Joona Tasanen

PRO GRADU – METHODS OF QUALITY ASSURANCE FOR PRODUCTIONALIZATION OF CLOUD SYSTEMS

JYVÄSKYLÄN YLIOPISTO
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA
2022

TIIVISTELMÄ

Tasanen, Joona
Laadunvarmistamisen menetelmät pilvijärjestelmien tuotannollistamisessa
Jyväskylä: Jyväskylän yliopisto, 2022, 57 s.
Tietojärjestelmätiede, pro gradu -tutkielma
Ohjaaja(t): Clements, Kati

Tiivistelmä

Tässä tutkimuksessa tutkittiin laadunvarmistamisen keinoja ja instrumentteja pilvijärjestelmien tuotannollistamista varten. Tutkielma koostuu kirjallisuus katsauksesta ja tapaustutkimuksesta. Tapaustutkimus oli kohdeyrityksen tilaama, laadunvarmistamisen keinojen ja instrumenttien löytämiseksi pilvijärjestelmien tuotannollistamiseen. Motiivina oli uusien teknologioiden tuomat haasteet laadukkaiden pilvipalveluiden tuottamiseen.

Kirjallisuuskatsauksen ja tapaustutkimuksen avulla löydettiin 27 metodia ja instrumenttia laadunvarmistamiseen. Näiden joukosta löytyi useita hyödyllisiä metodeja, kuten automaattinen ongelmien tulkinta ja suorituskykyjen mittaaminen. Nämä metodit tuovat käytännön hyötyjä pilvijärjestelmien laadunvarmistamiseen kohde yrityksessä. Jatkotutkimus aiheena tuloksia voidaan koestaa tai validoida ja uusia tapaustutkimuksia tehdä myös muissa yrityksissä.

Asiasanat: Quality Assurance, Cloud Computing, IS Success Model, Case Study, Productionalization, Telecommunication

ABSTRACT

Tasanen, Joona
Methods of quality assurance for productionalization of cloud systems
Jyväskylä: University of Jyväskylä, 2022, 57 pp.
Information Systems, Master's Thesis
Supervisor(s): Clements, Kati

In this study beneficial methods and instruments of quality assurance for productionalization of cloud systems were researched from the literature and from the case study done in the target organization. This case study was ordered by the target organization to help with quality assurance of cloud systems during productionalization. As implementing emerging technologies to new cloud systems provide increasing challenges to maintain high quality cloud services.

From the results of the literary review and the case study 27 methods and instruments of quality assurance were found. There were several beneficial methods, including problem determination with automation and benchmarking. These methods will provide practical help for quality assuring cloud systems in the target organization. To extend the research in the field these results could be tested, and more case studies done in different organizations.

Keywords: Quality Assurance, Cloud Computing, IS Success Model, Case Study, Productionalization, Telecommunication

FIGURES

TABLES

TABLE OF CONTENTS

1 INTRODUCTION

Cloud systems of emerging technologies including container clouds are getting massively needed with many emerging IT services like Internet of Things (IoT) applications, Network Function Virtualization (NFV) services, Mobile Edge Computing (MEC), fifth generation (5G) applications and services. These systems are in the forefront of running emerging services and applications and they have high requirements of latency and quality (Aral et al., 2019; Huedo et al., 2021). Furthermore, benefits of emerging technologies like containers go beyond enabling higher level resource utilization, containerization transform the data center from machine-oriented to application-oriented (Burns et al., 2016).

In this study quality assurance is researched in the context of adopting emerging technologies in cloud systems. This paper attempts to find methods or instruments which create net benefit for information system (IS) success in the target organizations cloud platforms. This net benefit of IS success is improved by change of quality measurements from three quality dimensions as defined by (DeLone & McLean, 2003). These quality measurements are defined later in this study.

Motivation for this research is that target organization is periodically adopting emerging technologies for their live production cloud environments. These environments are running their critical applications and the quality of these cloud systems is crucial for the successful use and deployment of the applications. Risk identified in the target organization is that during deployment of a new cloud system with emerging technologies; how is the quality of the system in use preserved or improved during adoption of these emerging technologies. In this paper we attempt to find methods to reduce this risk and improve quality assurance in productionalization of cloud systems.

This study is two-fold first there is literary review of quality measurements and methods used in productionalization of cloud systems related to these measurements. Secondly there is a case study which attempts to validate methods found from the literature and research for new methods and instruments. Research question in this study is as follows:

What methods of quality assurance are beneficial for productionalization of cloud systems?

Productionalization is defined as the deployment of a process or a procedure into a production environment (*Macmillan English Dictionary*, n.d.) or (business) the adoption of an approach or technology in a live production environment (*Wiktionary the free dictionary*, n.d.). It is seldom used terminology in scientific articles from search queries in the Google's Scholar system there was less than 100 results. Despite this it is common terminology in the target organization and used by their software vendors. Thus, in this research the term productionalization will be used to refer the process of deploying or adapting new cloud architecture and bringing it to production level i.e., serving live traffic.

Quality assurance is "the maintenance of a desired level of quality in a service or product, especially by means of attention to every stage of the process of delivery or production (Oxford Languages)". In this study we will focus on the productionalization part of the quality assurance i.e., deployment of new cloud platforms with emerging technologies.

This study aims to provide help for target organization with their quality assurance during productionalization of emerging cloud architectures. Most of the research done about quality assurance in information systems is not that recent. Furthermore, if connected to private cloud context this research is usually more focused in the migration from legacy systems to cloud and especially to public clouds. Previous research is also often done from perspective of adopting emerging technologies especially cloud-based services and not about quality assurance.

This study aims to extend the research about quality assurance to cloud systems and especially in the context of target organization, which is running applications on top of cloud systems and planning to migrate these workloads from more traditional cloud to containerized clouds. Aim of the research is to find methods for quality assurance by doing literary review from the context of quality assurance for productionalization of cloud systems and use this literature review as a method for conducting case-study in the target organization.

In this study earlier literature about quality and IS success were referred by different quality dimensions. These dimensions were defined in the context of target organization. Second part of the literature review, methods and instruments of quality assurance were listed in the context of productionalization of cloud systems.

This paper organizes as follows. Chapter 2 describes quality measurements. Chapter 3 describes methods and instruments used in productionalization of cloud systems. Defining what productionalization is and what methods are used for quality assurance during productionalization. Chapter 4 is the synthesis of quality measurements with the methods of productionalization. Chapter 5 describes the methods used in the literary review and the case study in the target organization. In chapter 6 results of the study are reported and in chapter 7 discussion about the results, limitations, and future research with the context of target organization is discussed.

2 QUALITY ASSURANCE

Purpose of this research is to find beneficial methods and instruments of quality assurance for productionalization of cloud systems in the target organization. Quality assurance is" the maintenance of a desired level of quality in a service or product, especially by means of attention to every stage of the process of delivery or production. (Oxford Languages)". Furthermore, operational definition in the context of the target organization, desired quality of emerging cloud platform is the capability of running new and existing applications with increasing requirements for high performance and latency criticality. Similar requirements as in the target organization were reported by (Huedo et al., 2021).

2.1 IS success theory

Information system (IS) success theory first originated by (DeLone & McLean, 1992) which is the basis for this research. IS success theory is well known and widely referred in information systems. In Google Scholar only original article by DeLone and McLean (1992) has been cited 15 882 times (checked 24th of April 2022). Research they did tried to define dependent variable i.e., IS success. Their research and modeling was done as an answer to one of the five issues identified by Peter Keen: What is the dependent variable? (Keen, 1980).

(DeLone & McLean, 1992) examined that there is not one measure for success but many. They concluded these measures to six major categories SYSTEM QUALITY, INFORMATION QUALITY, USE, USER SATISFACTION, INDIVIDUAL IMPACT and ORGANIZATIONAL IMPACT. Adapting IS success model by adding SERVICE QUALITY was done by at least Pitt et al. (1995). After revisiting the model 10 years later DeLone and McLean added SERVICE QUALITY to there IS success model and INDIVIDUAL AND ORGANIZATIONAL IMPACT were joined together as net benefits (DeLone & McLean, 2003, 2002). This upgraded model is the model which we use in this research. Reprinted model from DeLone and McLean (2003) in figure 1.

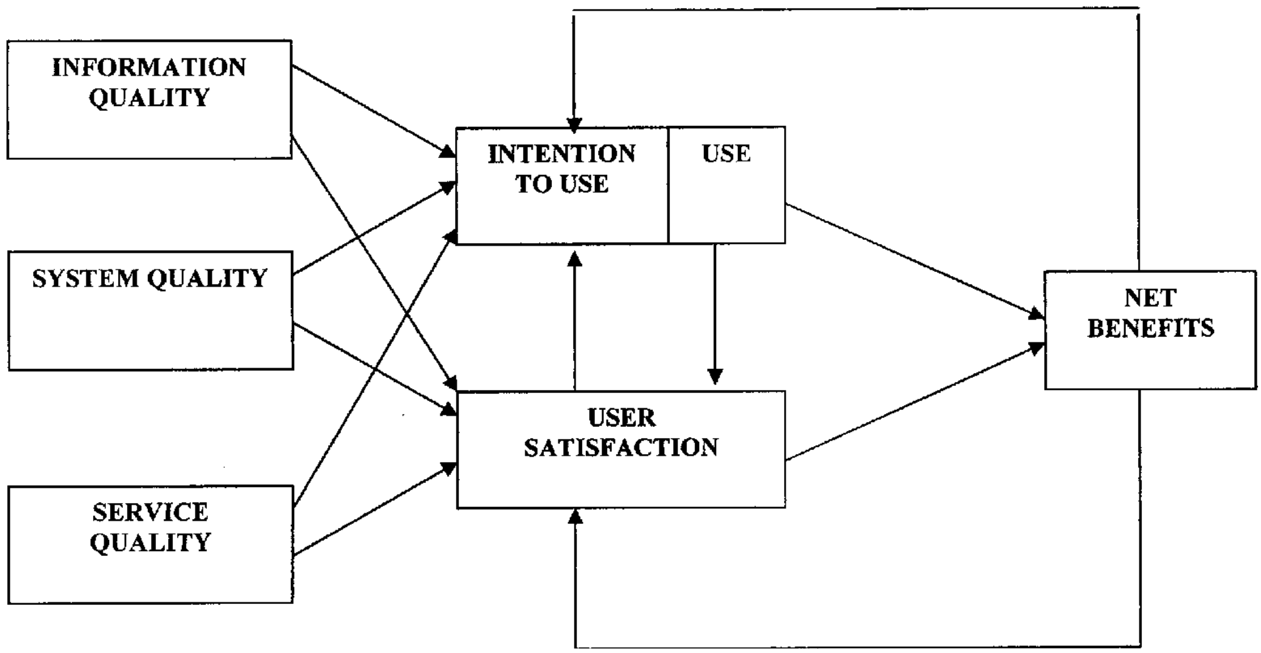

Figure 1 Updated D&M IS Success Model (DeLone & McLean, 2003)

Note: Reprinted from The DeLone and McLean model of information systems success: A ten-year update, by DeLone, W. H., & McLean, E. R., 2003, Journal of Management Information Systems: JMIS, 19(4), 9–30.

