Lappeenranta
University of Technology

Leah Riungu-Kalliosaari

EMPIRICAL STUDY ON THE ADOPTION, USE AND EFFECTS OF CLOUD-BASED TESTING

Doctoral dissertation for the degree of Doctor of Science (Technology) to be presented with due permission for public examination and criticism in the Auditorium 1382 at Lappeenranta University of Technology, Lappeenranta, Finland, on 12th of June 2014, at noon.

Acta Universitatis
Lappeenrantaensis 572

Supervisors
Professor Kari Smolander
Department of Software Engineering and Information Management
School of Industrial Engineering and Management
Lappeenranta University of Technology
Finland

Associate Professor Ossi Taipale
Department of Software Engineering and Information Management
School of Industrial Engineering and Management
Lappeenranta University of Technology
Finland

Reviewers
Professor Ilkka Tervonen
Department of Information Processing Science
University of Oulu
Finland

Professor Scott Tilley
Department of Computer Sciences
Florida Institute of Technology
USA

Opponent
Professor Tommi Mikkonen
Department of Pervasive Computing
Tampere University of Technology
Finland

ISBN 978-952-265-575-2
ISBN 978-952-265-576-9 (PDF)
ISSN-L 1456-4491
ISSN 1456-4491

Lappeenrannan teknillinen yliopisto
Yliopistopaino 2014

Abstract

Leah Riungu-Kalliosaari
Empirical study on the adoption, use and effects of cloud-based testing
Lappeenranta, 2014
105 p.

Acta Universitatis Lappeenrantaensis 572
Diss. Lappeenranta University of Technology
ISBN 978-952-265-575-2, ISBN 978-952-265-576-9 (PDF)
ISSN-L 1456-4491, ISSN 1456-4491

Cloud computing is a practically relevant paradigm in computing today. Testing is one of the distinct areas where cloud computing can be applied. This study addressed the applicability of cloud computing for testing within organizational and strategic contexts. The study focused on issues related to the adoption, use and effects of cloud-based testing.

The study applied empirical research methods. The data was collected through interviews with practitioners from 30 organizations and was analysed using the grounded theory method. The research process consisted of four phases. The first phase studied the definitions and perceptions related to cloud-based testing. The second phase observed cloud-based testing in real-life practice. The third phase analysed quality in the context of cloud application development. The fourth phase studied the applicability of cloud computing in the gaming industry.

The results showed that cloud computing is relevant and applicable for testing and application development, as well as other areas, e.g., game development. The research identified the benefits, challenges, requirements and effects of cloud-based testing; and formulated a roadmap and strategy for adopting cloud-based testing. The study also explored quality issues in cloud application development. As a special case, the research included a study on applicability of cloud computing in game development. The results can be used by companies to enhance the processes for managing cloud-based testing, evaluating practical cloud-based testing work and assessing the appropriateness of cloud-based testing for specific testing needs.

Keywords: Cloud computing, cloud-based testing, software testing, quality, cloud gaming, grounded theory

UDC 004.7:004.41:004.415.53

To the African girl child

Your right and access to education is a cause worth fighting for.

“The future must not belong to those who bully women. It must be shaped by girls who go to school and those who stand for a world where our daughters can live their dreams just like our sons.” – Barack Obama, at the United Nations General Assembly 2012

Acknowledgements

I consider it a blessing to have had the opportunity to carry out this research work. It would not have been possible without the support of many wonderful people. I am not able to mention everyone here, but I acknowledge and deeply appreciate all your invaluable assistance and support.

I would like to thank my supervisors, Professor Kari Smolander and Associate Professor Ossi Taipale, for their guidance, encouragement and contribution throughout this research. Thank you for all your efforts and for providing a great research environment.

I wish to thank the reviewers of this dissertation, Professor Ilkka Tervonen and Professor Scott Tilley for your valuable comments and feedback that helped me to finalize the dissertation.

I would like to express my gratitude to Dr. Ita Richardson for your warm reception and guidance during my research visit at the Irish Software Engineering Research Centre (Lero). Thank you also for your efforts as a co-author. Additional thanks to another co-author, Dr. Jussi Kasurinen, for your contributions.

For financial support, I would like to acknowledge the following sources: The Finnish Funding Agency for Technology and Innovation (TEKES), the Graduate School on Software Systems and Engineering (SoSE), European Union Regional Development Fund project administered by the Council of South Karelia, the companies participating in ESPA, STX and SOCES research projects, LUT Foundation and the Irish Software Engineering Research Centre (Lero) grant 10/CE/I1855.

I appreciate the support and assistance of my colleagues at the Department of Software Engineering and Information Management. Special thanks to Tarja Nikkinen, Ilmari Laakkonen and Petri Hautaniemi, for providing administrative and technical support.

My many friends, both near and around the globe, thank you for sharing my joys and frustrations. This dissertation is probably a little bit late, but better late than never. Thank you - Joan Mallai and family, Seija and Sakari Kiiskinen, and Pirkko and Martti Rouhiainen – for being like family away from home.

Special gratitude to my family, mother Isabella, siblings Kagendo, Mugambi, Murage, Martin and Edwin, parents-in-law Eila and Jorma Kalliosaari, brother-in-law Sami, and the extended family for unconditional support and encouragement. I fondly

remember my late father Eustace Riungu Muchiri, who highly upheld the value of education and encouraged all my academic endevours. Death did not allow him the opportunity to celebrate this milestone; I know he would have been very pleased.

My dear husband Jyrki, I am truly grateful for your love, patience, understanding and support.

Lappeenranta, March, 2014
Leah Riungu-Kalliosaari

List of publications

I. Riungu, L.M., Taipale, O., and Smolander, K., 2010. “Research Issues for Software Testing in the Cloud”, In Proceedings of the 2nd International Conference on Cloud Computing Technology and Science (CloudCom), pp. 557-564.

II. Riungu-Kalliosaari L., Taipale, O., and Smolander, K., 2013. “Software Testing as a Service: Perceptions from Practice”, Book chapter appearing in the book, Software Testing in the Cloud: Perspectives on an Emerging Discipline, IGI Global, pp. 196-215.

III. Riungu-Kalliosaari, L., Taipale, O., and Smolander, K., 2012. ”Testing in the Cloud: Exploring the Practice”, Special Issue on Cloud Computing for Software Engineering, IEEE Software, Volume 29, Issue 2, pages 46-51.

IV. Riungu-Kalliosaari, L., Taipale, O., Smolander, K. and Richardson, I. “Adoption and Use of Cloud-Based Testing in Practice”, Submitted to the Software Quality Journal.

V. Riungu-Kalliosaari, L., Taipale, O., and Smolander, K., 2013. ”Desired Quality Characteristics in Cloud Application Development”, Proceedings of the 8th International Joint Conference on Software Technologies (ICSOFT), pp. 303-312.

VI. Riungu-Kalliosaari, L., Kasurinen, J., and Smolander, K., 2013. ”Cloud Services and Cloud Gaming in Game Development”, Proceedings of Game and Entertainment Technologies (GET), 22-24.7.2013, Prague, Czech Republic, pages 197-206.

In this dissertation, these publications are referred to as *Publication I*, *Publication II*, *Publication III*, *Publication IV*, *Publication V*, and *Publication VI.*

After *Publication I,* my surname changed from Riungu to Riungu-Kalliosaari – due to getting married.

Symbols and abbreviations

API	Application Programming Interface
BPEL4WS	Business Process Execution Language for Web Services
COTS	Commercial-off-the-shelf
CRM	Customer Relationship Management
EU	European Union
GSD	Global Software Development
Haas	Human as a Service / Human -as-a-Service
HD	High-definition
IaaS	Infrastructure as a Service / Infrastructure -as-a-Service
ISO	International Organization for Standardization
IT	Information Technology
MDD	Model-Driven development
NIST	National Institute of Standards and Technology
OSS	Open Source Software
PaaS	Platform as a Service / Platform-as-a-Service
QoE	Quality of Experience
QoS	Quality of Service
SaaS	Software as a Service / Software-as-a-Service
SDLC	Software Development Life Cycle
SLA	Service Level Agreement
SME	Small and Medium sized Enterprise

SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
STaaS	Software Testing as a Service / Software-Testing-as-a-Service
SUT	System Under Test
TaaS	Testing as a Service / Testing-as-a-service
ToS	Terms of Service
VM	Virtual Machine
WSDL	Web Services Description Language
XP	eXtreme Programming

Table of Contents

1 Introduction

In the information technology (IT) industry, software products and services are constantly evolving due to changing technologies, trends and market requirements. Innovations towards service-oriented architecture (SOA) and software-as-a-service (SaaS) models have greatly affected the nature of software systems and organizations (Collard, 2009; Goth, 2008). As software products and services continue to evolve, it implies that the methods, tools and concepts to test them must also change. Over the last five years, a new paradigm called cloud computing has attracted the attention of many IT practitioners and researchers. There have been different descriptions of cloud computing (Armbrust et al., 2009; Buyya et al., 2009; Geelan, 2009; Mell & Grance, 2011). For example, Armbrust et al. (2009) look at cloud computing from the point of view of software applications delivered as services, while Buyya et al. (2009) pay attention to the technical concepts that cloud computing is based on, such as, parallel and distributed computing.

Mell and Grance (2011) define cloud computing in terms of its essential characteristics (e.g. measured service and resource pooling), service models (e.g. platform as a service), and deployment models (e.g. public cloud and hybrid cloud). Even though the definition of cloud computing has been subjected to many debates, cloud computing is viewed as a "promising paradigm that could enable businesses to face market volatility in an agile and cost-efficient manner" (Hassan, 2011, p.16). The definition by Mell and Grance (2011) is used in this dissertation.

Cloud computing follows the concept of utility computing to provide users with on-demand access to computing resources that are billed on pay-per-use basis (Buyya et al., 2009). There are many opinions about the purpose of cloud computing, how to adopt it and use it in practice. Many have wondered whether cloud computing is an old idea dressed in a new name, or whether they should consider it at all (Voas & Zhang, 2009). Some see it as way to cut down on expenditure, while others see it as a means to attain flexibility and focus on business goals. However, many people are unsure of what they could gain from cloud computing because it introduces more challenges, e.g., security. Cloud computing can be applied to support many business operations to achieve different goals. Therefore, it is important to understand cloud computing in a specific context within which it is applied.

The goal of this dissertation is to *increase empirical knowledge of the adoption, use and effects of cloud-based testing*. In this regard, we use the term cloud-based testing to refer to testing of software applications using infrastructure hosted in cloud computing environments. We address this goal by providing empirical results that help to define and understand cloud-based testing, as well as how it is applied in practice. Because testing is connected to development and quality, the dissertation also looks at cloud gaming and quality in the context of cloud application development.

This dissertation contains a series of empirical studies that focus on cloud-based testing. To provide more evidence on applying cloud computing in different contexts, the dissertation includes a study on cloud gaming. The cloud gaming example demonstrates the importance of considering the specific context within which cloud computing (and consequently, cloud-based testing) is applied. The results in the publications were obtained using qualitative research methods with data collected through interviews from different practitioners. The research method selected for this dissertation was grounded theory, which is an iterative research process for gathering and analysing data with the purpose of denoting theoretical concepts from the data (Corbin & Strauss, 2008).

The contributions of this dissertation are four-fold: (1) We define and understand cloud-based testing by describing the facets of testing in the cloud, and investigating the benefits, challenges, requirements and effects of cloud-based testing. (2) We provide insights on the applicability of cloud-based testing in practice and develop a roadmap and strategy for adopting cloud-based testing. (3) We explore quality issues in cloud application development. (4) We provide insights on the applicability of cloud computing within the gaming industry.

This dissertation consists of two parts, an introduction and an appendix containing the scientific publications. The introduction presents the background of the research area along with the research goal, research question, research method, overview of the publications, and a summary of the research contributions. The appendix is composed of the publications, which contain the research results in detail. Five of them have been through a scientific referee process and *Publication IV* is submitted to a journal for the evaluation process.

The introduction contains six chapters. Chapter 2 presents the literature review discussing the topics covered by the dissertation i.e. cloud computing, cloud-based testing, software quality assurance in the context of cloud computing, and cloud gaming. Chapter 3 describes the research process including the research questions and the research method. Chapter 4 summarizes the results contained in the publications. Chapter 5 discusses the contributions and their implications for research and practice. It also evaluates the research and mentions the limitations of the research. Chapter 6 summarises the contributions of dissertation and proposes future research topics.

2 Cloud computing and cloud-based testing

The purpose of this chapter is to set the scope of the study by describing the definitions and concepts related to this dissertation. The definitions and concepts are obtained from the related research and literature. Before going into the descriptions of cloud computing and cloud-based testing, a brief description of the software technology trends preceding cloud computing is presented.

2.1 Software technology trends during the 21st century

The software industry has changed significantly during the past decades. In an analysis of the future trends and implications for software engineering, Boehm (2006) identified eight trends that would befall the software industry. These trends are:

- increased emphasis on agility, usability and end value.
- increased emphasis on software criticality and dependability.
- increasing needs for commercial-off-the-shelf (COTS) systems and components, open source and legacy systems.
- model-driven development (MDD).
- the increasing interaction of software engineering and systems engineering.
- increasing global connectivity and need for systems to interoperate.
- increase of software-intensive systems of software.
- increasing computing capabilities across different platforms and applications.

By the start of the 21st century, the software industry had transformed from the use of formal and waterfall development procedures, to an escalated inclination towards agile methods such as Adaptive Software Development, eXtreme Programming (XP), Feature Driven Development, and Scrum (Boehm, 2006). Agile development methods primarily focus on building functional and usable software within short iterations. The agile manifesto was published in 2001, highlighting four main principles for agile software development: (1) individuals and interactions over processes and tools; (2) working software over comprehensive documentation; (3) customer collaboration over contract negotiation; and (4) responding to change over following a plan (Agilemanifesto.org, 2001).

Alongside agile methods, there was also an increase in open source software (OSS) and commercial-off-the-shelf (COTS) systems and components. There were some shortcomings with OSS and COTS such as difficulties with updating the software and the lack of "usability, dependability, interoperability and localizability in different countries and cultures" (Boehm, 2006). To address some of these problems, model-driven development (MDD) was applied. MDD focuses on developing software for a specific domain such as banks or automobiles, and it leads to the development of interoperable applications (Boehm, 2006). However, the main challenge with MDD was its inability to adapt to the changing software infrastructure caused by massive distribution, mobile computing and evolving Web objects (Boehm, 2006). To support the needs of the evolving software infrastructure requirements, there was a rise in the use of object-oriented models and service-oriented architectures (SOAs), which led to greater interaction between software systems.

Service-oriented architecture (SOA) became popular and was widely explored (Ebert, 2008; Gold et al., 2004; Goth, 2008; Sahoo, 2009). SOA was seen as "logical way of designing a software system to provide services either to end-user applications or other services distributed in a network, via published and discoverable interfaces" (Papazoglou et al., 2007, p.38). SOA is based on open Internet-based standards, such as Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL) and Business Process Execution Language for Web Services (BPEL4WS) and uses the Internet as the communication channel. The application of open standards to implement SOA helped to produce loosely coupled and interoperable software applications, referred to as services.

A service is "an act or performance offered by one party to another. Although the process may be tied to a physical product, the performance is essentially intangible and does not normally result in ownership of any of the factors of production" (Lovelock, 2000, p.3). Extending this definition to software engineering, service-based software, better known as software as a service (SaaS), is defined as software that is configured and executed to meet a specific set of requirements at a point in time and then discarded after use (Bennett et al., 2000). When delivering software as a service, a SaaS vendor handles the IT infrastructure, hosting, maintenance and support services (Sahoo, 2009; Sun et al., 2007), and provides fully functioning software that can be configured and used at delivery time (Turner et al., 2003). By using SaaS, end users are able to reduce the Total Cost of Ownership (TCO) while using software that is tailored to suit their unique needs and working environments (Olsen, 2006; Sun et al., 2007).

The use of SaaS influenced a service-oriented mindset across the IT industry. Organizations wanted to quickly deliver value to the users by providing software as a service (SaaS)-based applications (Sahoo, 2009; Turner et al., 2003). SaaS supported the evolution of software from being delivered as a product, to being delivered as a service that was provided to users on a demand basis. Over time, it became common for people to access online content with limited knowledge about the data centers hosting the content – a paradigm that evolved into cloud computing (Buyya et al., 2009).

2.2 Cloud computing

The advent of cloud computing generated many attempts to give it a descriptive, clear and understandable definition. Geelan solicited for cloud computing definitions from twenty-one experts (Geelan, 2009). For example, one of the experts defined cloud computing as Internet centric software which shifts the traditional single tenant software to scalable, multi-tenant, multi-platform, multi-network, and global software (Geelan, 2009). Another expert described cloud computing as being the capability to access resources and services needed to perform functions with dynamically changing needs, such that an application or service developer can request access from the cloud rather than a specific endpoint or named resource (Geelan, 2009).

Buyya and his team define "a cloud as a type of parallel and distributed system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resource(s) based on service-level agreements established through negotiation between the service providers and consumers" (Buyya et al., 2009, p.601). Parallel computing refers to multiple processors communicating with each other while located in the same machine (Sulistio et al., 2004). On the other hand, distributed computing has to do with different machines located in different physical locations and communicating with each other through networks such as the Internet (Sulistio et al., 2004). This definition emphasises two unique features of cloud computing:

- **Virtualization:** this is the capability that computing resources have to enable a single physical machine to function and present the illusion of many smaller virtual machines (VMs) running different operating system instances (Barham et al., 2003; Buyya et al., 2009). The VMs can host isolated operating system environments and make use of different portions of the resources – on the same physical machine (Buyya et al., 2009).

- **Dynamic provisioning:** a technique whereby the computing resources are started and stopped to meet the changing demand by spreading the load among the available virtual machines under the specified conditions (Buyya et al., 2009; Lu et al., 2013).

Armbrust and his team referred to cloud computing as "the applications delivered as services over the internet and the hardware and systems software in the data centers that provide those services" (Armbrust et al., 2009, p.4). They described the services delivered over the internet as the earlier known software as a service (SaaS); and the data center hardware and software as a cloud; which could either be a public cloud (available to everyone) or a private cloud (intended to a specific organization).

According to the National Institute of Standards and Technology (NIST), cloud computing is "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." (Mell & Grance, 2011, p.2). The NIST definition descomposes the cloud model into five essential characteristics, three service models and four deployment models (Figure 1).

The following text summarizes the essential characteristics, service models and deployment models of cloud computing as described by Mell and Grance (2011), and adapted from *Publication I*. Cloud computing has the following essential characteristics:

- **On-demand self-service:** Computing services such as servers, storage and virtual machines can be acquired automatically if and when needed by a user without human interaction with the cloud service providers.
- **Broad network access:** Computing services can be accessed over a network using different devices e.g. laptops and mobile phones.
- **Resource pooling:** Computing services are pooled for use by many users through a multi-tenant approach that enables the services to be allocated as per the demand of the users.
- **Rapid elasticity:** Computing services are unlimited and can quickly be scaled in and out as required by the customers.
- **Measured service:** Appropriate metrics like storage, active user accounts, and bandwidth are used to measure the usage of the computing services. This provides transparency of the utilized service to the cloud service provider and customer.

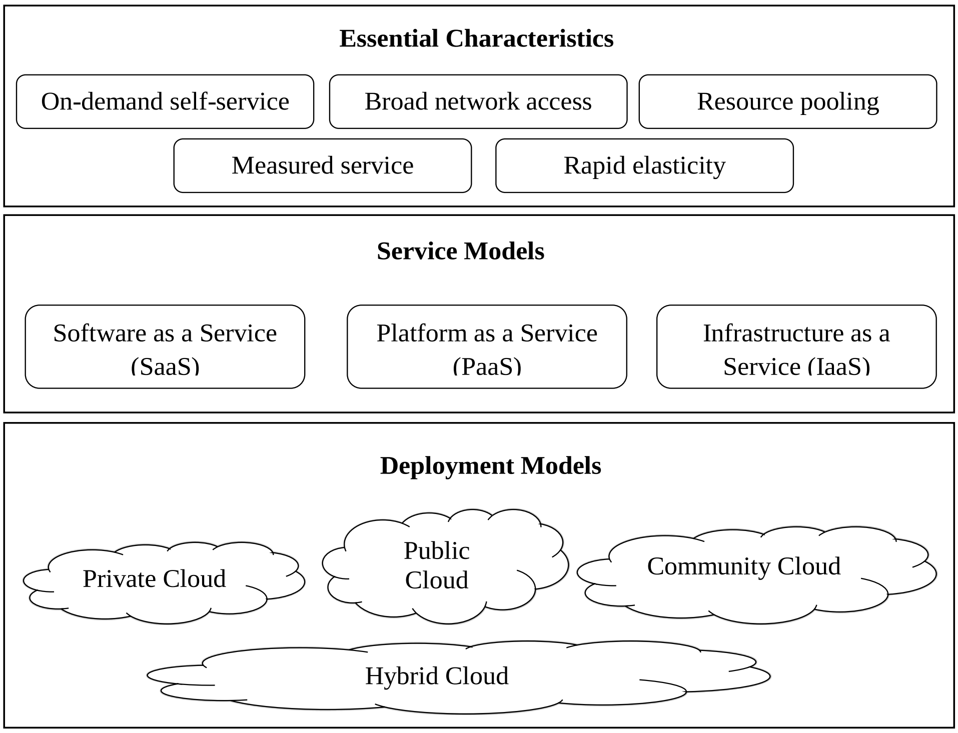


Figure 1: A visual overview of the NIST definition of cloud computing

The cloud contains three main layers referred to as service models (Mell & Grance, 2011). These include:

(1) **Software as a Service (SaaS):** Customers are able to make use of applications that are running in cloud environments, usually by means of a web browser. The customer does not have the rights to control or manage the cloud infrastructure that is running the software.

