KPI by proxy

MSc. Thesis Computer Science

Martijn van den Hoek

KPI by proxy

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

Martijn van den Hoek
born in Haarlem, the Netherlands

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

AI for Fintech Research
ING Bank N.V.
Frankemaheerd 1
Amsterdam, the Netherlands
www.ing.nl

KPI by proxy

Author: Martijn van den Hoek
Student id: 4600487

Abstract

Metrics are widely used in the software engineering industry and can serve as Key Performance Indicators (KPIs), which are used by management to make informed decisions and understand the performance of the organisation. Many companies measure themselves against industry-standard metrics, in addition to their own set of metrics. This thesis aims to investigate the relationship between these industry-standard metrics and the metrics that are additionally collected. Instead of focusing on the performance of a single organisation, the DORA report [16] focuses on the comparison of organisations. It measures the Software Delivery and Operational (SDO) performance of organisations by four key and industry-standard metrics representing two aspects, stability and throughput, of a software product. The use of one metric as a proxy for another metric or a thematic group of metrics is a common phenomenon. However, there rarely is evidence supporting the assumption that the proxy reflects the intended metric or represents the full thematic group. This thesis performs a single case study within ING, a large and highly digital bank in the Netherlands. It investigates the KPIs that are collected by the bank and analyses the relationships between those KPIs and the four metrics from the DORA report. It establishes a list of 27 KPIs that are in use by ING and shows that there are no correlations between the DORA metrics as collected within the bank and that these metrics show very little correlation with the metrics that ING collects additionally. Furthermore, it is established that nearly all metrics contain a bias at the organisational level and that these biases have a significant impact on the correlation between metrics.

Thesis Committee:

Chair:	Prof. Dr. A. van Deursen, Faculty EEMCS, TU Delft
University supervisor:	Dr. S. Proksch, Faculty EEMCS, TU Delft
Company supervisor:	E. Kula, Lab manager AI for Fintech Research lab, ING
Committee Member:	Dr. L.C. Siebert, Faculty EEMCS, TU Delft

Preface

Firstly, I would like to thank my academic supervisor Sebastian Proksch for his support and feedback during this thesis. Thank you so much for your help, guidance and thoughtful discussions over the last few months.

I would also like to thank Elvan Kula for giving me the opportunity to complete my thesis within ING, for her feedback on the thesis and for helping me understand and navigate the organisation. I would also like to thank Jerry Brons for his feedback, interesting discussions and help during the development of this thesis.

This thesis would not have been possible without the help of several ING colleagues. I would like to thank everybody who has participated in the interviews, provided me with additional data or has given me the opportunity to collect metrics from their systems. During my time at ING, I've also had the opportunity to work together with an amazing group of people at the AI for Fintech lab. I would like to thank all of you for your interesting talks, support and feedback.

Last but certainly not least, I would like to thank all my friends and family who have helped and supported me, not only during this thesis but during all of my studies. Thank you all so much for always being there for me, especially during the rough (and distanced) state of the world over the last 18 months.

This endeavour would not have been the same without the help and support of everybody mentioned above, whom I would like to thank again for everything they have done for me over the last months.

Martijn van den Hoek
Delft, the Netherlands
August 19, 2021

Contents

Chapter 1

Introduction

The practice of *DevOps* aims to remove the barrier between development (Dev) and operations (Ops). In the old days, a new release was developed and ”thrown over the wall of confusion” to be operated. This expression stems from the different objectives of developers and operations personnel: where the developers want to release new features as soon as possible, operations aims to achieve as much stability as possible. The practice of DevOps breaks down this wall by integrating the two disciplines, resulting in both higher throughput and stability [37].

Metrics are widely used in the software engineering industry. Areas on which metrics exist include, but are not limited to, software quality [33], the continuous delivery process [25] and agile processes [31]. Since DevOps combines the development and operations disciplines, the metrics used in this approach to software engineering come from different areas. Metrics can serve as *Key Performance Indicators* (KPIs) and are used by management to make informed decisions and understand the performance of the organisation [6]. This thesis defines KPIs as a subset of metrics that are found to be important by those who use them.

Despite this plethora of described metrics, using all of them at the same time is not an option. Implementing metrics takes effort and time and might require dedicated teams for building, operating and maintaining the monitoring tools or platforms. Secondly, measuring too many KPIs can result in information overload and make it harder for management to recognise which information is relevant for them [6]. Therefore, it is important to be able to make informed decisions about which metrics to collect.

Instead of focusing on the performance of a single organisation, the DORA report [16] focuses on the comparison of organisations. It measures the *Software Delivery and Operational* (SDO) performance of organisations by four key metrics: *Lead time for changes*, *deployment frequency*, *change fail rate* and *time to restore service*. Those four metrics relate to the three stages of the software life cycle: development, deployment and operation. It surveyed almost $1,000$ employees from all over the world and from different industries how their teams perform on these metrics and about their practices.

This thesis is a case study within ING, a large bank in the Netherlands. With 57.000 employees serving around 38.9 million customers in more than 40 countries, 1.005 million in profits and 90 per cent of primary customer interactions being digital [23], this organisation

is highly digital.

One of the objectives of this thesis is to understand how the DORA metrics behave within ING, given the description of how they behave between organisations as described by Forsgren et al. [16]. As an organisation, it is important to understand the metrics that are being measured and how they relate to each other, as things that are not informative should not be measured.

Within ING, many metrics are being measured currently and many more have been measured in the past. This thesis aims to understand how the DORA metrics relate to the ING metrics, being motivated from both the academic and corporate points of view. For the former, understanding how the four DORA metrics relate to metrics used in industry can provide valuable insights into how these industry-standard metrics relate to other metrics. For the latter, it is important to know how the organisation relates to this industry standard as measuring industry-standard metrics such as the DORA metrics without considering the context and needs of the organisation can lead to failure [6].

In industry, it is not always possible to measure the metric one would like to have access to. In such a scenario, it is possible to measure a metric that is believed to be related to the intended metric and use this related metric as a *proxy*. Another objective of this thesis is to analyse the use of proxies within ING and to investigate how the chosen proxies relate to the metrics they approximate. This is possible since some of the proxies are collected on an organisational scale, while the original metric is collected for only a subset of teams. Providing insights into the applicability of proxies can help businesses to make better-informed decisions around the use of proxies instead of the intended metrics: is it worth the effort to invest in measuring the intended metrics, or will measuring a proxy that is easier to calculate also suffice?

In short, this thesis aims to understand how the DORA metrics relate to each other within ING and to the metrics that are additionally collected by ING and to understand the usage and applicability of proxies for several metrics. This overarching goal resulted in the five research questions described below. The first two questions are exploratory in nature and are intended to get an understanding of DevOps metrics in literature and the usage of KPIs within ING. The third, fourth and fifth questions answer the overarching goal of this thesis.

RQ1 What DevOps metrics exist in the literature?

Before the case of ING is considered, this structured literature study creates an overview of the DevOps metrics that are reported in the literature. The corresponding chapter lays the foundation for the interviews that are used to answer RQ2.

RQ2 What KPIs are used by ING and why?

After the literature review has been completed, this research question aims to understand the KPIs that are used within ING and why they have been chosen. It generates insights into the process of selecting KPIs and provides the knowledge required for the remaining research questions. This question focuses specifically on KPIs instead

of on metrics, in contrast to the other research questions. It serves to get an understanding of what is most important to ING, while the other research questions intend to gain insight into the relationship between metrics within the organisation. As KPIs are a subset of metrics, those will be analysed in other research questions as well.

RQ3 How do the four metrics as identified by DORA [16] relate to the other metrics used by ING?

This is the first research question that can be brought back to the overarching goal of this thesis. It investigates the relationships between the ING-specific metrics and between the DORA metrics and the ING-specific metrics. To get a better understanding of the relationships between the metrics, this question does not only focus on the metrics themselves but also investigates the derivatives of the metrics.

RQ4 What is the influence of organisational structure on the relationships between the DORA and ING-specific metrics?

The previous research question has shown how the metrics relate to each other in absolute terms and how they change in relation to each other over time. This research question aims to understand if parts of these relationships can be explained by biases introduced via the organisational structure. In other words, does the use of these metrics differ per organisational unit, or are they used uniformly throughout ING? In the former case, a distinction will be made between relations that are the result of the metrics themselves and relations that are caused by biases.

RQ5 What proxies are used by ING and how do they relate to the metric they substitute?

Previous research questions have investigated the relationships between the available metrics. This research question uses the interviews of the second research question to understand which proxies are in use or have been in use in the past by ING. For each of the used proxies, this research question aims to generate insights into how the proxy relates to the actual metric it replaces when both the proxy and the actual metric are available.

While answering these research questions, this thesis has generated three main contributions. The first contribution provides an overview of the state of the art of DevOps metrics in scientific literature. The second and third contributions relate to the context of this thesis. To the best of our knowledge, this is the first thesis that investigates the relationship between metrics that are used in the highly regulated financial industry. The three main contributions have been listed below.

- **Overview of DevOps metrics from literature** This thesis has composed a list of 66 metrics that have been extracted from 31 pieces of scientific literature. This literature review provides insights into the state of the art of metrics in literature.

- **Exploratory interviews in a regulated context** This thesis performed exploratory interviews within a large and highly digital bank. The interviews have generated insights into the collection and usage of metrics within the regulated context of the financial industry
- **Thorough analysis of relationships between metrics used within the bank** Finally, this thesis has performed a thorough analysis of the relationships between the metrics that are in use by the bank. These metrics consist of both the industry-standard DORA [16] metrics and additional metrics that ING collects.

Chapter 2

Literature study of related work

This thesis aims to understand how the four DORA [16] metrics relate to the metrics used by ING, how the ING metrics relate to each other and how the proxies used within ING reflect the metrics they replace. As a first step in generating this understanding, this section creates an overview of the DevOps-related metrics that have been described in scientific literature. This investigation also aims to understand previous research that is related to this thesis. More precisely, it aims to answer the following research question:

Specification of research questions

RQ1 What DevOps metrics exist in the literature?

Having this understanding is important, as it serves as a basis for understanding the metrics that will be discussed in later chapters. A structured literature review has been performed to answer this research question. The remainder of this chapter will first discuss the methodology used to perform the literature review, followed by a discussion of the results of the review.

2.1 Methodology

Protocol 2.1: Selection criteria for literature items

- Item must be written in English.
- Item must be a paper, PhD thesis or master thesis.
- Item must do one of three things:
  - Introduce a new metric.
  - Introduce a new way of computing a metric.
  - Introduce a new insight into a metric or the relationship between metrics

Protocol 2.2: Definition of relevance for encountered metrics

- The metric relates to the development, deployment or operation of software.
- The metric is objective and does not involve human opinion.
- The metric can be collected automatically.
- The metric is not specific to a single build or pull request. This thesis focuses on DevOps as a whole and omits to look at specific builds or pull requests.

This section discusses the methodology applied during the structured literature review. All literature was retrieved from Google Scholar and for every query, the first four pages were evaluated for relevant papers. The pieces of literature were evaluated based on the selection criteria that have been listed in Protocol 2.1 and individual metrics were evaluated for their relevance by the definitions listed in Protocol 2.2. Literature had to satisfy all criteria to be evaluated. Books were excluded based on the assumption that all metrics available in books would also be available in other literature. The metrics found in literature surveys were only added to the results of this literature review if they were relevant to the thesis. This decision was made to limit the number of metrics collected during the review, as surveys had the potential to add very large numbers of non-relevant metrics.

The initial set of literature consisted of the *State of DevOps* report by Forsgren et al. [16] and the book about the same topic [15]. From this initial set of literature, several search queries were derived. The different queries were used to capture metrics from both DevOps, *continuous integration* (CI) and *continuous delivery* (CD). For the interest of time, multiple specific queries were used instead of a single expression to capture all papers. A list of the used queries has been depicted below.

Used queries

- DevOps metrics
- DevOps metrics Quality Model
- DevOps metrics literature review
- DevOps MTTR literature review
- DevOps metrics approximate
- Metrics Continuous Delivery
- Continuous Delivery metrics Quality Model
- Continuous Delivery metrics literature review
- Continuous Delivery MTTR literature review
- Continuous Delivery metrics approximate
- Continuous integration open source measure

Papers were only evaluated as part of the first query from which they resulted. Papers were first selected based on their title. If the title was in line with the used query, their abstract was read. For papers containing metrics, the section containing the metrics was scanned to see if they were relevant as defined by Protocol 2.2. All papers that satisfied those requirements were part of the initial set used for the snowball process. When all

papers had been collected, a one-level snowball process was performed both forward and backward.

Protocol 2.3: Processing of collected metrics

- Entries that were not relevant as defined by Protocol 2.2 were removed.
- Entries that were (near) duplicates were merged.
- When one metric was a subset of another metric, they were merged (e.g. ”Cyclomatic Complexity” and ”complexity metrics” became ”Cyclomatic Complexity”).
- When one metric was the time-sensitive variant of another metric, they were merged (e.g. ”Defects in production” and ”Defects in production over a certain timeframe” became ”Defects in production over a certain timeframe”).

After all papers were selected, they were read to identify the metrics that were mentioned in them. Reading a paper consisted of reading their results section and scanning other relevant sections such as the methodology for information about metrics. Some papers were removed from the set of collected papers, as the reading process revealed that they did not contain relevant metrics. When reading, all metrics mentioned in them were extracted. After all papers were read, the extracted metrics were evaluated and processed as described inProtocol 2.3.

The resulting metrics were then clustered into categories. The clustering process aimed to group metrics that were related. This way, multiple views on the same aspect of software were obtained.

2.2 Results

This section describes the results of the literature review. The first subsection provides details on the found literature and the used queries. The second subsection provides a summary of the evaluated literature. The third subsection provides the extracted metrics and their clustering.

2.2.1 Resulting literature

The literature review resulted in 31 pieces of relevant literature. Table 2.1 provides an overview of the number of resulting literature from each of the queries as a result of the queries directly, after snowballing and after reading the literature. The queries resulted in 22 pieces of literature directly. After snowballing, 48 pieces of literature were selected for further reading. After reading the results section of all papers, 31 pieces of literature were left and thus finally selected to be part of this literature review.

2.2.2 Summaries of evaluated literature

Bezemer et al. [4] evaluated how industry addresses performance within DevOps products. To perform this evaluation, they surveyed participants from different sectors of the industry

Query	Nr. items	After snowballing	After selection
Devops metrics	3	9	6
Devops metrics Quality Model	1	4	2
Devops metrics literature review	1	1	1
Devops MTTR literature review	3	4	4
Devops metrics approximate	1	3	2
Metrics Continuous Delivery	3	8	4
Continuous Delivery metrics Quality Model	0	0	0
Continuous Delivery metrics literature review	1	1	0
Continuous Delivery MTTR literature review	0	0	0
Continuous Delivery metrics approximate	2	2	1
Continuous integration open source measure	7	16	11
		Initially selected	**Selected**
	22	48	31

Table 2.1: Overview of the resulting literature. The first column contains the number of items resulting from each query. The second column contains the number of items after snowballing all items from each query. The third column contains the number of items left after reading the literature.

and got 26 full responses. They found that monitoring mainly focused on system-level metrics and that only one-third of respondents regularly performed performance evaluations.

Callanan and Spillane [7] described a case study of the transition towards DevOps of an Australian company. The goal of the company was to reduce the cycle time from weeks to less than a day. As a consequence of the reduced *cycle time*, the number of *changes* that were included in each new version was reduced. Due to this reduction, it was easier to detect defects and hotfixes could be deployed much faster.

Ebert et al. [11] described the DevOps tools and technologies used in 2016. Furthermore, they presented a case study of a company that transitioned to DevOps. As a result of the transition, the cycle time was reduced significantly. The case study reports an improved return on investment and more consistent software releases.

Forsgren and Humble [14] used the Economic Order Quantity (EOQ) model to describe performance in the context of DevOps. They identified two aspects of performance in the context of DevOps: *throughput* and *stability*. They used the lead time and deployment frequency measures as metrics for the throughput and the mean time to recover (MTTR) as a metric for stability. They applied this model to a survey of 7,522 IT professionals. They found that throughput and stability measures were correlated and identified three groups among participants: high, medium and low performers. Low performers tended to have low deployment frequency, large lead times and large MTTR. For the high performers, those results were reversed. They also noted that ”Short TTR reflects the ability of the organisation to achieve high levels of service stability.”[14].

The master thesis of Jain and Aduri [25] aimed to collect continuous delivery metrics from literature and compared them against the metrics used in industry to come up with

a checklist of metrics. They first performed a literature study. Next, they evaluated the usefulness of the collected metrics using an online questionnaire. From this questionnaire, they identified the limitations of the proposed metrics and found other metrics that are used in industry. Using this information, they compiled a checklist of metrics. Lastly, they performed interviews to validate the usefulness of the checklist. Only the metrics that were obtained from industry have been collected for this thesis, as Jain and Aduri [25] only ranked metrics that were obtained from Lehtonen et al. [34] in terms of their usability and this paper will be discussed later in this thesis.

The master thesis of Klint and Åkerström [28] studied the challenges of continuous delivery, the practices used to overcome those challenges and the metrics that can be used to monitor a continuous delivery pipeline. They performed a literature survey and found several metrics. As those metrics were obtained from other works referenced in this section, they have been omitted. They also interviewed developers at a company and got several metrics that the developers found useful. Those metrics have been included in the results of this literature review.

Lwakatare et al. [37] also presented a multivocal literature review. Besides scientific and multivocal literature, this review also contained data obtained from practitioners who took part in a workshop related to DevOps. The goal of the review was to compare DevOps to *agile*, *lean* and continuous deployment. The review also discussed the claimed effects of DevOps and the metrics that were used to support those claims. The authors claimed that "Effects of DevOps include the ability to release software quickly, frequently and with improved quality. However, the use of popular metrics such as deployment rate and cycle/lead time is insufficient to determine whether these effects arise from the implementation of DevOps or other approaches."[37] but that "While the supporting empirical evidence was poor, many sources (especially ML) stood over many of the presented claims. The lack of empirical evidence suggests that DevOps is still in its infancy. Additionally, most discussion of DevOps is confined to informal talks, workshops and events in forums outside more formal dissemination channels."[37]. This last claim requires to note that the paper was written in 2016.

The master thesis by Wu and Zhang [52] aimed to find proxies for the maintainability of software in the form of code metrics. They selected the MTTR as a metric for maintainability and performed a literature review to find code metrics that were related to maintainability. They selected the 10 most common metrics and established the correlation between each of them and the MTTR of a popular open-source software (OSS) project. They found four metrics that were most useful as a proxy for the MTTR.

The thesis by Arvedahl and Åkersten [2] investigated the goals when adopting DevOps, which practices within DevOps are critical for achieving those goals and the impact of adopting those practices. In their thesis, they performed a literature survey and a case study at multiple software organisations. For the case study, 12 practitioners with different experience levels were interviewed and surveyed. To measure the effect of adopting DevOps, they proposed several metrics extracted from the case studies.

The PhD thesis of Cogo [9] tried to understand DevOps and characterise it. Furthermore, it investigated how DevOps affects the outcomes of IT companies. They adopt metrics from another work, to which they add additional information. Using those metrics, they

measure the impact that DevOps has on organisations.

Farroha and Farroha [12] proposed a framework for transitioning the American Department of Defense to a DevOps approach to software development. Besides proposing several metrics that have to be monitored, they recommend placing more focus on the mean time to repair (MTTR) than on the mean time between failures (MTBF) for most failure types [12].

Ghaleb et al. [20] evaluated CI builds and tried to find factors that correlate with long build times. They study 104,442 CI builds from 67 GitHub projects. Besides the generally accepted reasons for longer builds, they find that builds lasting longer than 10 minutes had the following three characteristics: they were rerunning failed jobs, not using the cache and not finishing as soon as all jobs were finished. Among other things, they found that team size, test cases/KLOC and SLOC had a significant (p-value $< 2.2e^{-16}$) relation with the build time.

Gousios et al. [22] investigated pull-based development in OSS projects hosted on Github. By analysing 166,884 pull requests (PRs) from 291 projects, they investigated how often pull-based development is used, what the life-cycle of a pull request looks like, why some pull requests are not merged, which factors influence the decisions of core developers to merge a pull request and how long it takes them to make this decision. Using several metrics, they found that the test coverage and the size of the project did not correlate with the time it takes to merge a pull request, that the amount of source code in the project was one of three factors that could be used to predict if a pull request was going to be merged and that both the size of the project and its test coverage had a significant effect on the time it took to merge a pull request.

Islam and Zibran [24] aimed to find the factors that are related to build failures when using CI. To this end, they studied 3.6 million builds from 1,090 open-source projects. They only selected projects that had used CI for more than a year and that had performed at least 100 builds using the CI. They analysed metrics related to multiple dimensions of software development. This thesis only uses the project level metrics. They found that the sizes of projects and teams did not have a significant ($\alpha = 0.05$) correlation with the results of builds. Even though those metrics did not have a significant correlation with the outcome of a build, Ghaleb et al. [20] concluded that they had a significant correlation with the time it took to complete a build.

Jain et al. [26] investigated the effect of team size on the number of build failures and the effect of build failures on the productivity of developers. Only the first research question is relevant to this thesis, as it does not go into the productivity of developers. By analysing 3,702,595 builds of Java and Ruby projects performed using Travis CI, they found that there was an optimum team size which led to the lowest ratio of build failures over the total amount of builds.

Kerzazi et al. [27] analysed 3,214 builds over a six-month time frame within a large company. They also interviewed 28 software engineers from the company. They investigated the impact a build failure had on a project, the typical circumstances under which a build failure occurred and the factors that were associated with build failures. Concerning this last research question, they found that several factors were associated with more frequent build failures, but only the number of contributors per branch is suitable to be used in this thesis.

The technical report of König and Steffens [32] aimed to develop a quality model for DevOps. They collected several metrics from the literature and incorporated them into their quality model.

Kupiainen et al. [31] presented a literature review on papers containing case studies. The literature review focused on why metrics were used in agile processes and how they were used. The explanations mainly focus on why a certain category of metrics was used instead of on specific metrics.

Lee [33] provided an overview of the metrics used in software quality. Their list of metrics contains the formulas for the metrics. Although the metrics mainly focus on software quality, this paper includes useful metrics and formulas.

Lehtonen et al. [34] presented a case study on a single project within a company. They aimed to identify the data that could be collected from the used CD pipeline, which additional data should be collected and which new metrics could be introduced based on this data. They proposed several metrics that were enabled by the use of the implemented branching model and another metric that required additional code in the product to be tracked.

Lohrasbinasab et al. [36] performed a multivocal literature review on BizDevOps, a branch of DevOps that also integrates the business stakeholders into the DevOps cycle. As the literature review was multivocal, it also included sources that are not scientific literature. The review contains a list of KPIs that are often used in BizDevOps.

The master thesis by Maddila [38] is composed of a structured literature review and a survey among practitioners. It aimed to find agile and lean metrics that were mentioned in literature and were used in practice. They evaluated the purpose of using the metric and if the respondents are satisfied with the metric. If respondents were not satisfied, they investigated the reason for this. They first performed a structured literature review. From this review, they obtained a list of metrics that were put in the survey for practitioners. Furthermore, practitioners could add additional metrics they used. Only the metrics with which the respondents were satisfied were included in the results of this thesis.

Ordonez and Haddad [40] argued why it is important to collect metrics from software systems and discussed the metrics used in four large companies and the way they were used.

Prates et al. [43] performed a multivocal literature review on the metrics used in DevSecOps, the integration between security teams and DevOps. They found that the topic of DevSecOps was getting adopted by industry, but that scientific research was lacking behind. They found 11 grey literature articles and only two scientific articles containing metrics.

Rahman et al. [44] investigated the effects of adopting CI on software development and the difference of those effects between open source projects and proprietary projects. They proposed several metrics to measure those effects. They analysed 150 open-source and 123 proprietary projects before and after adopting CI. With $\alpha = 0.05$, they found that adopting CI in OSS led to higher normalised commit frequency and commit sizes, but this effect was not present in proprietary projects. They also found that with $\alpha = 0.001$, adopting CI led to larger normalised proportions of closed bugs and issues.

Saidani et al. [45] tried to predict the outcome of CI builds using evolutionary search. They evaluated 56,019 builds from 10 large OSS projects that used a particular CI service. The metrics that they extracted from the CI were are almost all focused on an individual

build, except for the ”project history” class of metrics. They showed that this class is often influential in the decision of their trained model on the outcome of a CI build.

Vasilescu et al. [50] evaluated 246 OSS projects to investigate the effect of adopting CI on the quality and productivity of the projects. They used a large number of metrics collected from the CI and issue tracker. They found that teams that use CI were more effective at merging pull requests from their core developers and the core developers of the teams that used CI were more likely to report bugs to the issue tracker. They found that a larger number of non-bug related issues corresponded to a larger amount of reported bugs, that older projects received fewer bug reports from core contributors and that projects with larger test files were more likely to receive bug reports from external developers.

Wnuk and Maddila [51] performed a systematic literature review of agile and lean metrics related to requirement engineering. They found 22 metrics, of which 13 were used in empirical studies while the other nine were merely mentioned or proposed. They also categorised the metrics as being related to the time aspect of software engineering or the quality aspect and found that only nine metrics related to quality while the other 13 related to the time aspect.

Yu et al. [53] aimed to identify the factors that caused latency in merging pull requests in OSS projects. They collected data from a repository of OSS projects and a particular CI service. They evaluated 40,848 pull requests from 40 OSS projects. A large number of metrics was used in their analysis. They trained three models, each of which used more metrics than the previous ones. The model that contained all metrics found that a large number of metrics had a significant ($\alpha = 0.001$) impact.

Yu et al. [54] extended Yu et al. [53]. They investigated the factors that influenced the latency of PRs and the decision to merge or reject a PR. They collected 103,284 PRs from 40 OSS projects. For the acceptance of PRs, they found that the age of the project had a significant ($\alpha = 0.01$) correlation. The amount of open PRs also had a significant ($\alpha = 0.05$) correlation. For the time it took to merge a PR, they found that the age of the project, the size of the integrator team and the number of open PRs had a significant ($\alpha = 0.01$) effect.

Zhao et al. [55] investigated the effects of adopting CI in OSS projects. They first investigated if the adoption of CI had an effect on the commit frequency in the projects. They found that adopting CI resulted in fewer non-merge commits, but more merge-commits. They found that both the number of non-merge commits and the number of authors of the project (team size) had a significant ($\alpha = 0.001$) relation with the number of merge commits. They then investigated the amount of PRs that were closed and the time it took to close a PR. They found that after the adoption of CI, the number of closed PRs did not significantly increase and that it took longer to merge a PR. A possible explanation would be the duration of the CI. They then investigated the effect of adopting CI on the number of closed issues. They found that the number of closed issues increased over time, but that the rate at which they were closed decreased after adopting CI. Finally, the authors found that adopting CI had a positive effect on the number of tests per build.

