Factors Affecting Cloud Infra-Service Development Lead Times: A Case Study at ING

Huijgens, Hennie; Greuter, Eric; Brons, Jerry; Doorn, Evert A. van; Papadopoulos, Ioannis; Martinez, Francisco Morales; Aniche, Maurício; Visser, Otto; van Deursen, Arie

Publication date
2019

Document Version
Accepted author manuscript

Published in
Proceedings of the International Conference on Software Engineering (ICSE)

Factors Affecting Cloud Infra-Service Development Lead Times: A Case Study at ING

Hennie Huijgens, Eric Greuter, Jerry Brons, Evert A. van Doorn
ING
Amsterdam, The Netherlands
hennie.huijgens,eric.greuter, jerry.brons,evert-jan.van.doorn@ing.com

Ioannis Papadopoulos*, Francisco Morales Martinez*
Delft University of Technology
Delft, The Netherlands
ioannis.papadopoulos@ing.com, f.j.moralesmartinez@student.tudelft.nl

Mauricio Aniche, Otto Visser, Arie van Deursen
Delft University of Technology
Delft, The Netherlands
m.f.aniche,o.w.visser, arie.vandeursen@tudelft.nl

***Abstract*—The development of Cloud Infra-Services has shifted over the past decade in the direction of a software code development process, also known as infrastructure as code (IaC). Contemporary continuous delivery settings in industry require fast feedback. As a consequence, companies need insight in time spent, especially in the development of such services. We examine a series of 28 Cloud Infra-Services within ING, and explore which factors affect their overall time to market and development time. An initial perception among several stakeholders in the Cloud Infra-Service development process, that Cloud Infra-Services within ING take longer than those in peer companies, is not confirmed by our benchmark. Development team members identified the time to internal market of services to be affected negatively by the portal where consumers can order a service and the Orchestration Workflows and by team dynamics. This perception is supported by additional metrics. We propose that promising ways to reduce lead time include reducing the complexity of the ING environment, by treating Cloud Infra-Services like regular software deliveries and by reducing the dependencies between teams in terms of tooling and collaboration.**

***Index Terms*—Cloud Infra-Services, Infrastructure as Code, Virtual Machine, SaaS, PaaS, IaaS, Continuous Delivery, ING**

I. Introduction

Cloud computing is a widely used model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services). This model can be rapidly provisioned and released with minimal management effort or service provider interaction [1] [2]. Cloud environments commonly consist of services, which are increasingly being developed entirely as code. In this study, we apply a software development approach to these deliveries. We focus on Cloud Infra-Services: services that enable the automated deployment of infrastructure [1]. If an organization wants to release Cloud Infra-Services rapidly, it is crucial that it knows which factors affect the time needed to develop these services. However few, if any, studies provide guidance on this subject. In this paper, we explore factors that affect the time to internal market and the development time of services related to infrastructure.

A cloud may contain various types of services. These services may include pieces of infrastructure, that users may order in the cloud (e.g. a database, a virtual machine with an OS, a network component). Such services can be developed as code using ways of working like continuous delivery, test-driven development, Dev/Ops and build/deployment automation, in order to automate as many of the parts of their life-cycle as possible [3] [4]. They are generally referred to as *infrastructure as code* (IaC) [5] [6]. IaC services can be divided into three types [1]: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).

For *Software as a Service* services, the capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser, or a program interface (e.g. Oracle Database as a Service).

For *Platform as a Service (PaaS)* services, the capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider (e.g. Microsoft SQL 2016 server stacks, Linux Redhat server stacks, or GlusterFS patterns).

For *Infrastructure as a Service* services, the capability provided to the consumer is to provision processing, storage, networks and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications (e.g. network components, or private networks).

As limited data was available for IaaS cloud infra services, the focus of our study is an analysis and benchmark of a series of SaaS and PaaS Cloud Infra-Services. To explore the time to market of such Cloud Infra-Services and the factors that affect it, we examine such services developed in the private cloud platform of ING, a Dutch, globally operating bank.

A. Background

ING is in the midst of a shift from finance-oriented to engineering-driven company. The infrastructure department of ING - ING Tech Infra - delivers the global digital self-service IT Infra platforms, to enable the bank to unite and operate as one. For the services that Tech Infra provides, virtualization of environments and infrastructure play a decisive role in providing information to customers and employees.

*Work completed during an internship at ING.

In recent years, ING implemented a fully automated release engineering pipeline for its software engineering activities. This pipeline facilitates more than 600 teams, that perform more than 2500 deployments to production each month on over 750 different applications. The pipeline is based on a model described by Humble and Farley [7] - and is known within ING as *CDaaS*, an abbreviation of *Continuous Delivery as a Service*. Within CDaaS, ING created two pipelines for their main technology platforms Windows and Linux.

One main goal of CDaaS is to support teams in maximizing the benefits of shared use of tools. The mindset behind CDaaS is to go to production as fast as possible, while maintaining or improving quality, so teams get fast feedback, and know they are on the right track. It forms the core of an ongoing transition within ING towards BizDevOps, a model were software developers, business staff, and operations staff work together in one small, agile team. The idea behind this is that such teams can develop software more quickly, be more responsive to user demand, and ultimately maximize revenue.

ING Tech Infra delivers its infrastructure products through a private cloud platform where consumers can order services, known as ING Private Cloud (the so-called IPC-portal). ING has decided to build its own private cloud, to comply with regulations in the financial sector. Private cloud refers to a model of cloud computing where IT services are provisioned over private IT infrastructure for the dedicated use of a single organization [1].

With IPC, ING controls the global pipeline of its infra-deliveries through four stages: Development, Test, Acceptance, and Production. In this study, we focus on the Cloud Infra-Services that are currently in Production. This means that an engineer in a BizDevOps team can order a Cloud Infra-Service from a web portal known as the IPC portal. By doing so, a part of the cloud infrastructure specifically developed to deploy a Cloud Infra-Service automatically deploys an instance of the service that is ready for use. We explore which factors affect the time to internal market and development time of the full Cloud Infra-Service, including these automated deployment processes.

