety)	0.957017	0.304481	0.693154	0.002267	0.977773	0.139394	0.304481	0.041051	0.693154	0.174783	0.730966	0.034384
03c (Churning users)	0.113385	0.639799	0.281882	0.080673	0.139394	0.982511	0.639799	0.62018	0.281882	0.899587	0.212294	0.486657
04a (Minimum effect size)	0.260003	0.983407	0.541433	0.027628	0.304481	0.639799	0.983407	0.306633	0.541433	0.743498	0.454931	0.240519
04b (Calculate minimum duration)	0.030754	0.306633	0.101066	0.164895	0.041051	0.62018	0.306633	0.988402	0.101066	0.508145	0.061667	0.799839
04c (Witholding results)	0.629047	0.541433	0.979964	0.006747	0.693154	0.281882	0.541433	0.101066	0.979964	0.343082	0.950675	0.082171
05a (Simultaneous experiments)	0.143898	0.743498	0.343082	0.057348	0.174783	0.899587	0.743498	0.508145	0.343082	0.983888	0.266232	0.396468
06a (Failure checks)	0.66255	0.454931	0.950675	0.003021	0.730966	0.212294	0.454931	0.061667	0.950675	0.266232	0.986401	0.051112
09a (Number of changes)	0.026008	0.240519	0.082171	0.298642	0.034384	0.486657	0.240519	0.799839	0.082171	0.396468	0.051112	0.978631
11b (Full-week effects)	0.001192	0.014685	0.004531	0.285833	0.001702	0.034264	0.014685	0.058729	0.004531	0.026193	0.002308	0.95895
12b (Confirm winning variant)	0.979201	0.260003	0.629047	0.001536	0.957017	0.113385	0.260003	0.030754	0.629047	0.143898	0.66255	0.026008
13a (Encourage more experiments)	0.17726	0.843746	0.406553	0.041109	0.212963	0.792565	0.843746	0.414844	0.406553	0.907143	0.323928	0.322968
14b (Higher level question)	0.271641	1	0.547197	0.037095	0.315616	0.677267	1	0.349168	0.547197	0.777826	0.464617	0.27583
15a (Share learning)	0.003705	0.061241	0.015569	0.650613	0.00537	0.167272	0.061241	0.330547	0.015569	0.122422	0.007394	0.533799
20a (Rerun experiment when results are marginal)	0.074335	0.505148	0.20238	0.109134	0.093797	0.867375	0.505148	0.760374	0.20238	0.74877	0.143104	0.598123
20b (Validation of experiments)	0.439406	0.733922	0.797396	0.01202	0.497075	0.411613	0.733922	0.164562	0.797396	0.492518	0.714704	0.131398

Table D.7: Have you ever encountered this problem yourself?

	01b (Falsifiable hypothesis)	02b (Direction of change in hypothesis)	02c (Guardrail metrics)	03a (Technical debt)	03b (Competitor safety)	03c (Churning users)	04a (Minimum effect size)	04b (Calculate minimum duration)	04c (Witholding results)	05a (Simultaneous experiments)	06a (Failure checks)	09a (Num
01b (Falsifiable hypothesis)	0.985874	0.88744	0.203373	0.599964	0.004248	0.134764	0.244844	0.448579	0.529755	0.337212	0.916199	0.725614
02b (Direction of change in hypothesis)	0.88744	0.985874	0.161494	0.454361	0.005055	0.100323	0.163229	0.331413	0.616531	0.248401	1	0.574856
02c (Guardrail metrics)	0.203373	0.161494	0.984504	0.522612	0.000154	0.860966	0.77942	0.506705	0.069446	0.710533	0.17273	0.400169
03a (Technical debt)	0.599964	0.454361	0.522612	0.976541	0.000403	0.394074	0.500827	0.924733	0.111878	0.595512	0.404805	0.784337
03b (Competitor safety)	0.004248	0.005055	0.000154	0.000403	0.981654	4.20E-05	2.97E-05	6.60E-05	0.007146	0.000249	0.003918	0.000448
03c (Churning users)	0.134764	0.100323	0.860966	0.394074	4.20E-05	0.984431	0.639292	0.365557	0.031402	0.589379	0.10248	0.276986
04a (Minimum effect size)	0.244844	0.163229	0.77942	0.500827	2.97E-05	0.639292	0.985384	0.578774	0.030273	0.971149	0.159663	0.341405
04b (Calculate minimum duration)	0.448579	0.331413	0.506705	0.924733	6.60E-05	0.365557	0.578774	0.989514	0.085914	0.691263	0.312407	0.707624
04c (Witholding results)	0.529755	0.616531	0.069446	0.111878	0.007146	0.031402	0.030273	0.085914	0.984357	0.078204	0.647326	0.171693
05a (Simultaneous experiments)	0.337212	0.248401	0.710533	0.595512	0.000249	0.589379	0.971149	0.691263	0.078204	0.985706	0.25854	0.452525
06a (Failure checks)	0.916199	1	0.17273	0.404805	0.003918	0.10248	0.159663	0.312407	0.647326	0.25854	0.989709	0.516566
09a (Number of changes)	0.725614	0.574856	0.400169	0.784337	0.000448	0.276986	0.341405	0.707624	0.171693	0.452525	0.516566	0.979608
11b (Full-week effects)	1	0.87135	0.396414	0.870086	0.027148	0.327106	0.453202	0.707855	0.49317	0.512646	0.867308	1
12b (Confirm winning variant)	0.88744	0.985874	0.161494	0.454361	0.005055	0.100323	0.163229	0.331413	0.616531	0.248401	1	0.574856
13a (Encourage more experiments)	0.427019	0.4776	0.050574	0.111203	0.032984	0.024165	0.03467	0.069252	0.657595	0.082296	0.444241	0.141813
14b (Higher level question)	0.773992	0.652405	0.369859	0.97967	0.006332	0.294848	0.561036	0.820669	0.370084	0.620907	0.703812	0.90448
15a (Share learning)	0.768086	0.868168	0.123326	0.271801	0.005141	0.066689	0.093955	0.206344	0.800962	0.177321	0.863052	0.363533
20a (Rerun experiment when results are marginal)	0.523253	0.630866	0.054125	0.096295	0.002669	0.022126	0.019372	0.070759	0.895108	0.056697	0.682774	0.161746
20b (Validation of experiments)	0.871905	0.693743	0.212243	0.753556	0.000587	0.13984	0.248769	0.558148	0.246585	0.329094	0.682896	0.949456