Cluster	Metric	Referencing papers	Papers providing formula or computation	Papers providing insight
Code metrics	Quality of code	[12]		
Code metrics	Cyclomatic complexity	[28, 40, 52, 33]	[33]	
Code metrics	Readability Metrics	[33]	[33]	
Code metrics	Weighted Methods per Class (WMC)	[40, 52]		[52]
Code metrics	Response For a Class (RFC)	[40, 52]		[52]
Code metrics	Coupling Between Objects (CBO)	[40, 52]		
Code metrics	Depth In Tree (DIT)	[40, 52]		
Code metrics	Lack of Cohesion over Operations (LCOM)	[52]		
Code metrics	Data Abstraction Coupling (DAC)	[52]		
Code metrics	Number of classes (NC)	[52]		
Code metrics	Number Of Children (NOC)	[40, 52]		
Code metrics	Source Lines of Code (SLOC)	[20, 24, 45, 50, 22, 40, 52]	[20]	[20, 45, 50, 22]
Code metrics	Test lines/KLOC	[20, 22]	[20]	[20, 22]
Code metrics	Test asserts/KLOC	[20]	[20]	[20]
Code metrics	Test cases/KLOC	[20]	[20]	[20]
Code metrics	Size of Test Code (STC)	[24, 50]		[50]
Defects	Defect removal efficiency	[25, 33, 43]	[33, 43]	[43]
Defects	Change failure rate	[28, 37, 9, 36]		
Defects	Intermitted errors	[28]		
Defects	Error discovery rate	[33, 36]	[33]	[33]
Defects	Defect escape rate	[36]		
Defects	Average fixed defects/working day	[40]		
Defects	Test Improvement (TI)	[40, 43, 33]	[43, 33]	[43, 33]
Defects	Total Defect Containment Effectiveness (TDCE)	[40]	[40]	
Defects	Total Released Defects (TRD)	[40]	[40]	
Defects	Customer-Found Defects (CFD)	[40, 25, 25, 36, 51, 50, 2]	[40]	
Defects	Total new post release problems opened during the month (NOP)	[40]	[40]	
Defects	Total post release problems that remain open at the end of the month (TOP)	[40]	[40]	
Defects	Normalized Proportion of closed bugs (NCB)	[44]	[44]	[44]
Defects	Number of bugs during development	[2]		
Defects	Average reported defects/working day	[40]		
Outage	Failure Rate (FR)	[40]	[40]	
Outage	Mean time to detection (MTTD)	[2, 25, 36]		
Outage	Mean time to recovery (MTTR)	[9, 12, 14, 25, 28, 33, 36, 37, 52]		[9, 12, 14, 52]
Outage	Mean time between failures (MTBF) / reliability	[12, 25, 33, 9, 37]	[33]	[12]
Outage	Mean time to failure (MTTF)	[33]	[33]	
Outage	Project fail history	[45]	[45]	[45]
Process	Normalized Proportion of closed issues (NCI)	[44]	[44]	[44]
Process	Normalized count of (non-merge red.) commits (NCC)	[44]	[44]	[44]
Process	Normalized commit size (NCS)	[44]	[44]	[44]
Process	Number of merged PRs over a timeframe	[50]		[50]
Process	Number of rejected PRs over a timeframe	[50]		[50]
Process	Count of merge commits over a timeframe	[55]		
Process	Number of opened issues over a timeframe	[55]		
Process	Number of closed issues over a timeframe	[55]		
Process	Number of opened PRs over a timeframe	[55, 53, 54]		[53, 54]
Process	Mean PR latency over a timeframe	[55, 28, 53]		
Process	How often code is checked in	[2]		
Process	Change volume	[36]		
Process	Release/deploy frequency	[2, 9, 12, 14, 28, 36, 37, 43, 34]	[34]	[9, 37, 34]
Process	Cycle time	[7, 11, 28, 37, 38]		[7, 11, 37, 38]
Process	Lead time	[9, 14, 28] [36, 38, 51]		[9, 14, 38]
Process	Development time	[34]	[34]	[34]
Process	Deployment time	[34, 36]	[34]	[34]
SCRUM	Oldest done feature (ODF)	[34]	[34]	[34]
SCRUM	Schedule Estimation Accuracy (SEA)	[40]	[40]	
SCRUM	User stories carried on to the next iteration	[51]		
SCRUM	Fastest Possible Feature Lead Time	[34]	[34]	[34]
Testing	Test effectiveness	[33]	[33]	[33]
Testing	Automated test pass percentage	[36, 33]	[33]	[33]
Testing	Test coverage	[40]		
Testing	Number of tests executed per build	[55]		
Testing	Test improvement in product quality	[33]	[33]	[33]
Security	Dependency freshness	[10]	[10]	[10]

Table 2.2: Overview of the metrics extracted from literature. Each metric is assigned to a cluster and contains the papers that reference it. This overview also shows the papers that contain a computation, formula or provide insight for each of the metrics.

2.2.3 Extracted metrics

The aforementioned literature resulted in 66 metrics after processing as described in Section 2.1. Clustering them resulted in seven different categories. During the clustering, related metrics were clustered together. For example, the metrics ”Source Lines of code (SLOC)” and ”Cyclomatic Complexity” both relate to the code of the applications and thus were clustered together as ”code metrics”.

During the processing of the collected metrics, all metrics related to security had been removed as they were hard to extract automatically. As this category of metrics was deemed

important and significantly different from the other metrics, the decision was made to search for additional security-related metrics that could be computed automatically. To quantify the use of outdated dependencies, the query ”metric dependencies outdated” was used to obtain the paper by Cox et al. [10]. This paper introduced a metric to quantify how outdated the dependencies of a project are.

Table 2.2 contains all identified metrics together with their assigned category and the pieces of literature mentioning them, providing computations or insights for them.

In summary, the literature survey has resulted in 31 pieces of literature that contained 66 metrics, which have been clustered into seven different categories. Each of the metrics is annotated with the papers it was found in, the papers that provided a formula or computation and the papers that provided insights into the metric. The full list of resulting metrics can be found in Table 2.2.

Chapter 3

Interviews

The literature review provided insights into the academic reporting on metrics for DevOps. The current chapter assumes another point of view and aims to understand the use of metrics within ING. As part of this understanding, the second research question has been broken down into three sub-questions:

Specification of research questions

RQ2 What KPIs are used by ING and why?

- What KPIs are or have been used by ING?
- Why are those KPIs used, or why are they not used anymore?
- Where are they calculated from?

Collecting metrics is a two-edged sword: on one hand, collecting too few metrics can create an incomplete picture of the state of the organisation, while on the other hand collecting too many metrics can create an information overload and distract from the important information [6]. Thus, it is important to know which metrics are being measured within an organisation and why they have been selected. This chapter aims to understand the metrics that either have been used in the past or are in use by ING. Understanding why these metrics have been selected or deprecated can enable the organisation to further improve processes. To support the analysis of collected metrics, this chapter also intends to gain insights into the sources from which the metrics are calculated.

To this end, interviews have been performed with several ING employees. This chapter walks the reader through the process of participant selection, interview design, the processing of the interviews and the grouping of extracted codes before the generated results are shown. Those results lay the foundation for the analyses performed in Chapter 4.

3.1 Organisational structure of ING

Before elaborating on the selection of participants, it is essential to get an understanding of the organisational structure of ING. The bank uses the Spotify model [29] for organising its

teams. This model consists of four main components: *squads*, *tribes*, *chapters* and *guilds*. Each of those components is shortly explained in the next paragraphs. The model itself is visualised in Figure 3.1, which has been adopted from Kniberg and Ivarsson [29].

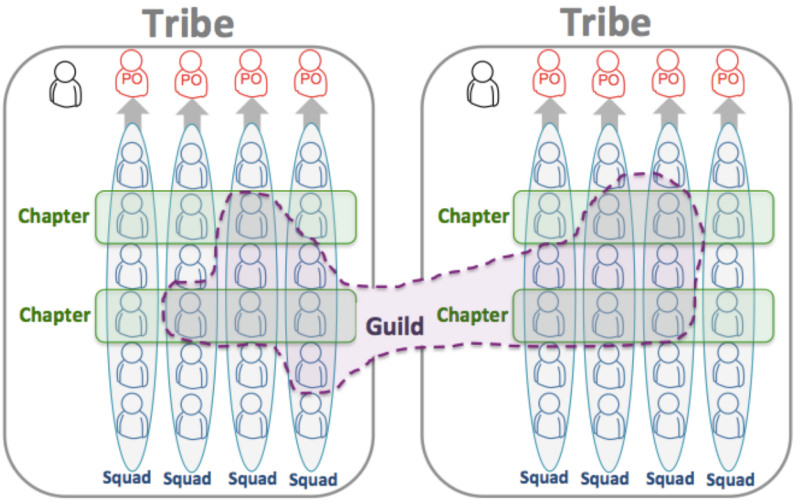

Figure 3.1: Visualisation of the Spotify framework. Adopted from Kniberg and Ivarsson [29]

Squad Squads are another name for autonomous, self-organising teams. Each squad is responsible for a part of the system and have all the different skills necessary to work on their part of the system. Each squad has a product owner (PO), who is responsible for what the squad does but not for how they do it.

Tribe Squads that work on similar products, components or services are organised together into tribes. Each tribe has a tribe lead, who makes sure that the squads of that tribe can function as optimally as possible. Within ING, tribes can additionally have an area lead. This leadership role is not part of the model described by Kniberg and Ivarsson [29] and the tribes can decide for themselves which responsibilities are given to this role.

Chapter Within tribes, employees with similar skills or functions are organised into chapters. These chapters are intended to enable communication and knowledge sharing between employees that face similar problems. As squads are autonomous, one squad may have already solved a problem that is very similar to that which another squad is struggling with. These chapters allow the sharing of such solutions. Each chapter has a chapter lead.

Guild Guilds are communities of people with similar interests. Like chapters, guilds are meant to enable communication and knowledge sharing between employees. Unlike the chapters, which are contained within tribes, guilds can span multiple tribes. It is often the case that similar chapters from different tribes are part of the same guild.

3.2 Selection of participants

This section contains a description of the selection procedure for participants. It has been a deliberate decision to not make any statements about the response rate of employees. Thus, this section will only report on the number of distinct employees that have been invited and the number of performed interviews. The reader shall be informed that the difference between those numbers can in part be explained by employees who forwarded the invitation to colleagues who were more involved with the use of metrics, resulting in a new invitation.

The initial stage of participant selection leveraged the knowledge of the company supervisor to identify several ING employees who had a leadership role within the organisation and whose area of expertise was likely to involve DevOps KPIs. Those employees were invited at the end of 2020 to participate in the interviews and the interviews from this first round were scheduled for January 2021. The second round of invitations was sent in February 2021 and the resulting interviews were scheduled in the same month. In total, 13 distinct ING employees have been invited for the interviews. As a result of those invitations, five interviews were scheduled in the first two months of 2021.

Function	**Count**
Tribe lead	2
Chapter lead	1
Area lead	1
Consultant Black Belt (Internal Process Consultant)	1

Table 3.1: Overview of the function of the interviewees

The functions within ING of the five interviewees have been depicted in Table 3.1. This information has been aggregated as describing the function of each interviewee could potentially leak identifiable information when combined with other information from this chapter. The following paragraph describes the teams of the interviewees, what they are working on and who their clients are. Interviewees are mentioned in no particular order. One of the interviewees is missing from this description, as a more interesting and important topic took presence during the interview.

Interviewed teams and their clients One interviewee builds a global reference platform for analytics at ING. It's the tribe's objective to give a perfect customer experience, make ING more data-driven and increase innovation and experimentation. However, their projects are focused on data science or machine learning and therefore place less emphasis on engineering and production environments. This focus is also reflected in their customers, as those are ING data scientists.

A second interviewee is part of the team that is responsible for a monitoring platform. This platform provides standardisation on top of different monitoring tools and provides insights into the reliability of client's applications so that they can remain in control of it. The platform also aims to facilitate better decision making. The team is not responsible for the data they are given or the availability of their client's applications. Clients can request

metrics and set their objectives for the metrics that are monitored. Their users are both technical and non-technical. Technical users are responsible for applications, which are often client-facing. Non-technical users are from independent auditors or are responsible for reporting information about a platform that enables sharing information between banks. How the platform is used is different for all users. Some teams designate a small number of people to get the graphs from the platform and determine together what those should look like. When a large incident is reported by the platform, management will try to prevent similar incidents in the future.

The team of a third interviewee builds a platform that includes a global CD pipeline. They intend to make sure that everything is code and that every change is performed using the pipeline. They provide other squads with the capability to work as automated as possible so that in turn those squads can measure themselves and set their targets. Their customers are all worldwide ING employees who have a function in IT.

A fourth team's objective is to make sure that applications are robust enough by focusing on availability and reliability. To do so, they offer several services related to testing and monitoring. Those tools only focus on deployed applications. Their users are nearly all software engineers from ING.

3.3 Interview design

The interviews were designed to be semi-structured [46] as the study is exploratory in nature. Each interview took 30 to 45 minutes. Before the interview, the participants had received an informed consent form. This form included an overview of the research and provided the interviewees with a description of how their information would be collected and processed and how anonymity and confidentiality would be achieved [21].

Protocol 3.1: Interview questions with estimated time per section

- ***5 min*** Introduction of research
  - Explanation of confidentiality and processing
- ***5 min*** Introduction
  - What is your function?
  - How are KPIs involved in your daily work?
- ***30 min*** Focused questions
  - What KPIs do you collect?
    - Why those?
    - What do you use them for?
    - Do you think they could have applications in other tribes?
    - Is there an overview or documentation of the used KPIs?
  - What are the inputs to the KPIs?
    - What limitations does this pose?
    - Have you investigated KPIs from other sources that do not have this limitation?
  - How do the KPIs you collect relate to ING as a whole?
- ***5 min*** Closing
  - Allow participant to ask questions
  - I will share the result of the interviews in the form of the chapter from my thesis.

Interview	**Duration (hh:mm:ss)**
1	00:21:40
2	00:37:59
3	00:32:16
4	00:32:01
5	00:34:26
	Total duration
	02:38:22

Table 3.2: Overview of the duration of the recordings of the performed interviews

During the interview, the interviewer intended to get answers to the questions posed in Protocol 3.1. As can be seen in that protocol, the interviews were divided into three main

sections: the introduction, the central part and the closing section. The goal of the central part was to answer the research question of the interview. To achieve this, the KPIs were approached from three different points of view: The KPIs themselves, the inputs to the KPIs and the relationship of the KPIs to ING as a whole.

The interviews were held using an online meeting platform and only the audio from the interviews was recorded. The five interviews resulted in 158 minutes of recorded audio. Table 3.2 displays the duration of each recording.

3.4 Processing of interviews

The recordings of the interviews were transcribed afterwards using multiple stages. Initial transcription was performed using an automated tool, which resulted in a baseline text. This transcription captured most of the conversation correctly but had trouble recognising technical terms in the audio. Therefore, the second round of processing was aimed at correcting the mistakes of the automated tool. This was done by playing the audio recording while reading the generated transcription. This way, the transcription was corrected. After this round of processing, the transcriptions contained the literal conversation between the interviewer and the interviewee. In a second pass over the data, the researcher listened again to each recording while reading the transcripts to be certain of a correct transcription and to get closer to the data. The audio recording was deleted after the correctness of the transcriptions had been established.

Interview	**Nr. of codes**
1	95
2	146
3	138
4	94
5	131
	Total nr. of codes
	604

Table 3.3: Overview of the number of codes resulting from each of the performed interviews

As the transcriptions at this point contained the literal conversations, they included the questions asked by the researcher together with a certain amount of verbose language that made further processing harder. To make further analysis faster and easier, the text of the researcher was removed from the transcriptions and the remaining text was summarised. The summaries aimed to reduce the amount of verbose language that was present in the transcriptions. Summarising the transcriptions also allowed the researcher to break up paragraphs of the interviewees' spoken text into smaller paragraphs that encapsulated their meaning. The summaries did not discard information, it merely served the purpose of making the transcripts easier to code in the next stage.

demonstration
When you want to show your reader an example of a quote with a code
42-1-2

Figure 3.2: Example of the card used to group the extracted codes. The top row depicts the code and the line below is the quote that this code came from. The last line identifies the code within the set of summaries. This example would be the second code extracted from the first quote of the 42nd interview.

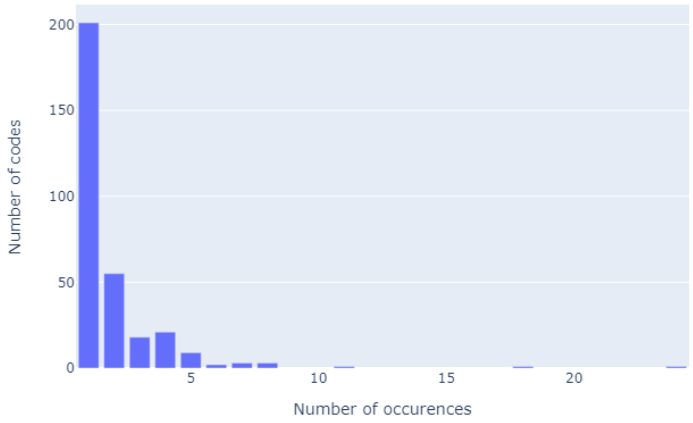

Figure 3.3: Frequency distribution of all the codes extracted from the interviews.

During this next stage, the transcriptions were coded using open coding [49]. Coding was performed by the researcher at the level of the message, instead of at the level of sentences or paragraphs as this was an exploratory study [19]. This resulted in quotes from the summaries with codes assigned. Table 3.3 depicts the number of codes that were extracted from each of the interviews. Figure 3.3 provides insight into the frequency distribution of the extracted codes. This figure shows that there is a long-tail distribution, where many codes occurred only once and a small number of codes occurred often. The full list of all extracted codes and their frequencies can be found in Appendix A. This appendix includes several codes that have been redacted for confidentiality reasons.

The codes were indexed with the interview they came from, the quote within the summary they were extracted from and the index within the list of codes extracted from this quote (first code, second code etc.). This information was encapsulated into a card to make grouping the codes easier. Such a card has been depicted in Figure 3.2.

3.5 Grouping of extracted codes

After the summaries were coded, the individual codes were grouped in an initial round of thematic grouping. The criteria for grouping the codes was loosely defined as that codes with the same or similar themes had to be grouped. For example, the codes ”Availability”, ”Uptime” and ”Downtime” were grouped under the theme ”Availability”. During this grouping, codes could be dropped from the study under certain conditions. The requirements for dropping a code have been stated in Protocol 3.2. In this initial state of grouping the codes, the reason for dropping codes was mainly that they were not relevant for the thesis.

Protocol 3.2: Rules for discarding codes

- If a quote has been assigned multiple codes
  - And at least one of those codes has already been clustered
  - And the quote can not be clustered with a different code
  - Then the code that can not be clustered can be discarded
- If a quote has been assigned multiple codes
  - And multiple codes get clustered together
  - Then all except the most applicable (to the cluster) of the codes from this quote in the same cluster can be discarded.
- A code is not relevant to the thesis and can be discarded if
  - It expresses an opinion on anything else than metrics.
  - It discusses internal architecture not related to metrics.
  - It discusses internal platforms not related to metrics.

The process of loosely grouping codes by theme resulted in a total of 21 themes where each theme consisted of a set of codes without further structure. During the initial thematic grouping of the codes, 56 codes were discarded. The remaining 547 codes were grouped into 21 themes. Table 3.4 provides numeric insights into the initial grouping of codes.

After this process had been completed, the intermediate product consisted of 21 themes, where each theme consisted of a set of codes. In the next processing step, each theme was considered independently. Within each theme, similar codes were placed together, creating loosely defined sub-themes. For each of the sub-themes, similar codes were grouped into a statement. By iteratively considering all other codes in the sub-theme, the initial statements were extended and merged into larger overarching statements. This resulted in a hierarchy of statements and codes. At the end of this process, the sub-clusters had been transformed into a single statement with an underlying hierarchy of statements and codes. In some cases, the statements of the sub-clusters could be merged until a single statement per theme was constructed. In other cases, the statements were sufficiently different from each other to prevent merging. This indicated that the initial theme should have been broken up into smaller themes. In the final stage of processing the interviews, the aggregated statements were rewritten to improve their readability. Thus, the results of this chapter are not the literal statements as extracted from the transcripts, but an edited version of those statements. Note that this editing phase did not change the meaning or content of the state-

ments. During this rewriting phase, it was discovered that the themes ”Interviewed teams” and ”Customers”, which describe the interviewed teams and their customers respectively, were so intertwined that it was decided to merge the two themes into a single overarching theme named ”Interviewed teams and their clients”.

During this iterative process, codes could again be dropped according to the rules of Protocol 3.2. In contrast to the initial thematic grouping, the most common reason for dropping codes in this step was that two codes from the same quote ended up being in the same cluster. Table 3.4 provides a numerical overview of the grouping process. This figure shows the extracted themes, the number of codes in each theme after the initial round of grouping, the number of codes that have been discarded in each theme while clustering the grouped codes and the final number of codes in each theme. It also shows the number of interviewees that contributed to the final statement of each of the themes.

	Nr. of codes			
Theme	**After initial grouping**	**Discarded during clustering**	**Final**	**Nr. of interviewees**
Feedback	14	0	14	2
Standardization	13	0	13	2
Impact/Business value	10	2	8	2
Interviewed teams	54	17	37	4
Customers	19	5	14	4
Maturity	54	13	41	5
DORA	33	19	14	3
Drawbacks	42	10	32	3
Functions of KPIs	37	5	32	4
Selection of KPIs	15	1	14	5
Sources of KPIs	57	6	51	4
Data collection	31	0	31	5
Used KPIs	33	2	31	4
Throughput	8	0	8	1
Automation	14	0	14	2
Adoption	8	0	8	3
Agility	31	3	28	4
Availability	38	7	31	3
MTTR/MTTF	17	5	12	2
Risk	8	0	8	3
Compliancy/Regulations	12	1	11	3
Discarded	56	96	-	-
Remaining nr. of codes	548	-	452	-

Table 3.4: Overview of the process of grouping and clustering codes. This table depicts the themes, the number of codes after the initial grouping, the number of codes that have been discarded while clustering, the final amount of codes per theme and the number of interviewees contributing to the final statement of the theme from left to right respectively. The bottom two rows depict the total number of discarded and remaining codes. Note that in the case of the centre column, the number of discarded codes is the sum of the column.

3.6 Results

This section reports the aggregate statements that have been generated for each of the extracted themes. To protect the privacy of participants, no verbatim statements have been included, nor have specific participants been mentioned within the aggregated statements.

Feedback The teams of two of the interviewed participants collect *feedback* on their product or service. This feedback can be collected using formal methods such as interviews, or using informal methods like a quick talk at the coffee machine. Feedback is especially important when you have a very demanding customer and it is valuable because it can lead to debate, inspire new KPIs and capture information that KPIs cannot express. Therefore, the human connection that collecting feedback offers is important.

Standardisation Participants discussed two ways to achieve the *standardisation* of reports and metrics. In one of the applications, data is collected from multiple sources and transformed into a standardised metric. In another application, data is stored in a standardised manner. The process of standardisation provides abstraction and enables the collection of metrics throughout the organisation.

Business value and impact It is important to know the *business value* that each activity generates for ING. This is hard to measure in practice, although it can be approximated for some activities by evaluating the profits that are generated by a related activity.

Interviewed teams and their clients This theme has been discussed in Section 3.2 when describing the interviewees.

DORA The DORA metrics have been used as a starting point for other metrics, but it is unlikely that they will have a relation with all ING KPIs. Collecting them on the scale of a global organisation requires much data collection and standardisation, which is likely a significant challenge for many companies.

The DORA metrics are useful when the goal is to improve the software delivery performance. Thus, the metrics relate to ING's goal of being able to move fast. When the DORA metrics are used, there should be an improvement in the DORA metrics on the organisational level, while there should be improvements that are linked to those metrics on lower levels of the organisation.

Among interviewees, there was consensus on the statement that in general, it is possible to achieve both throughput and stability at the same time. Some interviewees drew a parallel with the literature on the lean way of work, where throughput is increased by increasing quality. However, it was also noted that whether a company can achieve both at the same time depends on the business. The heavy regulations that ING is subject to might influence this interaction and therefore it might not be possible within the bank to achieve this enabling behaviour between throughput and stability.

Drawbacks The risk of using metrics is that people can start to cheat the metric by behaving differently when a metric is being measured. If this is the case, more explanation is needed on why the metric is being measured and how it is being used. For this reason, putting metrics on display can have consequences and doing so might not be a good idea if the goal is to improve something. On the other hand, observing this behaviour should not be a reason to not use metrics, but it should motivate to give explanations and have discussions.

Besides the risk of cheating metrics, there are several drawbacks to metrics in general. One of the drawbacks is that there is no set of KPIs that can be used for everything or that works for everybody, as different activities require different KPIs. Therefore it is important to understand what is being measured and why. On top of this, is not always possible to collect KPIs reliably.

Furthermore, KPIs do not cover the whole performance of applications as experienced by users, which is why collecting feedback is important. The manual evaluation of information also allows for interpretation, while collecting automated KPIs makes everything black and white. Another drawback is that metrics can be seen as private data and can be used to compare people or measure productivity, although this is frowned upon by some parties. Therefore, it is important that the data is treated carefully and that such applications are taken into account.

Functions of KPIs Within ING, there is a link between the goals of the organisation, why those goals exist and the KPIs that are collected. KPIs are used to set priorities and to assess the achievement of *service level objectives* (SLOs) and *service level agreements* (SLAs). They are very important in day-to-day activities and throughout the development cycle. Teams steer and check for the outcomes they want to achieve by making decisions based on KPIs but not by making them the primary goal.

Selection of KPIs It is important to first have the capacity to measure reliably, before starting to define KPIs. When this capacity is established, practices in the rest of the industry can serve as inspiration for selecting KPIs, but it is important to think about what is relevant for the team. It is important to understand what should be measured and what the aim of this measure is. This understanding also enables the use of a proxy if the original KPI cannot be measured.

There are three main reasons for selecting KPIs. They are either selected because it is known from experience that they are important, because higher management requires them to be measured or because there is an objective to improve something.

Sources of KPIs The KPIs that ING uses are calculated from many different sources. Some of them are collected by specialists from other platforms and are then sent to the teams via spreadsheets or presentations. Others are collected from the version control system and many metrics are calculated from the monitoring platform, the *IT Service Management* (ITSM) platform or the Continuous Delivery (CD) pipeline.

The monitoring platform collects raw metrics from various sources and consolidates them. Those sources range from end-to-end testing on production to the routing software,

provide different types of data and generate false positives differently. The platform creates data on *interruptions* from the first down- and first up-events it receives from the different sources.

The CD pipeline is event-driven and highly integrated. Because of that, it is measurable from beginning to end. Underneath the pipeline lies a data lake that allows the responsible team to calculate any metric they want. This was not possible in the old pipeline. This new pipeline is important because it will play a central role in software deployment. Automating both the software and the infrastructure will allow teams to focus on their objectives.

Data collection The collection of the underlying data is the first step when collecting metrics. However, doing so reliably on a global company scale is difficult, especially since the quality of this data is very important when it is used for the calculation of metrics.

Data collection can be performed manually or automatically. Manual data collection can be used to calculate a metric by itself, or it can be used to collect a part of a metric. One example of manual data collection is the process of gathering feedback. This can take the form of surveys, formal interviews or informal discussions. This way of collecting data is not allowed for metrics that need to be auditable, although many metrics are collected manually at first, and get automated later. Automated data collection has the advantage of being auditable. Correction of automatically collected data can be performed manually or automatically following predefined rules.

Used KPIs The interviews revealed a large number of used KPIs. This theme has been established to group related KPIs together. Some KPIs are used throughout the whole organisation, such as KPIs around cost and budget. The cost has multiple components such as employee costs and costs of the used platforms. Organisations also want to have insight into how happy their employees are. Some teams have a KPI on the number of *incidents* because it says something about the prioritisation and how the backlogs are filled. Others have a KPI on the timely response to reports from the risk department.

Different domains collect different metrics that best fit their area of expertise. The data-science domain uses metrics related to the use of their platform, how reliable it is, the number of deployed models and how long it takes to deploy a model. Squads that are interested in the *uptime*, *success rate* and *latency* metrics of their products can use the monitoring platform to get insights into them. Those metrics are collected by the monitoring platform, and they are the most important metrics that the platform collects. The success rate and latency metric apply to the APIs for sharing information between banks. For a tool that tests system resilience, there are KPIs on how to give feedback to teams and on not doing too much damage.