In information systems quality has three major dimensions: INFORMATION QUALITY, SYSTEM QUALITY and SERVICE QUALITY as described by (DeLone & McLean, 2002). In figure 1 these dimensions are representing independent variables on the left. These independent variables will affect users' intention to use, use and user satisfaction producing net benefits for the information system in this research this system would be cloud computing platform. IS success model was also adapted specifically for cloud computing purposes by (Lian, 2017), but in this paper the original IS success model will be used without adaptation. In their research they integrated IS success model with trust. So instead of use or intention of use with user satisfaction their dependent variables were trust and cloud satisfaction.

In another cloud related research Liu and Wang (2021) surveyed 165 automotive companies with their adapted model, where net benefits was replaced with performance of cloud-based marketing systems and found information quality, system quality and service quality to have positive effect on their dependent variable.

2.2 Quality measurements

Assuring quality of the three independent variables, net benefits of the emerging cloud platforms can be realized as is explained in IS success model (DeLone & McLean, 2003). To measure these variables Table 1 shows common measurements used in the literature.

Table 1 Quality measurements for the three independent variables

Variable	Measurement	Source
Information quality	Accuracy, precision, reliability, consistency of information provided	(DeLone & McLean, 2003; Pitt et al., 1995)
System quality	Reliability, data quality, functionality	(DeLone & McLean, 2003)
Service quality	Responsiveness, assurance, and reliability	(Pitt et al., 1995)

3 PRODUCTIONALIZATION OF CLOUD SYSTEMS

In chapter 3 first productionalization and cloud systems are defined. After which methods and instruments of productionalization are searched. These methods and instruments in this literary review are mainly from the context of a private cloud systems. Considering that target organization is currently running more traditional network functions virtualization clouds and will be migrating their production workloads to emerging containerized clouds, both of these technology contexts will be used in this study.

3.1 Definition of cloud systems and productionalization

In this study operational definition of cloud systems is cloud computing platform which consists of hardware and software components. Cloud systems transforms IT infrastructures into a utility (Vennam, n.d.). In this study cloud systems are mainly private clouds, private cloud defined in the citation below.

> A **Private Cloud** is a model of cloud computing where the infrastructure is dedicated to a single user organization. A private cloud can be hosted either at an organization's own data center, at a third party colocation facility, or via a private cloud provider who offers private cloud hosting services and may or may not also offer traditional public shared multi-tenant cloud infrastructure. (*What is a private cloud?*, 2022)

Productionalization is defined as the deployment of a process or a procedure into a production environment (*Macmillan English Dictionary*, n.d.) or (business) the adoption of an approach or technology in a live production environment (*Wiktionary the free dictionary*, n.d.).

3.2 Methods and instruments of quality assurance in productionalization

Five categories below were arbitrarily chosen by the context of the methods in the papers which were read in this literary review. Categories were discussed with the target organization's technological lead and team manager of the team responsible for deployment and maintenance of the cloud platforms and were deemed to be fitting for the purposes of the study.

3.2.1 Benchmarking

Main metrics used by Wang et al. (2018) in their research of comparing serverless computing providers AWS lambda, Azure Functions and Google Cloud Functions were cold start and warm start latency. These refer to process of launching a new function instance and re-using existing function instance respectively. Other metrics Wang et al. (2018) used were related to performance isolation investigation. These were CPU utilization (Python time.time() or NodeJS Date.now() functions), I/O throughput (dd command) and network throughput (iperf 3.13).

Das et al. (2018) presented benchmarking for edge computing platforms called EdgeBench. EdgeBench features three key applications: a speech/audio-to-text decoder, an image recognition machine learning model, and a scalar value generator emulating a sensor as listed by (Das et al., 2018). These applications were run on edge devices sent data to cloud. Metrics used for benchmarking with these applications were compute time, time-in-flight, end-to-end latency, payload size, CPU, and memory utilization.

3.2.2 Deployment

For resource deployment and management of microservice containers Lin et al. (2019) proposed use of ant colony optimization (ACO) algorithm. In their test results (executed in Alibaba Cluster Trace V2018) ACO algorithm outperformed more limited resource allocation algorithms due to three-dimensional optimization of the ACO. Optimization objectives used in their research were network transmission overhead, resource load of the cluster and reliability of the cluster services. Use of ACO relies on multiple iteration to achieve the best result due to its learning nature as concluded by Lin et al. (2019) quality of the solution in next generation can be improved as much as possible through the mechanisms of evaluation and feedback.

3.2.3 Application requirements

Application requirements such as fault-tolerance requirements, time constraints, container priorities and QoS classes were described in the taxonomy for container-based cluster orchestrations systems (Rodriguez & Buyya, 2019). These requirements are basis for application to run in a large-scale cloud system.

Furthermore, measuring the performance of these applications is important for quality assurance. Use of ISO 25010 to measure performance of cloud computing applications was extensively investigated in the context of cloud systems (Ravanello et al., 2014).

3.2.4 Resource utilization

Kuo et al. (2021) proposed architecture on how to maximize utilization of resources and how to improve consumption of resources, how to improve quality in production . In their case cloud testing platform.

Zhong and Buyya (2020) proposed heterogenous task assignment strategy (HTAS) for cloud workloads. They coupled four different application models with different priorities and node groups for task scheduling and scaling decisions to prevent resource wasting and overprovisioning while minimizing QoS degradation.

Rovnyagin et al. (2021) proposed use of re-scheduler which allows orchestrator to change the existing distribution scheme to match the current state of the cluster. Their paper included proposal of a new algorithm for re-schedulers that use reinforcement machine learning. This would allow re-scheduler to work in different situations and states of the container cluster.

3.2.5 Monitoring

In their research about delivering software with agility and quality in cloud environments Oliveira et al. (2016) proposed continuous quality assurance approach. This was mainly in relation to agile software delivery but included problem determination services are in the scope of this research. Environments where they did their demonstrations were Container Cloud and Openstack cloud environments. Demonstrations used data crawler to gather data about the current state of the cluster. The crawler parses log events and annotates each event with semantic information extracted from it. Furthermore, crawler also takes snapshots of the cloud state. Mainly the crawler is used for problem determination and dynamic Kibana pages can be produced from the data gathered by the crawler for specific incidents including semantically reasonable information tied to snapshotted at the time of the incident state of the cloud. Earlier research about a tool for automated problem determination used two-level approach of proactive high-level system health checks with rule-based probing (Huang et al., 2007).

Dynamic monitoring frequency or changing of monitoring frequency to match volatility of data. In objects like performance or resource usage this algorithm would increase frequency when object is changing more often and decreasing when it stays around the same value. (Mastelic & Brandic, 2015)

3.2.6 Security

Razaque and Rizvi (2017) talked about use of third party auditing (TPA) as a tool assist in guaranteeing data privacy. TPA generally means audition of cloud systems by some audit organization outside of the target organization.

ISO 27001 is international standard for information security management. In cloud systems it provides best practices and specifies requirements for implementing, maintaining, and improving information security management system (ISMS) in target organization (*ISO 27001*, 2019). Providing risk-based view for information security ISO 27001 can be certified and audited with ISO. ISO 27001 is widely used and its references in the literature are abundant in the context of cloud systems. Notably ISO 27001 is used for risk-based approach in cloud systems information security. It has been used for risk analysis (Alebrahim et al., 2014) and risk assessment (Weil, 2019) in cloud systems.

4 QUALITY ASSURANCE OF PRODUCTIONALIZATION OF CLOUD SYSTEM

Quality assurance in the context of productionalization of cloud systems is explained here. This is a conclusion of literary reviews done in chapter 2 and 3. This conclusion attempts to find answer to what methods of quality assurance are beneficial during productionalization of cloud systems.

In Table 2 methods and instruments which were found from the literature for productionalization of cloud systems were listed with their corresponding independent variable of quality measurement. Methods and instruments were matched to corresponding quality measurements namely information quality, system quality and service quality according to literature sources as shown in Table 1.

Table 2 Quality measurement synthesis

Instrument/Method	**Category**	**Description**	**Quality measurement**	**Articles referenced in**
Fault-tolerance requirements	Application Requirements	Fulfilling quality of service requirements of fault tolerance in instances	Service quality	(Rodriguez & Buyya, 2019)
QoS classes	Application Requirements	Enforcing quality of service classes in instances	Service quality	(Rodriguez & Buyya, 2019)
Container priorities	Application Requirements	Enforcing use of container priorities. This refers to instances having different priorities for accessing resources.	Service quality	(Rodriguez & Buyya, 2019)
ISO 25010	Application Requirements	Use of ISO 25010 to measure performance of cloud computing applications	Service quality	(Ravanello et al., 2014)

Autoscaling	Resource utilization	Using automatic scaling for dynamically changing workload through instance acquisition.	Service quality	(Kuo et al., 2021; Zhong & Buyya, 2020)
Use of re-scheduler	Resource utilization	Automatically or periodically re-scheduling of instances to better match the new status of the cluster.	System quality	(Rovnyagin et al., 2021)
ISO 27001	Security	Being compliant of ISO 27001 and actively auditing for it.	Information quality	(Alebrahim et al., 2014; Weil, 2019)
Third party auditor (TPA)	Security	Third party auditor in relation to security of the cloud systems. Third party will audit the system and its security requirements.	Information quality	(Razaque & Rizvi, 2017)
Problem determination by using data crawler	Monitoring	Automatic problem determination from logs and creation of dynamic Kibana page for given problem event.	Information quality	(Huang et al., 2007; Oliveira et al., 2016)
Dynamic monitoring frequency.	Monitoring	Changing of monitoring frequency to match volatility of data. In objects like performance or resource usage this algorithm would increase frequency when object is changing more often and decreasing when it stays around the same value.	Information quality	(Mastelic & Brandic, 2015)
Use of ant colony optimization (ACO) algorithm	Deployment	Use of ant colony optimization for instance deployment. Ant colony optimization is type of algorithm which attempts to mimic ants real life behavior to find optimal deployment path for virtual machines.	System quality	(Lin et al., 2019)
Cold start and warm start latency	Benchmarking	Benchmarking the launch time of instances. Cold start refers doing it for first time and warm start repeating same launch again.	System quality	(Wang et al., 2018)
CPU utilization	Benchmarking	Benchmarking CPU utilization metric in instances for performance validation.	System quality	(Das et al., 2018; Wang et al., 2018)

Memory utilization	Benchmarking	Benchmarking memory utilization metric in instances for performance validation.	System quality	(Das et al., 2018)
End-to-end latency	Benchmarking	Benchmarking end-to-end latency metric in instances for performance validation.	System quality	(Das et al., 2018)
Time-in-flight	Benchmarking	Benchmarking time-in-flight metric in instances for performance validation.	System quality	(Das et al., 2018)
I/O throughput	Benchmarking	Benchmarking I/O throughput metric in instances for performance validation.	System quality	(Wang et al., 2018)
Network throughput	Benchmarking	Benchmarking network throughput metric in instances for performance validation.	System quality	(Wang et al., 2018)

5 RESEARCH METHOD

In this chapter the target organization where the empirical part of this research was executed, and research methodologies used in this study are described.

