Project Management Methodologies in SaaS Deployment Projects

Master's Thesis
Department of Real Estate, Planning and Geoinformatics, School of Engineering, Aalto University

Espoo, 20 May 2015

Bachelor of Science in Technology
Ida Korhonen

Supervisor: Professor Seppo Junnila
Instructors: Professor Karlos Artto and M.Ec. Mikko Kallio

Aalto University
School of Engineering

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi

Abstract of master's thesis

Author Ida Korhonen

Title of thesis Project Management Methodologies in SaaS Deployment Projects

Degree programme Degree Programme in Real Estate Economics

Major/minor Real estate management **Code** M3003

Thesis supervisor Professor Seppo Junnila

Thesis advisor(s) Professor Karlos Artto, M. Ec. Mikko Kallio

Date 20.05.2015 **Number of pages** 66 + 10 **Language** English

This thesis focuses on studying Project Management Methodologies (PMM) in SaaS (Software as a Service) deployment projects. SaaS is a cloud computing service model and a new software delivery method which is increasingly popular on the IT market. PMM is a set or a combination of guidelines, practices and tools which can be utilized in the deployment projects in order to successfully manage the project execution and to deliver the project. This thesis studies the characteristics of PMMs that are most suitable in managing SaaS deployment projects, with an aim of finding the most effective project management methods, tools and practices which enhance the success of these projects. Additionally, this thesis studies the special characteristics of different SaaS deployment and architecture types in order to understand their possible impacts on the selected PMM. Finally, this thesis studies whether certain PMM activities or areas are specifically impacted or should be of specific concern in a typical SaaS deployment project.

The research approach of this thesis is a case study consisting of two case projects executed by one single case organization. The data for this research was collected by interviewing four case project managers and by conducting a survey for nine case project team members. Furthermore, the author of this thesis works as a project manager in the case projects which helped in making observations on the projects and in the analysis of the project related documentation.

Typically, when new technologies, methods or models are used in a project the lack of knowledge raises the risk of misunderstandings. Proper project planning, communication, training and transparency in the SaaS architecture, infrastructure, operations, and the deployment itself are critical for a successful SaaS solution, especially in the case of single-tenant SaaS solutions where client specific requirements are common.

In this thesis we identified that SaaS deployment differs from a typical (non-SaaS) software deployment and that SaaS projects have several characteristics that should be addressed when selecting a suitable PMM. The most important characteristics of the PMM were identified to be flexibility on requirements engineering and change management, iterative working, transparent and frequent communication and client involvement, increased collaboration and frequent demoing of the results. Additionally, multiple key considerations are brought up which should be taken into account when executing and managing a SaaS deployment.

Keywords cloud computing, SaaS, software deployment, project management, project management methodology

Aalto-yliopisto
Insinööritieteiden
korkeakoulu

Aalto-yliopisto, PL 11000, 00076 AALTO
www.aalto.fi
Diplomityön tiivistelmä

Tekijä Ida Korhonen

Työn nimi Projektinhallinta metodologiat SaaS ratkaisujen käyttöönottoprojekteissa

Koulutusohjelma Kiinteistötalous

Pää-/sivuaine Kiinteistöjohtaminen **Koodi** M3003

Työn valvoja Professori Seppo Junnila

Työn ohjaaja(t) Professori Karlos Artto, KTM Mikko Kallio

Päivämäärä 20.05.2015 **Sivumäärä** 66 + 10 **Kieli** Englanti

Tämä diplomityö tutkii projektinhallinta metodologien käyttöä SaaS (Software as a service) ratkaisujen käyttöönottoprojekteissa. SaaS on pilvipalvelumalli, jonka suosio on merkittävästi kasvanut viime vuosina. Projektinhallinta metodologia sisältää tärkeitä ohjeita, työkaluja ja käytäntöjä, joita voidaan hyödyntää projektien hallinnassa ja toimituksessa. Tässä työssä tutkitaan, mitkä metodologioiden ominaisuudet ovat sopivia SaaS käyttöönottoprojekteissa. Työn tavoitteena on löytää tehokkaimmat projektinhallinnalliset ohjeet, työkalut ja käytännöt SaaS projektien onnistuneeseen käyttöönottoon. Lisäksi työ tutkii SaaS:n käyttöönottomallien ja arkkitehtuurityyppien eroavaisuuksia sekä niiden tyypillisiä ominaisuuksia, jotta niiden vaikutukset metodologian valintaan ja hyödyntämiseen voidaan tunnistaa. Lopuksi työssä tutkitaan vaikuttaako tyypillinen SaaS käyttöönottoprojekti merkittävästi metodologioihin sekä niiden sisältämiin ohjeisiin, työkaluhin ja käytäntöihin.

Diplomityön tutkimusmetodologiana käytetään tapaustutkimusta, jossa tutkitaan yhden organisaation kahta SaaS esimerkkiprojektia. Tutkimuksessa haastatellaan neljää esimerkkitapausten projektipäällikköä sekä tehdään kysely yhdeksälle muulle esimerkkitapausten projektin jäsenelle. Diplomityön kirjoittaja toimi itse projektipäällikön roolissa esimerkkitapauksissa, mikä helpotti projektien tutkimista sekä dokumentaation analyysiä.

Diplomityö osoittaa, ettei SaaS:lle ole yksiselitteistä määritelmää, koska markkinoilla on lukuisia SaaS ratkaisuja, joilla on eri tyyppisiä arkkitehtuureja ja kokoonpanoja. Tyypillisesti, kun projektissa käytetään uutta teknologiaa, menetelmiä tai työkaluja tiedon puute kasvattaa väärinkäsitysten mahdollisuutta projektissa. Tämä pätee myös SaaS projekteihin. Huolellinen projektien suunnittelu, kommunikaatio, koulutus sekä läpinäkyvyys arkkitehtuurissa, infrastruktuurissa, tuessa ja itse asennuksessa ovat kaikki menestystekijöitä onnistuneessa SaaS käyttöönotossa. Tämä pätee erityisesti single-tenant SaaS ratkaisuihin, joka sisältää tyypillisesti räätälöityjä ominaisuuksia. Tässä diplomityössä tunnistimme, että SaaS:n käyttöönotto eroaa tyypillisestä (ei-SaaS) ohjelmistoprojektista. SaaS hankkeissa on useita ominaisuuksia, jotka täytyy huomioida sopivaa projektihallintamenetelmää valittaessa. Tärkeimmät ominaisuudet ovat joustavuus vaatimusten ja muutosten hallinnassa, iteratiivinen työskentely, läpinäkyvä ja säännöllinen kommunikaatio, tiivis yhteistyö ja jatkuva tulosten esittely projektin aikana. Lisäksi huomioimme tärkeät asiat liittyen SaaS ratkaisun ylläpitoon.

Avainsanat pilvipalvelut, SaaS, ohjelmiston käyttöönotto, projektinhallinta, projektinhallinta metodologia

Acknowledgements

This thesis was my last school project before graduation. Many sleepless nights it took, but here it finally is.

I would like to thank all of the people who helped me during this project. Especially, I want to thank my instructors Karlos Artto and Mikko Kallio, who patiently guided me through this process. Also, big thanks to my supervisor Seppo Junnila.

I want to thank my family, friends, and colleagues who were there for me when I needed the most.

For my dearest, I am grateful for all the support and love I have received.

Ida Korhonen
Espoo, Otaniemi, 25.05.2015

Contents

List of Abbreviations

AIPM	The Australian Institute of Project Management
API	Application Interface
APM	Association for Project Management
ASP	Application Service Provider
B2B	Business to Business marketing
CRM	Customer Relationship Management
DaaS	Development, Database, Desktop as a Service
DEV	Development
DSDM	Dynamic Systems Development Model
ERP	Enterprise Resource Planning
FaaS	Framework as a Service
FCC	The Federal Communications Commission (independent agency of the United States government)
FDD	Feature Driven Development
HaaS	Hardware as a Service
HTTP	Hypertext Transfer Protocol
IaaS	Infrastructure as a Service
IPMA	International Project Management Association
ISO	International Organization for Standardization
IT	Information Technology
ITIL	Information Technology Infrastructure Library
JAD	Joint Application Development
JSON	JavaScript Object Notation
LD	Lean Development
PaaS	Platform as a Service
PERT	Program Evaluation and Review Technique

PRINCE2	Projects IN Controlled Environments
PM	Project Manager
PMBoK	Association for Project Management Body of Knowledge
PMI	Project Management Institute
PMM	Project Management Methodology
PMO	Project Management Officer
PROD	Production
PTM	Project Team Member
RAD	Rapid Application Development
REST	Representational State Transfer
ROI	Return on Investment
RUP	Rational Unified Process
SaaS	Software as a Service
SB	Sandbox
SCM	Selling Chain Management
SDLC	Software Development Life Cycle
SIP	SaaS Integration Platform
SIT	System Integration Testing
SLA	Service Level Agreement
SOA	Service oriented architecture
SOAP	Simple Object Access Protocol
ST	Steering Team
UAT	User Acceptance Testing
WBS	Work Breakdown Structure
XaaS	Everything as a Service
XP	Extreme Programming

1 Introduction

1.1 Background and motivation

Our needs and behavior related to software usage have changed. One of the key changes is that the willingness for longtime investments and commitments is reduced. Solution needs to be available on-demand, anytime, and anywhere regardless of the used device. People are willing to collaborate and be social but at the same time the security and privacy concerns are addressed. And finally, change itself is a very important factor. No matter what the solution is, it needs to adjust to changes which in today's world are unavoidable.

These changes are reflecting to enterprises as well. Companies are willing to purchase more flexible, lightweight, accessible, on-demand, social, collaborative, and business need specific solutions instead of heavyweight and standard software systems which often require longtime investments with heavy installations and maintenance work. The emerging trends include computing utility services, remote working, online services, outsourcing, considering economic aspects and green IT. Additionally, phenomena such as Internet of Things and Industrial Internet are enabling everything to be connected to the internet and to each other forming an interactive network. (Pallis, 2010; Armbrust et al. 2009; Yang et al. 2013.) The term cloud has become familiar for people as services are digitalized and connected to the Internet. Cloud as an information technology related term is commonly known but the definition of cloud is not easy to find. Term cloud is also often used when people actually refer to the Internet. So what is cloud?

Cloud computing is a relatively new software service model which offers solutions to these changing needs and behavior, emerging trends, and phenomena (Pallis, 2010; Armbrust et al. 2009). Software as a Service (SaaS) is the most known service model of cloud computing that is becoming very popular on the enterprise software market. In the SaaS model, clients are able to use the software through a browser provided by a software vendor without the need to manage and maintain any hardware (Buyya et al. 2008; Furht & Escalante, 2010; Yang et al. 2013; Benlian et al. 2010.) SaaS entered the software market in mid 2000s and onwards the popularity has been rapidly increasing (Mäkilä et al. 2010). It has been forecasted that by the year 2016 12% of the worldwide software market will be moved to the cloud (Subashini et al. 2011). This is a significant change that is not only changing the software business but also the processes of both the enterprises adopting and the companies providing cloud services. Due to mentioned reasons the SaaS projects can be seen as important drivers of change (Papke-Shields et al. 2010).

Software deployment is a process covering all post-production activities which are needed in order for a customer to start using the software. The process activities vary based on the deployed product. The deployment is an important part of the system development life cycle which aims to deliver the developed product to the end users. (Dearle, 2007; Carzaniga, 1997.) This thesis focuses on SaaS application deployments executed as projects. Further, the focus is on enterprise SaaS application deployments in where a company purchases both the SaaS application and the deployment from the vendor. Enterprise SaaS deployments are extensive projects including business processes (Velte et al. 2010) with ranges of hundreds to thousands of users. In addition, the deployment usually requires configuration and customization work,

data transferring and integrations which increase the complexity of the project. (Mäntylä et al. 2011.)

Project management methodology (PMM) is a set or a combination of guidelines, practices, and tools which are used to successfully manage a project execution (Chin et al. 2012). It is highly important to identify the special characteristics of each project, so that the most suitable PMM and the project management methods can be identified and utilized. Already at this point the project can go wrong, if the choice of a PMM is not carefully considered. Additionally, all of the SaaS deployment related benefits may not be fully achieved if the project management is not suitable for the project type.

As SaaS is a rather new phenomenon of the software business, the subject is not widely researched. (Buyya et al. 2008; Yang et al. 2103). Additionally, software deployments are commonly seen as a part of the system development lifecycle (Pendharkhar et al. 2008) and therefore the amount of the deployment specific research is also quite limited. The field of project management is widely researched, but there is a lack of knowledge on the usage of project management methods and their success in practice (Papke-Shiedls et al. 2010). The lack of previous research and the importance of the subject makes the management of an enterprise SaaS deployment highly interesting topic to study.

In order to form a comprehensive outlook of the SaaS deployment project management in general, a literature review and an empirical case study research are conducted. The findings of this study will provide assistance and recommendations how to manage the execution of a SaaS deployment project. Especially in the continuously developing software business, it is highly important to be up to date in terms of skills and the toolset in order to succeed in the business and win competition between the other service providers.

First, a literature review is conducted with a purpose to define SaaS, software deployment process, software project management, and project management methodologies. Secondly, an empirical case study is conducted in order to investigate how are SaaS deployment projects managed and validate the typical SaaS deployment project characteristics identified in the literature research. The purpose is also to investigate how the PMMs are used in the deployment project and how effective and practical they are in the real life projects. Finally, the findings of the literature review and the case study are compared and discussed in chapter 5.

1.2 Objectives and scope of the study

The main objective of this study is to identify the characteristics of project management methodologies (PMMs) that are useful and effective in managing SaaS deployment projects. The information for the main objective is combined from a literature review and an empirical case study.

In order to answer to this question, first, the most popular service type of cloud computing, Software as a Service and its deployment process are defined and discussed. Next, software project management and project management methodologies are presented in general. Finally, the usage of PMMs in SaaS deployment project management is studied and discussed in order

to identify what should be taken into consideration when selecting and utilizing the PMM, and which of its characteristics are most important in managing the deployment.

In the empirical part, two case projects from a single case organization were selected to be used as the basis for the case study. These projects are SaaS deployment projects that the case organization has implemented or currently is implementing for one client organization. Generally this study focuses on examination of the project management of SaaS application deployment projects and the focus is on the vendor's deployment project activities. The empirical part is conducted by interviewing case organization's project managers and by conducting a survey for the project team members such as architects, testers, developers, technical, and business consultants. In addition, the author of this thesis has been working on the case projects from May 2014 which have enabled observations and investigation of the project related documentation.

1.3 Research problem and questions

Aim of this study is to present the usability and the applicability of PMMs in SaaS projects. The following research problem is appointed to this study:

> *What kind of project management methodologies are most suitable for managing SaaS deployment projects?*

Is further divided to two research questions:

1. *What are special characteristics of a SaaS deployment project?*

2. *What should be taken into account in project management activities, and what project management practices, tools, and methods should be implemented when managing a SaaS deployment project?*

1.4 Literature review and empirical analysis

The case study research was started before the literature study but they were completed as parallel activities. The intention of the literature study was to understand which characteristics of the project management methodologies are important in managing a SaaS deployment project.

As discovered, SaaS is a new emerging trend in the software business (Buyya et al. 2008) and therefore finding material for the study was occasionally challenging. In addition, the available literature of software deployment related research was quite limited. Instead, the field of project management is widely researched (Papke-Shields et al. 2010) and the author was able to collect large amount of subject related literature. The data for the literature study was collected from various sources e.g. articles, dissertations, academic journals, and academic studies which all related to cloud computing, project management field including methodologies and SaaS deployment projects. In addition, the literature review material is complemented with various reliable internet sources. The usage of the internet sources is justified due to the fact that the most current and the majority of the SaaS related information is available in the internet.

The research approach for this thesis is a case study consisting of two case projects executed by a single case organization. The selected cases are all deployment projects of a SaaS application and were selected for this study because the profiles of the cases matched the objectives of the study. The data for this research is collected by interviewing four case project managers and by conducting a survey for nine case project team members. Furthermore, the author of this thesis works as a project manager in the case projects which helped to observe the projects and analyze project related documentation.

1.5 Thesis outline

The research structure of this is study is divided into five main sections as follows:

1. Introduction
2. Literature review
3. Research methodology
4. Empirical research
5. Conclusions and discussion.

2 Literature review

This chapter presents the literature review of the thesis. Aim of the literature review is to identify the most important characteristics of a project management methodology in managing Software as a Service (SaaS) deployment projects. First, the chapter begins with section 2.1 providing definitions and presenting the special characteristics of SaaS. After this, the chapter continues with section 2.2, which studies the software deployment process and the typical features of a SaaS deployment process. Next, in section 2.3 the software project management field and its common practices are presented and discussed. Fourth, the project management methodologies and their usage in software projects are studied in section 2.4. Finally, section 2.5 summarizes and analyzes the findings from the literature review.

2.1 Software as a Service

This section focuses on studying a Software as a Service (SaaS) and its deployment process. The section starts with providing an overview of cloud computing and its most popular service model, Software as a Service. After this, SaaS and its typical characteristics are more thoroughly presented and defined. Next, the software deployment process, specifically a SaaS deployment process is presented and discussed in order to understand what requirements SaaS deployment sets to project management activities.

2.1.1 Overview of Cloud computing

Cloud computing, is currently one of the biggest trend of software business. It is a new software delivery model in where a cloud vendor provides ubiquitous and on-demand network access to a selection of configurable and customizable computing resources (Buyya et al. 2008). These resources can be e.g. networks, servers, storage space, applications, and services which can be easily and rapidly provisioned and released with minimal management effort or service provider interaction. (Yang et al. 2013.) In addition, through distributed computing cloud providers are able to provide processing power for the clients (Buyya et al. 2008.)