B. Problem Statement

Because ways of working like continuous delivery and Dev/Ops specifically require short iteration times, we are interested in examining how long it takes for a Cloud Infra-Service to be developed, from the moment a vendor releases it as a product to the moment customers can order it within ING. Our exploration resolves around the following questions:

RQ1: How does development time of the examined Cloud Infra-Services compare to other companies?

RQ2: What factors affect the time to internal market and development time of Cloud Infra-Services in continuous delivery settings?

RQ3: What actions can be taken to decrease time to internal market and development time of Cloud Infra-Services?

We use converging methods to answer these research questions, and aim to make the following contributions:

1) We propose a lightweight measuring technique of Cloud Infra-Services in a continuous delivery setting, based on a proven model for benchmarking software delivery portfolios [8].
2) We gather data on 28 deployed Cloud Infra-Services, and map these deliveries on a model for internal and external benchmarking purposes in order to identify good and bad deliveries.
3) We report a set of additional metrics related to usage, complexity and reliability of services once they have been deployed, to explore if they correlate with time to internal market and development time of the Cloud-Infra Service.
4) We survey stakeholders in the Cloud Infra-Service development process, to identify factors that influenced IPC PaaS and SaaS Cloud Infra-Services' internal time to market and development time.

The remainder of this paper is structured as follows. In Section II related work is described. Section III outlines the research design. The results of the study are described in Section IV. We discuss the results in Section V, and finally, in Section VI we make conclusions and outline future work.

II. Related Work

Cloud computing is a paradigm to deliver IT services as computing utilities, which run on data centers. It is enabled by advances in virtualization in computing, storage, and networking [9] [10]. Clouds provide users with services. Among other things, such services can be used to construct highly customized, software-defined environments that can support dynamic and data-driven applications. To the extent that they support deployments of services to consumers, such services can provide infrastructure [11]. Cloud computing and service oriented computing have a number of challenges. For example, Wei and Blake [12] identify maintaining high service availability, providing end-to-end secure solutions, and managing longer-standing service workflows. They also mention opportunities, such as service discovery through federated clouds, rapid service deployment, and agent-mediated ontology generation from co-located information.

To address challenges related to cloud computing, authors have proposed benchmarks at several levels of abstraction. For example, focusing on the deployment process, benchmarks have been proposed for *deployment methods* and *management platforms* for cloud services (e.g. [13] [14] [15] [16] [17]). Focusing on development, Palesandro et al. describe how Infrastructure as Code (IaC) emerges as a key enabler for cloud services, to manage infrastructure configurations. However, the complexity of the infrastructure life-cycle, the diverse resources that infrastructure configurations consist of, and demand for user-customizations complicate application of their approach [18]. More importantly, both methods fail to distinguish *build* and *delivery* phases of infrastructure services.

Scheuner et al. developed a benchmarking approach for IaaS deliveries [13], and introduced Cloud WorkBench (CWB) in [14]. They presented their results of a large-scale cloud evaluation analyzing more than 33,000 measurements in [15]. Bhattacharjee et al. developed CloudCAMP, a Model-driven Generative Approach for Automating Cloud Application Deployment and Management [19]. Additionally, Scheuner and Leitner describe a new execution methodology that combines micro and application benchmarks into a benchmark suite called RMIT Combined [16]. Although more specific, these benchmarks do not distinguish infra services from non-infra services, limiting their usefulness for our current exploration.

To benchmark the performance of SaaS and PaaS Cloud Infra-Services within IPC to a representative set of available data, we chose to use a software development-based model, known as the *Evidence-Based Software Portfolio Management-model* (EBSPM-model) [8] [20]. The EBSPM-model focuses on benchmarking software delivery portfolios. It is built on a repository of more than 500 finalized software deliveries in four different companies (two banking companies, one telecom company, and one billing software company). Using this model allows us to view the entire development cycle of a Cloud Infra-Service, and compare with similar deliveries in other companies on three key metrics.

III. Research Design

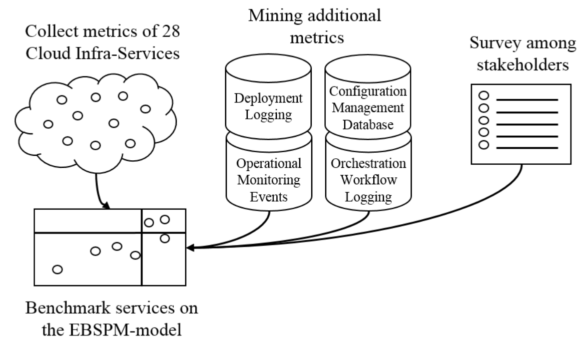
To better understand which factors affect lead time of Cloud Infra-Services — in terms of time to internal market, and components from which it is built, such as decision time, development time, and time before start of development — we use an exploratory mixed case study design consisting of the four steps depicted in figure 1. We will first describe our sample, and then discuss each step in turn.

A. Experimental Context

At the time of writing there are 38 Cloud Infra-Services available in Production in the IPC-portal. Most services are based on a vendor product (e.g. Red Hat Enterprise Linux 7 v1.0.8 for the RHEL7 delivery). They are also characterized by a platform (Linux or Windows). Upon deployment, an instance of the Cloud Infra-Service is automatically created in IPC by its cloud infrastructure, and registered as a configuration item in the configuration management database (CMDB). Such instances can have a variety of types (pattern, virtual machine with or without operating system, physical machine) and may have relations with middleware and / or applications as needed.

Within ING, teams are responsible for the delivery of each Cloud Infra-Service. These teams work agile, led by a Product Owner (a person responsible for the business value of the team). Teams usually work in close collaboration with other teams to create a service. We focused on a subset of 28 SaaS and PaaS Cloud Infra-Services that can be ordered directly in the IPC-portal (we excluded IaaS services from our scope due to the limited availability of data). An overview of services in scope can be found in the technical report [21].

Fig. 1. Overview of the Research Approach.



B. Collection of Metrics for Cloud Infra-Services

In order to plot each Cloud Infra-Service into the EBSPM model [8] [20], we collect three metrics: (1) lead time (e.g. development time, time to internal market), (2) effort (e.g. man hours spent, cost of a delivery), and (3) functional size (the latter being included as a normalizer). We do so by conducting open interviews with the Product Owner for each Cloud Infra-Service, asking them to provide (1) and (2). Point (3) is measured by one of the principal researchers, by counting function points in the IPC portal environment. We count functional size based on functionality delivered by the IPC portal itself, according to IFPUG guidelines [22].