5.1 Target organization

Target organization is large enterprise operating in telecommunication field. As for any telecom carrier minimal downtime and high level of sustained service operation is crucial for their business. In this master's thesis we are looking for to improve quality assurance during productionalization of on-premises clouds systems which are serving telecom applications. These cloud systems are regularly upgraded to adopt emerging technologies to maintain the competitive edge in telecom industry. During these upgrade phases underlying software and hardware architecture may be changed drastically. This brings a challenge where previously stable live workloads need to be migrated to new cloud system or to upgraded cloud system. Quality assurance of productionalization becomes crucial for achieving stable live production cloud system.

5.2 Research design

Literary review was mainly executed by using Scopus database. Keywords used in the search queries were cloud system, cloud computing, private cloud, IS success, quality, quality assurance, performance, Kubernetes, Openstack, virtualization, containers, and on-premises cloud. Search results were limited to information systems and computer science fields. Mainly results which were from respectable journals of at least score 1 according to Julkaisufoorumi database were chosen.

Empirical part was executed in the target organization. It consisted of qualitative case study on quality assurance methods and instruments for cloud systems of the target organization. In the literary review of this paper, from chapters two to four methods or instruments for quality assurance of cloud systems were found from the literature. With qualitative empirical research we attempt to find out answer to the research question: *what methods of quality assurance are beneficial for productionalization of cloud systems?* Even though Yin (2009, p. 4) describes case studies being more fitting for *how* or *why* research questions, he also describes case study method to be relevant if research question requires extensive and in depth description. This kind of description is exactly what is required to answer this research question in this study. These methods and instruments need to be explained in depth by the subjects to gain knowledge with which research question may be answered.

Research is done by conducting semi-structured interviews in the target organization. Firstly, results of the literary review from chapter 4 are validated in the target organization by conducting interviews to find out which of these found methods or instruments could be beneficial for cloud systems. Secondly, interviews are used to collect data about what other methods and instruments experts in the target organization believe that are beneficial for quality assurance in productionalization of cloud systems. Thirdly, interviews are used to find out what challenges or benefits there are in quality assurance in productionalization of cloud system, these are auxiliary questions to help interviewee bring up possible methods or instruments.

Interviews are commonly used in qualitative research and in case study research it is common to rely mostly on interviews for data collection (Myers, 2019, p. 29). So will be done in this study also. Semi-structural interview method was chosen because it gives some freedom for the subject to express and include their own thoughts while keeping the research somewhat limited, to keep the scope of the research in the research problem.

It is important to have sufficient access to data and interviewee subjects in the target organization (Yin, 2009, p. 27). For the purposes of this study there was relative abundance of possible interviewee candidates and from among these were chosen a group which was most representative of different platforms and expertise in the target organization. Thus, not every member of every team related to cloud systems was interviewed as it might have led to similar standpoints being repeated or this study being extended outside of the funded resources.

5.3 Methodologies

Semi-structured interviews were conducted on cloud systems experts in target organization. This included multiple teams responsible for several different types of clouds, including virtualization and containerization-based clouds. Although the target of the research was private clouds some of the interviewees also had public cloud expertise. Every one of the interviewees had more than a few years of expertise in the field.

Interviews were conducted in Finnish and transcript to English. Translation was done by best effort as word-by-word translation of some Finnish concepts was not feasible and optionally some concepts were confirmed from the participants during the interview. Recordings of the interviews were destroyed after the transcripts were done.

Preliminary questionnaire was given to the interviewees usually one day ahead of the interview shown in Appendix 2. This questionnaire included questions about role and expertise and allowed interviewees to mark a check box if they had experience about certain method or instrument or if they deemed them important. This was done to save some time during the interview and to provide basis for the questions one and two from where it was easier for subjects to begin explaining their views. Interview questions are shown in Appendix 1.

Interviews were held in Microsoft Teams application which was maintained by the target organization. As such interview recordings never left the systems or employee computers which were in control of the target organization. Duration of the interviews shown in Table 3 under results chapter.

6 RESULTS

In this chapter the research results of this case study are presented. Results are from interviews held in the target organization including preliminary questionnaire given to the participants. All interview participants are members of the target organization working in roles related to operations, development, or management in cloud systems in the target organization.

Results of preliminary questionnaire (Appendix 2) are listed in the Table 3. These results include the breakout of roles and experience in cloud systems among interviewees. Especially mentionable factor among the interviewees is that they represent multiple different teams in the target organization who are responsible of operating, developing, or managing more than twenty different clusters of the cloud systems. These cloud systems include different methods and technologies such as containerization and virtualization. All the interviewees have had at least mid-level experience regarding cloud systems. Beginner or junior level employees were not involved in this study.

In Appendix 1 among the interview questions there were definitions related to key terminologies of the research, which are cloud systems, quality assurance and productionalization. As a summary, cloud systems mean private clouds which utilize either virtualization or containerization technologies; quality assurance means maintaining desired level of quality in cloud system; productionalization means deployment of cloud systems as production system which services production workloads.

Results are divided to three sub-chapters according to quality measurements of independent variables, namely, information quality, system quality and service quality as defined by DeLone and Mclean's updated IS success model from (DeLone & McLean, 2003, 2002). Quality measurements were shown in the Table 1. These sub-chapters are further divided into sub-chapters of methods and instruments, and each method or instrument has its own sub-chapter.

Not all transcript interview material was used in the results. Quite a lot of omitting was done for stories or descriptions which were less relevant to the topic. Most informative key citations are included under each method or instrument sub-chapter. Repeating shorter remarks were also omitted if they did not bring new insight to the topic or at least strong confirmative comments.

One of the interviewees wanted to stress that their expertise was from the infrastructure point of view. This point of view is fitting to the scope of the research as it is still relevant for productionalization of cloud systems and their results are included in the results.

> When we talk about cloud systems, my experience is more from infrastructure side rather than services. For long time. I haven't been that much involved in individual services although I do have knowledge from those too. My perspective is more from the infrastructure side. In things which are more focused on cloud infrastructure and affect widely to services which are running on it. (S1, Management, Mid-level)

Table 3 Participants role and experience in cloud systems

	Role in cloud systems	Experience in cloud systems	Duration
S1	Management	Mid-level	22 min
S2	Management	Mid-level	34 min
S3	Development	Senior-level	26 min
S4	Operations	Mid-level	40 min
S5	Operations	Senior-level	37 min
S6	Development	Senior-level	27 min
S7	Management	Senior-level	29 min

6.1 Information quality

6.1.1 ISO 27001

As seen from the abundance of different articles about ISO 27001 in cloud systems, it wasn't surprising that ISO 27001 was in use in some of the cloud systems of the target organization. Although there were some cloud systems where ISO 27001 wasn't a practice. Some of the interviewees described that ISO 27001 is requirement for them from internal and external customers. So there seems to be different requirements from the cloud system's tenant point of view in different cloud systems.

Nevertheless, consensus among interviewees who had experience of using or certifying for ISO 27001 was very positive. Benefits which were described included systematic checks, helping with void of competency and being able to use same set of specifications for multiple platforms.

> ISO 27001 process in cloud systems will bring benefits with its systematic checks, but it is not obligatory, and it can fill in the void of competency. When that kind of model is used it enables wider look into the systems from multiple angles from the system management point of view. Of course, it will not take a stand for technical implementation. (S1, Management, Mid-level)

> ISO 27001 certification is now-a-days requirement from internal customers and maybe also from external customers. We can prove that we are doing [security] in certified way. We also have criteria which we need to match from the law point of view and this iso certification is big part of ascertaining that we fulfill these requirements from law. Benefits are also that different kind of platforms need to fit to ISO 27001. So, we can use the same specifications for all platforms. We don't need to think what we need to do for certain platform to match security requirements, because we have generic list of base requirements which we can use. (S5, Operations, Senior-level)

> ISO 27001 is foundation for quality assurance. It is like minimal requirement and is important basis for continues day to day work. (S7, Management, Senior-level)

6.1.2 Third party auditing

TPA was often mentioned in the context of ISO 27001 auditing. This auditing for ISO 27001 was done by a third party for target organization in some cloud systems and as such is of course use of TPA. But even in those cloud systems where ISO 27001 wasn't in use nor certified for, TPA was described as essential. Common theme among the answers was that TPA is beneficial because it gives visibility for developers and operators to their mistakes which they might otherwise be blind to.

Building of trust towards partners and customers was described as one of the benefits in TPA. This was connected to an ability to be able to prove that some other entity has validated the cloud system. Especially for security related audits TPA this ability was described as must have. Furthermore, TPA was taught to be better option because its use did not consume the in-house resources of the target organization and could be bought from outside.

> As a comment for TPA: scans or similar mechanism can be used which investigate the system from black box perspective. This perspective where someone is checking the system from outside is the main idea of TPA. Also, ISO auditing is similar in its perspective. Third party will come and ask questions about the system, but the one responsible for the system must solve them. (S1, Management, Mid-level)

> TPA is in use. Open-source projects or their upstream which we use are not audited by TPA, so we usually try to do this together with several architects and developers. Overall, the projects we use must be maintained and have strong community. In the code which we produce in-house especially for authentication and security related projects we have TPA. Benefits of TPA are that the developer who has been developing same system can have hard time to think outside the box and can be blind to issues in their own system. Furthermore, doing auditing in house from separate teams can be time consuming and budget straining so using TPA is usually better option. (S2, Management, Mid-level)

> TPA confirms that we have done correct things in correct manner. TPA report can be used to build trust towards the partners and customers. We can show that someone else has review the state of the cluster and can say that it fulfils certain criteria. It is essential especially in security side to show that also someone else agrees that the system is in certain [security] level. (S3, Development, Senior-level)

> TPA is good concrete example of how we can protect our cloud systems from quality deviations. When we are developing a new cloud system solution. TPA and security is good to have as separate path from the beginning. As we are responsible for our customers about [protecting them from] cyber-attacks etc. Even though we would not have quality deviations in the cloud systems, we should have TPA and dedicate resources for it. (S7, Management, Senior-level)

6.1.3 Problem determination by using data crawler

Experiences in problem determination from the logs including use of data crawler seemed to have some variance among the interviewees. Some of them described to have vast experience in the subject while others only a little. Some of the interviewees were not familiar with the term data-crawler, but the concept was familiar to them. Term data crawler came from the literature and may not be as widely used.

Problem determination was described as beneficial by everyone of the interviewees. These described benefits included: Identifying capacity problems and security threats. Reduced daily workload for operators due to better troubleshooting capabilities provided by problem determination. Ability to highlight anomalies from the baseline, which may be difficult for humans.

Furthermore, connected to these methods interviewees also proposed the use of artificial intelligence operations (AIOps) or mentioned that there are plans to include artificial intelligence to log monitoring. Interestingly AIOps was raised from different teams separately, as discussed with the interviewees there is not much interaction between these teams responsible for different cloud systems.