The provisioned cloud resources are accessible anywhere, anytime, and with various devices. Basically the user should only need a working network access in order to use the resource. This model brings many opportunities and reduces the dependency on owning and investing on IT resources. (Rao et al. 2011; Velte et al. 2010.) Cloud computing is a rather new diversified research field and the evolution and convergence of several independent computing trends and paradigms such as internet delivery, utility computing, scalability, virtualization, grid computing, peer to peer computing, distributed computing, storage, content outsourcing, security, and Web 2.0. (Pallis, 2010; Buyya et al. 2008.) Additionally, due to the characteristics mentioned above cloud computing can be seen as one of the key enablers for phenomena such as Internet of Things and Industrial Internet in where the main idea is everything connected and available in internet.

Cloud computing services are offered in various models from which the most known model is SaaS (Software as a Service). Other popular service models are IaaS /HaaS (Infrastructure /Hardware as a Service) and PaaS (Platform as a Service) (Mell & Grance, 2011), and less popular FaaS (Framework as a service) and DaaS (Development, Database, Desktop as a service). (Baran, 2008; Yang et al. 2013.) Baran also (2008) presents a term XaaS (Everything

as a Service) which covers all cloud services. These service models have common features such as the ease of provisioning and deployment, wide scalability options, multi-tenancy, and accessibility options easily via various devices. (Velte et al. 2010; Pallis, 2010.) The general layered architecture of these cloud services is presented in figure 1.

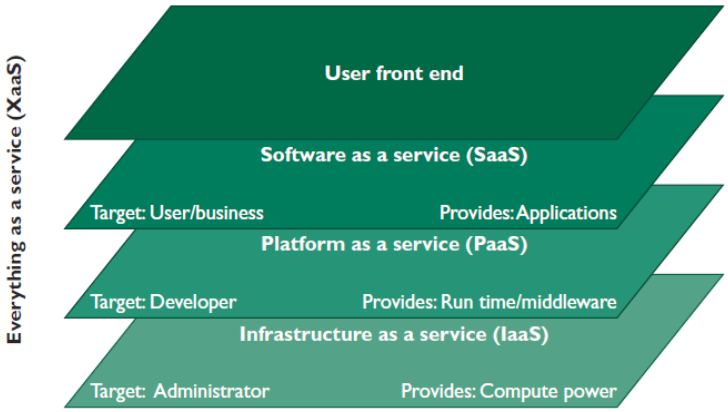

Figure 1 A general layered architecture of cloud infrastructures (Pallis, 2010).

A standard cloud solution, presented in figure 2, consists of several components that commonly include one or more clients, datacenters and distributed servers. Clients are various devices used by an end user who is willing to utilize the cloud computing resources. These devices can be e.g. computers, laptops, tablets, or mobile phones and they usually are categorized in three main types: mobile, thin and thick clients. Mobile clients are mobile devices, such as smartphones and tablets, whereas thin clients are computers with no internal permanent data storage. Thick clients are computers using a web browser and fixed internet access to connect to the cloud. (Velte et al. 2010.)

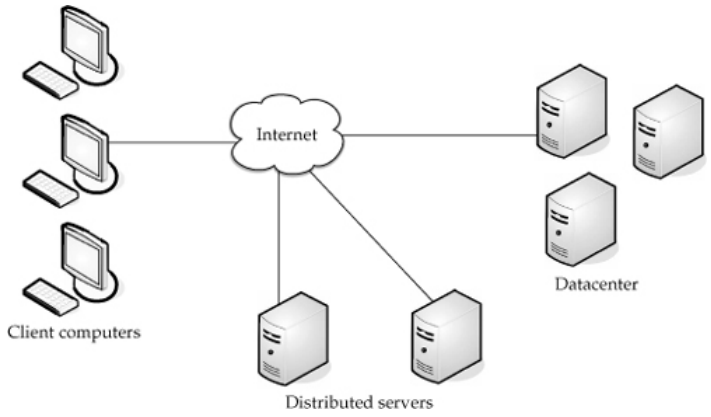

Figure 2 Standard cloud solution components (Velte et al. 2010).

2.1.2 Overview of Software as a Service

The most known delivery model of cloud computing commonly called Software as a service (SaaS) is a concept where a customer is able to use various cloud based software components in real time offered by the cloud vendors over the internet (Yang et al. 2013; Mäkilä et al. 2010; Furht & Escalante, 2010). After the purchase or payment of the usage fee the cloud components are easily accessible from various devices through either a thin client interface, such as a web browser (e.g. web-based email), or by a dedicated client program interface. Ideally, the customer simply needs to log in and start the usage of the software without a need to manage or control any of the cloud infrastructure components or individual software capabilities, except for possible adjustment of limited user-specific application configuration settings. (Mell & Grance, 2011; Velte et al. 2010; Yang et al. 2013.) SaaS can be seen as an on-demand solution that is replacing traditional software usage by reducing the need for the users to own and host hardware themselves (Furht & Escalante, 2010; Benlian et al. (2011) present an idea that SaaS can be considered as on-demand outsourcing in where a client sources software applications externally from their respective vendors.

Even though SaaS is replacing traditional software, many of the software products are still offered either in traditional on-premise model or in SaaS model. Whether the client should choose SaaS or on-premise model is always depended on the current situation and needs of the client. By comparing these two models, multiple differences can be identified which may bring assistance for the selection. The fast provisioning and deployment, accessibility, and economical benefits are usually considered as the key differences when it comes to choosing between SaaS and on-premise options. (Mell & Grance, 2011; Velte et al. 2010; Yang et al. 2013) Other key differences are related to licensing, location, and managing. For on-premise software the client typically purchases licenses, with an up-front cost or an owned outright whereas SaaS software are often licensed based on the usage. On-premise applications are installed onsite to client's premises whereas SaaS applications are remotely installed at the vendor's premises. Also, the application management and maintenance work with an on-premise application is done by the client IT department whereas the SaaS applications are entirely managed and maintained by the cloud vendor. (Carraro & Chong, 2006.)

According to Velte et al. (2010) SaaS and Service oriented architecture, commonly shortened as SOA, are quite similar in nature, due to the common feature of both being based on a services model. Whereas, Buyya et al. (2008) see SOA more as an enabler of cloud computing and SaaS. Additionally, SaaS is also compared to other on-demand and traditional internet based deployment models such as Application Service Provider (ASP). The key difference of SaaS is that it can be seen as a standardized solution and does not usually require purchase of licenses, whereas ASP and other internet based deployment models are used for tailored software and do require purchase of licenses. (Benlian et al. 2011; Mäkilä et al. 2010.) In a SaaS solution all customers are using a single instance of the same software base which is called multi-tenancy, whereas in ASP model each customer has their own private and isolated instances of the software (Buyya et al. 2008; Mäkilä et al. 2010; Carraro & Chong, 2006). Usually in case of an enterprise SaaS solution the client is also able to choose a single-tenant option in where the client is using an own software base. According to Mäkilä et al. (2010) many of the benefits of the cloud solution are achieved by the multi-tenancy architecture which enables scalability,

reduces operational and maintenance costs and increases the version upgrading. However, Mäkilä et al. (2010) also state the same benefits could possibly be achieved by spawning a new instance of the software for each client automatically instead of serving the clients from the same instance of the software.

2.1.3 Typical characteristics of SaaS

In this section the definitions and typical characteristics of SaaS are being studied and discussed in order to understand the deployment process of SaaS and how does it differ from a traditional software deployment process. Additionally, aim is to identify what should be considered when executing the SaaS deployment.

Definitions

According to Mäkilä et al. (2010) there is no single and generally accepted definition of the SaaS concept. However, typically five distinct characteristics associated to the definition of a SaaS solution can be identified:

1. SaaS solution is commonly used through a web browser

2. SaaS solution is a standardized solution and it is not tailor made for a specific customer

3. SaaS solution does not require installation of software on the customer premises

4. SaaS solution does not require special integration or installation work

5. SaaS solution's pricing model is based on the actual usage of the service i.e. on the pay-as-you-use model.

In addition to the listed characteristics above, the multi-tenancy architecture (Buyya et al. 2008; Mäkilä et al. 2010) is included in many SaaS definitions. The multi-tenant architecture is commonly considered as a technical solution to a business problem, not being an essential part of the SaaS business model itself. As described in previous section, in the multi-tenant architecture all customers are using a single instance of the same software base and the single instance of the common code and data definitions are on the vendor's server. (Benlian et al. 2011.) The common code cannot be customized which makes SaaS a standardized solution and reduces the possibilities of making client specific customizations and configurations. If customizations and configurations are required, they need to be implemented at the meta-data layer on top of this common code by using the interfaces that the SaaS vendor provides. (Carraro & Chong, 2006; Mäkilä et al. 2010; Benlian et al. 2011.) In a single-tenant option the client has an own software base which allows more customization and configuration to be done. The multi-tenant solution is commonly considered as a "pure" SaaS solution due to all of the cloud benefits can primarily be achieved with a multi-tenant architecture. (Carraro & Chong, 2006.)

Benefits and concerns related to SaaS

SaaS offers multiple benefits for both the clients and the vendors. Additionally, some concerns related to SaaS solutions have been addressed. The setup of the SaaS solution e.g. selected deployment type and the architecture does have an impact to the benefits and concerns. For

example by choosing a private deployment type, the client may not achieve all economical and flexibility benefits of cloud but at the same time may avoid some of the SaaS related concerns. In the following paragraphs the benefits and concerns related to SaaS are presented and discussed.

For the client SaaS enables the use of an application or a piece of software provided by the service provider. When it comes to identifying benefits of SaaS, the economical perspective is usually first to be examined. Generally, it is stated that the implementation, infrastructural, running, and personnel costs are reduced in SaaS model. (Velte et al. 2010; Armbrust et al. 2009; Yang et al. 2013.) The reduced costs are resultant of the characteristics of SaaS which are presented in the following paragraphs.

Due to SaaS being a standardized solution, the provisioning of is easy and fast. The installation is straightforward and usually no complex implementation is needed (Armbrust et al. 2009). The solution is also easy to take into use which means that the applications are up and running faster, with improved manageability and less maintenance work. (Velte et al. 2010). In today's highly competitive market, this shortened time to market period is a significant benefit which may increase future income while also reducing implementation and license costs. The SaaS related costs are usually based on the usage and there usually there is no upfront cost, however there is a lot of variance between application types and vendors (Carraro & Chong, 2006). Also, the version upgrades are provided to the clients automatically without any additional costs or needed installations (Armbrust et al. 2009).

One key benefit of Saas is accessibility, since users can access the services anywhere anytime with various devices (Yang et al. 2013). Additionally, the requirements for the end user´s device are decreased due to cloud services are light software and generally used via browser. Therefore, the need of disk space and device features is reduced which allows the use of light-weight mobile devices. (Velte et al. 2010.) Saas also enables easy data sharing and collaboration over internet (Armbrust et al. 2009) which is convenient for telecommuters and traveling remote workers, who can simply log in and use their applications anywhere (Velte et al. 2010). Such factors are becoming more and more relevant in the future as business is global and working is independent of time and place. One important benefit for company is scalability. This means that the company has the ability to easily provide or release resources. (Buyya et al. 2008). Enterprises can flexibly adjust resource types and capacity according to current needs on demand (Pallis, 2010) without the need to make long time commitments and investments. There has been discussion among clients that the cloud providers should be able to provide cloud resources as utilities. In this model, clients could purchase cloud utilities on-demand like other utilities (e.g. electricity). (Buyya et al. 2008.)

It can be concluded that more internal IT resources are in company's effective use because managing and maintaining the infrastructure is outsourced. (Velte et al. 2010.) In addition, managing and maintaining hardware and software resources is usually considered as a less pleasant work, so the motivation of these freed human resources will most likely also increase the general working moral.

For the vendors SaaS model opens up a possibility for creation of continuous stream of income and provides stronger protection of intellectual property. (Velte et al. 2010.) The first vendors adapting cloud into their offerings were large, well-known companies such as Amazon, Google, Microsoft, IBM, and Yahoo! which makes the purchase of cloud solutions quite a safe option. (Velte et al. 2010.)

Probably the most addressed issue of cloud solutions are security related concerns. (Benlian et al. 2011.) The vendors need to enforce customers' trust in them to ensure the security of their data. A knowledgeable vendor might be a significant factor for many clients when they are choosing whether the cloud is the best option for them or not. (Buyya et al. 2008.) Many of these vendors have strict privacy policies along with efficient security measures (e.g. cryptographic user authentication) and clients can use data encryption before storing the data in the cloud. With these methods the data may be more secure than if it were stored in company's own facilities and systems. (Armbrust et al. 2009; Velte et al. 2010.) Another security related point of view is that the company data would be even more secured since it is not physically located in one specific datacenter. This excludes the risk of data loss in case the datacenter is somehow damaged or dysfunctional.

Companies are concerned of the information security and privacy as their confidential data is managed somewhere, by someone in internet. This is a justified concern taking into account as nowadays security breaches are rather common, however the breaches are definitely not only limited to cloud solutions. (Yang et al. 2013; Benlian et al. 2011; Armbrust et al. 2009.) In order to enhance the security of clients´ data the cloud provider needs to ensure that even though the applications are hosted simultaneously for multiple users the data will be isolated and secured from each other (Buyya et al. 2008). Vendors also have to take into account differences in customers´ national legislation (Armbrust et al. 2009).

Cloud services are provided over internet which makes them dependent on an online internet access. This raises a reliability risk of a lack of a stable access to service. (Benlian et al. 2011.) However, it needs to be mentioned that nowadays there are extremely few work related activities an employee can perform without internet access. Therefore, it is fair to say that cloud solutions are occasionally unfairly pointed out due to this concern. This concern of reliability is universal since all business activity nowadays needs internet access, so this is not only a problem of cloud computing. (Armbrust et al. 2009; Yang et al. 2013.)

Even though there is a wide supply of cloud vendors and solutions, the clients may face vendor lock-in situations i.e. get stuck with vendors and their systems due to interoperable solutions. This of course is an issue for the current phenomena like Internet of Things, Industrial Internet and Big Data, which are based on the collaboration of systems and intelligent usage and sharing of data. (Pallis, 2010; Armbrust et al. 2009.)

The data lock-in might bring challenges in extracting data from cloud, but there are also issues with importing data to cloud through integrations with existing apps. The challenges with integrations are caused by the situation where the other application is hosted and located on-site and the other application lies somewhere in the cloud. In this situation security, performance and reliability issues may rise. (Velte et al. 2010.) Compatibility with other applications may

not always be sufficient and the implementation of integration is complex. This naturally requires a lot of effort and decreases one of the main benefits of cloud, which is easiness of taking the solution quickly into use.

Concern about the transparency of the cloud provider and the quality level of services have become important especially for clients who are adapting enterprise SaaS services. These concerns are addressed by Service Level Agreements (SLA) where the delivered services with terms and conditions are presented and agreed by both parties. (Buyya et al. 2008.) Performance level issues may occur in situations where e.g. the usage load of the cloud service varies substantially which leads to unpredictable performance of the service (Armbrust et al. 2009). Additionally, the data transfer and latency can present challenges (Yang et al. 2013).

One of the most difficult challenges in Cloud Computing is removing errors in these very large scale distributed systems. A common occurrence is that these errors cannot be reproduced in smaller configurations, so the debugging must occur at scale in the production datacenters. (Armbrust et al. 2009.)

The SaaS client may have to pay more to reach the expected level of service than initially anticipated the so-called “hidden costs”. SaaS’s architectural approach shifts specific investments to the client. The vendor, for example, does not customize the common code or data definitions on its servers, and the client is responsible for maintaining all the customized components. (Benlian et al. 2010.)

One of the greatest risks for the customer is strategic of nature. There is a risk that a company will lose critical resources and capabilities when sourcing applications via SaaS. This holds specifically true if business-critical applications and those that support a broad spectrum of key functional areas within an organization, including ERP, SCM or CRM systems, are outsourced. (Benlian et al. 2010.)

Unpredictable developments in the area of security may also cause challenges in the future. SaaS clients give the service vendor direct control of their data and of valuable assets without knowing exactly how this vendor will secure the data and which backup and disaster recovery procedures are in place. Service level agreements (SLA) can be used to indicate the exact data security levels that should be maintained. (Benlian et al. 2010; Subashini & Kavitha, 2011.)

2.1.4 SaaS Services

The SaaS market has a wide range of solutions provided by cloud vendors for companies to acquire. The variety of SaaS offerings is constantly increasing among the popularity and demand which are pushing the vendors to constantly provide more solutions. (Velte et al. 2010.) These solutions include SaaS applications, SaaS platforms, third-party SaaS add-ons, and SaaS integration tools (Hai & Sakoda, 2009). As vendors are pushed to enter the SaaS market many of the vendors are providing software that is technically SaaS, but do not fulfill the “pure” SaaS based business models features. (Mäkilä et al. 2010.) As discussed previously in section 2.1.2, in the "pure" form of SaaS, the software is a multi-tenant solution which is hosted and managed by the provider (Carraro & Chong, 2006).