Table D.8: How severe do you think this problem is?

	01b (Falsifiable hypothesis)	02b (Direction of change in hypothesis)	02c (Guardrail metrics)	03a (Technical debt)	03b (Competitor safety)	03c (Churning users)	04a (Minimum effect size)	04b (Calculate minimum duration)	04c (Witholding results)	05a (Simultaneous experiments)	06a (Failure checks)	09a (Num
01b (Falsifiable hypothesis)	0.983888	0.743498	0.343082	0.807484	0.631153	0.405471	0.017959	0.001419	0.899587	0.74877	0.266232	0.777826
02b (Direction of change in hypothesis)	0.743498	0.983407	0.541433	0.570784	0.412108	0.242423	0.038952	0.003682	0.639799	0.505148	0.454931	1
02c (Guardrail metrics)	0.343082	0.541433	0.979964	0.252391	0.157031	0.080256	0.139296	0.018046	0.281882	0.20238	0.950675	0.547197
03a (Technical debt)	0.807484	0.570784	0.252391	0.976327	0.859328	0.612597	0.011994	0.000799	0.918947	0.98069	0.189691	0.608075
03b (Competitor safety)	0.631153	0.412108	0.157031	0.859328	0.982648	0.743762	0.005483	0.000332	0.743762	0.884252	0.106915	0.453068
03c (Churning users)	0.405471	0.242423	0.080256	0.612597	0.743762	0.982511	0.002099	0.000106	0.500171	0.617685	0.048833	0.279323
04a (Minimum effect size)	0.017959	0.038952	0.139296	0.011994	0.005483	0.002099	0.966308	0.289822	0.013366	0.007854	0.143272	0.042438
04b (Calculate minimum duration)	0.001419	0.003682	0.018046	0.000799	0.000332	0.000106	0.289822	1	0.000974	0.00052	0.019731	0.003881
04c (Witholding results)	0.899587	0.639799	0.281882	0.918947	0.743762	0.500171	0.013366	0.000974	0.982511	0.867375	0.212294	0.677267
05a (Simultaneous experiments)	0.74877	0.505148	0.20238	0.98069	0.884252	0.617685	0.007854	0.00052	0.867375	0.984114	0.143104	0.546213
06a (Failure checks)	0.266232	0.454931	0.950675	0.189691	0.106915	0.048833	0.143272	0.019731	0.212294	0.143104	0.986401	0.464617
09a (Number of changes)	0.777826	1	0.547197	0.608075	0.453068	0.279323	0.042438	0.003881	0.677267	0.546213	0.464617	0.97793
11b (Full-week effects)	1	0.887949	0.562083	0.863806	0.738597	0.567269	0.06527	0.003882	0.923342	0.820524	0.498425	0.911889
12b (Confirm winning variant)	0.182173	0.094979	0.025514	0.329714	0.412108	0.639799	0.00044	1.74E-05	0.242423	0.316914	0.012844	0.119141
13a (Encourage more experiments)	0.907143	0.843746	0.406553	0.708024	0.534215	0.328384	0.023326	0.001982	0.792565	0.644167	0.323928	0.874079
14b (Higher level question)	0.645322	0.430042	0.171038	0.866505	0.980863	0.754636	0.006488	0.00039	0.754636	0.890632	0.11978	0.469548
15a (Share learning)	0.848176	0.913969	0.467045	0.66233	0.496789	0.305737	0.030688	0.002699	0.739794	0.599053	0.383762	0.94048
20a (Rerun experiment when results are marginal)	0.492518	0.733922	0.797396	0.367029	0.242022	0.129976	0.080682	0.009147	0.411613	0.306507	0.714704	0.731601
20b (Validation of experiments)	0.288132	0.471644	0.941997	0.209437	0.125009	0.061218	0.160582	0.021948	0.233799	0.163494	1	0.479892