(2) **Platform as a Service (PaaS):** In this case, customers are provided with programming and execution environments through which they can run and access applications of their own choice. Similar to the SaaS model, customers cannot control the underlying cloud infrastructure but have control over the applications they create and to a certain degree, configuration settings of the hosting environment.

(3) **Infrastructure as a Service (IaaS):** This is where computing services such as storage, processing and networks are provided by the IaaS provider for the customers to deploy and run their applications. IaaS gives a customer

the flexibility to control and run software over the computing environment.

In addition to the three service models mentioned in the NIST definition of cloud computing, Lenk et al. (2009) proposed two additional service layers, namely Human as a Service (HaaS) and supporting services. These layers can also be viewed as service models. Human as a Service (HaaS) incorporates crowdsourcing so that a crowd of people can use the cloud technology from different geographical places and work together to complete a task requiring effort from a large group of people. A good example of crowdsourcing is uTest, a company that provides software testing solutions to its customers through on-demand access to a community of professional testers (uTest, 2013). Another form of HaaS is Information Aggregation Services (IAS) that deals with generating a unified figure that represents a popular opinion of the crowd (Lenk et al., 2009).

Cloud computing can be implemented through different clouds, referred to as deployment models. These deployment models are:

- **Private cloud:** The cloud infrastructure operates mainly to serve one organization only and may be managed by the organization itself or an external cloud provider.

- **Community cloud:** Several organizations share the cloud infrastructure and provide services to a specific community that has similar needs.

- **Public cloud:** The cloud infrastructure is available for use by anyone and is usually owned by a large organization.

- **Hybrid cloud:** This is a combination of two or all of the above-mentioned clouds.

The above definitions describe cloud computing in terms of its composition (computing resources and services), distribution channel (internet/networks), acquisition approach (on-demand, pay-per-use) and stakeholders (providers, users). Cloud computing is enabling the delivery of computing as a utility (Armbrust et al., 2009), which Buyya, et al. (2009) envision to be a 5th utility in the future (in addition to water, electricity, gas and telephony).

The NIST definition of cloud computing has gained wide acceptance and is used in this study. The research also adopts the view about Human as a Service (HaaS) being another deployment model of cloud computing (Lenk et al., 2009), because it supports the delivery of services through crowdsourcing.

According to the literature, some benefits of cloud computing include:

- **Reduced costs:** Cloud computing provides quick access to computing resources without upfront capital investments for the users, which helps to reduce the total expenditure for an organization (Marston et al., 2011). Cloud computing also frees the users from the expense and effort of buying, installing and maintaining their own computing resources (Leavitt, 2009).
- **Availability:** Large cloud service providers that have abundant resources and redundant equipment may offer more availability than if the resources are hosted within the premises of smaller organizations (Leavitt, 2009).
- **Scalability:** Cloud computing enables the users to acquire and free the resources according to the changing demands (Marston et al., 2011).
- **Flexibility:** Leavitt (2009) claims that most cloud computing vendors don't require contracts and let users work with their services as needed, which makes the cloud a convenient way of getting additional resources for activities such as testing new applications and services.
- **Possibilities for new applications:** Cloud computing enables the availability and delivery of applications and services that might not have been possible earlier, such as data and resource intensive business analytics and parallel batch processing requiring huge amounts of processing power (Marston et al., 2011).
- Other benefits include lower costs of entry for smaller organizations, which boosts innovation for start-up companies (Marston et al., 2011); and the ability to integrate many applications and services into powerful composite systems, for example, Salesforce.com (Leavitt, 2009).

Many concerns have prevented the adoption of cloud computing. Some of these concerns are:

- **Security:** For many IT organizations, security and privacy have been primary concerns. For example, the privacy regulations in some European countries prohibit some types of personal data from being distributed outside the European Union (EU). This made organizations such as Microsoft and Amazon to locate some data centres in the EU and allow the users to select the data centre using geographical preferences (Sultan, 2011).
- **Loss of control:** Many IT organizations have been afraid of losing control of their operations to a third party who could change the underlying technology without the knowledge and permission of the customers (Marston, et al. 2011; Sultan, 2011).
- **Lack of cloud computing standards:** There have been concerns regarding the lack of interoperability between services from different cloud providers. The efforts to address this have come from cloud service providers such as Google and Microsoft, industry experts, independent organizations (e.g. the cloud computing interoperability forum) as well as the International Organization

for Standardization (ISO) (Marston et al., 2011). For example, the ISO has a study group that is working on cloud computing standards (ISO, 2013).

- **Vendor lock-in:** Due to the lack of standards, cloud providers offer their services through provider-specific interfaces (Sultan, 2011). This further limits the interoperability between the cloud services, resulting in a situation whereby users are 'stuck' with one cloud service provider, even when the users might wish to move to another provider.
- **Reliability:** Some of the cloud service providers have experienced service outages in the past, the most current being in August 2013 by Amazon (Amazon's online retail operation was off for about an hour) and Google (a four-minute blackout across all applications). Service outages can be a serious problem for the users and the cloud provider, especially in the case of prolonged outages.

Despite the concerns surrounding cloud computing, an increasing number of organizations have migrated some of their business operations to the cloud. Some of the areas in which cloud computing is being applied are (Sheehan, 2013):

- File storage, synchronization and sharing, e.g., dropbox.
- Financial services/eCommerce such as Amazon and eBay.
- Email, e.g., Gmail.
- CPU intensive applications across various domains such as healthcare, social media and advertising.
- Going global, i.e., utilizing different data centres across the world.
- Test and development, e.g., load testing.
- Short-term projects, for instance, setting up a microsite for a few months.
- Handling seasonal capacity, for example, when there is a surge in the number of users in need of your web applications.
- Business intelligence/big data for businesses to find out more about their prospects and customers.
- Website hosting.
- Conducting proof of concepts.
- Web and mobile applications for social media, online communities and blogging.
- Advertising, for example, by hosting advertising servers in the cloud.

2.3 Software testing

This section starts with a brief description of software testing, followed by cloud-based testing

2.3.1 What is software testing?

There are many definitions of software testing in literature. According to Myers (2004), testing is the process of executing a program with the intent of finding errors. By finding and removing errors, one ascertains that the program does what it is supposed to do and does not do what it is not supposed to do; hence improving the reliability and quality of the program (Myers, 2004).

Kit (1995) defines testing as verification and validation.

Verification is the process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase (IEEE/ANSI, 1990). The aim of verification is to answer the question: are we building the product right? Basic verification methods are inspections, walkthroughs and technical reviews. Checklists are verification tools that can be used to verify, for example, requirements, technical design and code.

Validation is the process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements (IEEE/ANSI, 1990). Validation ensures that the built product satisfies the user requirements (as described in the requirements specification) and that the product's behaviour is according to the expected behaviour (as stated in functional specification). Validation includes activities such as unit testing, integration testing, usability testing, function testing, system testing and acceptance testing.

Testing can be divided into two groups of testing methods: black-box testing and white-box testing. Black-box testing deals with the functional specifications (i.e. what the software can do) and white-box testing is concerned with the technical specifications (i.e. the internal structure of the software).

Black-box testing includes testing methods such as syntax testing (systematic method to generate valid and invalid inputs to a program), state transition testing (using finite-state machines to design the tests), equivalence partitioning (identifying equivalence classes and test cases), error guessing (using intuition and experience to guess), cause-effect graphing (transform a natural-language specification to a formal-language specification) and graph matrices (using graphs to represent and organize data) (Kit, 1995).

White-box testing methods, include, for example, statement coverage (executing each statement at least once), decision coverage (each decision or branch takes on all possible outcomes at least once), condition coverage (each condition in a decision takes on all possible outcomes at least once) and path coverage (all possible combinations of condition outcomes in each decision occur at least once) (Kit, 1995).

Testing techniques can also be divided into static and dynamic testing (Heiser, 1997). Static testing may also be referred to as static analysis, and it includes testing techniques that do not involve executing the software, e.g., reviews, walk-throughs, inspections, audits, program proving, symbolic evaluation and anomaly analysis. Dynamic testing refers to any testing technique that involves executing the software, for example boundary-based techniques, decision table-based methods, statistical techniques, path testing, and data flow testing (Heiser, 1997).

Kit's (1995) definition of software testing i.e. verification and validation, does not link software testing to any specific software development method or life cycle model and it is widely applied in software testing and quality practices (Taipale, 2007). For these reasons, this definition was used in this dissertation.

2.3.2 Cloud-based testing

By the year 2009, the IT industry was experiencing a shift towards service-oriented architectures (SOA) and software-as-a-service (SaaS) (see Section 2.1). Convinced about this shift, organizations such as Google started reconsidering their development and testing practices by taking into account the SOA and SaaS dynamics (Goth, 2008). With regard to testing, the terms "software testing as a service (STaaS)", "testing as a service (TaaS)", "testing in the cloud", "cloud testing" and "cloud-based testing" are now widely used as synonyms. As can be seen in *Publication II*, at the beginning of this study, the existing literature about cloud-based testing was in the form of industrial white papers and reports. However, as the field has evolved and gained popularity, scientific research has also been growing (Priyanka et al., 2012).

Cloud-based testing is essentially an intersection of two areas: software testing and cloud computing. Linthicum (2010) defines testing-as-a-service as the ability to test local or cloud-delivered systems using testing software and services that are remotely hosted. He emphasizes that while a cloud service requires testing unto itself, testing-as-a-service systems have the ability to test other cloud applications, web sites, and internal enterprise systems, and they do not require a hardware or software footprint within the enterprise.

In an industrial white paper, van der Aalst defines software testing as a service (STaaS) as "a model of software testing used to test an application as a service provided to customers across the Internet" (van der Aalst, 2009, p.1). STaaS enables daily operation, maintenance and testing support through web-based browsers, testing frameworks and servers.

A typical STaaS process involves interaction between a customer and STaaS provider in a demand and supply fashion (Figure 2). The customer sends a test demand through the Internet to the STaaS provider who handles the testing task and sends the customer the test results (van der Aalst, 2009). The provider needs to manage the test infrastructure and availability of the test service, as well as take into account the overall communication with the customer.

In conceptualizing testing in the cloud, we defined facets of testing in the cloud in *Publication I.* These are (1) the system or application under test is accessible online. The system under test (SUT) might be SaaS software or non-SaaS software. In addition, this includes testing at different test levels e.g. performance testing; (2) testing infrastructure and platforms are hosted across different deployment models of the cloud i.e. public, community, private or hybrid clouds; (3) testing of the cloud itself (Figure 3). To add on the human as a service (HaaS) layer onto testing in the cloud, crowdsourced testing is also a form of testing in the cloud whereby crowdsourced pools of testers from around the globe provide testing services to customers. A practical example of crowdsourced testing is uTest (uTest, 2013).

uTest (2013) uses a community of professional, crowdsourced testers to test mobile, web and desktop applications for multiple testing types (e.g. functional, security, load, localization, and usability). A project manager is assigned to work closely with the customer to understand the customer's testing needs, select and manage testers, assess the results and minimize overheads to the customer's in-house team.

Cloud-based testing introduces new requirements and features for the testing process (Gao et al., 2011). These include: (a) **Cloud-based testing environments** containing different computing resources, system infrastructures and tools that are scalable and can be provisioned for use as needed. (b) **Service level agreements (SLAs)**: Just like in other cloud services, service level agreements are used to describe the testing and quality assurance requirements e.g. reliability, availability, security and performance

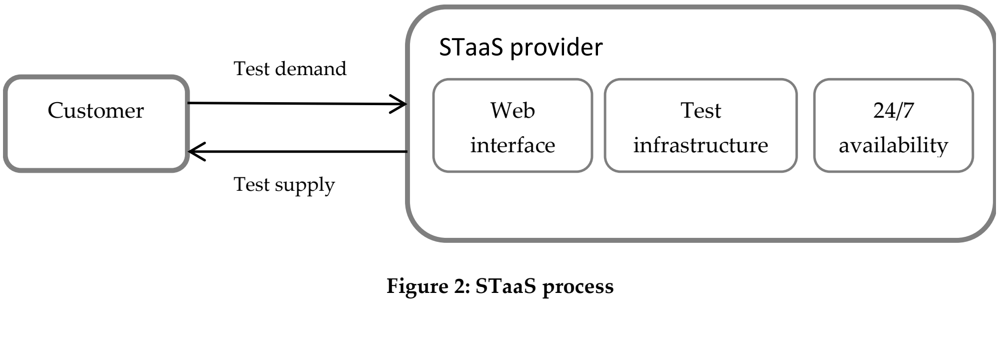


Figure 2: STaaS process

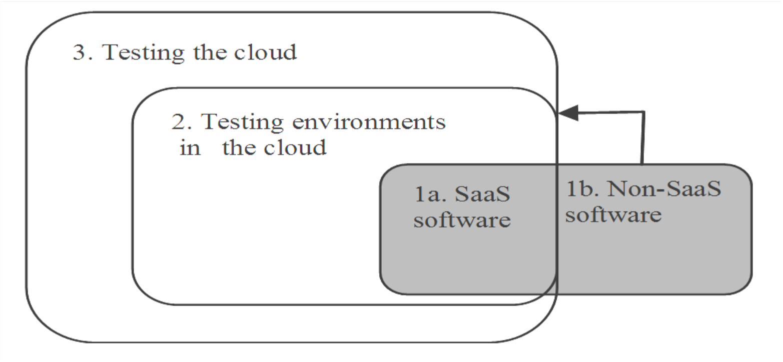


Figure 3: Facets of testing in the cloud

agreements. (c) **Price models and service billing**: The computing resources, infrastructures, and tools are charged along with the testing services based on pre-defined cost models. (d) **Large-scale cloud-based data and traffic simulation**: Applying and simulating online user accesses and traffic data is necessary in cloud-based testing, for example, during performance testing.

A cloud-based testing framework should be based on a reliable and scalable infrastructure that can handle changing resource demands with service transparency and data confidentiality (Wu et al., 2011). The capabilities of a cloud-based testing service can be supported by an architecture containing several layers. For example, Yu et al. (2010) developed a testing as a service (TaaS) architecture with five layers.

(1) **Service and contributor layer**. This layer allows interaction between the users and the TaaS platform, in order for the user to access the testing services.
(2) The **task management layer** acts as a testing service bus that handles all the activities associated with a particular test task e.g. checking the testing capabilities, scheduling and dispatching the tasks and publishing the services. This layer is also responsible for dispatching the scheduled tasks to the VMs and collecting the results.
(3) The **resource management layer** monitors the resources such as the physical and virtual machines, and assigns them to perform testing as required.
(4) The **test layer** generates the test cases, handles test execution and gets the results.
(5) The **database layer** stores the test tasks, service images, and bug tracking results.

According to Robinson & Ragusa (2011), there are various scenarios for utilizing cloud-based testing infrastructure. These are:

i. *Testing in the Cloud*: A simple model of using a single cloud infrastructure provider for hosting the software under test and test suite.

ii. *System under test in the Cloud*: The tester places the system under test on a cloud provider's infrastructure but hosts the test suite locally. The remote system under test receives the test payloads over the Internet. The software under test could be owned either by the tester or by a software-as-a-service (SaaS) solution under test.

iii. *Test suite in the Cloud*: The testing service is provided such that the cloud testing service provider hosts and controls the test suite outside the domain of the system under test.

iv. *Multi-site testing in the Cloud*: More than one provider hosts both the system under test and the test suite in different domains.

v. *Brokered multi-site testing in the Cloud*: The same as scenario IV but includes an intermediary for brokering test and infrastructure management requests and operations. The broker also provides support for resources selection, reservation and deployment.

Cloud-based testing supports testing across various testing levels including unit, integration, system and acceptance testing (Incki et al., 2012). It is also possible to perform different types of testing in the cloud e.g. functional, performance, security, compatibility and interoperability testing (Incki et al., 2012; Wu et al., 2011), model-based testing, symbolic testing, fault injection testing, random testing and privacy aware testing (Priyanka et al., 2012).

There has been an increase in the literature about testing in the cloud over the last few years (Incki et al., 2012; Priyanka et al., 2012). This demonstrates the importance of the topic and need for further investigation. The success of testing in the cloud will rely on how researchers and industry practitioners address all aspects related to its adoption, implementation, delivery and management.

2.4 Software quality and cloud application development

The key objectives of software engineering are reducing costs and improving the quality of the products (Osterweil, 1997). The quality of a software product is achieved as a result of actions performed throughout the software development life cycle (SDLC), including planning, design, implementation and testing (Seth et al., 2012). In particular, testing has a vital influence on the quality of a software product (Kasurinen et al., 2011; Tassey, 2002) and it needs to be taken into account to produce quality software. Other factors that affect quality are for example, outsourcing and communication (Kasurinen et al., 2011) as well as the work culture and individual attitudes (Wilson & Hall, 1998).

2.4.1 What is software quality?

Software quality is a complex concept that is difficult to define (Blaine & Cleland-Huang, 2008; Kitchenham & Pfleeger, 1996; Sommerville, 2001). When dealing with software systems, it is especially difficult to find a common and agreeable definition due to problems such as (Sommerville, 2001, p. 536):

i. The requirement specification should be oriented towards the characteristics of the product that customers wants. However, the development organization may also have requirements e.g. maintainability requirements which are not included in the specification.

ii. We do not know how to specify certain quality characteristics in an unambiguous manner, e.g., maintainability.

iii. It is very difficult to write complete software specifications. Therefore, even though a software product may conform to its specifications, users may not consider it to be a high-quality product.

The third problem mentioned above indeed demonstrates that software quality is a multi-dimensional concept that means different things for different people (Kusters et al., 1997). The ISO 9000 (2000) quality standard defines quality as the degree to which a set of inherent characteristics fulfil the requirements. In the ISO/IEC 25010 standard (ISO/IEC, 2010), software product quality is defined as the degree to which the software product satisfies stated and implied needs when used under specified conditions.

The ISO/IEC 25010 quality standard contains two parts for software systems - product quality model and quality in use model. The product quality model applies to computer systems and software products. It is made up of eight characteristics; functional suitability, reliability, performance efficiency, usability, security, compatibility, maintainability, and portability (ISO/IEC, 2010). Each of these characteristics contains related subcharacterictis e.g. availability as one of the subcharacteristics for reliability and interoperability as a subcharacteristic for compatibility.

The system quality in use model relates to the impact that a computer system or software product has on the users after it is used in a given context. It contains five main characteristics - effectiveness, efficiency, satisfaction, freedom from risk and context coverage together with their associated eleven subcharacteristics, such as, trust and flexibility (ISO/IEC, 2010). Factors that influence the quality in use of a system or product are: the quality of the software, hardware and operating environment as well as the users, tasks and social environment. This dissertation uses the ISO 25010 definition of software product quality and quality in use to describe the quality concept.

Garvin (1984) deduced different perspectives on product quality, and even though his work did not focus on software products, the different perspectives also apply in software engineering. He identified five different perspectives that define quality (Garvin, 1984):

- **Transcendent perspective:** This is a product's 'innate excellence' comprising of high standards that cannot be defined precisely but can be recognized through experience.
- The **product-based perspective** says that the product contains attributes that can be quantified and measured. Therefore, the quality is assessed objectively based on measurable attributes, instead of using preferences.
- The **user-based perspective** focuses on satisfying the individual preferences. However, users have different preferences, and it is not possible to accommodate all of them in one product. Therefore, a high-quality product is assumed to be one that meets the needs of a majority of the users.
- The **manufacturing-based perspective** aims at fulfilling the product requirements and specifications, and any deviation leads to a compromise in the quality.
- The **value-based perspective** evaluates the quality of a product in terms of costs and prices such that a quality product is one that delivers customer satisfaction at an acceptable price.

2.4.2 Cloud application development

Cloud computing is increasingly gaining appreciation for enabling easier, cost-effective and faster development of many applications, for example, e-business, collaborative research and development, enterprise computing infrastructures, healthcare, and military applications (Yau & An, 2011). Cloud computing strengthens the web as an application development environment leading to the delivery of dynamic web applications (Mikkonen & Taivalsaari, 2013). Cloud computing does not only enable application development and testing, but it also supports delivery of the developed and tested applications through (Yau & An, 2011):

- Service virtualization, hence providing easy application deployment and maintenance for service providers.
- Interfaces that provide users with access to the applications.

- Dynamic resource virtualization and allocation, enabling service providers to conduct Quality of Service (QoS) management.

The cloud provides a "complete technology stack covering the ground from database and security to workflow and user interface—so you can focus on assembling, building, and instantly deploying solutions" (Patidar et al., 2011, p.1010). A team from IBM observed that utilizing the cloud for application development can significantly reduce the time required to set up the development environment (Zhou et al., 2011). They developed a cloud platform that organizations can use for developing applications. The development environment incorporates standardized methods and tools that can help to improve the efficiency of application development processes and applications. According to Lu et al. (2010), Microsoft's Windows Azure platform can support the development of computation and data intensive scientific applications.