Throughput There has been a focus on *throughput* in the past. It was measured by the duration of sprints, and the lead time for changes. There are still teams that use those metrics and teams are interested in how they can improve by speeding up. However, it was established that throughput in a bank is mainly influenced by regulations, compliance

and risk. Thus, the throughput metrics were replaced by metrics on how many of the *risk controls* have been automated, see the automation KPI below.

Automation There is a detailed change process in place to prevent malicious changes. The detailed change process is intended to be a measure for risk control. Automating that risk control will reduce the amount of manual work required to perform a change, and will thus allow the throughput to be increased. For the process of automating it, there is a KPI on how much of the work around risk controls is still being done manually that measures how many of those risk controls have been automated. A lot of time is spent on increasing this automation of risk controls. All assets have the same risk controls.

Adoption Teams have KPIs on the adoption of their platforms. This adoption can be expressed in terms of the number of people using a platform or service or in terms of the increase or decrease in the number of users that use the product compared to an earlier measurement. Adoption does not only focus on the number of people that are actually using the product but also on the number of people that have the ability to use the product.

Agility The monitoring platform has collected agility metrics in the past. They were mostly used by one specific tribe. Those metrics might become important for them again in the future, or others might request them. They were computed from the DevOps platform and mainly related to deployments and commits such as the success status of deployments, the number of commits per deploy or the deployment frequency.

The deployment frequency does not say anything about the amount of business value that is being created but can say something about the performance and agility of a team. It is relevant in some domains, such as customer-facing domains. In domains where deployments mostly involve configurations, the frequency of those deployments is not relevant. The deployment frequency could not be measured reliably in the past, as engineers started to deploy to the development environment very frequently when the pipeline was automated. In the future, it might be possible to measure the deployment frequency directly using the new continuous integration pipeline.

As it is not yet possible to measure the deployment frequency, the number of changes is used as a proxy on the organisational level. This proxy is close to the *batch size*. However, as there are squads that use a *release train*, the number of changes does not always say something about a squad. In that case, the number of changes (or the batch size) says something about all squads on the train, but not about an individual squad. For an individual squad, the number of items that they put on the release train every sprint should be measured.

Availability *Availability* is important for both client-facing and internal applications. For the former, it is important because the organisation wants to be there for its customers and because ING is obligated by the Dutch National Bank to have a certain level of availability. The availability metrics are more important than the DORA metrics, although the importance of availability also depends on the application and the time of day. An incident during

prime-time will upset a large number of customers, while an incident of the same duration in the middle of the night will affect fewer people.

In the past, the number of incidents was used as a proxy for availability. It was discovered that this metric was prone to cheating, as clients complained about the availability of applications while there were no reported incidents. Because the number of incidents was being measured, people stopped reporting incidents. Currently, the availability is measured using uptime. The monitoring tool monitors the availability of assets, which can be applications or parts of applications, such as APIs. The monitoring platform registers when an asset is down and when it becomes available again. From this registered *downtime*, the uptime is calculated by subtracting it from the total amount of time in the month. The uptime is reported over windows and includes both the uptime percentage for the current month up to when it is requested, as well as a burndown graph calculated over the whole month. A distinction is made between planned and unplanned downtime. Unplanned downtime is due to incidents, planned downtime happens when a change is rolled out. Uptime is the most important metric for the users of the monitoring platform. Management uses the monitoring to decide if they can do a big release or not.

MTTR/MTTF The monitoring platform is developing the *MTTR* (Mean Time To Restore) and *MTTF* (Mean Time To Failure) metrics to add them to the platform. The MTTR is a very important metric and is computed from the interruptions, it is the average time between when an asset goes down and when it recovers. The MTTF is the average time between failures and is much larger than the MTTR.

One participant highlighted the relation between interruption MTTR and lead time of tickets (*incident MTTR*). Often a team notices an incident using the monitoring platform and starts to repair it even before a customer notices and creates a ticket. The time between those two events is part of the interruption MTTR, but not of the incident MTTR. Therefore, the incident MTTR for bugs does not say everything about the interruption MTTR.

Risk There exists a KPI on *risk* that is enforced down from the European Central Bank and in turn from higher management down to the teams. This KPI makes sure that there are certain controls in place and relates to ING's goal of being compliant while at the same time optimising the cost associated with risk. This process of mitigating risk and remaining compliant influences the speed with which software can be developed.

Compliance/regulations As discussed before, there are regulations around the risk that influence the speed at which software can be developed. Compliance with regards to software development on a more general level is also concerned with conforming to the licensing agreements of used libraries. Regulations also apply to the availability of certain assets. To comply with those regulations, the availability KPI needs to be auditable. In turn, this means that only automated data collection can be used for this metric.

Maturity The implementation and collection of KPIs consist of multiple stages and matures as it progresses. Teams start by deciding what KPIs they want to collect. This process

draws inspiration from what is going on in the industry, including the DORA report. The metrics from the industry are adapted to fit the needs of ING. After deciding what KPIs they want to collect, the squads decide where the metric should be calculated from and how the raw data should be collected. Initially, the collection of raw data is often manual. This stage requires aligning throughout the organisation that this metric needs to be collected and that it is good to report on it. After the manual data collection of the KPIs has been established, some of them are automated but this takes some time.

This process of implementing new KPIs never stops, as what is being measured changes over time. Causes of this change vary from users losing interest in a metric, to progressing insights into the most influential factors of the team's objectives. As a result of this process, some metrics are under development and metrics that need to be revised. What is being measured also depends on the tooling that is used. Thus, when the tooling changes, what is being measured changes as well. If a KPI cannot be measured directly, a proxy that is as close as possible to the original KPI is used. This proxy also depends on the tooling and might change when the tooling changes. If a new opportunity for measuring the KPI arises, it is used. As an example, ING is moving from an old CD pipeline that consisted of multiple tools to a single one that is fully integrated. This new pipeline includes an underlying evidence store which enables the pipeline to be fully measurable. This capability is made available to everybody by boarding them onto the new pipeline. The old pipeline did not provide this capability due to being connected by multiple tools. This transition is likely to enable the direct collection of the deployment frequency.

Summary In conclusion, the interviews have resulted in a list of 27 metrics that either are currently in use by ING or have been used in the past. These metrics and their usage status have been depicted in Table 3.5. This table contains three entries that require a note. Firstly, the number of incidents is both used in the past and used currently. In the past, it has been used as a proxy for the availability while it is currently being used to say something about the prioritisation and how the backlogs are filled. Secondly, the MTTR and MTTF have both been denoted to be currently in use, while the interviews indicated that they are under development. This decision has been made to simplify the different statuses of the metrics.

The two main sources for the metrics that have been discussed in this section are the ITSM platform for metrics relating to changes and incidents, and the monitoring platform for metrics relating to operations such as uptime or latency. The interviews have also indicated that the new CD pipeline in the future will enable teams to collect metrics such as the deployment frequency in a more reliable way.

Metric	Current/Past
Feedback	Current
Profits of related activity	Current
DORA	Current
Cost and budget	Current
Employee happiness	Current
Number of incidents	Past+Current
Timely response to risk incident	Current
Platform usage	Current
Platform reliability	Current
Number of deployed models	Current
Duration of model deploy	Current
Uptime	Current
Successrate	Current
Latency	Current
Giving feedback after resilience test	Current
Amount of damage done during resilience test	Current
Sprint duration	Past
Lead time for changes	Past
Automation of risk control	Current
Adoption	Current
Success status deployments	Past
Number of commits per deploy	Past
Deployment frequency	Past
Number of changes	Current
MTTR	Current
MTTF	Current
Risk	Current

Table 3.5: KPIs extracted from the interviews. For each metric, it is indicated if it is currently in use or has been used in the past.

Chapter 4

Data analysis

The overarching goal of this thesis is to understand how the DORA metrics relate to each other within ING, and how they relate to the metrics that are additionally collected by the organisation. It also aims to understand the usage and applicability of proxies for approximating KPIs. The previous chapters have laid the groundwork for these understandings by exploring the metrics that are reported in the scientific literature and investigating the KPIs that either are currently in use or have been used in the past by ING and where they are calculated from. As a final step towards reaching this goal, this chapter performs the data analysis required to answer the last three research questions, which have been depicted below. This section makes a distinction between the DORA metrics and the ING-specific metrics that are additionally collected within the bank. Both sets of metrics are collected within ING and this distinction is introduced with the sole purpose of structuring this thesis.

Specification of research questions

RQ3 **How do the four metrics as identified by DORA [16] relate to the other metrics used by ING?**
RQ4 **What is the influence of organisational structure on the relationships between the DORA and ING-specific metrics?**
RQ5 **What proxies are used by ING and how do they relate to the metric they substitute?**

This chapter describes the journey of answering those questions. As a first step, the available data sources will be explored and related to the sources of KPIs as extracted from the previous chapter. Having an understanding of the platforms that offer data, the metrics that are in use by ING are introduced in terms of their definitions. After the available data sources and the definitions of the metrics are known, the data sets that have been extracted from the available sources will be described. As a final preparation step, a description is provided on how the metrics used in this thesis are calculated from the collected data sets. After the metrics are calculated, this chapter performs the data analysis required to answer the aforementioned research questions. The first section explores the available data, the

three following sections describe a research question each. From this chapter onward, a distinction will be made between *concepts*, printed in italics, and ***metrics***, printed in bold italics, to improve the readability. This notation of metrics only applies after their definition has been provided to create a distinction between the conceptual understanding of a metric (denoted as a *concept*) and the actual defined metric (defined as a ***metric***). Furthermore, *concepts* will only be stylised the first time they are introduced, while ***metrics*** will be stylised every time they are used.

4.1 Preparation of data

Answering the three research questions of this chapter requires that the metrics are selected, collected and processed before they are evaluated. This section walks the reader through the process of selecting metrics to investigate, exploring available data sources, introducing the collected metrics, describing the available data sets and calculation of the metrics from those data sets.

Metric	Current/past
DORA	Current
Uptime	Current
Latency	Current
Success rate	Current
Number of incidents	Past+Current
Number of changes	Current
MTTR	Current
MTTF	Current
Lead time for changes	Past

Table 4.1: Summary of the IT-related metrics extracted from the interviews. For each metric, it is indicated if it is currently in use or has been used in the past.

4.1.1 Selection of investigated metrics

The interviews of Chapter 3 have resulted in a list of 27 metrics, which have been depicted in Table 3.5. However, it is not possible to investigate all 27 metrics for multiple reasons. Some of these metrics have been collected in the past and are no longer available. Others can not be collected from the internal ING systems, as they are delivered to the teams in different ways. Therefore, a selection has been made of metrics that are still available and that could be collected from the systems. These metrics all relate to IT processes within ING and have been depicted in Table 4.1. The remainder of this section will investigate these metrics further and elaborate on them.

4.1.2 Exploration of available data sources

Chapter 3 has shown that most of the metrics that have been or are in use are calculated from a small set of platforms. This section aims to equip the reader with a strong understanding of those platforms and what they have to offer in terms of available metrics.

Monitoring platform The interviews have indicated that the monitoring platform is an important source of metrics relating to the operation of applications. This platform has the ability to measure the uptime, latency and success rate metrics of onboarded applications. As this section aims to provide insights on the available data sources, the definitions of those metrics are out of scope and will be discussed in a later section. The monitoring platform measures those metrics for individual *assets*, which can be whole applications or parts thereof such as APIs. For this section, it suffices to know that each of the three metrics is expressed as a percentage and is thus bounded by: $0 \leq \text{metric} \leq 100$. The squad that is responsible for a specific asset can set a monthly objective for each of the metrics. The monitoring platform displays both values in red if the objective is not met and in green otherwise. It also displays a list with all interruptions that occurred in the selected month.

The monitoring platform exposes an internal API that allows the collection of the three aforementioned metrics, interruptions and the organisational structure, although this list is not exhaustive. The API allows requesting metrics from *resilience critical* assets only, or from all assets within a part of the organisation. Management has defined a list of 25 applications that have been defined as being resilience critical.

ITSM platform Some of the metrics mentioned in Chapter 3 are affiliated with incidents or changes and are recorded in the ITSM platform. Instead of interfacing directly with the platform to extract the raw data, this thesis leverages the Business Intelligence (BI) platform that is used within ING. This platform also allows the incidents and changes to be accessed and makes this data readily available so that no queries or API requests had to be created.

The BI platform provides access to incidents, changes and the lead time of stories. Again, the exact definition of those types of data will be discussed in later sections. The lead time that was available from the BI platform was only calculated on *stories* that were closed in the last nine months before the data was requested. For reasons discussed later, there was a need to have this data over a larger period than nine months. The squad that is responsible for the data was asked if the data over the full year of 2020 was available. They provided this thesis with the source data of the lead time, spanning the desired period.

4.1.3 Introduction of metrics in the context of ING

The previous section has introduced the platforms that have been used for the collection of metrics for this thesis. This section will define the metrics that have been collected in terms of their conceptual meaning and reason for inclusion when appropriate.

ING-specific definitions of DORA metrics

The DORA metrics are available from the BI platform within ING. However, they are not the metrics as defined in the report of Forsgren et al. [16]. The interviews in Chapter 3 established that measuring the DORA metrics as described by the DORA report is hard to do on a global company scale. The interviews also indicated that progress is being made and that measuring the DORA metrics should become possible with the new continuous delivery pipeline. However, at the moment this thesis was written, ING used proxies to measure the DORA metrics. This way of working allows the DORA metrics to be collected on an organisational scale. This subsection describes the proxies that are used by ING and how they relate to the original definitions of the DORA metrics as by Forsgren et al. [16].

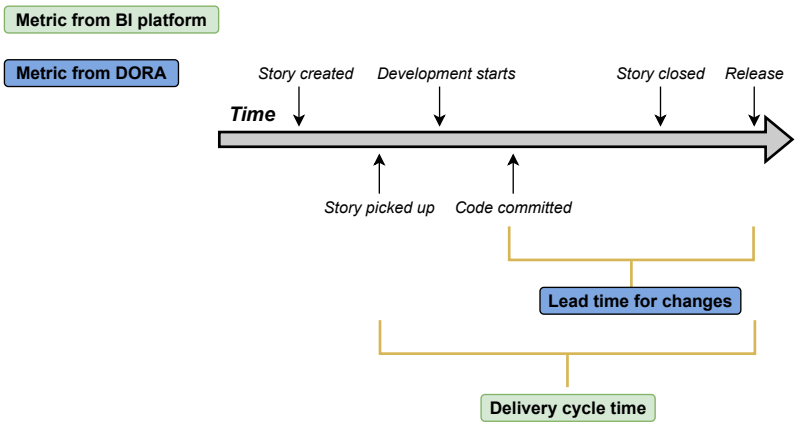

Figure 4.1: Visualisation of the delivery cycle time as available in the business intelligence platform and the lead time for changes as defined by DORA

Deployment frequency The DORA report defined the *deployment frequency* as how often an organisation deployed code to production or released it to end users [16]. Instead of measuring the number of deployments or releases, ING uses the number of change records assigned to the production environment that are registered in the ITSM platform. Change records contain a closure code, which can be either successful, successful with problems or failure. ING calculates the deployment frequency using the change records that were successful or successful with problems. This proxy has been discussed in Chapter 3.

Change failure rate The *change failure rate* has been defined by Forsgren et al. [16] as the percentage of changes to production or releases to end users that result in degraded service, such as service impairment or service outage and which require remediation such as a hotfix, rollback, fix forward or patch. Instead of looking at the effect of a release, ING uses the closure code of changes as a proxy for the effect of the change. The change failure rate is calculated as the number of changes that were successful with failure or had failed divided by the total number of changes in a month.

Time to restore service The time to restore service has been defined by the DORA report as how long it generally takes to restore service when a service incident or a defect that impacts users, such as an unplanned outage or service impairment, occurs [16]. The MTTR metric as discussed in Chapter 3 is following this definition. This metric is under development and will be part of the monitoring platform in the future. If squads want to have access to this metric, they need to onboard their assets to the monitoring platform. The proxy that ING uses for the MTTR is calculated from incident tickets obtained from the ITSM platform and thus is available for the whole organisation, instead of for only those who registered their assets. This proxy measures the time between when an incident is created and when it is resolved. It uses incidents that relate to the production environment. Within ING, the *incident MTTR* per month is calculated as the median time it took the squad to resolve the tickets that were resolved that month.

Lead time for changes The DORA report has defined the *lead time for changes* as how long it takes to go from code being committed to code successfully running in production [16]. Instead of tracking individual commits, ING approximates this metric by the *delivery cycle time*. The delivery cycle time is defined as the time between when a story is picked up and when that story is running in production. The relation between the delivery cycle time and the lead time for changes as defined by DORA has been depicted in Figure 4.1. To remain in line with the DORA report, the rest of this thesis will refer to the delivery cycle time as the lead time for changes.

Operation-plane metrics collected by ING

As described in the interviews of Chapter 3, there are several IT-related metrics that either have been or are being measured by ING to help them steer in the right direction. This section describes those metrics and their definitions.

Normalised latency objective Squads can measure the latency of requests made to their assets and define a threshold for the maximum allowed latency. The monitoring platform computes the percentage of requests for which the latency was smaller than the defined threshold. For example, suppose there is an asset with a latency threshold of 20 ms and out of 100 requests made to it, 5 had a latency of more than 20 ms. Then the monitoring platform computes the percentage as $(1 - 5/100) * 100\% = 95\%$. This metric is called "latency" within ING but to prevent confusion with the actual latency, this thesis will refer to it as the "normalised latency".

Squads can define a minimum value for the normalised latency that they expect from their asset. This is called the *normalised latency objective* within this thesis. The observant reader might notice that this thesis only reports on the normalised latency objective and not on the normalised latency itself. When performing the data analysis, it was discovered that there were too few data points of the normalised latency to make reliable statements about it. Therefore, it has been omitted.

Uptime As described in Chapter 3, the *uptime* of assets is defined as the percentage of time that they were available in a given month. This percentage is calculated by measuring when an asset becomes unavailable and when it becomes available again. The time between these two events is regarded as downtime, and the sum of all downtime of a month is subtracted from the total amount of time in that month, resulting in the absolute uptime. Dividing this by the total amount of time in that month results in the uptime percentage metric.

Uptime objective Squads can define the minimum uptime percentage per month that they require of each asset. This percentage is called the *uptime objective* and is defined per asset per month.

Uptime objective achieved As described in Subsection 4.1.2, the monitoring platform displays the uptime in green if the uptime of an asset is at least as large as the uptime objective, and red otherwise. This thesis translates that colour into a numerical value: it is 1 if the uptime is at least as large as the objective, and 0 otherwise. This metric is defined per month per asset and will be referred to as the *uptime objective achieved* metric.

Uptime margin The monitoring platform displays the uptime and the uptime objective per month per asset. This thesis makes the difference between the uptime and the uptime objective explicit by calculating the *uptime margin*. This is defined as the uptime minus the objective. This metric is not explicitly collected by the monitoring platform but has been added to this thesis to generate further insights.

Uptime resilience critical The monitoring platform allows the user to choose between seeing all assets, or only resilience critical assets. To get insights into how the resilience critical status of assets relate to other metrics, this state has been selected as a metric in this chapter. It is defined per asset and interpreted as 1 when the asset is marked as critical and 0 otherwise. It will be referred to as the *uptime resilience critical* metric.

Number of incidents As discussed in Chapter 3, the *number of incidents* has been used in the past as a proxy for the availability and is calculated from the ITSM platform. Currently, it is being used to say something about prioritisation and how the backlogs are filled. Given that incidents can have multiple priorities, this thesis will consider them both as an aggregate over all priorities and per priority.

Number of interruptions As described before, the uptime of assets is determined by registering when an asset becomes unavailable and when it becomes available again and using this time to calculate the total uptime percentage in that month. The time between going down and coming online again is called an interruption and is measured by the monitoring platform per asset per month.

Although the *number of interruptions* is not directly measured by the monitoring platform, it has been decided to add it to this thesis for two reasons. First, the number of

incidents is collected as a metric and contrasting this with the number of interruptions could generate valuable insights. Second, the number of interruptions can be thought of as the abstraction of the list with interruptions from the dashboard.

Interruption MTTR Chapter 3 indicated that the *interruption MTTR* is currently under development for incorporation in the monitoring platform. The metric is defined as the mean duration of an interruption for a given asset in a given month.

Interruption MTTF Similar to the interruption MTTR, the *interruption MTTF* is under development for the monitoring platform. It is defined as being the time between the end of an interruption and when the next interruption begins. It is defined per asset per month.

4.1.4 Description of available data sets

This section will elaborate on the data that has been collected from the different data sources and how it is prepared for further steps. All data discussed in this section has been collected for the full year of 2020.

Without going into further detail, it is important to know that future analysis requires all data points to have a column for the squad with which that data point is associated and the tribe to which that squad belongs. As mentioned before, the monitoring platform measures metrics per asset. To satisfy the aforementioned requirement, a mapping needs to be made between the names of assets and the squad they belong to. Additionally, there needs to be a mapping between the names of the squads and the tribe they are part of.

The API of the monitoring platform is leveraged to generate the former mapping. This API has the functionality to return the organisational structure. From this structure, the mapping between squads and assets is generated. The later mapping is extracted from the data that is obtained from the ITSM platform. This data includes not only the squad of each data point but the tribe as well. To generate the mapping, those two columns are extracted for each data point and added to the map. In case a squad is mapped to multiple tribes, a majority vote is used to obtain the most likely tribe. In case of a tie, the squad is marked as not having a tribe, and future data points belonging to this squad will be discarded.

	Dataset	Obtained from	Nr. of datapoints	Nr. of squads	Nr. of tribes
0	Incidents	BI	417597	870	133
1	Changes	BI	9020	1013	133
2	Cycle time	BI - custom datasheet	19003	531	97
3	Latency	Monitoring	5577	82	31
4	Uptime	Monitoring	9147	79	29
5	Interruptions	Monitoring	71527	116	34

BI = Business Intelligence platform, Monitoring = Monitoring platform

Table 4.2: Descriptive statistics for the available data sets.

Figure 4.2: Distribution of squads per tribe per data set. Each subplot represents a data set. The x-axis shows the number of squads per tribe, the y-axis shows the number of tribes of that size.

Now that the foundation for satisfying later data analyses has been laid, the different data sources can be introduced. Figure 4.3 displays the different types of available data. In this figure, a distinction is made between data obtained from the different sources. The middle column contains the two generated data sets used to map assets to squads and squads to tribes. The remainder of this section will describe each of the data sources separately. Table 4.2 provides a numerical summary of the data sets. It displays the source of each data set, the number of data points in it and the number of unique squads and tribes in the data set. Figure 4.2 extends this summary by displaying the distribution of the number of squads per tribe for each of the data sets.

The observant reader might have noticed that Subsection 4.1.2 mentioned that the monitoring platform also measures the success rate, while this type of data is not present in Figure 4.3. This type of data has been collected and has been involved in experimentation, but it was established that there were too few data points to use them for the final analysis in this thesis.

Cycle time The data set containing the cycle time is obtained from the squad that is responsible for it within the BI platform. Each entry in the data set contained the name of

the squad that did the change, together with its tribe. It also included the date at which the change was closed and the cycle time.

Incident The data set containing all reported incidents was obtained from the BI platform. Each reported incident included the name of the squad who was responsible for the affected application, together with the tribe of the squad. Each incident also had a priority between one and four, where a priority of 1 was most important. Each incident also included the date it was created and resolved and its time to restore in hours.

Change The data set containing the changes was obtained from the BI platform. Each change included the name of the squad that was assigned to the change, together with its tribe. Each change also included a closure status, indicating if the change was successful, partially successful or had failed. Each change also included the date at which the change had ended.

Uptime Uptime data was collected from the monitoring platform. As discussed before, the data was collected using the internal API. This API returned JSON objects. To create an overview that is easier to understand Figure 4.3 depicts a flattened and preprocessed version of the data extracted from the platform. Each request made to the platform was allowed to either request data on all assets or resilience critical assets only. For each asset within an organisational structure, both requests were made. Assets that appeared in both requests were marked as resilience critical, assets that did not appear in both were marked as not resilience critical. Each data uptime data point corresponds to the uptime of one asset in one month. The date is the last day of the month that the uptime relates to. Each uptime data point also included the name of the asset that is referred to. Each data point also includes the objective that the squad of that asset had set themselves for that month. Both the uptime and the uptime objective are expressed as percentages of the total amount of time in a month, and both are thus bound by: $0 \leq$ uptime (objective) ≤ 100. As an example, suppose a month with four weeks that contains 672 hours. If an asset has been down 1 hour in that month, the reported uptime is: $(1 - 1/672) * 100\% = 99.85\%$.

Latency Although the retrieved data is called ”latency” within ING, this is actually the normalised latency as described before. Similar to the uptime, it is collected using the internal API of the monitoring platform, and the data schema shown in Figure 4.3 depicts a flattened and preprocessed version of the data extracted from the platform. The data set containing information about the latency is nearly identical to that of the uptime data, with the only exceptions being that it includes the normalised latency of an asset for a month instead of the uptime and that the objective in each data point relates to the normalised latency instead of the uptime.

Interruption The API of the monitoring platform also allows retrieving all detected interruptions within a defined time frame. Each interruption includes the name of the asset that it refers to. It also includes the date and time at which the interruption started and ended.

The cause type of an interruption indicates if it was caused by an unexpected event, or if it was the result of a planned change. The cause type can also signal that the interruption was the result of a measurement error caused by one of the underlying monitoring tools. Lastly, each interruption also includes the date on which it took place.