Table D.9: Do you think this solution solves this problem?

	01b (Falsifiable hypothesis)	02b (Direction of change in hypothesis)	02c (Guardrail metrics)	03a (Technical debt)	03b (Competitor safety)	03c (Churning users)	04a (Minimum effect size)	04b (Calculate minimum duration)	04c (Witholding results)	05a (Simultaneous experiments)	06a (Failure checks)	09a (Num
01b (Falsifiable hypothesis)	0.985744	1	0.074293	0.498454	0.631363	0.404969	0.006613	0.003085	0.97048	0.790705	0.16856	1
02b (Direction of change in hypothesis)	1	0.985901	0.055608	0.362371	0.529936	0.337544	0.004434	0.002595	1	0.887521	0.139557	0.951356
02c (Guardrail metrics)	0.074293	0.055608	0.983971	0.180611	0.066707	0.239759	0.211986	0.177059	0.049266	0.007064	0.512239	0.035878
03a (Technical debt)	0.498454	0.362371	0.180611	0.976196	0.372432	0.943624	0.00104	0.000966	0.297728	0.056189	0.374845	0.264831
03b (Competitor safety)	0.631363	0.529936	0.066707	0.372432	0.98409	0.492177	0.000787	0.001072	0.450519	0.33231	0.187653	0.326881
03c (Churning users)	0.404969	0.337544	0.239759	0.943624	0.492177	0.984357	0.008345	0.006138	0.307215	0.102664	0.523277	0.258027
04a (Minimum effect size)	0.006613	0.004434	0.211986	0.00104	0.000787	0.008345	0.984465	0.72473	0.00474	2.36E-05	0.039915	0.004837
04b (Calculate minimum duration)	0.003085	0.002595	0.177059	0.000966	0.001072	0.006138	0.72473	0.988326	0.00326	3.39E-05	0.02339	0.002883
04c (Witholding results)	0.97048	1	0.049266	0.297728	0.450519	0.307215	0.00474	0.003326	0.984611	0.985255	0.13521	0.898244
05a (Simultaneous experiments)	0.790705	0.887521	0.007064	0.056189	0.33231	0.102664	2.36E-05	3.39E-05	0.985255	0.985504	0.020962	0.886654
06a (Failure checks)	0.16856	0.139557	0.512239	0.374845	0.187653	0.523277	0.039915	0.02339	0.13521	0.020962	0.989514	0.113801
09a (Number of changes)	1	0.951356	0.035878	0.264831	0.326881	0.258027	0.004837	0.002883	0.898244	0.886654	0.113801	0.980954
11b (Full-week effects)	0.967965	1	0.067874	0.141037	0.565613	0.282428	0.002629	0.004853	0.899154	0.966707	0.138097	0.739546
12b (Confirm winning variant)	0.985928	0.943728	0.02019	0.127894	0.460761	0.198912	0.000264	0.000335	0.868019	0.870971	0.06107	0.699835
13a (Encourage more experiments)	0.469351	0.346354	0.168413	0.963852	0.368782	0.955889	0.000979	0.000703	0.292724	0.050799	0.369421	0.267348
14b (Higher level question)	0.461424	0.370173	0.241846	0.957645	0.458777	1	0.006295	0.004826	0.327728	0.095709	0.492478	0.28798
15a (Share learning)	0.444809	0.340534	0.211033	0.75954	0.569778	0.874241	0.006018	0.007699	0.272784	0.143144	0.454149	0.20065
20a (Rerun experiment when results are marginal)	0.061142	0.042933	0.939378	0.073794	0.028017	0.156555	0.176954	0.11533	0.039537	0.001717	0.439775	0.034397
20b (Validation of experiments)	0.035142	0.028546	0.729993	0.046101	0.026167	0.102134	0.438556	0.310781	0.029266	0.001885	0.266847	0.023315

Table D.10: Will this solution help or hinder you when running experiments?