In order to succeed in utilizing the cloud for application development, there needs to be appropriate development frameworks, environments and tools. For example, Hosono et al. (2011) suggested a framework that streamlines and shortens the flow of development and production (or deployment) activities. Their framework is based on the concept of lifecycle management with supporting tools and roles for each phase of the development lifecycle. The service designer processes the requirements to lay out the design; the resource planner handles the virtual machine (VM) structure; the application implementer develops the application; the service administrator manages the running applications and the cloud administrator manages the virtual machines (VMs) on the physical servers.

When adopting cloud computing for application development, Sodhi & Prabhakar (2011) advocate that the organizations should first evaluate the various platforms and select the one that supports an application's most important quality characteristics. They name three types of development platforms.

i. **Traditional non-cloud platforms**: These are essentially the development platforms where data is stored in regular structured query language (SQL) relational databases, and the designer does not have to take into account any virtualization, elastic and distributed properties.

ii. **Virtualized platforms**: These are are similar to traditional non-cloud platforms, but use virtual machines (VMs) for developing and deploying the applications. The designer does not take into account properties such as elasticity of the platform.

iii. **Cloud-aware platforms**: These platforms hold virtual, elastic and distributed properties supported by robust data centres. Subsequently, the applications developed and deployed on these platforms can also be virtualized, elastic and distributed in nature.

Cloud-based application development requires that various aspects are considered in a different manner from developing applications in conventional software development. When developing applications in the cloud, the developers need to interact with the cloud providers, which is not the case with conventional software development. The developers need to follow formal and secure guidelines in order to align their code with the cloud provider's platform. Different service and deployment models have different technical, legal, security and trust boundaries that determine the kind of interaction that will exist between the developers and the cloud providers (Patidar et al., 2011). For example, using a private cloud provides more control and security to the developer as opposed to when using a public cloud.

2.4.3 Quality in the context of cloud application development

Developing and deploying applications in cloud computing environments introduces features that have significant impact on the quality of the applications. Some of these features are (Hobfeld et al., 2012):

- Increased network distance between the user and the service, e.g., if a user's geographical location is far from the data centre where the service is running from, he/she might experience unacceptable latency which will limit the level of interaction with the service.
- Service delivery implementation, e.g., how to deal with latency and bandwidth bottlenecks.
- Real-time resource management as users access and use the cloud services.
- Having multiple parties involved in providing the service where before there were one or two.
- Geographical distribution of the user base.
- Multi-party sharing and communication, which necessitates real-time requirements and measures for evaluating the quality of service.
- SLAs and pricing.

Lee et al. (2009) came up with a model for evaluating the quality of SaaS applications. They first looked at SaaS features (reusability, customizability, pay-per-use, data managed by providers, scalability and availability) and then derived a set of SaaS-specific quality characteristics (reusability, efficiency, reliability, scalability and availability). Then they defined metrics that could be used to measure each quality characteristic using specific formulae, value ranges, and relevant interpretations. For example, 'reliability' could be measured using the metrics 'service stability' and 'service accuracy'. Their quality model is relevant for both providers and customers because it focuses on SaaS features and therefore, provides guidelines for evaluating and assessing SaaS quality.

The quality of cloud applications can also be assessed by evaluating the quality of the components that make up an application (Zheng et al., 2010). The components are ranked based on their past performance while in use by other users. The results enable the user to select the components that would generate optimal performance for the relevant cloud application. In addition, the application designers can use the quality ranking to assess the performance of the components (Zheng et al., 2010).

Chauhan & Babar (2012) evaluated the quality characteristics of a cloud infrastructure for providing Global Software Development (GSD) teams with a service to support the tools in use. They identified the quality characteristics from studying existing GSD and cloud computing literature. For a cloud infrastructure to efficiently support the delivery of tools as services, it should contain these quality characteristics: multi-tenancy, versioning capabilities, compatibility with commercially available tools, ability to work on decentralized artifacts, support multiple databases, support multiple devices and comply with the SLAs. Chauhan and Babar (2012) also designed architectural solutions that could help to achieve the identified quality characteristics. For example, the architectural solution for handling 'compatibility with commercially available tools' is to ensure that the hosting infrastructure uses platform neutral Application Programming Interfaces (APIs).

Cloud computing services are considered to be relatively cheap, such that providing services at low prices ceases to be a competitive advantage (Durkee, 2010; Hobfeld et al., 2012). This means that cloud computing providers have to find other ways to differentiate themselves from their competitors. It is for this reason that many cloud service providers are looking to enhance the quality of service (QoS), and the quality of experience (QoE).

Quality of service is the ability of a service or an application to "satisfy the stated or implied needs of the user" (ITU-T, 2008a, p.3). QoE refers to "the overall acceptability of an application or service, as perceived subjectively by the end-user" (ITU-T, 2008b, p.2). QoS and QoE concepts complement each other, in that QoS deals on the technological aspects of an application or service, and QoE concentrates on the user satisfaction with the technological aspects of the particular application or service (Hobfeld et al., 2012).

Hobfeld et al. (2012) claim that quality of experience (QoE) has the potential to become the guiding paradigm for managing quality in the cloud. True to this claim, research on QoE and QoS in the cloud is growing. The following are some studies that have been conducted.

Hobfeld et al. (2012) discuss the challenges of QoE management for cloud applications. They evaluate the challenges based on the features of the cloud applications which are determined by the degree of multimedia intensity, interactivity, primary usage domain and service complexity. Cloud gaming, high-definition (HD) teleconferencing, and customer relationship management (CRM) services (e.g. Salesforce.com) require varying degrees of interactivity, which demand QoE requirements, for example, high degree of responsiveness.

Kafetzakis et al. (2012) developed a QoE-driven framework for effectively managing the quality of cloud computing environments. Their framework considers four aspects of quality. (1) System/hardware-related QoS aspects (IaaS management); (2) Network QoS aspects (PaaS QoS optimization); (3) QoE aspects (SaaS QoE optimization); and (4) Business-related aspects (Cloud provider operational excellence). The framework emphasizes on the QoE aspects in order to ensure that the cloud provider complies with the end user QoS expectations. By doing so, the cloud service provider guarantees that the end user's QoE is optimal to keep the end user happy with the service.

The Cloud2Bubble framework takes into account the content generated by users within a specific environment and then uses this information to evaluate the user's QoE (Costa et al., 2012). The aggregated information is used to improve the performance of the application and to improve the overall QoE when delivering personalized services to the users.

Other studies include, for example, a framework for measuring the quality of cloud services aimed at encouraging cloud services providers to better fulfill users' QoS requirements (Garg et al., 2013); a model that enables users to evaluate cloud service providers based on the QoE that service providers are able to provide (Qian et al., 2011); and an adaptive QoS framework that allows video-on-demand (VoD) service providers to target user-specific QoS requirements under different computing capacities and resources (Wang et al., 2010).

2.5 Cloud gaming

In support of the idea of 'adopting cloud computing for various needs within an organization', the research also includes a study on cloud gaming. The cloud gaming study serves to demonstrate the applicability of cloud computing in different organizational contexts to improve efficiency.

In recent years, the video game industry has grown into a large industry that is almost thrice the size of the music industry (PricewaterhouseCoopers, 2010). It was predicted that online and wireless games would overtake console/PC games in 2013 and they would be 36 percent larger by 2016 (PricewaterhouseCoopers, 2013).

Video games can be developed and deployed in cloud computing platforms and environments (Maggiorini & Ripamonti, 2011; Ojala & Tyrväinen, 2011; Ross, 2009). The cloud as a delivery model for video games can enable players to have access to a game through different platforms such as social media (Chang, 2010) and mobile phones (Moreno et al., 2012). In fact, cloud gaming is expected to "change the gaming landscape by reaching a huge audience of gamers who cannot afford expensive hardware and still would like to play the latest major new releases; provided however that the costs of high-speed Internet subscriptions will be affordable for this group of gamers; and that cloud gaming can provide them with a simple, user-friendly and latency-free experience" (Gaudiosi, 2013, p.2).

Even though cloud computing can enhance the games' availability and accessibility, one concern has been about its ability to support highly scalable and data-intensive games (Chan, 2010). In his study, Chan observed that an increase in background load and number of users affected the latency and scalability of the cloud-based games. Nevertheless, Ross (2009) described a supercomputer that might be a solution to the latency and scalability problems. The supercomputer handles all the computational details related to the gaming graphics and allows the end user to access the graphical details that can be supported by the available bandwidth (Ross, 2009).

Cloud gaming allows small organizations to compete with the large organizations (Gaudiosi, 2013; Ojala & Tyrväinen, 2011). Small organizations can utilize the cloud to develop and host their games with little upfront costs. Hence, they do not need to worry about the infrastructure supporting the games and instead focus on developing cutting-edge games (Gaudiosi, 2013) and suitable business models (Ojala & Tyrväinen, 2011). According to Ojala and Tyrväinen (2011), a successful business model is one that is continuously assessed as new game products and services come up, to ensure that the business model is beneficial to all players in the value chain (e.g. game licensors, server manufacturers, network operators, end users).

2.6 Summary

This chapter covered the background and scope of this dissertation by describing the concepts and literature related to cloud computing, cloud-based testing and quality aspects in the context of cloud application development. Cloud gaming is also presented and is used to support the idea about adopting cloud computing in different organizational contexts.

Technologies in the IT industry are constantly evolving. SOA and SaaS concepts progressed towards cloud computing whereby users can access services with limited knowledge about the data centers where the services are running from. There has been a variety of opinions and discussions about the adoption of cloud computing to enhance efficiency in organizations. The opinions about cloud computing have been both positive and negative, which is natural for any new trend. While adopting cloud computing has the potential to support many business and IT functions, the risks and complexities cannot be ignored. In this research, cloud computing is evaluated in the context of cloud-based testing. The objective is to develop a better understanding of how cloud-based testing can be applied to support different testing needs. The research also includes a study on cloud gaming to provide more evidence on the applicability of cloud computing in different organizational contexts.

3 Research problem and methodology

This chapter presents the viewpoints of the dissertation, which determine the research problem and the research questions. The chapter also includes a description of the research method, the rationale for applying it in this research, and the research process taken to realize the study.

3.1 The viewpoints of this dissertation

The success of a technology depends on its ability to be "created and propagated through industries" (Ebert, 2008, p.24). The notion that SOA and especially SaaS trends were here to stay influenced the motive to extend this service concept onto software testing and this dissertation began to study about software testing as a service (STaaS). In the course of the study, cloud computing as a delivery model for online services became more apparent, which made it imperative to contextualize the concept of software testing as a service within cloud computing.

Cloud computing is supporting different business operations by availing computing power and virtualization capabilities that would otherwise be impossible or too expensive to attain. Cloud service providers such as Google, Amazon, Microsoft and Salesforce provide cloud environments for developing and deploying applications and services. However, organizations would like to understand the challenges they should expect when migrating business operations to the cloud and how to mitigate these problems. Since there is a wide spectrum of aspects related to the adoption of cloud computing, it is imperative for one to select the area of focus.

This research investigates the applicability of cloud computing in application development and testing. The study seeks to understand (1) how cloud-based resources could support testing activities, as well as the factors that influence adoption of cloud-based testing and (2) how to achieve the quality characteristics that are desirable when developing cloud-based applications. The ISO/IEC 25000 series standards (ISO/IEC, 2005) are used in this dissertation as a starting point in exploring the desired quality characteristics. The research includes a study on cloud gaming to support the concept of applying cloud computing in various contexts.

3.2 The research problem and its shaping

The motivation for the research originated from observing the trends in the software industry; specifically SOA and SaaS (Dubey & Wagle, 2007; Gold et al., 2004; Turner et al., 2003). The goal was to explore and extend the service concept onto software testing. During the preparation of the research, there was very little research about software testing as service, and it became clear that this was an area worthy of further investigation. In the early stages of the research, there was an emergence of literature suggesting cloud computing as the delivery model for online services and this research began to study about the applicability of cloud computing in software testing. Testing relates to many activities of quality assurance and this led to a study about the quality aspects in the context of cloud application development.

The goal of this dissertation was to *increase empirical knowledge of the adoption, use and effects of cloud-based testing*. In particular, the research focused on cloud-based testing and quality aspects in the context of cloud application development.

The following text describes the research questions of this dissertation. They are addressed in more detail in the *Publications I – V.* In *Publication VI*, we discuss the applicability of cloud computing in game development, with the aim of demonstrating that the adoption, usage and effects of cloud computing should be considered within the specific context where it is applied. Table 1 shows a mapping between the research questions and respective publications.

Research Question 1: What are the research problems related to testing in the cloud?
Research Question 2: What are the conditions that affect software testing as a service?

These research questions address the relevant potential aspects associated with cloud-based testing. We answer the questions by identifying the benefits, challenges and requirements for cloud-based testing. In addition, we identify the important issues of testing in the cloud that are in need of further investigation.

Research Question 3: How can cloud-based testing be applied in practice?

Research question 4: What should be considered when adopting cloud-based testing?

These research questions focus on studying the real-life implementations of cloud-based testing. We provide answers to the questions by observing existing cloud-based testing practices and report on how the interviewed participants were applying cloud computing for testing. We also discuss the effects of cloud computing on software testing and develop a cloud-based testing strategy.

Research Question 5: What are the important quality characteristics when performing cloud application development and how can they be realized?

The aim of research question 5 is to identify the quality characteristics that software organizations find to be important when developing and deploying applications in the cloud. The observations are based on experiences by organizations that are utilizing the cloud to develop and/or deploy their software applications

Research Question 6: What are the factors that affect the adoption and use of cloud computing in game development?

Because cloud computing enables organizations to have quick access to infrastructure

Table 1: Research questions covered in each publication

Publication	Objective	RQ 1	RQ 2	RQ 3	RQ 4	RQ 5	RQ 6
I	Identifying the research issues for testing in the cloud	X					
II	Identifying the conditions that affect testing in the cloud	X	X				
III	Understand the effects of cloud computing on software testing		X	X	X		
IV	Study the aspects related adoption and utilization of cloud-based testing		X	X	X		
V	Analyse the quality-related aspects in the context of cloud application development					X	
VI	Analyse the aspects related to cloud gaming						X

at low costs, this seems to be an advantage that is much appreciated by small organizations. The objective of research question 6 was to explore the applicability of cloud computing in game development within small gaming organizations. The aim was to find out the arguments for and against cloud gaming in the context of small gaming organizations, and to provide a balanced and realistic view about the applicability of cloud computing for different business functions.

3.3 Research methods

Software engineering is a multidisciplinary field, spanning different disciplines involving software engineers and complex software systems (Easterbrook et al., 2008). Different methods can be applicable to a research problem. Unless researchers become more informed about the basic purposes of different research methods, they stand at a risk of selecting inappropriate research methods (Easterbrook et al., 2008). The process of selecting a research method for empirical research in software engineering should be carried out carefully so that researchers get the best out of their data. The choice of the research method may be determined by various factors such as: (1) the researcher(s) theoretical stance (2) researchers access to resources e.g. students/professionals as participants and (3) how closely the method relates to the research questions (Easterbrook et al., 2008).

There are different classifications of research methods (Hevner et al., 2004; Järvinen, 2004; March & Smith, 1995). Järvinen (2004) separates mathematical research methods from research methods concerning reality (Figure 3). This is because mathematical methods deal with symbol systems such as formal languages and algebraic units, which do not have direct reference to objects in reality. Research methods concerning reality are divided into two classes based on the research question, i.e., does the question ask about a part of reality or does it relate to utility of an innovation.

Research studies investigating about reality may include methods for theory development (i.e. conceptual-analytical approaches) or empirical research approaches. Empirical research approaches may involve theory-testing methods, for example, laboratory experiments, surveys, field studies, field tests etc., which investigate more about previously built theories, models or frameworks. Alternatively, theory-creating methods (e.g. case studies, ethnography, grounded theory, phenomenography etc.) may be applied where it is possible to collect data and develop new theories, models or frameworks. Considering Järvinen's classification, this dissertation falls under research investigating about reality using empirical methods for developing new theories, models and frameworks.

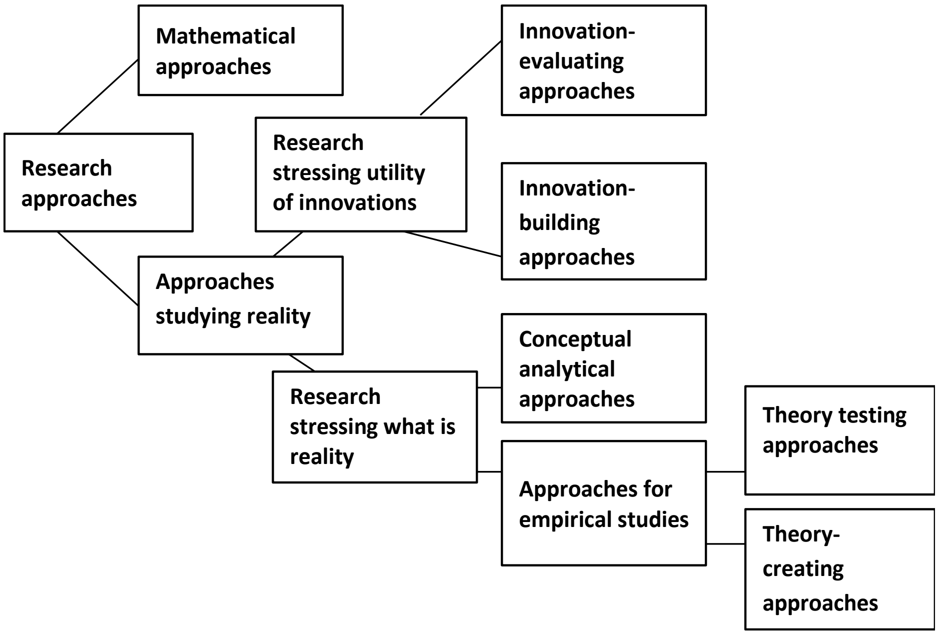

Mathematical approaches
Innovation-evaluating approaches
Research approaches
Research stressing utility of innovations
Innovation-building approaches
Approaches studying reality
Conceptual analytical approaches
Research stressing what is reality
Theory testing approaches
Approaches for empirical studies
Theory-creating approaches

Figure 3: Taxonomy of research methods (Järvinen, 2004)

3.3.1 Research perspectives

This section presents a brief description of the different philosophical perspectives of research. The four commonly known research perspectives are positivism, constructivism, critical theory and pragmatism (Easterbrook et al., 2008).

Positivism: In this perspective, a positivist assumes that the phenomenon is independent of the researcher and the research instruments (Orlikowski & Baroudi, 1991). The knowledge is based on inferences from observable facts that can be verified (Easterbrook et al., 2008). Positivists have a tendency to test or verify hypotheses, a characteristic that fits with carrying out controlled experiments that are commonly associated with the natural sciences (Easterbrook et al., 2008; Orlikowski & Baroudi, 1991). Survey research and case studies also often take the positivist view (Easterbrook et al., 2008), but case studies can also adopt other views.

Critical theory: Critical theorists take the view that research is a "political act, because knowledge empowers different groups within society, or entrenches existing power structures" (Easterbrook et al., 2008, p.291). Hence, critical theorists seek to challenge the norm by selecting research aimed at helping others or drawing attention to things that need to be changed (Easterbrook et al., 2008). In software engineering, the research method most commonly associated with the critical approach is action research (Easterbrook et al., 2008).

Constructivism: In constructivism, also referred to as interpretivism (Klein & Myers, 1999), there is a strong view that the reality cannot be separated from its human context (Easterbrook et al., 2008) such "that people create and associate their own subjective and intersubjective meanings as they interact with the world around them" (Orlikowski & Baroudi, 1991, p.5). Constructivists avoid making generalizations of a population from a sample and focus on understanding how people make sense of the phenomenon under investigation. Therefore, the results obtained from constructivist studies are tightly related to the context of the study. A constructivist perspective leads to a deep understanding of the phenomenon in a sample context, which can then be used to inform other contexts (Orlikowski & Baroudi, 1991).

Pragmatism: Pragmatists employ a view that focuses on how useful the knowledge is for solving practical problems (Easterbrook et al., 2008). The pragmatic perspective assumes a degree of relativism, interpreting knowledge or truth to be relative to the observer. Pragmatism focuses on practical knowledge, with a preference to applying mixed research methods in order to get a balanced view of the phenomenon under investigation.

This dissertation applied grounded theory with the constructivist perspective because (i) the aim was to identify new concepts and not to test or verify existing theories and (ii) the findings from the research are tied to the studied contexts, and could be used to inform other settings (Orlikowski & Baroudi, 1991).

3.3.2 Selection of the research methods

Below is a description of a few research methods that were considered prior to starting this dissertation, followed by the rationale for selecting the grounded theory method.

The Delphi method: The Delphi method uses three main features to gather and refine ideas from a group of experts (Dalkey, 1969):

- *Anonymous response* – A formal list of questions is used to get the opinions of participants. This helps to minimize the influence of dominant individuals.

- *Iteration and controlled feedback* – Several iterations are systematically conducted with controlled feedback between rounds whereby a summary of results from the previous round are communicated to the participants.

- *Statistical group response* – During the final round, the group arrives at a consensus that is determined by the average of the individual opinions. Obtaining a statistically calculated consensus helps to minimize the tendency towards conformity due to group pressure.