The majority of SaaS services provided are applications which can be divided in two categories: Enterprise/Line of Business services and Customer-oriented services. (Velte et al. 2010; Yang et al. 2013.) Enterprise/Line of business services are business solutions which are sold via a subscription service to companies and enterprises. These business services include business processes, like supply-chain management applications, customer relations applications, and other similar business-oriented tools. One well-known line of business SaaS application example is Salesforce.com which is used for customer relationship management. (Yang et al. 2013.) Customer-oriented services are offered to the general public on a subscription basis, commonly offered for free and supported by advertising. Customer-oriented services include e.g. web mail services, online gaming, and consumer banking, among others. Examples of customer-oriented SaaS services are the web-based email services which are offered by Microsoft (Hotmail), Google (Gmail), and Yahoo! (Yahoo Mail). (Velte et al. 2010.)

The next wave in the evolution of SaaS is the emergence of so called SaaS integration platforms (SIP): a platform that enables the combination of different SaaS services and thus create a new combined solution. This is called the third wave in software adoption in which SaaS evolves beyond standalone applications and becomes a comprehensive platform. Companies offering SIPs include Zoho, Sutisoft, Salesforce, Microsoft and Oracle. (West, 2010.)

Some of the best known global SaaS vendors are Cisco, Google, IBM, Amazon, Facebook, and Salesforce.com. Currently over 50 percent of the global SaaS market is shared between Salesforce.com, WebEx, and RightNow Technologies. (Competitive Insights, 2006; Tekrati, 2006.)

The Global SaaS market

SaaS model started to became popular around year 2005 (Mäkila et al. 2010) and onwards the popularity has been rapidly increasing. Nowadays, most of the people have heard about SaaS and also have used some SaaS solution even they might not have noticed it being delivered as SaaS. At the time it was forecasted that 10 percent of the enterprise software market would move to pure SaaS model by the year 2009. This 10 percent growth was not fully achieved, but the SaaS industry was growing at 40-50 percent rate annually. (Gens, 2008.)

It has been forecasted that by the year 2016 the global cloud computing market will grow to USD 95 billion and that 12% of the worldwide software market will be moved to the cloud (Subashini & Kavitha, 2011). According to Gartner, 16% of all enterprise software applications were delivered through the SaaS model by 2014 and SaaS revenue is expected to reach USD 22.1 billion by end of year 2015 as more companies are starting to invest into cloud. As can be seen in figure 3 according to Forrester Research the global SaaS software revenue is forecasted to reach USD 106 billion in 2016, increasing 21% over projected 2015 revenue levels. Forrester also forecasts that the global market size will continue to see strong growth and estimates the size of global SaaS market to reach USD 132 billion by the year 2020. (Kanaracus, 2015.)

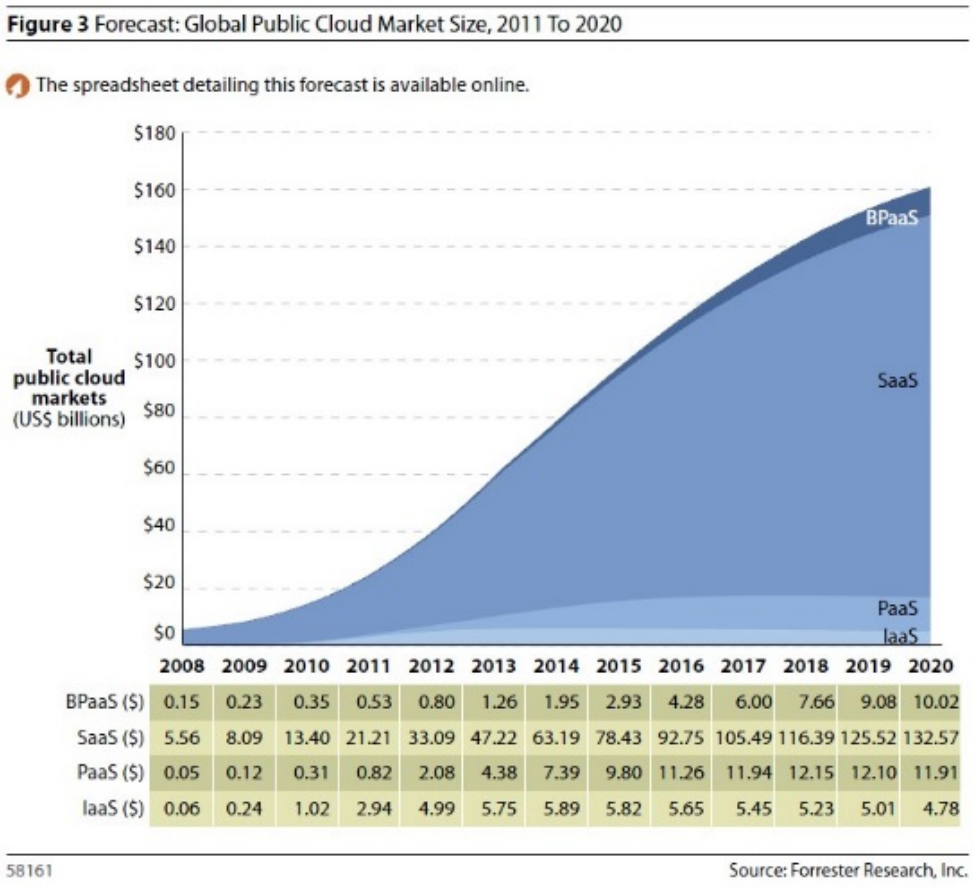

Figure 3 Global Public Cloud Market (Kanaracus, 2015).

2.2 SaaS Deployment Process

The purpose of this section is to study what is a SaaS deployment process and how to consider it as a project. First, software deployment process and its phases and activities are presented in general. After this, the special characteristics of a SaaS deployment process are discussed and the differences compared to a traditional software deployment process are studied. Finally, the execution of a SaaS deployment as a project is studied in order to identify suitable project management methods for the execution.

2.2.1 Software Deployment process

Software development life cycle (SDLC) is a term covering all activities of developing the software from initial planning to its retirement. SDLC is used in software engineering to describe a process for planning, creating, testing, and deploying a software system. Core activities of SDLC include requirements engineering, solution design, building, testing, debugging, deployment, and maintenance which are presented in figure 4. (Normand, 2015.)

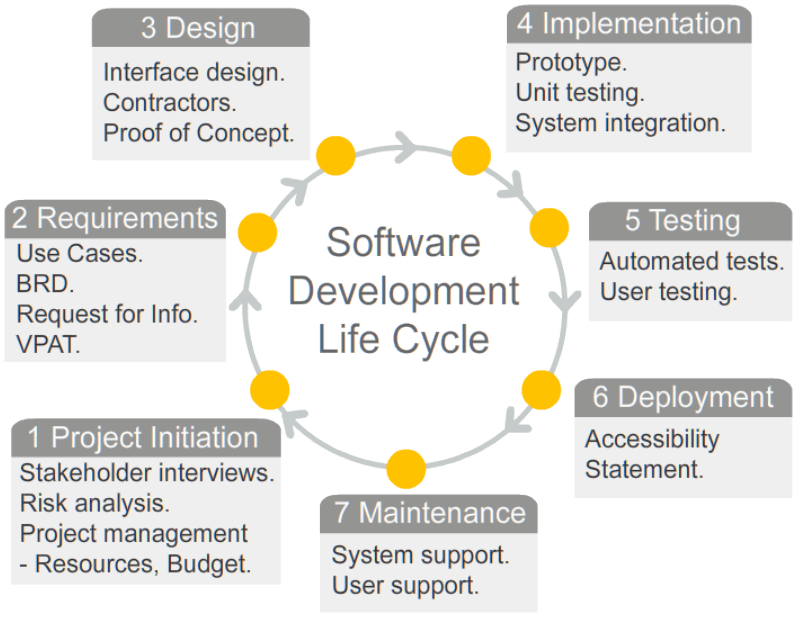

Figure 4 Software Development Lifecycle (Normand, 2015).

Software deployment is one of the core activities of SDLC. It is a complex process which covers all the needed activities in order to start using some pre-produced product (Carzaniga, 1997). The deployment is commonly called as delivery pipeline, consisting of multiple environments which usually are Development (DEV), System Integration Testing (SIT), User Acceptance Testing (UAT) and Production (PROD). The software is deployed through the environments starting from the lowest DEV finally to PROD in order to increase the quality of the product before it reaches the users. (Minick et al. 2006; UCISA, 2015; Carzaniga, 1997.)

Software deployment process covers all the activities and processes between the purchase and the execution. These activities include the release of the developed software, customer specific configuration and customization work, installation, and finally the activation of the software. After the installation, additional activities might be conducted which vary depending on the deployment situation and the software. These activities include e.g. monitoring, deactivation, version upgrading, reconfiguration, adaptation, and uninstalling. The objective of the deployment is always a software system ready to be used. The deployment activities are done after production and depending on the situation they can be performed by a service provider or by a customer. (Dearle, 2007; Carzaniga, 1997; Minick et al. 2006; UCISA, 2015.) Mäntylä & Vanhanen (2011) define the software deployment as a complex process, including customer interaction, integrations, configuring, installing, and testing the solution. This definition slightly differs from the previous definitions for example adding the testing activities as a part of the deployment process and highlighting the customer interaction.

All the customer specific requirements, functionalities, integrations, customizations and configurations for the pre-produced software are completed during the deployment process (Carzaniga, 1997; Mäntylä & Vanhanen, 2011.) Therefore, especially for a standardized solution, such as SaaS application, the deployment is a critical process where the customer

specific business requirements are fulfilled (Mäntylä & Vanhanen, 2011). According to Mäntylä & Vanhanen (2011) three process characteristics adding complexity to the deployment are existing integrations between clients' other systems, various configuration options, and a requirement for a complex pre-created data model. These characteristics are common for an enterprise SaaS deployments which makes the process more complex than a standard SaaS deployment. Every software contains its own specific characteristics which makes the deployment process unique. The deployment should be considered as a general process that will be customized based on the software to be deployed and the client specific requirements and characteristics.

2.2.2 Deployment process phases and activities

Every software deployment process phase contains activities which are needed to provide a software system to the end users. These phases and the activities defined by Carzaniga (1997) and Dearle (2007) are briefly presented in the following paragraphs and in the figure 5 below.

Release

After the software development process the product is ready to be released. In the release phase all operations in order to prepare a system for assembly and transfer to the customer site are performed. In addition, it is important to identify needed resources needed for operation at the customer site and planning of later phases of the deployment process.

Install and activate

In the activation phase, the software is activated usually with an execution command. More complex systems might require additional activation operations and they require confirmation that all supporting systems are ready to use.

In large software deployment processes the usage of multiple environments is common. This means that the vendor installs the working copy of the software on a production environment whereas other versions may be installed on different servers to a test environment, development environment, and disaster recovery environment.

Deactivate

Compared to activation, deactivation is the opposite action. In deactivation the executing components of a system are shut down. Usually before being able to perform deployment of a new software/version some deactivation operations are required. System deactivation is commonly called as application retirement or application decommissioning.

Adapt

In the adaptation phase the installed software is being modified. The difference between adaption and updating is that this phase is initiated by local events such as changing the environment of customer site, whereas updating is usually executed by an external software vendor.

Update / Version upgrade

As mentioned above, updating is a process where an earlier version of a software or part of it is replaced with a newer release.

Built-in updates: is a common term for system mechanisms that are checking new updates and installing them automatically. The process can be fully automated or the updating might require user initiation and can be controlled by the user. Usually these updates at least ask for user's approval before the installation is started.

As a new version of the software is released the version tracking system helps users to notice and install these updates on PCs and local networks. The version tracking can be web based, local or browser based depending on the software and client's requirements.

Uninstall

The inverse phase of installation is naturally uninstallation of a software. In this phase the software is removed and it usually also involves some reconfiguration of other software systems in order to remove the uninstalled system's files and dependencies.

Obsolescence

As the software is outdated and its support has been discontinued, the product life cycle becomes to an end and the software is being taken out of use.

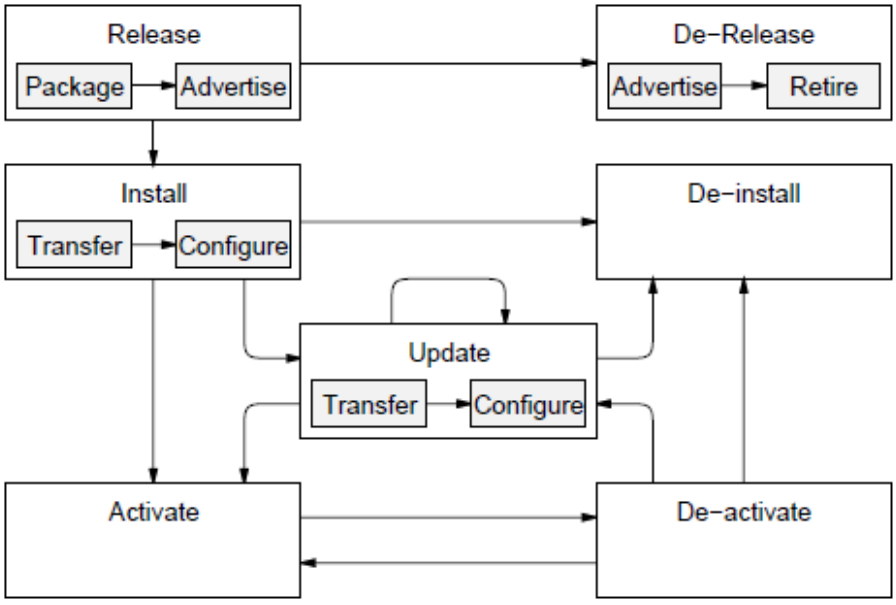

Figure 5 Software deployment process phases and activities (Carzaniga, 1997).

2.2.3 SaaS Deployment process

As mentioned above, every software contains its own specific characteristics and the deployment process should be customized based on the software to be deployed. This section presents the typical characteristics of a SaaS deployment process and studies how it differs from a traditional software deployment process. Some of these characteristics were partly presented already in section 2.1.2 but the important characteristics from the deployment perspective are revisited in the following section in order to study how the deployment should be managed.

The deployment of a SaaS application process covers all the post-production activities which are needed to deliver the application to the end users (Carzaniga, 1997). A key difference between traditional on premise software deployment process is that deployment of a SaaS application does not require onsite activities and can be centrally deployed in the cloud. Other major difference is that the delivery of the needed cloud components and the actual deployment activities are usually executed as separate projects. Additionally, the deployment has similarity with utility service provisioning and therefore the provisioning and deploying a SaaS application is easier and faster compared to on-premise software deployments.

Multi-tenancy reduces configuration and customization possibilities which reduces the scope and complexity of the process (Benlian et al. 2010). Enterprise SaaS application deployments are usually large processes, with ranges of hundreds to thousands of employees. In addition, the deployment usually requires configuration and customization work, data transferring, integrations, and possibly some custom development which increase the complexity of the deployment (Mäkilä et al. 2010).

SaaS deployment offers multiple benefits for both vendor and client organization. As mentioned, it should be noticed that the deployment type and the architectural decisions affects which of these benefits can be fully achieved. One of the main benefits is cost reduction of the delivery which means savings for both the vendor and the client. SaaS solutions are accessible anytime, anywhere, they are scalable, (Young et al. 2013.) For the client SaaS model reduces also the IT infrastructure and department related costs due to the services are hosted and maintained by the vendor (Carraro & Chong, 2006).

SaaS deployment reduces the risk of vendor and data-lock in which are common cloud service related risk (Velte et al. 2010). One of the benefits of SaaS solution is version upgrading. New versions of the solution are provided to clients as soon as they are released. Additionally, many of the SaaS solutions are updated frequently due to the vendor hosts and maintains the solution and therefore can perform development and testing activities more freely than in case of a standard software. (Carraro & Chong, 2006.)

Nowadays, the social aspect is getting more and more important even in the corporate world. One of the recognized benefits of SaaS is the offering of social and collaborative functionalities. The collaboration between users is only possible with centrally hosted software, which makes SaaS applications stand out in the market. In addition, it is important to notice that the social and collaborative features will be even more important in the future as the employees are increasingly born in the 1990s and later and used to utilize the social functionalities. (Armburst et al. 2009; Pallis, 2010; Tapscott, 2008.) In addition, Green IT is one of growing current market

trends and SaaS is seen as an excellent environmental choice from the ecological point of view. (Velte et al. 2010.)

The big trend is that focus in software projects in general is moving more and more into the deployment phase and customization or "tuning" of the software platform as opposite to the previous situation where the focus was more in the planning and development phases.

SaaS Deployment types

There are multiple deployment type options for SaaS solutions. The choice of the deployment type affects the deployment process and can be seen as one of the characteristics which distinguishes the SaaS deployment from a traditional deployment process. Additionally, the deployment type has an impact on the typical characteristics of SaaS. Some of the identified benefits may not be fully achieved with all deployment types whereas some of the SaaS related concerns may be avoided. The choice of most suitable deployment type is made based on the current situation and needs of a client organization. According to Mell & Grance (2011) three common options for deploying the solution are public, private, and hybrid cloud.

Public cloud

Public cloud is a cloud computing deployment type where the service provider offers resources to the general public over the internet via web applications and web services (Rao et al. 2011). Most commonly offered resources are applications and storage solutions (Mell & Grance, 2011).

Public cloud services may be provided as free of charge, but there are also chargeable services in the market. The chargeable solutions are usually offered as a pay-per-usage services and the customer is charged by the amount of use. (Rao et al. 2011.) Additionally, several of the free services have chargeable components or functionalities offering more features for the customer. Some service providers also offer a possibility for customers to purchase a direct also known as a private connection to the public cloud service. The private connection comes with more availabilities to influence on the customization and configuration aspects. (Gens, 2008.) In public cloud model the infrastructure is owned and operated by the service provider at their data center and the customer only needs a device and an access to internet. All of the customers share this same infrastructure pool and usually have only minor opportunities to configure and customize the solution or influence on security and availability related topics. (Velte et al. 2010; Buyya et al. 2008.)

