



# Data Driven Decisions

Validating and Supporting a Continuous Experimentation Development Environment

Ernst Mulders



# Data Driven Decisions

Validating and Supporting a Continuous  
Experimentation Development Environment

by

Ernst Mulders

to obtain the degree of Master of Science  
at the Delft University of Technology,  
to be defended publicly on Wednesday December 4th, 2019 at 11:00 AM.

Student number: 1504673  
Project duration: December 1, 2018 – September 30, 2019  
Thesis committee: Assistant Prof. dr. ir. G. Gousios, TU Delft, supervisor  
Prof. dr. A. van Deursen, TU Delft  
Assistant Prof. dr. A. Katsifodimos, TU Delft  
K. Anderson, ING, company supervisor

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.



# Abstract

The number of conducted A/B tests is growing throughout companies in software development. Many of these companies develop their own in-house Experimentation Platform to support these experiments. In this thesis we identify factors that influence the trustworthiness and soundness of A/B tests by conducting a literature review. We discuss nineteen influential factors categorised as essentials and pitfalls. Using the data of 268 experiments from ING we verify the trustworthiness of ING's own Experiment Platform, and conclude that there is room for improvement. Finally, we provide a method for developers and engineers to consider these factors during the experimentation phase by modelling them into a questionnaire containing 67 questions. These questions are grouped into three categories, which are referred to as the three A's of A/B Test validation: Availability, Analysability and Accuracy. To help administer this questionnaire we introduce the first Open-Source toolkit for this matter: ABvalidator.



# Preface

In front of you lies my master thesis, the pinnacle of my educational life. Perseverance was needed to get to this point, and not to be permanently side tracked by the companies I (co-)founded. I owe a thank you to the people whom helped me get to were I am today.

First of all, I would like to thank my university supervisor Georgios Gousios for supporting me from afar, more than 8,500 km away, and trusting me with the capability of working individually but still being there when needed. Furthermore, I would like to thank Arie van Deursen for checking in on my progress whenever visiting ING.

From ING I would like to thank Kevin Anderson for all the time spent on providing me with feedback on my progress, helping me out with retrieving datasets and giving interesting quotes about the Experiment Office and the state of experimentation at ING. My gratitude also goes out to Joost Bosman from ING Research and Tech for giving me the opportunity to do my internship at ING. I would also like to thank ING's Tetris squad - especially Priyanka Gujar and Raf Ulrix - for their warm welcome to ING Belgium and for answering all my questions. My ING thank you round would not be complete without thanking Hennie Huijgens; thank you for opening all doors I needed at ING, the good conversations and laughs, feedback on my progress and pushing me to deliver something of value to the scientific community.

I also would like to thank Pim Nauts from Bol.com for providing me with input on the state of experimentation at Bol.com. Furthermore, I would like to very much thank Lukas Vermeer, director of experimentation at Booking.com, for making time to answer all of my questions and reviewing my paper submission.

This would all not have been possible without the support from friends and family. I'm very thankful for having supporting parents that always believed in my capabilities. And I would like to thank my brother for paving the way to the TU Delft, and supporting me whenever possible.

Finally I would like to thank Simone for pulling me through my study endeavours and motivating me in pursuing a CS master. And for the incredible support you managed to give me over these last months, while being pregnant of our son. Otis, although you are still so small, I thank you for everything you have already given me over these past 8 weeks. You make me proud, and I will do everything to give you the best life possible: starting with finishing this masters.

*Ernst Mulders  
Amsterdam, October 2019*





*Any figure that looks interesting  
or different is usually wrong  
– Twyman's law*



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.1.1	A/B Testing . . . . .	2
1.1.2	Reaching the user . . . . .	3
1.1.3	Experiment Platform . . . . .	3
1.1.4	Scope of A/B testing . . . . .	3
1.1.5	Data types and collection . . . . .	4
1.2	Terminology . . . . .	4
1.3	Problem Statement . . . . .	5
1.4	Approach . . . . .	5
1.5	Main Contributions . . . . .	6
<b>2</b>	<b>Essentials and Pitfalls in A/B Testing</b>	<b>7</b>
2.1	Methodology . . . . .	7
2.2	Related Work . . . . .	7
2.3	Essentials and pitfalls . . . . .	8
2.4	Essentials . . . . .	8
2.4.1	A/A testing . . . . .	8
2.4.2	Statistical significance tests . . . . .	9
2.4.3	Statistical Power and confidence . . . . .	9
2.4.4	Randomisation and Sample Ratio Mismatch . . . . .	10
2.4.5	User group assignments . . . . .	10
2.4.6	Data aggregation . . . . .	11
2.5	Pitfalls . . . . .	11
2.5.1	Overall Evaluation Criteria . . . . .	11
2.5.2	Primacy and newness . . . . .	12
2.5.3	Page speed/latency . . . . .	12
2.5.4	Day of week effects . . . . .	12
2.5.5	Beacon (loss) . . . . .	13
2.5.6	Browser differences . . . . .	13
2.5.7	Carry over effects . . . . .	13
2.5.8	Robots . . . . .	13
2.5.9	Staged roll-out with Simpson's Paradox . . . . .	13
2.5.10	Device time . . . . .	14
2.5.11	Browser redirects . . . . .	14
2.5.12	Error checks . . . . .	14
2.5.13	Unplanned differences between variants . . . . .	14
<b>3</b>	<b>Continuous Experimentation at ING</b>	<b>15</b>
3.1	Experiment Platform . . . . .	15
3.1.1	Usage . . . . .	15
3.1.2	Hypothesis analysis . . . . .	16
3.1.3	Position within ING . . . . .	16
3.1.4	Webtrekk . . . . .	17
3.1.5	Technical background . . . . .	17

3.2	Essentials . . . . .	17
3.2.1	A/A Testing. . . . .	17
3.2.2	Statistical significance tests . . . . .	18
3.2.3	Statistical Power . . . . .	18
3.2.4	Effect size . . . . .	19
3.2.5	Randomisation and Sample Ratio Mismatch. . . . .	20
3.2.6	User group assignments . . . . .	22
3.2.7	Data aggregation. . . . .	22
3.3	Pitfalls . . . . .	23
3.3.1	Overall Evaluation Criteria . . . . .	23
3.3.2	Primacy and newness . . . . .	23
3.3.3	Page speed/latency . . . . .	23
3.3.4	Day of week effects. . . . .	23
3.3.5	Beacon (loss). . . . .	24
3.3.6	Browser differences . . . . .	25
3.3.7	Carry over effects . . . . .	25
3.3.8	Robots . . . . .	25
3.3.9	Staged roll-out with Simpson's Paradox . . . . .	25
3.3.10	Device Time . . . . .	25
3.3.11	Browser redirects . . . . .	26
3.3.12	Error checks . . . . .	26
3.3.13	Unplanned differences. . . . .	26
<b>4</b>	<b>Supporting Continuous Experimentation</b>	<b>29</b>
4.1	Introducing ABvalidator . . . . .	29
4.2	The three A's of A/B test validation . . . . .	29
4.2.1	Availability. . . . .	29
4.2.2	Analysability . . . . .	32
4.2.3	Accuracy. . . . .	34
4.3	Setup, distribution and usage . . . . .	37
<b>5</b>	<b>Results</b>	<b>39</b>
5.1	RQ1: Factors affecting trustworthiness and soundness of an A/B test . . . . .	39
5.1.1	Essentials . . . . .	39
5.1.2	Pitfalls . . . . .	40
5.2	RQ2: Factors impacting ING's Experimentation Platform . . . . .	41
5.2.1	Essentials . . . . .	41
5.2.2	Pitfalls . . . . .	42
5.3	RQ3: How can the factors influencing the trustworthiness and soundness of A/B tests be modelled into a toolkit to help support engineers with setting up A/B tests . . . . .	43
<b>6</b>	<b>Discussion</b>	<b>45</b>
6.1	Main Findings. . . . .	45
6.1.1	Factors influencing trustworthiness and soundness of A/B tests . . . . .	45
6.1.2	Integration of A/B testing . . . . .	45
6.1.3	Lack of validation . . . . .	45
6.1.4	Supporting experimenters . . . . .	45
6.1.5	Modelling influential factors. . . . .	46
6.2	Implications . . . . .	46
6.3	Culture . . . . .	46
6.4	Threads to Validity . . . . .	46
6.5	Recommendations . . . . .	46
6.5.1	Generic recommendations. . . . .	47
6.5.2	Recommendations for ING's EP . . . . .	47

---

<b>7 Conclusion and Future Work</b>	<b>49</b>
7.1 Conclusion . . . . .	49
7.2 Future work. . . . .	50
7.2.1 Validation of ABvalidator. . . . .	50
7.2.2 More test users. . . . .	50
<b>Bibliography</b>	<b>51</b>





# Introduction

Many associate data-driven decision making and controlled experiments with the field of medical research. However, these topics are nowadays also playing an important role within the software industry. And although the link between medical treatments and software development might look far fetched at first, software changes and updates are similarly used to improve from a current situation. Improving in software can have multiple outcomes, such as increasing sales, achieving more consumer satisfaction, boosting software performance, etc. Many different goals can be formulated, and often these goals are intertwined. To visualise what this means, it is easiest to think of a website with a blue buy button. An experiment could be performed to see if more users click the buy button when the colour of the button is changed to green. In order to achieve this two versions of the software can be released. In the first version (version A) the colour of the button remains the same, this version will be the control group. In the second version (version B), the button is changed to the proposed colour: green. This version will be the so called treatment group. This type of experimentation is known as A/B testing. The example directly shows one of the, modern, great advantages of software: its flexibility to change a product, even after it is shipped to customers. This flexibility helps companies to focus more into problems relevant to customers and therefore deliver value [28].

Changing a software product has two types of impact: technical and business consequences. The technical consequences relate to introducing or removing bugs, and boosting, or decreasing, performance. The business consequences however are more appealing to the imagination. In 2013 a small change in font colour on the Bing search engine resulted in a \$10M income increase [42], and two other, confidential, changes at Bing increased the ad revenues by about \$100M per year[42]. In 2015 ING conducted an experiment to change a header title on the consumer log-off page. Changing four words on this page resulted in a revenue increase of €200k per year, see figure 1.1.

On the contrary, in 2012 a glitch in the trading software of Knight Capital, an American global financial services firm, resulted in the company losing \$461M in about 45 minutes [3]. A more recent example is Cloudflare's outage on July 2<sup>nd</sup> 2019 [30]. The deployment of a single misconfigured rule within the Cloudflare Web Application Firewall caused the CPU to spike to 100% on their machines worldwide. Therefore it is safe to say that running a controlled experiment on a (software) change can be of great importance [46].

Controlled experiments are nowadays widely used by software driven organisations [16]. Companies such as Google, Booking.com, Microsoft, LinkedIn, Airbnb, Netflix, Yandex, Uber and Twitter test changes to their software products with a subset of real, unaware, users before being rolled out to all their users [32, 36, 60]. Given the amount of software these companies release to their production environment, this testing must happen on a continuous basis. This process is also known as Continuous Experimentation (CE).

## 1.1. Background

The technical and business impact of software changes results into two approaches to software experiments; Regression-Driven Experiments and Business-Driven Experiments [54]. 'Regression-Driven Experiment' is a label for experiments that are used for the mitigation of technical problems, performing health checks and testing scalability on production workload [54]. This is the case when there is only interest in the technical

## U bent uitgelogd. Speciaal voor u geselecteerd:

**Nu € 5 cadeau per € 2.500 extra spaargeld**



**Extra beloning**

Benut u graag een mooie kans om alles uit uw spaargeld te halen? Meld u nu aan en laat uw spaargeld groeien.

► [Meer informatie](#)

**Nu aanmelden**

**Vraag van vandaag**

3 februari 2015

**Hoeveel loon zou u inleveren in ruil voor behoud van uw huidige baan?**

Niets

Minder dan 5%

5 tot 10%

Meer dan 10%


Niet van toepassing

**Stem**

► Wist u dat... u met de '30-dagen Challenge' de uitdaging aan kunt gaan meer geld over te houden?

## U bent uitgelogd. Heeft u dit al gezien?

**Nu € 5 cadeau per € 2.500 extra spaargeld**



**Extra beloning**

Benut u graag een mooie kans om alles uit uw spaargeld te halen? Meld u nu aan en laat uw spaargeld groeien.

► [Meer informatie](#)

**Nu aanmelden**

**Vraag van vandaag**

3 februari 2015

**Hoeveel loon zou u inleveren in ruil voor behoud van uw huidige baan?**

Niets

Minder dan 5%

5 tot 10%

Meer dan 10%

Niet van toepassing

**Stem**

► Wist u dat... u met de '30-dagen Challenge' de uitdaging aan kunt gaan meer geld over te houden?

Figure 1.1: Above: the control variant of the ING log-off page experiment. Below: the winning treatment variant of the experiment. Changing four words in the title resulted in €200k income gain per year.

performance aspect of the software update (e.g. introduction of new bugs, or increased strain on available server capacity).

In the other approach, Business-Driven Experiments, the focus of the experiment is based on improving business metrics. The choice to run such an experiment is often initiated by a team lead or product manager [54], and driven by a certain business goal (e.g. increasing the conversion ratio in a web-shop).

This division into Regression-Driven and Business-Driven approaches, as introduced by Schermann et al. [54], suggests that an experiment can only be one of both. In practice however experiments are often initiated from the Business-Driven point of view, but the Regression-Driven aspects are actively monitored. More on this in paragraph 1.1.3.

### 1.1.1. A/B Testing

Both Business-Driven experiments and Regression-Driven experiments boil down to a very simple basis. A subset of users gets to interact with the new, updated, version of the software. This version is also known as the 'treatment variant' or 'variant B'. The rest of the users will remain to use the 'old' version of the software. This version is also known as the control variant or variant A. These two variants coexist for a variable amount of time and collect data on how users interact with these variants. Several different kinds of metrics can be collected [13, 37–39]. When the test period is completed, the metrics are analysed in order to detect causal relationships [39]. If a change is significant, and preferred, it can then be chosen as the new definitive variant



for all users.

This process of having variants A and B competing is known as A/B testing or split testing. Although the name A/B testing implies a limited number of two competing variants, multiple treatments can be ran against the control variant simultaneously.

### 1.1.2. Reaching the user

When a code update is ready to be tested, there are two different methods to get this update to users whilst also keeping the 'old' variant active as control variant[54]. The first method to achieve this is via a separate build. When working on the new feature an engineer replaces the old code with the new code on a feature branch. When the code is ready the entire application is compiled into a new build. At first this build is only distributed to a small subset of users. These users might be knowingly part of an opt-in beta program, but they can also receive this test version without them being aware. This latter scenario is for example implemented in the AppStore, where app developers can choose to release a new version of their apps to a small, steadily increasing, subset of their total user base. This so called 'phased release' spreads the release of the new version over a period of one week, giving the developers the possibility to halt a roll-out when technical issues arise[2]. Different build release tactics are known as Canary release, darklaunch or gradual roll-outs [54]. In the scenario where the treatment variant is favourable over the control variant, the build is released to all users and the feature branch can be merged into the master branch. If the control variant turns out to be favourable, a new build containing only the control variant is released to the users whom were previously on the feature build. The feature branch will not be merged into master.

The second methodology uses the help of so called feature toggles. Using this methodology all users remain on the same build. However the treatment variant feature can be toggled on and off. This toggling can happen client-side, server-side or both, based on information about the user or purely random for some users. A big advantage of feature toggles is the possibility to stop an experiment straight away when the treatment variant causes technical issues or significant decrease of important metrics (also known as guardrail metrics, see paragraph 1.1.5). A good example of stopping experiments automatically when guardrail metrics are violated is Booking.com's "The Circuit Breaker" which has the power of stopping an experiment within a second [62].

More on assigning users to the variants can be found in paragraph 2.4.5.

### 1.1.3. Experiment Platform

Before companies are able to run their own A/B tests some infrastructure requirements have to be met [32]. Fagerholm et al. [28] describes the back-end system needed to perform these tests. This back-end system should consist of an experiment database which is able to store raw data sent by the software application, experiment plans, including programmatic features of sample selection and the other logic needed to conduct the experiment. Furthermore it should be able to store the experiment results. This back-end system should have an API so the software application is able to communicate with it [28]. Gupta et al. notes that an Experiment Platform should also clearly communicate results from the experiments, add some level of quality control to ensure that the results are trustworthy and be flexible enough to support the diverse needs of future teams [32].

Although these back-end systems are available as commercial (e.g. Optimizely<sup>1</sup> or VWO<sup>2</sup>) and Open Source (e.g. Wasabi<sup>3</sup>) stand-alone products, many companies, such as Amazon, Booking.com, Facebook, Google, LinkedIn, ING and Microsoft, have developed their own internal variant [21, 32].

### 1.1.4. Scope of A/B testing

A/B testing is applicable to many facets of the software development process [26]. The most frequently mentioned experiment types are the User Interface (UI) changes, which are directly and clearly visible to the users (e.g. the ING experiment as can be seen in figure 1.1). However measuring improvements in ranking algo-

---

<sup>1</sup><https://www.optimizely.com>

<sup>2</sup><https://vwo.com>

<sup>3</sup><https://github.com/intuit/wasabi>

rithms and recommender systems are also popular applications of experimentation [20, 34]. Any software change that results in a consumer facing change is within the scope of A/B testing.

*“Almost every product change is wrapped in a controlled experiment. From entire redesigns and infrastructure changes to the smallest bug fixes, these experiments allow us to develop and iterate on ideas safer and faster by helping us validate that our changes to the product have the expected impact on the user experience.”* [62]

Lukas Vermeer, Director of Experimentation at Booking.com

### 1.1.5. Data types and collection

In order to compare the control variant with the treatment variant(s) data is required. This data contains information on how the user interacted with a variant, and what specific variant the user was presented.