The Delphi method requires a panel of experts to participate in rating the issues in order of importance. When beginning this research, the concept of testing in the cloud was very new to the industrial experts. This lack of sufficient know-how of the research topic by the experts was considered a hindrance to successfully applying the Delphi method at the beginning of the study because it might have limited the experts' abilities to make meaningful contribution in the rating process.

Survey research: The aim of survey research is to identify different characteristics such as plans, beliefs, attitudes, and behaviour in data collected from a large population (Fink & Kosecoff, 1985; Pfleeger & Kitchenham, 2001). Survey research often uses structured questionnaires for data collection. When conducting survey research, it is very important that the objectives of the research and the research questions or hypotheses are clear before beginning the empirical investigation. This research sought to explore the dynamics related to cloud-based testing. Since a clear hypothesis did not exist, survey research was not a practical option as a research method.

Case studies: The aim of case study research is to understand the dynamics happening within natural contexts (Eisenhardt, 1989). Case study research can involve a single case or multiple cases. The research questions used during case studies provide strong guidelines for selecting the cases and type of data to collect. A priori specification of constructs may also accompany the research question(s) but if the intention of the case study is to build theory, constructs need not be defined (Eisenhardt, 1989). During this research, the data was collected mainly through interviews with industrial practitioners from the participating organizations. *Publication V* and *VI* report on multiple case studies consisting of five and seven cases respectively.

Grounded theory: Grounded theory is a form of qualitative research method that investigates the reality about the topic being studied. The aim of grounded theory is to develop a theory derived from data that is systematically gathered and analyzed (Strauss & Corbin, 1990). A detailed description of the grounded theory method is provided in Section 3.4 of this chapter.

Although case studies and grounded theory are similarly classified as theory-building approaches, they have some fundamental differences such as:

i. Case studies take into account the context of the subject under investigation, hence providing little basis for generalisation. On the other hand, grounded theory investigates the phenomena at a great depth, such that the results might be suitable for other situations related to studied context (Jones et al., 2005).

ii. In case studies, the unit of analysis is usually described, while grounded theory provides a generalised explanation of the social process being investigated (Jones et al., 2005).

iii. When conducting case studies, it is possible to use existing theory or concepts (i.e. a priori constructs) to guide the research. In grounded theory, the theoretical concepts should be "grounded in and emerge from first-hand data" (Meyer, 2001, p.331).

Grounded theory was selected as the method for this research for the following reasons:

(a) Qualitative studies are useful when you cannot rely on your own previous experience and the literature to guide you in designing closed questions or when you want detailed information in the respondents' own words (Fink, 2003). As a qualitative method, grounded theory is useful in exploring substantive areas with little existing knowledge or areas that are widely known to gain novel understandings (Strauss & Corbin, 1990). Because there was limited research on cloud-based testing, grounded theory was selected in order to take advantage of its ability in enabling the identification of new theories and concepts (Seaman, 1999).

(b) Grounded theory focuses on generating theory through a process that provides depth, meaning and uniqueness to the question of interest (Fink, 2003). This allows the research to focus on the emerging concepts through a systematic analysis of the evidence from the collected data (Strauss & Corbin, 1990).

(c) There was a history of reliable support and successful studies using grounded theory within the research unit, e.g., (Kasurinen et al., 2010; Smolander, 2002; Taipale & Smolander, 2006).

(d) Using qualitative research may also be due to preference and/or experience of the researchers. Strauss & Corbin (1990) point out that some people are more oriented and temperamentally suited to doing this type of work. While working on her master's thesis, the dissertation candidate conducted case study research that she found to be interesting and exciting. This experience helped to build her interest in carrying out qualitative research.

3.4 Grounded theory

Grounded theory was first established in 1967 by Glaser and Strauss (Glaser & Strauss, 1967). The foundation of grounded theory is based on the concept of constant comparison, which means that the emergent theory is compared with the reality reflected in the data throughout the data collection and analysis phases. Later on, Glaser and Strauss published different versions of the grounded theory, resulting in two strands commonly termed as Glaserian (Glaser, 1992) and Straussian (Strauss & Corbin, 1990), developed in conjunction with Corbin. The main differences between the Glaserian and Straussian strands of grounded theory are found in the coding techniques, families and paradigms (Urquhart, 2001). Glaser criticised the Straussian version as being methodical such that it forced a theory out of the data instead of allowing the theory to "emerge" from the data. Strauss and Corbin oppose this claim, insisting that their version does not *"force"* the data; instead it allows the data to *"speak".* While Strauss and Corbin appreciated some of Glaser's criticisms and incorporated them into their second edition (Strauss & Corbin, 1998), they continue to disagree on other fundamentals. For example, Glaser says that the research question is identified during the coding phase, while Strauss and Corbin argue that a research question should be set in advance, mainly so that the study can keep a focus on the area under investigation. Although recognising Glaser's version, and hence keeping an "exploring" attitude, this research applied the Strauss and Corbin approach mainly because it allows the formulation of the research problem before commencing a grounded theory study. Hence, this helped to carry out the research with an "open mind" rather than "an empty head" (Dey, 1999, p.63).

Strauss and Corbin (1990) say that there is a reality that cannot actually be known, but is always interpreted. Thus, their version of grounded theory follows a coding paradigm to generate a theory from the analyzed data. The derived theory is believed to represent the reality as compared to theory obtained by merely combining ideas from one's experiences or suppositions.

Grounded theory has three main coding procedures that are used to analyze the data. These are: **open coding,** where concepts are classified according to their attributes and features; **axial coding,** where the identified attributes and features are used to establish relationships amongst concepts and **selective coding,** where the concepts are combined to build the theory (Strauss & Corbin, 1998).

Open coding: As implied by the name, the aim of open coding is to reveal the underlying meanings, ideas and thoughts within the concepts. The data is critically examined to point out the similarities and differences to uncover abstract concepts. The concepts are further grouped into categories based on similar properties and dimensions. It is important to find categories because they help in narrowing down the number of sections to focus on while at the same time predicting and explaining the emerging phenomena (Strauss & Corbin, 1998). Categories should be identified as early enough as possible because this makes it easier to start identifying unique properties and attributes that help to further develop subcategories.

Axial coding: This is the process of finding the underlying meaning within the categories. Axial coding is done mainly by "relating categories to subcategories along the lines of their properties and dimensions" (Strauss & Corbin, 1998). This procedure involves several basic tasks: (1) laying out the properties of a category and their dimensions, a task that begins during open coding, (2) identifying the variety of conditions, actions/interactions, and consequences associated with a phenomenon, (3) relating a category to its subcategories through statements denoting how they are related to each other, and (4) looking for clues in the data that denote how major categories might relate to each other (Strauss & Corbin, 1998). The researcher takes a deeper look at the categories, asking questions such as why, how, where, when, how come, with what results – leading to the discovery of linkages between categories.

One important thing to remember is that open and axial coding need not be carried out one after the other. On the contrary, they occur in parallel because there is always the potential of new concepts coming up while developing the relationships between the existing ones. Axial coding undoubtedly helps to further develop the categories and subcategories. However, when new information does not seem to add any meaning to the situation, this is referred to as "theoretical saturation" and it implies that there is no need to continue further category development (Strauss & Corbin, 1998).

Selective coding is the process of integrating the categories to find and clearly define the emergent theory. The theory is usually in the form of a core category that potentially explains what the research is all about. All other major categories should be related to the core category. Through previous analysis stages, it may be possible to see the main theme of the research emerging from the data. However, the grounded theory process is iterative which means that the emergent theory evolves as new insights become evident during the analysis. The theory can be explained in the form of theoretical models describing the structure in which the categories form the theory. Gasson (2004) describes two types of theoretical models i.e. a process model and a factor model. The process model contains stages of actions such that the core category reflects the stages with sub-categories and properties or states which show that the process has come to an end (Gasson, 2004). The factor model focuses on cause and effect relationships so that the core categories reflect "antecedent conditions, influences on and consequences of the construct being explored; and the relationships may indicate causality, association, process-sequence, or any pattern that the researcher finds useful" (Gasson 2004, p. 95).

Documenting the analysis process is helpful during data analysis. Strauss and Corbin (1990) strongly recommend the use of memos and conceptual diagrams during theory development. Memos can be written in a free-style manner, as the researcher wishes. This helps the researcher to creatively develop the ideas without feeling constrained to write in a polished fashion (Urquhart, 2001). The use of diagrams provides meaningful ways to view concepts from a bigger picture, hence enhancing the thought-pattern to derive and make conclusions.

As the research progresses, the researcher should aim at gathering data that best aids in further development of the evolving theory (Strauss & Corbin, 1998). In other words, it means that theoretical sampling should be used in order to achieve meaningful results. Theoretical sampling directs the researcher to focus on concepts that have proven theoretical relevance to the evolving theory (Järvinen, 2004). Generally, sampling should stop when theoretical saturation is reached, that is, until (1) no new or relevant data seem to emerge regarding a category; (2) the category is fully developed with regard to all its properties and dimensions exhibiting variation; (3) the relationships among categories are well established and validated (Strauss & Corbin, 1998).

How to evaluate a grounded theory study

The grounded theory research process poses some challenges related to the structure of the research process, the level of detail and how to portray the data in order to demonstrate evidence of the theory (Coleman & O'Connor, 2008). Just as with other qualitative research methods, grounded theory is dependent on the researcher, the purpose of the research, and the research method used (Corbin & Strauss, 2008). This makes it difficult to evaluate the quality of a grounded theory study. Corbin & Strauss (2008) consolidated a list of ten general criteria that can be used to evaluate a grounded theory study, as shown in Table 2.

3.5 Research process

The research process was divided into four phases (Figure 4). In Phase 1, we conducted two studies (*Publication I* and *Publication II*) that explored how industrial practitioners understood and perceived cloud-based testing. We wanted to know whether testing in the cloud was practical and if so, the requirements that would make it feasible. The studies identified the conditions affecting the adoption of testing in the cloud, and pointed out the existing research gaps. The results gathered and reported in this phase provided insights on the factors to evaluate when considering testing in the cloud, such as the requirements, benefits and challenges of testing in the

Table 2: Criteria for evaluating a grounded theory study

Criteria	**Description**
Fit	The theory should resonate or fit with the experience of both the professionals for whom the research was intended and the participants of the study. Participants need to see themselves in the theory even if not every detail applies to them.
Applicability	The findings should be useful and provide new insights. They can be used to develop policies, change practice and add knowledge.
Concepts	The findings should have substance so that they make sense to the professionals in the study area.
Contextualization of concepts	The concepts must be explained within the context that the reader can understand. The reader should not feel that something is missing from the story.
Logic	The logical flow of ideas leading to the findings should make sense to the reader.
Depth	The concepts should have depth of substance and findings should have the potential to make a difference in practice and policy.
Variation	Findings should contain different examples of cases to capture the complexity of real life.
Creativity	Findings are presented in a creative, consistent, flexible and innovative manner.
Sensitivity	The researcher should demonstrate his/her sensitivity to the participants and data by putting away bias and being objective.
Evidence of memos	Memos show the insights, questions, and depth of the research process.

cloud. In addition, we elicited the significant aspects that the organizations wanted to understand in order to carry out testing in the cloud successfully. The results from *Publication I* and *Publication II* led to further investigation in the second phase.

In Phase 2, we conducted two studies (*Publication III* and *Publication IV*) which extend the results obtained in Phase 1. The goal of the Phase 2 was to analyse the impacts of cloud-based testing when applied in real-life practice. The results in Phase 2 affirmed, and hence validated some of the views obtained in Phase 1. In the third phase, we conducted one study (*Publication V*) to complement the earlier studies. The study focused on the quality aspects in the context of cloud application development.

The dissertation candidate had the opportunity to participate in the SOCES research project (http://www2.it.lut.fi/project/SOCES/). The SOCES research project works with gaming organizations with the aim of improving their game development processes.

With many organizations migrating some of their business applications to the cloud, we wanted to find out how gaming organizations were approaching the so-called cloud computing 'hype' (Buyya et al., 2009) for their gaming applications. The objective was to understand the relevance and utilization of cloud computing within small and medium sized gaming organizations (*Publication VI*). In Phase 4, the dissertation includes the study on cloud gaming to demonstrate the relevance of cloud computing in different business contexts. In all the phases, the applied research method was qualitative data analysis using the grounded theory method.

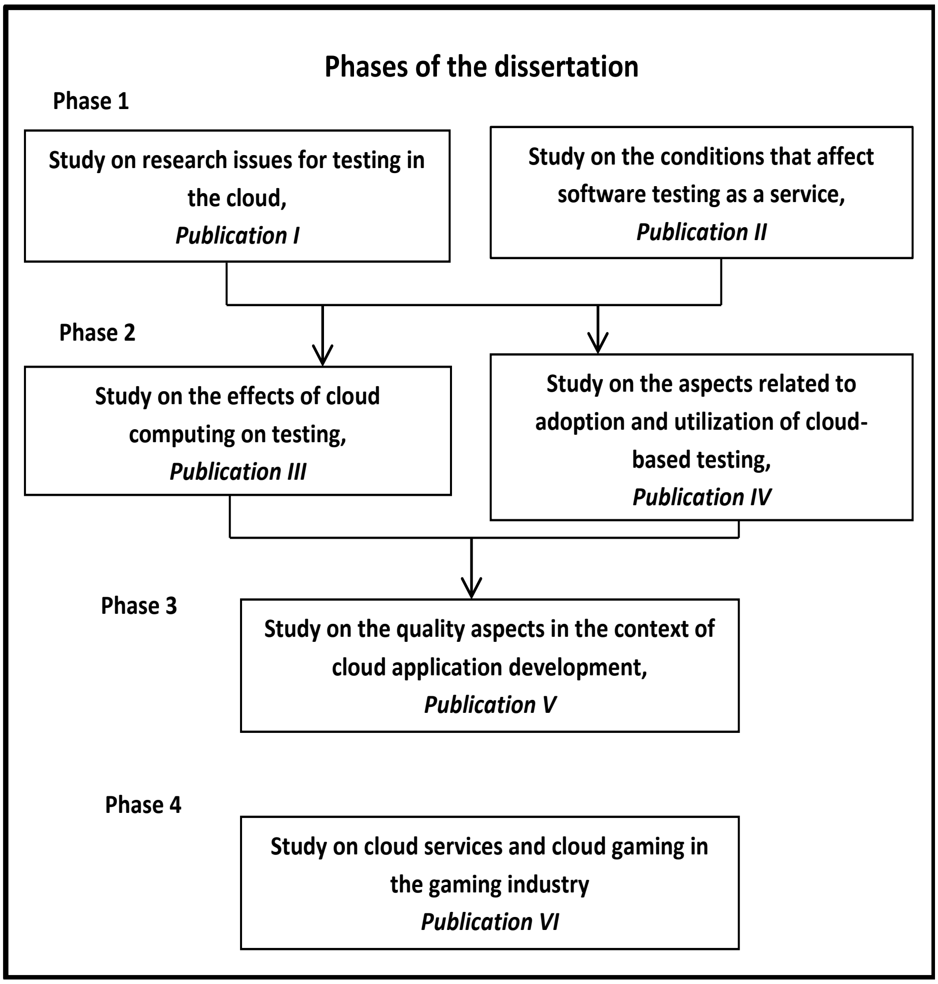

Phases of the dissertation
Phase 1
Study on research issues for testing in the cloud, Publication I
Study on the conditions that affect software testing as a service, Publication II
Phase 2
Study on the effects of cloud computing on testing, Publication III
Study on the aspects related to adoption and utilization of cloud-based testing, Publication IV
Phase 3
Study on the quality aspects in the context of cloud application development, Publication V
Phase 4
Study on cloud services and cloud gaming in the gaming industry Publication VI

Figure 4: Research process and phases

3.5.1 Data collection

The data related to cloud-based testing was collected using theme-based interviews during four interview rounds (Table 3). Interview rounds 2, 3 and 4 were held in cooperation with other dissertation studies, therefore some of the interviews contained questions outside the scope of this research. We conducted forty interviews with respondents from 23 organizations (Table 4). The interview questions were prepared in cooperation with other members of the research team, whereby some research members generated the questions, and other members reviewed and gave feedback. Involving different researchers in preparing the questions ensured sufficient coverage of the topics of interest. During the first interview round, an industrial practitioner from a testing organization also reviewed the questions. The dissertation candidate conducted a test interview with a practitioner from another software organization so as to (1) further ensure the validity of the questions (2) ensure that the interview was within the one hour limit and (3) familiarize herself with carrying out face-to-face interviews.

In the first and second interview rounds, we interviewed development and testing managers, as well as other people in leading positions. Respondents with managerial or leading roles were interviewed because they are responsible for directing the adoption of appropriate tools, methods, and concepts into the organizations. During the third round, we were mainly interested in interviewing employees with knowledge and responsibilities regarding new technologies in testing, and so we chose to interview the managers as well as the developers and testers. In the fourth interview round, we wanted to interview respondents with programming and testing

Table 3: Interview rounds and themes

Round	Themes
1	Software Testing as a Service (STaaS): Requirements, applicable products and services, effect on process model, suitable verification and validation types
2	Test policies, strategies and plans, testing work, software architecture, delivery models, new software development concepts
3	ISO/IEC 29119 software testing standard: Relevance and problems, cloud computing, software development and testing in the cloud
4	Software development, quality, testing, change management

Table 4: Organizations, interview rounds and interviewees

Org.	Business	Interview round(s)	Interviewee role
1	IT, R&D, and consulting services	1st, 2nd, 3rd, 4th	Performance testing unit leader Functional testing unit leader 1 Functional testing unit leader 2 Testing manager 1 Testing manager 2
2	Development of accounting software	1st, 2nd, 3rd	Software manager
3	Testing services and consultancy	1st, 2nd, 3rd	Testing and methodologies director
4	Development of the systems for work time data collection	1st, 2nd, 3rd	Project manager/test engineer
5	Development of the software for the energy market	1st, 2nd	Chief technology officer Mainline testing unit leader Tester
6	Quality assurance services	1st, 2nd	Chief executive officer
7	Testing and quality services	1st, 3rd	Vice president
8	Customized automation provider	2nd, 3rd	Head of software development and hardware design
9	Web-based products and services	2nd, 3rd	Head of software development
10	Building and construction software	2nd, 3rd	Manager - Product development and product packages Unit manager – Documentation, testing and release
11	Naval software systems development	2nd, 3rd	Manager - Software development and production
12	Consulting, teaching and cloud service brokerage services	3rd, 4th	Owner
13	Service development in banking	1st	Program manager
14	Information, logistics and mail communication	1st	Quality and processes manager
15	Testing services	1st	Testing manager Quality adviser
16	Software development consulting services	3rd	Chief technology officer
17	Cloud computing and service management	3rd	Founder and partner
18	Cloud services provider	3rd	System architect
19	IT services provider	3rd	Service development manager
20	Non-profit public IT agency	3rd	Manager – Computing environments group Manager – Software engineering group
21	Web-based services	4th	Research and Development manager Quality assurance manager
22	Cloud computing startup	4th	CEO Tester
23	IT services provider	4th	CTO

responsibilities within the organizations. We therefore requested the organizations to suggest the suitable candidates. Most of the recommended interviewees held managerial responsibilities, but they were also involved in the development and testing activities.

Theoretical sampling was applied in identifying organizations. Theoretical sampling is a process that allows the emerging theoretical understanding to influence the themes and sources for subsequent interviews (Strauss & Corbin, 1990). Theoretical sampling is particularly significant when investigating new areas because it helps to direct the researcher in choosing the data sources that add value to the theory formation (Strauss & Corbin, 1990). In our case, theoretical sampling enabled us to gather the views from industrial practitioners representing organizations that had longer experiences with cloud computing and cloud-based testing. We also applied snowball sampling (interview round 1 and 3) whereby an interviewee recommended a representative from another organization as a suitable respondent.

Five organizations that were contacted during the first interview round declined to give interviews. This was mainly because they did not think that cloud-based testing was suitable for their existing testing practices. Later on, three of these organizations participated in the second interview round. There seemed to be a growing interest in cloud computing. In particular, one of these organizations was particularly interested in topics related to cloud computing and SaaS. This was because some of the organization's customers had started demanding for SaaS-based services from the organization. Hence, the organization was undertaking some considerations in planning how to meet the customers' demands.

All except two interviews were face-to-face, conducted at the respondents' work locations. One of the exceptions was during the second interview round – through email – because the respondent was widely on the move and it was difficult to find a suitable time for a face-to-face interview. The other was during the fourth interview round – using Google Docs – because the interviewer and interviewees were located in different countries. Google Docs was chosen so that two persons from the organization would have access to the same document and answer the questions. Later on, the dissertation candidate accessed the document and downloaded it for archiving and analysis.

All face-to-face interviews were tape recorded and later transcribed for analysis. The transcribed text generated a sum of 518 standard A4 pages (Times New Romans Font, Size 12).

The data in Publication VI was collected from small and medium sized gaming organizations. Table 5 lists the interviewed organizations. The interviews were also theme-based. The themes included questions related to the development, testing and quality processes within the organizations as well questions concerning outsourcing, knowledge management and process development. Four interview rounds were conducted, all including the aforementioned themes. Each interview round involved respondents with similar roles and responsibilities i.e. project managers, programmers, upper management and game designers. The language of the interviews was mainly Finnish, in order to encourage the participants to discuss their views in their native language. The interviews were face-to-face, conducted at the respondents' work locations and were tape-recorded for transcription and analysis.