Benefits of the public cloud are mainly achieved by the volume of the users and the shared infrastructure. Public cloud provides an easy, on-demand, scalable and cost effective services (Rao et al. 2011). On the other hand due to the shared infrastructure, public cloud solutions are more vulnerable than private cloud solutions and therefore many enterprises are choosing private over public cloud. Public cloud solution may be the right choice in situations where e.g. an application is frequently used by many people, application code needs to be tested, incremental capacity is needed or collaboration projects are being executed. (Yang et al. 2013.) Examples of well-known public clouds include Amazon Elastic Compute Cloud (EC2), IBM's Blue Cloud, Sun Cloud, Google AppEngine and Windows Azure Services Platform. (Buyya et al. 2008.)

Private cloud

From a technical point of view public and private deployment types may only have minor or no difference in cloud architecture, but the security related topics vary. If the situation with public cloud is that the services are available for the public and communication is done over a non-trusted network. (Foley, 2008; Velte et al. 2010.) Private cloud differs from the public cloud by the operation of the cloud infrastructure. In a private cloud the infrastructure is dedicated to an organization so that the data and processes are managed within the organization. This reduces the security risks, legal requirements and network bandwidth restrictions. (Rao et al. 2011.) There are two variations of private clouds. The infrastructure can be managed and hosted internally or by a third-party (Mell & Grance, 2011). If the private cloud is implemented right it can improve business processes by offering elastic and service based solutions to the client (Rao et al. 2011).

The downside of the deployment of a private cloud is that it requires significant investments from the purchaser. Even though the services are cloud services there is still a need to invest on space, hardware and environmental control which also need to be periodically refreshed. In addition the possible security issues need to be carefully considered in order to be avoided. (Foley, 2008) Due to these reasons the private cloud solutions have faced criticism from the customers since it basically prevents achieving the beneficial economic model of cloud computing (Foley, 2008).

Companies have confidential data and processes. If these valuable assets are exposed to public it may cause tremendous damage to business and reduce the competitiveness. Therefore as mentioned, many companies choose private cloud solutions instead of public cloud. Private cloud is a suitable option when there is a need for data sovereignty, demand for consistency across services, more server capacity than needed, need for data center efficient improvement, or willingness to provide private cloud services. (Velte et al. 2010; Yang et al. 2013.)

Hybrid cloud

Hybrid cloud is basically a cloud environment which includes a mix of internal and external resources, combining public and private cloud components (Rao et al. 2011). Hybrid cloud requires on-premise and off-site infrastructure resources in order to form a comprehensive solution. One main advantage of hybrid cloud is that it offers multiple deployment benefits by spreading data and processes in various places. (Yang et al. 2013.) For example the advantage of public cloud's economic model is reached while the business confidential data can be processed safely in control on the private cloud. (Rao et al. 2011.) A downside for the model is that occasionally due to data and processes are managed in various places the tracking and the communication between the solutions can be difficult (Velte et al. 2010).

Hybrid cloud is an obvious and the best deployment solution in many cases. As mentioned before, organizations are willing to keep their confidential data and processes under their own control. In addition, many companies already own the needed resources for private cloud and these resources need to be utilized efficiently. (Rao et al. 2011.) These two reasons are usually the main drivers for companies to choose a hybrid cloud. Other examples for choosing hybrid clouds are situations where an organization is keen on purchasing SaaS but is concerned about

security and where a company is offering services that are tailored for different vertical markets. Organization can use a public cloud to interact with the clients but keep their own data secured within an internal private cloud. (Yang et al, 2013.)

Other deployment models

Other used deployment types are community cloud, distributed cloud, intercloud and multicloud which are briefly presented in the following sections.

Community cloud

Community cloud is a service model where a targeted group of users have a form of private cloud built and operated specifically for fulfilling their common requirements (Yang et al. 2013). In this model the cloud infrastructure is shared between these user organizations and managed internally or by a third-party, and it might be hosted internally or externally (Mell & Grance, 2011).

As with hybrid cloud, with community cloud the organizations can achieve both private and public cloud benefits. However, some benefits of the public cloud are not fully achieved due to the amount of the infrastructure users is less than in public cloud services (Mell & Grance, 2011). The main benefit and goal is that the targeted user group achieves most of the benefits of a public cloud with private cloud characteristics such as enhanced privacy, security, and policy features. (Yang et al. 2013; Velte et al. 2010.)

Situations where community cloud is a preferred deployment option are e.g. situations where government organizations need to share resources, group of healthcare facilities need a private HIPAA or group of telecommunication companies need to meet FCC regulations. (Yang et al. 2013.)

Distributed cloud

In a distributed cloud the computing resources are provided to a client by cloud data centers that are connected to a single network or hub service but are located in different places (Cunsolo et al. 2009).

Intercloud

The idea of an intercloud is based on the internet “networks of networks” and it can be seen as an interconnected global “clouds of clouds”. The main focus in this deployment type lies in the tight operation between public cloud service providers (Bernstein et al. 2009).

Multicloud

In a multicloud the client is combining and using multiple cloud services with same type of architecture. This is done mainly due to reduce reliance (lock-in) on specific vendors, to increase flexibility and scalability, and to reduce the possible risks by decentralizing the data and processes. The difference between multicloud and hybrid cloud is that multicloud uses multiple cloud services instead of multiple deployment types (King, 2014).

Executing the deployment

The deployment of a SaaS application is usually executed as a project. The deployment of a SaaS solution is a unique project which characteristics are combination of typical characteristics of SaaS, the specific product to be deployed, and the specific deployment process. Generally, a project is defined as a unique set of coordinated and controlled activities designed to produce a unique product, service or result (PMBOK). Project is a temporary one time effort with a defined beginning and an end. The project aims to produce and reach customer-specific deliverables and milestones with a specific level of criteria. (Morris, 2004.) In addition, like for any project there is an amount of money and resources available to execute the project in a given timeframe (Lester et al. 2014). This naturally highlights the importance of managing the execution efficiently.

A SaaS deployment is an IT project containing e.g. software, training, communications, conversion and deployment. (Taylor, 2004.) However deployment of a software to end users is not considered as an IT project in the traditional sense due to SaaS being a standardized solution, and commercialized as an easy-to-instantly take in to use solution and does not primarily contain software development (PMBOK). Further, deployment of a SaaS application is not categorized as a basic deployment process due to the installation of Saas is easy and usually no complex implementation is needed (Armbrust et al, 2009; Velte et al. 2010). SaaS applications are also faster to get up and running and easy to take into use (Velte et al. 2010; Buyya et al. 2008; Yang et al. 2013). The unique result of the deployment to be produced is a SaaS application ready to be used. The beginning and the end are set based on the provisioning date and when is the application moved to production / go-live date.

Generally, a project usually consist of five phases which are: initiation, planning and design, execution and construction, monitoring and controlling systems, and completion (Wysocki, 2014). As presented earlier in section 2.2.2 the traditional software deployment process phases differs from these phases. When the deployment is executed as a separate project and not as an automated deployment the applicable project phases need to be adjusted and planned in the initiation phase.

2.3 Software Project Management

After studying the typical characteristics of the deployment process of SaaS application and its execution as a project the suitable project management activities can be discussed. This section aims first to present the field of software project management and project management methodologies in general. After this, the suitable project management activities for a SaaS deployment project can be studied with a purpose to identify how to manage SaaS deployment project utilizing a suitable PMM and which of the characteristics of the PMM are most important for the project success.

2.3.1 Project management in general

The field of software project management is extensive. This thesis focuses on studying the software deployment process which is executed as a project. Therefore, in the following sections, the project management and project management methodologies are presented in general with a purpose to identify most suitable methods for managing the deployment.

In general, project management is defined as the process and activity of planning, organizing, motivating, and controlling resources, procedures and protocols to achieve the project objectives (Hill, 2010; Charvat, 2003) Traditionally, project management activities include four to five process groups which are usually used regardless of the project management methodology or terminology. These major process groups generally include:

1. Initiation
2. Planning or design
3. Production or execution
4. Monitoring and controlling
5. Closing (Hill, 2010; Wysocki, 2014.)

IT projects differ from other project types and therefore must have a unique set of project management tools and techniques to accomplish them successfully (Taylor, 2004). Usually the characteristics and requirements for the project management in an IT project are more complex and unpredictable (Papke-Shields et al. 2010.) In addition, requirements clarification, communication between stakeholders, high risk and uncertainty, constantly developing technologies, and interfaces to other software increases the challenges in IT project management (PMBOK). Many studies show that overall 80–90 percent of software projects do not fully reach the original objectives of projects (Taylor, 2004; Papke-Shields et al. 2010.) This may partly be resultant of the fact that the project management literature has not evolved the same pace as IT has. Therefore project managers are forced to apply the generic PM tools to IT project management with little or no success. Currently, most of the project management related material only covers the software development process of the IT system and do not offer specific guidance for example to the deployment phase. (Velte et al. 2010; Hill 2010.)

Project management standards

There is no comprehensive documented guide or manual available for a project management activities but there are many globally known project management standards developed attempting to provide the best practices in the area of project management (Papke-Shields et al. 2010). These standards have many commonalities which can be considered as best practices. Some examples of the project management standards are developed by the Project Management Institute (PMI), the Association for Project Management (APM), the Australian Institute of Project Management (AIPM), and the International Project Management Association (IPMA). Additionally, International Organization for Standardization's standards (ISOs), Projects IN Controlled Environments (PRINCE2), and Association for Project Management Body of Knowledge (PMBoK) are offering guidance for project management among the standards. (Papke-Shields et al. 2010; Lester, 2014.)

2.3.2 Deployment management

Deployment management includes building, testing, and delivering activities of a software service. Deployment management is not considered same as managing a software deployment project, instead it describes more the deployment and release management of an automated version upgrading process. UCISA ITIL (2015) describes the scope of release and deployment management includes the processes, systems and functions to package, build, test and deploy a

release into operation. Additionally, the goal of release and deployment management is to deploy releases into customers' usage and deliver value by ensuring the effective use of the service. The deployment management also makes sure that the support for the new service is in place e.g. handover to service operations and end user training are conducted, and solution design documentation is available. (Minick et al. 2006)

According to UCISA ITIL (2015) and Minick et al. (2006) the purpose of release and deployment management is to define and agree deployment plans with stakeholders, ensure that the content, maintenance, tracking and testing, change management, risk management, knowledge transfers and handovers to support related to each release and deployment packages are in place and conducted according to agreements and plans with the client. Additionally, they describe the deployment management has a lot benefits which include reducing costs and risks, delivering the changes fast and accordingly to client's needs, and improving the consistency of implementation.

2.3.3 Project management activities

The main responsibility of managing the project execution lies on the IT project manager (PM), who is responsible for analyzing the nature of the project and selecting the suitable project management methods by consulting other PMs and team members. Therefore the PM should be aware of different and most current project management methods. (PMBOK.) In addition, in larger projects the project management unit is commonly complemented with project management officer (PMO) providing assistance for the leading PM. Together the PM and PMO form a project management body for the project. Usually, there is also a sponsor or a partner appointed to every project who gives guidance and support in project management and steers the project throughout its lifecycle. (Hill, 2010; Velte et al. 2010.)

According to Hill (2010) the primary activities of the project managers include:

- Project initiation activities.
- Project team recruiting and managing.
- Plan and manage the project effort (cost, schedule, and resource utilization).
- Project stakeholder management.
- Contract execution and management.
- Risk management.
- Project deliverable quality and acceptance management.
- Project status and performance results reporting
- Project closing activities management.

2.4 Project Management Methodologies in managing SaaS deployment projects

Project management methodologies (PMM) are commonly used to enhance the project management. As learned earlier, IT projects are complex in nature and the managing is challenging (Papke-Shields et al. 2010). Therefore, a comprehensive and coherent management process is needed and the usage of a PMM is recommended. There are many definitions for a

project management methodology. PMBoK (4th Edition, glossary, page 437) defines the PMM as a system of practices, techniques, procedures, and rules used by those who work in a discipline. According to Chin et al. (2012) PMM is a combination of standards, articles, literature references whereas Cockburn (2000) describes the methodology as a set of project management elements such as practices, tools, and techniques. Goff comprehensively (2008) defines the PMM as a set of appropriate repeatable processes that help introduce consistency, flexibility, and efficiency while improving quality in managing projects. Additionally, Attarazadeh & Ow (2008) states a PMM typically consists of process descriptions, templates, roles and responsibilities, Life Cycles and Work Breakdown Structures, together with other support information.

Project management methods, tools, and practices

Most of the definitions describe the PMM as a set or a combination of guidelines, practices, and tools for managing the implementation of a project. These various project management tools, methods, techniques, guidelines, books, and processes were identified to be essential for project success already in the early 1900s (Taylor, 2004 & Morrison, 2004). As mentioned, based on the project's nature and features a project manager chooses the most suitable tools to be utilized in the execution (Hill et al. 2010; Velte et al. 2010). The most utilized and popular project management methods are Gantt Charts, PERT (Program Evaluation and Review Technique), Critical Path Method, Network Analysis and Earned Value, Scheduling techniques, Organizational issues, and Conflict management. These important tools are commonly seen as the foundation of project management regardless of the project type. However, the software development techniques are constantly improving, which forces these tools to be evolved as well. (Taylor, 2004; Charvat, 2003; Papke-Shields, 2010; Wysocki, 2014.)

Project management methodologies provide guidance, tools, and methods for project managers, key stakeholders, and team members throughout the lifecycle of a project. By identifying and utilizing a suitable methodology, project managers have a possibility to successfully execute the project and achieve the wanted results. In addition, the PMM unifies the project work which makes the collaboration between team members much easier. (Goff, 2008; Charvat, 2003; Attarzadeh & Ow, 2008.) Nevertheless, it must be noticed that the use of a methodology does not automatically produce project success. (Charvat, 2003.) A general assumption among project managers is that the usage of commonly accepted project management practices will enhance project performance. This assumption is supported by the evidence from previous research in the project management field which indicate that the usage of project management practices will improve the project performance. However, it has been stated that the evidence is limited and more research is needed in the specific area. (Papke-Shields et al 2010.)

Even though a suitable PMM is identified and taken in to use, the project manager still needs to have the capabilities to change and adjust the PMM according to the nature of the project and the current situation. It can be stated that none of the PMMs fits perfectly in any project or situation. In addition, almost in every project changes will occur and the PMM needs to be adjusted to these varying circumstances. (Attarzadeh & Ow 2008; Wysocki, 2014.) In project management field, there are a number of different approaches and methods for managing different kinds of project. According to Charvat (2003) a generic project methodology contains

at least nine basic elements which are: roles, skills, activities, techniques, tools, teams, deliverables, standards and quality measures. (Goff, 2008.)

The project management methodologies can be categorized into traditional and modern approaches. (Wysocki, 2014; Attarzadeh & Ow; 2008.) For a software project, commonly used methodologies are Waterfall (Traditional), Adaptive Project Framework, Agile Software Development, Crystal Methods, Dynamic Systems Development Model (DSDM), Extreme Programming (XP), Feature Driven Development (FDD), Information Technology Infrastructure Library (ITIL), Joint Application Development (JAD), Lean Development (LD), PRINCE2, Rapid Application Development (RAD), Rational Unified Process (RUP), Scrum, and Spiral methodology (Charvat, 2003; Goff, 2008; Wysocki, 2014; Attarzadeh & Ow, 2008.) All of these methodologies fall under the common traditional versus modern categorization.

2.4.1 Traditional Approaches

PMBOK (2010) defines the traditional project management as a set of techniques and tools that can be applied to an activity looking for an end product, outcomes or a service. A traditional approach includes phases to design, develop and deliver a product or service. These phases are to be completed step-by-step which reduces the flexibility and possibilities to changes. Additionally, it might cause delays in the project due to the next phase is not able to start before closing the previous. Most used traditional methodologies for software projects are Waterfall and RUP. However, it has been stated that traditional project management approach is not a perfect fit for a software development project due to the lack of flexibility and transparency. (Charvat, 2003; Goff, 2008; Wysocki, 2014; Attarzadeh & Ow, 2008.)

The following phases are included in the traditional project management methodology:

1. Initiation (requirements specification)
2. Planning and design
3. Execution (construction and coding)
4. Control and integration
5. Validation (testing and debugging)
6. Closure (installation and maintenance) (Charvat, 2003; Wysocki, 2014.)

These phases are not applicable for every project as shown in the figure 6. The applicability depends on the project type and the execution for example some projects do not have a structured planning or monitoring process, projects can be terminated before they reach completion, and occasionally some phases may be repeated multiple times.

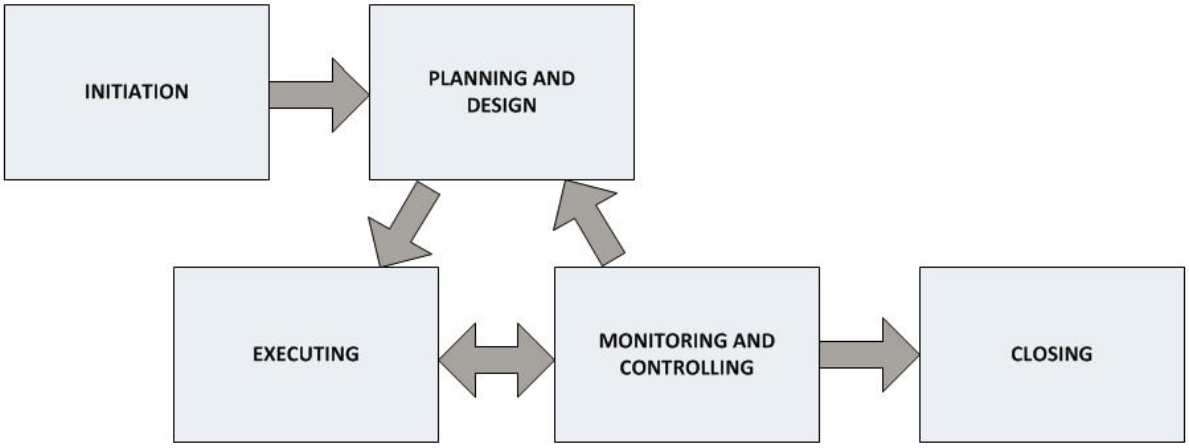

Figure 6 Typical development phases of a traditional project.