The collected data can consist out of many different kind of metrics. These metrics range from whether or not a user clicked on a button, to the total amount of items the user has in its shopping cart. Having the right metrics is critical to successfully executing and evaluating an experiment [15, 19, 21, 47]. When a specific metric is of great importance to the companies business, it can be considered as a guardrail metric, meaning that this metric is not allowed to deteriorate as a result of a given experiment. If a metric is used to evaluate the success of an experiment, it is part of the Overall Evaluation Criteria (OEC) [48] for that experiment.

For the collection of data, three different approaches are possible [39]. The first approach uses existing (external) data collection. Often software organisations already have some sort of data collection in place, either using an in-house developed solution or from 3rd party vendors such as Adobe Analytics<sup>4</sup> (formerly known as Omniture) or Webtrekk<sup>5</sup>. The user assignment information can be pushed into these systems, so it comes available for analysis. Although sending this information into these systems is often simple to set up [39], many of these platforms are not designed for the statistical analysis required for A/B tests. Manually pulling the data from these platforms and computing the analysis by hand can be expensive and impedes real time analysis. Kohavi et al. [39] therefore recommends to avoid this approach.

To overcome some limitations of the first approach, the collected data can be stored locally on servers or devices and must be sorted and aggregated before analysis can begin. This second approach is called 'Local data collection'. Although this solution scales for large software systems, collecting these logs for analysis without data loss becomes extremely difficult [32, 39]. This collecting and processing of data before analysis is called 'cooking' and results into a data log ready for analysis called a 'cooked log' [32]. Companies such as LinkedIn, Facebook and Airbnb use this approach by having a separate metrics framework, and importing these metrics into their Experiment Platform ('bring-your-own-data')[32].

The third approach is service-based collection, in which the service is the Experiment Platform. The advantage of this approach is centralising all observation data, making it easy for analysis. This means that the data collection is specifically designed to collect data for the Experiment Platform. From interviews with Booking.com, Bol.com and ING it is learned that this level of integration requires a great infrastructural investment, and is not easily implemented in an already running big software environment. At Booking.com the data aggregation is divided over two pipelines, one real-time the other through daily logs, to ensure the consistency of the data and minimise the possibility of data loss.

For all approaches it is important to note that data quality checks are the first step in analysing experiment results [31].

## 1.2. Terminology

In the literature for controlled experiments the terminology varies widely. To avoid this confusion for the rest of this report, some terminology is introduced.

**Experiment Platform:** the back-end system as described in paragraph 1.1.3.

<sup>4</sup><https://www.adobe.com/analytics-cloud.html>

<sup>5</sup><https://www.webtrekk.com>

**A/B test:** also known as "Split tests" or "Bucket tests". More information in paragraph 1.1.1.

**Overall Evaluation Criterion (OEC):** the quantitative measure for the experiment's objective [53].

**Variants:** Every possible user interface shown by the software is a variant. For the scope of this research only two variants will be used simultaneously, the control and treatment variant.

**Null hypothesis:** the hypothesis that the treatment variant does **not** outperform the control variant, and that any found differences are due to random fluctuations. Written as  $H_0$ .

**A/A test:** comparing variant A with variant A. A test methodology to check for errors and strange behaviour in the experiments. Also known as a Null Test [52].

**Guardrail metric:** a metric of such importance that it is monitored for every experiment and should not deteriorate.

### 1.3. Problem Statement

Small changes in software can lead to immense gains or loss of revenue for the company running the software [41]. Using controlled experiments these changes can be validated to ensure they are preferable, and minimise the risk on technical or business problems before the change is rolled out to all users. However, it should be known if the results of an experiment are trustworthy [32].

The trustworthiness and correctness of the Experiment Platform are essential when relying on experiments to make decisions in choosing variants for the production environment [10]. Currently no tooling is available to externally validate an Experiment Platform. Errors are only detected when the platform itself finds an indicator of unhealthy data. These anomalies are expensive to investigate [37], and can often take longer than running the experiment itself [10]. Furthermore an Experiment Platform checking itself is limited to the implementation of the Experiment Platform, which can also contain errors.

ING is heading towards supporting CE throughout its organisation (more on ING in chapter 3). In order to support this movement the validity of their in-house created Experiment Platform is crucial. This leads to the first two research question of this study:

**RQ 1:** *What factors affect the trustworthiness and soundness of an A/B test?*

**RQ 2:** *To what extent is ING's EP impacted by the factors influencing the trustworthiness and soundness of an A/B test?*

By answering *RQ1* and *RQ2* value is created for ING, however for other organisations working with CE the problem remains. Furthermore, checking ING's current situation provides a snapshot and does not provide an active control measure for future experiments within the Experiment Platform. Based on the desire to serve a more durable community wide solution the following research question is formulated:

**RQ 3:** *How can the factors influencing the trustworthiness and soundness of A/B tests be modelled into a toolkit to help support engineers with setting up A/B tests?*

### 1.4. Approach

In order to answer the research questions we performed a theoretical literature review [49] to gain knowledge in the current influential factors in A/B testing. By following the approach as introduced by Paré et al. [49] and DeLone et al. [14] we are able to identify relevant research in the field of CE and A/B testing. From this research the influential factors will be deducted to answer *RQ1*.

The results of *RQ1* will be used in an exploratory case study at ING. During this case study data from ING's EP is checked for the presence of the influential factors. The procedures used to check for these factors in

ING's Experiment Platform is described in chapter 3.

To answer the third research question a qualitative research [12] approach is chosen in addition to the already performed theoretical literature review. This qualitative research consisted of interviews with A/B test experts and data scientists from ING, Booking.com and Bol.com to find focal points for the automated tooling.

## 1.5. Main Contributions

Studies have been performed to highlight certain essentials and pitfalls in A/B testing, often by highlighting single example cases in which they were found. To my knowledge this study is the first time the literature on essentials and pitfalls in A/B testing is combined and checked against one specific Experiment Platform. Furthermore no Open-Source tool to help engineers with flagging mistakes in experiments has been reported, although recent research is done in this direction by Chen et al. [10].

- **Methodological contribution:** The first main contribution of this study are the methods used to validate an Experiment Platform. Steps taken in this research can be used by others to validate their Experiment Platform.
- **Empirical contribution:** The second contribution is in the results of the empirical case study. The findings from this study can be used as guidance for creating or adjusting an Experiment Platform as well as give an advance notice into the complexity that arises with creating an Experiment Platform.
- **Conceptual contribution:** Lastly this research contributes by giving the first development stages of a tool which can be used by engineers and data scientists to prevent them from making errors in experiments.

# 2

## Essentials and Pitfalls in A/B Testing

This chapter provides an insight into the essential requirements for, and common pitfalls of, conducting reliable A/B tests. It starts by presenting the results of a theoretical literature review. [14, 49], including related work. This is followed by an overview of the identified essentials and pitfalls, after which each factor is discussed in further detail.

### 2.1. Methodology

As mentioned in paragraph 1.4 we use a theoretical literature review as described by Paré et al. [49] to summarise the literature on influencing factors on the trustworthiness and soundness of A/B tests. As a starting point for this theoretical review we used the journal paper "Controlled experiments on the web: survey and practical guide" by Kohavi et al. [39]. By using the (reversed) snowballing methodology [5] we identified more relevant papers that contained empirical and conceptual studies that measure various dimensions and factors pertaining to trustworthiness and soundness of A/B tests and continuous experimentation. Furthermore newer papers have been found by searching on Google Scholar<sup>1</sup> with the following queries:

- "Continuous Experimentation" AND published in 2014 or later
- "A/B test" AND published in 2014 or later
- The above two in combination with the terms "validation", "trustworthiness", "soundness", "pitfall" or "essential"

Citations included in the obtained papers are also used, depending on their relevance to the topic.

### 2.2. Related Work

Much research can be found on a variety of aspects of controlled experimentation. A recent study by Fabijan et al. [26] shows that companies typically develop in-house Experiment Platforms, and that these platforms are of various levels of maturity. In recent years case studies have been published on the integration of continuous experimentation and A/B testing at companies. A 2013 case study by Deng et al. [16] at Microsoft's Bing shows the influence of pre-experiment data to reduce metric variability thereby reducing variance by about 50%, achieving the same statistical power with half of the users, or half of the experiment duration. The way of experimentation at Microsoft has been often described by Kohavi et al. [37–42]. In 2010 Tang et al. [60] performed a case study into the overlapping experiment infrastructure at Google. Lindgren et al. [45] presents an "interview-based qualitative survey exploring the experimentation experiences of ten software development companies", and found "that although the principles of continuous experimentation resonated with industry practitioners, the state of the practice is not yet mature". Although Crook et al. [13] encountered problems in every stage of the analysis pipeline at numerous websites, I found no literature into how to validate a current Experiment Platform against all the pitfalls and essentials mentioned in the literature. At WDSM '19 Chen et al. published a study on how to automatically diagnose some parts of A/B testing [10]. Two other recent

<sup>1</sup><https://scholar.google.com>, visited between December 2018 and September 2019

papers, published during this research, give some additional insights. A study by Fabijan et al. published at ICSE-SEIP '19 provides checklists to help novice data scientists and software engineers to become more autonomous in setting-up and analysing experiments, since depending solely on experts is neither scalable nor bulletproof [27]. At SIGKDD '19 a group of thirty-four experts published about the top challenges and pitfalls faced across the industry for running Online Controlled Experiments at scale [32].

### 2.3. Essentials and pitfalls

The identified factors that influence the trustworthiness and soundness of A/B tests can be grouped into two categories. The first category contains all factors that are by literature deemed essential for running a basic A/B test. If these factors are not in place, or wrongly implemented, the results of an experiment can be considered void. The other category contains factors described in literature as pitfalls. These factors can occur as result of faulty implementations of the essential factors, or by not having additional checks. The factors per topic can be seen in table 2.1, and will be discussed in more detail below.

Essential	Pitfall
A/A testing	Overall Evaluation Criteria (OEC)
Statistical significance tests	Primacy and newness
Statistical power and confidence	Page speed/latency
Randomisation and Sample Ratio Mismatch	Day of week effects
User group assignments	Beacon (loss)
Data aggregation	Browser differences
	Carry over effects
	Robots
	Device time
	Browser redirects
	Error checks
	Unplanned differences between variants

Table 2.1: Factors affecting trustworthiness and soundness of A/B tests divided per category.

## 2.4. Essentials

When starting with running experiments, a couple of topics are required in order to be able to safely determine the optimal variant. These topics are called the essentials of A/B testing.

### 2.4.1. A/A testing

A/A tests have been well covered in the literature [13, 31, 37–39, 52, 60, 65]. An A/A test is a randomised experiment with two identical variants, A and A, which should be exposed to the same control experience in an online controlled experiment [65]. A/A tests are highly recommended [37]. A/A testing helps to identify problems with the Experiment Platform, as well as to validate it. For example, metric balance is more difficult to assess during the A/B period than during the A/A period. This is because metric differences may result from a mixture of treatment effects and from unknown confounding factors [65]. When using a significance threshold of 95%, only one in twenty A/A tests should result in one of the A's turning out significant (due to chance) [29]. If too many, or few, metrics turn out to be statistically significantly different during an A/A test further investigation into issues is needed [13]. Other outcomes of regular A/A tests can include whether or not the collected data matches the system of record, and if the users are split according to the planned percentages [38, 39]. If it is found that users are not split accordingly, re-randomisation is required [31]. A/A testing is an essential feature for an experiment platform [24, 25].

### 2.4.2. Statistical significance tests

The P value is a measure of statistical evidence [29] introduced by Pearson [51]. After the introduction of the 5% significance boundary by Fisher [22], it has been a popular method to indicate the strength of evidences of scientific findings [21]. However, Fisher's choice for word "significant" was quite deliberate, and should be merely an indication of value in repeating the experiment [29]. A survey by Windish et al. [63] under medical residents shows that 88% of the participants shows fair to complete confidence in interpreting P values, however only 62% of these could answer an elementary P value interpretation question correctly.

When the P value is outside of the chosen confidence interval, the treatment variant shows a noteworthy finding. However when a metric comes with a borderline p value, it can be a sign of false positive [21].

In A/B testing two types of statistical hypothesis tests that yield a P value are commonly used: the T-test and the Chi-Squared test. The T-test tests an  $H_0$  for comparing two means, and requires a categorical variable (control or treatment) and a quantitative variable (e.g. the amount of items in a shopping cart) which should follow a normal distribution. The Chi-Squared test tests an  $H_0$  about a relationship between two variables, which can be used to determine if the difference made in the treatment variant has a relation to the measured Overall Evaluation Criteria (OEC). The Chi-Squared test is based on normally distributed data. For A/B testing it is assumed that the sample sizes are large enough that it is safe to assume the means have a normal distribution by the Central Limit Theorem [7, 8, 39]. When using the Chi-Squared distribution to interpret Pearson's Chi-Squared statistic the assumption has to be made that the discrete probability of observed binomial frequencies in Pearson's table can be approximated by the continuous Chi-Squared distribution. This assumption introduces an error, which should be corrected using the Yates's correction. However research by Haviland et al. shows that this correction is over conservative [33]. The normality assumption can also be tested using the Shapiro–Wilk test [57]. If P values are not uniform, it indicates an issue, for example an incorrect variance computation [31].

### 2.4.3. Statistical Power and confidence

Power describes the probability of correctly rejecting  $H_0$ , when it is false. Failing to reject a false  $H_0$  is known as a type II error. The probability of having a type II error is denoted with  $\beta$ . Thus power is described as  $1 - \beta$ . It is recalled from the terminology section that in A/B testing  $H_0$  describes the treatment variant not outperforming the control variant. In this context power means the probability to truthfully detect the treatment variant outperforming the control variant. From literature it is found that the desired power in A/B testing is at 80% [39].

The confidence level describes the probability that observed data lies within the described interval. For A/B testing this percentage is commonly set to 95% [39]. This means that in 5% of the experiments the  $H_0$  is true, but is still rejected. In A/B context this happens when the treatment variant is decided to outperform the control variant, although it does not. This is known as a Type I error.

In order to run a valid experiment with the chosen power of 80% and confidence level of 95% the number of participating users needs to be computed. To calculate this minimum sample size the following formula is used [61]:

$$n = \frac{16\sigma^2}{\Delta^2} \quad (2.1)$$

In equation 2.1  $n$  describes the sample size,  $\sigma$  is the standard deviation (the measure of variability), which comes from previous experiments, a pilot experiment or the Bootstrap method [23],  $\Delta$  is the sensitivity (the amount of change to be detectable, also known as the effect size). For example in the control variant the average spending of a customer is \$80, and this metric is chosen as the OEC. If it is decided that a 1% change in this OEC should be detected, the sensitivity becomes  $\Delta = \mu_0 * 0.01 = 80 * 0.01 = 0.8$ .

Typically web traffic has a tremendous variability, making it difficult to run an experiment with sufficient power to detect effects on smaller features [39]. Power analysis is an essential feature for an experiment of-fice [24, 25]. At Google so called "uniformity trials" are constantly ran to help determine the correct values for  $\Delta$  and  $\sigma$  [60]. These "trials" consist of A/A tests in which both the experiment size and the traffic is varying.

Multi-Variable Testing is found to be less beneficial compared to bi-variable designs [42].

#### 2.4.4. Randomisation and Sample Ratio Mismatch

The underlying assumption used for the statistical difference between the control and treatment version of the A/B test is the fact that both versions were visited by a random sample of end users. This means that having a good randomisation algorithm to select users for the experiment is critical [21, 39]. The following three properties for the randomisation algorithm [38, 39] are of importance:

- Any bias towards a specific variant should be avoided.
- Assignment of a revisiting user should be consistent. The user should be presented the same variant every visit for the duration of the experiment.
- When multiple experiments take place, there should not be any correlation between assigning users in the various experiments.

Although randomisation can be hard to achieve [50], it is found that random number generators in the most popular programming languages do satisfy the first point. However some conditions such as the moment of seed generation can influence this [39].

Another approach to randomisation of users is to use a hash based approach. In this approach a unique user specific identifier in combination with an experiment identifier is used as input for an hashing algorithm (e.g. MD5, SHA1). From the output hash the last bit is used to determine in what group the user should be (e.g. 0 for control, 1 for treatment). The benefit of this approach is the possibility to determine in what variant the user is classified, without storing this information into a database.

The soundness of the randomisation is of importance for data quality. The most effective data quality check for experiment analysis is the Sample Ratio Mismatch (SRM) test, which compares the observed user counts against the configured ratio [10, 31]. Whenever SRM is detected, the results are deemed invalid. SRM can be caused by many different reasons (e.g. incorrect filtering of bots, beacons not correctly working in some web-browsers, or logical implementation error within the experiment) [9, 37]. Hence the importance of the correctness of the randomisation: if the randomisation is skewed the other reasons are not automatically detectable.

There are two types of tests available that are commonly used to test for SRM: the Binomial Test and the Sample Size Ratio Test (also known as Multinomial Test) [9, 10]. Both tests are used to find the difference between an expected distribution and an observed distribution. The Binomial Test is two-tailed, since the control/treatment variant can have either too many or too few users. The Sample Size Ratio Test is given by equation 2.2 and results into the Chi-Squared statistic.

$$\chi_{K-1}^2 = \sum_{i=1}^K \frac{(O_i - E_i)^2}{E_i} \quad (2.2)$$

Equation 2.2 is known to be used at least at both Twitter [9] and LinkedIn [10].  $K$  denotes the number of variants.  $O_i$  is the number of users observed in variant  $i$ , and  $E_i$  is the number of expected users in variant  $i$ . Both tests yield the same P value when calculated for two variants. This P value can be checked against a threshold value to conclude if SRM occurs. At Microsoft this threshold is at 0.001 [43], Twitter uses 0.05 [9]. A lower threshold leads to more certainty about the actual presence of SRM (true positive), however it means that some cases of SRM might go undetected (false negative). When continuously performing A/A tests, the P values for these distribution tests should follow a uniform distribution [27, 31].