3.5.2 Data analysis

Qualitative data can amount to large volumes, which can be overwhelming for a researcher. Qualitative analysis software is particularly helpful in making sense of and managing the large amount of data. In our analysis, we used the qualitative analysis tool Atlas.ti (ATLAS.ti, 2013) which provides a compact way of handling the source files, with tools for linking, searching and sorting the data.

Applying the grounded theory method, the analysis in the Phase 2 progressed iteratively from the analysis started during Phase 1. During the analysis in Phase 1, it was clear that testing in the context of cloud computing was unclear, and it needed to be understood better. In addition, the research issues suggested various aspects that required further investigation, such as understanding how testing in the cloud affects the business, the business areas suitable for testing in the cloud, and how to verify the quality of applications that have been tested in the cloud, to mention but a few. Hence, in Phase 2, we sought to investigate the leads obtained from Phase 1.

Table 5: Organizations in *Publication VI*

	Product platform	**Company size**[1]
Case A	PC, Game consoles	Medium
Case B	Mobile	Small
Case C	Game consoles, PC	Medium
Case D	Mobile, PC	Small
Case E	Mobile	Small
Case F	PC	Medium
Case G	Web-based	Small

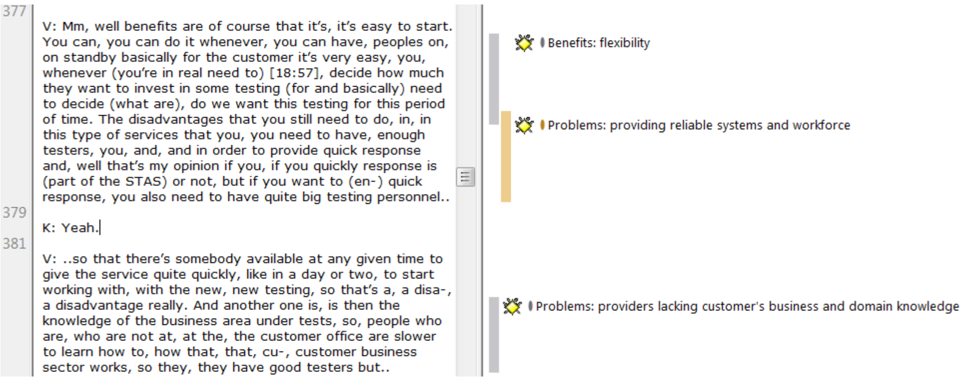
[1] Amount of people directly employed by the organization (European Commission, 2005)

Open coding mainly involves the researcher going through the data to identify the main points that the interviewee is talking about. Figure 5 shows an example of open coding. Having identified leads from Phase 1, the open coding continued in Phase 2 focused on the obtained leads. The leads dealt with the effects of cloud computing on software testing, the aspects related to the adoption of testing in the cloud, and the quality aspects related to testing in the cloud.

The goal of axial coding was to identify the relationships between the identified categories. The identified relationships were based on similarities, associations and differences between the categories. During axial coding, the analysis focused on the phenomena under investigation with a progression towards an emerging theory, usually in the form of a core category.

Selective coding is the process of identifying, refining, and explaining the core category (Strauss & Corbin, 1990). The core category might be one of the identified categories, or new one that may not have existed earlier. Other times, there might not be a single category that is broad enough to cover the entire phenomenon. In such a case, Strauss and Corbin (1990) advice that a conceptual idea covering all the categories should be developed. Figure 6 shows an example of a network diagram that was drawn in the process of axial and selective coding (Created on 4th January, 2012).

The leads we obtained from Phase 1 influenced the identified categories in the studies performed during Phase 2 (*Publication III* and *Publication IV)*. In *Publication III,* the core category was named *Effect of cloud computing on software testing.* The next view dealt with the aspects related to adoption of cloud-based testing. In *Publication IV,* we identified four main categories: *testing resources, utilisation of cloud-based testing resources, quality of cloud-based testing environments and quality of the system under test.* Due to the prevalent need for testing resources, the category *testing resources* initially appeared to be a strong candidate as a core category. However, during the process of



377
V: Mm, well benefits are of course that it's, it's easy to start. You can, you can do it whenever, you can have, peoples on, on standby basically for the customer it's very easy, you, whenever (you're in real need to) [18:57], decide how much they want to invest in some testing (for and basically) need to decide (what are), do we want this testing for this period of time. The disadvantages that you still need to do, in, in this type of services that you, you need to have, enough testers, you, and, and in order to provide quick response and, well that's my opinion if you, if you quickly response is (part of the STAS) or not, but if you want to (en-) quick response, you also need to have quite big testing personnel..

379
K: Yeah.

381
V: ..so that there's somebody available at any given time to give the service quite quickly, like in a day or two, to start working with, with the new, new testing, so that's a, a disa-, a disadvantage really. And another one is, is then the knowledge of the business area under tests, so, people who are, who are not at, at the, the customer office are slower to learn how to, how that, that, cu-, customer business sector works, so they, they have good testers but..

Benefits: flexibility
Problems: providing reliable systems and workforce
Problems: providers lacking customer's business and domain knowledge

Figure 5: Example of open coding in ATLAS.ti

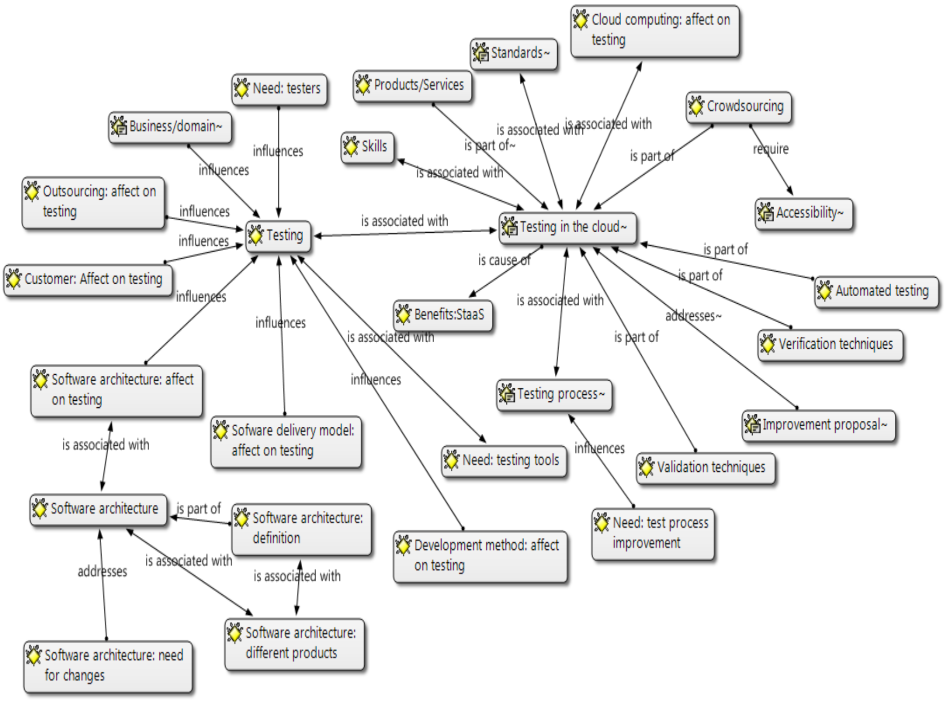


Figure 6: Example of a network diagram in ATLAS.ti

crystallising the theory, it emerged that the need for testing resources represented “antecedent conditions” occurring before *utilisation of cloud-based testing resources* – which eventually emerged as the core category. To identify the relationships between the categories, we applied the principle of ‘identifying the variety of conditions, actions/interactions, and consequences associated with a phenomenon (Strauss & Corbin, 1998). Figure 7 shows the identified relationships between the categories.

In phase three, we approached the lead about quality aspects related to testing in the cloud with a fresh perspective. It emerged from the data that the respondents discussed the quality aspects in the context of developing and hosting applications in the cloud, and not specifically focusing on testing. Since development and testing are interconnected phases of a software development life cycle, we approached the problem by first understanding the quality characteristics that the respondents discussed (*Publication V*). In the analysis, we derived the categories that affected how the important quality characteristics were targeted and achieved. These are *life-cycle model and tools, software application, important quality attributes, practices for handling requirements, and customer involvement.* We then developed a conceptual idea to cover all these categories and named it as *Activities that aid in attaining desired quality characteristics of cloud-based applications.*

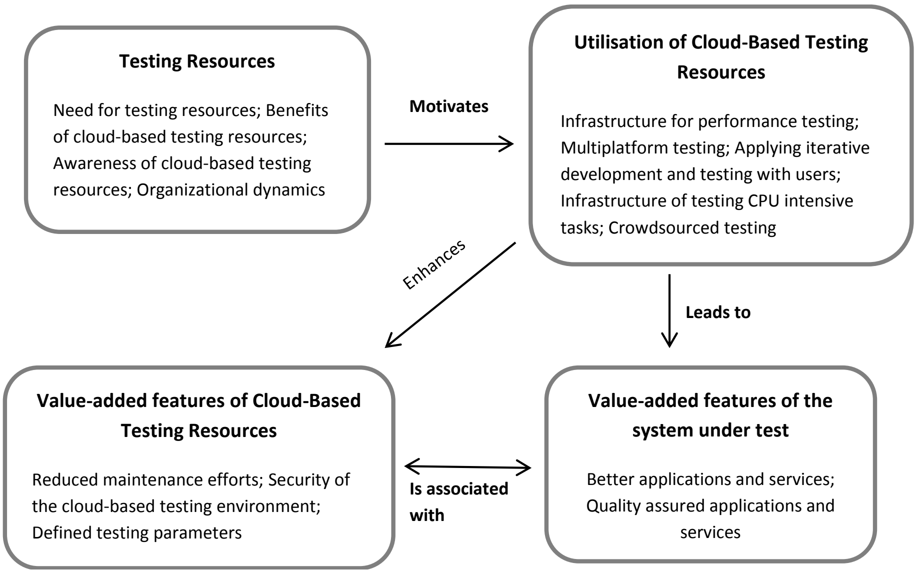
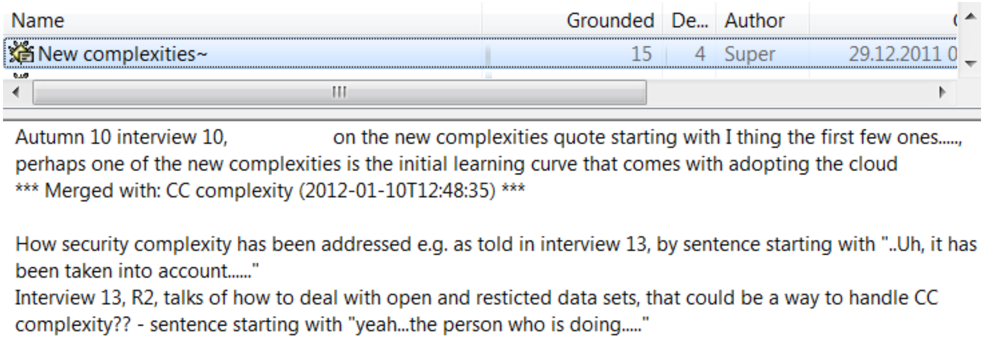

Testing Resources
Need for testing resources; Benefits of cloud-based testing resources; Awareness of cloud-based testing resources; Organizational dynamics
Motivates
Utilisation of Cloud-Based Testing Resources
Infrastructure for performance testing; Multiplatform testing; Applying iterative development and testing with users; Infrastructure of testing CPU intensive tasks; Crowdsourced testing
Enhances
Leads to
Value-added features of Cloud-Based Testing Resources
Reduced maintenance efforts; Security of the cloud-based testing environment; Defined testing parameters
Is associated with
Value-added features of the system under test
Better applications and services; Quality assured applications and services

Figure 7: Relationships between categories related to utilisation of cloud-based testing resources

The data analysis for *Publication VI* also applied the grounded theory method. Due to language limitations, another researcher handled the transcribed data and translated the data related to this study into English. He analysed the data, summarized the categories and observations onto a Microsoft Excel spreadsheet, and shared them with the dissertation candidate for discussion and feedback.

Memos were also written during the analysis (Strauss & Corbin, 1990; Urquhart, 2001). Memos are usually free-flowing ideas and thoughts regarding a code, concept or category (Hoda et al., 2012). In Atlas.ti, most of the memos were in form of comments attached to various codes, for example, as shown in Figure 8. The memos were also in form of text, written using Microsoft word. For example, during the analysis for *Publication IV,* we wrote this text as our initial impression about one of the organizations. "They are developing a cloud application. The team is using their previous experiences in deciding the development approach and tools. The development was taking place at the time of the interview. Changes [in requirements] would be addressed along the way. Quality requirements are adjusted according to the performance of the end product/service and the user preferences. Quality section in the interview has interesting views."



Name	Grounded	De...	Author	(
New complexities~	15	4	Super	29.12.2011 0

Autumn 10 interview 10,   on the new complexities quote starting with I thing the first few ones....., perhaps one of the new complexities is the initial learning curve that comes with adopting the cloud
*** Merged with: CC complexity (2012-01-10T12:48:35) ***

How security complexity has been addressed e.g. as told in interview 13, by sentence starting with "..Uh, it has been taken into account......"
Interview 13, R2, talks of how to deal with open and resticted data sets, that could be a way to handle CC complexity?? - sentence starting with "yeah...the person who is doing....."

Figure 8: Example of a comment attached to a code, serving as a memo

3.5.3 Finishing and reporting the dissertation

In all the Phases, the reporting was done by writing scientific publications that are included in this dissertation. The goal of Phase 1 was to establish a general understanding about testing in the cloud and to identify the potentially interesting research issues. *Publication I* discusses the research issues in detail, and *Publication II* reports about the conditions that affect testing in the cloud.

The study continued in Phase 2 with a deeper analysis of testing in the cloud. *Publications III* discusses the effects of cloud computing on software testing and proposes a roadmap towards cloud-based testing. *Publication IV* evaluates the aspects related to the adoption and utilisation of testing in the cloud, and presents a cloud-based testing strategy that may help organizations as they perform some of the testing activities in the cloud. In Phase 3, *Publication V* presents an analysis of the activities that organizations can undertake in order to target important quality characteristics of their cloud-based applications.

Phase 4 contains a study on cloud gaming. The aim of phase 4 is to show an example of the utilization of cloud computing in a different business context.

3.6 Summary

This chapter described the research problem, the research method and the research process used to accomplish this dissertation work. Table 6 summarizes the research phases that made up the research process.

Table 6: The summarized research phases

Phase	Phase 1	Phase 2	Phase 3	Phase 4
Research questions	What are the research problems related to testing in the cloud? What are the conditions that affect software testing in the cloud?	How can cloud-based testing be applied in practice? What should be considered when adopting cloud-based testing?	What are the important quality characteristics when performing cloud application development and how can they be realized?	How do game organizations apply cloud services and cloud technology in game development?
Instruments and protocols for data collection	Theme-based interviews	Theme-based interviews	Theme-based interviews	Theme-based interviews
Data analysis	Grounded theory analysis using Atlas.ti	Grounded theory analysis using Atlas.ti	Grounded theory analysis using Atlas.ti	Grounded theory analysis using Atlas.ti, and category tables
Reporting	*Publication I* *Publication II*	*Publication III* *Publication IV*	*Publication V*	*Publication VI*

4 Overview of the publications

This chapter presents an overview of the most important results contained in the publications included in this dissertation. The six publications, attached as an appendix, contain the results in detail. *Publications I, III, V,* and *VI* are published separately in peer-reviewed scientific conferences and journals. *Publication II* is published as a chapter in a peer-reviewed book. *Publication IV* is in the publication process of a journal. This chapter briefly discusses each of the publications, including its research objectives, main results and relation to the whole dissertation.

4.1 Publication I: Research issues for software testing in the cloud.

4.1.1 Research objectives

The objective of this publication was to elicit important research issues in the context of software testing in the cloud. The paper aimed at being a resource for researchers interested in the topic – to provide a starting point on potential topics for further investigation. It served as an initial contribution towards describing cloud-based testing.

4.1.2 Results

At the beginning of this research, we found that the literature on software testing in the cloud was scarce (Riungu et al., 2010). The material available was mainly in the form of industrial white papers and reports. As we collected the data, we continued to search for and read more content on the topic. In an attempt to contribute to a better understanding about testing in the cloud, this paper described a conceptual idea on the facets of testing in the cloud (Figure 9).

We conceptualized cloud-based testing as consisting of three facets (1) the system or application under test is accessible online. This might be SaaS software or non-SaaS software. In addition, this includes testing at different test levels e.g. performance testing; (2) testing infrastructure and platforms are hosted across different deployment models of the cloud i.e. public, community, private or hybrid clouds; (3) testing of the cloud itself. Cloud environments should be tested and measured for their performance, availability, security and scalability in order to support efficient delivery of services (Spirent, 2010).

We classified the research issues into three categories: Application issues, management issues, and legal and financial issues (Table 7). Application issues included aspects related to handling the applications tested in the cloud. Management issues covered the matters dealing with how to handle testing in the cloud with respect to the flow of work and business impacts. Legal and financial issues incorporated aspects dealing with handling the test data, as well as how to attach the monetary value to cloud-based testing services.

4.1.3 Relation to the whole

The results of this study identified issues that needed attention from both research and industrial communities and described a conceptual understanding about how to carry out cloud-based testing. The identified research issues showed that industrial practitioners and researchers needed a better understanding of a broad range of aspects related to successful implementation and delivery of testing in the cloud. Therefore, in an attempt to address the research gaps, other studies in this dissertation

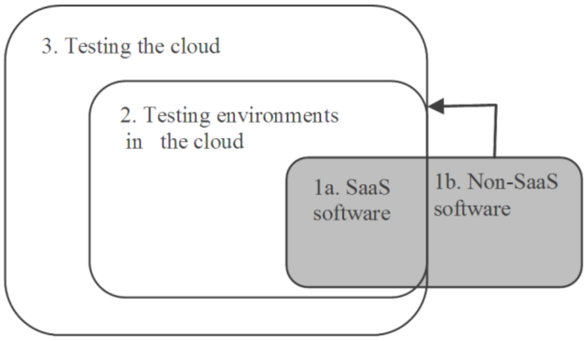


Figure 9: Facets of testing in the cloud

Table 7: Research issues for software testing in the cloud

Application issues
1. Applications suitable for online software testing
2. Providing a ready online performance testing package for any customer
3. Quality checks for applications that have been tested on the Internet
4. Harmonizing test processes across multiple players
5. Online testing solutions for e-business applications How to handle test data. Where does it come from? Who owns it? How is a system under test made accessible to the tester? What if signing of Non-Disclosure Agreements (NDAs) is required?
Management issues
6. How to create a big enough available pool of testers
7. Effects of software testing as an online service on the customer's business. Change management issues during the processes of adopting software testing as an online service.
Legal and financial issues
8. How to handle test data. Where does it come from? Who owns it? How is a system under test made accessible to the tester? What if signing of Non-Disclosure Agreements (NDAs) is required?
9. Pricing models and service descriptions for online software testing services

followed the leads provided by the results, and investigated them further. For example, *Publication III* studies about the effects of cloud computing on the delivery and support of software testing work, contributing to a better understanding of the effects of software testing as an online service on the business, which falls under the management issues. Similarly, *Publication IV* proposes a cloud-based testing strategy, which takes into consideration concerns related to adoption of cloud-based testing e.g. SLAs and test data management (under legal and financial issues).

4.2 Publication II: Software testing as a service: Perceptions from practice

4.2.1 Research objectives

This publication presented a study whose aim was to evaluate the conditions that affect software testing as an online service. The purpose was to investigate what the industry believed to be vital components for carrying out testing in the cloud effectively. The research question for the study was "What are the conditions that affect software testing as an online service?" The study was an explorative qualitative study, with data collected through interviews with industrial practitioners.

4.2.2 Results

During the analysis, we evaluated the conditions in terms of requirements, benefits, and challenges of software testing as service. We also considered the conditions that make it feasible for realizing software testing as a service. In addition, we evaluated the applicability of cloud computing as the delivery model for software testing as a service. Table 8 presents a summary of the evaluated conditions.

Table 8: Conditions that influence software testing as a service

Requirements	
Domain knowledge	Important especially for mission critical systems, is required throughout the development lifecycle, testing included. Difficult for such systems to be tested by external parties.
Infrastructure	Mainly includes cloud computing as a testing environment, and as a hosting platform for testing resources
Security	Data security across networks, confidentiality of customer data.
Pricing	Service level agreements, transparency
Communication	Meetings, video conferences, telephone calls, emails, formal software test management systems
Testers' skills	Testers would need to develop new and/or better skills, such as communication and global project management skills among others.
Benefits	
Reduced costs	No need to invest in testing servers, acquire testing resources as needed and pay only for what you use. Less license fees also help to reduce costs
Flexibility	A customer can obtain testing services only when needed, and pay only for what is used.
Access to global markets	Market base for both provider and customer becomes bigger
Challenges	
Test data management	Who owns data? Where is it stored?
Project and change management	How to manage multiple testing projects across different platforms, different customers, and/or even different providers.
Service level agreements	Customers should be assured of the reliability of the services
Enabling conditions	
Standards	Applications and systems based on standards are easy to test. Testing parameters are predictable due to standards.
Verification and validation methods	Several of these methods can be tested, but the possibility to test them should be thoroughly considered and care should be taken to avoid misunderstandings.
Cloud computing	
Applicability	Enables scaling up and down as per demand and enhances performance. May increase complexity hence increase need for testing.
Data governance	To be considered across various geographical regions
Research issues	
Various	Discussed in *Publication II*

4.2.3 Relation to the whole

This exploratory study revealed the different aspects that affect software testing as a service. At the beginning of this study, it was evident that organizations dealing with mission critical systems did not find testing in the cloud to be a viable option. Such organizations placed a strong emphasis on the need for testers to possess sufficient domain knowledge, and were therefore not willing to let an unknown tester to test their systems. However, as the research progressed, some of these organizations became interested because they were internally following the trends on SaaS and cloud computing.