C. Benchmark Cloud Infra-Services

We plot the Cloud Infra-Services collected in the former step on the EBSPM-model. The results of this step are (1) a *research repository* with basic metrics of the services over time, and (2) an inventory of *good practice Cloud Infra-Services* (services that performed better than average on both Development Time and Cost) and *bad practice Cloud Infra-Services* (services that performed worse than average on both Development Time and Cost). We discuss the resulting plot and metrics in the next section.

D. Mining of Additional Metrics

The benchmarking metrics discussed above relate to the time necessary to build cloud *infra* services, services that enable the automated deployment of each cloud service. We seek to explore whether *post-deployment* characteristics of the Cloud Infra-Services, particularly *usage*, *complexity* and *reliability*, can be related to cost, development time and functional size (as used in our benchmarking procedure). Given the exploratory nature of this study, we do not have specific hypotheses with regards to influence of these metrics on the performance of Cloud Infra-Services.

We measure usage as the number of deployments of configuration items with a specific Cloud Infra-Service within IPC overall and within the past year, and the total amount of configuration items that were active during the past year (configuration items which were decommissioned were included if their time of decommissioning within the past year). Complexity refers to the duration of the deployment workflow of the Cloud Infra-Service, the number of deployment steps needed for that service in the main orchestration layer, and the number of workflow orchestration tools used in deploying the service. Reliability refers to the number of monitoring events registered by ING's automated event monitoring per Cloud Infra-Service, averaged over configuration items.

To derive the monitoring metrics, we combine data from the *deployment registry* (all configuration items that were deployed within IPC since its launch), the *configuration management database*, the *event monitoring datawarehouse* (registrations of events on configuration items generated by automated monitoring and logging processes) and the *deployment orchestration logging* (all of the workflow steps invoked by the central orchestration layer within ING Infra). We duplicate entries for configuration items in both registries, and check assignment of configuration items to Cloud Infra-Services with experts within ING. We then subset monitoring and workflow logging data when appropriate (e.g. to select successful deploys or events with certain severity), and use timestamp data to infer whether a configuration item was active during the time period, and to isolate the deployment steps in the logged workflow data. Refer to [21] for a more detailed description of the procedure and script.

E. Survey Among Cloud Infra-Service Stakeholders

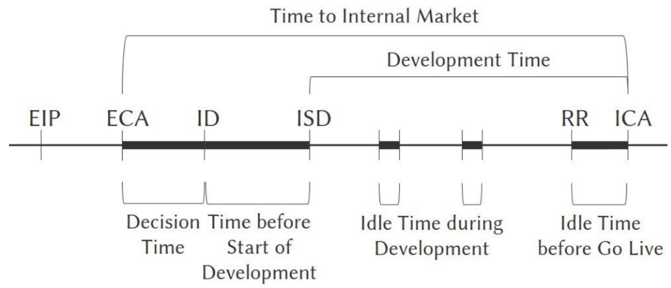
We want to measure which factors the members of the teams that develop the Cloud Infra-Services identify as affecting the time to internal market of these deliveries. To that end we conduct a survey, which focuses on three parts: the duration of the development of the Cloud Infra-Service, idle time prior to the start of development, and the perceived complexity of a Cloud Infra-Service.

In the survey, we first ask which Cloud Infra-Service the stakeholder is most involved in developing, and what his or her role in the development process is. We ask about 11 aspects of the development process that could affect time to market, as depicted below in Table III. These 11 aspects derive from discussion sessions with Product Owners of a variety of Cloud Infra-Services within IPC, which were aimed at identifying a typology of steps that can generically be said to be taken in the development of IPC Cloud Infra-Services.

Each of the 11 aspects is addressed in a survey question that asks to what extent a respondent agrees with a statement, on a 1 to 5 point Likert-scale (strongly agree - agree - neutral - disagree - strongly disagree - don't know). Each survey question is accompanied by the follow-up question "Can you please explain the choice you made to us?" See the technical report [21] for a detailed overview of the survey questions.

We sent the electronic survey to 275 members of ING Tech Infra squads that were involved in one or more Cloud Infra-Services in scope of this study; 28 responses were obtained on 28 unique services. We did not offer any reward to increase the participation in the survey. Based on the responses, we calculated several indicators in order to interpret the results of the survey. Note that the first three are measures of the central tendency, CV is a measure of variability.

Fig. 2. Overview of Timelines in Cloud Infra-Services.



Explanation of abbreviations: *EIP* (External Information on Product); the date when the first information of a product is made available by a vendor. *ECA* (External Consumer Availability); the date when a product is made generally available for consumers by a vendor. *ID* (Internal Decision); the date when a decision was made to start developing a Cloud Infra-Service. *ISD* (Internal Start of Development); the date when a Cloud Infra-Service development team put the first user story in the backlog management system into a sprint. *RR* (Ready for Release); the date when a complete productized build was ready according to its Definition of Done. *ICA* (Internal Customer Availability); the date when a Cloud Infra-Service became generally available for internal consumers on the IPC portal.

1) *Percent Agree* or *Top-2-Box*; the percentage respondents that agreed or strongly agreed.
2) *Top-Box*: the percentage respondents that strongly agreed.
3) *Net-Top-2-Box*; the percentage respondents that chose the top 2 bottom responses subtracted from the top-2 top responses.
4) *Coefficient of Variation* (CV); also known as relative standard deviation; the standard deviation divided by the mean. Higher CV-values indicate higher variability.

We also code the free format text from the surveys to examine whether the provided responses confirmed observations from the survey analysis. We do so using an open card sort [23] with three phases. In the *preparation phase*, we create cards for each survey question commented on by the respondents. In the *execution phase*, cards are sorted into meaningful groups with a descriptive title. Finally, in the *analysis phase*, abstract hierarchies are formed in order to deduce general categories and themes. Our card sort is open, meaning we have no predefined groups; instead, we let the groups emerge and evolve during the sorting process. We apply a number of subsequential steps in the card sort. The fifth author tags the first half of answers. The sixth author tags the second half. Results are reviewed and discussed in a group discussion with the other authors.