#### 2.4.5. User group assignments

Related to the randomisation of users is the actual assigning of users to a variant. The assigning method enables the experiment software (e.g. the site being tested) to execute a different code path for different end users [39]. Three commonly used assignment methodologies are discussed.

##### Traffic splitting

Traffic splitting is an assignment method that works for web applications. When a users visits the site running the experiment, a load balancer (or proxy server) divides the traffic between servers or clusters of servers (depending on the size and traffic of the site). One server (cluster) runs the control variant of the software,



the other server (cluster) runs the treatment variant. The main advantage of this approach is the assignment method being not intrusive, meaning that no additional changes to the code are required to implement the experiment. Traffic splitting uses the new build methodology as described in 1.1.2. Drawbacks of this methodology include the costs required to test small changes, the requirement for the control set-up to be able to cope with 100% of traffic load in case an error in the treatment variant emerges, increased complexity of running multiple experiments simultaneously and the increased risk of detecting differences due to varieties in the infrastructure set-up[39].

### **Client-side assignment**

The most popular assignment methodology [39]. Client-side assignment is not limited to web-only applications and can therefore be useful for A/B tests that influence cross-platform users (e.g. users using both the web-app and a mobile app). The assigning process can take place on a separate "assignment server". The client side software connects to this server in order to determine what variant needs to be shown to the user. The assigning process can also be based on the hash based approach. This methodology is easy to implement for developers of the client side software. Disadvantages include the additional latency introduced by connecting to the assignment server (when used), and the ability of end users (with technical know-how) to determine that they are part of an experiment.

### **Server-side assignment**

With server-side assignment the server assigns the user to a variant, and sends the content required for that variant towards the client. This means that the software on the web server has to be heavily adjusted to support experimentation [39]. This methodology is used by Google for their experimentation [60]. Advantages include it being an extremely general method, the experimentation code is in the best logical place and the experimentation is completely transparent to end users [39]. The disadvantages are related to the methodology being very intrusive [39].

### **2.4.6. Data aggregation**

In order to support A/B testing, logging the right data should be an integral part of the development process [31]. Different methods are possible to collect raw data such as using existing data collection, local data collection or service-based collection [39]. When an error occurs in the collection of data it renders the experiment invalid, and experimenters should be blocked from viewing the results to prevent incorrect conclusions [31]. At Booking.com two separate pipelines are set-up to prevent issues in data aggregation [36]. An in-depth description on data aggregation can be found in paragraph 1.1.5.

## **2.5. Pitfalls**

By having the essentials of A/B testing in place, the experimentation can start. However having these essentials right does not guarantee trustworthy and correct outcomes. From literature 13 pitfalls are identified which are still common to occur.

### **2.5.1. Overall Evaluation Criteria**

The OEC is the criteria which is used to determine the performance difference between the variants of the A/B test. Picking a good OEC is essential in the overall business endeavor [41]. The goal for a good OEC is to include factors that predict long-term goals [21], for example predicted lifetime values and repeat visits. Short-term goals, such as clicks, should be avoided [13]. As an example an experiment is introduced where the goal is to increase sales of a product directly from the homepage. If it is decided that the OEC is the number of clicks on a revenue generating link (e.g. the "buy now" button), the results of the A/B test could be flawed. Imagine the homepage in variant A not containing any pricing information in contrast to variant B. The number of clicks on the revenue link might drop, but the conversion rate to actually buying might increase. This scenario has been seen happening on the front page of Microsoft's Office Online site [13]. Another example explaining the reason behind long-term OECs comes from Amazon [37]. An experiment was performed to increase success of e-mail campaigns. The chosen OEC was based on purchases whose sessions were referred by the e-mails. This experiment however resulted in sending more e-mails to the (future) customers. Short-term the OEC was achieved, however long-term customers started complaining and unsubscribing from the e-mails.

It is important to note that long-term OECs are not the same as long-duration testing. The latter could harm a company's agility, and it is complicated to correctly track users over a longer period of time [32].

The second OEC related pitfall is a focus on a specific, small, area of a User Interface. Generating more attention to a specific area can be easily achieved by highlighting it (e.g. making it bold, changing the background colour, etc.). However it is important to choose an OEC that incorporates metrics from the whole-page, in order to prevent improving a specific area but deteriorating other areas of the page.[13]. In order to adjust for this behaviour, a good OEC can be extended with a penalty term [39]. Guardrail metrics also help to prevent harmful impact on important metrics.

Finally it is of importance to determine the OEC before the experiment is started (a planned comparison). Not doing so increases the risk of a type I error (finding what appears to be significant results by chance) [38].

### 2.5.2. Primacy and newness

Primacy and newness are two contrary effects which can influence the outcome of an A/B test [10, 21, 38, 39]. Both effects have to do with changes in important structural parts of the site or software product, e.g. the navigation menu. When experienced users of a site are confronted with a new navigation menu they may be less efficient at first, giving an advantage to the control version [41]. This is called a primacy effect. The newness effect, also known as novelty effect, is in contrast with the primacy effect, and describes experienced users being intrigued and therefore clicking everywhere introducing this newness bias, giving an advantage to the treatment variant [41]. The newness effect also describes the difference between seeing a change or feature for the first time, and having used it more often. For example users being open to notifications at first, but disabling them later on [34]. This newness bias is sometimes associated with the Hawthorne Effect [4]. Both effects can be assessed by generating a delta graph between the control and treatment variant. An equation for this delta graph is introduced by Chen et al. [10]:

$$\Delta\%^{[t,t]} = \beta_0 + \beta_1 \frac{1}{t^\alpha} + \beta_2 \frac{1}{t^\gamma} \quad (2.3)$$

Where  $\Delta\%^{[t,t]}$  stands for the percentage of impact between day  $t$  and day  $t$ , being the single day impact.  $\alpha$  and  $\gamma$  should be chosen in such a manner that  $\frac{1}{t^\alpha}$  is a slow-decay term and  $\frac{1}{t^\gamma}$  a fast-decay term. Chen et al. have found  $\alpha = 0.35$  and  $\gamma = 2$  as suitable values [10]. When a Multiple Linear Regression is ran on equation 2.3 with the first week of data a primacy or newness effect can be flagged whenever the following three conditions hold:

- (a) The linear model captures the effect trend well ( $R^2 \geq 0.8$ )
- (b) The fitted line is monotonic in  $t$
- (c) The largest impact is statistically significantly different from the smallest impact

### 2.5.3. Page speed/latency

Website performance, or speed, is critical [38, 42, 59]. Added latency can adversely impact the user experience [56, 65]. A decrease of sales by 1% was seen at Amazon after a 100ms slowdown of the website [44]. Google reports similar significant impacts on key metrics after slowdown experiments [55]. Hence an extra delay in a variant will likely cause a decrease in click-through rate and other metrics [13]. Therefore it is important that if time is not directly part of the OEC, it is ensured that a version is not losing due to added latency [38, 39].

### 2.5.4. Day of week effects

As discussed, an import part of a successful A/B test lies in the number of users involved in the experiment. When running an experiment with lots of users it could occur that the minimum required of users is reached within a few days or even hours. However behaviour of users can differ based on the day of week. Many sites have different user segments on their site in weekends compared to weekdays [38]. This is why it is of importance to always run an experiment in full weeks (e.g. 25 days would become 4 weeks). However, experiments that run for too long (e.g. over 125 days) can be unreliable due to factors such as cookie churn [38].

### 2.5.5. Beacon (loss)

In order to track the OEC in both variants some software needs to be added to collect the data. For example when using the client side assignment as described in paragraph 2.4.5 for a web application, a bit of JavaScript code is required that sends the metrics to the Experiment Platform. This code is called a beacon. Errors with this beacon could lead to loss of data, therefore influencing the results of the experiment. Zhao et al. [65] describes a scenario in which the beacon code was at a relatively lower position in the treatment variant compared to the control variant. This results in more time before the beacon was fired. For users with a slow internet connection or old browser this meant that they could have already abandoned the page before the beacon fired, resulting in them being underrepresented in the treatment variant.

For mobile applications beacon loss can occur due to optimisations within the mobile operating system. These events are often only sent to the experiment office when a device is connected to the WiFi, and with a maximum of buffer on the device this can lead to loss of data when a device is not connected to WiFi for a longer period of time [21]. This is for example one of the important challenges of Booking.com, since many of their app users are abroad and therefore do not have an active cellular data subscription.

### 2.5.6. Browser differences

Browsers can differentiate in how JavaScript, CSS or HTML is executed and shown to the user. Kohavi et al. [40] shows how Chrome, Firefox and Safari are aggressive about terminating requests on navigation away from the current page, leading to a non-negligible percentage of measured events never making it to the Experiment Platform. However Internet Explorer continues to execute beacon requests even after navigation, which makes click tracking more reliable but skews the data in comparison with other browsers [41]. Therefore breaking down patterns by the browser type used may highlight problems that appear in some browsers, but not in others [13].

### 2.5.7. Carry over effects

The result of performing many experiments on a software product is the increased likelihood of a user participating in more than one experiment. In the situation where a user impacted by a first experiment is also selected for a second experiment carry over effects can occur [41]. For example when a user has had a really bad experience in the first experiment (e.g. encountered a severe bug), it impacts the behaviour in the second experiment. Carry over effects have been measured that last for 3 months [41]. Experiment platforms should try to avoid these carry over effects [65].

### 2.5.8. Robots

Automated visitors, robots, on an experiment can cause misleading results due to their behaviour being not representative for human behaviour [13]. In the literature examples are described of robots generating 100 click events per minute for a duration of 2.5 hours [13]. Hence it is understandable that robots can introduce significant skew to render assumptions invalid [39]. Although some type of robots are difficult to identify [13], they should be excluded from the experiment. However, as long as robots are distributed uniformly over the control and treatment their relative impact is small [37].

### 2.5.9. Staged roll-out with Simpson's Paradox

When a staged roll-out is chosen for an experiment, e.g. to mitigate the chances of exposing users to faulty variants, the risk of the Simpson's Paradox [58] is introduced. This means that the treatment variant can outperform the control variant on every individual day, however when the weighted average is made the control variant wins [13, 37]. An example is given in table 2.2.

	Friday C/T split: 99% / 1%	Saturday C/T split: 50% / 50%	Total
C	$\frac{20,000}{990,000} = 2.02\%$	$\frac{5,000}{500,000} = 1.00\%$	$\frac{25,000}{1,490,000} = 1.68\%$
T	$\frac{230}{10,000} = 2.30\%$	$\frac{6,000}{500,000} = 1.20\%$	$\frac{6,230}{510,000} = 1.20\%$

Table 2.2: Conversion Rate for two days. Each day has 1M customers, and the Treatment (T) is better than Control (C) on each day, yet worse overall. Table, data and description by Crook et al. [13]

### **2.5.10. Device time**

When using time stamps based on the internal clock of the users' client incorrect data points can be introduced [65]. This can occur due to the lack of clock synchronization on the users' device. When this time stamp is involved in the OEC, it can influence the results of the experiment.

### **2.5.11. Browser redirects**

There are several different methodologies to present a user to a treatment variant. One of these methodologies is browser redirect. This is done by using the `http-equiv="REFRESH"` meta tag in HTML [37]. However this methodology results in significant under-performance of the variant. This is due to a difference in performance, bots (some bots can not follow the redirection) and the redirects being asymmetric [37].

### **2.5.12. Error checks**

Running automatic checks for errors, e.g. syntax error checks, leads to fewer broken experiments being run [60]. Exposing less users to broken version can reduce the risk of negative carry over effects [41]. Systems like Booking.com's Circuit Breaker help mitigate these problems [62]. Error checks can also include tests such as: does the experiment have a control variant, does the control variant divert on the same set of traffic as the experiment [60]. Therefore assisting experimenters in the process of setting up an experiment.

### **2.5.13. Unplanned differences between variants**

In order to compare the control and treatment variants, it is understandably of importance that the control and treatment variant only differ where it is intended (the change the experiment is set up for). Any unintended difference can skew the results of the experiment [37]. Therefore it is of importance to be able to detect changes other than the tested change. Kohavi et al. encourages to drill-down on hourly data to detect any unexpected behaviour, as well as use screen scrapers to take screen shots of the pages shown to the users on a regular basis. These screen shots can be used later on to help identify differences.

## Continuous Experimentation at ING

ING is a large globally operating bank. ING believes that banking products are becoming commodities, and that customer experience is the only way to differentiate in the future. Industry leaders like Amazon, Apple, Facebook and Google offer access to platforms where customers connect to one another and to businesses, and where they spend more and more of their time [1]. To remain relevant to their customers ING wants to create a similar experience for which they say it is key to work toward one global and scalable IT infrastructure with a modular approach for easy plug-and-play connections [1]. This strong focus on software and customer underlines the importance of A/B testing to ING.

### 3.1. Experiment Platform

At ING the Business-Driven Experiment approach is being used. The required back-end system as described by Fagerholm et al. [28] is an internally developed system called the Experiment Office (named Experiment Platform after this).

#### 3.1.1. Usage

Since its launch in March 2016 to July 1st 2019 over 268 A/B tests have been initiated in the Experiment Platform. The number of initiated experiments is rapidly increasing as can be seen in figure 3.1. In 2016 only 17 A/B tests were conducted using the Experiment Platform, however in 2018 this number already increased to 120 that year.

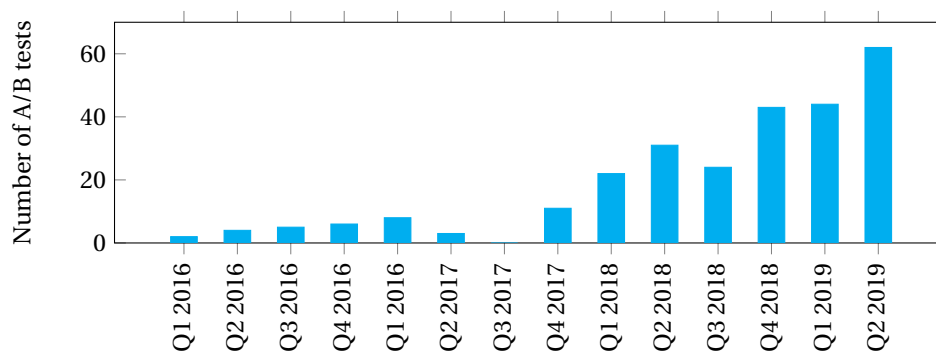


Figure 3.1: Number of A/B tests run by ING's Experiment Platform per quarter (Q).

The Experiment Platform (EP) is geared towards two types of users: content creators and developers. The content creators (or 'Customer Journey Experts' in ING's jargon) are users whom use the ING Content Management System to display information on the ING.nl main website. Using the EP, they can write different copy and use altering images in order to determine what works best for their OEC. The developer group has a broader use case for the EP. With the help of a, deprecated, Angular code snippet or

API endpoints, they can use the EP to have full control over what code is executed in their application based on the variant. Out of the total of 268 A/B Tests, 167 were initiated by content creators and 101 by developers.

Currently the focus of the team (or 'squad' in ING's jargon) responsible for the EP is on increasing the adoption and awareness of the EP with other squads. The 268 A/B tests are initiated by 76 unique squads. Given that ING has over 700 squads, there is clearly room for improving this figure.

### 3.1.2. Hypothesis analysis

As mentioned in the introduction section, part of an A/B test is formulating a Null Hypothesis. This hypothesis should be about how and why the treatment variant does not outperform the control variant. In order to get a better grasp on how users of the Experiment Platform set up their experiments, a simple analysis of these hypothesis is made. In figure 3.2 the user interface is shown which is used to formulate the hypothesis. A template option is available, or a free text can be used. Both options result in one sentence being the hypothesis.

Figure 3.2: ING Experiment Platform user interface for writing the hypothesis when creating a new experiment

In table 3.1 the word count of all words in the hypothesis can be found. From this word count it stands out that the default template is frequently used and the hypothesis are not only formulated in English but also in Dutch. Furthermore the high scoring words "clicks" and "CTR" (Click Through Rate) indicate a more business-driven approach towards the experiments at ING.

Word	Count	Part of template
Lead	212	Yes
Measured	211	Yes
More	211	Partially
Clicks	93	No
Page	58	Partially
Increase	43	Partially
CTR	43	No
Meer	42	Partially
Replacing	35	Partially
Button	33	No
Customers	32	No

Table 3.1: Most occurring words in 268 hypothesis. Partially means the word is not in the fixed part of the template, but suggested by the placeholder text of the input field.

### 3.1.3. Position within ING

Room, and hence budget, for maintaining and improving the Experiment Platform goes hand-in-hand with the position it has within the organisation. In order to further improve the EP, budget and dedication is required. However this budget is only made available when the EP is actively used (and a higher-up manager believes in its use-case). Active usage in its turn can be limited due to engineers requiring more features. This creates a negative spiral which could turn out worrisome for the platform. To get a better grasp of the current situation the Experiment Platform is facing the following topics are explained; competition with 3rd party vendors within ING, and the current maintenance and development position. In paragraph 3.1.4 the current

analytics data logging is discussed, which shines a light on limiting features of the EP. Using the Experimentation growth model by Fabijan et al. [25] ING's experimentation would qualify as in between the "crawl" and "walk" stage.