The organizations participating in this study based their views on their general understanding about software testing and trends such as SOA, SaaS and cloud computing. The results from this study formed a basis for the subsequent studies. To get a deeper insight on testing in the cloud, the subsequent studies targeted organizations that used cloud computing for deploying and developing applications and services.

4.3 Publication III: Testing in the cloud: Exploring the practice

4.3.1 Research objectives

The objective of this publication was to understand the impacts of cloud computing on software testing. To do this, we conducted interviews with respondents from organizations that were using the cloud. The goal was to build a deeper understanding on the results obtained from *Publication II,* by studying cloud-based testing in real-life practice.

4.3.2 Results

The results of this study focused on the real-life practices involving cloud-based testing. The study highlighted several companies providing cloud-based testing services, including Soasta, Zephyr, Skytap, uTest, CloudTestGo, to mention but a few. The study also proposed a practical roadmap that industrial practitioners could use for establishing testing in the cloud.

The analysis resulted in elaborating on the effects of cloud computing on software testing – classified into three categories:

(1) ***The effects of cloud-based testing on actual testing.*** Organizations can obtain cloud-based infrastructure and resources quickly and cost effectively to achieve performance, scalability and stress testing that is more efficient. In addition, cloud-based testing promotes faster development cycles because the overall testing times become shorter. Furthermore, it is possible to replicate the production environments using cloud-based testing environments, which helps to generate more realistic testing results.

(2) ***The effects of cloud-based testing on delivery and support of testing services***: The cloud hosts a wide variety of testing tools and environments that organizations can get on-demand. This provides organizations with the advantage of being able to choose and even experiment with various options for testing purposes. Cloud-based testing encourages more collaborative interaction between developers and testers because they all have simultaneous access to the systems, as opposed to situations where developers handle the system first before the testers. Additionally, due to the quick pace of setting up cloud-based testing resources, this helps organizations to focus on the customer needs more efficiently.

(3) ***Cloud-based testing challenges****:* In addition to the challenges mentioned in *Publication II,* we identified other challenges of cloud-based testing. For example, when dealing with different systems that need to work together, it calls for attention to additional testing parameters. Other challenges are how to handle load balancing, network latency and multitenancy when performing testing in the cloud.

This publication also proposed a roadmap that organizations could follow when evaluating the viability of cloud-based testing. The roadmap included five actions i.e. *understanding cloud computing* – in order to determine whether cloud-based testing is a viable option, *conducting pilot projects* – to explore the potential benefits with less risks, *developing elaborate strategies* – which help organizations in identifying possible approaches for adopting cloud-based testing, *enhancing team interaction and preparing for complexities* – in order to be better prepared for, and adequately address the challenges, and finally, *enhancing cooperation between research and industry* – so as to focus on cloud-based testing aspects that are relevant for the software industry.

4.3.3 Relation to the whole

In *Publication II*, cloud computing is mentioned as the primary means through which software testing can be delivered and acquired as an online service. This study focused on discussing the effects of cloud computing on testing. The study extended and confirmed the 'perceptions' from *Publication II,* concerning the challenges of testing in the cloud. The results of this study contribute towards the "understanding of current cloud testing practices, support tools, software processes for cloud applications, and software tools for cloud applications" (Grundy et al., 2012).

This publication suggests a roadmap that can be a starting point for organizations when evaluating the viability of cloud-based testing. Because this roadmap is basic, we expounded more from it, and developed a cloud-based testing strategy presented in *Publication IV.*

4.4 Publication IV: Adoption and utilisation of cloud-based testing in practice

4.4.1 Research objectives

This study focused on aspects related to the adoption and utilisation of cloud-based testing in different organizational contexts. We wanted to generate a balanced view of cloud-based testing right from pre-adoption to utilization. The goal was to understand how organizations were approaching and utilizing cloud-based testing. Hence, the publication was based on two research questions: (1) What are the motivating factors that cause organizations to adopt cloud-based testing? (2) How is the cloud used to test applications, services and systems?

From the analysed results, this publication also carries on from *Publication III* to develop a cloud-based testing strategy that may assist organizations during decision-making processes towards adopting the cloud for testing purposes.

4.4.2 Results

In this study, we identified four categories related to the adoption and utilization of cloud-based testing in practice, i.e., *testing resources, utilisation of cloud-based testing resources, value-added features of cloud-based testing resources, and value-added features of the system under test.*

Testing resources covered the motivations and reasons that drive organizations to adopt cloud-based testing resources. Organizations adopt cloud-based testing resources for several reasons, for example, the *need for testing resources* is an important motivation for adopting cloud-based testing resources. This is appreciated because accessing these resources from the cloud is quick, on-demand and cost effective. Another motivation for adopting cloud-based testing is the desire to take advantage of the *benefits of cloud-based testing resources*, such as, flexibility and access to global markets. We also observed that organizations were more willing to consider the prospects of cloud-based testing as they became more *aware of cloud-based testing resources.* In addition, small and medium sized enterprises (SMEs) were more open to adopting cloud-based testing in order to take advantage of the cheap costs.

Utilisation of cloud-based testing resources described how cloud-based testing resources are put to use after they have been adopted within the organizations. Cloud-based testing resources can be utilized to perform *performance testing* as well as testing applications across multiple platforms. Cloud-based testing encourages *iterative development and testing with users,* so that an application's developed features can be released and it is tested by the users. Due to their ability to process large computations, cloud-based resources can also be used for *testing CPU intensive tasks*. Another way of utilizing cloud-based testing resources is by organizing a pool of testers to perform testing in what is referred to as *crowdsourced testing*.

Value-added features of cloud-based testing resources described the value propositions created when utilising cloud-based testing resources. These include *reduced maintenance efforts, secure cloud-based testing environments,* and *defined testing parameters.*

Cloud-based testing increases the opportunities for producing applications and services with various value propositions. *Value-added features of the system under test* described the value propositions that cloud-based testing added on the tested applications and services. Cloud-based testing supports iterative development which aids in fostering a continuous improvement approach, hence resulting in *better applications and services*. In addition, since the test environment imitates the real production environment, this helps to better assure the *quality of the tested applications and services.*

Based on the results from this and previous publications, we developed a cloud-based testing strategy (Figure 10). The strategy includes five main activities: *evaluating the line of business*, *assessing the needs* of the organization, *identifying and selecting* the appropriate service delivery specifications e.g. service provider, *utilizing the cloud service* to test, and making the necessary adjustments to improve the cloud-based testing experience. The strategy includes activities that support the process of adopting cloud-based testing, and can be applied according to the needs of an organization.

4.4.3 Relation to the whole

This study described (1) the motivating factors that drive organizations to adopt cloud-based testing, (2) the various ways in which cloud-based resources are utilised for testing, (3) the favourable consequences of utilising cloud-based testing resources, (4) the positive effects of cloud-based testing on the system under test. The results showed that cloud-based testing can be used for various testing needs within different

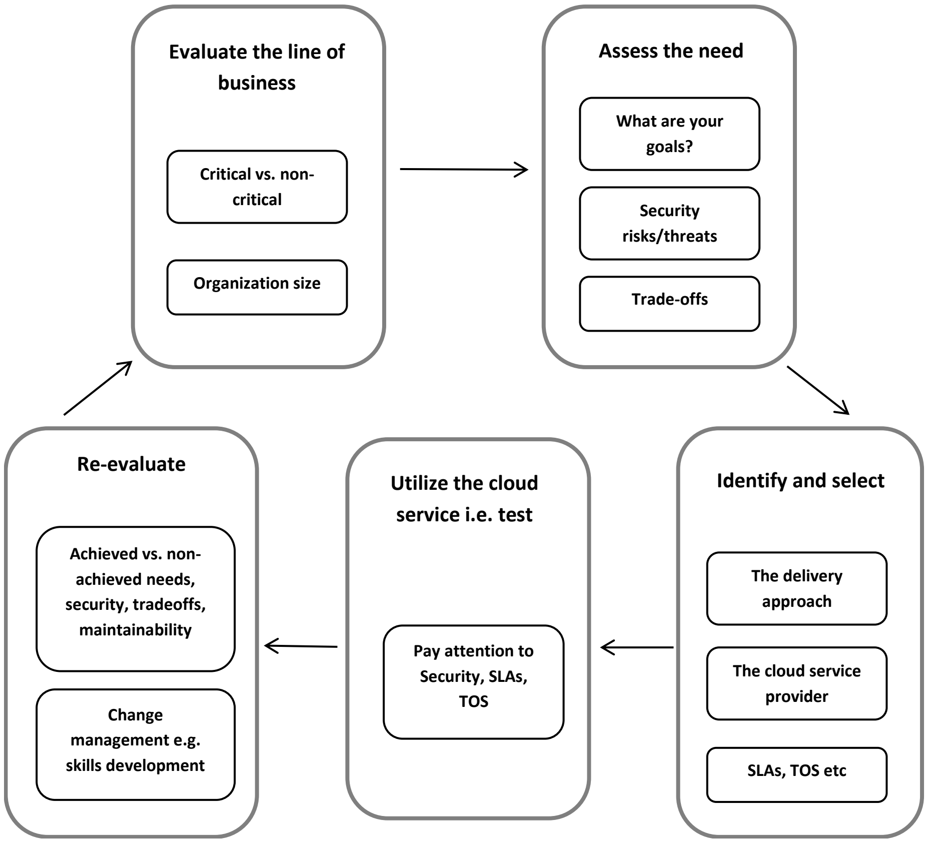

Evaluate the line of business
Critical vs. non-critical
Organization size
Assess the need
What are your goals?
Security risks/threats
Trade-offs
Identify and select
The delivery approach
The cloud service provider
SLAs, TOS etc
Utilize the cloud service i.e. test
Pay attention to Security, SLAs, TOS
Re-evaluate
Achieved vs. non-achieved needs, security, tradeoffs, maintainability
Change management e.g. skills development

Figure 10: Cloud-based testing strategy

organizations. The cloud-based testing strategy presented in this study can be applied when organizations are working towards adopting and applying cloud-based testing.

4.5 Publication V: Desired quality characteristics in cloud application development

4.5.1 Research objectives

When collecting data related for previous publications, the respondents also discussed the quality aspects in the context of developing and hosting applications in the cloud, and not specifically focusing on testing. We took this as a lead, and since development and testing are interconnected phases of a software development life cycle, we approached the problem by seeking to understand the quality issues.

The objective of this qualitative case study was to explore the quality-related aspects in the context of cloud application development. Developing cloud applications entails that the developer uses a number of pre-defined architectural structures, resources, interfaces or service access and discovery mechanisms. This may emphasize certain quality requirements such as the need for strict conformance and high interoperability. Therefore, it is important to know the desired quality and to understand how it reflects on the way cloud applications are developed and vice versa. In other words, the development of cloud applications needs to consider the quality expectations. We used the quality characteristics described in the ISO/IEC 25000 series (ISO/IEC, 2010) to see what quality characteristics were important when developing cloud applications, and what the organizations did in order to achieve the desired quality goals.

4.5.2 Results

This publication mainly focused on the quality characteristics that are important for cloud application development and the activities that aid in achieving these quality characteristics. The results showed that the desired quality varied among the organizations, ranging from availability, functional suitability, reliability, security, operability, usability to performance efficiency. However, usability was important in all the organizations. Despite the differences in desired quality, the organizations involved three activities geared towards attaining the desired quality characteristics. These activities involve the *selection of a suitable life-cycle model*, during which *the customer is engaged* and *the most suitable tools* are used. The organizations incorporated these activities to establish supportive working practices for acquiring the desired quality.

The life-cycle models were such that they allowed the developers to interact with the customers and cloud providers. Interacting with the customers helped in improving requirements, and consequently, the functionality of the software. This was especially useful for enhancing the usability of the developed applications. On the other hand, interacting with the cloud provider helped the developers to align the functionality of their (developers') applications according to the specific platforms within which the applications are developed. The developers used development tools that deemed as most relevant for the software being developed. For example, by selecting development frameworks suitable for building web-based applications, a couple of the organizations were able to focus on developing the important application features.

4.5.3 Relation to the whole

This publication offered a quality perspective when developing cloud-based services and applications. The study described the activities that the organizations incorporated to develop cloud applications. The use of cloud computing platforms for developing and hosting applications necessitate the need for close interaction between the cloud platform providers and the application developers.

Building cloud applications is no doubt a complex process, which requires multidisciplinary techniques for providing traceable links between development activities and the desired quality. The activities that organizations undertake to achieve important quality characteristics can be combined with cloud-based testing in order to produce the targeted quality.

4.6 Publication VI: Cloud services and cloud gaming in game development

4.6.1 Research objectives

The objective of this qualitative study was to understand the relevance and utilization of cloud computing within small and medium sized gaming organizations. To do this, we interviewed respondents from seven organizations developing different types of games such as PC, mobile and web-based games as well as games requiring the use of game consoles.

4.6.2 Results

The results of this study explained the status of cloud services and cloud gaming from the perspective of small and medium sized gaming organizations.

Firstly, we found that small and medium sized gaming organizations knew about cloud services and cloud gaming. In general, the organizations used cloud computing to cut down on costs associated with infrastructure and server-side services. They used cloud-based tools to support document sharing and project management systems as well as cloud-based server and rendering farms. In addition, even though most of the organizations had plans for making use of cloud services and applying cloud gaming in the future, they considered the current cloud computing technology to be unreliable in supporting their main game products except for simple data storage needs and server-side services.

Secondly, the respondents acknowledged the fact that cloud computing had the potential to convert gaming products to services, and hence enabling access to the games by a wider customer base. However, the respondents expressed concerns about the lack of clear business models and success stories that would encourage gaming organizations to adopt cloud computing. The respondents opined that the effects of cloud computing on gaming businesses were not clear because the monetization of the technology (Yamakami, 2012) was not evident from the existing cloud services and cloud gaming implementations. Even so, to make money, one organization that developed web-based games relied on in-app purchases (Tyni et al., 2011) embedded within their free-to-play online games.

Thirdly, even though the organizations developed games on different platforms, and were of different sizes, these factors did not seem to affect the respondents' views towards cloud services and cloud gaming. Respondents from all the organizations knew about the cloud services and cloud gaming models and the organizations had some plans for adopting cloud computing in the future.

Lastly, besides general applicability of the cloud services and cloud gaming models in the game industry, the organizations expressed that cloud computing posed some potential changes in the gaming industry; in that cloud computing would steer gaming products towards services, and user groups towards game-related communities. Consequently, this would make community building, visibility in the social media and community management to be vital activities in the gaming industry. In addition, the focus on user communities would encourage more collaboration between the gaming organizations and their users, such that the users would become more influential in enhancing existing gaming services and developing new ones.

4.6.3 Relation to the whole

This study provides insights on how cloud computing affects small and medium-sized gaming organizations, their products, their access to global markets, and the cost structure. The benefits and hindrances of cloud adoption are also discussed in the context of cloud gaming. Cloud computing could enable organizations to increase efficiency and profits across different business spectrums. As in the previous studies, this study also showed that the adoption of cloud computing could be encouraged by demonstrating successful examples and developing clear business models for conducting successful cloud-based businesses.

This study is included in this dissertation to provide more evidence on the importance of considering the domain of the organization when introducing cloud services, including cloud-based testing. The study revealed that the gaming professionals had concerns about cloud computing, but not so much with regard to security as is popular with organizations in other business and application domains. The gaming organizations were not fully convinced about the current technological possibilities in the cloud for their game products. The unique needs and requirements of gaming products should be put into consideration in order develop successful cloud gaming services and applications. Consequently, cloud-based testing tools, methods and procedures intended for the gaming industry should be tailored to the unique features of the gaming industry.

4.7 About the joint publications

For *Publication I - II,* the dissertation candidate designed and implemented the data collection instruments, collected the interview data, and wrote major parts of publications.

For *Publication III – V,* the dissertation candidate participated in the design and implementation of the data collection instruments, participated in the collection and analysis of the data and wrote major parts of the publications.

For *Publication VI,* the dissertation candidate participated in the design and implementation of the data collection instruments. The dissertation candidate was partially involved in data collection because the language used during the interviews was Finnish. For the data analysis, she was involved in refining the analysed data after the second author translated it into English. The dissertation candidate wrote major parts of the publication.

5 Contributions, implications and limitations

This chapter extracts the results from the different publications and presents them in a conclusive summary, followed by their implications for practitioners and researchers. After that, the dissertation is evaluated and the limitations of the research are presented.

5.1 Contributions

This research focused on aspects related to the adoption and utilization of cloud computing in software testing and application development. In particular, the research concentrated on testing in the cloud and quality aspects in the context of cloud application development. The research also included a study on cloud gaming. This discussion about the results is aligned with the research questions stated in 3.1.

The information gathered during the separate studies addressing these research questions helps in understanding the process of adopting cloud computing for application development and testing. The results documented in the publications aim at enhancing the adoption processes related to cloud-based testing, which should enable organizations to apply cloud-based testing to increase profits, efficiency and productivity.

5.1.1 Definitions and perceptions of cloud-based testing

The first goal of the research was to establish an understanding of cloud-based testing by answering to the research questions: *What are the research problems related to testing in the cloud? What are the conditions that affect software testing as a service?* This goal was accomplished through *Publication I* and *Publication II*. In *Publication I,* we outlined the facets of testing in the cloud, dealing with ways in which cloud-based testing can be performed: (1) Availing the application under test online for testing; (2) hosting the testing infrastructure in the cloud and (3) testing the cloud itself. Robinson & Ragusa (2011) proposed several scenarios for performing cloud-based testing, all of which fall under the first and second facets.

In addition, the respondents were interested in knowing more about how to deal with different aspects related to testing in the cloud, which were discussed as research issues for software testing in the cloud. The respondents wanted to know, for example, what applications are best suited for testing in the cloud, how to deal with the data security, and how to predict the costs for implementing testing in the cloud. Other researchers have addressed some of the research problems. For example, Ciortea et al. (2010) describe a pricing model for their cloud-based testing service whereby the users are charged according to their test goal specifications.

Publication II covered the conditions that affect testing in the cloud. Cloud computing was discussed as the delivery model for online testing services. The benefits of cloud-based testing included flexibility, cost reduction and access to global markets. In terms of the challenges, the results revealed that the participants were concerned about how to effectively govern test data and provide services under reliable SLAs. In addition, migrating to cloud-based testing was seen as a complex process calling for efficient project and change management procedures. Parveen and Tilley (2010) address the complexities of migrating testing to the cloud based on the type of testing to be done and depending on the application being tested. *Publication III* adds on to the challenges related to handling load balancing, network latency, multitenancy and increase in testing complexities when testing interrelated systems.

The results also showed that successful implementation of cloud-based testing needed to have some factors in place, e.g. sufficient testing infrastructure, security, transparent pricing models, and efficient communication between stakeholders. It appeared that the need for domain knowledge influenced the acceptance of testing in the cloud. However, with increased awareness, even organizations that were opposed to the idea of cloud-based testing began to become interested. The increased awareness has generally been on the rise globally, as can be seen through increased attention to cloud-based testing by researchers and industry practitioners (Incki et al., 2012; Priyanka et al., 2012).

5.1.2 Cloud-based testing in practice

The second goal of this research was to provide empirical insights on how cloud-based testing is applied in practice. The research questions here were: *How can cloud-based testing be applied in practice? What should be considered when adopting cloud-based testing?*

The results in *Publication III* and *Publication IV* indicate that cloud computing is used for performance, scalability and stress testing. In addition, provisioning cloud-based testing infrastructures is fast and cost-effective, thus enabling organizations to conduct feasibility tests as well as testing across various platforms. Other ways for performing cloud-based testing are usability testing, testing CPU intensive tasks, and crowdsourced testing. These findings contribute to growing inquiries about ways to perform cloud-based testing, putting into consideration the testing levels e.g. system testing, and testing types, such as functional, compatibility and privacy aware testing (Incki et al., 2012; Priyanka et al., 2012; Wu et al., 2011).

Publication III focused on the effects of cloud computing on testing. Cloud-based testing was reported to reduce testing times, and the overall development cycles. Cloud-based testing resources can scale to imitate production environments, and this helps organizations to generate more realistic test results, and thereby improving the testing outcomes. The practitioners found that by performing cloud-based testing, testers and developers were able to work better, because they all had equal access to the system. Setting up resources for cloud-based testing is fast and this provided the practitioners with more time for dealing with essential business matters.