IV. RESULTS

We report results from 1) the analysis of collected Cloud Infra-Services, 2) the benchmarking of the services, 3) the analysis of additional metrics, and 4) the survey performed among stakeholders of the services in scope.

A. Inventory of Cloud Infra-Services

We recorded the collected data as described in Section III in a repository. Figure 2 gives an overview of applicable time

TABLE I
TIMELINE STATISTICS

	Time to Internal Market	Development Time	Decision Time	Time before SoD
Count	28	28	12	14
Max	36	16	26	17
Mean	10.12	6.96	5.58	2.64
Median	6.28	6	3	1
Min	3	3	0	0
Standard Deviation	7.79	3.45	4.49	7.75

Count indicates the number of deployments that are included in the metric above. Not for all deployments metrics were found related to Decision Time and Time before Start of Development. The measurements Max, Mean, Median, and Min are in months.

lines for infra-services within ING, such as Time to Internal Market and Development Time. As can be seen in Table I, the Development Time of Cloud Infra-Services varied from 3 to 16 months, with an average duration of 6.96 months. Two types of idle time occur. First, Decision Time - the time between a product being available for consumers and the internal decision taken within ING to start a project - varied from 0 to 26 months, with an average of 5.58 months. Second, Time before Start of Development - the time between the a decision by ING to start a project and the actual start of development - varied from 0 to 17 months, with an average of 2.64 months. The other expected two types of idle time (during development and before go live) could in theory also occur, but interviewees found it difficult to provide accurate information on them.

B. Benchmark Cloud Infra-Services

We mapped the 28 services on the EBSPM-model [8] [20], with Development Time projected on the vertical axis, and cost projected on the horizontal axis, as shown in Figure 3. The EBSPM-model is based on open source repository holding data of approx. 500 software projects [24], of which 352 in several banking companies and business domains comparable to ING. It can therefore be used as a reliable benchmark in our study.

Revisiting RQ1 — How does lead time of the examined Cloud Infra-Services compare to other companies? — shows that on average the subset of 28 services performed 17% better on duration and 41% better on cost than the average of the EBSPM-repository, based on a repository of 500 finalized software deliveries in four comparable companies.

> Observation 1: Our study does not confirm the initial perception among many ING-stakeholders that Cloud Infra-Services within ING take more time than those in peer companies, instead ING services show on average a 17% shorter Development Time than software deliveries in the EBSPM-repository.

The colors of the different Cloud Infra-Services in Figure 3 indicate that services with a longer-than-average Development Time (indicated on the vertical axis) also tend to have a longer Time to Internal Market (indicated by the color range from blue at the top to red at the bottom). Longer Decision Time and Time before Start of Development seem to go together with longer Development Time.

> Observation 2: Taken together, average Decision Time and average Time before Start of Development exceed average Development Time. This suggests that examining the preliminary stage of service development in more detail may yield improvements in lead time.

C. Mining of Additional Metrics

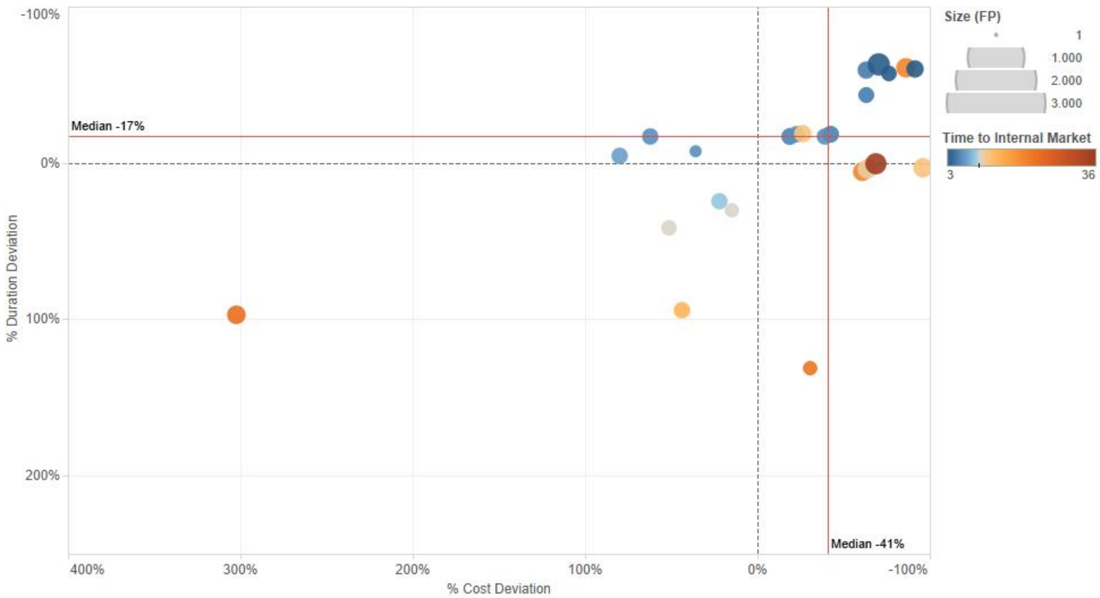
To assess what factors affect the lead time of Cloud Infra-Services in continuous delivery settings (RQ2), we derived metrics for usage, complexity, and reliability of Cloud Infra-Services in the context of IPC. We collected data for 26 deployed services (for two services we were unable to retrieve data). We explored whether correlations exist between usage, complexity, reliability, and benchmarking metrics time to internal market, cost and functional size. For each of these categories of metrics, descriptive statistics are included in the technical report [21].

1) Correlations between metrics: Given our relatively small sample size (26 data points), calculation of a correlation matrix may suffer from low statistical power, which may lead to inflated correlation coefficients [25]. However, since our research is exploratory in nature, we report the correlations as possible directions to explore. Due to redundancy of indicators for usage and complexity, we report only the most relevant dimensions here. A more extensive correlation table is included in the technical report [21]. The various reliability metrics correlate highly with each other. Cloud Infra-Services with a higher average of acknowledged events per configuration item also have a higher average of incidents per configuration item and a higher number of severity 5 events per production configuration item, all $r(26) = .59$, all p-values $< .05$. This pattern of results matched what one would expect based on monitoring, where more severe events are formally recognized more often.