2.4.2 Modern Approaches

Contrary to traditional approach, modern methodologies do not focus on linear processes. Instead, they are incremental learning methodologies which enables fast reactions to changes. These methodologies have only a few rules, practices, and documents. Additionally, compared to traditional approaches the modern approach highlights constant customer involvement and communication. Modern approaches are identified to be successful in IT software projects due to their flexible, adaptive, customer involvement, and transparent nature. Most used modern project management approaches used for software projects are Extreme Programming (XP), Scrum, Crystal methodology, Dynamic Systems Development Methodology (DSDM), Rapid Application Development (RAD), Adaptive software development, Lean development, and Feature-driven development. (Charvat, 2003; Wysocki, 2014; Goff, 2008; Attarzadeh & Ow, 2008; Schwaber et al. 2007.)

2.4.3 Selecting a Methodology

There are no exact guidelines for choosing a suitable methodology for a specific project but many recommendations for the selection are given. Generally it has been advised that large and complex projects require more formalized and heavy weight PM methodologies to control the execution. As the number of stakeholders arises, the heavier methodology is needed due to the increasing need for coordination, communication, and guidelines in order to the project to keep on track. Papke-Shields et al. 2010.) Additionally, project specific features such as industry, objectives, duration, and the budget are important to be taken into account. According to Attarzadeh & Ow (2008) important features to be considered while choosing a methodology are:

- Budget
- Team size
- Used technology

- Tools and techniques
- Project criticality
- Existing processes.

As mentioned, Charvat (2003) defines that a generic project methodology contains at least nine basic element which are roles, skills, activities, techniques, tools, teams, deliverables, standards, and quality measures. For the PMM selection he identifies ten general requirements e.g. budget, team and project size, project criticality, technology used, documentation, training, best practices, tools and techniques, examination of existing processes, and software.

2.5 Literature review findings

Aim of the literature review was to identify the most important characteristics of a project management methodology in managing Software as a Service (SaaS) deployment projects. This section presents a summary and analysis the findings from the literature review.

2.5.1 Software as a Service (SaaS)

SaaS is a new software delivery model which has become very popular during last years and it is replacing standard software among enterprises (Subashini & Kavitha, 2011). The SaaS market has a wide range of solutions which include SaaS applications, SaaS platforms, third-party SaaS add-ons, and SaaS integration tools (Hai et al. 2009). The majority of the provided SaaS services provided are applications which can be divided in two categories: Enterprise/Line of Business services and Customer-oriented services. (Velte et al. 2010; Yang et al. 2013.) SaaS is not widely researched due to it being a rather new software model (Yang et al. 2013). There is no single and generally accepted definition of the SaaS concept but five common characteristics included most of the definitions are:

1. SaaS solution is commonly used through a web browser

2. SaaS solution is a standardized solution and it is not tailor made for a specific customer

3. SaaS solution does not require installation of software on the customer premises

4. SaaS solution does not require special integration or installation work

5. SaaS solution's pricing model is based on the actual usage of the service i.e. on the pay-as-you-use model (Mäkilä et al. 2010).

Additionally, multi-tenant architecture is included in most of the definitions. The multi-tenancy makes SaaS as a standardized solution and reduces the customization and configuration possibilities. In a multi-tenant architecture all of the customers are using a single instance of the same software base and the single instance of the common code and data definitions are on the vendor's server. (Mäkilä et al., 2010; Benlian et al. 2011.) Other architectural option for cloud solution is a single-tenant architecture which increases the customization and configuration possibilities.

Many software are offered in SaaS and also in traditional on-premise model. Main differences between these two options are related to costs, accessibility, provisioning and scalability, light implementation project, licenses, managing and hosting, and location aspects. These differences are also seen as the major benefits of SaaS compared to the on-premise model. (Velte et al. 2010; Yang et al. 2013; Carraro & Chong, 2006.) For the client the main benefits of SaaS are the economic benefits and a flexible and scalable solution. For the vendor SaaS opens up new business possibilities and also opens up possibility for creation of continuous stream of income and provides stronger protection of intellectual property. (Velte et al. 2010.)

As mentioned, many SaaS related benefits but also concerns were identified from both client and vendor perspective. Most addressed issues of cloud solutions were security related and performance related concerns (Benlian et al. 2011; Buyya et al. 2008), dependency on internet, strategic risks (Benlian et al. 2011), and the vendor and data lock-in (Pallis, 2010; Armbrust et al. 2009). Important is to notice that choice between the deployment type and the multi versus single-tenant architecture have impact on the achieved benefits and possible concerns (Mäkilä et al., 2010; Benlian et al. 2011).

2.5.2 Deployment of Software as a Service

Software deployment is a complex process and one of the core activities of the SDLC (Software Development Life Cycle). Software deployment process covers all the activities and processes between the purchase and the execution which are needed in order to start using some pre-produced product. The common process phases and activities are: release, install and activate, deactivate, adapt, update / version upgrade, uninstall, and obsolescence. (Carzaniga, 1997; Dearle, 2007; Mäntylä & Vanhanen, 2011).

Usage of multiple environments is recommended in the deployment process. The software is deployed through the environments in order to increase the quality of the product before it reaches the users. (Carzaniga, 1997; Dearle, 2007; Mäntylä & Vanhanen, 2011; Minich et al. 2006; UCISA ITIL, 2015). All the customer specific requirements, functionalities, integrations, customizations and configurations are completed during the deployment process (Carzaniga, 1997; Mäntylä & Vanhanen, 2011.) which makes the process highly important, especially for a standardized solution, such as SaaS application. Three process characteristics that are adding complexity to the deployment are existing integrations between clients' other systems, various configuration options, and a requirement for a complex pre-created data model which are common for an enterprise SaaS deployments (Mäntylä & Vanhanen; 2011). Finally, it was stated that every deployment process is unique due to the specific characteristics of the deployed software and the process activities should be adjusted to fit these characteristics.

Overall the standard software deployment process phases described, can be considered as applicable for a SaaS deployment but some differences and similarities of the activities can be identified. Additionally, these standard deployment phases and activities are more applicable in case of releasing and deploying a new version of already pre-deployed product to the end users.

The deployment of SaaS application has a lot of special characteristics, possible benefits and concerns which distinguishes the deployment from a standard on-premise process. The deployment of SaaS has similarity with utility service provisioning and therefore, in principle,

the provisioning and deployment of a SaaS application should be easier and faster compared to on-premise software deployments. Other major differences of SaaS deployment are the lack of required onsite activities, the enhanced version upgrading process, and the limited configuration and customization possibilities. (Mäkilä et al., 2010; Benlian et al. 2011; Velte et al. 2010; Yang et al. 2013; Carraro & Chong, 2006.)

Most important benefits from the client's perspective were identified to be the economic benefits i.e. reduced costs in deployment and in later managing and maintaining phases. Additionally, the fast provisioning, light and easy installation and deployment project, and rapid version upgrading. For the vendor, major benefits also include the reduced costs and efforts of delivery which enables the vendor to execute multiple projects same time and in shorter period. On the other hand the light installation and deployment process can be seen as a downside from the vendor's perspective due to the implementation projects are smaller from scope and contractual, and point of view.

As any cloud solution, SaaS can be deployed in many types from which three most common are public, private, and hybrid cloud. All the deployment models have own benefits and shortages and the choice need to be done by the client and based on the current needs. It should be noticed that the choice between multi-tenancy and single-tenancy and the deployment type affect the achieved benefits and possible occurred challenges.

In general software deployment process is considered as a part of the SDLC (Software Development Life Cycle). However, deployment of an SaaS service differs form a software deployment process and is commonly executed as a separate project. The deployment can be seen as an IT project containing e.g. software, training, communications, conversion and deployment. (Taylor, 2004.) However, the deployment of is not considered as an IT project in the traditional sense due to SaaS being a standardized solution, and commercialized as an easy-to-instantly take in to use solution and does not primarily contain software development (PMBOK, 2010).

2.5.3 Managing the deployment project

The deployment of a SaaS application is commonly executed as a separate IT project. In general the project management is a process and activity of planning, organizing, motivating, and controlling resources, procedures and protocols to achieve the project objectives (Hill et al. 2010; Charvat, 2009). Usually the project management activities are divided in five processes which are initiation, planning or design, production or execution, monitoring and controlling, closing (Hill et al. 2010). These phases differs quite significantly from the traditional software deployment process phases and therefore the management of a deployment project should be carefully planned and adjusted accordingly to the deployed product and the project specific features.

IT projects differ from other project types and therefore must have a unique set of project management tools and techniques for the management. IT projects are considered more complex and unpredictable in nature. Studies have shown that overall 80–90 percent of software projects do not fully reach the original objectives of projects. (Taylor, 2004; Papke-Shields et al. 2010;

PMBOK, 2010) Due to these reasons a comprehensive and coherent management process is needed and the usage of a PMM is recommended when managing IT projects.

Project management methodologies (PMM) are commonly used to enhance the project management. There are many definitions and many different methods available for a project management methodology but generally it is described as a set or a combination of guidelines, practices, and tools for managing the implementation of a project (PMBoK, 2010; Papke-Shields et al. 2010; Attarzadeh & Ow, 2008). The most utilized and popular project management methods are Gantt Charts, PERT (Program Evaluation and Review Technique), Critical Path Method, Network Analysis and Earned Value, Scheduling techniques, Organizational issues, and Conflict management. The project management methodologies can be categorized into traditional and modern approaches. (Goff, 2008; Attarzadeh & Ow, 2008; Wysocki, 2014.) There are no exact guidelines for choosing a suitable methodology for a specific project but many recommendations for the selection are given. Generally it has been advised that large and complex projects require more formalized and heavy weight PM methodologies to control the execution. Ten common requirements have been identified to be important while selecting a suitable PMM e.g. budget, team and project size, project criticality, technology used, documentation, training, best practices, tools and techniques, examination of existing processes, and software. (Charvat, 2003; Attarzadeh & Ow, 2008.)

From a SaaS deployment perspective the selected PMM should address and meet the requirements and characteristics of the solution and the deployment process. Also, the considerations should be taken into account while choosing the PMM.

If we address the ten common requirements identified to be important while selecting a suitable PMM for the SaaS deployment project, we see that for many of them are project specific features which cannot directly be answered. These include e.g. budget, team and project size, project criticality, documentation, training, best practices, tools and techniques, and examination of existing processes. The features which can at least partly be addressed are the used technology used and the deployed software.

We can see that the SaaS specific characteristics are important to be known when planning and executing the deployment project. These characteristics include SaaS being a new software model, the easiness of provisioning, high scalability and accessibility of the solution, standardized nature which reduces the customization and configuration work, no installation of software on the customer premises needed, and no special integration or installation work needed. Additionally, main concerns related to the SaaS deployment should also be considered while choosing the PMM. These include the security related and performance related concerns, dependency on internet, strategic risk, and the vendor and data lock-in. Important is to notice that choice between the deployment type and the multi versus single-tenant architecture have impact on the achieved benefits and possible concerns.

3 Research methodology

This chapter presents the research methods used in this thesis. First, in section 3.1, the used research approach is presented and discussed. Section 3.2 and 3.3, include justifications for the research method and introductions of the selected case organization and its projects. After this, the descriptions of the data collection and analysis methods are presented in sections 3.4 and 3.5. Finally, in section 3.6 the reliability and the validity of the research are analyzed and discussed.

3.1 Research approach

SaaS is a rather new phenomenon in the software business and therefore the previous research about the subject is quite limited (Buyya et al. 2008; Yang et al. 2103). Software deployments are commonly seen as a part of the system development lifecycle and therefore the amount of subject specific research is also quite limited. The field of project management is widely researched, but there has been statements that the usage of project management methods and their success in practice is not widely researched (Papke-Shiedls et al. 2010). This study extends the previous research in SaaS, software deployment, and project management fields.

The research approach for this thesis is a qualitative case study research which is complemented by observations conducted by the author. According to Yin (1994) a qualitative case study may be applied in situations where the existing definitions and theories do not provide unambiguous answer to appointed research questions. The case study approach enables to collect practical data from real life customer projects from various stakeholders. (Tellis 1997; Meyer, 2001) The collected empirical data was compared to the literature research findings.

The analysis of this study was done for one case organization and three projects. Commonly it is suggested that multiple case studies are included when a case study approach is used (Meyer, 2001). In order to ensure a comprehensive understanding of the subject Yin (1994) highlights the importance of using multiple data collection methods and sources in a single-case study. In order to gain a holistic picture of the subject, multiple sources of data were used for gathering the empirical data e.g. project team member interviews, surveys, and discussion. In addition, the author of this thesis was working in the case projects which allowed to conduct observations during a long time period. Observation period enabled author to have multiple discussions with the team members, review the project related documentation, take part on the project activities, organization's processes, and make several important observations of the research subject.

3.2 Case project selection

The case projects, hereafter called as case Alpha and Beta, were selected from one case organization. The case organization is one of the top vendors providing and implementing SaaS solutions. The selected cases are all deployment projects of a SaaS application and were selected for this study because the profiles of the cases matched the objectives of the study. In addition, the selected case projects are quite extensive and the examination of these projects offers valuable information to the organization from its project business. The major reason for choosing the organization was that the author is working in the company and in the case projects and therefore access to people and data was enabled. Also, this opened the possibility to conduct

observations which enhances the quality, amount, dimensions, and format of the collected empirical data.

The selected cases were executed to a single client and were all part of client's extensive Program. Even though from the Program's perspective the case projects were seen as Phases, the vendor executed them as separate projects. The case projects had differences in objectives, scope, project activities, management, project organization, duration, budget, execution situation, and results. Additionally, many lessons were learned in the first case, project Alpha which were taken into account in later cases. The selection of the two cases was done by the author with the help of colleagues within the company.

3.3 Case projects overview

As mentioned, all the three case projects were selected from the same case organization and were also executed to the same client. The case organization acts in Finland but some of the project team members were located in different countries. Additionally, majority of the project activities were conducted in Finland but occasionally workshops or meetings were held abroad. As some of the team members were not located in Finland, many of the meetings, communication, and project activities were done remotely. Some of the project team members, such as business and technical consultants were brought to Finland to attend project kick-offs, conduct analysis and design workshops, and configure the software. Both of the case projects, Alpha and Beta are currently ongoing. The timeline of the case projects is presented further in section 4.1.2, in figure 7.

3.4 Data collection

In all researches the amount, validity, and relevance of collected data plays a significant role. Yin (1994) has listed six major sources of evidence for data collection in the case study protocol: documentation, archival records, interviews, direct observation, participant observation, and physical artifacts. According to Yin, not all of these sources are needed to be used in every case study. In this study the used sources of evidence were documentation, interviews, direct and participant observations, and surveys. All the data was collected between 2014 -2015 from two case projects.

3.4.1 Interviews

The qualitative interviews were used to collect practical data and experimental information from the project managers in order to answer the research question. The major part of the data was collected via four interviews which were conducted in Finland in March - April 2015. The interviews were conducted in the case organization's office or remotely over the phone.

3.4.2 Survey

To complement and compare the collected data from the interviews the author developed a set of questions and conducted a survey for project members (PTMs). The survey contained questions regarding the characteristics of a SaaS deployment process and the suitability of utilized project management methods in the project. Additionally, the PTMs were asked to rate the identified used project management methods. Survey was answered by nine of the team

members. Survey respondents, the project team members, were selected from different roles backgrounds and organization levels in order to form a comprehensive overview of the project.

3.4.3 Observations and discussions

During the data collection phase there were no specific observations arranged due to the fact that author was working as a PM in the project. The working period started in May 2014 and is still ongoing at the present day.

3.5 Data analysis

After conducting the interviews, discussions, surveys, documents gathering, and observations the author had an extensive amount of material related to the research subject. In order to form a comprehensive understanding of the case projects the author started to explore the material carefully in a phased approach. First, the related documentation was reviewed and the observation findings were documented in plain text, tables, charts, and idea maps. Next, the author started to review the interview and survey results. After forming an outlook of the gathered data the author started an in-depth analysis of each data collection method material. The analysis of the material was done by carefully reviewing the material several times, making comparison between the materials, creating notes, and discussing about the findings with project the team members. In addition, based on the questions presented and answers of the respondents four themes were identified. These themes were used to form a base structure for the empirical part analysis and findings. Additionally, the themes allowed grouping the findings which helped to point out convergences and new information.

3.6 Validity, Reliability and Limitations of the Research

3.6.1 Validity

The empirical study consists of four interviews conducted during spring 2015, survey results from nine participants, observation period, and an analysis of project related documentation. The validity of the study results is based on these data collection methods and a careful analysis of the collected data.

3.6.2 Reliability and limitations

The main limitation of the research is the fact that the research is based on only one two case projects. Therefore it is uncertain whether the findings of this thesis are relevant if projected to another projects in the future. Also, as deployed solution was a single-tenant, instead of a multi-tenant solution all the SaaS related benefits and risks were not able to be identified based on the empirical research.