#### Development and maintenance

The development of the Experiment Platform started on March 23rd 2015 with the commit "First commit initial project". The first experiments on the platform were conducted in March 2016. After development of the platform the Panama squad, responsible for the development, dissolved, and maintenance of the platform came in the hands of the Tetris squad, located in Belgium. This squad is led by Raf Ulrix, and has recently taken on more development capacity in order to start development on new features after the summer of 2019. From the Analytics tribe Kevin Anderson is the driving force behind the Experiment Platform both during the development phase as well as the current maintenance phase.

#### Competing platforms

Although the Experiment Platform is an internally developed solution at ING, there is competition from a 3rd party solution: Adobe Target (AT). Squads using AT should migrate towards the Experiment Platform, however this process is slow. From a conversation with Raf Ulrix it is learned that this is mostly due to AT having a greater feature set, and that especially squads in ING Belgium are still using AT. All research in this paper is only focused on the Experiment Platform, AT is considered out of scope.

#### 3.1.4. Webtrekk

Webtrekk helps to connect, analyze and activate user and marketing data across all devices<sup>1</sup>. ING uses business monitoring by Webtrekk to collect information about the use of ING webpages and mobile applications by ING customers and prospects. Hence Webtrekk has valuable information on how users interacted with the software. However this information is currently not linked to the Experiment Platform. In regards to saving user information the EP is only capable of registering visits and successes (e.g. user clicked a button). Every visitor not reaching the success action is therefore considered as failed.

*The Experiment Platform only has user visit counts and binary data: success or failure.*

Two ING engineers are currently occupied with making a link between the Experiment Platform and Webtrekk. This link would transfer the user variant information into Webtrekk, enabling Webtrekk to output the A/B testing results. However currently this is a manual and cumbersome task.

#### 3.1.5. Technical background

The Experiment Platform is a Java application. It uses a DTAP (Develop, Test, Acceptance and Production) approach for environments separation. In the Test, Acceptance and Production environment the system runs on Tomcat servers, using Cassandra as database. Releases are automated through an internally ING developed pipeline called "CDAas".

From the EP documentation it is learned that: *In the production environment the EP runs on 4 different Tomcat instances, divided over 2 different datacenters.*

## 3.2. Essentials

In this section the availability of the essential features as described in paragraph 2.4 is discussed. Some of the essential features are validated on data generated by past experiments with the platform, others are based on interviews with ING experts or by inspection of the EP source code.

### 3.2.1. A/A Testing

To consistently check the data quality of the EP A/A tests should be run on a regular basis. This is not the case at ING. A/A Tests have been conducted, one example is the "Smoke test op MING" (more information provided in paragraph 3.2.5). The EP however does not explicitly keep track if the experiment was set up as A/A.

---

<sup>1</sup><https://www.webtrekk.com/>

For developers it is possible to set-up an A/A test through the EP. They can use its API to determine whether or not a user is in control or treatment, and always show the same features/interface for both variants. This is also the case for the content creators. As can be seen in figure 3.3 the interface for starting an experiment through the CMS seems to force the treatment variant to have a different URL. This difference in URL could enable the Hawthorne effect, however when a user is exposed to a variant other than the control the URL remains the same, thereby mitigating this effect.

The screenshot displays the user interface for creating an experiment. It is organized into two main sections: 'Control' and 'Variant'.  
 - The 'Control' section includes an 'Original URL' text input field with the value 'https://test.mijn.ing.nl/particulier/mijn-gegevens-en-instellingen/betaalpas/index.html' and a 'Description' text area with an information icon.  
 - The 'Variant' section includes a 'Variant URL' text input field with the value 'https://test.mijn.ing.nl/particulier/mijn-gegevens-en-instellingen/betaalpas/index\_variant1.html' and another 'Description' text area with an information icon.  
 - At the bottom of the interface, there is a button labeled '+ Add variant'.

Figure 3.3: ING Experiment Platform user interface for content creators starting an experiment for pages generated through the CMS.

### 3.2.2. Statistical significance tests

By examining the Java source code of ING's Experiment Platform it is found that only the Chi-Squared method is used for determining if the  $H_0$  holds or should be rejected. The computation of the Chi-Squared value in the EP is done by a custom implementation, with no trace of Yates' correction. In order to validate this custom implementation all calculations have been re-executed in Python by using the *chi2\_contingency* package from *scipy.stats*<sup>2</sup>.

When using the *chi2\_contingency* package with the correction parameter set to **false**, the Chi-Squared values for all experiments were identical to the 10<sup>th</sup> decimal, with only one experiment having an equal value to the 2<sup>nd</sup> decimal. With the Yates correction parameter set to **true**, this number drops to only 104 out of 268 experiments having the same Chi-Squared value. This means that the p value for 164 experiments differs, which could result in a different outcome to whether or not the treatment variant is significantly outperforming the control variant. By computing the p values, and using the threshold of  $p < 0.05$  for significance, it is determined whether or not the lack of Yates's correction caused any difference in outcome within the current 268 experiments. This was not the case, however this scenario could occur with future experiments.

By using the Shapiro-Wilk test<sup>3</sup> we determined that 124 experiments can correctly be considered as having normally distributed data. 77 Experiments show non-normally distributed data. The other experiments have too little data to compute the test.

### 3.2.3. Statistical Power

As mentioned earlier, the statistical power of an experiment is an important part of the significance of the experiment. To validate if ING's EP is correctly checking for statistical power, equation 2.1 is used. ING's EP does not require experimenters to enter a number for the sensitivity  $\Delta$  they want to achieve, therefore the statistical power is checked for a  $\Delta$  of 1%, 5% and 10%. For determining the  $\sigma$  of every experiment the Bootstrap method [23] is used.

Table 3.2 shows that even for a sensitivity of detecting changes as big as 10% in success ratio there are experiments in which not enough users participated to statistically correctly detect this change. This is related to ING's EP not monitoring unused or test experiments. However, as can be seen in table 3.2, there is

<sup>2</sup>[https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.chi2\\_contingency.html](https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.chi2_contingency.html)

<sup>3</sup><https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.shapiro.html>



Experiments	$\Delta 0.01$	$\Delta 0.05$	$\Delta 0.10$
Not enough power	110	40	17
Not enough power, and significant according to EP	22	4	1

Table 3.2: Number of experiments not having enough users to statistically correctly detect 1%, 5% or 10% sensitivity. The second row indicates the number of experiments that did not have enough power, but for which the EP showed the user a winning variant with the message that it was significant.

one experiment (width id content-247916) that did not have enough users to detect the 10% change, but is considered by the EP to have a significant result. This experiment has had 67 users, but should have had 4,839 users to have enough power. According to the EP’s data the control variant registered 3 successes and the treatment 0. As can be seen in figure 3.5, it concludes the control variant as statistically significant and being the best without giving any warning about the lack of power. The four experiments that are significant according to the EP, but lack the power with  $\Delta = 0.05$  can be found in figure 3.4.

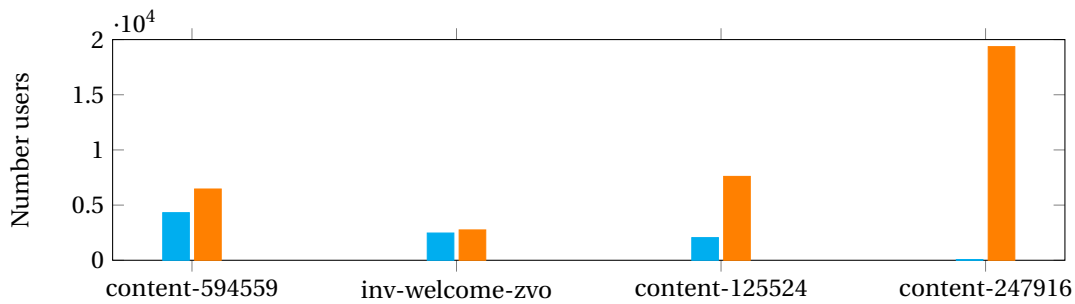


Figure 3.4: The 4 experiments significant according to EP, but lacking the required user numbers for enough power with  $\Delta 0.05$ . Left column (cyan) shows actual visitors, right column (orange) shows required visitors for enough power.

### 3.2.4. Effect size

Since the data of the EP is either success or failure, it is ordinal data. Therefore Cliff’s delta [11] can be used to determine the effect size between the control and treatment groups. Cliff’s delta determines how often values in one distribution are different (larger or smaller) than in the distribution it is compared to. This delta, or  $d$  is given by:

$$d = \frac{\sum_{i,j} [x_i > x_j] - [x_i < x_j]}{mn} \tag{3.1}$$

In equation 3.1  $n$  is the size of the first given distribution and  $m$  of the compared distribution with items  $x_i$  and  $x_j$  respectively. The square brackets  $[\ ]$  are Iverson brackets [35], meaning the value within the brackets is either 1 (true) or 0 (false). The absolute value of  $d$  indicates the level of effect size, as can be seen in table 3.3.

Cliff’s $d$	Interpretation
0 - 0.147	Negligible
0.148 - 0.329	Small
0.330 - 0.473	Medium
0.474 - 1	Large

Table 3.3: Interpretation of Cliff’s delta

To compare the effect size of the experiments at ING two distributions are created. One for the control variant, and one for the treatment variant. These distributions are filled with the number of successes per day for the respective variant. In order for the computation of Cliff’s delta to work these distributions should have the same number of items, and both be non-zero. In 82 of the 268 A/B Tests at ING this was not the case. Some of these 82 experiments were only created but never received data, others had an extra day of measurements in either the control or treatment variant. The results of the remaining 186 experiments in regards to the effect size can be seen in table 3.4.

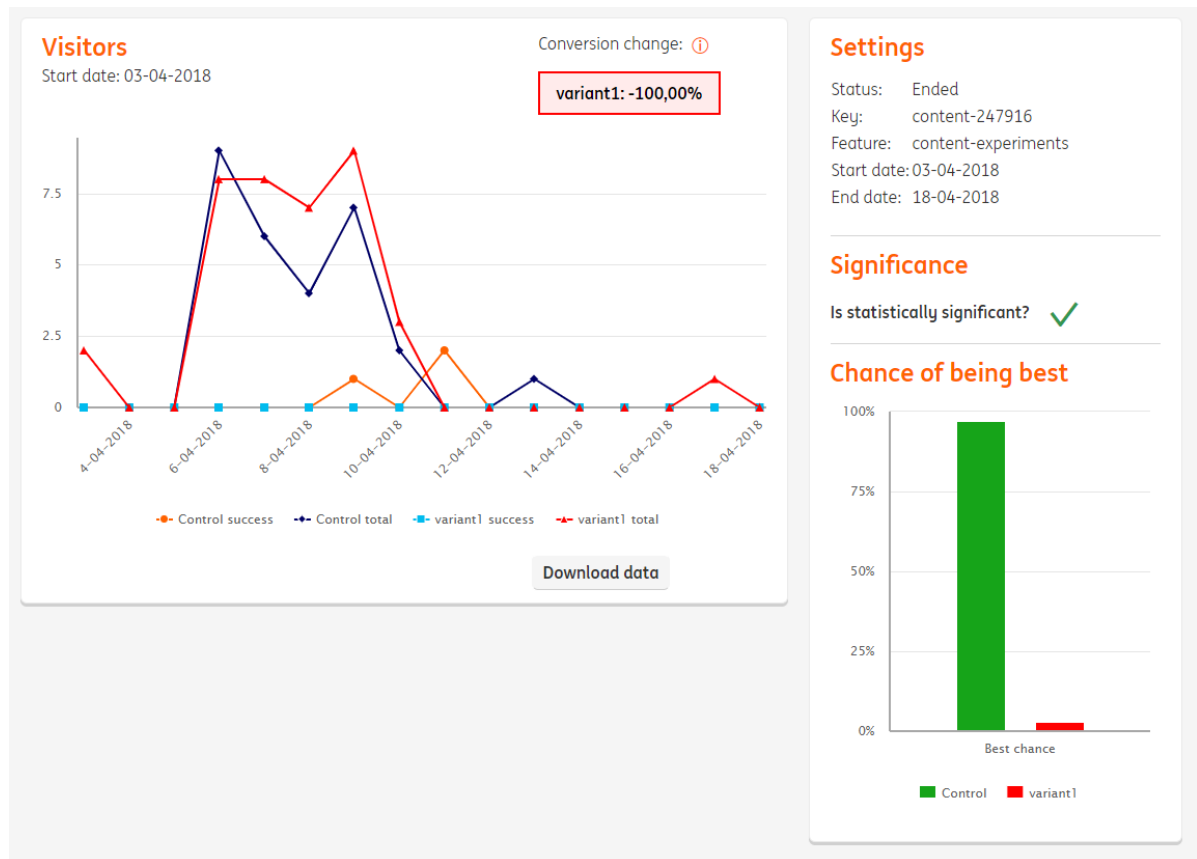


Figure 3.5: ING's Experiment Platform Dashboard for experiment content-247916. No warning in regards to insignificant power is given.

Effect size	Experiment count
Large	22
Medium	8
Small	28
Negligible	128

Table 3.4: Number of experiments per effect size

### 3.2.5. Randomisation and Sample Ratio Mismatch

In ING's EP the concept of SRM is taken into account. Every experiment has an "srmTests" object per variant containing the computed p value and whether or not the variant should be rejected (boolean) based on the p value. The documentation of ING's EP mentions: "We reject the null hypothesis if the p value is lower then **0.0001**" and provides an Excel sheet created by Kohavi [43] on how the calculation is done. The mention of the threshold of 0.0001 is off by a factor of 10 compared to what Kohavi uses in the Excel sheet, however in the source code of the EP the (correct) value of 0.001 is implemented.

To validate whether or not the SRM implementation of ING's EP is correct, both tests from paragraph 2.4.4 are run against the data for all experiments. Because ING's EP does not enforce users to stop an experiment before showing results, many experiments are (partially) terminated on the code side but not on the EP side. In some cases this results in the chosen variant still reporting back visitor counts, in contrast to the killed variant. This skews the data stored in the EP. Hence before running any tests the data is sanitised by assuming the experiment was stopped the day before any of the variants received zero visitors, and discarding that day and forward.

With the sanitised data the Binomial Test and the SRM Test are performed. The Binomial test is based on the *binom\_test* method of the *scipy.stats* Python package. The SRM Test is computed according to equation

2.2. The results of these tests can be found in table 3.5. From this table it is learned that the Binomial Test and SRM Test yield the same results, as was expected. However the EP has three more experiments flagged as SRM for  $p < 0.001$ . When looking into detail in those three specific experiments (*ivy\_sliding\_cats*, *ivy\_sticky\_cta\_d* and *ivy\_mortgage\_assist*) it is found that all three of these experiments have manually selected winner variants in the EP. Whenever a winning variant is selected by a user of the EP, all traffic is directed towards that variant resulting in the other variants not receiving any traffic. However when computing the SRMTest the EP does not take the date the winner variant was selected into account, and therefore detects SRM due to the losing variant(s) not receiving traffic after this date. This is obviously a False Positive. The EP does not compute an SRM result for  $p < 0.05$ .

	<b>p &lt; 0.001</b>	<b>p &lt; 0.05</b>
Binomial Test	13	32
SRM Test	13	32
Experiment Platform	16	Not computed

Table 3.5: Number of experiments where SRM is detected based on thresholds  $p < 0.001$  and  $p < 0.05$ . The Experiment Platform row shows the number as computed by the Experiment Platform.

Out of the 13 experiments that have SRM, seven are considered to have a significant winning variant according to the EP. Although the SRM should void the results for these experiments, the EP does not communicate the SRM to the platform user creating the impression of a successful experiment. As can be seen in figure 3.6, *ivy\_new\_hsi* clearly has SRM. However the team decided to roll-out the new-design variant. In most cases the SRM is not so clearly visible, and therefore the warning is of even more importance.

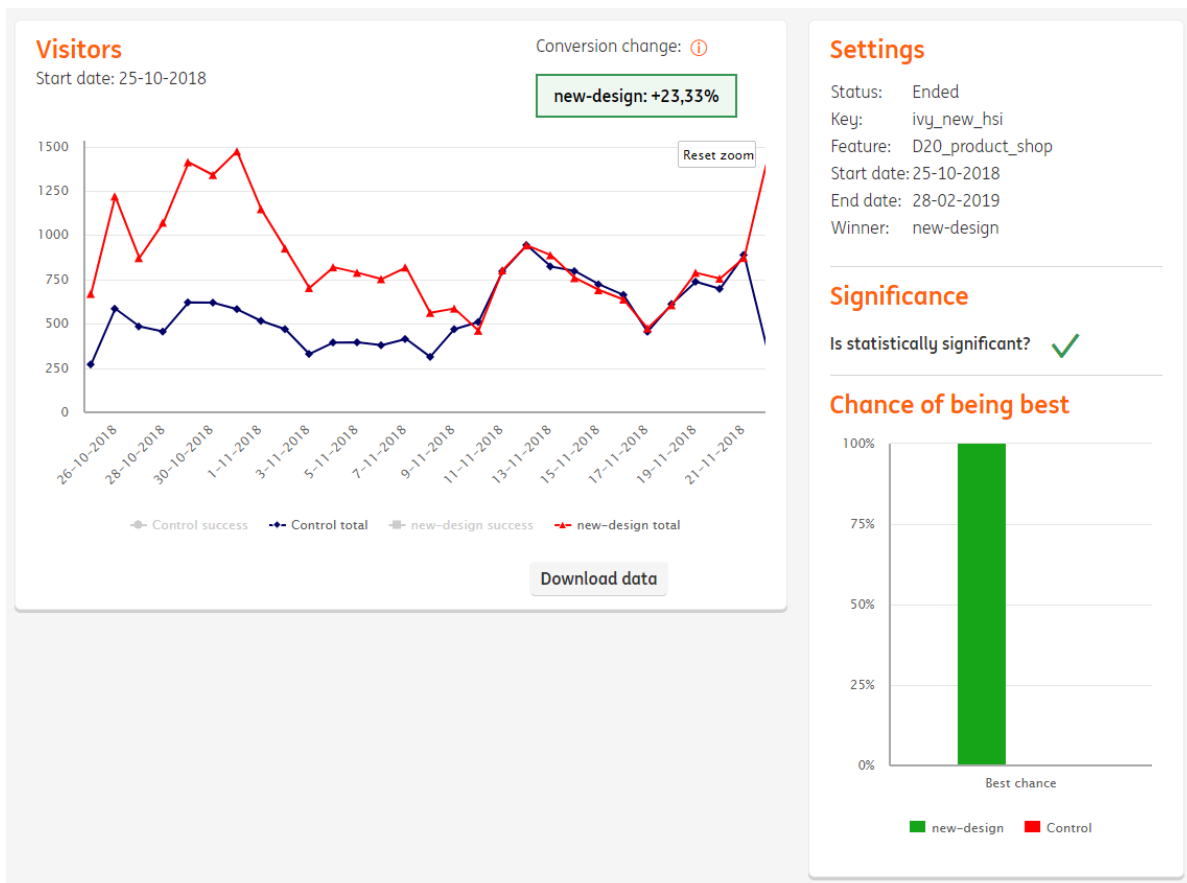


Figure 3.6: ING's Experiment Platform Dashboard for experiment *ivy\_new\_hsi*. This experiment has SRM, however no warning is given to the user and the significance is still shown with a green checkmark suggesting everything went according to plan. The visitor count chart is cut off on 21-11-2018, because on that day the winner variant was selected by a user and the day after the control variant went to 0 visitors.