> Observation 3: Our study shows that more popular Cloud Infra-Services have workflows that consist of a greater number of components, and Cloud Infra-Services with a longer deployment time register more incidents per configuration instance, on average.

Two correlations between metrics for usage, reliability and complexity are noteworthy. First, the more configuration items were active during the past year, the more components were included in the deployment workflow of the main orchestration tool, $r(26) = .69$, p-value $= .002$. This shows that more frequently used Cloud Infra-Services incorporate more tools in their workflows. Second, the longer a deployment of a Cloud Infra-Service in the main orchestration tool took on average, the more incidents were registered on configuration items for that Cloud Infra-Service, $r = .71$, p-value $= .002$.

Fig. 3. The Cloud Infra-Services in scope of this exploratory study mapped on the EBSPM-model.



In this figure the deployment time of the 28 deliveries from our study is mapped on the EBSPM-model; a model based on a subset of more than 500 finalized software deliveries from five different companies. The figure above shows only the 28 Cloud Infra-Services in scope of this study. Each service is shown as a circle. The larger the circle, the larger the service is (in functional size). Color indicates a longer Time to Internal Market (the more red, the longer; this varies from 3 to 36 months). The position of each service in the matrix represents the Cost and Development Time deviation of the service relative to the benchmark, expressed as percentages. The horizontal and vertical 0%-lines represent zero deviation, i.e. services that are exactly consistent with the benchmark. A service at (0%, 0%) would be one that behaves exactly in accordance with the benchmark; a service at (-100%, -100%) would cost nothing and be ready immediately; and a service at (+100%, +100%) would be twice as expensive and take twice as long as expected from the benchmark.

This shows that Cloud Infra-Services with, on average, longer deployments have more post-deployment incidents registered. We observed no correlations between the benchmarking metrics and usage, complexity and reliability.

D. Survey Results

Our survey on factors that affected development time of Cloud Infra-Services was active during two weeks. During this time, 10.2% of the 275 who were invited to participate responded, yielding 28 completed questionnaires. The respondents include 22 Cloud Infra-Service Engineers (78.6%), 5 Product Owners (17.9%), and 1 Chapter Lead (3.6%) .

The respondents indicated their level of agreement or disagreement towards 11 statements (questions Q03 through Q13 in the survey). They did so on 1 to 5 point Likert-scales, or resorted to an *"I don't know"* option if they were unsure whether the aspect mentioned in the question affected the time to delivery of their infra cloud service. See Table III for descriptive statistics and bar-charts depicting the spread of scores for each survey question.

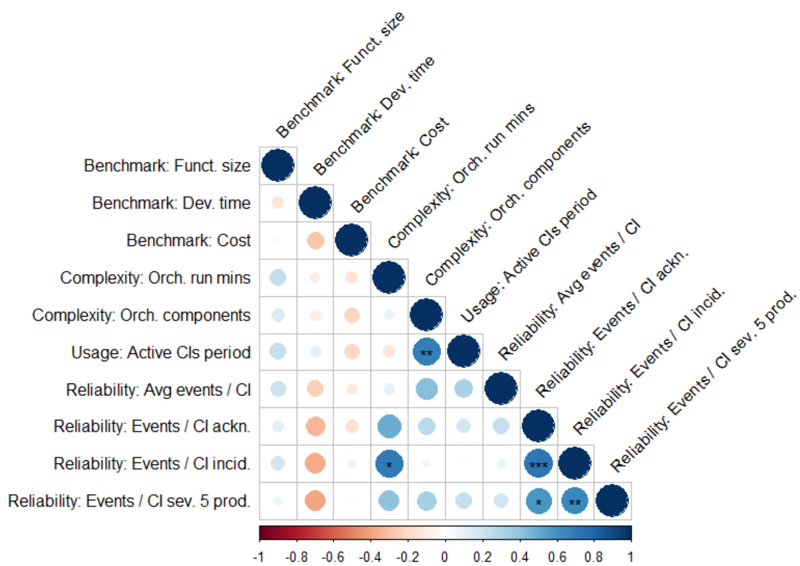
In order to understand any causes behind the survey scores, we coded the free format text resulting from the survey. The most mentioned codes per question are shown in Table II. The analysis of the survey scores and the text coding resulted in the following observations.

1) IPC-portal and Orchestration Workflows: A relatively high number of respondents agreed with the statements that the IPC-portal (Q03) and Orchestration Workflows (Q04) were obstacles for the delivery of the Cloud Infra-Service they worked on (67 and 68 percent agreement, respectively). As one of the respondents put it: "The portal is working very slow and it is annoying" [P07]. On the other hand, Service Delivery aspects (Q09) were considered the least hindering of all measured aspects (13 percent agreement).

TABLE II
MOST MENTIONED CODES PER QUESTION

Question	Code Description
Q3	Setting up IPC portal to consume the new service (9)
	Dependencies on other teams (5)
	Issues with orchestration tools (4)
Q4	Issues with orchestration tools (5)
	Dependencies on other teams (3)
	Complexity of the infra delivery (3)
Q5	Backup capabilities / miss-alignments in requirements (4)
Q6	Second day operations are time consuming (5)
	Complexity of the infra delivery (4)
Q7	Not unified CMDB Models and other issues with them (8)
Q8	The process of risk and security is too complex (9)
Q9	Documentation issues (3)
Q10	Pricing/Charging/License (6)
Q11	Dependencies on other teams (12)
Q12	Bug fixing and testing (10)
	Test Resources (3)
Q13	Decision making (4)
Q14	Pricing/Charging/Licence (4)

Fig. 4. Correlation Matrix of selected Benchmarking, Usage, Complexity and Reliability Metrics.



This correlation matrix depicts correlations between the most important benchmarking, usage, complexity, and reliability metrics for each Cloud Infra-Service in our study. The size of the circle represents the magnitude of the correlation. The *, ** and *** superscripts represent p-values associated with the correlation, of > .05, > .01 and > .005, respectively. The color of the circle represents the direction (blue for positive, red for negative) of the correlation. The correlations depicted are a subset of a correlation matrix containing all metrics. This matrix was corrected for multiple comparisons using a Benjamini Hochberg correction [26]. See the technical report [21] for more details.