> Logging systems or log service where logs are gathered, and events are based on that. Big system where there is a lot of events, this kind of centralized service can be useful. It enables identifying capacity problems and data security threat. Personally, I feel that this is very interesting in problem determination. This kind of logging events can be triggered from known problem cases. Systematic use of this enables the system to mature for better quality. (S1, Management, Mid-level)

> Data crawler or anomaly detector has been in use, but I don't have personal experience with that, but I see it beneficial, because human can detect certain things but seeing the baseline and detecting anomalies from that baseline can be difficult. Highlighting these anomalies is valuable. So familiar but haven't used it myself. Anomaly detection and AIOps are useful for mitigating human weakness. Machine is more capable of finding patterns from masses. (S2, Management, Mid-level)

> Data-crawler and automatic problem determination is very good idea. I have had very little experience so far but there is huge potential to reduce the operational load. When operational load is being reduced from daily works like in troubleshooting, it will reduce rush and hassle during troubleshooting. Problems can be solved more straightforward and smooth ways. (S4, Operations, Mid-level)

> Problem determination experiences have been limited but I see it as important. If we want to keep our cloud system stabile it is good to gather and make data visible. It will help our operations. From KPI standpoint, if we collect data and make some events

visualized and alarms from them it will help quality assurance. It will also make troubleshooting quicker. (S7, Management, Senior-level)

Somewhat connected to problem determination one of the interviewees described how situational awareness and predicting situations are valuable for cloud systems. Main point seemed to be how problem determination as a monitoring method should give visibility to the status of the cloud system. What problems are occurring and what is the status of the monitored components. They also stressed how it would be beneficial if this information would also be visible for the tenants of the cloud system. This would be beneficial because it will reduce the number of inquiries to the status of the cloud system and allow more focus on development.

> Secondly how state information of the cloud environment and how its actions and components are monitored. All metrics etc. which are accessible for applications, cloud platforms are also applications and need metrics. Cloud platform needs metrics in a same was as applications running on top of it. Even though those applications are monitored with metrics, also cloud platform needs to be monitored and alarms created from metrics in a same way. Cloud platform can cause surprises if it is not monitored in a same way as applications are. Cloud platforms are not anymore traditional servers or black boxes. In my opinion cloud environments are software and their requirements for metrics and monitoring are the same as for software. Related to this situational awareness what is happening to the software right now is valuable for the operating team to predict situations and can use anomaly detection to find out what is happening now. Metrics or monitoring data is gathered and used and if possible, it is shared to users also. It helps to focus to development of cloud environment, when users can see what is going on without asking. It makes it easier when you can offer visibility for users what components have possible problems going on and why. (S2, Management, Mid-level)

6.1.4 Error tracking tools

Error tracking tools are also connected to problem determination, but also enable the historical view of the errors and problem management. These tools were brough out by one of the interviewees and said to be beneficial if there is own software development.

> If there is own software development problem management is beneficial for quality. In example use of Sentry. When quality is not only the quality from the customer point of view but also development point of view. Error tracking tools like Sentry where developer doesn’t need to manually grep logs to something which has happened earlier, or which kind of environment is in question. It helps a lot with quality assurance when from Sentry you get information about user agents, and events only related to that same error occurrence and the environment where it is happening. [Sentry is error tracking software. Like New Relic]. (S2, Management, Mid-level)

6.1.5 Dynamic monitoring frequency

Dynamic monitoring frequency was commonly described as beneficial resource saving method. When there is not as much traffic going on there is no need for high monitoring frequency. Also, with low monitoring frequency there may be some errors which may not be visible, so benefit of being able to dynamically change the frequency was described to allow some problems to be detected. This was described as beneficial for troubleshooting in cloud systems.

> Metrics and monitoring are resource consuming, so if there are no real-time requirements, well-adjusted dynamic monitoring frequency can save resources. Basic but valuable. (S2, Management, Mid-level)

> Dynamic frequency monitoring in practice if there is not much going on in the system it doesn't need so many data point to show what is going on. If there is events or traffic spikes etc. then information may be needed from more frequent time interval. It makes it easier for saving resources if there are known times when the system is in reduced capacity. (S3, Development, Senior-level)

> Dynamic monitoring frequency, we have seen cases when the frequency of monitoring hasn't been enough for troubleshooting but would have helped for problem solving if it was high enough. Especially in networking related situation can have situations where monitoring frequency is not high enough. Although for data collection higher frequency may be problematic. Dynamically reacting and changing monitoring frequency to abnormal events would be beneficial. (S4, Operations, Mid-level)

> Dynamic monitoring frequency might possibly be reasonable. If it works it is super good, we could get the essential information out of the logs. Would also be good for saving resources. (S6, Development, Senior-level)

6.1.6 Dynamic documentation

One of the interviewees brought out that in their cloud system they have been doing dynamic documentation. They described benefits to include that documentation will represent the production state of the cloud systems rather than time of documentation state. This would reduce the inadequate documentation and thus improve the information quality of the cloud system according to the interviewee.

> Dynamic documentation is promising tool. In our cloud system we are moving towards using Netterrain or similar for documentation. Benefit of dynamic documentation is that it represents the production state in real time rather than representing the state when the documentation was written down. Sometimes documentation is old due to many factors, but if dynamic documentation is in use, it will prevent the situations where some documentation is forgotten and not updating. Or some process documentation is inadequate. (S4, Operations, Mid-level)

6.1.7 Security and scanning related methods

Security scanning was said to be very essential. This included scanning the components of the cloud system and possible problems in them. Finding these problems like old software versions or publicly accessible keys was said to be crucial. This kind of scans could be done against common vulnerabilities databases as described by the interviewee. Increased security and ability to make sure that the software is up to date would be beneficial for the cloud systems as described by the interviewee.

> Security scanning is very essential. On top of scanning the components and their problems we can scan from the outside and figure out possible configuration problems and other things which can negatively affect data security. And that way to find problems before they become problems. As one example if there are keys which can be accessed by public APIs, it is crucial to be able to notice them before someone else. (S3, Development, Senior-level)

> Also, DDoS protections systems if there is need for them. Utilizing them before there is forced situation where there is too much traffic coming. Also at least some of these products or services [for DDoS protection] work also as content delivery network (CDN), which would be two birds with one stone kind of thing. Allowing content to be delivered more closer to the user and load balancing of it. (S3, Development, Senior-level)

> Overall, when deploying CI/CD pipeline there should be quality assurance there. Testing the code which is on the way to the system and binding several scans against it, like to check that software components are up to date. So, if developer has chosen older software, but there is new available it should create notification about. Also, checking against common vulnerabilities (CVE) databases if chosen software libraries have vulnerabilities which are common knowledge. Creation of necessary tickets from these. And something like this. (S3, Development, Senior-level)

6.1.8 Competency and training related methods

One of the common themes among most of the interviewees was to talk about competencies or importance of trainings. Complexity of the cloud systems was said to require competencies and every day working without enough competencies in the team was said to be difficult. Including troubleshooting, development, and day to day operations.

According to the interviewees there should be enough senior level experience in the team with the junior level employees. There should be adequate number of trainings or otherwise the productionalization phase could be slowed down. Competency challenges were not only limited to the team responsible for the cloud system. There was also mentioned that vendor's competency or familiarity with the cloud system is also crucial for quality.

Furthermore, it was said to be beneficial if the team responsible for operating the cloud system had also participated in building of it or is actively developing on top of it. So, if there is unexpected problems the team could possibly be more able to solve these problems due to better familiarity with the system.

> I have noticed in the everyday work that competency is crucial. When there is transformation to cloud services, private or public type, it requires its own competencies. In cloud infrastructure or cloud applications operations side. It is recommended to have at least one architecture level expertise among juniors in the team when developing cloud services. Competency is greatest challenge now. It is broadly visible. (S1, Management, Mid-level)
>
> More the team responsible for maintaining has experience of building the better. If team responsible for operating the cloud environment is participating to the building of the cloud environment wider knowledge and experience that team has about the cloud environment and how it is running. So, when unexpected problems happen in production the team is more capable of predicting probable causes even though the problems would be new. Also connected to this is how large group has experience about the cloud environment. How sure we are about when unexpected events happened that the team could find solutions to the problem. Even if there is vendor support, it is a little bit searching for competencies from their side too, if they haven't been in touch with our environments for somewhile. It is quicker [to solve problems] if team has operation competency about the cloud environment. Potential lack of competency or experience of cloud environment is a challenge. (S2, Management, Mid-level)
>
> The one who does the work affects the quality a lot. Competencies and trainings for the future are very important. Otherwise, we cannot operate or develop cloud systems if we don't have know-how how to do things. (S5, Operations, Senior-level)
>
> People who are participating in productionalization of cloud systems. Their competencies are a challenge if there is not enough training for certain aspects it can slow down the productionalization phase. Systems which we are working on are very complex. It is a challenge to try to make this package easier. Automation will reduce the mistakes done by humans. But the challenges are in the competencies. (S5, Operations, Senior-level)

6.1.9 Centralized status dashboard

Somewhat also related to some of the comments about problem determination and monitoring earlier in this chapter there was also clear indication for need of an instrument of overall status of the cluster. This centralized dashboard would be beneficial by showing the status of all the clusters which the team is responsible for and their current alarms and maybe also historical alarms.

This dashboard should show instant picture of the situation and anomalies should be promptly visible. Including capacity management and status in the clusters. How much there are resources left i.e., CPU and Memory and their status should also be non-uniform memory access (NUMA) aware.

> There should be centralized dashboard and metrics visibility for alarms in multiple clusters. Present situation and maybe also history of previous state. More monitoring done by experience the better. (S6, Development, Senior-level)

> Capacity management should be automatized. There should all the time be visibility how much capacity there is, and it should be more granular not just CPU and Memory but also NUMA awareness. Single dashboard view to current capacity, total and how much is free. Also, capacity prediction to the future. New capacity increases may take a year now a days. (S6, Development, Senior-level)

> Clear and indicative quality metrics should be available. There should be clear visibility to problems. There should be instant picture of the situation and anomalies should be visible promptly. (S7, Management, Senior-level)

6.2 System quality

There were few comments about system quality from the interviews for benchmarking in general. Some of the instruments described in 6.2.2 to 6.2.6 were in benchmarking category. These comments were included in the results because they could be descriptive for most if not all the instruments in mentioned sub-chapters.

Benchmarks were described to be very beneficial and important, because they allow comparison of the new cloud system to the older production cloud systems. And being useful for understanding the capacity and limitations of the cloud system. Furthermore, this kind of benchmarks were said to not be available from the cloud system vendors, so having their own benchmarks was seen as beneficial.

> Benefits of benchmarks are that we can easily compare new and old production systems. Whenever we deploy new platforms, we can do comparison are we going for better or worse. From all these benchmarks, when we try to increase the quality of the cloud systems it is important to have this kind of benchmarks to where we can compare. Usually, this kind of benchmarks are not available from the vendors and as such having your own is beneficial. (S5, Operations, Senior-level)

> What is useful in cloud benchmarking. Measuring or benchmarking the metrics in empty cloud creates us a quality targe and understanding of the capacity and limitations of the cloud system. It will help us in the future to answer to what is the required quality. (S7, Management, Senior-level)

6.2.1 Use of re-scheduler

Re-scheduler had only few comments and they were mostly only descriptive of the use-case. Re-scheduler helps the maintainability of the system and can ascertain that most recent images of instances are in use, making sure that the necessary security updates are done imperceptibly.