In order to address the need for supportive guidelines for adopting cloud-based testing, *Publication III* suggests a simple, five-step roadmap towards adopting testing in the cloud, and *Publication IV* provides a cloud-based testing strategy for use by organizations adopting cloud-based testing. The strategy involves five steps that advocate for continuous selection, evaluation and assessment of the cloud-based testing tasks, results, activities and stakeholders. Existing cloud adoption strategies are generic and may be applied when adopting cloud computing for different purposes (Khajeh-Hosseini et al., 2012; Zardari & Bahsoon, 2011). The strategy in *Publication IV* is unique because it is specifically aligned to support the adoption of cloud-based testing.

5.1.3 Quality in the context of cloud application development

Considering quality in relation to cloud computing, the third goal of the research was to understand the quality characteristics that cloud application developers deemed to be important. The research question in *Publication IV* was: *What are the important quality characteristics when performing cloud application development and how can they be realized?*

The results indicate that the important quality characteristics are different for different products and services, depending on the application being developed, the infrastructure being used for development, and the purpose of the application. Availability, functional suitability, reliability, security, operability, usability and performance efficiency were important. An interesting observation was the fact that usability was deemed to be important by all participants. This might be because there is a need to create a good first impression in order to capture and retain the attention of the end user (Orehovacki, 2011).

In order to achieve the desired quality, the results show that the organizations applied suitable tools, involved the customer when needed and selected appropriate development methods. The developers used tools and development frameworks that were most relevant for their specific applications. Customer input helped to improve the requirements, which led to incorporating better quality in the developed applications. Agile methods were most preferred because they allowed the organizations to react to changes, and hence, keep up with the quality expectations. The observed tendency towards agile practices correlates to the extended agile process model proposed by Patidar et al. (2011). The model aims at enhancing the effectiveness of developing cloud-based software by encouraging interaction between the developers, cloud providers and customers.

5.1.4 Cloud gaming in practice

The fourth goal of the research was to understand the relevance and utilization of cloud computing within small and medium sized gaming organizations. *Publication VI* focused on cloud gaming to answer the research question: *What are the factors that affect the adoption and use of cloud computing in game development?*

The results showed that cloud computing is also applicable in the gaming world, for the same reasons as in other business and application domains, i.e., cutting down costs and easy (flexible) access to computing infrastructure. The interviewed gaming organizations used cloud computing for simple tasks such as data storage and were of the opinion that cloud computing was not mature enough to support the technological demands of a reliable video game. These views support the notion that latency and scalability problems seem to be the most prevalent concerns with regard cloud-based games (Chang, 2010; Ross, 2009).

The participating organizations were aware of cloud computing, and acknowledged that cloud computing had the potential to change how games were distributed, leading to a greater amount of access and interaction between gaming organizations and the users. Overall, the cloud business models for cloud gaming were unclear, which was seen as a hindrance to adopting cloud computing for game development. Other researchers have also highlighted the need for clear business models to support the adoption of cloud computing in the gaming industry (Moreno et al., 2012; Ojala & Tyrväinen, 2011).

5.2 Implications for practice

This research focused on cloud-based development and testing, which is increasingly becoming relevant in software practices today. The dissertation identified the conditions that affect testing in the cloud (*Publication II*). The conditions included aspects such as the benefits, challenges, requirements, and enabling conditions related to testing in the cloud. The aim was to gather the factors to consider when thinking about the applicability of testing in the cloud. Organizations dealing with mission-critical systems require that the testers possess sufficient domain knowledge of the respective systems. Due to this restriction, such organizations were not willing to have their software tested by anonymous persons in the cloud. A potential solution would be to have private clouds dedicated for the individual testing needs of an organization.

In order to satisfy customer expectations,cloud testing service providers should also come up with transparent pricing models that truly reflect the worth of their work. Transparent pricing models also enable the customer to predict costs and budget accordingly. When exploring the possibility of testing in the cloud, organizations should also pay attention to security. Just as with other cloud services such as SaaS, a cloud-based testing service needs to be safe, bearing in mind the security of the test data and test results. Organizations can make take different approaches to test for security. For example, Zech et al. (2012) have developed a mechanism for testing the security of cloud computing environments based on potential risks and expected changes.

Test data management is a crucial part of any testing activity. With regard to testing in the cloud, the issue of test data management becomes even more important. Some testing tasks may require the use of actual customer or production data. However, the rules and regulations in some countries prohibit the customers from supplying sensitive or production data to third parties. A solution to this problem may be the development of new models that would generate almost "identical" test data to facilitate productive testing results. With regard to problems related to differences in data governance across various geographical areas, a viable solution is to allow the customer to choose where they would like to have their data stored, e.g., Amazon allows its customers to choose between five different regions.

Publication II also discusses cloud computing as the delivery model for software testing and *Publication I* gives the conceptual view about testing in the cloud i.e. the system under test is accessible online, testing infrastructure is hosted in the cloud, or testing of the cloud itself. Organizations should evaluate which approach best suits their testing needs as determined by internal requirements and guidelines. *Publication*

III and Priyanka et al. (2012) mention several cloud-based testing service providers that organizations can choose.

Publication III continued with the focus on testing in the cloud, and studied the effects of cloud computing on software testing. The results showed that cloud-based testing helps to perform more efficient performance, scalability and stress testing, while shortening the time it takes to perform testing. Cloud-based testing environments can be provisioned to replicate the production environments, which yield in more realistic testing results. The results also revealed that cloud-based testing provides organizations with the various options for addressing their testing needs. Organizations can therefore easily try out the different options before deciding, for example, on which testing tools to use. Organizations benefit from cloud-based testing because they do not have to go through long acquisition processes when in need of new testing infrastructure. This provides room for the organizations to focus on the business-critical aspects. Experiences from industry support these findings. For example, Sogeti reports that its cloud-based testing service is helping customers to reduce testing costs, shorten testing times and and improve quality (SOGETI, 2012).

Publication III also addressed the need for practical steps for adopting cloud-based testing. The publication proposed a roadmap that organizations can use when evaluating the viability of cloud-based testing. The roadmap is a five-step process which emphasizes on the need of *understanding cloud computing* first, then *conducting pilot projects* to 'test and see' how cloud-based testing would work. Organizations should *develop elaborate strategies* that can be followed when adopting cloud-based testing. Furthermore, they should focus on adequately addressing the challenges of cloud-based testing by *enhancing team interaction, preparing for complexities* and *enhancing cooperation between research and industry* in order to develop solutions for problems associated with cloud-based testing.

Publication IV focused on the adoption and utilization of cloud-based testing in practice. The results covered four categories that addressed different aspects of cloud-based testing. The first dealt with the motivations and reasons for adopting cloud-based testing resources. The second discussed different ways of utilizing cloud-based testing. The third and fourth categories covered the value-added features that cloud-based testing brings to testing resources and to the system under test. These results can help organizations in learning more about cloud-based testing, and evaluating their testing needs in order to identify the most beneficial way to apply cloud-based testing in practice.

The roadmap in *Publication III* recommends that organizations should come up with elaborate strategies to assist in adopting cloud-based testing. *Publication IV* carries on by developing a cloud-based testing strategy to assist organizations as they adopt and utilize cloud-based testing. The strategy advocates for organizations to evaluate their businesses and assess their needs so as to identify and select applicable and efficient cloud-based testing implementations. When applying cloud-based testing, organizations can employ a process improvement mindset to constantly re-assess the cloud-based testing experience and make the necessary changes. Continuous re-assessment is also recommended in other cloud adoption strategies (Khajeh-Hosseini et al., 2012; Zardari & Bahsoon, 2011).

Publication V focused on cloud application development practices. The study highlights the important quality characteristics for cloud applications, and the activities that contribute towards realizing the desired quality. When developing cloud applications, organizations should select the most suitable life-cycle models and tools that best support the development processes. They should constantly engage the customer in order to develop applications that bring the most value to the end users. At the same time, organizations should collaborate with the cloud provider in order to align the applications with the platform requirements at all times. Usability was found to be a vital quality attribute for cloud applications. This is an important observation because usability is connected to QoE and QoS management, which seems to be crucial when dealing with cloud applications and services (Garg et al., 2013; Hobfeld et al., 2012).

The results of *Publication VI* discussed about the relevance and utilization of cloud computing in small and medium-sized gaming organizations. The results can be used in comparable organizations to develop the processes for supporting the use of cloud computing (including cloud-based testing) in data-intensive software services and products. Essentially, even though there are concerns about latency and bandwidth problems (Chang, 2010; Gaudiosi, 2013), it seems that cloud computing is expected to transform gaming products into services, and provide a wider distribution channel for delivering the games. Gaming organizations can take these results to prepare for the expected changes as they develop and test their game products.

5.3 Implications for further research

Cloud computing has become more common within the software industry and there is need for academic research to address different research issues associated with it. The aim of *Publication I* was to discuss the research issues for software testing in the cloud. The research issues dealt with application and management specific details regarding testing in the cloud, together with the legal and financial matters. The study aimed at bringing the research issues to the attention of the research community, and served as a call for further research on testing in the cloud. As such, *Publication I* has been a source of input for several researchers thereafter, for example, (Incki et al., 2012; Priyanka et al., 2012; Tung & Tseng, 2013).

This dissertation addressed the applicability of cloud computing for testing within organizational and strategic contexts. The study identified the factors to consider when adopting cloud-based testing, such as, the benefits, risks and requirements. These results can be used as basis for researching on the practical-level problems related to cloud-based testing, such as, daily management of cloud-based testing (Wu et al., 2011), evaluating practical cloud-based testing work (Yu et al., 2010), and assessing the appropriateness of cloud-based testing for specific testing needs (Robinson & Ragusa, 2011).

The dissertation observed that small and medium sized organizations were more willing to use cloud-based resources for testing needs. This can be attributed to the fact that cloud computing provides a quick and economical way of accessing the required infrastructure. On the other hand, large organizations employ strategic approaches when adopting cloud-based testing. In general, the size of an organization affects how it approaches and utilizes cloud computing (Marston et al., 2011). The size of the organization seems to have an effect on how cloud-based testing is adopted, and this could be investigated in order to address the unique characteristics of different organizations.

Cloud computing introduces a new paradigm for developing and delivering software applications. *Publication V* evaluated the quality characteristics that organizations want to fulfil when they develop their cloud-based applications, along with the activities that they undertake to meet the desired quality attributes. The scope of this study could be extended to further look at how cloud-based software development affects development, testing and quality of cloud software applications, systems and services, e.g., by considering the different trust boundaries across different cloud service models, and looking at what developers need to do to ensure that their code is secure.

The study on cloud gaming demonstrates that it is important to consider the business or application domain before adopting cloud computing. The results indicate that cloud computing does not adequately address the technological needs of the gaming industry. It is no wonder that Ross (2009) came up with a supercomputer designed specifically for processing gaming applications. There is need for further research on how cloud computing can sufficiently support gaming applications, and this can be combined with developing cloud-based testing procedures that are suitable for the needs of the gaming industry.

5.4 Evaluation and validity threats of the research

This section evaluates the research and summarizes the main limitations and threats to the validity of the research. The research is evaluated using Corbin & Strauss (2008) criteria for evaluating a grounded theory study.

5.4.1 Evaluation of the research

This research was qualitative in nature and applied grounded theory as the main research method. Corbin & Strauss (2008) recommend evaluating a grounded theory study based on ten criteria: fit, applicability, concepts, contextualization of concepts, logic, depth, variation, creativity, sensitivity, and evidence of memos (Section 3.4). This subsection evaluates the dissertation against these criteria. Some of the criteria are combined because they are related to each other.

Fit, applicability and depth: Publications reporting the results of this research were shared with the participants and were presented in various events, e.g., meetings, seminars, workshops and international conferences. The practitioners recognized their own experiences in the concepts described by the results, and this helped to build our confidence in the validity of the research results. The practitioners found the results to be relevant and had plans of using the results in their work. For example, two of the practitioners provided feedback via email as follows:

> *"Thanks for the journal paper, interesting read! I have shared it with the team here too."* (Founder and partner, Organization 3 featured in *Publication III*).
>
> *"Thank you. I will definitely read this in upcoming days. Hopefully the results of the research will support me with defining the focus for next years."* (Performance testing unit leader, Organization featured in *Publication I, II, III,* and *IV*)

Concepts, conceptualization of concepts, logic, and creativity: All publications except *Publication IV* have been published in international conferences and journals, and in a book. They have gone through the scientific review process which provides critics on the results and how they are presented. The scientific review process has been a constructive input towards describing the results in understandable, clear, and logical concepts. Discussions with the supervisors during data analysis and reporting also helped to formulate the results concretely.

Variation: The results described in the publications are grounded from data that was collected through interviews. The publications contain many interview quotations reflecting thoughts, ideas and experiences of practitioners representing different companies, business domains and applications.

Sensitivity: When collecting data through interviews, a researcher might encounter difficulties in discerning the reality and relevance of the answers provided. The use of open-ended questions provided the opportunity to clarify more on the questions, so as to gather a variety of descriptions and answers regarding one question.

Evidence of memos: Memos were used during the analysis as described in subsection 3.5.2.

5.4.2 Limitations of the research

This subsection contains a summary of the threats to validity of the research in relation to construct validity, external validity, reliability and reactivity (Maxwell, 2005; Runeson & Höst, 2009).

Construct validity: Threats to construct validity are concerned with the extent to which the researcher is able to study and evaluate the concepts as intended (Runeson & Höst, 2009). At the beginning of this research, the threat to construct validity related to the rather new and limited understanding about software testing in the cloud. In *Publication I* and *II,* the respondents' interpretation of software testing in the cloud relied on their perceptions. However, we interviewed respondents from different business domains, each expressing their opinions, and thereby providing us with a balanced view about the topic. To improve the construct validity, *Publication III, IV,* and *V* gathered views from organizations that had applied cloud computing. The results presented in *Publication III* and *IV* complemented and extended the earlier results, and this helped to increase the trustworthiness of the results.

External validity: Threats to external validity relate to the generalizability of the results and the researcher tries to evaluate whether the results can apply in other situations outside the studied context (Runeson & Höst, 2009). The main limiting context variable is that all except one organization involved in this dissertation were located in Finland. Different views might have come from organizations in countries with different cultures. However, the data was gathered through several interview rounds, which allowed us to have a variety of views from different participants. This helped in establishing the general factors of the different concepts addressed in each publication.

Reliability: Reliability deals with the extent to which the researcher might influence the repeatability of the study with the same results (Runeson & Höst, 2009). In qualitative studies, the researcher is the main instrument of the research, and this usually poses the danger of researcher bias. Creswell and Miller (2000) mention triangulation as an approach for ensuring the reliability of the findings. Triangulation deals with gathering a common perspective from different information sources. There are four types of triangulation techniques involving data sources, theories, methods and investigators (Creswell & Miller, 2000). We applied triangulation among different researchers in this dissertation. Different researchers were involved during the preparation of the interviews and during data collection at different phases. In addition, during the data analysis, discussions were held with the supervisors of this dissertation and other researchers. We also applied triangulation across data sources by interviewing respondents from different organizations and analysing the data. In addition, other sources of written data, for example, memos, reports, and web sites were used.

Reactivity: Reactivity refers to a researcher's potential to influence the interview atmosphere during the interview (Maxwell, 2005). The data collected for this dissertation made use of open-ended questions, introducing the risk of the researcher imposing his/her opinions on the interviewees. If the respondents required clarification about the questions, the researchers exercised caution so as not to influence the views that the respondents would express.

6 Conclusions

This dissertation made use of empirical research methods to study the applicability of cloud computing in application development and testing. This chapter summarizes the contributions of this research and outlines the directions for future work.

6.1 Contributions and summary

The scope was limited to cloud-based testing together with quality aspects in the context of cloud application development and cloud gaming. The dissertation consisted of four main phases. The first phase studied industry practitioners' perceptions about testing in the cloud, and gathered research issues for testing in the cloud. The second phase studied about the effects of cloud computing on software testing, along with observing how organizations adopt and utilize cloud-based testing. Additionally, this phase developed a roadmap towards testing in the cloud, and a cloud-based testing strategy. The third phase focused on quality, and evaluated the activities that help to achieve desired quality during cloud application development. The fourth phase provided more insights on cloud adoption from a different angle, i.e., cloud gaming and utilisation of cloud computing in the context of the gaming industry.

The results in this research provide high abstraction concepts and constructs that may be used in more detailed studies related to the application of cloud-based testing in specific contexts. The results for both researchers and industrial practitioners can be summarized as follows:

- The concept of cloud computing is wide and extensive; therefore, it would be most beneficial to study it in specific contexts, such as testing, application development, and game development.
- Cloud-based testing can be applied to increase efficiency and profitability of testing practices within different organizations.
- Organizations appreciate the fact that cloud-based testing enables quick and on-demand access to required testing infrastructure.
- Cloud-based testing should be aligned to different requirements as determined by the needs of the organization, for example, issues related to security and test data management should be well addressed.
- Organizations with great security and trust requirements, such as those dealing with mission-critical systems can also adopt cloud-based testing through dedicated private clouds.
- Cloud-based testing affects:
    - The acquisition model, which emphasizes software services instead of software products.
    - The business model, which causes a shift from licence based fees to pay-per use pricing.
    - The access model, whereby, dedicated testing environments are replaced by cloud-based testing environments.
    - The technical models of testing, for example, new opportunities for performance testing
- The challenges of adopting cloud-based testing can be demystified by following roadmaps and strategies that provide guidelines to support the decision-making processes.
- When deploying or developing applications in cloud environments, some quality attributes, such as, security and usability become very important. However, the type of application or business domain is also key determinants to the kind of quality characteristics that are important.
- Cloud computing still requires maturing in order to support the technological needs required for developing, deploying and testing cloud-based gaming applications.

In summary, cloud computing is relevant and applicable for testing and application development, as well as other areas such as game development. Research on cloud-based testing has grown within the last few years, during which this research was conducted. The results of this research provide a better understanding towards the adoption, utilization and effects of cloud-based testing for both practitioners and researchers. The study on cloud gaming shows that there are different requirements for adopting cloud computing. Therefore, the unique characteristics of the context within which to apply cloud computing, and consequently cloud-based testing, should always be considered.

6.2 Future research topics

Sections 5.2 and 5.3 mentioned the various research implications. The effectiveness and efficiency of cloud-based testing requires more research and evidence to draw reliable conclusions. In particular, empirical evaluations in industrial settings would provide more knowledge and evidence to facilitate effective cloud-based testing.

There is a need for research on the methods, tools, or processes for managing, planning and assessing cloud-based testing as it is applied in practice. To evaluate the real benefits and limitations of cloud-based testing, there should be studies conducted in real-life projects and processes, if possible. It is important to study how practitioners can benefit from the strengths of cloud-based testing without risking important aspects such as security, control and technology requirements.

This research has not addressed the issue of designing respective cloud-based testing implementations. This is an important research area because it would lead to: (1) developing cloud-based testing techniques to fit with different requirements, e.g., mission-critical systems and gaming applications; (2) performing different testing types in the cloud and provide concrete results on how to deal with issues such as security, latency and bandwidth bottlenecks.

The study on quality in the context of cloud application development (*Publication V*) could be extended with in-depth studies focusing on how to measure, and test for different quality characteristics of cloud-based applications. For cloud applications, the developer and the cloud provider share the responsibility of ensuring secure software development. Further research could address how this responsibility can be accounted for in the process of application development, testing and deployment. A wider sample of organizations could be included, or a survey could be conducted, to widen the scope and impact of the study in understanding the desired quality characteristics of cloud applications along with testing techniques that can be used to test corresponding quality characteristics.

References

Agilemanifesto.org, 2001. *Manifesto for Agile Software Development*. [Online] Available at: http://agilemanifesto.org/.

Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I. & Zaharia, M., 2009. *Above the Clouds: A Berkeley View of Cloud Computing*. Electrical Engineering and Computer Sciences, University of California at Berkeley.

ATLAS.ti, 2013. *Qualitative Data Analysis*. Available at: http://www.atlasti.com [Accessed 2 October 2013].

Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I. & Warfield, A., 2003. Xen and the Art of Virtualization, In Proceedings of the 19th ACM Symposium on Operating Systems Principles, pp. 164-177.

Bennett, K., Layzell, P., Budgen, D., Brereton, P., Macaulay, L., & Munro, M., 2000. Service-Based Software: The Future for Flexible Software, In Proceedings of the Seventh Asia-Pacific Software Engineering Conference, pp. 214-221.

Blaine, J.D. & Cleland-Huang, J., 2008. Software Quality Requirements: How to Balance Competing Priorities. *IEEE Software*, 25(2), pp.22-24.

Boehm, B., 2006. A View of 20th and 21st Century Software Engineering, In Proceedings of the 28th International Conference on Software Engineering, pp. 12-29.

Buyya, R., Yeo C.S., Venugopal, S., Broberg, J. & Brandic, I., 2009. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25(6), pp.599-616.

Chan, D., 2010. *On The Feasibility of Video Gaming onDemand in Wireless LAN/WIMAX*. Burnaby, BC, Canada: Unpublished.

Chang, T., 2010. Gaming Will Save Us All. *Communications of the ACM*, 53(3), pp.22-24.

Chauhan, M.A. & Babar, M.A., 2012. Cloud Infrastructure for Providing Tools as a Service: Quality Attributes and Potential Solutions, In Proceedings of the IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture, pp. 5-13.