2) Second Day Operations: Although answers of both Q06 and Q07 are scattered, respondents say that second day operations — updates of the system after deployment — are time consuming in general, and a lack of unified CMDB models is perceived as an obstacle: "The optional software capabilities should be part of the CDaaS (application) workflow and not of the Cloud capabilities" [P05].

Observation 4: The IPC-portal and the Orchestration Workflows were seen by a large percentage of respondents as negatively affecting time to internal market.

3) Security, Risk, Compliance, and Governance: The process with regard to risk, security and compliance (Q08) is perceived as complex by many respondents: "It takes weeks to set up security scans, pentests, get approval for documents..." [P11]. Governance related aspects (e.g. decision-making, rules & regulations) are not perceived as impediments, as indicated by a Net-Top-2-Box score of 0%.

Observation 5: The process with regard to risk, security and compliance is perceived as complex by a large number of respondents.

4) Service Delivery: Documentation, service component description, service specification, and training are not perceived as obstacles, as indicated by a Net-Top-2-Box score of -52%.

5) Finance and Governance: Financial (Q10) and Governance (Q13) related aspects received more *"I don't know"* answers than the other statements (10 each, in total). It might be the case that respondents are less aware or familiar with these aspects, given their less-technological nature.

V. Discussion

In this study we identify five topics that we see as relevant to establishing the time to internal market of Cloud Infra-Services: data quality considerations, using appropriate benchmarks for projects, reduction of decision time, reduction of dependencies between teams and tools, and assessing the implementation of security measures. We discuss these topics below for each of our research questions, giving attention to considerations with regards to validity where necessary.

A. How does lead time of the examined Cloud Infra-Services compare to other companies? (RQ1)

Our results indicate that with regards to development time, the Cloud Infra-Services within IPC perform 17% better than other deliveries in the benchmark repository. This shows that in terms of cost and development time, ING is doing well. Though ING internal customers may experience the development process as slow, our benchmark suggests other organizations with projects of comparable size do about as well, if not worse in terms of development time.

We based our benchmarks on interview data, which introduces several issues with regards to validity. First, we relied on the memory of the Product Owner to obtain timepoints and estimates of cost and effort. These data were not adequately

TABLE III
OVERVIEW OF THE SURVEY ANALYSIS.

Interview Question	Likert Distribution	Number of Respondents	Percent Agree	Top-Box	Net-Top-2-Box	CV
Q4. Orchestration Workflows related aspects (e.g. Workflows/Automation for Virtual Machine, Operating System, Network, System Accounts, Storage, Configuration Registration) hindered the delivery of <Cloud Infra-Service choice>.		25	68%	20%	44%	31%
Q3. IPC-portal related aspects (e.g. setting up the IPC-portal to consume new service) hindered the delivery of <Cloud Infra-Service choice>.		27	67%	7%	48%	27%
Q11. Team dynamics related aspects (e.g. dependencies on other teams, cultural differences, many team changes, age of teams, difference in expertise) hindered the delivery of <Cloud Infra-Service choice>.		27	63%	30%	37%	31%
Q12. Service Verification and Testing related aspects (e.g. Optimization, Bug Fixing, Test Resources, Test Automation) hindered the delivery of <Cloud Infra-Service choice>.		24	50%	17%	17%	36%
Q7. Operations related aspects (e.g. Monitoring, Configuration Scanning, CMDB Model) hindered the delivery of <Cloud Infra-Service choice>.		22	45%	9%	0%	35%
Q6. Second Day Operations related aspects (e.g. Install optional software, SelfService capabilities) hindered the delivery of <Cloud Infra-Service choice>.		24	42%	13%	0%	35%
Q8. Security, Risk & Compliance related aspects (e.g. OSG, BIA, Risk Assessment, SEM-I, TSCM-I, Vulnerability Scanning, Penetration Testing, Certificate Management) hindered the delivery of <Cloud Infra-Service choice>.		23	39%	13%	4%	36%
Q10. Financial related aspects (e.g. Procurement, License Metering, Pricing & Charging) hindered the delivery of <infra-delivery choice>.		18	39%	17%	-6%	40%
Q13. Governance related aspects (e.g. Decision-making, Rules & Regulations) hindered the delivery of <Cloud Infra-Service choice>.		18	33%	11%	0%	32%
Q5. Stack Definition related aspects (e.g. Capabilities for Backup, APIs, Agents) hindered the delivery of <Cloud Infra-Service choice>.		22	32%	5%	-5%	30%
Q9. Service Delivery related aspects (e.g. Documentation, Service Component Description, Service Description, Service Specification, Training + Instruction Movies) hindered the delivery of <infra-delivery choice>.		23	13%	0%	-52%	34%

Table sorted on percentage agreed. Column 'Likert Distribution' shows a graph of the distribution on a 1-5 point Likert scale for each question with from left to right the values 'Strongly Agree', 'Agree', 'Neutral', 'Disagree', and 'Strongly Disagree'. See the Technical Report for an extended overview of the survey setup and the survey questions.

administered for several projects, and team composition (including the Product Owner role) may have changed during or after development. Our metrics should therefore be seen as rough estimates. Additionally, interviews with Product Owners were conducted by the investigators. This may have affected interview results. We attempted to minimize such effects by asking for factual information and followed a standardized protocol for the interview.

B. What factors affect the lead time of Cloud Infra-Services in continuous delivery settings? (RQ2)

Reviewing our benchmarking data, we saw that a relatively long period of decision time precedes the decision to start developing a solution. On average, decision time spans half a year, with the most extreme case recorded spanning over two years. Development time is equally long but has a smaller spread, suggesting that there is value in examining the decision making process in more detail.

Although we found no significant correlations with deployment time, cost, or functional size, our additional metrics did show that the more a Cloud Infra-Service was used over the past year, the more complex it gets. In itself, this does not say much about development time. However, Cloud Infra-Services with a longer deployment duration (an indicator of complexity) had more incidents occur per CI over the past year. Taken together with the first finding, this suggests an increase in the complexity of Cloud Infra-Services may lead to less reliable Cloud Infra-Services after deployment.