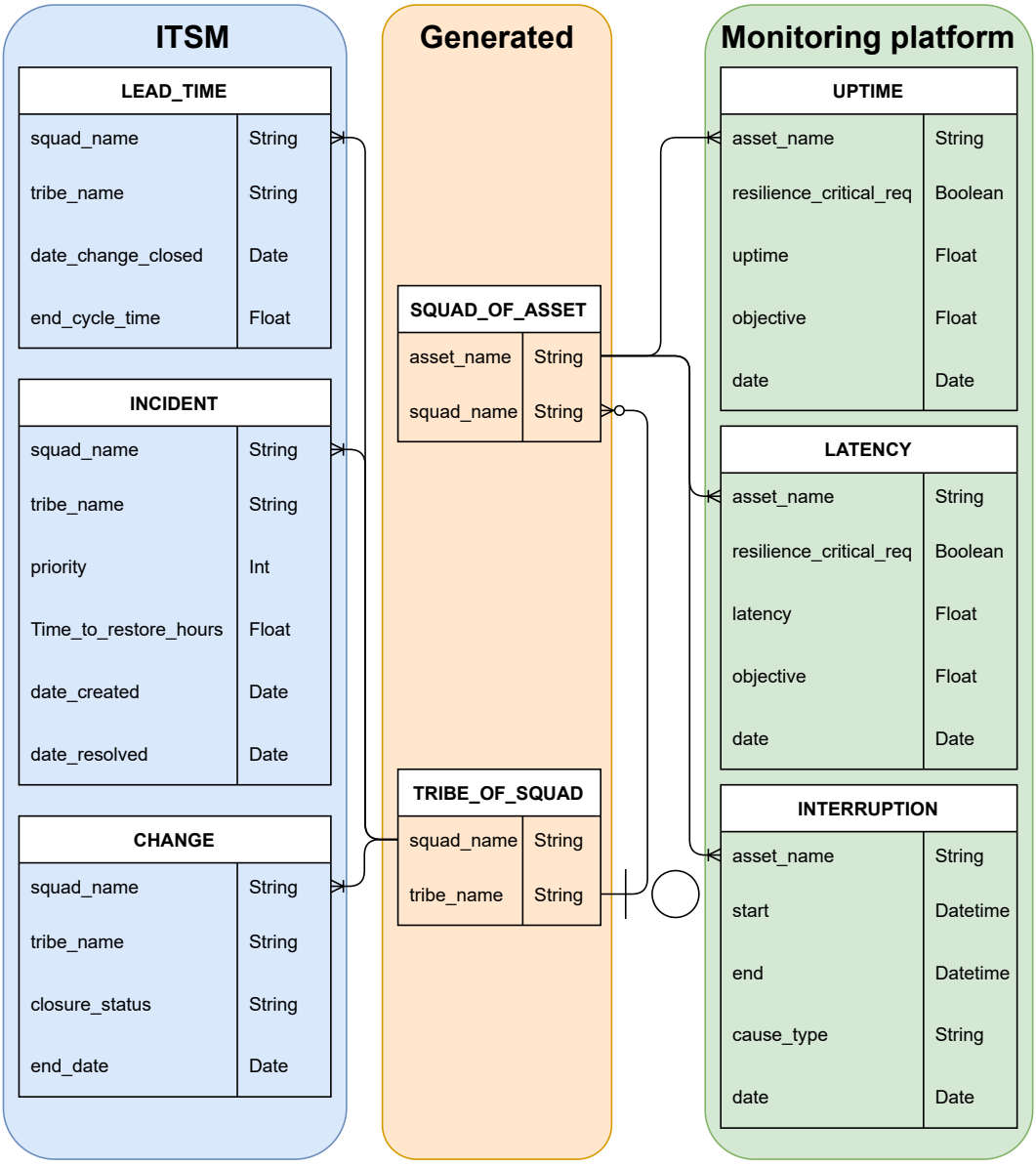

Figure 4.3: Entity-relationship diagram from the available data sources and the platform they are obtained from. The entities in the middle are generated and serve to map assets to squads and squads to tribes.

4.1.5 Calculation of metrics from available data sets

Previous sections have introduced the metrics that are measured by ING and described the available data sets. This section describes how the metrics are calculated from the available data sets as described in Figure 4.3. All metrics are aggregated per squad per month. The level of aggregation of the squads has been chosen as this was the smallest organisational unit that was available. The metrics are aggregated per month as this was the smallest possible granularity at which some of the metrics were defined. For most of the metrics, the median metric per month is calculated. In the initial phase of this thesis, there have been experiments using the mean instead of the median. Using the mean resulted in a much smaller amount of significant correlations than when using the median. Thus, the decision has been made to use the median when possible. This decision means that some of the metrics used in this thesis differ slightly from their official definition. For example, the mean time to restore (MTTR) has in this thesis become the median time to restore.

Calculation of ING-specific DORA metrics

This subsection describes how the DORA metrics are calculated according to the ING-specific definitions of those metrics as described before.

Deployment frequency This metric is calculated from the data set with changes. The ***deployment frequency*** per squad per month is calculated by grouping the changes by those two columns and counting the number of entries in each group. This metric only considers changes that were successful or partially successful according to their closure code. One data point consists of the squad name, tribe name, month and the number of changes performed by that squad in that month.

Change failure rate The input to this metric is the same as for the ***deployment frequency***. Again, the changes are grouped by squad and month. To compute the ***change failure rate***, the number of changes that were partially successful or had failed is divided by the total number of changes in the group. Thus, if two changes are performed by a squad in a month and only one is rolled out successfully, the change failure rate becomes 0.5. One data point consists of the squad name, tribe name, month and the ***change failure rate*** of that squad in that month.

Incident MTTR This metric is calculated from the data set containing the incidents. For reasons that will be described later, this thesis uses two ways of grouping the incidents when calculating the ***incident MTTR***. The first approach aggregates all incidents regardless of their priority. The incidents are grouped by their squad and the month of their start date. For each group, the median MTTR in seconds of the incidents is calculated. Thus, one data point consists of the squad name, tribe name, month and the median MTTR of the incidents of that squad in that month. The second approach aggregates incidents for each priority individually. In this case, the incidents are grouped by their priority as well. This results

in data points that consist of the squad name, tribe name, priority, month and the median MTTR of the incidents of that priority for that squad in that month.

Lead time for changes This metric is calculated from the data set containing lead times. The objects are grouped by their squad name and the month of the change. For each group, the mean lead time is calculated. Note that in contrast to many of the other metrics, the mean is used instead of the median. This is done as ING uses the mean as a proxy instead of the median. One data point consists of the squad name, tribe name, month and the mean ***lead time for changes*** of that squad in that month.

Metric	Calculated from	Nr. of datapoints after aggregation
Incident MTTR	Incidents	8131
Deployment frequency	Changes	8985
Change failure rate	Changes	8985
Lead time for changes	Cycle time	2945

Table 4.3: Descriptive statistics for the generated DORA metrics.

Descriptive statistics Table 4.3 provides the descriptive statistics of the generated DORA metrics, indicating the source of each metric and the number of data points after aggregation. The number of data points before aggregation is the same as the number of data points in the used data set as described in Table 4.2. The number of data points for the ***number of changes*** and the ***change failure rate*** after aggregation is the same, as each aggregated data point in the former metric automatically results in a data point in the latter. Aggregation does not affect the number of unique squads or tribes from the used data set.

Calculation of ING-specific metrics

Similar to the calculation of the DORA metrics above, this section provides the calculations of the metrics that either have been or are in use by ING. Where the previous section described the DORA metrics as used by ING, this section describes the metrics that are extracted from the interviews.

Normalised latency objective This metric uses the data set of normalised latency and requires little preparation as the data is already measured per asset per month. Assets are mapped to squads using the intermediate table described in Figure 4.3. Only data points with a positive latency objective are considered. Next, the data points are grouped by squad and month. For each group, the median ***normalised latency objective*** is calculated. One data point consists of the name of the squad, the month and the median ***normalised latency objective*** of that squad in that month.

Uptime The input to this metric consists of the uptime data set and requires little preparation as the data is already measured per asset per month. Assets are mapped to squads using the intermediate table described in Figure 4.3. Only data points for which both the uptime and uptime objective are between the bounds of 0 and 100 are considered. The inputs are grouped by the squad and month. For each group, the median ***uptime*** is calculated. One data point consists of the name of the squad, the month and the median ***uptime*** of that squad in that month.

Uptime resilience critical The input for this metric is the same as the input for the uptime metric and the same filtering applies. The inputs are grouped by the squad and month. For each group, the mean ***uptime resilience critical*** metric is calculated by calculating the mean of the "resilience critical" fields of the aggregated assets. In all following figures, this metric has been marked with an asterisk to indicate that this metric uses the mean instead of the median. Would the median have been used, this metric would only have reflected whether more than half of the assets in a group were resilience critical or not. One data point consists of the name of the squad, the month and the mean ***uptime resilience critical*** metric of that squad in that month.

The observant reader might wonder why this metric is only calculated for the uptime and not for the normalised latency, as both data sets include the information around resilience critical assets. Since this feature is defined at the level at the asset and many assets occur in both data sets, the metrics are nearly identical. Thus, the "normalised latency resilience critical" metric has been omitted for brevity.

Uptime objective This metric also uses the uptime data set as input. Assets are mapped to squads using the intermediate table described in Figure 4.3. Only data points for which both the uptime and uptime objective are between the bounds of 0 and 100 are considered. Data points are grouped by their squad and month and the median ***uptime objective*** is calculated for each group. One data point consists of the name of the squad, the month and the median ***uptime objective*** of that squad in that month.

Uptime objective achieved The input to this metric consists of the data set with uptime data. Assets are mapped to squads using the intermediate table described in Figure 4.3. Only data points for which both the uptime and uptime objective are between the bounds of 0 and 100 are considered. Each data point is extended with the value 1 if the uptime for that asset in that month is at least as large as the objective and 0 otherwise. The enhanced inputs are then grouped by their squad and month. For each group, the mean ***uptime objective achieved*** metric is calculated. The reasoning for using the mean instead of the median is similar to that of the ***uptime resilience critical*** metric. Similar to the ***Uptime resilience critical*** metric, this metric is annotated with an asterisk in the figures of this thesis. One data point consists of the name of the squad, the month and the mean ***uptime objective achieved*** of that squad in that month.

The observant reader might have noticed that the ***uptime objective achieved*** is part of this thesis, but the "standardised latency objective achieved" is not. As described before,

there was too little data on the standardised latency and thus this could not be used to calculate the ”standardised latency objective achieved” metric.

Uptime margin The data set containing the uptime data has been used to calculate this metric and the same filtering rules apply as mentioned before. For each data point, the margin is calculated by subtracting the uptime objective from the uptime of that asset in that month. The extended data points are grouped by their month and squad and the median ***uptime margin*** is calculated for each group. One data point consists of the name of the squad, the month and the median ***uptime margin*** of that squad in that month.

Number of incidents The metric is calculated using the data set with the incidents. Similar to the calculation of the ***incident MTTR***, two ways have been used to calculate this metric. The first one groups the incidents by squad and month and counts all incidents in each group. This results in data points that consist of the squad name, the month and the ***number of incidents*** of that squad in that month. The second one groups the incidents by squad, month and priority and then counts all incidents in each group. This results in data points that consist of the squad name, the priority, the month and the ***number of incidents*** of that priority of that squad in that month.

Number of interruptions The data set containing the interruption is used to calculate this metric. Duplicate interruptions were removed and the remaining interruptions were mapped to a squad using the intermediate table described in Figure 4.3. Interruptions that could not be mapped to a squad were removed. Interruptions that were the result of a measurement error were removed, this value was contained in the ”cause type” field of the interruption data. After this processing, the remaining interruptions were grouped by the squad and month and the number of interruptions in each group was counted. One data point consists of the name of the squad, the month and the ***number of interruptions*** of that squad in that month.

Interruption MTTR The input to this metric is the same as for the metric around the ***number of interruptions*** and the same processing steps apply. For each input, the MTTR is calculated by subtracting the start time from the end time and converting the resulting interval into seconds. The input is then grouped by the squad and month. For each group, the median ***interruption MTTR*** is calculated. One data point consists of the name of the squad, the month and the median ***interruption MTTR*** in seconds of that squad in that month.

Interruption MTTF This metric leverages the extended input of the ***interruption MTTR*** metric as its own input. The MTTF for each input is calculated by first calculating the duration between the end time of the interruption and the previous interruption of the same squad, followed by the subtraction of the MTTR of the incident. This results in the time between the end of the previous incident and the start of the current one. The calculated results are grouped by squad and month. For each group, the median MTTF is calculated.

One data point consists of the name of the squad, the month and the median ***interruption MTTF*** in seconds of that squad in that month.

Metric	Calculated from	Nr. of datapoints after aggregation
Normalised latency	Latency	506
Uptime resilience critical*	Uptime	511
Uptime objective achieved*	Uptime	511
Uptime margin	Uptime	511
Uptime objective	Uptime	511
Uptime	Uptime	511
Nr. of incidents	Incidents	8131
Nr. of interruptions	Interruptions	523
Interruption MTTF	Interruptions	480
Interruption MTTR	Interruptions	523

Table 4.4: Descriptive statistics for the generated ING metrics. Metrics with an asterisk have been aggregated using the mean.

Descriptive statistics The descriptive statistics of the generated ING metrics have been depicted in Table 4.4. This table shows the data set used to calculate each metric and the number of data points after aggregation. The number of data points before aggregation is the same as the number of data points in the used data set as described in Table 4.2. Aggregation does not affect the number of unique squads or tribes from the used data set.

4.2 RQ3: Relationship between DORA and ING metrics

Now that the available data and metrics have been described, the last three research questions can be answered. The first remaining research question aims to understand the relation between the DORA metrics and the ING-specific metrics. These insights are generated in multiple steps. First, the relationship between the DORA metrics within ING is investigated. Next, the correlations between the ING metrics themselves and the DORA metrics are investigated to evaluate how the values of those metrics relate. In the final step, the relationships between the derivatives of those metrics are investigated to generate insight into how they change in relationship to one another over time. Thus, the third research question has been broken up into three sub-questions, as depicted below.

Specification of research questions

RQ3 How do the four metrics as identified by DORA [16] relate to the other metrics used by ING?

- **RQ3.1** How do the four metrics as identified by DORA[16] relate to each other within ING?
- **RQ3.2** How do the ING-specific metrics correlate with themselves and with the DORA metrics?
- **RQ3.3** How do the derivatives of the ING-specific metrics correlate with themselves and with the derivatives of the DORA metrics?

4.2.1 RQ3.1 Independence of DORA metrics

Previous sections have introduced the used data sets and have indicated how ING measures the DORA metrics. The DORA report by Forsgren et al. [16] claims that the four metrics they use measure four aspects of SDO Performance. This claim would imply that the four metrics are independent of each other. After all, if two metrics are dependent on one another, measuring only one of the two could be sufficient.

As discussed in the introduction, measuring more metrics than required is not desired due to the cost of measuring metrics and the risk of causing an information overload. Thus, it is important to know that these four metrics are independent. Furthermore, establishing that the proxies that ING uses to approximate the DORA metrics are independent would strengthen the confidence in those proxies. This independence is investigated in two different ways. First, the DORA metrics are analysed by performing *Principal Component Analysis* (PCA) [18]. PCA aims to reduce the dimensionality of the input data while minimising the amount of information that is lost by performing this reduction. If the DORA metrics, which have four dimensions due to being composed of four metrics, can be projected into a space of three dimensions without losing a significant amount of information, this would mean that the four metrics are not independent and that only three metrics would suffice. Secondly, a multiple linear regressor [17] is used to predict each one of the DORA metrics from the other three. The regressor aims to find a linear combination of the three

other metrics that best approximates the fourth metric. This thesis hypothesises that if that is possible for any of the metrics, it is redundant and can be replaced by the linear combination of the other three metrics.

This section first describes the steps that were taken to prepare the DORA metrics for further analysis. Then, it describes the procedure and results of performing principal component analysis and finally, it describes the use of a linear regressor on the DORA data set.

Preparation of metrics Before both types of analysis are performed, the four DORA metrics that have been described in Section 4.1.5 are joined using an inner join on the squad name and month. This results in a data set that only contains those combinations of squads and months that have a value for all four of the metrics. This filtering step is necessary as using all data points resulted in a data set that was too sparse to use in the following analysis steps. Therefore, the decision has been made to only include combinations of squad name and month that have a value for all four of the metrics.

Principal Component Analysis PCA is performed using the PCA module of scikit-learn[1]. Before the analysis is performed, the data is scaled using the StandardScaler of scikit-learn[2]. The PCA module is then used to create the decomposition of the data. This decomposition results in four extracted dimensions that are orthogonal to each other. The relative amount of explained variance is extracted from each component. If the four dimensions of the original data were not orthogonal, one of the components of the decomposition will explain significantly more variance than the others. On the other hand, if all four components explain roughly the same amount of variance, the original data was already more or less orthogonal.

Figure 4.4 depicts the relative amount of variance that each of the four extracted components explains. A pie chart has been selected for the visualisation because of the small dimensionality of the data. Often, a scree plot is used to depict the amount of variance that each principal component explains. However, as this thesis only has four dimensions, using a pie chart provided insight into the components in a way that was easier to understand. From this figure, it can be derived that, although there is one component that explains slightly more variance than the others, the metrics are more or less orthogonal. Selecting the three components that explain the most variance would mean that 17.9 per cent of variance would become unexplained. Therefore, it is concluded that the original metrics are orthogonal.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

[2] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

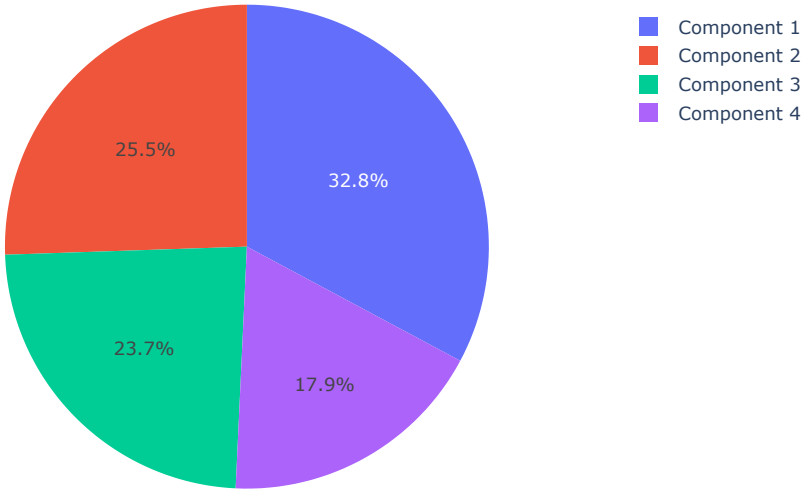

Figure 4.4: PCA analysis of the DORA metrics. This figure depicts the amount of variance that is explained by each of the dimensions created by the PCA.

Linear regression Linear regression was performed using the LinearRegression module from scikit-learn[3]. Four different models were trained, one for each one of the metrics. When training for one of the metrics, that metric was used as the dependent variable, while the other three were used as independent variables. For each of the models, the coefficient of determination R^2 was extracted. This coefficient indicates the percentage of variance in the dependent variable that can be explained using the independent variables and is bound by: $0 \leq R^2 \leq 1$.

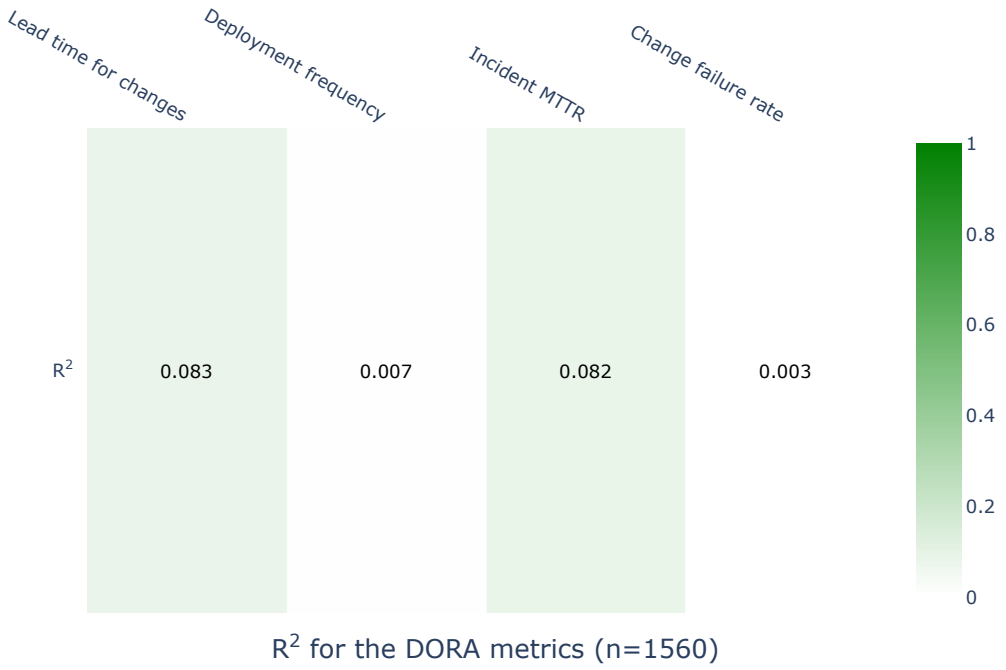

Figure 4.5: Coefficient of determination R^2 when using each of the DORA metrics in turn as the dependent variable and the others as independent variables while training a linear regression model.

Figure 4.5 depicts the result of the linear regression analysis. Each column in this figure depicts the R^2 value when that metric is used as the dependent variable. Using this heat map visualisation allows the four metrics to be contrasted with each other. The analysis indicated that at most 8.3 per cent of the variance could be explained when using the ***lead time for changes*** as the dependent variable. Given the small maximum R^2 value, it is concluded that it is not possible to predict one of the DORA metrics from the others.

[3] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

Findings RQ 3.1 Relation of DORA metrics within ING

This section has investigated if the four DORA metrics as measured by ING are independent and orthogonal to each other. Principal Component Analysis (PCA) has been used to establish that the four DORA metrics are more or less orthogonal to each other. Using a linear regressor, it has been shown that none of the four metrics is a linear combination of the other three metrics. Therefore it is concluded that the four DORA metrics as measured by ING are independent and orthogonal.

4.2.2 RQ3.2 Correlations between metrics

The previous section has established that the four DORA metrics as measured by ING are independent. This section investigates the correlation between the DORA metrics, the ING metrics and between the ING metrics and the four DORA metrics. The second investigation aims to get insights into how the ING metrics relate to each other, while the last investigation aims to get insights into how well the four DORA metrics can explain the ING metrics. The correlations are calculated between pairs of metrics. The metrics are joined on their squad and month using an inner join, such that only combinations of squads and months remain that have a value for both of the metrics. Subsequently, the Spearman correlation [48] and the p-value of the correlation is calculated using the stats module from Scipy[4]. For each pair of metrics, the number of entries in the joined table is also recorded.

Range of r	**Interpreted strength**
1.0	Perfect
$0.8 \leq r < 1.0$	Very strong
$0.6 \leq r < 0.8$	Moderate
$0.3 \leq r < 0.6$	Fair
$0.1 \leq r < 0.3$	Poor
$0.0 \leq r < 0.1$	None

Table 4.5: Interpreted strengths of Spearman correlation coefficient r. Adapted from Chan [8] and Akoglu [1]

In the analyses, correlations with a p-value of at most 0.05 will be considered to be significant. Table 4.5 has been adapted from from Chan [8] and Akoglu [1] and shows the interpreted strengths of the Spearman correlations. This thesis will only consider correlations that are at least fair according to this table to be relevant. Analysis later on in this chapter will show that many of the significant correlations are at most poor according to this table. Reporting on those correlations as well would mean that the results include many correlations with minor effect sizes. Therefore, the decision has been made to not report on poor correlations.

Table 4.4 and Table 4.3 show that 10 ING metrics and four DORA metrics have been collected for this thesis. This means that when investigating the correlations between all ING metrics, there are $10^2 = 100$ correlations to report, 55 of which are unique due to symmetry. Thus, the decision has been made to report the correlations using heat maps. Each row and column label depicts the name of the metric, together with the total amount of data points that were available for this metric. Each tile of the heat map has three rows, displaying the correlation strength r, the p-value of the correlation and the number of entries in the joined data set respectively from top to bottom. Tiles with bold text contain significant correlations ($p \leq 0.05$).

[4] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.spearmanr.html

Correlations between DORA metrics

Given the previous section, an investigation of the correlations between the DORA metrics might seem superfluous. Later sections will investigate the effect of ING's organisational structure on the relationship between the DORA metrics. To be able to perform this analysis, a baseline comparison is needed to show the effect of the biases. Thus, this analysis is superfluous on its own, but necessary in the context of this full thesis.

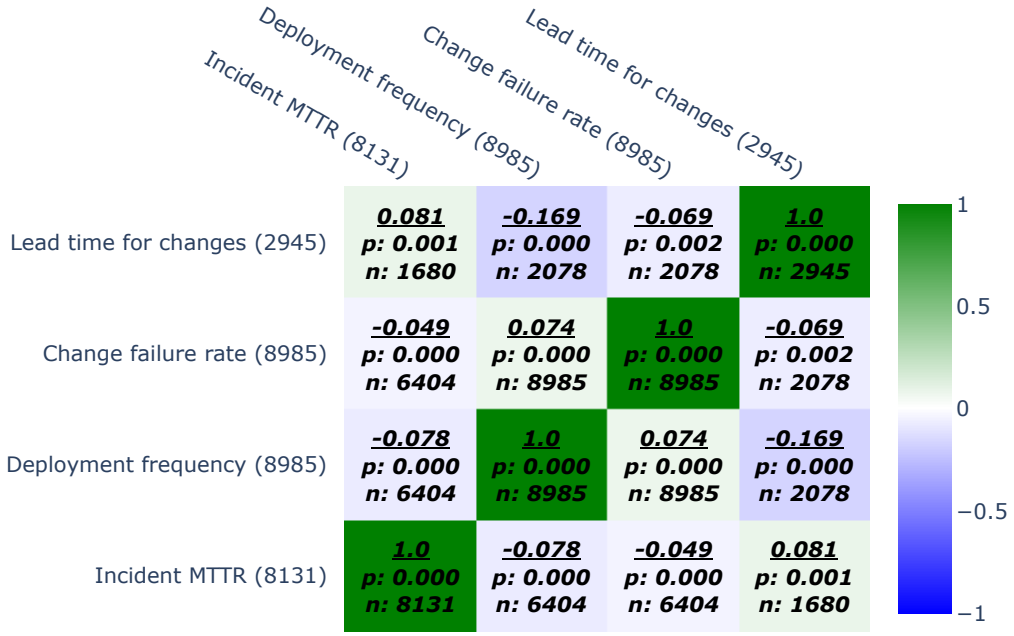

Figure 4.6: Pairwise Spearman correlation between the DORA metrics. Each square includes the correlation strength, the p-value and the number of data points used. Each metric is annotated with the total amount of data points that were available for this metric. Bold correlations have a p-value of ≤ 0.05.

To investigate the correlations between the DORA metrics, all four of them have been treated as described at the beginning of this section. This resulted in a four-by-four heat map containing the correlations between the DORA metrics. This heat map has been depicted in Figure 4.6. The pairwise inner join of metrics resulted in at least 1680 data points, occurring between the ***incident MTTR*** and the ***lead time for changes***. This figure shows the correlations between pairs of DORA metrics. Although all of the correlations are significant, none of them can be considered to be relevant, as the largest correlation (except self-correlations) has a strength of -0.169. This is the only correlation that can be considered poor according

to Table 4.5, all others are classified as being nonexistent. Thus, it can be concluded that there are no relevant correlations between the four DORA metrics as measured by ING.

Correlation between ING-specific metrics

This subsection investigates the correlations between the ING-specific metrics. Generating insights into the relationships between the metrics that are being measured can provide valuable information about how teams measure themselves and how the things they measure relate to each other. Such insights can help the organisation to make better-informed decisions about the metrics they want to collect, or stopped collecting. Furthermore, understanding these relationships is useful when investigating the relationship between the DORA metrics and the ING metrics in the next subsection. The investigation of the correlations between the ING-specific metrics follows the same methodology as explained at the beginning of this section. The metrics generated in Section 4.1.5 are pairwise joined using an inner join to create combinations of months and squads that have a value for both metrics of the selected pair. The correlations between the pairs of metrics have been depicted using a heat map, where each metric includes the number of data points, and each tile represents the correlation, the p-value and the number of data points used to calculate those between pairs of metrics.

Figure 4.7 depicts the correlations, p-values and sample sizes for the correlations between the ING metrics. The strongest negative correlation (except for self-correlations) can be observed between the ***interruption MTTF*** and the ***number of interruptions***. This correlation is very strong and significant. This negative correlation was expected, as having more interruptions must decrease the time between interruptions. The strongest positive correlation can be found between the ***uptime objective*** and the ***normalised latency objective***. From this observation, it follows that squads that set themselves high standards for standardised latency also set themselves high standards for uptime. Other pairs of metrics with high correlations can be found between the uptime-related metrics. Another observation is that the ***interruption MTTR*** has a fair but significant negative correlation with most of the other metrics, except for the ***interruption MTTF***. Thus, squads that have larger ***interruption MTTR*** are slightly more likely to have fewer interruptions and more time between interruptions.