As mentioned in paragraph 2.4.4 it is of importance to ensure the randomisation of users follows the expected distribution. Having 13 experiments with SRM, it is impossible to say whether or not the SRM in these experiments is introduced thanks to the randomisation or due to any other factor.

In order to get an indication on the randomisation table 3.6 shows the five most visited experiments and whether or not SRM occurs. When computing the top ten visited experiments, four show SRM for  $p < 0.05$ . From the EP change-log it is learned that on 31-10-2017 a deploy to the platform is made that included the following change: "*GET /active-variants now allocates users to a random variant instead of the least visited one*". A conversation with Kevin Anderson confirms that until this release the EP programmatically ensured all variants were receiving equal amounts of traffic, meaning that for some time interval all traffic would go to one variant and for the next time interval the other variant would get all traffic. This feature introduces all kinds of biases, and is correctly removed in favour for a true random generator.

Experiment	Start date	SRM p < 0.05	Total number of visitors
Smoketest_homepage_MING	20-07-2016	No	267.7M
dba-sticky-cta	28-03-2019	No	3.1M
investment-news	08-07-2016	No	1.69M
mg-image-abtest	04-10-2017	No	1.29M
LogoffZakelijk	11-12-2017	Yes	0.81M

Table 3.6: Top 5 of most visited experiments since start of ING's Experiment Platform

When computing the top-10 of most visited experiments that started after 31-10-2017, six out of ten experiments are showing SRM. Although this does not give a definitive conclusion about the randomisation, it is worth noting that for its randomisation the EP source code relies on the *ThreadLocalRandom* Java function, for which the Oracle documentation states: "A random number generator isolated to the current thread."<sup>4</sup>. From paragraph 3.1.5 it is recalled that the EP runs on four different server instances, divided over two different datacenters, hence utilising multiple different threads violating the constraint of the *ThreadLocalRandom* function.

Whether or not the randomisation is actually the problem of some of the SRM cases could be tested by sending generated traffic to a test experiment. However ING's EP works with logged-in users, and only ten test users are made available for this research.

### 3.2.6. User group assignments

ING's Experiment Platform uses the client-side assignment methodology discussed in paragraph 2.4.5. This methodology supports running A/B tests on a variety of platforms, not limited to web application only.

However, as mentioned in paragraph 3.2.5, the EP currently only works with logged-in users. Given that for this research only ten test users are available, no conclusive research can be performed to validate the correctness of the user group assignment.

Working solely with logged-in users is however a perfect use-case for the hash based approach, which would render any randomisation problems obsolete.

### 3.2.7. Data aggregation

From the methods mentioned in paragraph 2.4.6 ING's EP uses the existing data collection in combination with the service-based collection. It is of importance to recall that the ING's EP relies solely on the visitor counts and success/failure rates. Measuring whether or not a visit by a user was a success or not is done in two different manners and depends on whether the experiment was set-up by a content creator or a developer. In the latter case the 'success' is initiated by an API call to the EP. However when a content creator initiates an experiment through the CMS the success/failure is registered through a more complex system of data collection. All of ING's front-end services communicate with a dedicated NGINX monitor server to log events by users. This monitor server propagates this information into two different pipelines for aggregation. The first one is the Webtrekk instance (more in paragraph 3.1.4) which stores the data in a processed manner. The other pipeline is a KAFKA BUS<sup>5</sup> which makes these events available to other services. One of those

<sup>4</sup><https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ThreadLocalRandom.html>

<sup>5</sup><https://kafka.apache.org/>

services is the EP. Whenever an event is related to an experiment created in the CMS, it is processed by the EP. Such an event can consist of the assignment of a user to a specific variant, or the registration of a success event.

The sparse number of test users available for this research are not enough to compute a real test with large visitor numbers which could be used to find any possible data loss. An interesting future research could be the comparison of data as seen in the Webtrekk collection versus the data in the KAFKA BUS. A similar comparison has been performed in Booking.com's double pipeline approach, and revealed inconsistencies.

### 3.3. Pitfalls

Similar to the analyses of the essentials in paragraph 3.2, in this chapter the occurrence of the pitfalls from literature at ING is investigated.

#### 3.3.1. Overall Evaluation Criteria

As mentioned earlier, ING's EP relies on binary success data to make conclusions when comparing variants. This means the OEC is always the success ratio of the variant. However how this 'success' is defined differs per experiment.

For all experiments created by content creators a visit is considered a success whenever during the visit the users navigates to a predefined page, and/or generates a predefined event. Since developers are able to send a 'success' call to the API whenever they deem the visit successful, they have more variability in what exactly is a success.

From the EP's data only the hypothesis gives some insight into what the success criteria might have been. Table 3.1 shows the most occurring words in the hypothesis. The two most occurring words not part of the template are 'Clicks' and 'CTR' (Click Through Rate). From paragraph 2.5.1 it is recalled that a common pitfall in choosing an OEC is focusing on short-term goals: clicks [13]. For ING's case this pitfall therefore seems to be applicable.

Aside of the template, no regulation or checking of OECs is performed in the EP.

#### 3.3.2. Primacy and newness

In order to detect any primacy or newness effects in the 268 experiments conducted within ING's EP the methodology introduced in paragraph 2.5.2 is used. For  $\alpha$  and  $\gamma$  the proposed values of 0.35 and 2 by Chen et al. [10] are used. One experiment flags for the primacy and newness effect for four different variants within the experiment. However, as figure 3.7 shows, the data of this experiment is unhealthy.

ING's EP has no form of automatically detecting primacy or newness effects.

#### 3.3.3. Page speed/latency

Within ING's EP variant load times are not monitored, therefore no validation on this part can take place. This means that whenever a variant would have an increase or decrease in load times the EP would be unaware. An ING expert confirms that page load time is however present within Webtrekk, but it is considered to be an unreliable metric which would render the usage for the EP void.

#### 3.3.4. Day of week effects

As mentioned in paragraph 3.2.5 ING's EP does not enforce users to turn off the experiment in the EP. This results in an experiment still running according to the EP, however only one variant is still receiving traffic, meaning the user actually already decided the experiment is over and chose a winner. Using the methodology described in 3.2.5 the actual end date of all experiments is determined. In figure 3.8 the duration of all experiments are plotted. From this information it is found that 5 experiments have run for more than 120 days which is considered to be too long [38]. One experiment has already been running for more than 572 days, and was still running on the day the data was collected (July 1<sup>st</sup> 2019).

In order to determine if the day of week effect could be applicable on the experiments, a modulo 7 opera-

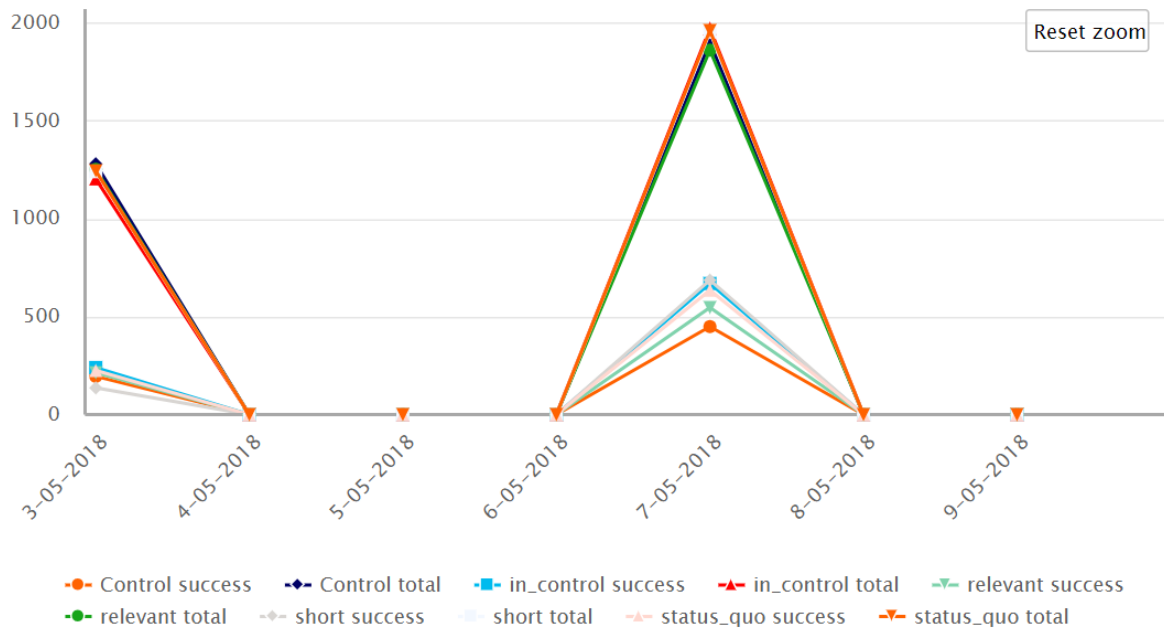


Figure 3.7: Experiment is showing unhealthy data since no visits have been registered between 03-05-2018 and 07-05-2018.

tion has been applied to the duration. Whenever the modulo 7 returns 0, the experiment has run for an exact number of weeks, therefore it is considered that the day of week effect is not present within that particular experiment.

51 out of 207 experiments that run for more than 0 days, are vulnerable to the day of week effect. ING's EP does not give any warning in regards to day of week effects.

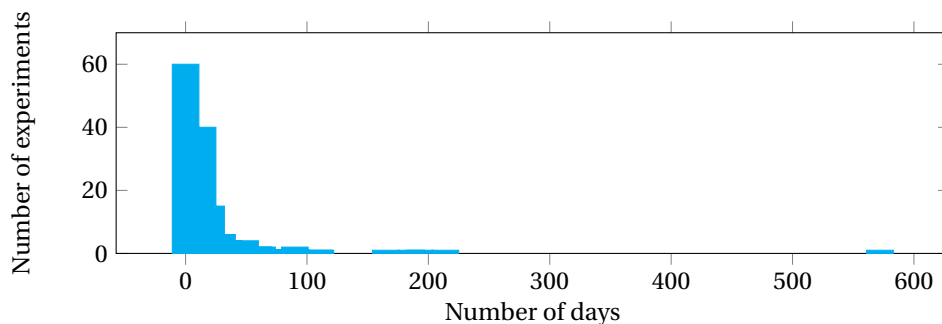


Figure 3.8: Number of experiments plotted against the number of days they have run. The experiment running for 572 days was actually still running when the data-set was created for this plot on July 1<sup>st</sup> 2019.

### 3.3.5. Beacon (loss)

Problems with the beacon are complicated to find since the result of a malfunctioning beacon is unhealthy data. This data problem would typically result in SRM, giving an indication to investigate further. As mentioned in paragraph 3.2.5, SRM is occurring within experiments at ING, but no warning of SRM is given to the user of the EP.

Mobile applications are currently not using ING's EP, so handling of these batched events cannot be validated.

### 3.3.6. Browser differences

Information on what browsers are used by participants of the experiments are not stored within ING's EP. Therefore any influences of specific browsers can not be detected by the EP. Browser information is however available within Webtrekk.

### 3.3.7. Carry over effects

ING has a broad range of consumer facing software products and experiments which can be run on all of these products. Although a typical end-user does not use all of these systems, the chance still exists that a user is exposed to more than one experiment at the same time. Since this chance increases when more experiments are run simultaneously a Gantt chart is made to verify whether or not this scenario could occur. As can be seen in figure 3.9, multiple experiments do happen at the same time. To narrow down the chance of a user actually interacting with more than one experiment simultaneously the number of overlapping experiments per squad is computed. Table 3.7 shows the squads with the most internally overlapping experiments.

Squad	Number of experiments	Overlaps
Panama	22	49
DoItExpress	29	29
Cashless	20	20
SquadVermogen	34	19
DigiChan	14	16
Ivy	9	10
ActiPay	14	6
persoonlijkvermogensadvies	7	3
expleen	6	3
FutureFit	9	2

Table 3.7: Top-10 squads with most overlapping experiments within the squad. Every time an experiment overlaps with a different experiment is considered an overlap. More overlaps than experiments is an indication of many experiments run at the same time. Panama is the squad that built ING's EP, the high number of overlaps is explained by testing experiments in the production setting.

Although start and end date of experiments are clearly stated in ING's EP, no warning system is in place whenever a squad starts multiple experiments. The fact that overlapping experiments within software made by particular squads occurs, gives an indication that carry over effects might occur. However it does not prove it happened. Proving this could be done by rerunning some experiments without the parallel experiments, and investigating the differences that might occur.

### 3.3.8. Robots

Within ING's EP no mitigation of robot traffic is made. Whether or not this filtering occurs higher-up in ING's firewall is unknown, however in both occasions it would result in an equal division of robot traffic over the control and treatment variants. This would only be of relative small impact [37].

### 3.3.9. Staged roll-out with Simpson's Paradox

In order for the Simpson's Paradox to occur the possibility to perform a staged roll-out is a boundary condition. In ING's EP the option to release a feature through a staged roll-out is present, however this option is separated from the A/B test feature. A staged roll-out from ING's EP will not provide variant significance information. The possibility to slowly ramp up one variant of an A/B test is not existent.

This set-up means that Simpson's Paradox can not occur within ING's Experiment Platform.

### 3.3.10. Device Time

In order to validate whether or not device times issues can arise, a set of tests has been conducted. Using the EP's API endpoints we tried to register visits and successes for ended experiments, as well as visits and successes in the past. Neither of those succeeded. Investigation of the source code reveals that the EP, correctly, uses server time to register a visits or success and checks whether or not the experiment is still active.

### 3.3.11. Browser redirects

Although developers using ING's EP could implement a browser redirect for showing a different variant, neither developers nor content-creators are dependent on browser redirects. From the EP's documentation developers are encouraged to not implement a redirect for showing different content. ING's EP does not keep track of the Referer header in requests, making us unable to detect the occurrence of browser redirects. The Referer header is however used to narrow down the audience of an A/B test.

### 3.3.12. Error checks

From the error checks as described by Tang et al. [60] only the check for a control variant is partially supported. ING's EP enforces developers and content creators to have a control variant. However any checks on whether or not this control variant actually is implemented correctly is lacking.

As seen in paragraph 3.2.5 SRM checks are implemented, however the results are not shown to the user. This means that any errors occurring might be flagged by the SRM check, but the EP's user stays unaware.

### 3.3.13. Unplanned differences

Unplanned differences are difficult to automatically check. During the validation process required for the paragraphs in this chapter one experiment stood out. Experiment *content-517397* is initiated by a content creator and has the following hypothesis: "Replacing 'make an appointment' with 'call me now' will lead to increasing clicks on cmn button measured by click"<sup>6</sup>. According to the EP, variant 1 shows a statistically significant infinite ( $\infty$ ) improvement over the control variant. The treatment variant has had 77 visitors, of which 32 reached the success state. However the control variant had 73 visitors, with 0 visitors reaching the success state. Since this is a CMS based experiment, problems with the beacon implementation are not expected. An unplanned difference is highly likely causing the 0 successes for the control variant.

---

<sup>6</sup>Actual sentence: "By replacing maak een afspraak voor bel mij nu will lead to increasing clicks on cmn button measured by click"





Figure 3.9: Experiments plotted with start date and duration. The increasing number of experiments results in an increased chance of multiple experiments running at the same time.



# 4

## Supporting Continuous Experimentation

The previous two chapters show how easily mistakes can be made in Continuous Experimentation. These mistakes are not limited to the set-up and analysis of experiments, but can also easily surface during the development of an Experimentation Platform. Since bad data can be actively worse than no data [27], a semi-automated check on healthy data can be of importance.

In this chapter the basis for creating an Open-Source A/B Test validation tool is made, resulting in the answering of **RQ3**.

### 4.1. Introducing ABvalidator

Providing support for a Continuous Experimentation Environment can only be achieved in a semi-automated manner. This automation makes it possible to capture the knowledge, currently divided over papers, researchers and industry experts. Furthermore this automation makes the solution viable for the growth in the sheer number of experiments conducted, as for example is seen at ING (see 3.1).