A possible explanation for this problem is that an eventual larger number of configuration items means a greater (anticipated) demand for custom functionality, which complicates Cloud Infra-Service development. Such a complexity-based explanation matches the results of the survey, in which problems with workflow tools are prominently mentioned as factors that impede development time. A second prominent factor mentioned in the survey is collaboration between teams; apparently increased dependencies in tooling go together with increased dependencies in collaboration. In an organization that aims to implement infrastructure as code, what effects such dependencies have seems like an important topic to investigate.

As we have seen in our discussion of RQ1, our benchmarking data suffers from non-response by product owners. With regards to our data mining efforts, we were unable to conclusively verify completeness of data for each system we mined. We could not tie deployments, event monitoring or orchestration logging to specific configuration items for Cassandra Keyspace or Oracle DBaaS resulting in missing data for these infra deliveries. In addition, our sample size was small, making any conclusions based on correlations tentative at best. Moreover, we learnt monitoring is optional for certain classes of configuration item, meaning our event data is likely incomplete. We were also not able to conclusively verify whether monitoring for all Cloud Infra-Services was stored in the datawarehouse we mined. Severity categories do not seem to be used systematically for monitoring. For these reasons, we resolved not to report counts of monitored events without any classification of organizational relevance. Finally, our monitoring data only went back one year, while some infra deliveries were more mature than others. In sum, although we are confident that the conclusions we draw are optimal given the available data, a more systematic approach of data storage with regards to Cloud Infra-Services in both development and deployment would greatly increase ING's ability to draw conclusions regarding the IPC environment. The limited sample did not offer us an opportunity to test the mediating influence of complexity on lead time.

The survey we sent out suffers from two main issues. First, we sought a representative, stratified sample of ING engineers who worked on the Cloud Infra-Services in our study. This was complicated by staffing changes within teams, leading us to e-mail all employees of the Infra department at ING. A list of who worked on which delivery when would have made it easier to target a representative, stratified sample. Additionally, we built our survey to gather information on categories, of which team leaders indicated they were process steps in developing an infra delivery. The extent to which these categories where adequately understood by our respondents may vary from category to category. We were unable to verify the extent to which this was the case.

C. What actions can be taken to decrease lead time of Cloud Infra-Services? (RQ3)

Our results are largely specific to IPC, and do not generalize well to other environments within or outside ING. Yet, based on our answers to RQ1 and RQ2 — as described in Section V.A and V.B — we identify four general take-away messages that may be of general benefit in reducing time-to-market and development time of Cloud Infra-Services:

1) Reduce the complexity of the environment by treating Cloud Infra-Services just like regular software deliveries; e.g. make the use of standardized, automated delivery pipelines (such as CDaaS) mandatory.
2) Do follow-up research into the possibilities to reduce the dependencies of other teams (e.g.: security, workflow orchestration), since this is mentioned by many stakeholder as the biggest obstacle for time-to-market.
3) Ensure good process data quality as a precondition for well-informed decision-making; make the use of a standardized backlog management tool, mandatory from the start of a service (e.g. the creation of an epic) and beyond, and formally track decision moments.
4) Examine the decision-making process more closely; the greatest impact on the time-to-market of Cloud Infra-Services can be realized in the decision-making phase and the period prior to the start of the development.

D. Threats to Validity

Like many applied researchers, we have had to sacrifice experimental control for studying an in vivo phenomenon. In doing so, several factors impacted the validity of our results. We have already discussed several points related to construct validity and internal validity above, in summarizing answers to our research questions.

1) External Validity: The results of this study are based on the current situation with ING Infra. Because of the complexity of the environment and the relatively low levels of standardization in processes and tooling, conclusions from the current study have limited external validity. At the same time, this study yields a number of concepts that were shown to be related to the time of internal deployments. These can be mapped onto other organizations with cloud-based infra services.

2) Study Reliability: As a general note, the infra deliveries we examined were developed over a period of years. This development period spanned several large organizational changes and efforts at restructuring, efforts which were ongoing at the time of this study. Teams changed, with members being reassigned or leaving, and the structure of the infra environment changed. Additionally, changes in the various data sources (particularly event data stored in the monitoring datawarehouse), and the necessity for stakeholder management in a project of this scope make it difficult to repeat this process exactly. However, by scripting our analyses and making them repeatable and documenting all our efforts in detail in the technical report, we have made every effort to enable others to replicate the steps we followed.

3) Incompleteness of monitoring data: Because monitoring data proved incomplete, we did not count numbers of events in isolation. Instead, we focused on the impact of events for ING by counting the events per configuration item that were acknowledged by an operator after being generated by an automated monitoring tool, events assigned an incident number, and events assigned a severity number ranging from 0 (least severe) to 5 (most severe). The choice for these metrics was made by project stakeholders, together with subject matter experts on monitoring. The relevance of the metric for ING and availability of data were used as criteria.

VI. Conclusions

We performed an exploratory case study on 28 Paas and SaaS Cloud Infra-Services deployed at ING Tech Infra, in order to examine how time-to-market of such services can be shortened. We benchmarked 28 Cloud Infra-Services with peer group software deliveries, mining additional metrics from four data sources, and from a survey among stakeholders. Based on these, we propose that time to internal market may benefit from reducing the complexity within which development teams operate, both in terms of tools and dependencies between teams, from a more detailed consideration of the time necessary to reach a decision to start developing, and from more structural registration of Cloud Infra-Service related data.

A. Directions for Future Research

We see this study as offering several interesting directions for future work. First, the software development-based perspective we applied in benchmarking IPC PaaS and SaaS Cloud Infra-Services provides a straightforward way of quantifying the full lead time of an automatically deployed cloud service. We aim to incorporate a more diverse range of metrics into this model in the future, including data on IaaS components, agile team performance, decision making, and idle time. This will lead to a more fine-grained model, that should be generically applicable across organizations.

Second, we see merit in exploring the differences between the development of applications and infrastructure components. In this study, we have assumed that function points are a useful proxy for the functionality of a service. However, Cloud Infra-Services can involve more dependencies between cloud infrastructure components than applications normally have. Such dependencies may not be countable as function points. Future examination could test an adapted version of our benchmarking model, in which functionality indicators are matched to Cloud Infra-Service complexity.