The interviews in Chapter 3 have indicated that the ***number of incidents*** has been used as a proxy for the availability. This analysis shows that it does not correlate with availability. It only has a fair correlation with the ***number of interruptions***. Section 4.4 will explore this correlation further. An interesting observation to make is that the ***uptime resilience critical*** metric does not have correlations with any of the other metrics. In other words, squads that work on more resilience critical assets do not have higher uptime, uptime objectives, or an inclination towards achieving their latency goals. The correlations that the ***normalised latency objective*** has with the other metrics indicate that squads that have higher normalised latency objectives tend to have higher uptimes and uptime objectives and are more likely to achieve their uptime objectives. They also tend to need slightly less time on average to recover from an interruption, as indicated by the negative correlation with the ***interruption MTTR***.

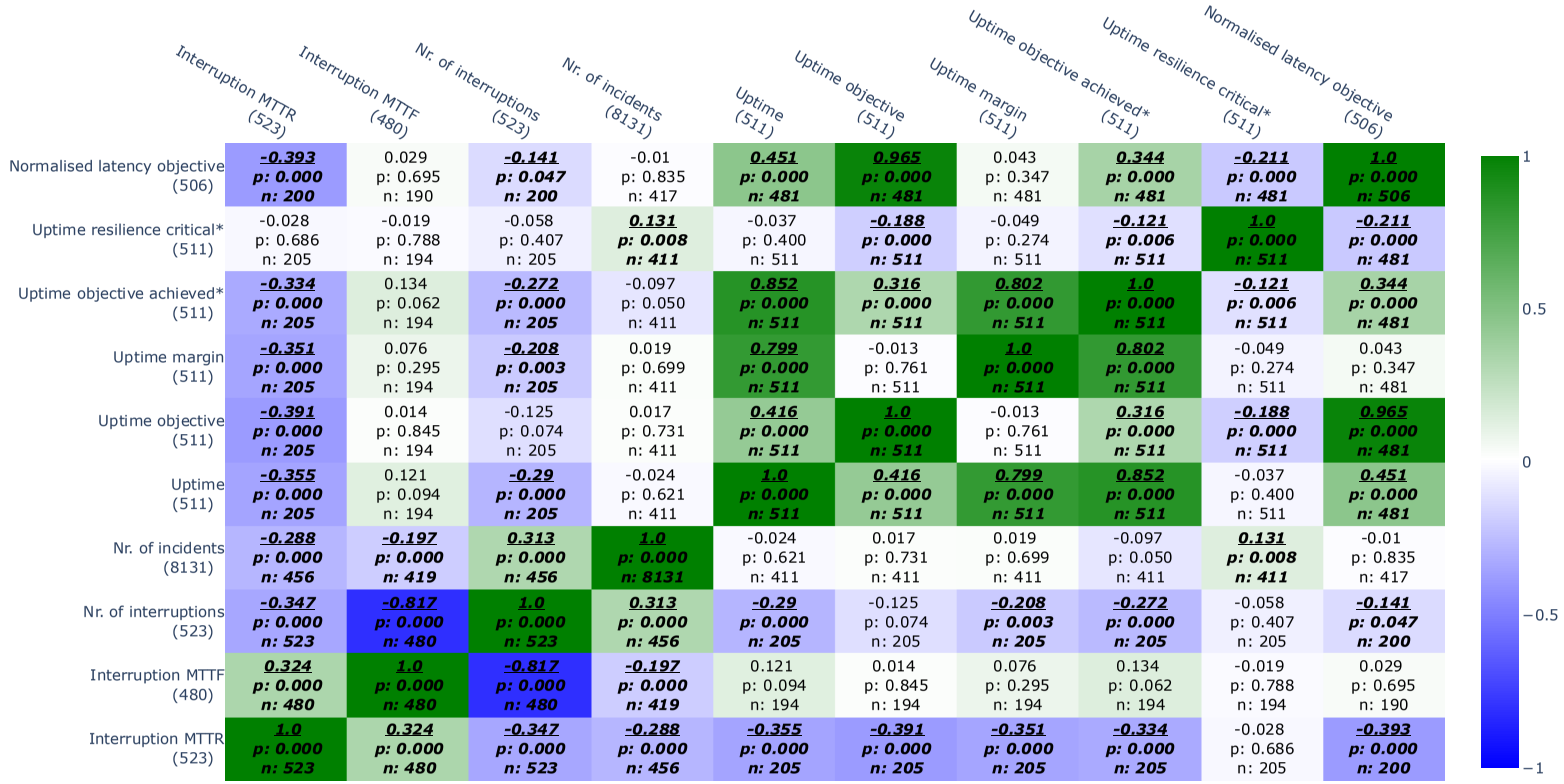

Figure 4.7: Pairwise Spearman correlation between the ING metrics. Each square includes the correlation strength, the p-value and the number of data points used. Each metric is annotated with the total amount of data points that were available for this metric. Bold correlations have a p-value of ≤ 0.05. Metrics with an asterisk have been aggregated using the mean.

Correlations between DORA and ING metrics

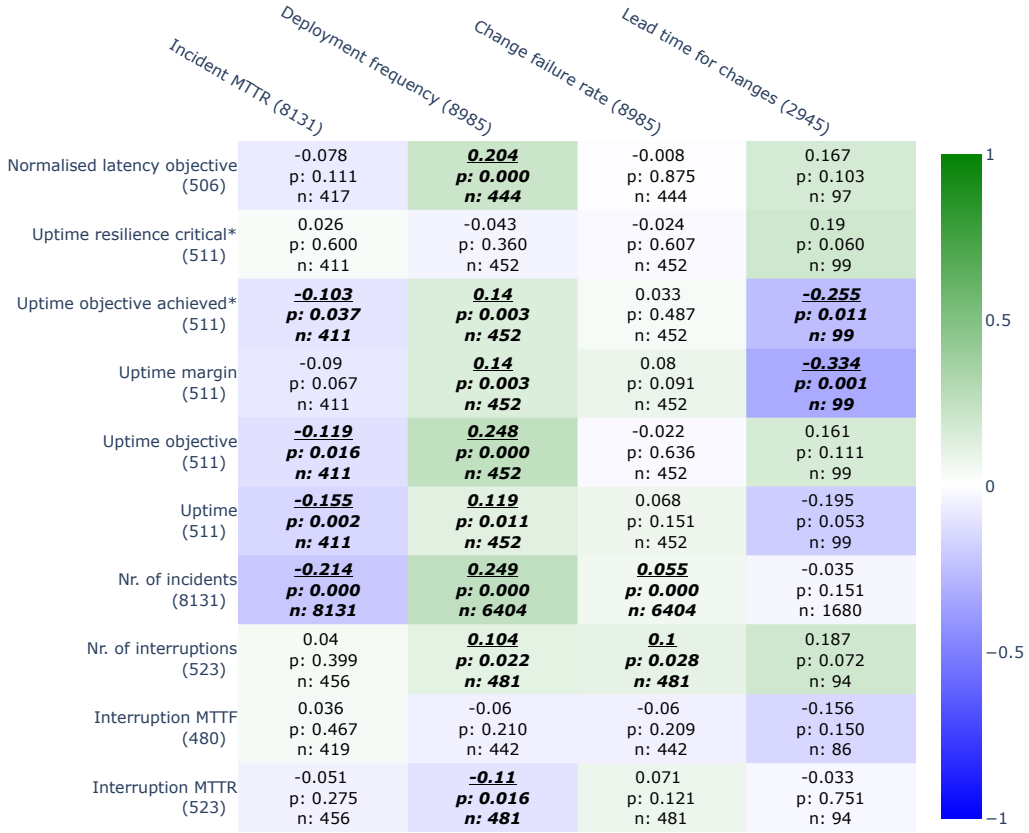

Figure 4.8: Pairwise Spearman correlation between the DORA metrics and ING metrics. Each square includes the correlation strength, the p-value and the number of data points used. Each metric is annotated with the total amount of data points that were available for this metric. Bold correlations have a p-value of ≤ 0.05. Metrics with an asterisk have been aggregated using the mean.

Previous analyses have generated insights into the independence of the DORA metrics and the correlations between the ING metrics. The following analysis aims to combine those steps by investigating the correlations between the DORA metrics and the ING metrics. Gaining insight into the relationship between those two sets of metrics can provide valuable information for future researchers by showing how the industry-standard set of DORA metrics relates to a set of metrics that are collected within an organisation to fit their specific needs. The DORA report states that their metrics relate to stability and throughput, so investigating their relationship to the metrics used for indicating similar themes within

ING can shed light on how generalizable those metrics are. Furthermore, as ING uses proxies for the DORA metrics that can be calculated on an organisational scale, gaining insights into how these proxies relate to comparable metrics that are being measured directly can enable informed decision making about the metrics that should be collected. Similar to previous investigations of the correlations between metrics, pairs of metrics are joined using an inner join on their month and squad to create data points that include both metrics for a given combination of squad and month. The results are reported using a heat map, where each tile includes the strength of the correlation, the p-value of that correlation and the number of data points used to calculate it.

Figure 4.8 shows the results of this analysis. Although the figure shows that there are a large number of significant correlations, there is only one correlation that is both significant and has an effect size of at least 0.3. This correlation is between the ***lead time for changes*** and the ***uptime margin*** and is negative. Thus, squads that have a larger lead time for changes in a given month have a smaller margin between their uptime objective and the uptime they achieved.

Findings RQ 3.2 Correlations between DORA and ING metrics

This section has shown that there are no correlations between the DORA metrics and that there are several correlations between the ING metrics, most of which can be explained by the definitions of the metrics. Most importantly, the ***interruption MTTR*** has small negative correlations with other metrics, the ***number of incidents*** and ***number of interruptions*** have a fair correlation and ***the standardised latency objective*** correlates with the ***uptime***, ***uptime objective*** and ***uptime objective achieved*** metrics. In contrast to those findings, it has been established that the DORA metrics have only one correlation with the ING metrics. This correlation is fair and appears between the ***lead time for changes*** and the ***uptime margin***.

4.2.3 RQ3.3 Correlations between derivatives of metrics

The previous section has shown that there is a small number of correlations between the ING metrics that can not be explained by the definitions of those metrics. It has also shown that the DORA metrics have very little correlation to any of the metrics. These analyses only investigated the relationships between the actual values of the metrics. In this sub-question, the notion of time will be introduced into the relationships between the metrics. The previous analyses only investigated the actual values of the metrics in a given month, without taking the order of months into account. The analysis of this sub-question investigates how the pairs of metrics change in relationship to each other from one month to the next. Introducing this notion creates a more complete picture of the relationships between metrics.

The methodology of this process is largely the same as that of establishing the correlations between metrics. Instead of using the inputs directly, the inputs are first grouped by their squad only. For each squad, this results in a list of months for which data points are available. Months that are missing and that are between the minimum and maximum month

of a squad are linearly interpolated. In other words, if a squad has a metric for month 3 and month 6, months 4 and 5 are assumed to have values that are the linear interpolation between months 3 and 6. After all months between the minimum and maximum month for a squad have a value for the metric, the derivative of the metric is calculated by subtracting the value of the previous month from the following one. Figure 4.9 provides a visualisation of this process. The interpolation of months has been implemented to aid in the calculation of the derivative. Would there have been no interpolation and would the derivative have been calculated over the whole gap between months, the later data point would have had a too-large derivative. Would the average of the gap have been used as the derivative, a part of the change in the metric would have been lost. Once the derivatives of the metrics have been established, performing the correlation study between them followed the same procedure as explained in the previous section.

As the procedure of this section is largely the same as that of Subsection 4.2.2, this section will use the same interpretation of Spearman's *r* value as described in Table 4.5 and use the same structure and reasoning when it comes to reporting the results of the analyses. As was the case in the previous section, this section will again report on the 100 correlations between the ING metrics and thus leverage the expressive power of heat maps as described before.

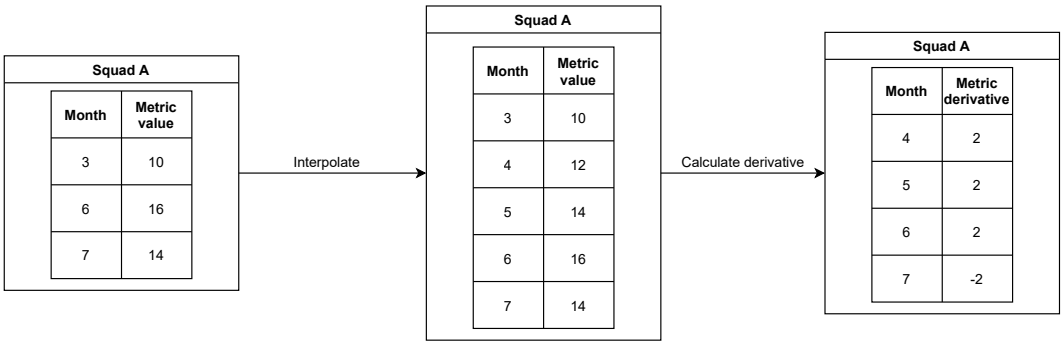

Figure 4.9: Visualisation of the process of interpolation and calculation of the derivative for a fictional squad A

Correlations between derivatives of DORA metrics

Subsection 4.2.1 has established that the DORA metrics as measured by ING are independent. This subsection investigates how the DORA metrics change in relation to each other. Given that the metrics are independent, the hypothesis is that there will be no significant correlations between the derivatives of the metrics.

Figure 4.10 depicts the results of this analysis. This figure shows that, except for self-correlations, there are two types of correlations between the derivatives of the DORA metrics. The first type adheres to the hypothesis, as those correlations have large p-values, indicating that the correlations are not significant. On the other hand, the second type of correlation has a small p-value, but a near-zero correlation strength. In conclusion, the

derivatives of the DORA metrics do not show significant correlations with noteworthy effect sizes.

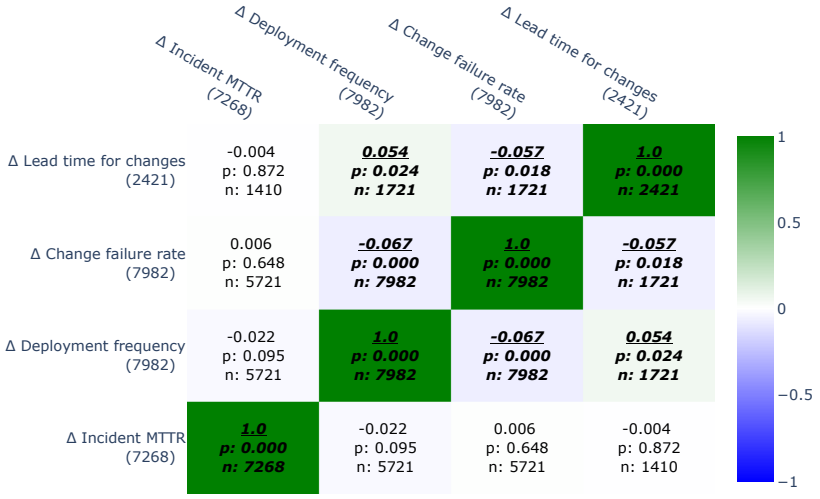

Figure 4.10: Pairwise Spearman correlation between the derivatives of the DORA metrics. Each square includes the correlation strength, the p-value and the number of data points used. Each metric is annotated with the total amount of data points that were available for this metric. Bold correlations have a p-value of ≤ 0.05.

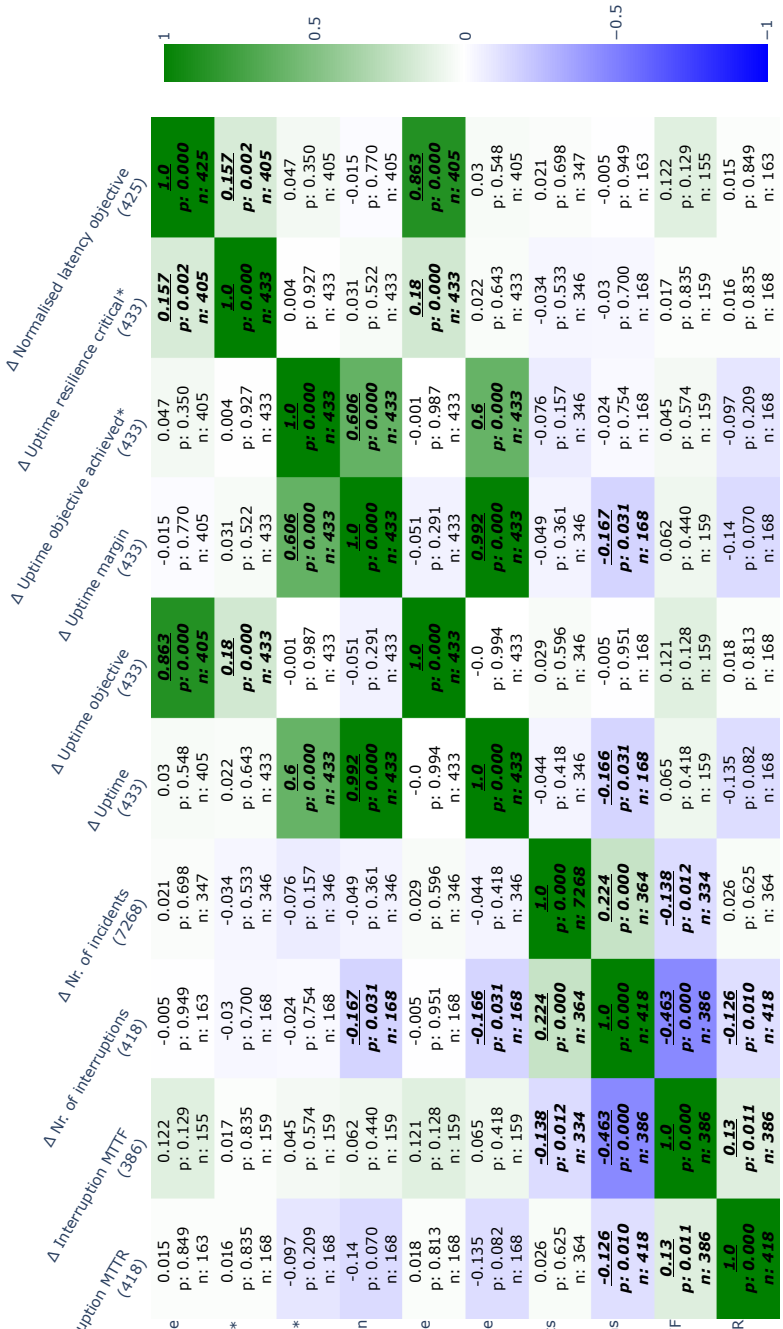

Figure 4.11: Pairwise Spearman correlation between the derivatives of ING metrics. Each square includes the correlation strength, the p-value and the number of data points used. Each metric is annotated with the total amount of data points that were available for this metric. Bold correlations have a p-value of ≤ 0.05. Metrics with an asterisk have been aggregated using the mean.

Correlations between ING-specific metrics

Given the findings from the previous subsection and the findings from the previous section that there are only a small number of ING metrics that correlate together for which the definition of the metric does not provide an answer, it is interesting to investigate how the DORA metrics change together.

Figure 4.11 depicts the correlations, p-values and sample sizes for the study of the derivatives of the ING metrics. The fair correlations of the ***interruption MTTR*** that were observed in Figure 4.7 have disappeared in this experiment, indicating that although ***interruption MTTR*** correlates fairly with the other metrics, having a change in the median ***interruption MTTR*** does not mean that the other metrics change as well. The earlier analysis of the ING metrics showed that the ***interruption MTTF*** correlated strongly with the ***number of interruptions***, as was hypothesised from the definition of the metrics. The current analysis strengthens this observation, as this correlation is also present between the derivatives of the two metrics. These results show that the ***uptime*** correlates positively with the ***uptime objective achieved*** and ***uptime margin*** metrics. They also show that the ***uptime objective achieved*** and ***uptime margin*** metrics correlate with each other. Finally, there is a very strong and significant correlation between the ***normalised latency objective*** and the ***uptime objective***, indicating that squads who increase or decrease one of those objectives also modify the other one in the same direction.

Correlations between DORA and ING metrics

In the previous analyses, it has been shown that there are no correlations between the derivatives of the DORA metrics and that there are only a few correlations between the ING metrics, most of which could be derived from the definitions of the involved metrics. The current analysis extends those findings by investigating the correlations between the derivatives of the DORA metrics and the ING-specific metrics. Given that the DORA metrics have no correlations between their derivatives and the ING metrics have them only sporadically, the hypothesis for this analysis is that there are few to no correlations.

This hypothesis is confirmed by the results shown in Figure 4.12. As was the case when analysing the correlations between the derivatives of the DORA metrics, there are mostly correlations that are not significant or with a near-zero effect. This figure shows three empty squares in the column of the ***lead time for changes***. For those correlations, there were respectively 80, 82 and 82 data points that resulted from the inner join of the data sets. As the derivatives of these data points were constant, no correlation could be calculated. Thus, the squares of those pairs of metrics are empty.

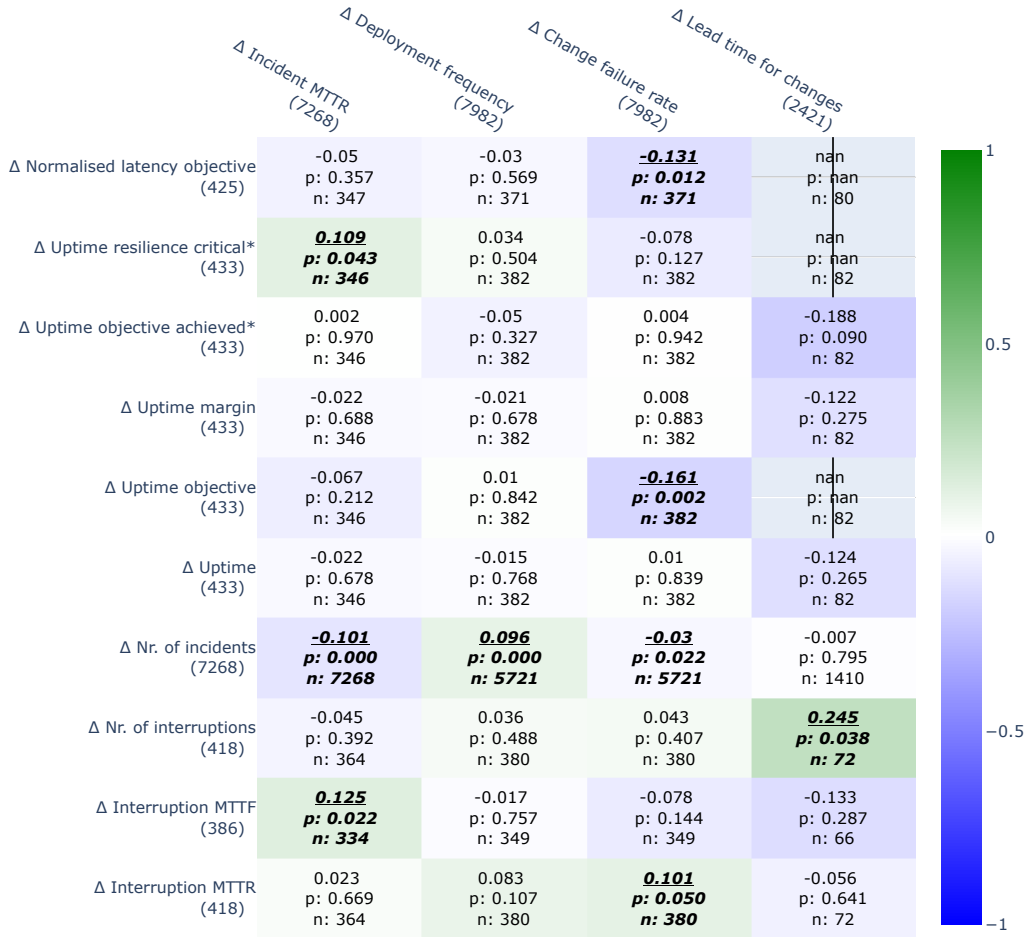

Figure 4.12: Pairwise Spearman correlation between the derivatives of the DORA and ING metrics. Each square includes the correlation strength, the p-value and the number of data points used. Each metric is annotated with the total amount of data points that were available for this metric. Bold correlations have a p-value of ≤ 0.05. Metrics with an asterisk have been aggregated using the mean.

Findings RQ 3.3 Correlations between the derivatives of metrics

This subsection has investigated the correlations between derivatives of the DORA metrics, the ING metrics and between the DORA and ING metrics. It has been established that there are no correlations between the derivatives of the DORA metrics and that there are only a few between the derivatives of the ING metrics. Most of the found correlations for the ING metrics could be brought back from the definitions of the metrics. However, it was found that squads that modify their objective for the uptime often also modify their objective for the normalised latency in the same direction. When analysing the correlations between the DORA metrics and the ING metrics, it was established that there are no correlations between them, indicating that the metrics change independently from each other.

4.2.4 Conclusion

Firstly, Principal Component Analysis (PCA) has been used to establish that the four DORA metrics are more or less orthogonal to each other. Using a linear regressor, it has been shown that none of the four metrics is a linear combination of the other three metrics. Therefore it is concluded that the four DORA metrics as measured by ING are independent and orthogonal.

The Spearman correlation coefficient has been calculated and with a significance level α of 0.05, it has been established that there are no correlations between the DORA metrics. Furthermore, it has been established that there are several correlations between the ING metrics, most of which can be explained by the definitions of the metrics. Finally, it has been established that the DORA metrics have only one correlation with the ING metrics, his correlation has a fair strength.

To complement these findings, the correlations between derivatives of the DORA metrics, the ING metrics and between the DORA and ING metrics have been investigated. It has been established that there are no correlations between the derivatives of the DORA metrics and that there are only a few between the derivatives of the ING metrics. Most of the found correlations for the ING metrics could be explained by the definitions of the metrics. Finally, it was established that there are no correlations between the derivatives of the DORA metrics and the derivatives of the ING metrics.

4.3 RQ4: Influence of organisational structure on the relationships between DORA and ING-specific metrics

The previous research question focused on the relationships between metrics in terms of their direct values and derivatives. Several correlations were found between metrics, some of which could be explained by the definition of the involved metrics. During the analysis, the hypothesis was raised that this could be caused by biases in the use of metrics. During the interviews in Chapter 3, it had already been mentioned that the users of the monitoring platform use the platform differently and it was hypothesised that this could also be the case for the other metrics. Understanding which correlations are the result of biases and which

ones aren't can help in establishing a more complete picture of the behaviour of metrics within an organisation. Metrics that correlate after the removal of biases are intrinsically related, while metrics that correlated before bias removal but no longer do afterwards are likely to be correlated by the way people use them or how important people find them. This research question aims to investigate if the aforementioned hypothesis is indeed true. As a first step, the metrics are analysed to establish whether they contain a bias or not. If it has been established that a bias is present, the effect of this bias on the results of RQ3 will be investigated. To this end, the fourth research question has been broken up into two sub-questions.

Specification of research questions

RQ4 What is the influence of organisational structure on the relationships between the DORA and ING metrics?
RQ4.1 Do the metrics contain biases?
RQ4.2 What is the effect of biases on the correlations found in RQ3?

4.3.1 RQ4.1 Existence of tribe biases in used metrics

This sub-question investigates if there is a bias present in the DORA and ING metrics. To do so, the bias is first formally defined before an analysis strategy is introduced to investigate if the biases are present. The biases were defined at the level of the tribe. Defining the biases at the squad level would have made the analysis more sound, but as each squad has at most 12 data points for each metric, this would have resulted in data sets that are too small for the chosen approach. The definition of the biases requires that several notations are introduced first.