> If we think about our chaos monkey as re-scheduler. In our own cloud environment, we can ascertain that most recent operating system images are in use and make sure that necessary security updates can be done to the cluster imperceptibly. Usually this will not include performance changes, but those can be done also. More so we look this from the perspective that cloud users must understand that actions may be done in the cloud environment and these changes should not be visible in the application. From the users' perspective operating system updates should happen automatically. (S2, Management, Mid-level)

> Re-scheduler helps the maintainability of the system status to keep it in certain level that is more of a quality assurance, taking care that the system stays in good condition. (S3, Development, Senior-level)

6.2.2 Cold start and warm start latency

For cold and warm start latencies only, few interviewees had experience in using them in benchmarking. Benchmarking for these was thought to be beneficial because it can be important knowledge how long launching some of the services will take before they are ready for action. And in auto-scaling this information may be crucial so the time how long it takes for this scaling to proceed is known. If this time is not known well and is only an educated guess it may lead to unexpected situations or failures.

> Cold and warm start latencies are related to if in example auto-scaling is in use and system reacts to increased traffic situation that how fast changes [in capacity] will be put to practice. And, if there will be updates done in the system these latencies affect how quickly these changes are in use. In practice this is good to know how long these changes takes to optimize the system to work smoothly. In other cases [if no benchmarks have been done] these are only educated guesses and it may lead to operations taking longer than expected and situations where services are taken in to action before they are ready. (S3, Development, Senior-level)

> Warm and cold start latencies should be quite close to each other. Every boot should be close together in latency if instance is cloud native. This could be quality metric also. (S6, Development, Senior-level)

Noteworthy that one of the interviewees didn't see these latencies as inherent as they said that cold start and warm start latencies are not as relevant as they can fluctuate for many reasons.

> [Cold start and warm start latency] I don't see metering of launch times inherent in any case, it can fluctuate for many reasons. (S1, Management, Mid-level)

6.2.3 CPU and Memory utilization

CPU and Memory benchmarks were grouped together by most of the interviewees, and it makes sense as both are usually going hand in hand as black box metrics. These were described as basic but essential for cloud systems. CPU and Memory benchmarks are beneficial because they take care of the capacity management. As in most of the cloud systems CPU and Memory are the most important resources for capacity. Also constantly monitoring for CPU and Memory metrics enables the use of auto-scaling.

Furthermore, benchmarking for CPU and Memory metrics can be used for stress testing. Stress testing can be used for finding weak points or bottlenecks. Daily monitoring of these benchmarks may be beneficial, and history of these benchmarks can be used to determine trends in capacity usage. As if there would be clear trend of increase of resource consumption, it could be noticed early enough, and resources increased, or risk mitigated.

> Then these metering things. CPU and Memory are crucial from the cloud infrastructure point of view. These are benchmarks are taking care of capacity management. Then workloads will not be starved out by lack of resources. Furthermore, if there is overlap in CPU usage or memory runs out etc., these issues can be found out. Also, when we are thinking from the point of view of one virtual machine monitoring CPU and Memory utilization is important, but of course it affects only that instance. (S1, Management, Mid-level)

> CPU and Memory metrics are basic black box metrics but are valuable and if they are used together with some white box metrics can bring benefits. Neither black or white box metrics alone are not the main point, but all together more metrics help to find problems more easily. CPU and Memory metrics are easy to measure and help to find where to look for problems. (S2, Management, Mid-level)

> CPU and Memory utilization monitoring is essential, it enables the use of auto-scaling or other things. And if something special happens like high CPU usage or resource straining then something can be done accordingly. Also, while stress testing [CPU and Memory monitoring] can be used for finding weak points or bottlenecks. (S3, Development, Senior-level)

> Stress testing has been done; those are of course related to benchmarking. Daily monitoring of the metrics should be also done. Accounting of how much CPU and Memory usage is at certain points in time and creation of statistics from these. And we can see how much of resources are in use at which point in time. And we can determine if there will be future need for increasing resources at some point of time. In example if we see that in three-month period there has been increase in CPU usage in the database. Then we know if there is trend for resource increase in resource utilization, we can determine that there is pressure to increase resources for the database. And this increase can be done ahead of time without any affects to the operation of the service. And same for the stress test that we can ascertain that the system is running as expected with certain amount of traffic. So, we can ascertain ahead of time if there is expected large increase in the traffic for some time, we can test the system before that event. And we can fix the problems ahead of time. There is also a point that stress tests are as good as one can make it. If you cannot make it to match the production traffic, then it might give false image of how well system is working even though in reality it might not be adequate for production traffic. (S3, Development, Senior-level)

6.2.4 End-to-end latency

End-to-end latency was seen as especially beneficial from the end user or customer perspective. As it was described as one of the methods to ascertain that the application is working in expected way.

> End to end latency is of course very important, it is related to how cloud system is working from the end user perspective. So, is it working smooth enough? Do queries take too long time etc.? (S3, Development, Senior-level)

> End to end latency is no brainer, of course latency needs to be measured starting from the customer. It is not enough that our API has good latency, it doesn't help external actions. It is one of the metrics which need to be measured to find the bottle necks in the system. (S2, Management, Mid-level)

> End to end latency is very good for measuring end user experience. If example in DNS services running on top of the cloud, during performance validation end to end latency is most important metric to be measured. We can see which factors affect the traffic and make sure that the customer gets the best possible experience. (S4, Operations, Mid-level)

6.2.5 Time in flight

Interviewees did not have much experience with time-in-flight benchmarks. One of the interviewees said it relates to optimization of end-to-end latency and enables determining how long certain queries take.

> Time-in-flight relates to optimization of [end to end latency], enables determining in which part of the queries takes time. (S3, Development, Senior-level)

> We should have more benchmarking as a part of automation and daily [tasks]. I have little experience in containers, so time in flight or cold and warm start latencies I don't know how critical they are as high availability should have taken the action [of recovery in possible failure case]. (S7, Management, Senior-level)

6.2.6 I/O and Network throughput

I/O and Network throughput were often grouped together by interviewees and as they are both measuring the throughput of a metric it is understandable. Mostly benefits were seen from the troubleshooting or problem finding perspective. Furthermore, helping with locating bottlenecks and preventing slowdowns of the cloud system.

> I/O throughput and Network throughput are important especially in telecommunication applications. It is crucial to have enough bandwidth and traffic throughput. When these are monitored and benchmarked, we can locate events from these traffic profiles, and we can conclude where there is interference in the traffic. (S1, Management, Mid-level)

> Network throughput maybe as a metric to see if the problem is related to the users or some abnormal behavior. If example CPU and Memory are fully utilized but network throughput is in the steady level it could help to find the problem from inside the system rather than outside. (S2, Management, Mid-level)

> I/O and Network throughput helps to find bottlenecks. Enables figuring out how much data is transferred in which parts of the system and that way find the weak points of the system. In example if somewhere there isn't enough capacity to move data, slowdown of the system can be prevented [when these weak points are found]. (S3, Development, Senior-level)

6.2.7 Use of ant-colony optimization

Ant-colony optimization for deployment didn't receive many comments from the interviewees. There wasn't much of experience nor knowledge about it. Although it was mentioned that due to the nature of cloud systems in the target organization there might not be much need for deployment optimization. As in the target organization the deployment path of the instances is so well know there is diminishing returns in trying to optimize it with algorithms.

> I am not familiar with term ant colony optimization but finding optimal deployment path is usually not as necessary in our systems, because it the path of deployment is already well known. This although could be more useful in networking. (S1, Management, Mid-level)

> I have read about ant-colony optimization but don't remember now for what it is used. (S3, Development, Senior-level)

6.2.8 Risk-aware methodology

Risk-aware methodology was recognized as important and beneficial for cloud systems. Weak areas should be located and mitigated.

> Firstly risk-aware methodology is important in critical observation of cloud systems, locating weak areas and mitigation of them. (S1, Management, Mid-level)

6.2.9 Infrastructure as a code model

One of the interviewees brought up the use of infrastructure as a code (IaaC) model as beneficial for cloud systems. They stressed the benefit of how the cloud system will stay on expected configuration. Manual configuration changes which are not done by use of automation can deviate the cloud system from expected configuration. To ascertain that this doesn't happen it would be beneficial to use IaaC model.

Regardless of that the manual configuration may sometimes be quicker or might be simpler to execute. This IaaC model was seen as more beneficial because changes will stay over upgrades and other changes during the life cycle of the cloud system.

> Thirdly I believe that most important is IaaC model, all infrastructure is documented as code and then running production environment is not manually changed and we can be certain that running production environment matches the configuration in the code and that there is no deviation between configuration files and what is in the production. All manual configuration eventually leads to problems down the road. Latter one is one of the most important lessons learnt from our unit, even if manual fix to production can be quick it can lead to problems down the road due to production not anymore matching to the state of the configuration files. So, it is beneficial to change is to the configuration code. Then changes will stay correctly over upgrades and other changes done later. This is also important if there are employee changes, to make sure that everything continues working as expected. (S1, Management, Mid-level)

6.3 Service quality

6.3.1 Fault-tolerance requirements

Fault-tolerance requirements provide the cloud system with means to specify the required high availability. It enables the availability of the applications during the failures. Fault-tolerance requirements were described to be beneficial and essential by many of the interviewees.

> Availability of services and fault tolerance of the services are emphasized. Foundation of high available solution in cloud infrastructure is that it allows simultaneous use of multiple copies of the service. I think that is the most important factor that there is network and cluster architecture that allows high availability in example in situations where one cluster goes down. Infrastructure must be implemented in a way that it enables [fault-tolerance].
>
> Fault-tolerance comes from the basis that there is more than one component available for service, it can be spread to a wider area. Then one faulty thing will not stop the service like failure of switch, virtual machine, Kubernetes cluster or load balancer. So, capacity can be lost but the service stays available. (S1, Management, Mid-level)

> Fault-tolerance requirements I understood in a way that the cloud platform sets up requirements for the applications which applications must follow. If you have cloud native applications, they will be treated differently than traditional virtual machines. Resources are managed differently and there can be different requirements for the platform, in case of high availability in example. At some point, some part of the cloud system may be broken and this needs to be taken in to account for application requirements. (S2, Management, Mid-level)

> Fault-tolerance requirements increase reliability of the system. In practice it enables availability of the system during failures. It is essential part of the system. (S3, Development, Senior-level)

6.3.2 QoS classes

There was only some experience among the interviewees about the use of quality of service (QoS) classes. They were seen as situational and depending on use-case. Some of the interviewees described QoS classes as being beneficial for end-to-end network optimization and in certain workloads like real-time speech or video payloads. Another use-case where they were seen as beneficial was if there are two types of applications of which one of them processes data in patches and doesn't require that processing to be done instantly. QoS classes would provide the possibility to allow more urgent traffic to be served first and thus improving the quality of the service.