Finally, we have conducted this study in a banking environment. Such environments can be expected to have regulations that go beyond those in other sectors. We hope to use our benchmarking model for Cloud Infra-Services in other sectors, so as to provide a standardized comparison within a more homogeneous population. This will enable more confident conclusions with regards to the performance of Cloud Infra-Service development processes.

Our study provides an exploration into the development of the infrastructure of the ING Private Cloud. We have seen that the development speed of PaaS and SaaS Cloud Infra-Services is on par with a sample of other software deliveries. We have also identified several promising directions that ING can explore to further accelerate the time needed to go from vendor release to customer ready Cloud Infra-Service. We hope our findings will help build the better clouds of tomorrow.

Acknowledgments

The authors would like to thank ING Tech Infra and its employees for their willing contributions to this project.

References

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," *NIST Special Publication 800-145, Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899-8930*, 2011.

[2] OpenGroup, "The Open Group Cloud Ecosystem Reference Model - The Cloud Ecosystem Reference Model," 2018. [Online]. Available: http://www.opengroup.org/

[3] D. Spinellis, "Don't install software by hand," *IEEE Software*, vol. 29, no. 4, pp. 86–87, July 2012.

[4] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano, "Devops," *IEEE Software*, vol. 33, no. 3, pp. 94–100, May 2016.

[5] M. Hüttermann, *Infrastructure as Code*. Berkeley, CA: Apress, 2012.

[6] A. Wittig and M. Wittig, *Amazon Web Services in Action*. Manning Press, 2016.

[7] J. Humble and D. Farley, *Continuous Delivery, reliable software releases through build, test and deployment automation*. Addison-Wesley, 2010.

[8] H. Huijgens, A. van Deursen, and R. van Solingen, "The effects of perceived value and stakeholder satisfaction on software project impact," *Information and Software Technology*, vol. 89, pp. 19 – 36, 2017.

[9] R. Jain and S. Paul, "Network virtualization and software defined networking for cloud computing: a survey," *IEEE Communications Magazine*, vol. 51, no. 11, pp. 24–31, November 2013.

[10] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities," in *2008 10th IEEE International Conference on High Performance Computing and Communications*, Sept 2008, pp. 5–13.

[11] M. Abdelbaky, J. Diaz-Montes, M. Unuvar, M. Romanus, I. Rodero, M. Steinder, and M. Parashar, "Enabling Distributed Software-Defined Environments Using Dynamic Infrastructure Service Composition," in *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, May 2017, pp. 274–283.

[12] Y. Wei and M. B. Blake, "Service-Oriented Computing and Cloud Computing: Challenges and Opportunities," *IEEE Internet Computing*, vol. 14, no. 6, pp. 72–75, Nov 2010.

[13] J. Scheuner, J. Cito, P. Leitner, and H. Gall, "Cloud workbench: Benchmarking iaas providers based on infrastructure-as-code," in *Proceedings of the 24th International Conference on World Wide Web*, ser. WWW '15 Companion. New York, NY, USA: ACM, 2015, pp. 239–242.

[14] J. Scheuner, P. Leitner, J. Cito, and H. Gall, "Cloud work bench – infrastructure-as-code based cloud benchmarking," in *2014 IEEE 6th International Conference on Cloud Computing Technology and Science*, Dec 2014, pp. 246–253.

[15] P. Leitner and J. Cito, "Patterns in the Chaos&Mdash;A Study of Performance Variation and Predictability in Public IaaS Clouds," *ACM Trans. Internet Technol.*, vol. 16, no. 3, pp. 15:1–15:23, Apr. 2016.

[16] J. Scheuner and P. Leitner, "A Cloud Benchmark Suite Combining Micro and Applications Benchmarks," in *Companion of the 2018 ACM/SPEC International Conference on Performance Engineering*, ser. ICPE '18. New York, NY, USA: ACM, 2018, pp. 161–166.

[17] E. Folkerts, A. Alexandrov, K. Sachs, A. Iosup, V. Markl, and C. Tosun, "Benchmarking in the Cloud: What It Should, Can, and Cannot Be," in *Selected Topics in Performance Evaluation and Benchmarking*, R. Nambiar and M. Poess, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 173–188.

[18] A. Palesandro, M. Lacoste, N. Bennani, C. Ghedira-Guegan, and D. Bourge, "Mantus: Putting Aspects to Work for Flexible Multi-Cloud Deployment," in *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*, June 2017, pp. 656–663.

[19] A. Bhattacharjee, Y. Barve, A. Gokhale, and T. Kuroda, "Technical Report - CloudCAMP: A Model-driven Generative Approach for Automating Cloud Application Deployment and Management," 2018.

[20] H. Huijgens, R. van Solingen, and A. van Deursen, "How to Build a Good Practice Software Project Portfolio?" in *Companion Proceedings of the 36th International Conference on Software Engineering*, ser. ICSE Companion 2014. New York, NY, USA: ACM, 2014, pp. 64–73.

[21] H. Huijgens, E. Greuter, J. Brons, E. A. van Doorn, I. Papadopoulos, F. M. Martinez, M. Aniche, O. Visser, and A. van Deursen, "TUD-SERG-2018-003 - Factors Affecting Cloud Infra-Services Development Lead Times: A Case Study at ING," 2018. [Online]. Available: https://se.ewi.tudelft.nl/tr.html

[22] IFPUG, "IFPUG FSM Method: ISO/IEC 20926 - Software and systems engineering - Software measurement - IFPUG functional size measurement method," 2009.

[23] S. Fincher and J. Teneberg, "Making sense of card sorting data," *Expert Systems*, vol. 22, no. 3, pp. 89–93, 2005.

[24] H. Huijgens, "Evidence-Based Software Portfolio Management (EBSPM) Research Repository," 2017. [Online]. Available: https://doi.org/10.4121/uuid:42fd1be1-325f-47a4-ba39-31af35ca7f75

[25] T. Dybå, V. By Kampenes, and D. I. K. Sjøberg, "A systematic review of statistical power in software engineering experiments," *Information and Software Technology*, vol. 48, pp. 745–755, 2006.

[26] "Controlling the false discovery rate: A practical and powerful approach to multiple testing." *Journal of the Royal Statistical Society Series B*, vol. 57, no. 1, pp. 289–300, 1995.