4 Empirical research

This chapter presents the findings of the studied case projects. First, the case organization and the studied case projects are introduced in section 4.1. After this, section 4.2 presents the empirical findings related to SaaS application deployment process and its management methods.

4.1 Case organization and projects

4.1.1 Introduction of the Case Organization

The studied projects in this thesis are deployments of a SaaS application executed by the case organization. The name of the company will be kept anonymous, as well as the client, and their employees. The respondents, project managers (PMs) and project team members (PTMs) are employees of the case organization, working in the case projects. The case organization is a large information technology company that has many subsidiaries in different countries. The case organization manufactures and markets computer hardware and software, and offers infrastructure, hosting and consulting services in many business areas. A significant amount of the case organization's business is project business which makes successful project management a highly important topic for the case organization. This has been recognized by the case organization and the project management processes are constantly being improved. Therefore the findings of this study will provide valuable information about the current project management situation, guidance for possible enhancements, and suggested methods to be used for future projects.

This study focuses on investigating the usage of project management methodologies in the deployment of the SaaS application. Therefore, the focus is on case organization's deployment process related activities and the activities related to the client are not presented and the projects will be hereafter called as cases Alpha and Beta. The data was collected via conducting interviews and surveys, gathering documents, and making observations. As this thesis concentrates to study the overall deployment of the SaaS application without specifying the product, the findings may be considered as applicable for a general SaaS deployment regardless of the business area. Additionally, as learned from the literature in chapter 2, every deployment process is unique and therefore the findings should be reviewed as general guidelines in the execution of the SaaS deployment project.

4.1.2 Case projects

The studied case projects are deployments of a SaaS application covering users and functionalities from three business streams with integrations to existing solutions, data transfer, and some custom development activities performed by the case organization. The case projects are called Alpha and Beta. The deployed application is the case organization's single-tenant SaaS application which was deployed in a public cloud model. The deployment projects are part of an extensive Program which consist of three phases. Alpha and Beta were scoped based on the first two phases. The case organization was delivering and executing the technological parts of the deployment and this study focuses on studying the case organization's project activities. Project timeline for case Alpha and Beta can be seen in figure 7.

The first project, case Alpha was launched in late 2013 with a scope comprising the configuration of standard application functionalities, enabling functionalities for the one

business stream, building two integrations, data transfer, change management support, and training activities. Case Alpha being the first project to be implemented was also the most extensive of the cases in terms of the scope, project team, and duration. This was due to that many of the standard application related functionalities and activities were done in case Alpha. The second project, case Beta was launched in spring 2014 and its scope consisted of two business streams, building five integrations, continuing the data transfer, conducting trainings, and making enhancements to the existing solution.

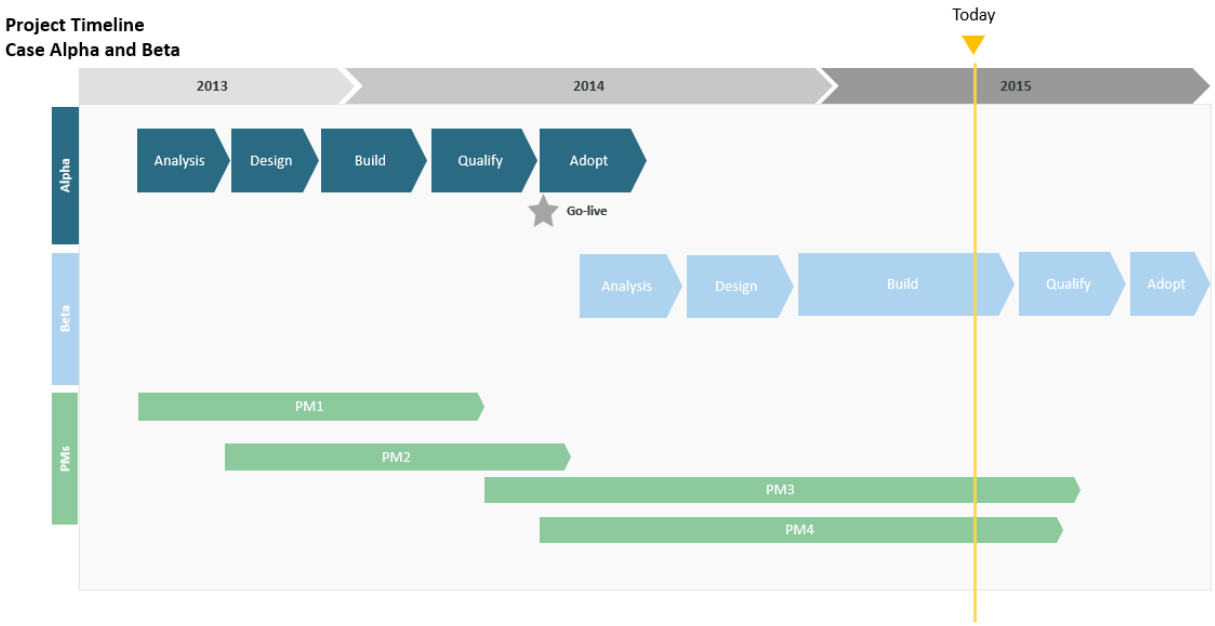

Figure 7 Project timeline, Case Alpha and Beta.

Project managers

Overall the case organization had four project managers working in cases Alpha and Beta. These project managers will be hereafter called as PM1, PM2, PM3, and PM4. The number of the PMs refers to the onboarding order. PM1 was the first one to enter the projects whereas PM4 has been the latest addition to the team.

Not all of the PMs were working in all of the cases. Figure 7 above presents the working periods of the project managers. In case Alpha, the work was started in December 2013 with leading PM1 and PM2 as a PMO. In the end of case Alpha, in spring 2014, the PM1 was handing the responsibilities over to PM3, who was about to complete Alpha and to start to lead the first phases of case Beta. Also the PM4 entered the case projects in spring 2014.

The author of this thesis (PM4) has been working in the case projects from May 2014 to the present day. During this time the author has gained a good knowledge level and valuable

experience of the case projects and project management in general. The author has had several discussions with all of the project managers and other team members. In addition, access to project related material and participation in project activities helped in forming an overall picture of the case projects and writing this thesis.

Project organization

The project's organizational structure was different in case Alpha and Beta. The project organization was formed based on the scope and the objectives. Generally the team roles and supporting functions included:

1. Business partner
2. Project manager (PM)
3. Project manager officer (PMO)
4. Lead architect
5. Developers
6. Testers
7. Technical consultants
8. Functional consultants
9. Change management consultants
10. CSM
11. Business partner
12. Account manager
13. Application support team
14. Financial department
15. Security assessment team
16. Legal department

Research approach

The research was conducted in two parts. First, the project managers' thoughts and experiences were studied in order to identify the special features of a SaaS deployment process and the suitability of the utilized project management methods in the project. All of the four project managers were selected to be interviewed. Next, a survey was conducted for other project team members. The survey contained questions regarding the characteristics of a SaaS deployment process and the suitability of utilized project management methods in the project. Additionally, the PTMs were asked to rate the used project management methods. Survey was answered by nine of the team members.

4.1.3 Case Alpha

Case Alpha was the first project executed by the case organization. The execution of both case projects was managed using case organization's traditional project management methodology which will be studied more in section 4.2. As mentioned above, case Alpha's scope comprised the configuration of standard application functionalities, enabling functionalities for one business stream, building two integrations, data transfer, change management support, and training activities.

Overall, the target was to complete the in-scope activities by the plan and create a good foundation for the following case Beta which made the scope wider. Also, many of the activities done in case Alpha were to be completed only once and the results could be utilized later in case Beta. According to the utilized PMM, case Alpha was executed in six different functions which were:

- Configuration / Customization without any code changes
- Integration
- Data Migration

These three functions above were divided further in to following phases:

1. Analyze
2. Design
3. Build
4. Qualify
5. Adopt

- Change Management including Training
- Roll Out
- Overall Project Management

The client purchased the SaaS application and the deployment activities from the case organization in 2013. After the purchase, case Alpha was launched in late 2013 with analysis and design phases. The project management team consisted of a leading project manager PM1 and a supporting PM2 (PMO). Additionally, a business partner was appointed to the project who provided guidance and steered the project. Later, in spring 2014 the leading PM1 was handing the responsibilities over to PM3 and also PM4 entered the project. After the Alpha go-live in 2014 June, the PM1 was off-boarding from the projects.

Overall the project progressed quite well according to the plans. As usual in a project, some changes and challenges also occurred in case Alpha. These changes and challenges were mainly related to the overall functional design and the integrations. The client requested changes and additions to the overall solution design after the build phase had started and some parts of the integration design were affected and had to be updated. This had impact on the project plan e.g. in terms of scope, timeline, activities, and resources. Additionally, the data transfer plans were slightly changed as a more effective approach was identified during the project and was taken into use. All of these occurred events were taken into account when planning and executing case Beta. Section 4.2, includes more discussion and findings related to the project management events.

4.1.4 Case Beta

Case Beta was launched in spring 2014 and the original scope comprised enabling functionalities for two business streams, building five integrations, continuing the data transfer,

conducting trainings, and making enhancements to the existing solution. Case Beta aimed to extend and enhance the solution delivered in case Alpha. As the project went on, the scope was reduced due to changes in the requirements and moving two of the integrations into later projects. The building phase of Beta is currently ongoing while the complete execution plans for case Beta are not confirmed yet. Currently it seems that case Beta will be completed with the same six different tracks as in case Alpha:

- Configuration / Customization without any code changes
- Integration
- Data Migration

These three functions above were divided further in to following phases:

1. Analyze
2. Design
3. Build
4. Qualify
5. Adopt

- Change Management including Training
- Roll Out
- Overall Project Management

In case Beta the case organization's project team was smaller than in case Alpha. The leading project manager PM1 was leaving the project and was replaced by PM3 for both cases Alpha and Beta during spring 2014. PM2 was staying in the projects as responsible for PMO activities. Additionally, third project manager, PM4 was on-boarding to the projects. Changes occurred also in other team members due to changes in the scope.

Case Beta was conducted in separate packages and therefore there were some project activities inside the project phases which needed to be repeated. For example, before the analysis phase, the created project plan was only covering a detailed plan for analysis and design phase whereas in case Alpha the whole project plan was done before starting the analysis phase.

In case Beta the requirements from the business streams were not as clear as in case Alpha. In addition, lessons learned from case Alpha indicated that some design decisions were better left to be done after clarifying all of the requirements in order to avoid changes in later project phases. Therefore, it was decided that the project would be conducted in small packages. This enabled a more agile approach to the project and made it more flexible for changes. Currently the analysis and design phase for case Beta has been completed and the first part of the implementation phase is ongoing. One project management option for the second part of the implementation to use an agile approach.

4.2 Deployment of a Software as a Service application

The findings of the case studies were divided into four themes: characteristics of the SaaS deployment, managing the SaaS deployment, SaaS deployment project phases, and evaluating the used project management methods, tools, and practices. All of the findings are presented and discussed in the following sections. First, the findings from the project manager (PM) interviews and the survey results from the project team members (PTM) are presented. Also, the findings are complemented with the author's observations, learnings, and with the data collected from the project related material. The majority of the collected empirical data is applicable for both cases Alpha and Beta. The possible differences are presented in every section.

4.2.1 Characteristics of the SaaS deployment process

The key benefits of a SaaS application were described to be easy access and usage via browser, branding and customization possibilities, supporting many users, and enabling integrations.
First, the interviewed PMs were asked about the characteristics of SaaS and the deployment project. They were asked to describe how the SaaS and the deployment process differs from a traditional software project and what kind of requirements SaaS added to the project management.

SaaS versus standard software

SaaS as a term was familiar to all of the PMs and PTMs, but for most of them case Alpha was the first SaaS project they have been working in. This supports the statements that SaaS is a new phenomenon in the field of software business. The arising popularity and importance of knowing the nature of SaaS was mentioned multiple times.

Compared to a standard software deployment multiple differences were mentioned. Most important differences mentioned were related to provisioning, customization and configuration, managing, and maintaining the software. The respondents described SaaS solutions as fast and easy to provision, with limited customization possibilities, and outsourced maintenance and hosting work. Additionally, one PTM commented that the outsourced management and hosting affect the project scope because it moves the focus closer to the client specific needs and business requirements and away from the needed IT resources. One of the PMs supported this statement by commenting that the start of a SaaS deployment project is different compared to a standard deployment since the vendor needs to ensure that all the needed computing resources are in place for the deployment activities. Onwards, this PM sees that the process has more similarity with a standard process.

One of the PMs and one of the PTMs mentioned that SaaS differs from traditional software project from a contractual and implementation point of view. In SaaS there are separate contracts and implementations for the actual SaaS solution and for the deployment. This comment was also supported by the literature findings.

All the PMs commented that the due to the fact that the deployed application was a single-tenant solution, there were less differences compared to a standard software deployment process. The single-tenancy had an impact on e.g. provisioning process, scalability, and customization and

configuration possibilities. The provisioning and scalability was not as fast and easy as in multi-tenant solution whereas the customization and configuration possibilities were significantly more extensive compared to the multi-tenant solution.

The usage of multiple environments is recommended in software deployment process to ensure the quality of the deployed product. Overall five environments were used in cases Alpha and Beta: Development (DEV), Sandbox (SB), System Integration Testing (SIT), User Acceptance Testing (UAT) and Production (PROD). In the case projects the usage of multiple environments was very beneficial due above mentioned reasons and because the scopes included many different functions and activities which required usage of multiple environments in order to get them completed and tested before releasing them to production. Some of the activities were related to data transfer, version upgrades, and customization and configuration work. In order to maximize the utilization of the environments the vendor provided a plan for the environments usage which was adjusted and agreed with the client.

PM4 stated that the most significant feature which separates SaaS projects from traditional software projects is how the support functionalities are organized. Usually the set up team is part of the project team and the support activities are done by project team members, whereas in SaaS there is a centralized remote support team solely taking care of the support. The case organization has a support tool available in the internet. All the issues, functionality requests, and version upgrade related matters were handled through this tool. The support tool related benefits were e.g. easy to access, traceability of the tickets, online service available 24/7, root cause analysis available for issues, and a quick response time.

PM3 mentioned that the deployment process was more transparent compared to on-premise deployment and control of the releases was better. New releases (version upgrades) were requested and provided via the online support tool. The client simply needed to raise a ticket and the support team proposed an upgrade time which the client either approved or requested a new proposal. After this the upgrade was executed by the support team in the given timeframe. Overall the version upgrading was a fast and easy process with a quite short downtime period.

It was interesting to notice that most of the comments related to SaaS specific characteristics were from PMs whereas the PTMs mostly commented that the project felt to them as a standard software project. Additionally, the PMs had different opinions about what are the most significant differences and similarities of SaaS compared to a standard software.

Addressing concerns and preventing possible challenges

Every project faces concerns or challenges during the execution. In order to prevent possible SaaS related concerns and challenges the case organization wanted to educate and provide information for the project team members, especially for the project managers. The cases organization provided information about the cloud solutions, previous projects (e.g. lessons learned), multiple examples, educational material, education related to possible risks, and offered subject related education. In addition, the case organization has an internet portal which provides extensive amount of material e.g. process charts, examples, templates, tools, guidelines, and practices for different project and software types which can be utilized in project executions.

The initial level of knowledge about SaaS and the deployed application characteristics varied among the project team members. As mentioned, all of the PMs and PTMs knew about SaaS but only few had previously been working in a SaaS project. Consultants who had previously worked with the solution naturally knew more and the members not been working with the application nor SaaS obviously did not have the same level of knowledge. Overall it can be stated that after working in the case projects the knowledge level has significantly increased among the team members.

The business partner of the projects was the escalation point for possible issues. He worked closely with the project managers who were reporting the project status frequently. Issues related information was communicated internally and externally in weekly meetings, via status reports and in the Steering team meetings.

The case organization had identified key concerns related to SaaS adoption. These concerns were presented to the project members and they included:

1. Data security and privacy
2. Data and transaction integrity
3. Regulatory compliance
4. Integrations to existing solutions
5. Cloud provider transparency and performance level
6. Dependable delivery of required high capability
7. Viability of cloud providers
8. ROI of cloud not yet verified
9. Vendor and data lock-in
10. Cross-border data restriction. (Saugatuk Technology, 2013.)

Based on the project nature, the case organization identified that two of the concerns would possibly be relevant for the case projects. These were data security and privacy and integrations to the existing solutions. Both of the concerns were addressed as early as possible and managed by the service provider. Mitigation actions included, e.g. conducting security assessments and detailed analysis and design workshops for the needed integrations. Additionally, all the possible risks and issues were tracked and managed by the project Steering team.

PMs identified that main concerns the client had were related to the security, and integrations to existing solutions. These concerns were addressed and the case organization was handling the concerns by providing more information about the solution, organizing meetings, conducting integration specific workshops, and answering to possible questions. Additionally, both the case organization and the client had security processes in place which were followed by the project members. These security process activities were not all specific to SaaS, instead some were standard for every project but all of the activities were applicable for the case projects.

As found from the literature in chapter 2, integrations to an existing solution are usually complex and require data transfer activities. Careful design work with realistic planning were done to ensure a successful implementation. In case Alpha's scope there were two integrations whereas

in case Beta's scope altogether five integrations were to be build. From integration build and data transfer activities point of view, Beta was more complex and extensive than case Alpha.