> About containers, if the goal is end-to-end network optimization the quality-of-service classes will certainly prove to be usable. Although in my role I haven't had much experience with them. (S1, Management, Mid-level)

> QoS classes have not been really required in the work what we do, and our experience is limited in example video payloads. This may be useful in more real-time applications. QoS classes are more and more important if there is real-time speech or other similar applications in use. Especially when connected to relevant metrics. Maybe also if container priorities are used dynamically according to the QoS classes they could be beneficial. (S2, Management, Mid-level)

> Quality of service relates to which kind of system is in question. I can say that if there are worker instances which are processing data in patches and there is no hurry for this data to be processed. In that kind of cases QoS classes can be used and resources given to other instances with priority and these workers can have lower priority and do processing with the leftover resources. This way capacity can be more effectively utilized and there are less hardware requirements. Same jobs can be done with less devices. Good but situational. (S3, Development, Senior-level)

6.3.3 Container priorities

One of the interviewees described container priorities as having administrative benefits and requiring less monitoring. Other interviewees saw them as good but situational or reasonable.

> Container priorities are more of an administrative benefit rather than application quality benefit. In nutshell those instances which you want to turn off last will turn of last. Benefit is also that it requires less monitoring due to having clear priorities for services. (S2, Management, Mid-level)

> Similarly, container priorities are good but situational. It will define which containers have priority for getting the resources and which will get the rest of the unused resources. (S3, Development, Senior-level)

> Container priorities could sound reasonable. As in QoS it needs to be in use in the whole network, this might be similar case with container priorities. (S6, Development, Senior-level)

Clearly in one of the teams responsible for cloud systems the idea for container priorities was different from others. They saw no need for them as in general their cloud systems should not have situations where there is competition between resources and as such there is no need for container priorities. So there was clear difference between some cloud systems being more focused on resource savings where others had more focus for performance.

> Container priorities I think that the importance is that we will not have competition for resources in our cloud systems. I don't see our cloud systems to ever have such situations. As our cloud is for telecommunication, we need assure that there is sufficient resources and capacity for whoever uses the cloud system, and we must ascertain that these resources are available. So, I don't see it as that important. (S7, Management, Senior-level)

6.3.4 Autoscaling

There was experience of autoscaling usage by two interviewees. One of them shortly described why autoscaling is beneficial that "let's automate everything which we can automate". There should be no need for humans to do scaling if it can be done automatically. This reduction in human workload was also mentioned by another interviewee.

Furthermore, they described autoscaling to be quality of life type of thing for cloud systems. Enabling accessibility of the system during increase usage of the cloud system. And being beneficial due to avoiding possibly failures due to lack of resources type of situations.

> For auto-scaling, let's automate everything which we can automate. Why we would use humans to do scaling if service profile allows to do it automatically. (S2, Management, Mid-level)

> Auto-scaling when there is more load requirement, it increases the resources. If by some reason number of users increases, auto-scaling enables accessibility of the system. In practice by using [auto-scaling] you can avoid failures and reduces the workload. These possible workloads usually also come at unpleasant time. It makes life easier. (S3, Development, Senior-level)

6.3.5 ISO 25010

ISO 25010 didn't have much of actual use experience among the interviewees especially in the context of application performance measurements in cloud systems. ISO 25010 was said to be good tool for verifying cloud systems to common standards. And there were some elements which were deemed beneficial in ISO 25010. These included end-to-end compatibility, reliability and matching the requirements given by the application developers.

> ISO 25010 has been looked in to at some point. ISO standard are good tools for verifying clouds to common standards. Verification brings more end-to-end compatibility which enables quality. ISO 25010 is good for guaranteeing the quality of

> the cloud and matches the requirements given by application developers. (S4, Operations, Mid-level)

> ISO 25010 I don't have experience but there are elements which might be beneficial for us like compatibility and reliability. This should be organization level. These should be part of our goals in planning phase. Similarly, how we are doing ISO 27001. Also, usability and should be in discussion for our team like how we can enable better usability with automation. (S7, Management, Senior-level)

6.3.6 Separation of production and pre-production.

One of the possible methods for service quality assurance was separation of production and pre-production cloud systems. This was described by two of the interviewees and seen as crucial. This kind of separation to pre-production has benefits of allowing changes to be tested in near production like environment before they are brought to the production. This will ascertain that non tested changes will not go to production and possibly cause failures and issues.

> Secondly separation of production and pre-production environment is crucial. It allows introduction of changes to production in controlled manner. (S1, Management, Mid-level)

> Then also of course utilization of development and staging environments. Code of the program is published in those environments first. Also, if changes are done to the platform they should be done first to these development and staging environments. Changes should be tested there for a time depending on the extend of the changes and when it works sufficiently enough there it will be deployed to production. So, in this way we can ascertain that non tested things will not go to production. (S3, Development, Senior-level)

6.3.7 Application performance monitoring

Application performance monitoring (APM) was considered as beneficial. This would include mapping out application logic to understand how application is connected to the environment like databases. These internal activities could be monitored and issues like database queries taking too long time could be found. Use of APM may require information from the application vendors as sometimes there is not much visibility to application layer in cloud systems.

> Furthermore, more and more there is own software development some kind of applications performance monitoring to fulfil metrics. In example Elastic or similar application performance monitoring (APM) similarly to Prometheus these need to be instrumented to receive desired data. In example, build a map how different things are connected to each other from the application logic perspective. From where the application is connecting to databases etc. Also, to get these e.g. database queries to application performance monitoring. If queries take too long and so on. Effect of these depend on how big role own software development is in the team. If these can be done more beneficial, they are with production operation. (S2, Management, Mid-level)

White box monitoring where applications internal activity is known. In example Prometheus metrics are gathered from things which have been deemed useful. And that there is not only black box monitoring but there is also knowledge of applications internal metrics and problems which might occur. In example database writing or which kind of actions are happening inside the instance. These help with looking inside the service or application but will require the knowledge of the internal logic of the service or application. In our cloud environment it might require knowledge from the vendors of the applications to be useful. Prometheus and Grafana are good tools. Grafana is good to use so Prometheus queries don't need necessarily to be written by oneself or if are those can be saved to Grafana Dashboards as graphs. Where monitoring and accessing them will be easier. But at minimum metrics should be available so when the problems in production do occur these can be utilized to create graphs or alarms. (S2, Management, Mid-level)

7 DISCUSSION

In this chapter, results and their limitations are discussed. Furthermore, answers to the research question and implications for practice are explored. Purpose of this study was to find beneficial methods or instruments of quality assurance for productionalization of cloud systems in the target organization.

7.1 Research limitations

In this study search results were limited to private cloud systems. Research papers which referred to cloud system which are connected from internet or are cloud services in nature like AWS were excluded from the literary review. This was because the need for the case study in the target organization was for private cloud systems.

Research was mainly executed in one organization and research methods used did not include any quantitative methods. Thus, results may not be applicable to other organizations without additional research validation in other organizations.

There should have been separation between experiences in network function virtualization and containerization technologies during the interviews. It would have allowed better matching of experiences as some of the experiences were clearly from either containerized cloud systems or network function virtualization cloud systems.

7.2 Theoretical contribution

In this study multiple methods and instruments for quality assurance in productionalization of the cloud systems were found. Most of the methods and instruments found from the literature received affirmative or supportive comments from the experts interviewed in this study. Although some of these methods and instruments were not as well known or the experience of using them was limited among the interviewees.

Furthermore, interviewees described multiple other beneficial methods and instruments from their own experience and knowledge in cloud systems. These methods and instruments are not new findings for quality assurance, but results are indicative for these being beneficial in the context of productionalization of cloud systems. Cloud systems were also defined to be in the context of private cloud systems. Public and hybrid cloud systems were not included.

Theoretical contribution from new findings of this study is how relevant these methods or instruments are to the productionalization of cloud systems context. But this would require additional research to ascertain the validity of these new findings. These new methods and instruments found in this study were not validated by searching for relevant literature. As this study was mostly done for the practical contribution purposes for the target organization.

In Tables 4, 5 and 6 quality assurance methods and instruments found in this study are listed. These tables are divided by information quality, system quality and service quality respectively.

7.2.1 General

As some of the findings did not fit to pre-determined categories and were general in nature, these were listed under general category in the Tables 4, 5 and 6.

Most emphasized answers after asking for methods of quality assurance among the interviewees and mentioned by nearly all of them was competency. Having competent team of operators or developers was seen as very beneficial. Results indicate that for quality assurance there should be emphasis on either acquiring enough competencies to the teams or ensuring that there are enough trainings for emerging technologies. As complexities may easily get out of hand, especially if there are already complexity related problems with current technologies used in the cloud systems.

Other methods reported by interviewees were risk-aware methodology, infrastructure as a code (IaaC) model and separations of production and pre-production cloud system environments. Risk-aware methodology is important for critical observation of the cloud systems for locating weak areas and mitigating them. IaaC model was described in the results as there should be no manual configuration of the cloud system. As everything should be documented in the code. This would be beneficial to keep the cloud system's configuration more stable as if there are changes those changes will not be wiped away during updates as would be the case if configuration would have been done manually. Separation of the pre-production cloud system from the production cloud system was emphasized in the results. This would ensure that if changes produce unexpected outcomes these problems would not be affecting the actual production.

There was also short remark about benefits of dynamic documentation by one of the interviewees. Benefits of dynamic documentation include that documentation will represent the production state of the cloud systems rather than time of documentation state.

7.2.2 Benchmarking

Benchmarking was emphasized a lot in the in results by interviewees. Benchmarking in the literature was done in the context of edge computing platforms by Das et al. (2018) and in the context of serverless platforms by Wang et al. (2018). Cloud systems need to be able to provide the required performance or capacity for applications to be able to serve the end users or the customers efficiently and with minimal latency. Benchmarks can be used to test these performance and capacity requirements. To find the limitations of the cloud systems namely the bottlenecks or weak spots. As benchmarks are not often provided by the cloud system vendors. Recording them and benchmarking the cloud systems was seen as beneficial by the interviewees.

CPU and Memory benchmarking was seen as essential and the basis of benchmarking in the cloud systems. End-to-end latencies were said to be beneficial especially from the end user or customer perspective. As benchmarking for end-to-end latency would be closest to the latency which the customer will experience. I/O and Network throughput benchmarks were seen as valuable for troubleshooting and locating and preventing bottle necks.

Connection between cold and warm start latencies with autoscaling was similarly identified in the results as well as in the literature. Autoscaling is extremely time sensitive and lag in instance acquisitions i.e., start latencies could impact the service performance (Zhong & Buyya, 2020).

7.2.3 Monitoring

Problem determination in cloud systems is time consuming and requires resources from automation development to be done effectively (Huang et al., 2007). This seemed to be also the case in the results where the benefits of automatic problem determination were recognized but they were not as widely used. Nevertheless, there was consensus to the benefits of using problem determination.

Problem determination can be used for identifying capacity problems and security threats. Predicting problem situations and providing situational awareness to the status of the cloud systems. Providing abilities to highlight anomalies from base lines of data. Give tools and support for troubleshooting of problems or raised anomalies in cloud systems. Main benefits included reduced workloads due to better troubleshooting capabilities (one of the interviewees mentioned that troubleshooting could be about half of the workload) and reduction of inquiries to the status of the cloud systems. These were very much in line with the literature describing quicker problem resolution and reduced problem search space as some of the benefits (Oliveira et al., 2016).