In this section, $m_{uncorrected}$ is used to denote the *uncorrected metric* and $m_{corrected}$ to denote the *corrected metric*. $\bar{m}_{global}$ denotes the mean or median of the metric over the whole data set. Whether this notation denotes the mean or median depends on the metric that is being evaluated. Subsection 4.1.5 describes which aggregate is used for each metric. Furthermore, let $\bar{m}_{tribe}$ denote the mean or median of the metric when considering a specific tribe. This value is obtained by computing the mean or median of all data points of a specific tribe for that metric. Finally, let b_{tribe} denote the bias of a tribe for a given metric.

This section assumes that the uncorrected metric is composed of the corrected metric, plus a tribe-based bias: $m_{uncorrected} = m_{corrected} + b_{tribe}$. Furthermore, it is assumed that $\bar{m}_{tribe}$ is composed of the mean or median of the metric, plus the tribe-based bias: $\bar{m}_{tribe} = \bar{m}_{global} + b_{tribe}$. This definition of biases is based on baseline predictions as described by Koren [30]. These baseline predictions are described in the context of users and items. The goal of the baseline predictions is to predict how a user is going to rate an item. The baseline is computed by summing the global mean rating, the deviation of the user from the global mean user rating and the deviation of the item from the global mean item rating. In the context of this thesis, the same strategy is applied backwards. Instead of creating a baseline

prediction that includes the biases of users and items, the corrected metric is computed by accounting for the *tribe-based biases*. The strategy to assert the presence of tribe-based biases assumes that, if there were tribe-based biases present in the metrics, it would be harder to predict the tribe of a data point after the tribe-based biases had been removed. To this end, the data sets were first processed to remove the biases, after which a *Support Vector Machine* (SVM) [5] was trained on both the processed and unprocessed data sets.

Using the definitions provided earlier, the biases can be removed from the data as follows:

$m_{corrected} = m_{uncorrected} - b_{tribe}$
$b_{tribe} = \bar{m}_{tribe} - \bar{m}_{global}$
$m_{corrected} = m_{uncorrected} - (\bar{m}_{tribe} - \bar{m}_{global})$

First, $\bar{m}_{global}$ was calculated over the full data set of a metric. To compute $\bar{m}_{tribe}$ for each tribe, the data points were grouped by their tribe and the mean or median is calculated. In the final step, the bias in each data point was removed by subtracting the difference between the global mean or median of the metric and the mean or median of the respective tribe for that metric. This processing step resulted in two variants of the data sets of each metric: one with the biases present and one with the biases removed. To assert the presence of the biases, an SVM was trained on both variants of the data. The data is first scaled using the quantile transformer of sklearn[5]. After the data had been scaled, it was split into 10 stratified folds. The training set of each fold was oversampled using the random oversampler from imblearn[6]. The training data was then used to train an SVM from sklearn[7] with polynomial kernel and a maximum of 1000 iterations. The trained model was then used to predict the tribes of the test set of the current fold. To measure how well the model performs, the F1 score [47] on the test set was calculated. By running this analysis on both variants of each data set, the F1 score for the regression was obtained for when a model had access to the biases and when these had been removed.

Given that stratified 10-fold sampling was used, this procedure resulted in two data sets with 10 entries each, where each entry was the F1 score of a run of an SVM. Box plots have been used to visualise the results of the experiments. This strategy of visualisation allows displaying not only the median F1 score of the two variants of the data set but also the distribution of the scores in terms of their quartiles. By displaying the boxplots of the data set with and without the biases next to each other, insight is created into how the removal of the bias affects the ability of the SVM to predict the tribe.

[5] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.QuantileTransformer.html

[6] https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.RandomOverSampler.html

[7] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

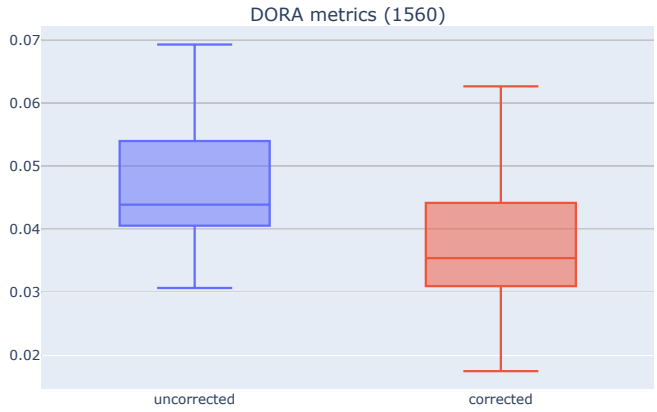

Figure 4.13: F1 scores of 10-fold stratified cross-validation on corrected and uncorrected DORA metrics.

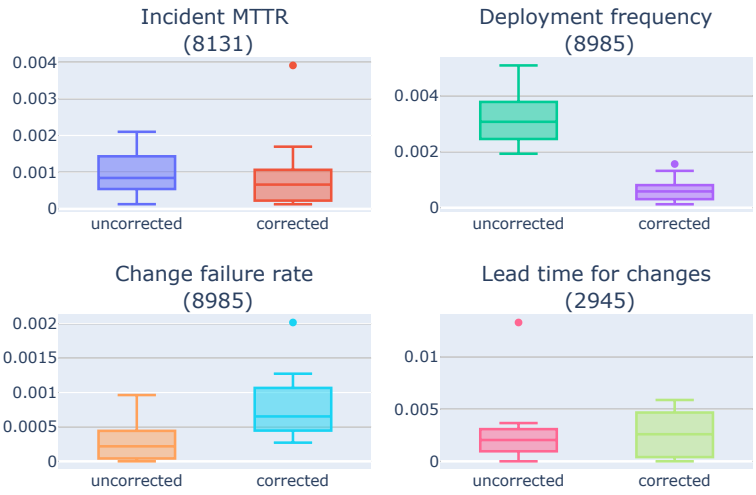

Figure 4.14: F1 scores of 10-fold stratified cross-validation on corrected and uncorrected DORA metrics when evaluating them separately.

Existence of biases in DORA metrics

First, the DORA metrics were checked for the presence of tribe-based biases. When doing so, all four metrics were considered at the same time. Four data points of the four metrics were joined on their squad and month to get a data point that includes all DORA metrics. This resulted in a single data point that has a month, a squad, a tribe and four columns with the uncorrected metrics. Figure 4.13 depicts the results of this experiment on the DORA metrics. This figure shows that it is hard to predict the tribe from the uncorrected DORA metrics, but removing the tribe bias from the data results in an even lower F1 score. Thus, there has been a small tribe-based bias in the DORA metrics, which resulted in a slightly higher prediction score originally.

In the analyses that are to follow, the metrics will be investigated separately and not combined such as was done with the DORA metrics. To correct this mismatch, the experiment was repeated for each one of the four DORA metrics separately. Thus, this new experiment used four data sets where each data set consisted of data points with a squad name, month, tribe name and value for one of the DORA metrics. Figure 4.14 depicts the results of this second experiment. This figure shows that in most of the metrics the F1 score becomes lower when removing the tribe-based bias. This is not the case for the ***change failure rate***. In the case of that metric, removing the tribe-based bias increases the F1 score slightly. Thus, most of the DORA metrics include a tribe-based bias.

In conclusion, most of the DORA metrics contain a tribe-based bias and this bias becomes more evident when evaluating the combination of all four DORA metrics together instead of separately.

Figure 4.15: F1 scores of 10-fold stratified cross-validation on corrected and uncorrected ING metrics. Metrics with an asterisk have been aggregated using the mean.

Existence of biases in ING-specific metrics

As it has been shown that the DORA metrics include a tribe-based bias, an subsequent question would be if this bias remains contained to the DORA metrics only, or if it is present in the ING metrics as well. To this end, the same investigation was carried out for the ING metrics. This investigation is performed for each of the metrics separately, as joining all ING metrics using an inner-join, which is required to generate the combined data set, results in a data set that is much smaller than when each metric is considered independently.

Figure 4.15 shows the box plots for the F1 scores when training an SVM on both variants of the ING metrics. The figure shows that it is hard to predict the tribe from the ING metrics and that for most of them it is even harder to get a correct prediction when the tribe-based bias is removed. However, the ***uptime*** and ***uptime objective achieved*** metrics break this trend. Correcting the tribe bias for the ***uptime*** metric does not change the F1-score significantly and increases the interquartile range significantly. In the case of the ***uptime objective achieved*** metric, removing the tribe bias improves the F1-score but also increases the interquartile range significantly. Given that the ***uptime*** has a significant correlation with the ***uptime objective achieved*** metric, as described in Subsection 4.2.2, the inconsistency in the former might be the cause for the result of the latter.

Findings RQ 4.1 Existence of tribe-based biases in both ING and DORA metrics

This section has investigated the presence of tribe-based biases in both the DORA and ING metrics by training an SVM on corrected and uncorrected variants of the metrics. It has been shown that the DORA metrics contain a tribe-based bias when evaluating all four metrics at the same time. It has also been shown that except for two metrics, all ING metrics contain a bias as well. The only two exceptions are the ***uptime*** and ***uptime objective achieved*** metrics. When considering these findings, it has to be noted that predicting the tribe of a data point is hard to do, even when the tribe-based bias has not been removed.

4.3.2 RQ4.2 Effect of tribe biases on the correlation between metrics

The previous section has established that there is a tribe-based bias present in the DORA metric and almost all ING metrics. Now that this has been established, this section aims to understand what the effect of this bias is on the results in earlier sections. To this end, this section performs the same type of analysis as in Subsection 4.2.2. The difference between the two sections is that where the earlier section investigated the correlations between pairs of metrics directly, this section will first remove the bias from the metrics as described in Subsection 4.3.1 before the correlations are calculated. This section starts by investigating the effects of the biases on the correlations between the DORA metrics. After the DORA metrics have been investigated, the ING metrics are subject to the same type of analysis. When the groundwork has been laid in the form of those two analyses, attention is brought to the correlations between the DORA metrics and the ING metrics after the biases have been removed.

The effect of biases on the correlations of DORA metrics

Previous sections have shown that the DORA metrics are independent, contain a tribe-based bias and that both the metrics themselves and their derivatives do not correlate with each other. This analysis aims to investigate if this still holds after the aforementioned tribe-based biases have been removed.

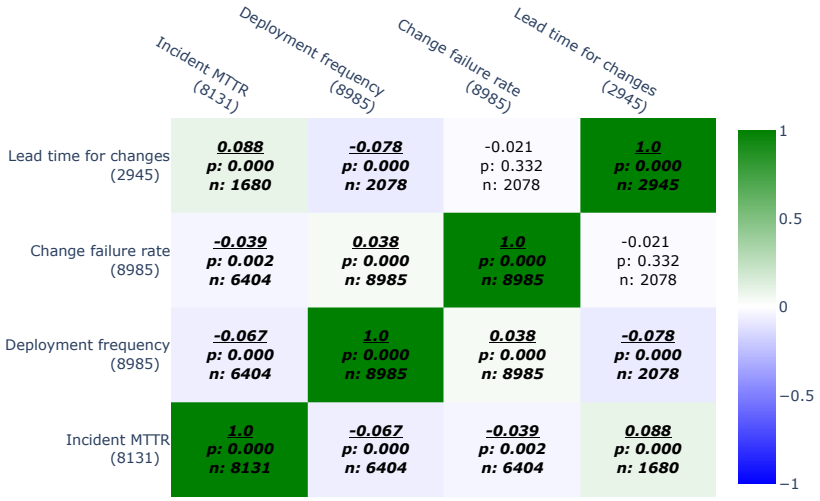

Figure 4.16: Pairwise Spearman correlation between the DORA metrics after correcting for tribe-based bias. Each square includes the correlation strength, the p-value and the number of data points used. Each metric is annotated with the total amount of data points that were available for this metric. Bold correlations have a p-value of ≤ 0.05.

Figure 4.16 depicts the correlations between the DORA metrics after the tribe bias has been removed. This analysis shows that there are no correlations between the corrected versions of the metrics. Contrasting this figure with Figure 4.6 shows that the correlation between the ***lead time for changes*** and the ***change failure rate*** has lost its significance and that the strongest correlation now has a strength of only 0.088.

The effect of biases on the correlations of ING-specific metrics

Investigating the effect of tribe-based biases on the ING metrics is particularly interesting as the interviews already indicated that some users use the monitoring platform differently. These different usages could be reflected in the values of the metrics. Uncovering biases in the ING-specific metrics can also serve as an indication of which metrics are adopted by the organisation but are used differently by different tribes.

Figure 4.17 shows the correlations between the ING metrics after the tribe-based biases have been removed. In contrast to the correlations with the tribe-bias included from Subsection 4.2.2, there is no correlation between the ***interruption MTTR*** and the other metrics. Thus, the previously observed correlations have been the result of tribe-based biases. Besides those correlations, contrasting the two figures results in the observation that five pairs of metrics were correlated in terms of their untreated values but which lost their correlation after accounting for tribe biases. This behaviour means that the metrics are correlated only because of the way tribes use them and not because of their inherent relationship to each other. Besides correlations that disappeared after correcting for the biases, some metrics remained correlated. This means that they are inherently related to each other. Besides metrics that lost or retained their correlation during the process of bias removal, two pairs of metrics gained a correlation. They were not correlated in Subsection 4.2.2, but show fair negative correlations in Figure 4.17. These findings have been summarised in Figure 4.18, correlations that disappeared after correction are red, correlations that appeared are green and the correlations that were significant and had an effect size of at least 0.3 have been depicted in blue.

Figure 4.17: Pairwise Spearman correlation between the ING metrics after correcting for tribe-based bias. Each square includes the correlation strength, the p-value and the number of data points used. Each metric is annotated with the total amount of data points that were available for this metric. Bold correlations have a p-value of ≤ 0.05. Metrics with an asterisk have been aggregated using the mean.

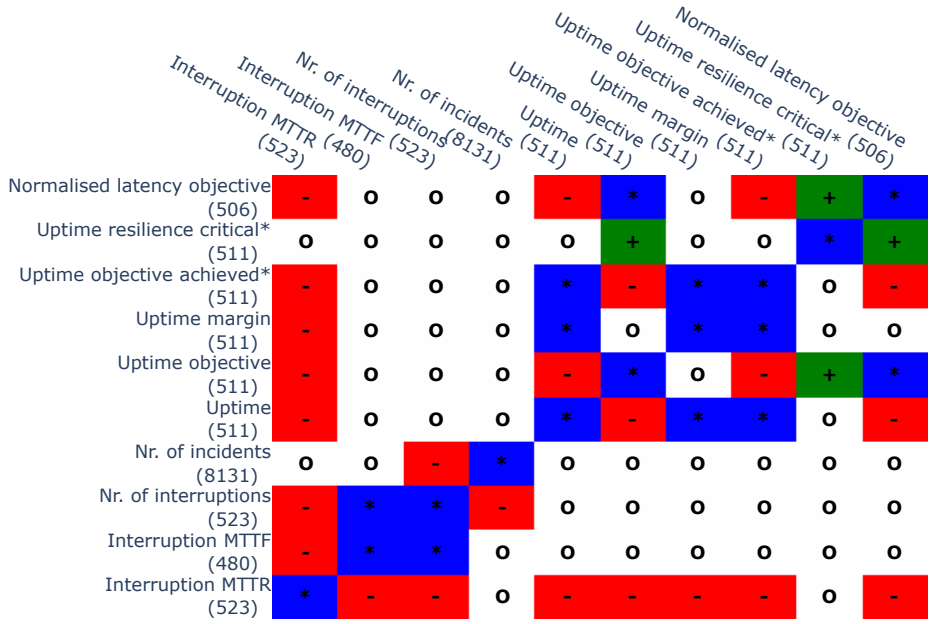

Figure 4.18: Comparison of the correlations before and after bias correction. Red: correlation disappeared after correction, green: Correlation appeared after correction, Blue: Correlation remained despite correction. Metrics with an asterisk have been aggregated using the mean.

The effect of biases on the correlations between DORA and ING metrics

Now that the effect of tribe-based biases has been investigated for the DORA and ING metrics separately, the only thing that remains to do is to investigate its effect on the correlations between the DORA and ING metrics.

	Incident MTTR (8131)	Deployment frequency (8985)	Change failure rate (8985)	Lead time for changes (2945)
Normalised latency objective (506)	***-0.115*** ***p: 0.019*** ***n: 417***	***0.236*** ***p: 0.000*** ***n: 444***	0.002 p: 0.970 n: 444	0.193 p: 0.058 n: 97
Uptime resilience critical* (511)	***0.207*** ***p: 0.000*** ***n: 411***	***-0.108*** ***p: 0.022*** ***n: 452***	0.017 p: 0.711 n: 452	-0.04 p: 0.696 n: 99
Uptime objective achieved* (511)	-0.083 p: 0.093 n: 411	***0.213*** ***p: 0.000*** ***n: 452***	0.035 p: 0.456 n: 452	-0.119 p: 0.242 n: 99
Uptime margin (511)	-0.048 p: 0.331 n: 411	***0.117*** ***p: 0.013*** ***n: 452***	***0.097*** ***p: 0.040*** ***n: 452***	0.024 p: 0.815 n: 99
Uptime objective (511)	***-0.191*** ***p: 0.000*** ***n: 411***	***0.254*** ***p: 0.000*** ***n: 452***	-0.022 p: 0.637 n: 452	***0.324*** ***p: 0.001*** ***n: 99***
Uptime (511)	-0.066 p: 0.180 n: 411	***0.16*** ***p: 0.001*** ***n: 452***	***0.093*** ***p: 0.047*** ***n: 452***	0.018 p: 0.857 n: 99
Nr. of incidents (8131)	***-0.184*** ***p: 0.000*** ***n: 8131***	***0.24*** ***p: 0.000*** ***n: 6404***	***0.056*** ***p: 0.000*** ***n: 6404***	0.028 p: 0.250 n: 1680
Nr. of interruptions (523)	-0.034 p: 0.470 n: 456	0.007 p: 0.871 n: 481	0.063 p: 0.171 n: 481	***0.22*** ***p: 0.033*** ***n: 94***
Interruption MTTF (480)	0.094 p: 0.056 n: 419	0.046 p: 0.338 n: 442	-0.022 p: 0.651 n: 442	***-0.385*** ***p: 0.000*** ***n: 86***
Interruption MTTR (523)	0.031 p: 0.512 n: 456	-0.004 p: 0.928 n: 481	-0.014 p: 0.764 n: 481	-0.016 p: 0.881 n: 94

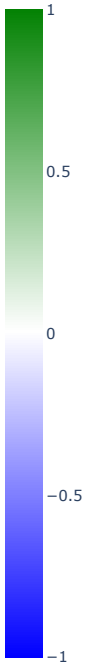

Figure 4.19: Pairwise Spearman correlation between the DORA metrics and the ING metrics after correcting for tribe-based bias. Each square includes the correlation strength, the p-value and the number of data points used. Each metric is annotated with the total amount of data points that were available for this metric. Bold correlations have a p-value of ≤ 0.05. Metrics with an asterisk have been aggregated using the mean.

Figure 4.19 shows the correlations between the DORA and ING metrics after removing the biases. The figure shows that several correlations were at most poor (less than 0.3) in the first analysis that has now become insignificant and that there are several correlations for which the reverse holds. The correlation between the ***lead time for changes*** and the ***uptime objective***, which was the only correlation with an effect of at least 0.3 in the previous analysis, has lost its significance after the removal of the biases. Besides this correlation disappearing, two new correlations have appeared after removing the biases. The ***lead time***

for changes now shows a small positive correlation with the ***uptime objective*** and a negative fair correlation with the ***interruption MTTF***. It has to be noted that these correlations have been established based on a relatively small number of data points in comparison to the other significant correlations in the figure.

Findings RQ 4.2 Effects of tribe-based biases on correlations

This section has shown the effect of tribe-based biases on the correlations between the DORA, ING and DORA and ING metrics. Removing the biases did not result in new correlations between the DORA metrics, there was only one correlation that lost its significance. The correlations between the ***interruption MTTR*** and other ING-specific metrics that were reported previously disappeared after removing the biases, together with other previously found correlations. The analysis showed that only two new correlations emerged when accounting for biases. Those were found between the ***uptime resilience critical*** metric on one hand and the ***uptime objective*** and ***normalized latency objective*** metrics on the other hand. Investigating the effect of biases on the correlations between the DORA metrics and the ING metrics revealed that the only correlation that was found before had been caused by biases, but two other correlations emerged during the analysis. All three events were in relation to the ***lead time for changes*** DORA metric.

4.3.3 Conclusion

This research question has first investigated the presence of tribe-based biases in both the DORA and ING metrics by training a Support Vector Machine (SVM) on corrected and uncorrected variants of the metrics and has subsequently investigated the effect of those biases on the correlations between metrics. The first part of this approach has shown that both the DORA metrics and all but two of the ING-specific metrics contain tribe-based biases. The second part investigated the effect of the tribe-based biases on the correlations between the DORA metrics themselves, between the ING-specific metrics themselves and between the DORA metrics and the ING-specific metrics. Removing the biases did not result in new correlations between the DORA metrics, there was only one correlation that lost its significance. It was found that many of the earlier observed correlations between the ING-specific metrics were caused by tribe-based biases and that only two new correlations emerged when accounting for biases. Finally, it has been established that the only correlation between the DORA and ING-specific metrics that was reported earlier was the result of tribe-based biases. However, two new correlations emerged during the analysis.

Therefore, it is concluded that both the DORA and ING-specific metrics contain biases and that these biases are responsible for many of the correlations that are observed when investigating the unprocessed values of the metrics.

4.4 RQ5: Applicability of used proxies

Previous sections have focused on the relationship between all DORA, ING and DORA and ING metrics in terms of direct values, derivatives and the effect of tribe-based biases. The interviews in Chapter 3 and the descriptions of metrics in Section 4.1.3 also indicated another relation between metrics, namely the use of one metric as a proxy for another. In short, the ***number of incidents*** has been used as a proxy for the availability in the past within ING, and the BI platform currently uses the ***incident MTTR*** as a proxy for the ***interruption MTTR***. In this section, the metric that is actually measured will be referred to as the source of the proxy, while the metric that is being approximated will be referred to as the target of the proxy. These two proxies are used to answer the final research question and therefore compose its sub-questions, as depicted below.

Specification of research questions

RQ5 What proxies are used by ING and how do they relate to the metric they substitute?
- **RQ5.1** Is the number of incidents a good proxy for availability?
- **RQ5.2** Is the incident MTTR a good proxy for the interruption MTTR?

For both proxies, it is the case that the proxy source is being measured across the whole organisation, while the proxy target is being measured using the monitoring platform for the assets of a subset of all squads. This distinction allows this thesis to investigate the relationship between the proxy source and target for the squads that measure the proxy target for their assets. As both proxy sources are calculated from the incident data, it is possible to split the incidents by their priority before performing the analysis. This improves the granularity of the results. To investigate the relationship between the proxy source and target, the incidents are processed as describes in Subsection 4.1.5. Note that the processing splits the incidents by priority as described in that section. This analysis only includes the correlations between the uncorrected metrics and between the metrics after correcting for tribe-based biases, as earlier investigations have shown that the derivatives of metrics do not yield much extra information.

4.4.1 RQ5.1: Number of incidents as a proxy for availability

Chapter 3 indicated that in the past the ***number of incidents*** has been used as a proxy for the availability, which is now measured using the ***uptime***. This proxy is no longer used as it was established that it was vulnerable to cheating. Despite this finding, it was decided to investigate its applicability nevertheless, as this could still generate insights into how the number of incidents relates to the availability.

The analysis in Subsection 4.2.2 has shown that the overall ***number of incidents*** only correlated with the ***number of interruptions***. Figure 4.20 depicts the correlations between the ***number of incidents*** per priority and the metrics related to uptime and provides several insights. Firstly, this correlation was caused by the incidents with priority three. Secondly,

several correlations become evident when considering each priority separately but are hidden when considering all priorities at the same time. The number of incidents of priority three correlates fairly positively with the ***number of interruptions*** and negatively with the ***interruption MTTR***. The number of incidents with priority four correlate positively with the ***uptime resilience critical*** metric and negatively with the ***normalised latency objective*** and the ***uptime objective***.

As has been shown in Subsection 4.2.3, some correlations can be caused by tribe-based biases instead of by the relation of the metrics themselves. Thus, the same experiment will be repeated for the derivative of the number of incidents split by priority. Figure 4.21 shows the result of this analysis. The only correlation that remains is the correlation between the ***number of incidents*** of priority three and the ***number of interruptions***, the others have disappeared after the correction. In contrast to the uncorrected variant of this analysis, the correlation between the total ***number of incidents*** and the ***number of interruptions*** has disappeared and only the correlation between the number of incidents of priority three and the ***number of interruptions*** remains.

Findings RQ 5.1 Applicability of number of incidents as a proxy for availability

In the past, the ***number of incidents*** has been used as a proxy for availability. This analysis has shown that it does not correlate with availability-related metrics.
However, this section has shown that the number of incidents of priority three and four have correlations with other ING metrics when considering the priorities separately. Most of those correlations are caused by tribe-based biases, with the only exception being the correlation between the ***number of incidents*** of priority three and the ***number of interruptions***.

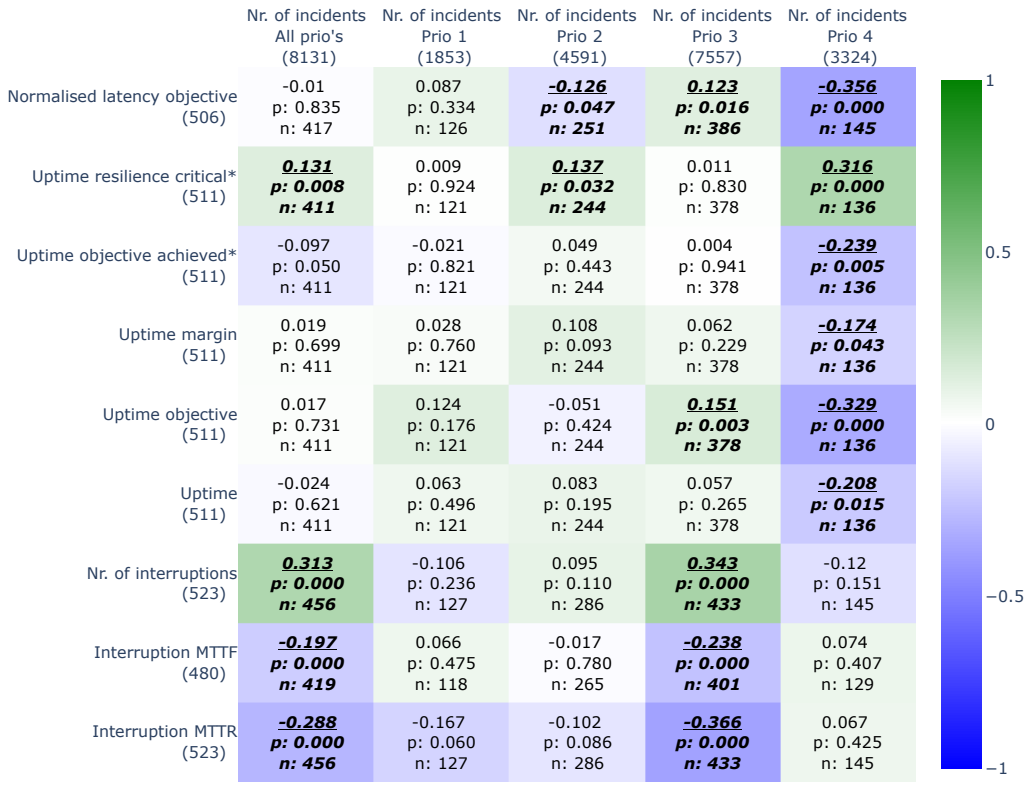

Figure 4.20: Pairwise Spearman correlation between the number of incidents per priority and ING metrics. The left column depicts the aggregate over all priorities and is the same column as shown in earlier sections. Each square includes the correlation strength, the p-value and the number of data points used. Each metric is annotated with the total amount of data points that were available for this metric. Bold correlations have a p-value of ≤ 0.05. Metrics with an asterisk have been aggregated using the mean.