As learned earlier, the deployment type and the architectural setup affects the achieved benefits and also possible concerns related to SaaS. Additionally, as SaaS is a new delivery model and the characteristics are usually not fully known possible concerns or questions related to the architecture and SaaS setup were addressed in the beginning of the project in order to avoid misunderstandings. For example the multi versus single-tenant aspects and the deployment type, i.e. private versus public cloud, may be unclear and confusing to the clients.

All of the PMs addressed the importance of clarifying the SaaS and the product specific requirements both to the vendor PTMs and to the client. Now the initial knowledge level varied among the team members and the client, as mentioned earlier. It would have been a significant improvement for the project success if the characteristics of SaaS and the application would have been better clarified in the early phases of the project. This would have hastened and enhanced completion of some project activities (e.g. analysis and design work) and prevented change requests in the later phases of the project.

As mentioned, some of the PTMs were working in different countries. Additionally, some of the client team members were working abroad. This had an impact on the project activities but it was not a SaaS specific challenge, instead remote working is always challenging regardless of the project type. Main challenges related to remote working is the ability to fully participate and concentrate on the meetings and the time zone differences. Overall, both PTMs and PMs stated that overall the global team collaboration worked very well and the team spirit was quite good throughout the projects.

The deployment of a simple SaaS application is an easy remotely completed process. The case projects were enterprise SaaS application projects which increases the complexity and the requirements of the project. It was learned from the case projects that the need of onsite consultants was occasionally identified to be quite high, because fulfilling the business requirements requires close cooperation with the client. Especially for the project kick-offs, analysis and design workshops, build phase activities, and training the need for onsite participation was seen important. The on-site participation was seen to reduce needed time and effort, enhancing the communication, building the team spirit and customer relationship and involving the customer. It was clearly noticed that onsite face to face conducted workshops provided more and more valuable output than the remote collaboration. This is mainly due the fact that if workshops are attended remotely, the participants cannot give a full commitment to the workshops and the concentration is always challenging. According to the PMs these on-site activities were playing an important role in the project success. Therefore, when there was a chance to organize onsite activities the opportunity was always used.

It became clear in a quite early stage that the time zone differences between the team members were causing some challenges in the collaboration and communication. The time zone differences affected mostly in finding suitable meeting times and in daily communication. Scheduling for example, status meetings and client meetings were very hard due to eastern and western participants. As some were complaining of early mornings, for some it was already late

night. Also, basic communication such as emailing, chatting or calling was occasionally hard due to reasons described above. The challenges of different time zones were handled by scheduling meetings early and by adjusting the daily working hours based on the project situation.

On the other hand it can be considered that the time zone differences were a good thing for the project since almost in every hour someone was working in the project in the so called follow-the-Sun principle. This enabled fast reaction time for possible changes and issues, constant progress, and scheduling activities which needed to be done while no one was using the environments (e.g. testing and upgrading activities).

From the results it can be summarized, that one major reason causing challenges was the unclarity of the characteristics related to a SaaS application and the nature of the project. Additionally, the integrations to existing solutions with data transfer activities were the most challenging activities of the projects. The unclarity of the requirements and the effort required to building the integrations should be taken into account when planning and managing a SaaS deployment.

4.2.2 Managing the SaaS Deployment project

After the specific features of SaaS were surveyed, the following questions were related to project management. Questions aimed to discover what should be taken in to account in managing a SaaS deployment project and how the project management activities did differ from a traditional project. This chapter presents the results and opinions regarding the utilized PMM, its suitability to the project, and considerations whether an agile method would have been more suitable for the case projects.

Selection of the PMM

The cases Alpha and Beta were executed by utilizing case organization's methodology which can be considered as a traditional approach. The methodology was not a SaaS deployment specific methodology, yet it was slightly adjusted to fit the project characteristics and requirements e.g. project being a deployment process, the SaaS application, and phased project execution. Additionally, the methodology was adjusted according to the client's processes and needs. For example, the project phases were named according to the client's naming convention and the phase deliverables and outcomes were specified according to the cases. Even though the case projects were executed as separate projects, they had many common activities, resources, features, and dependencies.

The implementation methodology divided the project in to five phases and provided guidance, processes, practices, and tools for all of the phases. In addition to these, the case organization required many standard project activities, tools, and processes to be completed throughout the project life cycle.

All interviewed PMs and PTMs were asked about the project management methodology used in the project. All identified the used project management methodology being case organization's commonly used and comprehensive methodology. Most of the PMs and PTMs had been using the method previously. The respondents did not have strong opinions about the

PMM, however the benefits and shortages were identified. PM1 stated that the methodology was very extensive and heavy and therefore fits best large scale projects with a larger project team. Otherwise the utilized methodology was deemed to be working sufficiently.

Characteristics of the deployment of SaaS

PMs were asked whether the SaaS aspect affected the project management. All of the PMs commented that the cases had similarities to a standard software delivery, but also multiple differences were specified. PM1 and PM2 who were working in case Alpha stated that the initiation phase activities were most affected by the SaaS characteristics. In the initiation phase the needed resources were to be provisioned, configured, and taken into use in order to start the deployment activities.

The provisioning process in SaaS should generally be very straightforward and fast as the scalability is one of the key benefits of the cloud. PM2 stated that this should be taken in to account in the initiation and planning phases of a SaaS deployment project. Both PM2 and PM1 commented that for deploying a multi-tenant and a simple SaaS solution, the need for a formal project execution and, management, and a large project team should be questioned. However, for the enterprise SaaS deployments the need was identified and the usage was justified due to a more complex nature and larger scope of these projects.

As mentioned, the characteristics of SaaS were not so clear which caused some changes during the projects. Change control process was quite heavy and slow in the utilized PMM. PMs stated that allowing more changes and managing them in a more agile way were top priority modifications they would have done to the PMM.

SaaS was affecting the project management activities when tasks and responsibilities were identified in creating a project plan and a Work Breakdown Structure (WBS). Both the plan and the WBS were created in the early stage of the projects, already in the planning phases. For case Alpha, the tools were covering all of the project phases, whereas in case Beta the WBS was created separately for almost every phase. However, the analysis and design phase activities were combined.

SaaS is a standardized solution with limited customization and configuration possibilities. However, many of the SaaS solutions are single-tenant solutions allowing the customers to make more configuration and customizations than in a multi-tenant solution. In case Alpha and Beta, the deployed application was a single-tenant solution which increased the amount of client specific customizations and configurations.

Cloud solutions and their implementation model are rather new and the organizational structure does not always cover all the specific roles and activities. The project was executed by two major line of business functions inside the case organization. All of the roles and activities were not easy to identify and to address the responsible persons, for example, while creating a project plan and a WBS. The same situation was also noticed during other project activities. In these situations the required role or the responsible person was named by the project manager or in the Steering team meetings. Overall, from the respondent's opinion, the collaboration between the business lines was successful.

Generally, managing the initiation phase and the issues were seen to be most affected by the SaaS aspect. Otherwise the project management activities were experienced to be quite standard.

Considering an agile approach

Additionally, the respondents were asked whether they think an agile approach would have been more suitable and effective for the project management. First to be mentioned, every respondent seemed to have heard of or to know what agile methods are in general.

The answers were quite aligned with both PMs and other team members. There were no strong opinions that the agile methodology would have worked better on this project. Hence, possible benefits of using an agile approach were identified by quite a many of the respondent. All of the respondents knew that changes will always occur regardless of the project and agile methods would bring assistance to managing these changes. Project objectives and targets were clearly defined in the beginning which supported the use of a traditional method.

Among the PMs some justifications for using an agile methodology included e.g. flexibility on change process, iterative work, constant communication, and better customer involvement. One PM stated that a modified agile approach would have been the best option. The benefit of an agile method would have been the flexibility on requirements clarification and a lighter change management process. In case Alpha the traditional step-by-step approach was discovered not to be the best option in the design phase because in the early project stage all the characteristics of the application and the cloud model were not entirely known. This caused some modifications to the design in later phases, which obviously required extra effort.

In few of the PTMs' opinion agile would have been a more suitable approach considering reporting, flexibility and responsiveness to changes. Also, even though agile was not suggested to be a solution, the author concluded from the answers that agile would have increased transparency, collaboration, and enhanced internal and external communication, which were addressed as developments areas in the project management.

One interesting consideration related to agile was that in both case Alpha and Beta, the case organization used demoing and sneak peeks in order to ensure that the client understood the product characteristics and also was satisfied with the features and functionalities. The demos were held during analysis and design phases in order to help with designing the application whereas the sneak peeks were held after the case organization had already built some functionalities and was willing to review the current results with the client. This approach carries similarity with the so called sprint release demos which are used in agile projects in which the current version of the product is demoed after every sprint. Additionally, few of the PMs have said that the sneak peeks were in a critical role while building some of the functionalities.

4.2.3 SaaS deployment project phases

One important part of the study was to figure out the deployment process e.g. the deployment type, project phases, and project activities.

The deployed solution was a SaaS application which was deployed as a public cloud and a single-tenant solution. The application is case organization's own solution and it has been in the

market for many years. After the initial deployment all the clients receive new versions of the application which are frequently released to production.

As learned in the literature chapter 3, a software deployment process includes the release, install and activate, deactivate, adapt, update (version upgrade), uninstall, and obsolescence phases. The used PMM divided the deployment project in to five separate phases (also called as sub-phases): analysis, design, build, qualify, and adopt. In addition the preliminary (initiation) phase was included in the planning. It can be seen that the software deployment process phases and the case projects PMM phases do not match. Instead, the PMM phases fit to the traditional project management approach. A comparison of the process phases can be seen in chart X.

According to the used PMM, all of the project phases should include specific activities and outputs as presented below:

Preliminary/Initiation phase

- Complete all needed activities for preparing the project to start e.g. contract negotiations, contracting, procurement process, planning, and preparations

Analysis

- Detailed project plan
- Identification of key named stakeholders and roles
- Develop change management plan and approach
- Train project team
- Define and collect data to be migrated

Design

- Conduct process workshops
- Design all needed functionalities
- Design integrations
- Develop test cases
- Design documentation
- Ensure availability of resources/stakeholders
- Complete project plan

Build

- Configuration of all designed elements
- Build integrations
- Data transfer
- Reporting

Qualify (and Pilot)

- Development
- Testing activities
- Training
- User acceptance testing (UAT)

Adopt

- Training
- Handover to support.

According to PMs, this process approach fitted the project execution quite well. One adjustment which would have made the approach better was more flexible change process. Additionally, PM1 questioned the need for this heavy process since the case projects were quite small. In PM1's opinion the process activities are most suitable for a large scale and complex project. However, as mentioned the SaaS application deployment is a rather complex and extensive project and therefore the used PMM fitted the project execution. In agile approach the process is much more flexible and changes may occur in all process phases. This would have been the motive for selecting an agile approach, otherwise the used process was sufficient.

4.2.4 Evaluating the utilized PM methods, tools, and practices

The remaining questions were about the used project management methods, tools, and practices with a purpose to find out which of them were experienced to be the most effective and important, considering the project progress and success. Additionally, the respondents were asked to rate the most important and effective methods used, tools, and practices.

PMs identified WBS, internal status meetings, project wikis for document sharing and management (both internal and external), Gantt chart and a detailed project plan as the most important factors for the success of a project.

A detailed WBS was created for the whole project in the beginning of case Alpha. For case Beta, a WBS was created separately for each project phase, except for the analysis and design phases the WBS was combined. The WBS contained detailed information about the project related activities, tasks, roles, and responsible parties/persons. WBS was used to keep track on the overall project progress and it was constantly monitored and maintained by the project managers e.g. in status meetings. Based on the research results, the PMs gave more value to the importance of WBS compared to PTMs. The WBS contained information such as task, assigned person/party, duration, start and end date, progress, comments etc.

Weekly status meetings were held every week with an intention to have a status check with the project team. The importance of these internal meetings was highlighted by both the PMs and the PTMs. In addition, project managers held weekly project status meetings with the client. In the meetings, the PMs presented a weekly status report to the client. The report was updated weekly and included information of completed and planned tasks, possible issues and mitigation actions. PMs stated that these weekly meetings with the client were experienced to be very important from the project success point of view.

The Steering team (ST) meetings were used in both case Alpha and Beta. Participants of the ST consisted of the case organization and client employees. The meetings were held approximately once in a month. The meeting material was prepared by the case organization and was reviewed in the meeting. All the project status related topics, decisions, issues, changes, and any other business related topics were discussed during the ST meetings.

To summarize, based on the interviews and surveys the most important project management tools, methods, and practices were identified to be the weekly internal and external status meetings, steering team meetings, status reports, WBS, and a detailed project plan.

5 Conclusions and discussion

This final chapter presents the results and conclusions of this study. It begins with a presentation of the answers to the two research questions in section 5.1. After this, the last section, 5.2 includes discussion and ideas for possible future research of the findings.

5.1 Results of the study

The main objective of the study was to identify the characteristics of PMMs that are applicable to and effective in managing SaaS deployment projects. The general research problem of this thesis was:

> *What kind of project management methodologies are most suitable for managing SaaS deployment projects?*

Further, the research questions appointed to this study were as follows:

1. *What are the special characteristics of a SaaS deployment project?*
2. *What should be taken in to account in project management activities and what project management practices, tools, and methods should be implemented when managing a SaaS deployment project?*

This section presents answers to the research questions. In the following paragraphs the findings from both the literature review and the empirical research are combined, presented and discussed in order to identify answers to the research questions. Overall, the results from this study were more extensive and coherent than was originally expected and they provide more than guidance for utilizing and choosing a PMM. This can be seen from the research problem and questions which all concentrate on the utilization of the PMM, whereas the results provide recommendations for overall project management related activities in a SaaS deployment project.

As a result of this study many project management related activities, methods, tools, considerations and general advises can be presented and recommended to be used for managing a SaaS deployment project. These recommendations are mainly intended as a guidance for the vendors executing the deployments, but can also provide valuable information to other parties. The recommendations can be divided into three themes which are the project management methodology, project management tools, methods, and practices and recommendations related to general advice and guidance.

5.1.1 Empirical findings

This section include a summary and analysis of the findings from the studied case projects. The results are presented and analyzed in the following paragraphs.

SaaS versus standard software deployment

Based on the results, SaaS was identified to be a new trend in the field of software business that most people have heard of but of which most have no previous experience. Additionally, the increasing popularity of SaaS and the importance of knowing it was mentioned. Multiple differences compared to a standard software delivery model were identified e.g. easy and fast to provision, light implementation project without onsite work, limited customization and configuration possibilities, outsourced maintenance and hosting, and reduced delivery costs. All the PMs mentioned that the deployed solution was a single-tenant solution which affected the achieved benefits and concerns related to the SaaS application.

Additionally, one major difference is that in SaaS model there are usually different contracts and implementation projects for the actual solution (including the needed IT resources) and the deployment activities. Therefore, in SaaS deployment projects the focus is more concentrated on the business requirements and processes because the IT infrastructure is implemented in a separate project.

The deployment of SaaS was stated to be more transparent compared to an on-premise deployment and the management of releases was better. For SaaS solutions the support is provided as a centralized remote function which differs from a traditional support setup. In the case projects the case organization had a support tool available online which was used in all support related activities. The usage of multiple environments was recommended in the deployment project which is also common for on-premise deployments. In case projects overall five environments were used to enhance the quality of the deployed product before it reaches the end user.

SaaS related concerns

It was mentioned that every project faces some kind of challenges and concerns during the life cycle. This was also known by the case organization and therefore the possible concerns and risks were addressed and mitigation actions were done in order to prevent challenges in the project phases.

The common concerns related to adopting SaaS solutions were identified by the case organization. These included:

1. Data security and privacy
2. Data and transaction integrity
3. Regulatory compliance
4. Integrations to existing solutions
5. Cloud provider transparency and performance level
6. Dependable delivery of required high capability
7. Viability of cloud providers
8. ROI of cloud not yet verified
9. Vendor and data lock-in
10. Cross-border data restriction. (Saugatuk Technology, 2013.)

Data security and privacy and integrations to existing solutions were identified to be possible risk factors for the projects. Both concerns were addressed in the beginning of the project and mitigation actions were conducted.

The PMs identified that the major concerns the client had were related to security and integrations to existing solutions. The amount of required integrations and data transfer were noticed to be challenging in the project planning phase. Therefore, the analysis and design phases were carefully planned and multiple workshops for the integration design were conducted.

The unclarity of the solution characteristics is always concerning and might cause troubles in the project activities. The characteristics of SaaS were not fully clear which caused some concerns and required extra effort in the project. The PMs stated that the knowledge level related to SaaS varied a lot among the project team member and the client employees and it would have been improvement if the knowledge level would have been better already in beginning of the project.

Some of the vendor employees and the client employees were working in different countries. This caused challenges due to the time zone differences and remote working. Even though SaaS deployment does not primarily require onsite activities, it was discovered that face-to-face meetings and onsite working enhances the quality and output of working. On the other hand the time zone differences were seen as a benefit since almost in every hour someone was working in the project. This enabled fast reaction time for possible changes and issues, constant progress, and scheduling activities which needed to be done while no one was using the environments (e.g. testing and version upgrading activities).

From the results it can be summarized, that one major reason causing challenges was the unclarity of the characteristics related to a SaaS application and the nature of the project. Additionally, the integrations to existing solutions with data transfer activities were the most challenging activities of the projects. The unclarity of the requirements and the effort required to building the integrations should be taken into account when planning and managing an SaaS deployment.