The following paragraphs introduce ABvalidator, an automated toolkit to validate experiments and Experiment Platforms, is made. ABvalidator will be, for as far as I'm aware, the first and only automated Open-Source tool for this matter. ABvalidator combines the literature gathered in chapter 2, scripts used for the research of chapter 3 and the knowledge and feedback of industry leaders. ABvalidator is a tool that combines pragmatic questions about the experiment set-up with in-depth statistical calculations to cover the broad scope of A/B Testing.

### 4.2. The three A's of A/B test validation

As outlined in chapter 2 many different aspects of A/B testing can influence the validity of an experiment. Combining the essentials and pitfalls from this study with the three key checklists introduced by Fabijan et al. [27], an extensive list of topics is made. These topics range from whether or not data is available to validation of the results on a dataset. Two type of users are identified: experimenters and maintainers. The experimenter is a user whom runs a single experiment and wants to externally validate the results of his or her experiment. On the other hand, the maintainer is part of the development team of the Experiment Platform and is looking for pointers on how to improve the platform. Depending on the type of user the number of relevant questions varies.

Fabijan et al. [27] use a segregation of topics based on before, during or after the experiment. However since the questions used in this study are set-up for a validation purpose, this categorisation does not suit the purpose. Therefore I introduce three new categories: the three A's of validating A/B tests: **Availability**, **Analysability** and **Accuracy**. Down below these A's will be explained in more detail.

#### 4.2.1. Availability

Since A/B testing is about making conclusions based on data, it is of importance that the required data to make these conclusions is available. The following checks should be performed.

### Experiment hypothesis should be defined and falsifiable

Every experiment should have a hypothesis describing what exactly the to be tested change is, the impact that is expected and why it is expected that way [27]. Furthermore it is important that this hypothesis is actually falsifiable, otherwise the experiment will not be conclusive. In platform context, it is expected that the EP asks for this hypothesis and supports experimenters with a template. Questions for this topic can be found in table 4.1.

#	User	Question
<b>1a</b>	E	Did you specify a hypothesis describing the change, expected impact and reasoning behind the expected impact?
	M	Does the EP require to input a hypothesis that describes the change, expected impact and reasoning behind the expected impact?
<b>1b</b>	E	Is your hypothesis falsifiable?
	M	Does the EP check whether or not a hypothesis is falsifiable? Or is there a system in place to verify whether or not hypothesis are falsifiable (e.g. by having them checked by peers)?

Table 4.1: Questions regarding the experiment hypothesis for Experimenters (E) and Maintainers (M).

### Metrics and their expected movement are defined

Metrics need to be chosen that reflect the data quality and success of the experiment [24, 25, 31]. Guardrail metrics should also be in place [27]. Similar to the hypothesis, the direction of change of the metric should also be defined. Questions for this topic can be found in table 4.2.

#	User	Question
<b>2a</b>	E	Are you able to gather data on the exact metric that you want to impact?
	M	Is the EP capable of tracking the metrics that experimenters want to impact?
<b>2b</b>	E	Did you in advance specify the direction of change of your metrics?
	M	Are experimenters forced to in advance specify the direction of change of their metrics?
<b>2c</b>	E	Did you keep track of any guardrail metrics that should not be impacted negatively?
	M	Are guardrail metrics automatically tracked for every experiment?

Table 4.2: Questions regarding the experiment metrics for Experimenters (E) and Maintainers (M).

### The risk associated with the experiment is managed

Fabijan et al. describes three risk scenarios that need to be managed when running an experiment [27]. First off the emergence of technical debt should be taking into account in the life cycle of running an experiment. Secondly when running an experiment with real users, the possibility that competitors get an early glimpse at the direction of product development is present, plans to mitigate this risk should be in place. The third risk point focuses on a set of users having a (very) bad experience as a result of the experiment which makes them churn. Questions for this topic can be found in table 4.3.

### The minimum effect size and experiment duration are set

For the experiment to be decisive a certain minimum number of participants is required. This number is based on the minimum effect size ( $\Delta\%$ ) the experimenter wants to be able to detect. Based on historical data and the minimum effect size the duration of the experiment can be determined. Questions for this topic can be found in table 4.4.

#	User	Question
3a	E	Have you considered the technical debt introduced by having an experiment in your code?
	M	Are experimenters made aware of the risk created by the technical debt introduced by running experiments?
3b	E	Did you think about and/or mitigated the risk of competitors getting early glimps at your product development by running the experiment?
	M	Are experimenters made aware of the risk of competitors getting early glimps at new product ideas in development?
3c	E	Are you aware that a (very) negative experience for users in the experiment due to some unforeseen fault could make them churn?
	M	Does the platform remind experimenters that they work with actual live users, and that any (very) negative experience for these users could result in making them churn?

Table 4.3: Questions regarding the experiment risks for Experimenters (E) and Maintainers (M).

#	User	Question
4a	E	Did you specify the minimum percentage of change, the effect size, you wanted to measure on the metrics in advance?
	M	Are experimenters asked for the minimum effect size they want to measure on their metrics?
4b	E	Is the minimal duration of the experiment calculated in advance based on the effect size and historical data?
	M	Does the EP calculate the minimal duration of an experiment based on the effect size and historical data?
4c	M	Does the EP withhold results from the experimenters until the minimal required users have participated to detect the set effect size?

Table 4.4: Questions regarding the experiment effect size for Experimenters (E) and Maintainers (M).

### Overlap with related experiments is handled

Although overlapping between experiments can not completely be eliminated through pre-experiment coordination and testing [27], analysis to detect such interactions among experiments need to be in place. Questions for this topic can be found in table 4.5.

#	User	Question
5a	E	Do you run multiple experiments simultaneously? And if so, are you aware that they might influence each other?
	M	Does the EP warn experimenters for experiments running simultaneously that can effect each other?

Table 4.5: Questions regarding experiments overlapping for Experimenters (E) and Maintainers (M).

### Criteria for alerting and shutdown are configured

A running experiment can have an unintentionally significant negative impact on the business, often when the guardrail metrics are violated. Therefore it is of importance that an experimenter or the Experiment Platform is aware of this scenario and able to shut down a running experiment. Questions for this topic can be found in table 4.6.

#	User	Question
6a	E	Did you in advance specify boundaries per metric for when the experiment should be shut down to prevent negative impact?
	M	Does the EP monitor guardrail metrics and warn experimenters or shut down the experiment when these guardrail metrics are negatively impacted?

Table 4.6: Questions regarding experiment shutdown criteria for Experimenters (E) and Maintainers (M).

#### Required telemetry data can be collected

Usually telemetry logging is put in place for debugging or testing purposes [64]. This logging does however not necessarily result in data that reveals how the product is actually used [27]. The creation of a centralised catalogue of (log) events as described by Barik et al. [6] is recommended [27]. Questions for this topic can be found in table 4.7.

#	User	Question
7a	E	Is the data to support your hypothesis available?
	M	Is the experimenter able to select the metrics deemed required for the experiment?
7b	M	Are the metrics collected through a dedicated system for experimentation?

Table 4.7: Questions regarding the collection of metrics for Experimenters (E) and Maintainers (M).

#### 4.2.2. Analysability

The next step in validating an experiment or EP is whether or not all prerequisites are in place to be able to analyse the data in a trustworthy manner.

#### Are A/A tests performed

As mentioned in paragraph 2.4.1 A/A tests are an essential feature for an EP [24, 25]. Not only should the feature be present, a systematic routine for performing continuous A/A tests should be in play [39]. Question for this topic can be found in table 4.8.

#	User	Question
8a	E	Did you perform an A/A test with only the control variant to verify the experiment is set-up correctly?
	M	Are A/A tests performed on a regular basis?
8b	M	Are the p-values for the A/A tests close to uniformly distributed?
8c	M	Is one of the A's significant accordingly with the significance threshold? (e.g. for $\alpha = 0.05$ one in twenty A/A Tests should be significant)

Table 4.8: Questions regarding A/A Tests for Experimenters (E) and Maintainers (M).

#### Experiment design to test the hypothesis is decided

In the design of an experiment it is determined if an experiment is about testing a single change (one-factor-at-a-time) or multiple changes with a MultiVariate Test (MVT). Analysis and interpretation of MVT's are more difficult [39]. Questions for this topic can be found in table 4.9.

#	User	Question
9a	E	Did you specify before the experiment if you are testing one change or multiple at once?
	M	Does the EP require the experimenter to specify in advance if a single change is tested or more than one at once?

Table 4.9: Questions regarding experiment design for Experimenters (E) and Maintainers (M).

#### Experiment owners for contact are known

For every experiment that is ran a group of experiment owners should be identified. It is even recommendable to have multiple owners for a single experiment [27]. This ensures availability of at least one in a scenario that requires an urgent action by an operations engineer [27]. Questions for this topic can be found in table 4.10.

#	User	Question
10a	E	Are multiple people in your team responsible for the experiment you are running?
	M	Is the EP aware which team runs an experiment?
10b	E	Do the operations engineers know your team is responsible for this experiment?

Table 4.10: Questions regarding experiment owners for Experimenters (E) and Maintainers (M).

#### Possibility of early stopping early is evaluated

Understandable one might want to look at the data and analysis during an experiment. This *peeking* might lead to early stopping the experiment because the results are interpreted by the experimenter or because some p-values already show significance. However stopping an experiment early has several pitfalls [13, 21]. If an EP wants to facilitate early stopping, some statistical tests are required [17]. Questions for this topic can be found in table 4.11.

#	User	Question
11a	E	Were you able to look at conclusive results during the runtime of the experiment?
	M	Are experimenters able to look at conclusive intermediate results during the runtime of the experiment?
11b	M	Are statistical tests in place that allow for early stopping?

Table 4.11: Questions regarding early stopping experiments for Experimenters (E) and Maintainers (M).

#### Metrics are examined for selecting the winner

A winning variant should only be designated after a set of checks. First of the data quality should be ensured. Second, feature metrics should be checked to confirm the feature is working as expected. Third, the long-term benefit for users and business should be evaluated. And finally the guardrail metrics should be checked to confirm that no negative impact is occurring [27]. Questions for this topic can be found in table 4.12.

#### In-depth analysis has been performed

An experiment does not end with selecting a winning variant and implementing it for all users. The data from the experiment can be analysed in more depth to come up with new ideas for feature improvements and insights. Questions for this topic can be found in table 4.13.

#	User	Question
12a	E	Did you (visually) check the data from the experiment and looked for anomalies?
	M	Does the EP perform an health check on the data before appointing a winner?
12b	E	Did you verify the winning variant is actually working as expected?
	M	Does the EP ask the experimenter to confirm the winning variant is working as expected?
12c	E	Did you verify the guardrail metrics are not negatively impacted before implementing the winning variant?
	M	Does the EP check the guardrail metrics before presenting a winning variant?

Table 4.12: Questions regarding winning metrics for Experimenters (E) and Maintainers (M).

#	User	Question
13a	E	Did you analyse the results of the experiment to come up with possible future improvements that can be tested?
	M	Does the EP encourage experimenters to analyse the results of an experiment for possible new features that can be tested through experiments?

Table 4.13: Questions regarding analysing the results for new features for Experimenters (E) and Maintainers (M).

### Coordinated analysis of experiments is done

Experiments do not have to be about isolated feature improvements. Fabijan et al. [27] shows that often experiments are part of a coordinated initiative to answer a higher-level business question. If this is the case, the result of an experiment should also be evaluated in this broader higher-level view. Questions for this topic can be found in table 4.14.

#	User	Question
14a	E	Is this experiment part of a set of experiments designed to answer a higher-level business question?
	M	Does the EP allow for experiments to be grouped when they are designed to answer a higher-level business question?
14b	E	Did you analyse the results of this experiment in the broader higher-level view?

Table 4.14: Questions regarding coordinated analysis of the experiment for Experimenters (E) and Maintainers (M).

### Experiment learnings are institutionalised and shared

Not seldom learnings are drawn from experiments. Sharing these learnings within the organisations helps to design future experiments and steer *intrapreneurship* [18, 27]. Questions for this topic can be found in table 4.15.

### 4.2.3. Accuracy

The final step is to validate whether or not the computed results are accurate. This step is the most crucial and will provide the most value. Using ABvalidator both experimenters and maintainers can double check the validness of the results.



#	User	Question
15a	E	Do you have the possibility to share learnings from your experiment in a structured manner with colleagues?
	M	Does the EP facilitate a way for experimenters to share learnings and knowledge about their experiments?

Table 4.15: Questions regarding experiment learnings for Experimenters (E) and Maintainers (M).

### Randomisation quality is sufficient

As described in paragraph 2.4.4 the correctness of randomisation is important. The best method to do this is running A/A Tests, and monitoring the uniform distribution of the p values. Giving the nature of AB-validator however it is not plausible to simply run multiple A/A Tests in a production environment. Therefore, using data from an actual experiment, the quality of the randomisation can be determined using a two tailed binomial test. Thresholds used to calculate if the randomisation holds are 0.05, 0.01 and 0.001. Questions for this topic can be found in table 4.16.

#	User	Question
16a	E & M	How many users have participated in the experiment in total?
16b	E & M	How many users participated in the variant you want to check?
16c	E & M	What percentage of total users should have been allocated to this variant (0-1)?

Table 4.16: Questions regarding randomisation quality for Experimenters (E) and Maintainers (M).

### No serious data quality issues are present

Using the data of an actual experiment the overall health of the data from that experiment can be checked. This is done by performing the Sample Size Ratio Test which is, recalling from paragraph 2.4.4, given by:

$$\chi_{K-1}^2 = \sum_{i=1}^K \frac{(O_i - E_i)^2}{E_i} \quad (4.1)$$

In this equation, which is known to be used at Twitter [9] and LinkedIn [10],  $K$  denotes the number of variants,  $O_i$  is the observed frequency in variant  $i$ , and  $E_i$  is the expected count in variant  $i$ . Equation 4.1 results into the Chi-Squared statistic, and yields the same P value as the two tailed binomial test when computed for two variants. This step of the tool can take in user counts from more than two variants, and detect whether or not SRM is present. The question for this topic can be found in table 4.17.

#	User	Question
17a	E & M	Enter the path of the csv file with user counts per variant

Table 4.17: Questions regarding data quality for Experimenters (E) and Maintainers (M).

### Novelty effects are excluded

As described in paragraph 2.5.2 novelty effects can void the validness of the results of the experiment. It is recalled that this novelty effect can be assessed by generating a delta graph between the control and treatment variant. This delta graph is introduced by Chen et al. [10] and given by:

$$\Delta\%^{[t,t]} = \beta_0 + \beta_1 \frac{1}{t^\alpha} + \beta_2 \frac{1}{t^\gamma} \quad (4.2)$$

In equation 4.2  $\Delta\%^{[t,t]}$  stands for the percentage of impact between day  $t$  and day  $t$ , being the single day impact.  $\alpha$  and  $\gamma$  should be chosen in such a manner that  $\frac{1}{t^\alpha}$  is a slow-decay term and  $\frac{1}{t^\gamma}$  a fast-decay term. Chen et al. have found  $\alpha = 0.35$  and  $\gamma = 2$  as suitable values [10]. When a Multiple Linear Regression is ran on equation 4.2 with the first week of data a primacy or newness effect can be flagged whenever the following three conditions hold:

1. The linear model captures the effect trend well ( $R^2 \geq 0.8$ )
2. The fitted line is monotonic in  $t$
3. The largest impact is statistically significantly different from the smallest impact

The question for this topic can be found in table 4.18.

#	User	Question
18a	E & M	Enter the path of the csv file containing the metric values for the experiment that need to be checked for the novelty effect

Table 4.18: Questions regarding the novelty effect for Experimenters (E) and Maintainers (M).

### Skewed data is treated

Following the possible detection of unhealthy data it is worthwhile to look for any outliers skewing the data [27]. These outliers can be caused by legitimate special users, which might make it useful to look at metrics for different sub-populations. ABvalidator determines outliers by computing the Z-score for every value in the dataset. This Z-score is given by  $z = (X - \mu) / \sigma$ , in which  $X$  is the observed value. When the absolute value of  $z$  is greater than 3, the observed value  $X$  is considered an outlier. The question for this topic can be found in table 4.19.

#	User	Question
19a	E & M	Enter the path of the csv file containing the metric values for the experiment that need to be checked for outliers

Table 4.19: Questions regarding outliers for Experimenters (E) and Maintainers (M).

### Validation of experiment outcome is conducted

Although ABvalidator is able to, partially, validate experiment results, the strongest form of validation is the reproducibility of the experiment. Whenever an experiment only marginally impacts the metrics, it should be executed again to validate the results and learnings [27]. Questions for this topic can be found in table 4.20.

#	User	Question
20a	E	Did the experiment only marginally impact the metrics?
	M	Does the EP encourage experimenters to rerun the experiment when the results only marginally impact the metrics?
20b	E	Did you rerun the experiment, preferably with higher power, to validate the results and learnings?

Table 4.20: Questions regarding validation for Experimenters (E) and Maintainers (M).

### Statistics

The corner stone of A/B testing is statistics. By recalculating the values for the required power of the experiment, the accuracy of the given experiment can be determined. Questions for this topic can be found in table 4.21.

#	User	Question
21a	E & M	What is the standard deviation (sigma) from pre-experiment data? (enter 0 to use the bootstrap method on the current experiment data)
21b	E & M	What is the sensitivity percentage you want to be able to detect?
21c	E & M	What is the path to the experiment data csv?

Table 4.21: Questions regarding the statistics for Experimenters (E) and Maintainers (M).

## 4.3. Setup, distribution and usage

The foremost goal of ABvalidator is to publicly share the gathered knowledge. This goal automatically points to open sourcing the tool, and hence it is published on GitHub. The language of choice is Python, thanks to its broad support for data handling and mathematical libraries.