For monitoring of the cloud systems interviewees described three new findings for the context of productionalization. These were error tracking tools, centralized status dashboard and application performance monitoring (APM). Error tracking tools allow problem management and enable historical view of the errors. Some of these tools were mentioned by name like Sentry and New Relic. Centralized status dashboard was more of a concept described by two interviewees. This was not an instrument which was in use per se, but rather a need for such instrument. This instrument would provide status of multiple clusters in one dashboard. Including information like capacity situation and alarms for problems. APM was described as beneficial for allowing internal activity of the applications to be monitored.

7.2.4 Application requirements

For application requirements (Rodriguez & Buyya, 2019) described methods of fault-tolerance, QoS classes and container priorities in their taxonomy of container based cloud orchestration systems. These methods did get affirmative comments in the results and some of them were described to be basis of the cloud systems like fault-tolerance requirements.

Use of ISO 25010 for measuring application performance described by Ravanello et al. (2014) did not have many experiences among the interviewees. But there were comments, that it could possibly be beneficial in some elements like compatibility and reliability.

7.2.5 Resource utilization

Autoscaling and re-scheduling methods were clearly in use only in some of the systems but by these interviewees responsible for those systems they thought them as beneficial in productionalization. Although it seems that the interviewees understanding and experience in re-scheduling was different from the literature. As in literature it was more of a resource utilization methods (Rovnyagin et al., 2021). Interviewees were mainly describing benefits in maintainability. For autoscaling results simply stated its benefits, autoscaling to be beneficial was also reported in the literature (Zhong & Buyya, 2020).

7.2.6 Security

ISO 27001 was in use in some of the cloud systems of the target organization. What was similar between the results of this study and the literature were the requirements as a reason for using ISO 27001. Alebrahim et al. presented structured method for risk analysis for identifying threats and security requirements according to ISO 27001 standard (Alebrahim et al., 2014). Similar methods were in use in some of the cloud systems at the target organization and seen as beneficial. Weil mentioned protecting customer data and meeting regulatory complains as reason for adopting ISO 27001 (Weil, 2019). Similarly in the results law and customers' requirements were given as reason for adopting ISO 27001 standards.

Third party auditing (TPA) among the interviewees was often brought out in the context of ISO 27001. Especially in some teams ISO 27001 certification seemed to be the only TPA methods in use. In literature TPA had role at guaranteeing data privacy in cloud computing environments and TPA was usually brough up in the context of security (Razaque & Rizvi, 2017). This seemed to be in-line what interviewees described of TPA. As TPA was emphasized in the context of authentication and security. Furthermore, in the results TPA was also described to be beneficial in the context of auditing open-source projects.

Furthermore, for security scanning was proposed by the interviewees as one of the methods for quality assuring cloud system. This would include finding of possible vulnerabilities or old software versions. And was seen as essential for ensuring that the cloud system is up to date.

7.2.7 Deployment

Use of ant colony optimization algorithm for deployment optimization was not known to any of the interviewees. Deployment optimization was also not seen as important due to nature of the cloud systems in the target organization. Thus, similar needs as described in the literature did not exist in the target organization (Lin et al., 2019).

Table 4 Quality assurance methods and instruments of information quality

Instrument/Method	Category	Description	Connection of results to the literature	Articles referenced in
ISO 27001	Security	Being compliant of ISO 27001 and actively auditing for it.	Agrees with previous literature	(Alebrahim et al., 2014; Weil, 2019)
Third party auditor (TPA)	Security	Third party auditor in relation to security of the cloud systems. Third party will audit the system and its security requirements.	Agrees with previous literature	(Razaque & Rizvi, 2017)
Problem determination by using data crawler	Monitoring	Automatic problem determination from logs and creation of dynamic Kibana page for given problem event.	Agrees with previous literature	(Huang et al., 2007; Oliveira et al., 2016)
Error tracking tools	Monitoring	Tracking the errors seen in the logs of the cloud system	-	New finding
Dynamic monitoring frequency.	Monitoring	Changing of monitoring frequency to match volatility of data. In objects like performance or resource usage this algorithm would increase frequency when object is changing more often and decreasing when it stays around the same value.	Agrees with previous literature	(Mastelic & Brandic, 2015)
Dynamic documentation	General	Documenting cloud system with automation so changes will be documented in real-time improving data accuracy and reliability.	-	New finding
Security and scanning related methods	Security	Running scans against the cloud system to find possible vulnerabilities and weak spots	-	New finding
Competency and training related methods	General	Ensuring that required level competency is present in the team and trainings for emerging technologies are done	-	New finding
Centralized status dashboard	Monitoring	Having centralized view to the status of the cloud system and its alarms and capacity.	-	New finding

Table 5 Quality assurance methods and instruments of system quality

Instrument/Method	**Category**	**Description**	**Connection of results to the literature**	**Articles referenced in**
Use of re-scheduler	Resource utilization	Automatically or periodically re-scheduling of instances to better match the new status of the cluster.	Agrees with previous literature	(Rovnyagin et al., 2021)
Cold start and warm start latency	Benchmarking	Benchmarking the launch time of instances. Cold start refers doing it for first time and warm start repeating same launch again.	Agrees with previous literature	(Wang et al., 2018)
CPU utilization	Benchmarking	Benchmarking CPU utilization metric in instances for performance validation.	Agrees with previous literature	(Das et al., 2018; Wang et al., 2018)
Memory utilization	Benchmarking	Benchmarking memory utilization metric in instances for performance validation.	Agrees with previous literature	(Das et al., 2018)
End-to-end latency	Benchmarking	Benchmarking end-to-end latency metric in instances for performance validation.	Agrees with previous literature	(Das et al., 2018)
Time-in-flight	Benchmarking	Benchmarking time-in-flight metric in instances for performance validation.	Agrees with previous literature	(Das et al., 2018)
I/O throughput	Benchmarking	Benchmarking I/O throughput metric in instances for performance validation.	Agrees with previous literature	(Wang et al., 2018)
Network throughput	Benchmarking	Benchmarking network throughput metric in instances for performance validation.	Agrees with previous literature	(Wang et al., 2018)
Use of ant colony optimization (ACO) algorithm	Deployment	Use of ant colony optimization for instance deployment. Ant colony optimization is type of algorithm which attempts to mimic ants real life behavior to find optimal deployment path for virtual machines.	Results were limited, but had some disagreement with previous literature	(Lin et al., 2019)
Risk-aware methodology	General	Having risk-aware approach for operations and development	-	New finding
Infrastructure as a code model	General	Executing changes and managing cloud system in IaaC methods	-	New finding

Table 6 Quality assurance methods and instruments of service quality

Instrument/Method	**Category**	**Description**	**Connection of results to the literature**	**Articles referenced in**
Fault-tolerance requirements	Application Requirements	Fulfilling quality of service requirements of fault tolerance in instances	Agrees with previous literature	(Rodriguez & Buyya, 2019)
QoS classes	Application Requirements	Enforcing quality of service classes in instances	Agrees with previous literature	(Rodriguez & Buyya, 2019)
Container priorities	Application Requirements	Enforcing use of container priorities. This refers to instances having different priorities for accessing resources.	Agrees with previous literature	(Rodriguez & Buyya, 2019)
Autoscaling	Resource utilization	Using automatic scaling for dynamically changing workload through instance acquisition.	Agrees with previous literature	(Kuo et al., 2021; Zhong & Buyya, 2020)
ISO 25010	Application Requirements	Use of ISO 25010 to measure performance of cloud computing applications	Results didn’t confirm previous findings and were limited	(Ravanello et al., 2014)
Separation of production and pre-production.	General	Using pre-production environment for testing the changes before they are implemented to production	-	New finding
Application performance monitoring (APM)	Monitoring	Monitoring the performance of the application by using APM	-	New finding

7.3 Practical contribution

For target organization results indicate that most beneficial methods or instruments for quality assurance for productionalization of cloud systems would be problem determination and benchmarking related methods. As was seen from the results troubleshooting and reducing time of troubleshooting will have impact in quality.

Implementing problem determination should be continuous work where known problem cases would be turned to automatically identifiable cases. So, they would be more easily noticed and predicted, improving the time which it takes to solve these problems or incidents. Results listed possible use cases including identifying capacity problems, security threats and highlighting anomalies from the monitored base line. Reduction in troubleshooting would positively affect the daily workload in the organization and remaining time could be used for development. Thus, improving the overall quality of the cloud systems.

Problem determination could also be paired with another good finding, which was centralized status dashboard. Implementing such a dashboard may have similar effect as it would allow quicker access to information which otherwise will take some time to get to. And in some cases, this information may not be found before incidents. Incidents especially outside of work hours are unnecessary strain in the workload of operators and developers.

Benchmarking as seen from the results was described to be very beneficial. As these benchmarks will not always be available from the vendors it is important to understand the limitations of the cloud system and to be able to compare old systems to new ones. Benchmarking is helpful in finding bottle necks. These would also help in troubleshooting. One of the points mentioned in the results was that benchmarks should be constantly monitored, because if there is no monitoring for the benchmarks their value is diminished to only new installations of the cloud systems.

Furthermore, IaaC model was described to be beneficial as manual configuration changes to cloud systems, which are not done by use of automation nor implemented to the code of the system can deviate the cloud system from expected configuration. This will prove to be issue during updates or further changes in the cloud system.

Some of the other methods and instruments from the results could also be looked for but were not as relevant and are thus not included in the practical contribution.

7.4 Future research

To extend the knowledge of quality assurance for productionalization of cloud systems. In the future research the focus could be more in the industry wide survey for methods and instruments to give better quantitative insight to if all the same methods and instruments are used in same way as seen in this study. Furthermore, the scope of the study was strictly in the private cloud systems, but as in target organization also public cloud and hybrid cloud systems are in use, so these could be included in further studies.

Net-benefits resulting from use of methods and instruments – independent variables i.e., information quality, system quality and service quality were not tested during this research. These as shown in the (DeLone & McLean, 2003, 2002) updated version of IS success model also in Figure 1 should contribute for net-benefits of the cloud systems. Thus, in future research methods and instruments found in this research could be tested for their contribution for net-benefits in the scope of cloud systems to confirm the results of this study.

8 CONCLUSION

In this study beneficial methods and instruments of quality assurance for productionalization of cloud systems were researched from the literature and from the case study done in the target organization. This case study was ordered by the target organization to help with quality assurance of cloud systems during productionalization. As implementing emerging technologies to new cloud systems provide increasing challenges to maintain high quality cloud services.

Literary review was done about IS success, quality measurements and methods of productionalization in cloud systems. From the results of the literary review and the case study 27 methods and instruments of quality assurance were found. With these findings answers to the research question and implications for practice were explored. Purpose of this study was to find beneficial methods or instruments of quality assurance for productionalization of cloud systems in the target organization.

There were several beneficial methods which will provide practical help for quality assuring cloud systems in the target organization. Results indicated that most beneficial methods or instruments for quality assurance for productionalization of cloud systems would be problem determination and benchmarking related methods. As was seen from the results troubleshooting and reducing time of troubleshooting will have impact in quality.