Ciortea, L., Cristian Zamfir, S.B., Chipounov, V. & Candea, G., 2010. Cloud9: A Software Testing Service. *ACM SIGOPS Operating Systems Review*, 43(4), pp.5-10.

Coleman, G. & O'Connor, R., 2008. Using Grounded Theory to Understand Software Process Improvement: A Study of Irish Software Product Companies. *Information and Software Technology*, 49(6), pp.654-67.

Collard, R., 2009. Performance Innovations, Testing Implications. *Software Test & Performance Magazine*, 6(8), pp.19-20.

European Commission, 2005. *The New SME Definition: User Guide and Model Declaration, Enterprise and Industry Publications*. : http://ec.europa.eu/enterprise/policies/sme/files/sme_definition/sme_user_guide_en.pdf.

Corbin, J. & Strauss, A., 2008. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. 3rd ed, Sage Publications.

Costa, P.M., Pitt, J., Cunha, J.F.e. & Galvao, T., 2012. Cloud2Bubble: Enhancing Quality of Experience in Mobile Cloud Computing Settings, In Proceedings of the 3rd ACM Workshop on Mobile Cloud Computing and Services, pp. 45-52.

Creswell, J.W. & Miller, D.L., 2000. Determining Validity in Qualitative Inquiry. *Theory into Practice*, 39(3), pp.124-30.

Dalkey, N.C., 1969. *The Delphi Method: An Experimental Study of Group Opinion*. Santa Monica: The Rand Corporation.

Dey, I., 1999. *Grounding Grounded Theory: Guidelines for Qualitative Inquiry*. Academic Press.

Dubey, A. & Wagle, D., 2007. *The McKinsey Quarterly*. Available at: http://www.mckinsey.de/downloads/publikation/mck_on_bt/2007/mobt_12_Delivering_Software_as_a_Service.pdf [Accessed 20 April 2011].

Durkee, D., 2010. Why Cloud Computing Will Never Be Free. *Communications of the ACM*, 53(5), pp.62-69.

Easterbrook, S., Singer, J., Storey, M.-A. & Damian, D., 2008. Selecting Empirical Methods for Software Engineering Research. In F. Shull, J. Singer & D.I.K. Sjoberg, eds. *Guide to Advanced Empirical Software Engineering*. London: Springer-Verlag. pp.285-311.

Ebert, C., 2008. A Brief History of Software Technology. *IEEE Software*, 25(6), pp.22 - 25.

Eisenhardt, K.M., 1989. Building Theories from Case Study Research. *Academy of Management Review*, 14(4), pp.532-50.

Fink, A., 2003. *The Survey Handbook*. 2nd ed, SAGE Publications.

Fink, A. & Kosecoff, J., 1985. *How to Conduct surveys A Step-by-Step Guide*. SAGE Publications.

Gao, J., Bai, X. & Tsain, W.-T., 2011. Cloud Testing - Issues, Challenges, Needs and Practice. *Software Engineering: An International Journal*, 1(1), pp.9-23.

Garg, S.K., Versteeg, S. & Buyya, R., 2013. A Framework for Ranking of Cloud Computing Services. *Future Generation Computer Systems*, 24(4), pp.1012-23.

Garvin, D.A., 1984. What does "Product Quality" Really Mean? *Sloan Management Review*, (4), pp.25-43.

Gasson, S., 2004. Rigor in Grounded Theory Research: An Interpretive Perspective on Generating Theory from Qualitative Field Studies. In *Rigor in Grounded Theory Research*. pp.79-102.

Gaudiosi, J., 2013. *Cloud Gaming Europe 2013*. Available at: http://www.videogamesintelligence.com/cloud-gaming-europe/pdf/CloudGamingEuropeReport.pdf [Accessed 28 May 2013].

Geelan, J., 2009. *Twenty-One Experts Define Cloud Computing*. Available at: http://virtualization.sys-con.com/node/612375 [Accessed 17 April 2013].

Glaser, B., 1992. *Basics of Grounded Theory Analysis: Emergence vs Forcing*. Sociology Press.

Glaser, B. & Strauss, A., 1967. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine Transaction.

Gold, N., Mohan, A., Knight, C. & Munro, M., 2004. Understanding Service-Oriented Software. *IEEE Software*, 21(2), pp.71-77.

Goth, G., 2008. "Googling" Test Practices? Web Giant's Culture Encourages Process Improvement. *IEEE Software*, 25(2), pp.92-94.

Grundy, J., Kaefer, G., Keung, J. & Liu, A., 2012. Software Engineering for the Cloud. *Focus: Guest Editors' Introduction, IEEE Software*, 29(2), pp.26-29.

Hassan, Q.F., 2011. *Demistifying Cloud Computing*. Available at: http://www.crosstalkonline.org/storage/issue-archives/2011/201101/201101-Hassan.pdf [Accessed 27 February 2012].

Heiser, J.E., 1997. An Overview of Software Testing, IEEE Autotestcon Proceedings, pp. 204-211.

Hevner, A.R., March, S.T., Park, J. & Ram, S., 2004. Design Science in Information Systems Research. *MIS Quaterly*, 28(1), pp.75-105.

Hobfeld, T., Schatz, R., Varela, M. & Timmerer, C., 2012. Challenges of QoE Management for Cloud Applications. *IEEE Communications Magazine*, 50(4), pp.28-36.

Hoda, R., Noble, J. & Marshall, S., 2012. Developing a Grounded Theory to Explain the Practices of Self-Organizing Agile Teams. *Empirical Software Engineering*, 17(6), pp.609-39.

Hosono, S., Jiafu H., Xuemei L., Lin L., He H. & Yoshino, S., 2011. Fast Development Platforms and Methods for Cloud Applications, In Proceedings of the 2011 IEEE Asia-Pacific Services Computing Conference (APSCC), pp. 94-101.

IEEE/ANSI, 1990. *IEEE Standard Glossary of Software Engineering Terminology, 610.12-1990*.

Incki, K., Ari, I. & Sözer, H., 2012. A Survey of Software Testing in the Cloud, In Proceedings of the IEEE 6th International Conference on Software Security and Reliability Companion (SERE-C), pp. 18-23.

ISO/IEC, 2005. *ISO/IEC 25000 Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - System and Software Quality Models*.

ISO/IEC, 2010. *ISO/IEC 25010 Systems and Software Engineering - Software Product Quality Requirements and Evaluation - Quality Models for Software Product Quality and System Quality in Use*.

ISO, 2000. *ISO 9000: Quality Management Systems - Fundamentals and Vocubulary*. International Organization for Standardization.

ISO, 2013. *Distributed Application Platforms and Services (DAPS)*. Available at: http://www.iso.org/iso/jtc1_sc38_home [Accessed 15 October 2013].

ITU-T, 2008a. *Definitions of Terms Related to Quality of Service, Recommendation ITU-T E.800*.

ITU-T, 2008b. *Vocabulary for Performance and Quality of Service, Recommendation P.10/G.100*.

Jones, M.L., Kriflik, G.K. & Zanko, M., 2005. Grounded Theory: A Theoretical and Practical Application in the Australian Film Industry, In Proccedings of the International Qualitative Research Convention.

Järvinen, P., 2004. *On Research Methods*. Opinpajan Kirja.

Kafetzakis, E., Koumaras, H., Kourtis, M.A. & Koumaras, V., 2012. QoE4CLOUD: A QoE-driven Multidimensional Framework for Cloud Environments, In Proceedings of International Conference on Telecommunications and Multimedia (TEMU), pp. 77-82.

Kasurinen, J., Taipale, O. & Smolander, K., 2010. Test Case Selection and Prioritization: Risk-based or Design-based?, In Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), Article No. 10.

Kasurinen, J., Taipale, O., Vanhanen, J. & Smolander, K., 2011. Exploring Perceived Quality in Software Organizations, In Proceedings of the Fifth International Conference on Research Challenges in Information Science, pp. 1-12.

Khajeh-Hosseini, A., Greenwood, D., Smith, J.W. & Sommerville, I., 2012. The Cloud Adoption Toolkit: Supporting Cloud Adoption Decisions in the Enterprise. *Software - Practice & Experience*, 42(4), pp.447-65.

Kit, E., 1995. *Software Testing in the Real World: Improving the Process*. Addison-Wesley, Reading, MA.

Kitchenham, B. & Pfleeger, S.L., 1996. Software Quality: The Elusive Target. *IEEE Software*, 13(1), pp.12-21.

Klein, H.K. & Myers, M.D., 1999. A Set of Principles for Conductiong and Evaluationg Interpretive Field Studies in Information Systems. *MIS Quarterly*, 23(1), pp.67-94.

Kusters, R.J., Solingen, R.V. & Trienekens, J.J.M., 1997. User-Perceptions of Embedded Software Quality., 1997. In Proceedings of 8th International Workshop on Software Technology and Engineering Practice, pp. 184-197.

Leavitt, N., 2009. Is Cloud Computing Really Ready for Prime Time? *Computer*, 11(2), pp.10-13.

Lee, J.Y., Lee, J.W., Cheun, D.W. & Kim, S.D., 2009. A Quality Model for Evaluating Software-as-a-Service in Cloud Computing, In Proceedings of the 7th ACIS International Conference on Software Engineering Research, Management and Applications, pp. 261-266.

Lenk, A., Klems, M., Nimis, J., Tai, S. & Sandholm, T., 2009. What's Inside the Cloud? An Architectural Map of the Cloud Landscape, In Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, pp. 23-31.

Linthicum, D.S., 2010. *Cloud Computing and SOA Convergence in your Enterprise: A Step-by-Step Guide*. Addison-Wesley.

Lovelock, C.H., 2000. *Services Marketing*. Prentice Hall Europe.

Lu, T., Chen, M. & Andrew, L.L.H., 2013. Simple and Effective Dynamic Provisioning for Power-Proportional Data Centers. *IEEE Transactions on Parallel and Distributed Systems*, 24(6), pp.1161-71.

Lu, W., Jackson, J. & Barga, R., 2010. AzureBlast: A Case Study of Developing Science Applications on the Cloud, In Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing, pp. 413-420.

Maggiorini, D. & Ripamonti, L.A., 2011. Cloud Computing to Support the Evolution of Massive Multiplayer Online Games. Vilamoura, Portugal, In Proceedings of the International Conference on ENTERprise Information Systems, pp. 101-110.

March, S.T. & Smith, G.F., 1995. Design and Natural Science Research on Information Technology. *Decision Support Systems*, 15(4), pp.251-66.

Marston, S., Li, Z., Bandyopadhyay, S., Zhang J. & Ghalsasib, A., 2011. Cloud Computing: The Business Perspective. *Decision Support Systems*, 51(1), pp.176-89.

Maxwell, J.A., 2005. *Qualitative Research Design: An Interactive Approach*. Sage Publications.

Mell, P. & Grance, T., 2011. *The NIST Definition of Cloud Computing*. NIST Special Publication 800-145. The National Institute of Standards and Technology (NIST).

Meyer, C.B., 2001. A Case in Case Study Methodology. *Field Methods*, 13(4), pp.329-52.

Mikkonen, T. & Taivalsaari, A., 2013. Cloud Computing and its Impact on Mobile Software Development: Two Roads Diverged. *Journal of Systems and Software*, 86(9), pp. 2318-2320.

Moreno, C., Tizon, N. & Preda, M., 2012. Mobile Cloud Convergence in GaaS A Business Model Proposition, In Proceedings of the 45th Hawaii International Conference on System Sciences, pp. 1344-1352.

Myers, G.J., 2004. *The Art of Software Testing*. 2nd ed. John Wiley & Sons, Inc.

Ojala, A. & Tyrväinen, P., 2011. Developing Cloud Business Models: A Case Study on Cloud Gaming. *IEEE Software*, 28(4), pp.42-47.

Olsen, E.R., 2006. Transitioning to Software as a Service: Realigning Software Engineering Practices with the New Business Model, In Proceedings of the IEEE International Conference on Service Operations and Logistics, and Informatics, pp. 266-271.

Orehovacki, T., 2011. Perceived Quality of Cloud Based Applications for Collaborative Writing. In *Business Systems and Services: Modeling and Development, Information Systems Development*. pp.575-86.

Orlikowski, W.J. & Baroudi, J.J., 1991. Studying Information Technology in Organizations: Research Approaches and Assumptions. *Informations Systems Research* , 2(1), pp.1-28.

Osterweil, L.J., 1997. Software Processes are Software too, Revisited: An Invited Talk on the Most Influential Paper on ICSE 9, In Proceedings of the 19th International Conference on Software Engineering, pp. 540-548.

Papazoglou, M.P., Traverso, P., Dustdar, S. & Dustdar, S., 2007. Service-Oriented Computing: State of the Art and Research. *IEEE Computer*, 40(11), pp.38-45.

Parveen, T. & Tilley, S., 2010. When to Migrate Testing to the Cloud. In Proceedings of the 2nd International Workshop on Software Testing in the cloud (STITC), as part of 3rd IEEE International Conference on Software Testing, Verification and Validation (ICST), pp. 424-427.

Patidar, S., Rane, D. & Jain, P., 2011. Challenges of Software Development on Cloud Platform, In Proceedings of the World Congress on Information and Communication Technologies (WICT), pp. 1009-1013.

Pfleeger, S.L. & Kitchenham, B.A., 2001. Principles of Survey Research: Part 1: Turning Lemons into Lemonade. *Software Engineering Notes*, 26(6), pp.16-18.

PricewaterhouseCoopers, 2010. *Global Entertainment and Media Outlook: 2010-2014*. PriceWaterhouseCoopers.

PricewaterhouseCoopers, 2013. *Global entertainment and media outlook: 2012-2016*. [Online] Available at: http://www.pwc.com/gx/en/global-entertainment-media-outlook/segment-insights/video-games.jhtml [Accessed 22 May 2013].

Priyanka, Chana, I. & Rana, A., 2012. Empirical Evaluation of Cloud-based Testing Techniques: A Systematic Review. *ACM SIGSOFT Software Engineering Notes*, 37(3), pp. 1-9.

Qian, H., Medhi, D. & Trivedit, K., 2011. A Hierarchical Model to Evaluate Quality of Experience of Online Services hosted by Cloud Computing, In Proceedings of the 12th IFIP/IEEE International Symposium on Integrated Network Management, pp. 105-112.

Riungu, L.M., Taipale, O. & Smolander, K., 2010. Software Testing as an Online Service: Observations from Practice. In Proceedings of the Third International Conference on Software Testing, Verification, and Validation Workshops, pp. 418-423.

Robinson, P. & Ragusa, C., 2011. Taxonomy and Requirements Rationalization for Infrastructure in Cloud-based Software Testing, In Proceedings of the 3rd IEEE International Conference on Cloud Computing Technology and Science, pp. 454-461.

Ross, P.E., 2009. Cloud Computing's Killer App: Gaming. *IEEE Spectrum*, 43(3), pp.14-14.

Runeson, P. & Höst, M., 2009. Gudelines for Conducting and Reporting Case Study Research in Software Engineering. *Empirical Software Engineering*, 14(2), pp.131-64.

Sahoo, M., 2009. IT Innovations: Evaluate, Strategize and Invest. *IT Professional*, 11(6), pp.16-22.

Seaman, C.B., 1999. Qualitative Methods in Empirical Studies of Software Engineering. *IEEE Transactions on Software Engineering*, 25(4), pp.557-72.

Seth, F.P., Mustonen-Ollila, E., Taipale, O. & Smolander, K., 2012. Software Quality Construction: Empirical Study on the Role of Requirements, Stakeholders and Resources, In Proceedings of the 19th Asia-Pacific Software Engineering Conference, pp. 17-26.

Sheehan, M., 2013. *15 Top Cloud Computing Use Cases*. Available at: http://www.hightechdad.com/2013/03/22/15-top-cloud-computing-use-cases/ [Accessed 13 September 2013].

Smolander, K., 2002. Four Metaphors of Architecture in Software Organizations: Finding out the Meaning of Artchitecture in Practice, In Proceedings of the International Symposium on Empirical Software Engineering (ISESE 2002), pp. 211-221.

Sodhi, B. & Prabhakar, T.V., 2011. Assessing Suitability of Cloud Oriented Platforms for Application Development, In Proceedings of the 9th Working IEEE/IFIP Conference on Software Architecture (WICSA), pp. 328-335.

SOGETI, 2012. *STaaS - Software Testing as a Service*. Available at: http://www.sogeti.com/staas [Accessed 13 November 2013].

Sommerville, I., 2001. *Software Engineering*. 6th ed. Harlow: Pearson Education Limited.

Spirent, 2010. *The ins and outs of cloud computing and its impacts on the network*. Available at: http://www.spirent.com/~/media/White%20Papers/Broadband/PAB/Cloud_Computing_WhitePaper.ashx [Accessed 10 June 2010].

Strauss, A. & Corbin, J.M., 1990. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. SAGE Publications.

Strauss, A. & Corbin, J.M., 1998. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. 2nd ed. SAGE Publications.

Sulistio, A., Yeo, C.S. & Buyya, R., 2004. A Taxonomy of Computer-based Simulations and its Mapping to Parallel and Distributed Systems Simulation Tools. *Software: Practice and Experience*, 34(7), pp.653-73.

Sultan, N.A., 2011. Reaching for the "Cloud": How SMEs can Manage. *International Journal of Information Management*, 31(3), pp.272-78.

Sun, W., Zhang, K., Chen, S-K., Zhang, X. & Liang, H., 2007. Software as a Service: An Integration Perspective, In Proceedings of the Fifth International Conference on Service-Oriented Computing, pp. 558-569.

Taipale, O., 2007. *Observations on Software Testing Practice*. Doctoral Thesis. Lappeenranta: Acta Universitatis Lappeenranta University of Technology.

Taipale, O. & Smolander, K., 2006. Improving Software Testing by Observing Practice, In Proceedings of the 5th ACM-IEEE International Symposium on Empirical Software Engineering (ISESE), pp. 262-271.

Tassey, G., 2002. *The Economic Impacts of Inadequate Infrastructure for Software Testing*. Final Report, U.S. National Institute for Standards and Technology.

Tung, Y.-H. & Tseng, S.-S., 2013. A Novel Approach to Collaborative Testing in a Crowdsourcing Environment. *Journal of Systems and Software*, 86(8), pp.2143-53.

Turner, M., Budgen, D. & Brereton, P., 2003. Turning Software into a Service. *Computer*, 36(10), pp.38-44.

Tyni, H., Sotamaa, O. & Toivonen, S., 2011. Howdy Pardner!: On Free-to-Play, sociability and Rhythm Design in FronterVille, In Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, pp. 22-29.

Urquhart, C., 2001. An Encounter with Grounded Theory: Tackling the Practical and Philosophical Issues. In E.M. Trauth, ed. *Qualitative Research in IS: Issues and Trends*, pp.104-140.

uTest, 2013. *uTest*. Available at: http://www.utest.com/ [Accessed 7 May 2013].

van der Aalst, L., 2009. *Software Testing as a Service (STaaS)*. Available at: http://www.tmap.net/Images/Paper%20STaaS_tcm8-47910.pdf [Accessed 6 May 2009].

Wang, X., Du Z., Liu X., Xie H., & Jia, X., 2010. An Adaptive QoS Management Framework for VoD Cloud Service Centers, In Proceedings of the International Conference on Computer Application and System Modeling (ICCASM 2010), pp. V1-527-V1-532.

Wilson, D.N. & Hall, T., 1998. Perceptions of Software Quality: A Pilot Study. *Software Quality Journal*, 7, pp.67-75.

Voas, J. & Zhang, J., 2009. Cloud Computing: New Wine or Just a New Bottle? *IT Professional*, 11(2), pp.15-17.

Wu, J., Wang, C., Liu, Y. & Zhang, L., 2011. AGARIC - A Hybrid Cloud Based Testing Platform, In Proceedings of the International Conference on Cloud and Service Computing, pp. 87-94.

Yamakami, T., 2012. Revenue-generation pattern analysis of mobile social games in Japan. PyeongChang, In Proceedings of the 14th International Conference on Advanced Communication Technology (ICACT), pp. 1232-1236.

Yau, S.S. & An, H.G., 2011. Software Engineering Meets Services and Cloud Computing. *Computer*, 44(10), pp.47-53.

Yu, L., Tsai, W-T., Chen, X., Liu L., Zhao Y., Tang L., & Zhao W., 2010. Testing as a Service over Cloud, In Proceedings of the Fifth IEEE International Symposium on Service Oriented System Engineering, pp. 181-188.

Zardari, S. & Bahsoon, R., 2011. Cloud Adoption: A Goal-oriented Requirements Engineering Approach. In Proceedings of the 2nd International Workshop on Software Engineering for Cloud Computing, pp. 29-35.

Zech, P., Felderer, M. & Breu, R., 2012. Towards a Model Based Security Testing Approach of Cloud Computing Environments, In Proceedings of the 6th International Conference on Software Security and Reliability Companion (SEREC-C), pp. 47-56.

Zheng, Z., Zhang, Y. & Lyu, M.R., 2010. CloudRank: A QoS-Driven Component Ranking Framework for Cloud Computing, In Proceedings of the 29th IEEE Symposium on Reliable Distributed Systems, pp. 184-193.

Zhou, N., An, D.P., Zhang, L.-J. & Wong, C.-H., 2011. Leveraging Cloud Platform for Custom Application Development, In Proceedings of the 2011 IEEE International Conference on Services Computing (SCC), pp. 584-591.