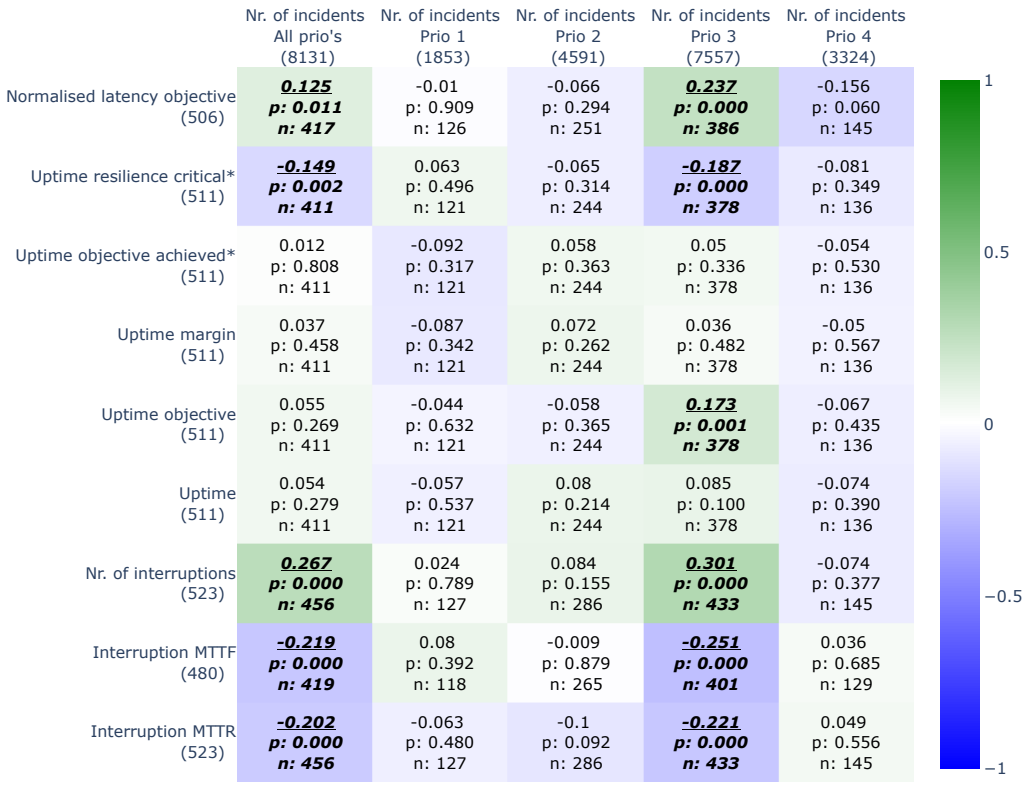

Figure 4.21: Pairwise Spearman correlation between the number of incidents per priority and ING metrics after correcting for tribe-based bias. The left column depicts the aggregate over all priorities. Each square includes the correlation strength, the p-value and the number of data points used. Each metric is annotated with the total amount of data points that were available for this metric. Bold correlations have a p-value of ≤ 0.05. Metrics with an asterisk have been aggregated using the mean.

4.4.2 RQ5.2: Incident MTTR as a proxy for the interruption MTTR

Section 4.1.3 has described how the ***incident MTTR*** is used as an organisational-wide proxy for the ***interruption MTTR***. This thesis investigates this relation both in terms of uncorrected and corrected metrics when splitting the incidents on their priority.

The correlations between the ***incident MTTR*** per priority and the ING metrics have been depicted in Figure 4.22. This figure shows that splitting the incidents by priority does not reveal correlations that are hidden in the aggregated data set in the left column and thus that the ***incident MTTR*** does not correlate with anything. Another observation is that the ***number of incidents*** does not correlate with the ***incident MTTRs*** of any of the priorities. In other words, it is not the case that when a squad has fewer incidents in a given month, those incidents are more or less severe. The many significant (although poor) correlations of incidents with a priority of three can likely be attributed to the amount of data that is available for incidents with this priority. As can be seen from the descriptions of the metrics, there are many more data points for incidents with priority three than there are for the other priorities (over 7500 versus at most 4591). Given that the analysis of the uncorrected values of the metrics does not provide new insights, the same analysis is repeated with the metrics after correcting them for the tribe-based biases. Figure 4.23 shows this analysis after removing the biases. In contrast to Figure 4.22, this figure does include a fair and significant correlation. This correlation appears between the ***incident MTTR*** with priority two and the ***uptime resilience critical*** metric.

Findings RQ 5.2 Applicability of incident MTTR as a proxy for the interruption MTTR

The ***incident MTTR*** is used as an organisational-wide proxy to the ***interruption MTTR***. This analysis has investigated the correlation between the ***incident MTTR*** when splitting it by the priority of the incident and the ING metrics, looking at both the uncorrected and corrected metrics.
This section has shown that the ***incident MTTR*** does not have correlations with any of the ING metrics when considering the values directly and only correlates fairly with the ***uptime resilience critical*** metric after removing the tribe-based biases.

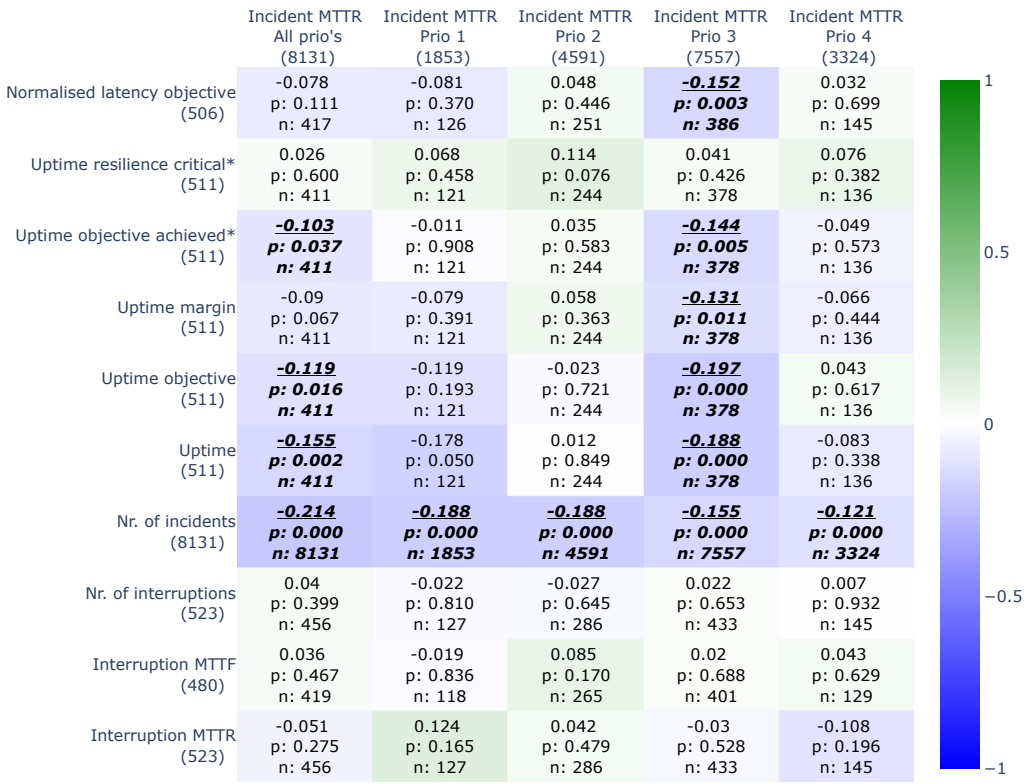

Figure 4.22: Pairwise Spearman correlation between the incident MTTR per priority and ING metrics. The left column depicts the aggregate over all priorities and is the same as the incident MTTR column of Figure 4.8. Each square includes the correlation strength, the p-value and the number of data points used. Each metric is annotated with the total amount of data points that was available for this metric. Bold correlations have a p-value of ≤ 0.05. Metrics with an asterisk have been aggregated using the mean.

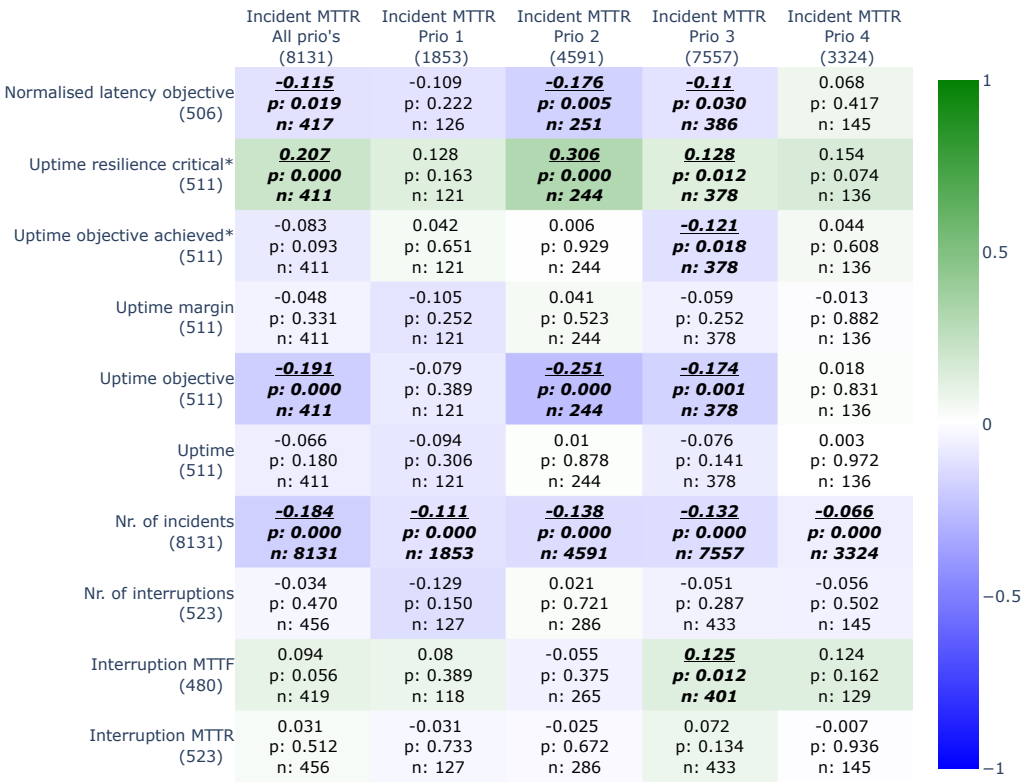

Figure 4.23: Pairwise Spearman correlation between the incident MTTR per priority and ING metrics after correcting for tribe-based bias. The left column depicts the aggregate over all priorities. Each square includes the correlation strength, the p-value and the number of data points used. Each metric is annotated with the total amount of data points that were available for this metric. Bold correlations have a p-value of ≤ 0.05. Metrics with an asterisk have been aggregated using the mean.

4.4.3 Conclusion

This thesis has discovered two proxies that either currently are or have been in use by ING and that could be evaluated as both the proxy and the metric it is intending to approximate was available for several squads. This thesis has evaluated the use of the ***number of incidents*** as a proxy for the availability and the use of the ***incident MTTR*** as a proxy of the ***interruption MTTR***. In these proxies, incidents are tickets in the IT service management system and interruptions indicate downtime of operational assets and incidents can have a priority between one and four, where one is the most important.

It has been shown that neither of the proxies correlates with the metric they are intending to approximate when splitting the incidents based on their priority. The ***number of incidents*** does not correlate with availability-related metrics, but it does correlate with the ***number of interruptions***. The number of incidents of priority three and four have correlations with other ING metrics when considering the priorities separately but most of those correlations are caused by tribe-based biases, with the only exception being the correlation between the ***number of incidents*** of priority three and the ***number of interruptions***. The ***incident MTTR*** is used as an organisational-wide proxy to the ***interruption MTTR***. This thesis has shown that the ***incident MTTR*** does not have correlations with any of the ING-specific metrics when considering the values directly and only correlates fairly with the one metric after removing the tribe-based biases.

In conclusion, the ***number of incidents*** is used as a proxy for the availability and the ***incident MTTR*** is used as a proxy of the ***interruption MTTR***, but neither of them correlates with the metric they substitute. There is only a correlation between the ***number of incidents*** and the ***number of interruptions***.

Chapter 5

Discussion

This chapter provides a discussion on the findings from this thesis in terms of actionable insights for industry, and ING in particular. Based on the findings of this thesis, this chapter also proposes future work to answer the questions that were encountered but could not be answered. Finally, this chapter provides an overview of the things that threatened the validity of this thesis.

5.1 Actionable insights and future work

This section provides the actionable insights gained from this thesis and suggests directions of future work. No explicit distinction has been made between the two, as some generated insights are actionable the way they are now but also invite future work to take a closer look at their underlying mechanisms or causes. For the sake of brevity, this section will use ”product” to denote both software products and services.

Leverage ITSM tickets to improve products but not to measure them This thesis has investigated metrics calculated from ITSM incident tickets and how they relate to objective metrics. This relationship is relevant, as they portray two sides of software products: The ITSM data is generated by humans, while the objective data is measured by automated tools. Thus, this relationship portraits how users see the product versus how it behaves objectively.

The metrics that are calculated from ITSM incident tickets have been investigated as proxies for objective metrics. It has been shown that both proxies did not correlate with the metric they were supposed to substitute, both before and after the correction of tribe-based biases. Additionally, the interviews in Chapter 3 have established that collecting feedback from users is important and this importance is further stressed by Forsgren et al. [15]. The same interviews have also established that reliably collecting objective metrics on the scale of global organisations is difficult to achieve and that automated data collection is required to make metrics auditable.

In short, both ITSM data and objectively collected metrics have their own respective functions and advantages but can not be used to substitute each other. On one hand, ITSM data is often abundantly available without the need for additional measuring platforms or

tools and reports on problems that people have encountered. On the other hand, objective metrics are auditable and can continuously monitor assets. This implies that organisations should leverage the advantages of each of those two classes of metrics. ITSM data can be used to improve products, but not to make objective statements about them. Objective metrics can be used to generate insights into the operation of products, but does not correlate with what is perceived by users.

Decouple responsibilities of metric collection and usage Selecting the right metrics to collect and use is important for organisations. On one hand, collecting too little metrics can create an incomplete picture of their product, while collecting too many metrics can create conflicting views or information overload.

Metrics can be grouped into themes, such as stability or throughput. One might expect that metrics from the same group behave similarly within an organisation. For example, a product that performs well at one aspect of stability could be expected to also perform well at other aspects of stability. This thesis has shown that this assumption does not necessarily hold. Within the single case study of ING, metrics from similar classes did not show linear correlations with each other, did not change together over time and did not correlate after accounting for biases introduced by the organisational structure. This finding implies that decisions need to be made about which metrics to use for measuring each aspect of a software product. Such decisions prevent the collection of contradicting metrics when assessing the different aspects of software products, but simultaneously raise the issue of who should decide which metrics should be collected. On one hand, one could argue that this decision should be made by higher management to create an uniform set of metrics that is measured across the organisation. On the other hand, doing so would infringe the autonomy of teams as promoted by the agile manifesto [3]. Additionally, the interviews in Chapter 3 have indicated that standardisation of reports and metrics enables the collection of metrics throughout the organisation and that the automated collection of metrics makes them auditable, which might be a requirement in certain regulated industries.

As with many problems, the best solution probably lies in between those two extremes. One solution might be to create an organisational-wide catalogue of standardised metrics, from which the individual teams get to pick the metrics they find most important for their product. This way, standardisation and a degree of uniformity are guaranteed while teams partially maintain their autonomy. A second and more team-oriented approach could be to let teams decide for themselves which metrics they collect, but have management assure that all important aspects of the product are covered by metrics. This approach provides more autonomy to the teams, at the cost of standardisation and uniformity of metrics throughout the organisation.

In conclusion, the responsibilities of collecting and using metrics should be decoupled. Two approaches are recommended to achieve this decoupling. Introducing an organisational-wide catalogue of standardised metrics provides standardisation and uniformity of metrics at the cost of autonomy of the squads, while having management only verify that all areas of metric collection are covered can provide more autonomy to squads at the expense of standardisation.

Cautiously trade frequency of failure for duration of outage The availability of a software product is determined by the frequency of failure and the duration of outages. Organisations need to make a careful trade-off between these two metrics, as both this thesis and literature have shown that they each come with advantages and disadvantages. Balancing these metrics can allow organisations to achieve both optimal throughput and availability.

This thesis has shown that the ***interruption MTTR*** and ***interruption MTTF*** have a fair and positive correlation and that this correlation does not exist between their derivatives. It can be attributed to the behaviour of people and is not inherent to the metrics themselves, as it disappears after the removal of tribe-based biases. Furthermore, lower ***interruption MTTR*** has been shown to correlate with larger ***uptime*** and an increased values for the ***uptime objective achieved*** metric. This implies that squads who fail more often tend to recover from failure quicker and that this quicker recovery is associated with desirable outcomes in terms of availability. These findings are in line with the recommendations made in literature. Forsgren et al. [15] states that it is more important to be able to recover quickly from a failure than it is to prevent failures, as failures are inevitable in contemporary complex software projects and Farroha and Farroha [12] recommend to place more focus on the mean time to repair (MTTR) than on the mean time between failures (MTBF). On the other hand, Facebook has changed their motto from "Move fast and break things" to "Move fast with stable infrastructure" [35]. Their focus has shifted towards stability of the interfaces that developers use to connect with Facebook products. This shift was partially motivated by the realisation that their overall throughput could increase if existing software would need less repeated repairs. Furthermore, the interviews have indicated that ING is obligated by the Dutch National Bank to have a certain level of availability. However, one could argue that what really matters is that this obligation is met, independent from if it is caused by frequent short outages or sporadic long ones. It also has to be noted that Forsgren et al. [15] has shown that organisations that perform better in terms of stability also tend to perform better in terms of throughput.

Thus, organisations are advised to focus on recovering rapidly from a failure instead of on never failing but not to take this focus to the extreme, as this might impede the organisation in the long run. Additionally the amount of downtime that is acceptable, and thus the risk that organisations are willing to take, is dependent on the context of the organisation. Therefore, organisations should critically analyse how fast they want to move and how much they want to risk breaking things.

Implement *enabling practices* on an organisational scale Implementing *enabling practices*, such as cloud infrastructure or an extensive continuous delivery pipeline, can enable improvements in both stability and throughput. Improving these features of the software products allows organisations to deliver better products faster to customers.

This thesis has shown that the four DORA metrics, which measure throughput and stability, are independent from each other when evaluating them between squads. This implies that there is no correlation between the throughput of squads, and the stability of their product. The cross-organisational analysis of the DORA report [16] has shown that throughput and stability move in tandem between organisations. Thus, this thesis concludes that stability and throughput move in tandem between organisations, but not between squads. A likely

explanation for this could be that the usage of *enabling practices* often differs between organisations but not between squads. This hypothesis is in line with the DORA report, which recommends a number of *enabling practices* to improve an organisation's performance in terms of the four metrics. This thesis has show that such practices are already being implemented at the organisational level of ING, such as the continuous delivery pipeline to which all personnel with and IT function should be onboarded and which should improve the throughput by automating the process of risk management. Additionally, this thesis has shown that the context of an organisation can have an effect on the software engineering practices. One example of this are the regulations that have an effect on the throughput.

As was the case with the decision of which metrics to measure, implementing *enabling practices* on an organisational scale runs the risk of violating the autonomy of teams. However, one could argue that enforcing centralised tooling and practices violates this autonomy less than enforcing metrics to collect, given that generally teams already share tooling and practices.

Given the aforementioned findings and the recommendations of the DORA report [16], organisations are advised to investigate the applicability of *enabling practices* within their own context and to implement the most suitable practices across their organisation. It is recognised that this advise partially violates the autonomy of teams. Therefore, organisations should also make the decision if they are willing to trade this part of autonomy of their teams for *enabling practices* that are likely to increase the SDO performance of the organisation.

Investigate code-quality metrics as proxies for both interruption and incident MTTR
There exists a gap in the current understanding of how code-quality metrics relate to objective and user-perceived outcomes of software products in a corporate setting. Filling this gap can help organisations to understand which aspects of their software's code-quality influence the objective and user-perceived outcomes of their products, enabling data-driven improvement of software products.

Wu and Zhang [52] investigated the relation between MTTR and several code-related metrics. They used the GitHub tickets of an open source project as the source of their MTTR, which is close to the ***incident MTTR*** used in this thesis. They found four metrics that had a moderate correlation with the MTTR. Paulson et al. [41] has compared three closed-source projects to three open-source projects and found that defects in open source projects are found and repaired more rapidly compared to closed-source projects. This raises the question if the findings of Wu and Zhang [52] hold in a corporate environment. Additionally, this thesis has shown that there is no correlation between the MTTR of incidents and the MTTR of objectively measured interruptions.

Thus, two gaps in the current state of research have been identified. Firstly, the relation between code-quality metrics and incident MTTR in a corporate environment is unclear due to the differences between open source and closed source projects. Secondly, the relation between code-quality metrics and objectively measured interruption MTTR has not been established, as it has been shown that there is a mismatch between interruption and incident MTTR. Filling these gaps in research could help organisations to understand the effect that

code-quality metrics have on the reported incidents and the objective behaviour of their products. Such understanding can enable the data-driven improvement of products.

In short, previous research has investigated the use of code-quality metrics as proxies for incident MTTR in OSS projects and has established a difference between open source and closed source projects. This thesis has established that the ***incident MTTR*** and ***interruption MTTR*** do not correlate. Therefore, future work should investigate the use code-quality metrics as proxies for both interruption and incident MTTR in a corporate environment.

Replicate research across industries A gap in the current understanding of the relation between different properties of organisations and their usage of metrics has been established. Filling this gap can firstly help organisations better handle their regulations. Secondly, gaining understanding of the relationship between the different properties and the usage of metrics can generate more specific recommendations for organisations, based on their individual properties.

The interviews in Chapter 3 have indicated that the strict regulations that ING is subject to have a significant effect on the use and collection of metrics. For example, manual data collection is not allowed for metrics that need to be auditable and the risk KPI is enforced down from the European Central Bank. This thesis has performed a single case study within the financial services industry, but there are many more industries that are subject to strict regulations such as the aviation, automotive and medical industries. Thus, part of the gap consists of the knowledge surrounding the effects of regulations on the usage of metrics in different heavily regulated industries. Filling this gap could allow organisations to learn from the approaches used in other industries to handle the regulations.

This gap in research is also reflected in the DORA report. During the interviews in Chapter 3, some interviewees noted that achieving both throughput and stability at the same time might not be possible within ING due to the heavy regulations it is subject to, but that this duality is likely to hold in general. However, the report by Forsgren et al. [16] reported that 12 percent of the companies at which the participants worked were active in the financial services industry and that 79 percent of respondents in the report were from North America, the UK or the EU. It is assumed that the financial services industry in those regions is subject to similar regulations as ING. Therefore, a gap exist in the current understanding of the effect that heavy regulations have on the duality between stability and throughput. Better understanding this effect can help to generate recommendations that are better tailored towards the specific properties of organisations. For example, organisations from heavily regulated industries could get different recommendations than organisations from other industries.

During the writing of this thesis, a discussion point was raised about the possibility that the DORA report includes an confirmation bias. The duality between stability and throughput has been shown among organisations that existed while the survey was performed. However, it might be the case that there also exist organisations that had similar values for the metrics but went bankrupt after the survey was completed. Thus, there is a gap in understanding how the DORA metrics relate to the future performance of organisations. Increased knowledge about the relation between the DORA metrics and future performance

of organisations can help organisations to make better informed decisions about their use of the metrics.

As has been shown by Levy [35], companies can shift their focus from throughput to stability when they mature. However, the DORA report does not make a distinction between new and established companies. Thus, there is a gap in the current understanding of how the four metrics relate to each other over time as an organisation becomes more mature. Understanding how the age of an organisation impacts its focus can help to create recommendations that are tailored to the specific properties of organisations.

In conclusion, there is a gap in the current understanding of metric usage across industries and of the DORA metric in relation to different properties of organisations. Future research should replicate this thesis and the DORA report while differentiating on the aforementioned properties of organisations to fill these gaps, which can help to create recommendations that are better tailored to the specific properties of organisations.

5.2 Threats to validity

5.2.1 Internal validity

Setup of literature study The literature review was constructed over three iterations of work. In the initial iteration, only the queries containing the term ”DevOps” were used. During this iteration, it turned out that the terms ”DevOps” and ”Continuous Delivery” were often used interchangeably in literature. Thus, the second iteration repeated all queries, but with ”Continuous Delivery” instead of ”DevOps”. After completing this iteration, it was found that the collected metrics did not cover opensource projects. Thus, a third iteration was performed using the query ”continuous integration opensource measure”. The total of those three iterations resulted in the results as described in this thesis. This structure of iterations together with the use of one level of bidirectional snowballing poses the risk that this literature survey was not exhaustive and did not catch all relevant literature. However, given that the resulting metrics can be clustered into diverse topics, we are confident that this risk has been mitigated sufficiently.

Selection of interview participants Out of the 13 distinct employees that have been invited, five participated in an interview. This low number of participants can introduce the risk of not being exhaustive. It is possible that there are tribes or areas within ING that use metrics which would have been informative for this thesis, but which have not been discovered. By inviting and interviewing employees with leadership positions in areas that are strongly related to IT and by leveraging the organisation-specific knowledge of the company supervisor, this risk has been mitigated.

Processing of interviews by only one author The interviews of Chapter 3 have been held and analysed by the author of this thesis. This poses a risk to the validity of the results, as those rely on the interpretation of the interviews of only one person. However, requesting the university supervisor to validate the results was not possible, as the information from the interviews was confidential and could thus not be shared with non-ING employees.

5.2.2 Construct validity

Calculation of biases Subsection 4.3.2 used the paper by Koren [30] to calculate the biases and remove them from the collected metrics. The aim of this analysis was to determine which correlations are caused by the behaviour of tribes and which correlations are inherent to the metrics themselves. However, this methodology can only distinguish between those two cases when there are tribes that do not have the bias. If there are two metrics that are unrelated, but for which the behaviour of all tribes makes them correlate, the analysis will show that the two metrics are correlated. This thread has been mitigated by collecting the metrics from as many ING tribes as possible.

Calculation of derivatives Subsection 4.2.3 calculated the derivatives of metrics and investigated the correlations between them. The calculated derivatives were of the first order and used interpolation to handle missing data. These derivatives did not take into account that there can be a delay in the interaction between metrics. For example, it could be the case that an increase in deployment frequency results in an increase in incidents two months later. This analysis does not capture such delays. This thesis has established the relationships between the derivatives that manifest in the same month. Analysing the potential delayed interactions is beyond the scope of this thesis.

Sporadic correlations This thesis has matched metrics based on their squad name and the month during which the metric was collected. However, some data sets had a small amount of matching data points. Figure 4.8 for example shows that between the incident MTTR and the ING metrics there are more than 400 matching data points, but between the lead time for changes and the ING metrics there are often less than 100 data points used to calculate the correlation. This introduces the risk of reporting correlations that occur by random chance. To mitigate this risk, the number of used data point and the p-value have been reported for each of the correlations and only correlations with a significant p-value have been reported. The latter reduces the risk of reporting correlations that occurred by chance, while the former allows the reader to get and impression of how reliable the reported correlation is.