To extend the research in the field these results could be tested, and more case studies done in different organizations. Furthermore, methods and instruments found in this research could be tested for their contribution for net-benefits in the scope of cloud systems to confirm the applicability of these methods.

REFERENCES

Alebrahim, A., Hatebur, D., & Goeke, L. (2014). Pattern-based and ISO 27001 compliant risk analysis for cloud systems. *2014 IEEE 1st International Workshop on Evolving Security and Privacy Requirements Engineering (ESPRE)*, 42–47.

Aral, A., Brandic, I., Uriarte, R. B., De Nicola, R., & Scoca, V. (2019). Addressing Application Latency Requirements through Edge Scheduling. *International Journal of Grid and Utility Computing*, *17*(4), 677–698.

Burns, B., Grant, B., Oppenheimer, D., Brewer, E., & Wilkes, J. (2016). Borg, Omega, and Kubernetes. *Communications of the ACM*, *59*(5), 50–57.

Das, A., Patterson, S., & Wittie, M. (2018). EdgeBench: Benchmarking Edge Computing Platforms. *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*, 175–180.

DeLone, W. H., & McLean, E. R. (1992). Information Systems Success: The Quest for the Dependent Variable. *Information Systems Research*, *3*(1), 60–95.

DeLone, W. H., & McLean, E. R. (2003). The DeLone and McLean model of information systems success: A ten-year update. *Journal of Management Information Systems : JMIS*, *19*(4), 9–30.

DeLone, W. H., & McLean, E. R. (2002). Information systems success revisited. *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, 2966–2976.

Huang, H., Jennings, R., Ruan, Y., Sahoo, R., Sahu, S., & Shaikh, A. (2007). PDA: a tool for automated problem determination. *Proceedings of the 21st Conference on Large Installation System Administration Conference*, 1–14.

Huedo, E., Montero, R. S., Moreno-Vozmediano, R., Vázquez, C., Holer, V., & Llorente, I. M. (2021). Opportunistic Deployment of Distributed Edge Clouds for Latency-Critical Applications. *International Journal of Grid and Utility Computing*, *19*(1), 2.

ISO 27001. (2019). ISO. https://www.iso.org/standard/54534.html

Keen, P. G. W. (1980). MIS research: Reference disciplines and a cumulative tradition. *ICIS 1980 Proceedings*, *9*. https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1016&context=icis1980

Kuo, J. Y., Lin, H. C., & Liu, C. H. (2021). A Study of Resource Utilization Improvement on Cloud Testing Platform. *KSII Transactions on Internet And*. https://www.koreascience.or.kr/article/JAKO202123755970599.page

Lian, J.-W. (2017). Establishing a Cloud Computing Success Model for Hospitals in Taiwan. *Inquiry: A Journal of Medical Care Organization, Provision and Financing*, *54*, 46958016685836.

Lin, M., Xi, J., Bai, W., & Wu, J. (2019). Ant Colony Algorithm for Multi-Objective Optimization of Container-Based Microservice Scheduling in Cloud. *IEEE Access*, *7*, 83088–83100.

Liu, Y., & Wang, T. (2021). Quality factors and performance outcome of cloud-based marketing system. *Kybernetes. The International Journal of Cybernetics, Systems and Management Sciences*, *51*(1), 485–503.

Macmillan English Dictionary. (n.d.). Retrieved March 21, 2022, from https://www.macmillandictionary.com/dictionary/british/productionalization

Mastelic, T., & Brandic, I. (2015). Data Velocity Scaling via Dynamic Monitoring Frequency on Ultrascale Infrastructures. *2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom)*, 422–425.

Myers, M. D. (2019). *Qualitative Research in Business and Management*. SAGE.

Oliveira, F., Eilam, T., Nagpurkar, P., Isci, C., Kalantar, M., Segmuller, W., & Snible, E. (2016). Delivering software with agility and quality in a cloud environment. *IBM Journal of Research and Development*, *60*(2–3), 10:1-10:11.

Pitt, L. F., Watson, R. T., & Kavan, C. B. (1995). Service Quality: A Measure of Information Systems Effectiveness. *MIS Quarterly*, *19*(2), 173–187.

Ravanello, A., Desharnais, J.-M., Bautista Villalpando, L. E., April, A., & Gherbi, A. (2014). Performance Measurement for Cloud Computing Applications Using ISO 25010 Standard Characteristics. *2014 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement*, 41–49.

Razaque, A., & Rizvi, S. S. (2017). Privacy preserving model: a new scheme for auditing cloud stakeholders. *Journal of Cloud Computing*, *6*(1), 1–17.

Rodriguez, M. A., & Buyya, R. (2019). Container‐based cluster orchestration systems: A taxonomy and future directions. *Software: Practice & Experience*, *49*(5), 698–719.

Rovnyagin, M. M., Dmitriev, S. O., Hrapov, A. S., & Kozlov, V. K. (2021). Algorithm of ML-based Re-scheduler for Container Orchestration System. *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, 613–617.

Vennam, S. (n.d.). *What is cloud computing?* Retrieved March 14, 2022, from https://www.ibm.com/cloud/learn/cloud-computing

Wang, L., Li, M., Zhang, Y., Ristenpart, T., & Swift, M. (2018). Peeking behind the curtains of serverless platforms. *2018 USENIX Annual Technical Conference (USENIX ATC 18)*, 133–146.

Weil, T. (2019). Risk Assessment Methods for Cloud Computing Platforms. *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, *1*, 545–547.

What is a private cloud? (2022, March 23). VMware. https://www.vmware.com/topics/glossary/content/private-cloud.html

Wiktionary the free dictionary. (n.d.). Retrieved March 21, 2022, from https://en.wiktionary.org/wiki/productionalization

Yin, R. K. (2009). *Case Study Research: Design and Methods*. SAGE.

Zhong, Z., & Buyya, R. (2020). A Cost-Efficient Container Orchestration Strategy in Kubernetes-Based Cloud Computing Infrastructures with Heterogeneous Resources. *ACM Transactions on Internet Technology*, *20*(2), 1–24.

APPENDIX 1 INTERVIEW QUESTIONS

Interview questions

In this research:
Cloud system is defined as private clouds which utilize either virtualization or containerization technologies
Quality assurance means maintenance of desired level of quality in the cloud systems
Productionalization means deployment of cloud system as a production environment which servers production traffic.

From those methods or instruments, from which you had experience. What benefits these methods or instruments have brought to quality assurance for productionalization of cloud systems?

From those methods or instruments, from which you did not have experience of, do you think some of them would possibly be useful in quality assurance for productionalization of cloud systems?

What other methods or instruments you can name related to quality assurance of cloud systems?

Which kind of challenges do you see in quality assurance for productionalization of cloud systems?

Which kind of benefits do you see in quality assurance for productionalization of cloud systems?

Haastattelu kysymykset

Tässä tutkimuksessa:
Pilvijärjestelmällä tarkoitetaan privaatti pilviä, jotka hyödyntävät joko virtualisointi tai konttiteknologioita.
Laadunvarmistamisella tarkoitetaan halutun laatutason ylläpitämistä pilvijärjestelmissä.
Tuotannollistamisella tarkoitetaan pilvijärjestelmän käyttöönottoa tuotantoympäristöksi, joka palvelee tuotantokuormia.

Niistä menetelmistä tai instrumenteista, joista sinulla oli kokemusta, mitä hyötyjä nämä menetelmät tai instrumentit ovat tuoneet pilvijärjestelmän laadunvarmistamiseen tuotannollistamisen yhteydessä?

Niistä menetelmistä tai instrumenteista, joista sinulla ei ollut kokemusta, pidätkö jotakin tai joitakin niistä mahdollisesti hyödyllisinä laadunvarmistamiseen pilvijärjestelmän tuotannollistamisessa?

Mitä muita menetelmiä tai instrumentteja voit nimetä pilvijärjestelmän laadunvarmistamiseen liittyen?

Minkälaisia haasteita näet laadunvarmistamisesta pilvijärjestelmien tuotannollistamisen yhteydessä?

Minkälaisia hyötyjä näet laadunvarmistamisesta pilvijärjestelmien tuotannollistamisen yhteydessä?

APPENDIX 2 INTERVIEWEE PRELIMINARY QUESTIONNAIRE

Pick the one which most closely represents your role and level of expertise.

	Operations	Development	Management
Role in cloud systems	☐	☐	☐

	Intermediate	Mid-level	Senior-level
Expertise in cloud systems	☐	☐	☐

A: Which of these methods and instruments you have experience?
B: Do you think this method or instrument is important in quality assurance of cloud systems?

A. Mistä näistä menetelmistä tai instrumenteista sinulla on kokemusta?
B. Pidätkö tätä menetelmää tai instrumenttia tärkeänä pilvijärjestelmien laadunvarmistamisessa?

Instrument/Method	**Category**	**Description**	**A**	**B**
Fault-tolerance requirements	Application Requirements	Fulfilling quality of service requirements of fault tolerance in instances	☐	☐
QoS classes	Application Requirements	Enforcing quality of service classes in instances	☐	☐
Container priorities	Application Requirements	Enforcing use of container priorities. This refers to instances having different priorities for accessing resources.	☐	☐
ISO 25010	Application Requirements	Use of ISO 25010 to measure performance of cloud computing applications	☐	☐
Autoscaling	Resource utilization	Using automatic scaling for dynamically changing workload through instance acquisition.	☐	☐
Use of re-scheduler	Resource utilization	Automatically or periodically re-scheduling of instances to better match the new status of the cluster.	☐	☐
ISO 27001	Security	Being compliant of ISO 27001 and actively auditing for it.	☐	☐
Third party auditor (TPA)	Security	Third party auditor in relation to security of the cloud systems. Third party will audit the system and its security requirements.	☐	☐

Problem determination by using data crawler	Monitoring	Automatic problem determination from logs and creation of dynamic Kibana page for given problem event.	☐	☐
Dynamic monitoring frequency	Monitoring	Changing of monitoring frequency to match volatility of data. In objects like performance or resource usage this algorithm would increase frequency when object is changing more often and decreasing when it stays around the same value.	☐	☐
Use of ant colony optimization (ACO) algorithm	Deployment	Use of ant colony optimization for instance deployment. Ant colony optimization is type of algorithm which attempts to mimic ants real life behavior to find optimal deployment path for virtual machines.	☐	☐
Coldstart and warmstart latency	Benchmarking	Benchmarking the launch time of instances. Coldstart refers doing it for first time and warmstart repeating same launch again.	☐	☐
CPU utilization	Benchmarking	Benchmarking CPU utilization metric in instances for performance validation.	☐	☐
Memory utilization	Benchmarking	Benchmarking memory utilization metric in instances for performance validation.	☐	☐
End-to-end latency	Benchmarking	Benchmarking end-to-end latency metric in instances for performance validation.	☐	☐
Time-in-flight	Benchmarking	Benchmarking time-in-flight metric in instances for performance validation.	☐	☐
I/O throughput	Benchmarking	Benchmarking I/O throughput metric in instances for performance validation.	☐	☐
Network throughput	Benchmarking	Benchmarking network throughput metric in instances for performance validation.	☐	☐