ABvalidator works by asking the user the set of questions defined in paragraph 4.2 as can be seen in figure 4.1. Every question is accompanied by a short description on the topic, helping out the user to answer the question as intended. Some questions will also provide information on how data should be structured before it can be processed by ABvalidator.

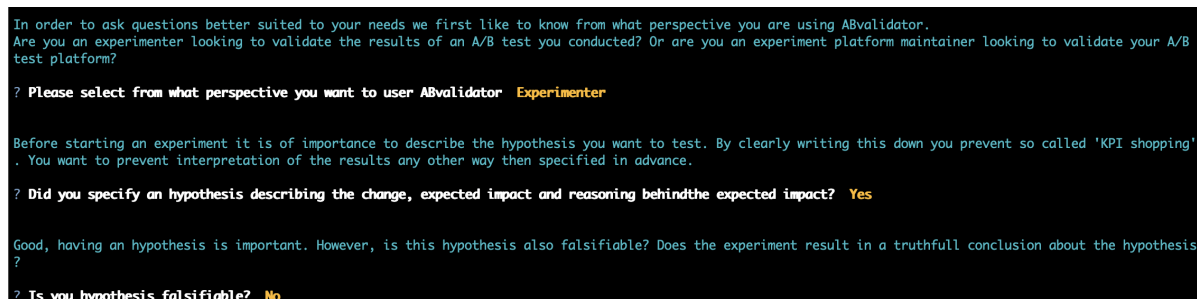


Figure 4.1: ABvalidator question interface

After answering the questions ABvalidator will show a pass or no-pass for every question. Whenever a question is not answered to satisfaction, a more information link is provided to ensure the user is able to gather more information on the topic to encourage improving the outcome. This results interface can be seen in figure 4.2. ABvalidator is Open-Sourced and available through:

<https://github.com/ernstmul/ABvalidator>

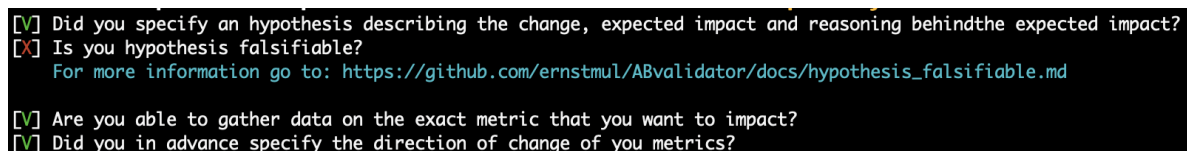


Figure 4.2: ABvalidator results interface



# 5

## Results

In this chapter we briefly summarise the results as presented in the previous three chapters.

### 5.1. RQ1: Factors affecting trustworthiness and soundness of an A/B test

The result of our theoretical literature review is a list containing 19 factors influencing the trustworthiness and soundness of an A/B test. This list is divided over two categories: essentials and pitfalls. Factors that are described by literature to be at least required for running an A/B test are considered essential. Six factors are in this category. As a result of implementing these essentials and running A/B tests pitfalls will arise. These pitfalls consists of mistakes that are commonly made in A/B testing. We have identified 13 factors in this category. The factors are listed in table 2.1.

In the rest of this section we will give a summarised description of these factors. The full description can be found in chapter 2.

#### 5.1.1. Essentials

##### A/A testing

The procedure of testing two identical variants against each other. A/A tests help to identify problems with the experiment setup and Experiment Platform, by rejecting the null hypothesis only as often as the chosen threshold ( $\alpha$ ) indicates (e.g. one in twenty tests for  $\alpha = 0.05$ ). Furthermore the P values yielded by the statistical tests from the A/A test should follow a uniform distribution [31].

##### Statistical significance tests

To compare the results of the variants within the experiment statistical tests, such as the T-test and the Chi-Squared test, are used. These tests yield a P value that, in comparison to the chosen threshold ( $\alpha$ ), indicates whether or not the null hypothesis can be rejected and if the tested change is significant. Issues can arise when these tests are incorrectly performed, or when the underlying assumption of a normalised data set are not correct.

##### Statistical Power and confidence

Power describes the probability of correctly rejecting the null hypothesis, when it is false. The confidence level describes the probability that observed data lies within the described interval. In A/B testing the commonly used value for power is 80% and 95% for the confidence level [39]. Using these values the minimum sample size can be determined by using equation 2.1.

##### Randomisation and Sample Ratio Mismatch

Sample Ratio Mismatch (SRM) describes the scenario in which the sample size per variant is not consistent with the pre-defined ratio. This scenario can be detected using the Sample Size Ratio Test, which can be seen in equation 2.2. Whenever SRM is present within the results of an experiment, it is an indicator of a problem with the experiment (e.g. a not correctly working beacon). Therefore the SRM test is a powerful tool to check the healthiness of the experiment results. A sound randomisation for dividing users over the variants is therefore required, otherwise the error in randomisation will trigger the SRM test.

### User group assignment

When users are participating in an experiment it is of importance that these users are maintained in the same variant throughout the experiment. Having users alternating between variants can result in measuring behaviour not introduced by the change that is to be tested. Three types of user assignment are available: traffic splitting, client-side assignment and server-side assignment.

### Data aggregation

In order to support A/B testing logging the right data should be an integral part of the development process [31]. Any loss of data, errors in the aggregation of the data or unavailability of the data can lead to void the results of the experiment.

## 5.1.2. Pitfalls

### Overall Evaluation Criteria

The Overall Evaluation Criteria (OEC) is the metric, or set of metrics, used to evaluate the performance difference between the variants. The goal for a good OEC is to include factors that predict long-term goals[21], for example predicted lifetime values and repeat visits. Short-term goals, such as clicks, should be avoided [13]. Selecting an incorrect OEC can result in optimising for non intended scenario. Furthermore the OEC should be selected before the experiment, in order to avoid 'KPI shopping'.

### Primacy and newness

Primacy and newness describe the change in the behaviour of experiment participants whom frequently use the software. An experienced user might be less efficient when a segment of navigation is changed. This is called the primacy effect. On the contrary the newness, or novelty, effect describes a user being intrigued because he notices the change, and starts exploring the change (in stead of regularly using it). The newness effect also describes users at first positively interacting with a change, but ignoring it later on. This newness effect can be detected using equation 2.3.

### Page speed/latency

The load time of the content is of great importance in how the user experiences the variant he is presented. When this page speed is not part of the OEC, any speed differences between variants should be mitigated.

### Day of week effects

The specific day of the week (weekend versus mid-week), and even the hour of day (day time, night time) can have influence on the type of users of the variant. Therefore it should be ensured that an experiment is run over an exact number of weeks to prevent day of week effects.

### Beacon (loss)

The beacon is the piece of software that communicates the collected metrics back to the Experimentation Platform. Any problems with this beacon (e.g. a varying position in a web page) should be avoided, since this skews the aggregation of data.

### Browser differences

The differences in how browsers render and handle web pages can influence the results of an experiment. It should be ensured that specific browsers do not over populate one specific variant.

### Carry over effects

When multiple experiments are conducted simultaneously on the same software application the scenario can occur in which a participant is enrolled in more than one experiment. This participant is then influenced by what he is using/seeing in both experiments, leading to a so called carry over effect. Experiment Platforms should try to avoid these carry over effects.

### Robots

Automated visitors, robots, on an experiment can cause misleading results due to their behaviour being not representative for human behaviour. Especially when these robots are only present in one variant.

### Staged roll-out with Simpson's Paradox

When choosing a staged roll-out strategy for an experiment, it can occur that one variant outperforms the other on every single day, yet not overall as can be seen in table 2.2.

### Device time

Relying on the device time of the user can lead to incorrect data points, and should therefore be avoided.

### Browser redirects

Browser redirect is a methodology to direct a user towards the treatment variant. Although this method was commonly used, it should be avoided since it leads to significantly under performance of the treatment variant.

### Error checks

Running automatic checks for errors, e.g. syntax error checks, leads to fewer broken experiments being run [60]. Whenever a specific variant is showing errors, this should be detected otherwise the results can be skewed.

### Unplanned differences between variants

Although an experiment is started to test a specific change, it can occur that variants show more differences than planned. It is advised to have tooling in place that help detect these unforeseen differences.

## 5.2. RQ2: Factors impacting ING's Experimentation Platform

We have used the factors identified in chapter 2 to validate ING's in-house created Experimentation Platform (EP). This validation is done by using the data of 268 A/B tests conducted by ING in the period ranging from January 1st 2016 to July 1st 2019. In this section we summarise the results of this validation per factor.

### 5.2.1. Essentials

#### A/A testing

ING's EP supports the possibility to perform A/A tests. A/A tests have been conducted at ING, however they are not specifically labelled as A/A test, nor ran on a regular basis.

#### Statistical significance tests

By investigating the source code of ING's EP, we have found that only the Chi-Squared method for statistical analysis is implemented. The custom Chi-Squared implementation is made on the assumption that all data is normally distributed. No Yates correction is present. Furthermore running the Shapiro-Wilk test on all 268 experiments shows that this normality assumption is not valid for at least 77 of these experiments.

#### Statistical Power

ING's EP does not require experimenters to enter the sensitivity ( $\Delta$ ) they want to be able to measure, nor is the standard deviation ( $\sigma$ ) available through pre-experimentation. Therefore we have calculated the required statistical power for all 268 experiments using  $\Delta = 0.01$ ,  $\Delta = 0.05$  and  $\Delta = 0.10$ , and by using the bootstrap method for determining  $\sigma$ . When looking for the detection of the largest change ( $\Delta = 0.10$ ) 17 experiments are identified that do not have the minimum required number of participant to be able to detect such a change. One of these experiments is however still considered by ING's EP to have a winning variant. Although post-hoc power analysis are not favoured over calculating the power before the experiment, the results of the calculation give an indication that improvement in guarding the required power is needed. All statistical power results can be found in table 3.2.

#### Randomisation and Sample Ratio Mismatch

By running both the binomial and the SRM test using a threshold of  $\alpha = 0.001$  (similar to what is used within ING's EP), we identified 13 cases of SRM. Although the User Interface of ING's EP does not show any signs of flagging SRM to the experimenter, the SRM test is performed in the back-end service. ING's EP itself has flagged 16 cases of SRM. The 3 experiments which got flagged by ING's EP, but not by us turned out to be skewed thanks to the selection of a winning variant during the experiment. ING's EP supports the winning variant selection feature, however does not use this information when computing the SRM calculation. The SRM in the remaining 13 experiments is likely due to using a randomisation function which is suitable for the use on one server, however ING's EP is divided over four different server instances.

#### User group assignments

ING's EP uses the client-side assignment methodology. Although this assignment methodology supports a variety of platforms, currently only web-based applications are using their EP. Further testing was not possible due to the limitation of only 10 test users to ING's testing environment.

#### Data aggregation

ING's EP uses the existing data collection in combination with the service-based collection. It is of importance to recall that the ING's EP relies solely on the visitor counts and success/failure rates.

### 5.2.2. Pitfalls

#### Overall Evaluation Criteria

Since ING's EP relies on custom success events it is not possible to directly analyse the impacted metrics. However a word analysis on all 268 entered hypothesis reveals that the two most occurring words not in the pre-defined template are "Clicks" and "CTR" (Click Through Rate). This reveals that the pitfall of focusing on short-term goals, such as clicks, is applicable to ING.

#### Primacy and newness

By using the methodology described in paragraph 2.5.2 only 1 out of all 268 experiments flags for the primacy and newness effect. However the data of this experiment is found to be unhealthy.

#### Page speed/latency

No monitoring of variant load times is present within ING's EP. Webtrekk however does store this information, but it is considered to be unreliable.

#### Day of week effects

From the 268 experiments at ING, 207 have run for more than 0 days, making them vulnerable to the day of week effect. When performing a modulo 7 operation on the duration of these 207 experiments, 51 are found to be exposed to the day of week effect. 5 Experiments have run for more than 120 days, which is considered to be too long [38].

#### Beacon (loss)

Since problems with the beacon would result in unhealthy data, and since we have already found that SRM is present within ING's EP, potentially due to the randomisation, no further results about problems with the beacon are made. It is however notable that the beacon code snippet provided in the documentation of the EP is deprecated.

#### Browser differences

No browser information is stored within ING's EP.

#### Carry over effects

Experiments can be initiated throughout ING's software interface, limiting the possibility of carry over effects. It is however found that the 268 experiments are conducted by 76 unique squads, with the top-10 squads having numerous overlaps between their experiments. This increases the chances of having carry over effects. ING's EP has no warning system in place to help guide experimenters in avoiding these effects.

#### Robots

No mitigation for robot traffic is in place, however the influence of any robot traffic is considered to be small.

#### Staged roll-out with Simpson's Paradox

ING's EP has a staged roll-out feature, however this feature is separated from the A/B feature of the platform. Therefore Simpson's Paradox can not occur within ING's EP.

#### Device Time

By manipulating calls to the EP's API endpoint we tried to register information on ended experiment as well as for days in the past. This did not succeed, and by investigating the EP's source code we confirmed that it relies on server time.



#### Browser redirects

ING's EP does not rely on browser redirects to navigate participants to specific variants, and using them is discouraged by the platform's documentation. The EP does not keep track of Referer headers, therefore making it not possible to determine if browser redirects are used.

#### Error checks

ING's EP only supports an error check on the presence of a control variant. No further checks or warnings to experimenters are in place.

#### Unplanned differences

One experiment initiated within ING's CMS did not receive any success events for the control variant. Since it is a CMS based experiment, beacon errors are not to be expected. Therefore this experiment is considered to have an unplanned difference.

### **5.3. RQ3: How can the factors influencing the trustworthiness and soundness of A/B tests be modelled into a toolkit to help support engineers with setting up A/B tests**

By combining the influential factors identified in chapter 2 with the factors specified in the three checklists paper by Fabijan et al. [27] a list of 21 factors is created. These factors describe the reason why they invalidate the outcomes of the experiment, and what mitigation or detection could be introduced to increase awareness or prevent the problem from occurring all together. Therefore the influential factors are modelled into a set of 67 questions. From this set of questions 28 are aimed specifically at experimenters, 30 at maintainers and an additional 9 questions at both. These 9 questions aimed at both experimenters and maintainers are numerical. By requesting input from the users, they will lead to a computation that verifies if certain thresholds for trustworthiness are achieved or if scenarios are flagged. The remainder of questions is in a closed format, and are introduced for the sole purpose of creating awareness.

To easily administer the questionnaire a python command-line Open-Source tool is introduced: ABvalidator. By first determining the type of user, Experimenter or Maintainer, the correct sub-set of questions is asked. Each question topic is accompanied by a short descriptive paragraph (see figure 4.1). After all questions are answered the results are shown (see figure 4.2). For every questions that is not answered to satisfaction, a url is provided to a page where more information on the topic can be found.



# 6

## Discussion

In this chapter we discuss the main findings and discuss their implications. Furthermore we discuss the culture of A/B testing, the threads to validity of this study and recommendations for improving ING's Experimentation Platform.

### 6.1. Main Findings

By conducting this research we have identified five main findings, which are discussed in detail below.

#### 6.1.1. Factors influencing trustworthiness and soundness of A/B tests

The basic concept of A/B testing is easy to explain, and many companies report attractive outcomes. This might lead to developers, engineers and product managers perceiving A/B testing as 'low hanging fruit' in pursuing a more optimised (and profitable) software product. Our literature study however shows that there are actually many factors that influence the trustworthiness and soundness of an A/B test. The presence of these factors, or the lack of their mitigation rules, can invalidate the entire result of an experiment and even all experiments run within the Experiment Platform. Our literature study shows that the occurrence of these factors are widespread, and that these factors are found at major software organisations.

#### 6.1.2. Integration of A/B testing

The first two A's of our three A's of A/B test validation are about the availability of data and analysability of that data. In short this means that the correct data to come to a data driven decision must be present, as well as the correct tooling and methodologies to compute the results. Ensuring that both availability and analysability are on point requires commitment of the organisation wanting to conduct the experiments in their software environment. From literature and meetings with companies about A/B testing we found that organisations that do not have 'experimentation in their DNA' struggle to later fully commit to experimentation. Setting up experimentation at a software organisation requires budget and commitment by higher management.

#### 6.1.3. Lack of validation

Our study shows that ING is, to some extent, aware of the influential factors in trustworthy and sound A/B tests. By having implemented a SRM test, the most important check on unhealthy data is present. However when the test flags SRM, the result of the test is not shown to the experimenter. This means that there is no actively functioning validation within ING's EP. This shows that an in-house developed Experimentation Platform can be used more and more, without any functioning validation of the results in play. Thereby jeopardising the results all together.

#### 6.1.4. Supporting experimenters

Many of the influential factors can be mitigated by correctly instructing and training experimenters to flag certain errors. For example literature speaks about novice experimenters being less likely to spot SRM when looking at data. Therefore supporting experimenters in helping setting up correct experiments (e.g. by assisting in formulating an hypothesis, or guarding key metrics) is of use. Furthermore an Experiment Platform

should support an experimenter by showing when an influential factor is flagged within the data of the experiment.

### 6.1.5. Modelling influential factors

We have found that information on the influential factors of trustworthiness and soundness in A/B testing is currently scattered over numerous websites, research papers and industry experts. By formulating, mostly, closed questions that look for the presence of mitigation measures for the described influential factors, we are able to model the body of knowledge in an Open-Source toolkit that can be used by experimenters.

## 6.2. Implications

Our results show that a lot factors can influence the validity of an A/B test and its outcome. Although many arguments can be made for companies to develop their own in-house Experiment Platform, our study shows that there is a not to be neglected level of risk involved in doing so. Whenever an organisation decides to put its own EP into practice, validating it thoroughly is not a superfluous luxury.