It can be argued that the amount of experiments performed on each metric in combination with the value of $\alpha = 0.05$ makes this thesis prone to finding false positives. This argument could be constructed as follows: Each of the ING metrics is compared to the nine other ING metrics in terms of its raw value, its derivative and its value after correcting for tribe-biases. Choosing $\alpha = 0.05$ means that one in 20 correlations are the result of random chance. Therefore, at least one of the 27 experiments performed on the ING metric could be reported to be significant based on chance. This issue could be mitigated by applying Bonferroni adjustments [42], which corrects the value of α for the number of experiments performed. One of the problems with this approach is that it allows the number of performed experiments to influence the interpretation of the results [42]. As as example, the correlation between metrics A and B could be significant in the current version of this thesis, but adding an unrelated metric C and computing the correlation between A and C could make the first correlation become insignificant. Therefore, it has been decided to not

use this correction, although it is recognised that sporadic correlations could be a thread to validity to this thesis.

5.2.3 External validity

Data collected over the pandemic The data used in Chapter 4 has been collected over the year 2020. On March 12th 2020, the Dutch government urged employees to work from home as much as possible due to the SARS-CoV-2 pandemic [39]. This could have had an effect on the metrics that have been collected for this thesis. Although there has been no way to mitigate the impact of this factor, Microsoft has reported that the productivity of their employees remained stable or slightly increased on the organisational level when measuring the productivity in terms of opened and closed pull requests[13]. Thus, it is possible that this factor had a minor effect on the findings of this thesis.

Chapter 6

Summary

Many organisations involved in software engineering collect metrics, the most important of which can serve as Key Performance Indicators (KPIs). Those KPIs are used by management to make informed decisions and understand the performance of the organisation. Organisations often collect industry-standard metrics, in addition to their own set of metrics. The DORA report [16] defines four key and industry-standard metrics that represent the aspects of stability and throughput of software products. Those four metrics are used to compare organisations in terms of their Software Delivery and Operational (SDO) performance. This thesis has performed a single case study within ING, a large and highly digital bank in the Netherlands. It investigated the DORA metrics as measured within ING and established that the four metrics are independent from each other. After composing a list of ten IT-related KPIs that are used by ING, it has shown that some of these metrics correlate, but that a significant amount of these correlations are due to biases introduced via the organisational structure, as different organisational units use metrics differently. It has also shown that there are no correlations between the DORA metrics and the metrics that ING collects additionally. The use of one metric as a proxy for another metric or a thematic group of metrics is a common phenomenon within industry. This thesis has investigated how two proxies relate to the metric they are intended to approximate. It has been found that neither of these proxies shows a significant correlation of considerable effect size with the metric it approximates. This case study has resulted in six actionable insights and recommendations for future work, the three main ones of which will be highlighted. The first insight finds that metrics calculated from IT Service Management tickets and objectively measured metrics both can bring value to an organisation, but that there is no correlation between them. The second insight highlights the trade-off between failing fast and the stability of the product. The third insight discusses the complexities of which metrics should be measured and who should make that decision. It proposes two approaches, one of which optimises for standardisation of metrics while the other one favours metrics autonomy of teams. Three contributions have been made by this thesis. Firstly, the state of the art of DevOps metrics in scientific literature has been established by means of a structured literature review. Secondly, exploratory interviews within the context of the heavily regulated financial services industry have been performed to investigate the KPIs that ING uses. Finally, a thorough analysis of the relationships between metrics used within the bank has been performed.

Bibliography

[1] Haldun Akoglu. User's guide to correlation coefficients. *Turkish Journal of Emergency Medicine*, 18(3):91–93, September 2018. ISSN 2452-2473. doi: 10.1016/j.tjem.2018.08.001. URL https://www.sciencedirect.com/science/article/pii/S2452247318302164.

[2] Martin Arvedahl and Christopher Åkersten. Exploring the Criticality and Impact of DevOps Practices. Master's thesis, Chalmers University of Technology; University of Gothenburg, Gothenburg, 2018. URL https://odr.chalmers.se/handle/20.500.12380/300540. Accepted: 2019-11-12T09:45:39Z.

[3] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas. Manifesto for Agile Software Development, 2001. URL https://agilemanifesto.org/.

[4] Cor-Paul Bezemer, Simon Eismann, Vincenzo Ferme, Johannes Grohmann, Robert Heinrich, Pooyan Jamshidi, Weiyi Shang, André van Hoorn, Monica Villavicencio, Jürgen Walter, and Felix Willnecker. How is Performance Addressed in DevOps? In *Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering*, ICPE '19, pages 45–50, New York, NY, USA, April 2019. Association for Computing Machinery. ISBN 978-1-4503-6239-9. doi: 10.1145/3297663.3309672. URL https://doi.org/10.1145/3297663.3309672.

[5] Christopher Bishop. Sparse kernel machines. In *Pattern Recognition and Machine Learning*, Information Science and Statistics, pages 325–358. Springer-Verlag, New York, 2006. ISBN 978-0-387-31073-2. URL https://www.springer.com/gp/book/9780387310732.

[6] David A. Bishop. Key Performance Indicators: Ideation to Creation. *IEEE Engineering Management Review*, 46(1):13–15, 2018. ISSN 1937-4178. doi: 10.1109/EMR.2018.2810104. Conference Name: IEEE Engineering Management Review.

[7] M. Callanan and A. Spillane. DevOps: Making It Easy to Do the Right Thing. *IEEE Software*, 33(3):53–59, May 2016. ISSN 1937-4194. doi: 10.1109/MS.2016.66. Conference Name: IEEE Software.

[8] Yiong Chan. Biostatistics 104: Correlational Analysis. *Singapore medical journal*, 44:614–9, January 2004.

[9] Gabriel Silva Cogo. *Understanding DevOps: From its enablers to impact on IT performance*. Thesis, Texas Tech University, Lubbock, August 2019. URL https://ttu-ir.tdl.org/handle/2346/85394. Accepted: 2019-11-01T15:18:44Z.

[10] J. Cox, E. Bouwers, M. van Eekelen, and J. Visser. Measuring Dependency Freshness in Software Systems. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, volume 2, pages 109–118, May 2015. doi: 10.1109/ICSE.2015.140. ISSN: 1558-1225.

[11] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano. DevOps. *IEEE Software*, 33(3):94–100, May 2016. ISSN 1937-4194. doi: 10.1109/MS.2016.68. Conference Name: IEEE Software.

[12] B. S. Farroha and D. L. Farroha. A Framework for Managing Mission Needs, Compliance, and Trust in the DevOps Environment. In *2014 IEEE Military Communications Conference*, pages 288–293, October 2014. doi: 10.1109/MILCOM.2014.54. ISSN: 2155-7586.

[13] Denae Ford, Margaret-Anne Storey, Thomas Zimmermann, Christian Bird, Sonia Jaffe, Chandra Maddila, Jenna L. Butler, Brian Houck, and Nachiappan Nagappan. A Tale of Two Cities: Software Developers Working from Home During the COVID-19 Pandemic. *arXiv:2008.11147 [cs]*, July 2021. URL http://arxiv.org/abs/2008.11147. arXiv: 2008.11147 version: 2.

[14] Nicole Forsgren and Jez Humble. DevOps: Profiles in ITSM Performance and Contributing Factors. SSRN Scholarly Paper ID 2681906, Social Science Research Network, Rochester, NY, October 2015. URL https://papers.ssrn.com/abstract=2681906.

[15] Nicole Forsgren, Jez Humble, and Gene Kim. *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*. IT Revolution, March 2018. ISBN 978-1-942788-35-5.

[16] Nicole Forsgren, Dustin Smith, Jez Humble, and Jessie Frazelle. 2019 Accelerate State of DevOps Report. Technical report, 2019. URL http://cloud.google.com/devops/state-of-devops/.

[17] David A. Freedman. *Statistical Models: Theory and Practice*. Cambridge University Press, New York, UNITED STATES, 2009. ISBN 978-0-511-60336-5. URL http://ebookcentral.proquest.com/lib/delft/detail.action?docID=461133.

[18] Karl Pearson F.R.S. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, November 1901. ISSN 1941-5982. doi: 10.1080/14786440109462720. URL https://doi.org/10.1080/14786440109462720. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/14786440109462720.

[19] D. R. Garrison, M. Cleveland-Innes, Marguerite Koole, and James Kappelman. Revisiting methodological issues in transcript analysis: Negotiated coding and reliability. *The Internet and Higher Education*, 9(1):1–8, January 2006. ISSN 1096-7516. doi: 10.1016/j.iheduc.2005.11.001. URL http://www.sciencedirect.com/science/article/pii/S1096751605000771.

[20] Taher Ahmed Ghaleb, Daniel Alencar da Costa, and Ying Zou. An empirical study of the long duration of continuous integration builds. *Empirical Software Engineering*, 24(4):2102–2139, August 2019. ISSN 1573-7616. doi: 10.1007/s10664-019-09695-9. URL https://doi.org/10.1007/s10664-019-09695-9.

[21] P. Gill, K. Stewart, E. Treasure, and B. Chadwick. Methods of data collection in qualitative research: interviews and focus groups. *British Dental Journal*, 204(6):291–295, March 2008. ISSN 1476-5373. doi: 10.1038/bdj.2008.192. URL https://www.nature.com/articles/bdj.2008.192. Number: 6 Publisher: Nature Publishing Group.

[22] Georgios Gousios, Martin Pinzger, and Arie van Deursen. An exploratory study of the pull-based software development model. In *Proceedings of the 36th International Conference on Software Engineering*, ICSE 2014, pages 345–355, New York, NY, USA, May 2014. Association for Computing Machinery. ISBN 978-1-4503-2756-5. doi: 10.1145/2568225.2568260. URL https://doi.org/10.1145/2568225.2568260.

[23] ING Groep N.V. ING profile 1Q/2021, May 2021. URL https://www.ing.com/MediaEditPage/ING-profile-1Q2021.htm.

[24] M. R. Islam and M. F. Zibran. Insights into Continuous Integration Build Failures. In *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*, pages 467–470, May 2017. doi: 10.1109/MSR.2017.30.

[25] Aman Jain and Raghu ram Aduri. Quality metrics in continuous delivery : A mixed approach. Master's thesis, Blekinge Institute of Technology, Karlskrona, 2016. URL http://urn.kb.se/resolve?urn=urn:nbn:se:bth-12804.

[26] Romit Jain, Saket Kumar Singh, and Bharavi Mishra. A Brief Study on Build Failures in Continuous Integration: Causation and Effect. In Chhabi Rani Panigrahi, Arun K. Pujari, Sudip Misra, Bibudhendu Pati, and Kuan-Ching Li, editors, *Progress in Advanced Computing and Intelligent Engineering*, Advances in Intelligent Systems and Computing, pages 17–27, Singapore, 2019. Springer. ISBN 9789811302244. doi: 10.1007/978-981-13-0224-4_2.

[27] N. Kerzazi, F. Khomh, and B. Adams. Why Do Automated Builds Break? An Empirical Study. In *2014 IEEE International Conference on Software Maintenance and Evolution*, pages 41–50, September 2014. doi: 10.1109/ICSME.2014.26. ISSN: 1063-6773.

[28] Anders Klint and Vilhelm Åkerström. Continuous Delivery: Challenges, Best Practices, and Important Metrics. *LU-CS-EX*, 2020. ISSN 1650-2884. URL http://lup.lub.lu.se/student-papers/record/9021590.

[29] Henrik Kniberg and Anders Ivarsson. Scaling Agile @ Spotify, October 2012. URL https://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf.

[30] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 426–434, New York, NY, USA, August 2008. Association for Computing Machinery. ISBN 978-1-60558-193-4. doi: 10.1145/1401890.1401944. URL https://doi.org/10.1145/1401890.1401944.

[31] Eetu Kupiainen, Mika V. Mäntylä, and Juha Itkonen. Why are industrial agile teams using metrics and how do they use them? In *Proceedings of the 5th International Workshop on Emerging Trends in Software Metrics*, WETSoM 2014, pages 23–29, New York, NY, USA, June 2014. Association for Computing Machinery. ISBN 978-1-4503-2854-8. doi: 10.1145/2593868.2593873. URL https://doi.org/10.1145/2593868.2593873.

[32] Leon König and Andreas Steffens. Towards a quality model for devops. *Continuous Software Engineering & Full-scale Software Engineering*, page 37, 2018.

[33] Ming-Chang Lee. Software Measurement and Software Metrics in Software Quality, 2013. URL /paper/Software-Measurement-and-Software-Metrics-in-Lee/80c92c609d0d6cfd3c959d69eba4f98b6d2a3a46.

[34] T. Lehtonen, Sampo Suonsyrjä, Terhi Kilamo, and T. Mikkonen. Defining metrics for continuous delivery and deployment pipeline. In *SPLST*, 2015.

[35] Steven Levy. Mark Zuckerberg on Facebook's Future, From Virtual Reality to Anonymity. *Wired*, April 2014. ISSN 1059-1028. URL https://www.wired.com/2014/04/zuckerberg-f8-interview/. Section: tags.

[36] Iraj Lohrasbinasab, Prameet Bhakta Acharya, and Ricardo Colomo-Palacios. BizDevOps: A Multivocal Literature Review. In Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Chiara Garau, Ivan Blečić, David Taniar, Bernady O. Apduhan, Ana Maria A. C. Rocha, Eufemia Tarantino, Carmelo Maria Torre, and Yeliz Karaca, editors, *Computational Science and Its Applications – ICCSA 2020*, Lecture Notes in Computer Science, pages 698–713, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58817-5. doi: 10.1007/978-3-030-58817-5_50.

[37] Lucy Ellen Lwakatare, Pasi Kuvaja, and Markku Oivo. Relationship of DevOps to Agile, Lean and Continuous Deployment. In Pekka Abrahamsson, Andreas Jedlitschka, Anh Nguyen Duc, Michael Felderer, Sousuke Amasaki, and Tommi Mikkonen, editors, *Product-Focused Software Process Improvement*, Lecture Notes in Computer Science, pages 399–415, Cham, 2016. Springer International Publishing. ISBN 978-3-319-49094-6. doi: 10.1007/978-3-319-49094-6_27.

[38] Kalyan Chakravarthy Maddila. Potential metrics for Agile and Lean : Systematic Literature Review and Survey. Master's thesis, Blekinge Institute of Technology, Karlskrona, 2015. URL http://urn.kb.se/resolve?urn=urn:nbn:se:bth-1916.

[39] Ministerie van Volksgezondheid, Welzijn en Sport. New measures to stop spread of coronavirus in the Netherlands - News item - Government.nl, March 2020. URL https://www.government.nl/latest/news/2020/03/12/new-measures-to-stop-spread-of-coronavirus-in-the-netherlands. Last Modified: 2020-04-14T17:08 Publisher: Ministerie van Algemene Zaken.

[40] M. J. Ordonez and H. M. Haddad. The State of Metrics in Software Industry. In *Fifth International Conference on Information Technology: New Generations (itng 2008)*, pages 453–458, April 2008. doi: 10.1109/ITNG.2008.106.

[41] J.W. Paulson, G. Succi, and A. Eberlein. An empirical study of open-source and closed-source software products. *IEEE Transactions on Software Engineering*, 30(4): 246–256, April 2004. ISSN 1939-3520. doi: 10.1109/TSE.2004.1274044. Conference Name: IEEE Transactions on Software Engineering.

[42] Thomas V Perneger. What's wrong with Bonferroni adjustments. *BMJ : British Medical Journal*, 316(7139):1236–1238, April 1998. ISSN 0959-8138. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1112991/.

[43] Luís Prates, João Faustino, Miguel Silva, and Rúben Pereira. DevSecOps Metrics. In Stanisław Wrycza and Jacek Maślankowski, editors, *Information Systems: Research, Development, Applications, Education*, Lecture Notes in Business Information Processing, pages 77–90, Cham, 2019. Springer International Publishing. ISBN 978-3-030-29608-7. doi: 10.1007/978-3-030-29608-7_7.

[44] Akond Rahman, Amritanshu Agrawal, Rahul Krishna, and Alexander Sobran. Characterizing the influence of continuous integration: empirical results from 250+ open source and proprietary projects. In *Proceedings of the 4th ACM SIGSOFT International Workshop on Software Analytics*, SWAN 2018, pages 8–14, New York, NY, USA, November 2018. Association for Computing Machinery. ISBN 978-1-4503-6056-2. doi: 10.1145/3278142.3278149. URL https://doi.org/10.1145/3278142.3278149.

[45] Islem Saidani, Ali Ouni, Moataz Chouchen, and Mohamed Wiem Mkaouer. Predicting continuous integration build failures using evolutionary search. *Information and Software Technology*, 128:106392, December 2020. ISSN 0950-5849. doi:

10.1016/j.infsof.2020.106392. URL http://www.sciencedirect.com/science/article/pii/S0950584920301579.

[46] C. B. Seaman. Qualitative methods in empirical studies of software engineering. *IEEE Transactions on Software Engineering*, 25(4):557–572, July 1999. ISSN 1939-3520. doi: 10.1109/32.799955. Conference Name: IEEE Transactions on Software Engineering.

[47] Marina Sokolova, Nathalie Japkowicz, and Stan Szpakowicz. Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation. In Abdul Sattar and Byeong-ho Kang, editors, *AI 2006: Advances in Artificial Intelligence*, Lecture Notes in Computer Science, pages 1015–1021, Berlin, Heidelberg, 2006. Springer. ISBN 978-3-540-49788-2. doi: 10.1007/11941439_114.

[48] C. Spearman. The Proof and Measurement of Association between Two Things. *The American Journal of Psychology*, 100(3/4):441–471, 1987. ISSN 0002-9556. doi: 10.2307/1422689. URL https://www.jstor.org/stable/1422689. Publisher: University of Illinois Press.

[49] Anselm L. Strauss. *Qualitative Analysis for Social Scientists*. Cambridge University Press, Cambridge, 1987. ISBN 978-0-521-33806-6. doi: 10.1017/CBO9780511557842. URL https://www.cambridge.org/core/books/qualitative-analysis-for-social-scientists/1EBB3B490B28C39D7A33EB12A58B211B.

[50] Bogdan Vasilescu, Yue Yu, Huaimin Wang, Premkumar Devanbu, and Vladimir Filkov. Quality and productivity outcomes relating to continuous integration in GitHub. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, ESEC/FSE 2015, pages 805–816, New York, NY, USA, August 2015. Association for Computing Machinery. ISBN 978-1-4503-3675-8. doi: 10.1145/2786805.2786850. URL https://doi.org/10.1145/2786805.2786850.

[51] Krzysztof Wnuk and Kalyan Chakravarthy Maddila. Agile and lean metrics associated with requirements engineering. In *Proceedings of the 27th International Workshop on Software Measurement and 12th International Conference on Software Process and Product Measurement*, IWSM Mensura '17, pages 33–40, New York, NY, USA, October 2017. Association for Computing Machinery. ISBN 978-1-4503-4853-9. doi: 10.1145/3143434.3143437. URL https://doi.org/10.1145/3143434.3143437.

[52] Xinhao Wu and Maike Zhang. An empirical assessment of the predictive quality of internal product metrics to predict software maintainability in practice. Master's thesis, Blekinge Institute of Technology, Karlskrona, 2020. URL http://urn.kb.se/resolve?urn=urn:nbn:se:bth-20149.

[53] Y. Yu, H. Wang, V. Filkov, P. Devanbu, and B. Vasilescu. Wait for It: Determinants of Pull Request Evaluation Latency on GitHub. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*, pages 367–371, May 2015. doi: 10.1109/MSR.2015.42. ISSN: 2160-1860.

[54] Yue Yu, Gang Yin, Tao Wang, Cheng Yang, and Huaimin Wang. Determinants of pull-based development in the context of continuous integration. *Science China Information Sciences*, 59(8):080104, July 2016. ISSN 1869-1919. doi: 10.1007/s11432-016-5595-8. URL https://doi.org/10.1007/s11432-016-5595-8.

[55] Y. Zhao, A. Serebrenik, Y. Zhou, V. Filkov, and B. Vasilescu. The impact of continuous integration on other software development practices: A large-scale empirical study. In *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 60–71, October 2017. doi: 10.1109/ASE.2017.8115619.

Appendix A

Frequencies of codes extracted from the interviews

Code	code frequency
Maturity	24
Source of KPIs	18
Differentiation	11
Manual data collection	8
Distribution of responsibility	8
KPIs as a driver	8
Uptime	7
Selection of KPIs	7
Automated data collection	7
Understanding purpose of KPIs	6
Sources of KPIs	6
Agility metrics	5
Number of changes	5
Organizational structure	5
User satisfaction	5
Availability	5
Compliancy	5
Offered services	5
Relation to ING	5
Testing — Pillars	5
Business value	4
Reason for existence	4
Single central storage	4
Under developement	4
Cheating	4
Data collection — Challenge	4
Customer definition	4

Deployment frequency — Relevance	4
MTTR	4
Objectives	4
Impact	4
Differences between tools	4
Importance of KPIs	4
Regulations	4
Definition of customers	4
Function of KPIs	4
Use of proxies	4
Deployment frequency	4
Risk KPI	4
Users	4
Performance	4
Interpretation of DORA	3
DORA metrics	3
Throughput	3
Standardized reporting	3
Duration	3
Relation between release frequency and nr. changes	3
Relation between MTTR and lead time	3
Data-driven	3
Standardized metrics	3
Relation between KPIs and metrics	3
Availability — Evolution	3
Monitoring platform	3
Transition of platforms	3
PSD2	3
Assessment	3
Project performance	3
Adoption	3
Cheating metrics	3
REDACTED	2
Deployment frequency — Agility	2
Production	2
Opinion of DORA	2
Automated data gathering	2
Importance of capability	2
Objective	2
Auditability	2
Importance of automation	2
Automation KPI	2
Differentation	2

Assesment	2
Importance	2
Reasons for using nr. of incidents	2
e2e testing	2
Reliability	2
Downside of KPIs	2
Downside of automation	2
Release train instead of CD	2
Example	2
Long term direction	2
MTTF	2
Advantage of automation	2
Adoption KPI	2
Event driven	2
Risk of metrics	2
Success rate — subject	2
Human aspect	2
Purpose definition	2
REDACTED	2
Metrics as a goal	2
Connection to infra	2
Standardizing	2
Opinion on thesis assumptions	2
Hierarchy of metrics	2
Interrest	2
Case-to-case differences	2
Cost	2
REDACTED	2
REDACTED	2
Biggest user	2
Quality of data	2
Batch size in release train	2
Latency	2
DORA tradeoff	2
DORA tradeoff context specific	2
Indicating underlying problem	2
Latency — subject	2
Setting priorities	2
Independent auditors	2
Availability — Nuance	2
Availability — Example	2
Increase in adoption	2
Usage	2

Incident	2
Organization	1
Optimization	1
Organizational wide	1
Most important	1
Opinion on mainstream role of KPIs	1
Opinion on large change process	1
MTTR — Importance	1
MTTR — Maturity	1
MTTR — Source of KPI	1
Manual correction	1
Manual data collection — Drawback	1
Manual upgrades	1
Measurable end to end	1
Measuring impact is challenging	1
Monitoring	1
REDACTED	1
Prioritization	1
Not entirely deprecated	1
Number of deployments	1
Number of incidents	1
REDACTED	1
Preventive	1
Achieving goals	1
Privacy	1
Protecting employees	1
Similarity	1
Size of organization	1
Sliding window	1
Software delivery performance	1
Source of KPI data is difficult	1
Standard metric	1
Standardized	1
Started with throughput metrics	1
State of the organization	1
Struggle	1
Success rate	1
Theory vs. Reality	1
Timeliness KPI	1
Uptime — Definition of asset	1
Uptime — Objective	1
Uptime — calculation	1
Uptime — definition	1

User feedback	1
Visualization	1
Work field	1
Work in progress	1
project orientation	1
source of KPI data	1
Setting objectives	1
Self-monitoring	1
Role of pipeline	1
Relevance	1
Providing capabilities	1
Purpose	1
Purpose of control	1
Purpose of pipeline	1
Reach	1
Reason for using metrics	1
Relation batch size deployment frequency	1
Relation between DORA and lean	1
Relation to industry	1
Release frequency	1
Reliable metrics	1
Robustness	1
Reporting	1
Resilence	1
Respond to incidents in time	1
Responsibility	1
Retrospective	1
REDACTED	1
Risk	1
Risk controls	1
Risk profile KPI	1
Risks vs downsides when using metrics	1
MTTR — Definition	1
Importance of standardization	1
MTTF — relation to MTTR	1
Data collection — struggle	1
Considerations	1
Contract based testing	1
Control	1
REDACTED	1
Cost of employees	1
REDACTED	1
Customer experience	1

DORA	1
DORA coverage	1
DORA metrics as origin	1
DORA relation to project performance	1
Damage KPI	1
Data collection — Reliability	1
Data protection	1
Difference between project and software delivery performance	1
Data source	1
Decommissioned	1
Definition of purpose	1
Definition of work	1
Demand	1
Dependencies between APIs	1
Dependency	1
Dependency on tools	1
Deployment frequency — Business value	1
Deployment frequency — Example	1
Deployment frequency — Not reliable	1
Deployment success	1
Deployment time	1
Considerate damage	1
Consequence of not controlling risk	1
Compliancy — Challenge	1
Compatibility	1
Advantage of contract based APIs	1
Agility	1
Alignment	1
Application perspective	1
Automatic correction	1
Automation	1
Automation — Definition	1
Automation — Simplification	1
Automation —Risk controls	1
Availability and Scalability — Overlooking	1
Availability and scalability KPI	1
REDACTED	1
REDACTED	1
Availability — Internal	1
Awareness	1
REDACTED	1
Batch size	1
Binary	1

Burndown graph	1
Business value — Hard to measure	1
REDACTED	1
Change process	1
Changing culture	1
Collection	1
Collection of KPIs is difficult	1
Commercial impact	1
Commits per deployment	1
Development cycle	1
Different rules	1
MTTF — Definition	1
Informal discussions	1
Highly integrated	1
Human connection	1
Ideal architecture	1
Impact of compiancy	1
Implication of people cheating metrics	1
Importance of availability	1
Importance of microservices	1
Importance of quality of data	1
Adapt to customer	1
Importance of tracebility	1
Improvement of tracebility	1
Improving	1
Increase innovation and experimentation	1
Insights	1
Diversity	1
Interest	1
Interruption	1
Judgement	1
KPI on adoption	1
KPI selection	1
KPIs on throughput	1
KPIs relate to objectives	1
Knowledge exchange	1
Lead time	1
Lead time for changes	1
Level of formality	1
Limitation of DORA metrics	1
Limits of KPIs	1
REDACTED	1
Granularity	1

REDACTED	1
Global platform	1
Domain definition	1
Downtime — Decision making	1
Downtime — Differentiation	1
Downtime — granularity	1
Downtime — planned	1
Employee happyness	1
Estimate impact	1
Evolution of KPIs	1
Evolution of pipeline	1
Exception	1
Expectation of DORA	1
Expectations of customers	1
Explaining metrics	1
REDACTED	1
False positive	1
Feedback KPI	1
Feedback as inspiration	1
Feedback can lead to debate	1
Feedback goes beyond KPIs	1
Feedback is valuable	1
Feedback — Customers	1
Finance KPI	1
Focus of department	1
Fragmented pipeline	1
Future interest	1
Giving feedback KPI	1
Global continues delivery pipeline	1
Throughput KPI	1

Table A.1: Frequency counts of the codes extracted from the interviews. Some entries have been redacted to maintain confidentiality.