The case study at ING pointed out that numerous findings question the validity of the earlier 268 conducted experiments. To ensure that the validity of future experiments is increased, work on the platform is required (described in more detail in paragraph 6.5).

## 6.3. Culture

In this report a lot has been said about CE and A/B testing. However there is an elephant in the room: culture. In the first practical Online Controlled Experiments Summit (held December 2018) companies such as Microsoft, Google, Twitter, LinkedIn, Airbnb, Facebook, Netflix, Uber, Yandex, Lyft, Amazon and Booking.com noted that creating an experiment-driven product development culture in an organisation is currently a big challenge [32]. And not only these companies and literature describe this problem. During meetings with ING, Booking.com and Bol.com culture and adoption of CE within the organisation was frequently addressed as an important topic.

It is understandable that investing a lot of time on a feature you believe in, only to have it shut down thanks to the results of an experiment, is hard. This phenomenon is referred to as the Semmelweis Reflex [32]. It takes effort for an organisation and its teams to embrace experimentation, A/B testing and data driven decisions.

## 6.4. Threads to Validity

In this study we identified nineteen factors influencing the trustworthiness and soundness of A/B tests from literature. Although these factors are found and described by other (big) software organisations the composed list is not guaranteed to be conclusive. Neither does having mitigation rules in place for these factors ensure a trustworthy and sound outcome of an A/B test.

The case study at ING identified aspects to improve in their Experimentation Platform, however this research is conducted by the author of this paper who interned at ING. Although the company and university supervisor glanced over the used code and outcomes, no in-depth verification of the results is conducted.

The ABvalidator toolkit is created in attempt to raise the awareness for future experimenters and maintainers. The questionnaire used by ABvalidator is in itself however never validated to ensure that administering it does lead to demonstrably more trustworthy and sound results. The same goes for the case study at ING. Many of the influential factors are found to exist within ING's EP, however it is not validated whether or not fixing them leads to more trustworthy and sound results.

## 6.5. Recommendations

Based on the findings of this research we have recommendations for software organisations working with their own EP in general, and ING in specific.

### 6.5.1. Generic recommendations

We recall Twyman's law from the preface of this thesis: *Any figure that looks interesting or different is usually wrong*. This encapsulates the reasoning behind our first recommendation: reproduce important findings. As seen in this study, many factors influence the possible outcome of an A/B test. Being able to reproduce the outcome of an experiment is a strong indicator of the outcome being trustworthy and sound. Furthermore we recommend to systematically run A/A tests in order to be able to spot problems with the EP as soon as possible. The third generic recommendation is to guide experimenters as much as possible within the EP (e.g. by not allowing peeking, or making sure all mitigation rules are in place), and provide structured assistance to educate engineers and developers new to experimentation. Finally we recommend to correctly train maintainers and developers of an EP, to ensure they are aware of the risks that come along with setting up experiments.

### 6.5.2. Recommendations for ING's EP

First and foremost it is of importance that the randomisation issues within ING's EP are fixed. After this the generic recommendations apply. For ING in particular combining the Webtrekk data with the EP data will enable better experiments, which are more focused on what the developers and/or content creators try to achieve with their hypothesis. As far as previous experiment results go, I would recommend to look forward and not jeopardise the gained trust of ING's engineers into the platform. Perhaps a new master student could try to validate the old experiments in hindsight by combining the Webtrekk data with the data from the EP.



# 7

## Conclusion and Future Work

Continuous Experimentation, and A/B testing in particular, helps software organisations to make data driven decisions when altering their (software) products. Many examples can be found of experiments which revealed that making a slight adjustment to the software product can lead to immense income gains. However it is also found that there are many factors influencing the trustworthiness and soundness of these outcomes, possibly resulting losing revenue when increments where to be expected.

In this final chapter of this thesis we revisit the research questions, come to conclusions and provide pointers for future work.

### 7.1. Conclusion

In this thesis our three main research questions have been answered through combining a literature survey with data from practice:

#### **RQ1: What factors affect the trustworthiness and soundness of an A/B test?**

Nineteen factors are identified in literature that influence the trustworthiness and soundness of an A/B test. These factors are divided over two categories: essentials and pitfalls. Essentials are factors that are considered the bare minimum to run a valid A/B test. Factors within this category are: systematically running A/A tests, accepting or rejecting the null hypothesis by using the correct methods for determining significance between variants, in advance determining the required number of participants of an experiment, sound randomisation of participants between variants and consistent group assignment. Pitfalls on the other hand are influential factors which can occur, even when the essentials are correctly implemented. Factors within this category are: choosing suitable Overall Evaluation Criteria, primacy and newness effects, differences in page speeds (latency), day of week effects, errors with the beacon, differences in browsers, carry over effects, the presence of robots, Simpson's Paradox when using a staged roll-out, differences in device times, problems with browser redirects, early on error detection, and unplanned differences between variants.

#### **RQ2: To what extent is ING's EP impacted by the factors influencing the trustworthiness and soundness of an A/B test?**

By using the data from 268 A/B test experiments conducted between January 2016 and July 1st 2019, we could identify that ING's EP is, to some extent, impacted by 4 out of 6 essential factors. The remaining two essential factors (user group assignments and data aggregation) could not have been tested due to limitations in the test environment.

From the pitfalls, three factors have been identified to occur within ING's EP (Overall Evaluation Criteria, day of week effects and unplanned differences). Three factors do not occur, are correctly mitigated or can not occur due to the set-up of the EP (Primacy and newness, staged roll-out with Simpson's Paradox and device time). For the remaining seven factors no definitive conclusion can be made, either because the correct data is not present, or due to the test environment limitations.

#### **RQ3: How can the factors influencing the trustworthiness and soundness of A/B tests be modelled into a toolkit to help support engineers with setting up A/B tests?**

In order to achieve the goal of making the factors influencing the trustworthiness and soundness of A/B tests easy accessible to developers and engineers, we have decided to model these factors in a set of 67 questions. To make the questions more relevant to the users' perspective they are divided over two type of users: experimenters (28 closed questions) and maintainers (30 closed questions). The remaining 9 questions are aimed at both experimenters and maintainers, and are numerical input questions used to validate if the computed results are accurate. To help structure the questionnaire we introduced the three A's of A/B test validation: Availability, Analysability and Accuracy. The 67 questions are grouped by these categories. For easy administering of the questions we introduce ABvalidator, an Open Source Python command line toolkit.

## **7.2. Future work**

The results of this study not only provides future work for the maintainers of ING's Experiment Platform, but also for future research.

### **7.2.1. Validation of ABvalidator**

As pointed out in this research we were able to identify many influential factors, and make them accessible through ABvalidator. Although our case study shows that ABvalidator is able to identify these factors in an operational Experiment Platform, there is still research needed if mitigating all these factors really does lead to more trustworthy and sound A/B tests.

### **7.2.2. More test users**

During our case study at ING, we were not able to fully test all of the influential factors due to limitations in the test environment. Having ample of test users available would make checking those factors worthwhile.



# Bibliography

- [1] ING Bank annual report 2017. URL <https://www.ing.com/web/file?uuid=85dd95ab-b69f-43da-9d04-4e2d54e53c6a&owner=b03bc017-e0db-4b5d-abbf-003b12934429&contentid=42780>. (Date last accessed 27-February-2019).
- [2] Phased release for automatic updates now available. URL <https://itunespartner.apple.com/en/apps/news/31070842>. (Date last accessed 25-February-2019).
- [3] Knight capital glitch loss hits \$461m. URL <https://www.ft.com/content/928a1528-1859-11e2-80e9-00144feabdc0>. (Date last accessed 21 March 2019).
- [4] John G Adair. The hawthorne effect: a reconsideration of the methodological artifact. *Journal of applied psychology*, 69(2):334, 1984.
- [5] Deepika Badampudi, Claes Wohlin, and Kai Petersen. Experiences from using snowballing and database searches in systematic literature studies. In *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering*, page 17. ACM, 2015.
- [6] Titus Barik, Robert DeLine, Steven Drucker, and Danyel Fisher. The bones of the system: a case study of logging and telemetry at microsoft. In *2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)*, pages 92–101. IEEE, 2016.
- [7] Dennis D Boos and Jacqueline M Hughes-Oliver. How large does n have to be for z and t intervals? *The American Statistician*, 54(2):121–128, 2000.
- [8] George EP Box, J Stuart Hunter, and William G Hunter. Statistics for experimenters. In *Wiley Series in Probability and Statistics*. Wiley Hoboken, NJ, 2005.
- [9] Robert Chang. Detecting and avoiding bucket imbalance in a/b tests. URL [https://blog.twitter.com/engineering/en\\_us/a/2015/detecting-and-avoiding-bucket-imbalance-in-ab-tests.html](https://blog.twitter.com/engineering/en_us/a/2015/detecting-and-avoiding-bucket-imbalance-in-ab-tests.html). (Date last accessed 8-July-2019).
- [10] Nanyu Chen, Min Liu, and Ya Xu. How a/b tests could go wrong: Automatic diagnosis of invalid on-line experiments. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 501–509. ACM, 2019.
- [11] Norman Cliff. Dominance statistics: Ordinal analyses to answer ordinal questions. *Psychological bulletin*, 114(3):494, 1993.
- [12] John W Creswell and J David Creswell. *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications, 2017.
- [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*, pages 1105–1114. ACM, 2009.
- [14] William H DeLone and Ephraim R McLean. Information systems success: The quest for the dependent variable. *Information systems research*, 3(1):60–95, 1992.
- [15] Alex Deng and Xiaolin Shi. Data-driven metric development for online controlled experiments: Seven lessons learned. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 77–86. ACM, 2016.
- [16] Alex Deng, Ya Xu, Ron Kohavi, and Toby Walker. Improving the sensitivity of online controlled experiments by utilizing pre-experiment data. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 123–132. ACM, 2013.

- [17] Alex Deng, Jiannan Lu, and Shouyuan Chen. Continuous monitoring of a/b tests without pain: Optional stopping in bayesian testing. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 243–252. IEEE, 2016.
- [18] Kevin C Desouza. *Intrapreneurship: managing ideas within your organization*. University of Toronto Press, 2011.
- [19] Pavel Dmitriev and Xian Wu. Measuring metrics. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, pages 429–437. ACM, 2016.
- [20] Pavel Dmitriev, Brian Frasca, Somit Gupta, Ron Kohavi, and Garnet Vaz. Pitfalls of long-term online controlled experiments. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 1367–1376. IEEE, 2016.
- [21] Pavel Dmitriev, Somit Gupta, Dong Woo Kim, and Garnet Vaz. A dirty dozen: twelve common metric interpretation pitfalls in online controlled experiments. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1427–1436. ACM, 2017.
- [22] Anthony WF Edwards. Ra fischer, statistical methods for research workers, (1925). In *Landmark Writings in Western Mathematics 1640-1940*, pages 856–870. Elsevier, 2005.
- [23] Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. CRC press, 1994.
- [24] Aleksander Fabijan, Pavel Dmitriev, Helena Holmström Olsson, and Jan Bosch. The evolution of continuous experimentation in software product development: from data to a data-driven organization at scale. In *Proceedings of the 39th International Conference on Software Engineering*, pages 770–780. IEEE Press, 2017.
- [25] 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, 2018.
- [26] 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)*, pages 68–72. IEEE, 2018.
- [27] Aleksander Fabijan, Pavel Dmitriev, Helena Holmström Olsson, Jan Bosch, Lukas Vermeer, and Dylan Lewis. Three key checklists and remedies for trustworthy analysis of online controlled experiments at scale. In *Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice*, pages 1–10. IEEE Press, 2019.
- [28] Fabian Fagerholm, Alejandro Sanchez Guinea, Hanna Mäenpää, and Jürgen Münch. Building blocks for continuous experimentation. In *Proceedings of the 1st international workshop on rapid continuous software engineering*, pages 26–35. ACM, 2014.
- [29] Steven Goodman. A dirty dozen: twelve p-value misconceptions. In *Seminars in hematology*, volume 45, pages 135–140. Elsevier, 2008.
- [30] John Graham-Cumming. Cloudflare outage caused by bad software deploy (updated). URL <https://blog.cloudflare.com/cloudflare-outage/>. (Date last accessed 3-July-2019).
- [31] 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)*, pages 1–109. IEEE, 2018.
- [32] Somit Gupta, Ronny Kohavi, Diane Tang, and Ya Xu. Top challenges from the first practical online controlled experiments summit. *ACM SIGKDD Explorations Newsletter*, 21(1):20–35, 2019.
- [33] Mark G Haviland. Yates’s correction for continuity and the analysis of  $2 \times 2$  contingency tables. *Statistics in medicine*, 9(4):363–367, 1990.

- [34] Henning Hohnhold, Deirdre O'Brien, and Diane Tang. Focusing on the long-term: It's good for users and business. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1849–1858. ACM, 2015.
- [35] Kenneth E Iverson. A programming language. In *Proceedings of the May 1-3, 1962, spring joint computer conference*, pages 345–351. ACM, 1962.
- [36] Raphael Lopez Kaufman, Jegar Pitchforth, and Lukas Vermeer. Democratizing online controlled experiments at booking. com. *arXiv preprint arXiv:1710.08217*, 2017.
- [37] Ron Kohavi and Roger Longbotham. Unexpected results in online controlled experiments. *ACM SIGKDD Explorations Newsletter*, 12(2):31–35, 2011.
- [38] Ron Kohavi, Randal M Henne, and Dan Sommerfield. Practical guide to controlled experiments on the web: listen to your customers not to the hippo. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 959–967. ACM, 2007.
- [39] 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, 2009.
- [40] Ron Kohavi, David Messner, Seth Eliot, Juan Lavista Ferres, Randy Henne, Vignesh Kannappan, and Justin Wang. Tracking users' clicks and submits: Tradeoffs between user experience and data loss. *Redmond: sn*, 2010.
- [41] Ron Kohavi, Alex Deng, Brian Frasca, Roger Longbotham, Toby Walker, and Ya Xu. Trustworthy online controlled experiments: Five puzzling outcomes explained. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 786–794. ACM, 2012.
- [42] Ron Kohavi, Alex Deng, Roger Longbotham, and Ya Xu. Seven rules of thumb for web site experimenters. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1857–1866. ACM, 2014.
- [43] Ronny Kohavi. Srm calculation excel file. URL <http://bit.ly/srmCheck>. (Date last accessed 8-July-2019).
- [44] Greg Linden. Make data useful, 2006.
- [45] Eveliina Lindgren and Jürgen Münch. Software development as an experiment system: a qualitative survey on the state of the practice. In *International Conference on Agile Software Development*, pages 117–128. Springer, 2015.
- [46] Eveliina Lindgren and Jürgen Münch. Raising the odds of success: the current state of experimentation in product development. *Information and Software Technology*, 77:80–91, 2016.
- [47] Widad Machmouchi and Georg Buscher. Principles for the design of online a/b metrics. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 589–590. ACM, 2016.
- [48] Herzl Marouni. Design of experiments using the taguchi approach. *Quality Progress*, 34(9):111, 2001.
- [49] Guy Paré, Marie-Claude Trudel, Mirou Jaana, and Spyros Kitsiou. Synthesizing information systems knowledge: A typology of literature reviews. *Information & Management*, 52(2):183–199, 2015.
- [50] Stephen K Park and Keith W Miller. Random number generators: good ones are hard to find. *Communications of the ACM*, 31(10):1192–1202, 1988.
- [51] Karl Pearson. X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302): 157–175, 1900.
- [52] Eric T Peterson. *Web analytics demystified: a marketer's guide to understanding how your web site affects your business*. Ingram, 2004.

- [53] Ranjit K Roy. *Design of experiments using the Taguchi approach: 16 steps to product and process improvement*. John Wiley & Sons, 2001.
- [54] Gerald Schermann, Jürgen Cito, Philipp Leitner, Uwe Zdun, and Harald C Gall. We're doing it live: A multi-method empirical study on continuous experimentation. *Information and Software Technology*, 99:41–57, 2018.
- [55] Eric Schurman and Jake Brutlag. Performance related changes and their user impact. In *velocity web performance and operations conference*, 2009.
- [56] Eric Schurman and Jake Brutlag. The user and business impact of server delays, additional bytes, and http chunking in web search. In *Velocity Web Performance and Operations Conference*, 2009.
- [57] Samuel Sanford Shapiro and Martin B Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611, 1965.
- [58] Edward H Simpson. The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 13(2):238–241, 1951.
- [59] S Sounders. High performance web sites: Essential knowledge for front-end engineers, 2007.
- [60] Diane Tang, Ashish Agarwal, Deirdre O'Brien, and Mike Meyer. Overlapping experiment infrastructure: More, better, faster experimentation. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 17–26. ACM, 2010.
- [61] Gerald Van Belle. *Statistical rules of thumb*, volume 699. John Wiley & Sons, 2011.
- [62] Lukas Vermeer. Moving fast, breaking things, and fixing them as quickly as possible. URL <https://medium.com/booking-com-development/moving-fast-breaking-things-and-fixing-them-as-quickly-as-possible-a6c16c5a1185>. (Date last accessed 19-June-2019).
- [63] Donna M Windish, Stephen J Huot, and Michael L Green. Medicine residents' understanding of the biostatistics and results in the medical literature. *Jama*, 298(9):1010–1022, 2007.
- [64] Ding Yuan, Soyeon Park, and Yuanyuan Zhou. Characterizing logging practices in open-source software. In *Proceedings of the 34th International Conference on Software Engineering*, pages 102–112. IEEE Press, 2012.
- [65] Zhenyu Zhao, Miao Chen, Don Matheson, and Maria Stone. Online experimentation diagnosis and troubleshooting beyond aa validation. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 498–507. IEEE, 2016.