Managing the deployment

The case organization has created a project management methodology which was used in managing the case projects. This PMM was not specifically meant for managing a SaaS project but it was slightly adjusted to fit the project specific requirements. Overall the PMs and the PTMs stated that the utilized PMM was fitting the project execution quite well. PMM divided the project activities in six to functions:

- Configuration / Customization without any code changes
- Integration
- Data Migration

Three functions above were divided further to following phases:

1. Analyze
2. Design
3. Build
4. Qualify
5. Adopt

- Change Management including Training
- Roll Out
- Overall Project Management.

The key characteristics of managing of a SaaS deployment project were identified based on the empirical results. As mentioned, typically the SaaS solution and the deployment are implemented in separate projects. This affected the project management and increased the criticality of the beginning of the project where all the needed cloud resources were to be delivered to the client. Even though the solution and deployment were conducted separately they had multiple dependencies and tight cooperation between the projects was needed. Due to separate execution and an easy deployment it was discussed whether the SaaS deployment project require heavy project management activities. The conclusion was that in the SaaS deployment the scope and nature of the project is significantly more extensive and complex and therefore in those cases there is a need for formal project management activities.

Due to the unclarity of SaaS specific characteristics it was recommended that the change management procedure should be flexible in SaaS projects. Also, some of the project related tasks, activities, and roles were not so easy to identify and assign due to the unclarity of characteristics and the fact that SaaS is a new software delivery model. Even though the IT resources and the deployment are delivered in separate projects as mentioned above, tight cooperation is needed which is a new delivery model for the vendor organizations.

As a standardized solution SaaS offers limited configuration and customization possibilities. This need to be taken into account in project management activities. Additionally, the management of the client's expectation is highlighted due to many clients are used to that all software will and can be tailored entirely according to their specific needs and business requirements. The limitations of the solution should be clearly addressed in the initiation phase.

SaaS is stated to integrate well to existing solutions. However, integrations and the needed data transfer are always complex and require lot of effort and careful planning. This should be taken into account in the project management.

SaaS affects the organizational structure and the vendor's and client's organization's processes. Many clients are used to full tailoring possibilities of purchased software. Traditionally the implementation project has been extensive and the purchased software is entirely customized and configured for the specific client or the software is particularly developed for a specific client. As learned, in a standardized solution such as SaaS, the customization and configuration possibilities are quite limited. This might require the client to adjust its business processes when SaaS solutions are adopted in to business functions.

The suitability of an agile project management methodology was not strongly commented but some benefits of an agile approach were identified which would have enhanced the project performance. Most important benefits were the flexibility on requirements and change management, iterative working, constant and informal communication, better customer involvement, transparency of the work, increased collaboration, and frequent demoing of the results.

Many project management methods, tools, and practices were identified to be suitable and successful in the SaaS deployment project management. These included the internal and external weekly status meetings, weekly status reports, Steering team meetings, WBS, detailed project plan, and project wikis for document management and sharing.

According both to the PMs and PTMs the weekly status meetings where one of the most important project management activities in the cases in which all the project related topics, activities, issues, and team related topics were discussed. All of the PMs mentioned them as important, but the PTMs highlighted their importance more and even commented the weekly meetings to be the key factor for project success. Further, as mentioned weekly status meetings were held both internally and with the client. All of the project members, PMs and PTMs were participating the internal meetings whereas PMs were only attending the client meetings. Therefore, it was not a surprise that PMs valued the external meetings occasionally even more than the internal meetings. Additionally, the status reports which were related to these weekly meetings were described to be an effective management tool in the project.

5.1.2 Recommendations for utilizing the project management methodology

Even though SaaS has already increased its popularity and offerings significantly, best practices et cetera have not matured yet. Additionally, there are many concerns and features (also in future) which could be addressed/solved in order to better meet the client's requirements and to achieve a better market position. Therefore, there is a lot of room for research to be done on the subject and a lot of potential benefits for companies willing to invest on studying the area right now.

It is strongly recommended that for a SaaS deployment management a suitable project management methodology (PMM) is utilized. This methodology should fit to both SaaS deployment specific characteristics and project specific characteristics. The deployment specific characteristics include fast and easy provisioning, vagueness of requirements and characteristics, light implementation project no need for on-site activities, limited customization and configuration possibilities, rapid onlinke2016Sae version upgrading process, outsourced hosting and maintenance work and reduced delivery costs. Additionally, from the deployment point of view, major findings were centralized support functions, higher transparency in release management, and separate delivery of the SaaS solution and deployment activities.

General features of a project to be considered when choosing a PMM which also should be taken into account in a SaaS deployment project are e.g. budget, team and project size, project criticality, documentation, training, best practices, tools and techniques, and examination of existing processes, technology used and the deployed software. From SaaS point of view, most

important of these features are training, examination of existing processes, technology used and the deployed software.

In the case studies the used PMM included project phases and the activities as follows:

Preliminary/Initiation phase

- Complete all needed activities for preparing the project to start e.g. contract negotiations, contracting, procurement process, planning and preparations

Analysis

- Detailed project plan
- Identification of key named stakeholders and roles
- Develop change management plan and approach
- Train project team
- Define and collect data to be migrated

Design

- Conduct process workshops
- Design all needed functionalities
- Design integrations
- Develop test cases
- Design documentation
- Ensure availability of resources/stakeholders
- Complete project plan

Build

- Configuration of all designed elements
- Build integrations
- Data transfer
- Reporting

Qualify (and Pilot)

- Development
- Testing activities
- Training
- User acceptance testing (UAT)

Adopt

- Training
- Handover to support.

These phases and activities differ a lot from the traditional software deployment process which includes additional phases and activities e.g. release, install and activate, deactivate, adapt, update / version upgrade, uninstall and obsolescence. When comparing the case project and the traditional software deployment process phases some similar activities can be identified e.g. installation, activation, adaption, version upgrading, data transfer, configuration, testing, training and handover to support activities. However, the deployment process phases are only covering the phases and activities which solely cover the deployment and therefore the project management aspect with various additional features is not considered. Therefore, the deployment process activities do not cover all the needed activities for a deployment project.

In the case projects, the used methodology was a traditional one which is more widely used in a software development than in a deployment project. However, as mentioned the PMM fitted the project execution well. Some suggested enhancements were the flexibility on requirements and change management, iterative working, constant and informal communication, better customer involvement, transparency of the work, increased collaboration and frequent demoing of the results.

It can be concluded that the traditional approach suits an SaaS deployment project well, but some enhancements to the PMM are beneficial. Additionally, SaaS related concerns and risks should be addressed and mitigated as early and as well as possible. Major concerns were related to security, performance level, vendor and data lock-in, complexity of integrations and data transfer and unclarity in solution related characteristics.

Other general advice to the SaaS deployment project management were also identified. It is highly recommended to consider the usage of multiple environments in the deployment project. Project team, roles, tasks, and responsibilities should be clarified in the early project phase in order to avoid situations where resources for occurred tasks are not appointed.

Clear communication and expectation management is also an important part of the project management activities. It is highly important to inform the client that the choice of the deployment type and the architectural setup affects the achieved benefits and possible concerns. SaaS is a new delivery model and therefore in the beginning of the project, SaaS related education and information should be provided to the project team members, client and other relevant stakeholders. Due to the same reason the analysis and design phase might require more time and effort, which should be taken into account in the project planning.

To conclude, based on the findings the most important characteristics of a project management methodology in managing SaaS deployment projects include e.g. transparency and constant communication with the project team and the client, flexibility on change management, detailed requirements engineering, enabling iterative work, frequent demoing of the results and training,

In addition, the most important project management methods, tools, and practices that could be utilized in the SaaS deployment projects were identified to include weekly status meetings with

the project team and the client, status reports, Steering team meetings, detailed WBS, overall project plan and project wikis for documents sharing and management.

Additionally, based on the findings a suitable PMM for managing a SaaS deployment project would most likely be a methodology which contains a combination of characteristics from both the traditional and agile methodologies. For example, it would be good to have characteristics of an agile approach in the analysis and design phases so that the build activities could be started before the design is entirely completed.

5.2 Discussion and future research

This section includes the discussions and future research possibilities of this thesis. The expectations, findings, author's considerations, and future research suggestions are presented in the following paragraphs.

5.2.1 Characteristics of SaaS

Both literature and empirical findings supported the statements that SaaS is a new software delivery model and one of the biggest trend in the software business that is increasing its popularity. The amount of previous research and gained experience from SaaS were described to be limited.

The term "pure" SaaS appeared multiple times in the study, in the literature review and in the empirical part. So called pure SaaS solution is a multi-tenant solution deployed usually in the public cloud model. It was discovered that most of the cloud related benefits were achieved with the pure SaaS solution. However, it was also stated that the cloud benefits could be achieved with a single-tenant option, too. In the SaaS solutions, the single-tenancy is a commonly selected model and therefore it would be important to study if this statement holds true. Additionally, the selected deployment type (e.g. public, private, or hybrid cloud) was also affecting the achieved benefits and avoided risks. The deployment types and their impact on the SaaS characteristics could be studied together with the multi versus single-tenant aspects. Especially in the single-tenant cloud model it is critical that SaaS infrastructure is carefully planned, analyzed and explained to relevant stakeholders in advance prior to making decisions to acquire the SaaS. This is because fundamental changes to SaaS architecture are challenging to achieve afterwards in case it is found out the selected SaaS model or its architecture does not fit well to target environment. Such pitfalls could be for example related to data security, connectivity and flexibility to make changes to the SaaS solution. This view is supported by findings from the case projects.

Compared to a standard on-premise software, several differences can be identified in SaaS. These differences include e.g. easy and fast to provision, light implementation project without on-site work, limited customization and configuration possibilities, outsourced maintenance and hosting and reduced delivery costs.

Additionally, one major difference is that in SaaS model there are usually different contracts and implementation projects for the actual solution (including the needed IT resources) and the

deployment activities. Therefore, in SaaS deployment projects the focus is more concentrated on the business requirements and processes, because the IT infrastructure is implemented in a separate project.

The deployment of SaaS was stated to be more transparent compared to an on-premise deployment and the management of releases was better. For SaaS solutions the support is provided as a centralized remote function, which differs from the traditional support setup. In the case projects the case organization had a support tool available online and it was used in all support related activities. The usage of multiple environments was recommended in the deployment project which is also a common practice with the on-premise deployments, too. In case projects in total five environments were used to enhance the quality of the deployed product before it reaches the end user.

5.2.2 SaaS Deployment process

As mentioned, SaaS is a new delivery model and the characteristics are not usually fully clear to the client. This is the main reason for possible challenges and concerns related to the deployment which may occur e.g. in the analysis or design phase as some of the design decisions cannot be done or are done based on wrong assumptions. Both of these situations affects the progress of the project and the activities related to the later project phases.

Literature defines the software deployment process as being a part of the system development life cycle which covers all the post-development activities that are needed to deliver the solution to the end users. In the case projects the deployment was considered more as a normal software project. This can be seen from the differences in the project phases and activities. The software deployment activities are the release of the developed software, customer specific configuration and customization work, installation, activation, monitoring, deactivation, version upgrading, reconfiguration, adaptation and uninstalling. Whereas the case project phases were analysis, design, build, qualify and adopt. As mentioned earlier, the standard software deployment process is usually considered as a part of SDLC activities and therefore it provides quite general guidance for the process activities. Additionally, it was stated that every software contains its own specific characteristics and the deployment should be considered as a general process that will be customized based on the software to be deployed and the client specific requirements and characteristics.

Overall the standard software deployment process phases described can be considered as applicable for a SaaS deployment, yet some differences and similarities of the activities can be identified. Additionally, these standard deployment phases and activities are more applicable in case of releasing and deploying a new version of an already pre-deployed product to the end users.

All the customer specific requirements, functionalities, integrations, customizations, and configurations are completed during the deployment process which makes the process highly important, especially for a standardized solution, such as a SaaS application. Three process characteristics that are adding complexity to the deployment are existing integrations between client's other systems, data transfer, possible custom development, various configuration

options, and a requirement for a complex pre-created data model, which are common for an SaaS deployments. Additionally, SaaS application deployments are usually large processes, with the number of employees ranging from hundreds to thousands.

The complexity of integrations and data transfer activities were more highlighted in the empirical findings than in the literature. Additionally, the client of the case projects addressed the importance of both the integrations and the data transfer. It was stated that without these the achieved business benefits would have been significantly reduced.

The usage of multiple environments in the deployment process was recommended in both literature and in empirical findings. The software is deployed through the environments in order to increase the quality of the product before it reaches the users.

The deployment of SaaS affects the organizational structure and the processes of the vendor and the client organization. It is important that client understands that SaaS is a standardized solution with limited customization and configuration possibilities. Therefore, especially in case of deployments, the client might need to reduce the business requirements or adjust its business processes according to the functionalities of the solution. In an on-premise solution the deployment process is significantly different requiring e.g. more effort, resources, onsite activities, custom development and tailoring work. Additionally, the major difference in SaaS solutions is that the solution (e.g. required cloud components) and the deployment activities are commonly completed as separate projects. Therefore, also the vendor needs to adjust its processes and organizational structures in order to be able to deliver this new software model. However, the case studies showed that even though the projects are implemented separately by the same vendor, the need for a constant and tight collaboration is required.

Personally, I would see SaaS as an enabler for small and medium businesses to provide software to the common public and to the enterprises. Earlier software companies needed to make large investments in their IT infrastructure and resources in order to be able to provide and produce software for clients. Nowadays, it requires rather small investments from companies willing to develop and offer a cloud solution. They can simply purchase needed IT resources from a cloud computing vendor to develop their own solutions. This, of course, increases the amount and quality of the cloud solutions available in the market. Additionally, the more there are solutions available in the market, the more vendors need to address clients' demands and keep the prices on a competitive level.

The literature material considers the deployment process as a part of the SDLC whereas in the case study the deployment was clearly considered as a separate project. Additionally, as part of the SDLC the project management methods are chosen for the whole SDLC and the same methods are commonly used in the deployment process as well. This means that the methods are selected based on the SDLC characteristics and not based on the deployment process. It seems that the software business is moving more into tuning and customizing of pre-produced solutions. Therefore, the importance of deployment management will increase in the future and most likely project management methods, especially for the deployment, will be created and addressed.

Literature does not highlight the data transfer activities as much as was done in the empirical study. The data transfer and integrations building are seen as key factors in the case projects. Without existing data the benefits of the solution might be tremendously reduced. For me a lack of highlights of data transferring is quite surprising, because it should be quite obvious that especially in the enterprise software deployments some data always exists already and it usually needs to be available also in the future. In my opinion if the data transfer is not possible many companies might not achieve the benefits or see the business value of the new solution and are willing to continue using the old systems.

Additionally, the security and other data transfer related aspects need to be taken into account. As SaaS solution is in a remote Cloud environment users need to have means to connect to the Cloud environment securely and reliably from corporate networks, office LANs and personal computers. To compare, in an on-premise solution it would be very likely that a corporation had existing VPN connections to its server rooms. For each SaaS solution a corporation acquires, it needs to solve the connectivity and data transfer needs separately. This increases the need for SaaS vendors to provide reliable, secure and flexible connectivity and data transfer solutions.

5.2.3 Managing the deployment

The studied SaaS deployments were executed as projects which can be considered as IT projects with limited resources and time. The objective of the deployment project is a ready to use solution. IT projects are usually complex in nature and therefore using a suitable project management methodology is recommended. These observations were supported by literature and the case study.

Choice of the PMM was more widely discussed in the literature review than in the empirical study. This is mainly due to the fact that in the case projects the used PMM was created and recommended by the case organization itself. For software projects it is common that the PMM is pre-defined by vendor or client organization and not selected by the project manager, as literature guides. Additionally, in many projects the project preparations are already started and the PMM selected when the project manager enters the project.

Generally it was stated that the more complex and extensive a project is, the more heavy and comprehensive the PMM should be. For the PMM selection the literature suggests to consider specific project features such as: budget, team and project size, project criticality, documentation, training, best practices, tools and techniques, and examination of existing processes, technology used and the deployed software.

As presented before, the phases and activities of a SaaS deployment differ significantly from those of a traditional software deployment process which includes phases and activities e.g. release, install and activate, deactivate, adapt, update/version upgrade, uninstall and obsolescence. Comparing the case project and the traditional software deployment process phases some similar activities can be identified e.g. installation, activation, adaption, version upgrading, data transfer, configuration, testing, training, and handover to support activities. However, the deployment process phases are only covering the phases and activities of a process which solely covers the deployment and therefore the project management aspect with various

additional features is not considered. Therefore, the deployment process activities do not cover all the needed activities for a deployment project.

Additionally, the term deployment management appeared in both the literature and in the empirical study. However, it is more used in the version upgrading process where a new version of pre-deployed solution is released to the end users. The studied deployment projects were initial deployments of the solution and therefore different. In the case of a SaaS solution, the version releases are managed and deployed by the centralized hosting and support team. New versions are released quite frequently and even during the case projects couple of new versions were released.

By comparing and combining the findings it can be concluded that some SaaS specific characteristics and consideration can be identified. Additionally, some characteristics and considerations regarding the PMM are common for every project type and should be adjusted to each project.

Based on the empirical results, some characteristics of agile methods were suggested to fit well in managing SaaS deployment projects which included the flexibility on requirements engineering and change management, iterative working, constant and informal communication, better customer involvement, transparency of the work, increased collaboration, and frequent demoing of the results. Also, the research of utilizing a mix of the traditional and agile project management would be an excellent subject for future research. Also the literature suggested that an agile method would be suitable for software projects which usually require tight collaboration, flexibility on changes, iterative work and customer involvement.

Personally, I see some challenges regarding the SaaS deployment management. The deployment process is described as a light and an easy project but especially for enterprise solutions this is not the case. As mentioned, there are many requirements, integrations, data transfer, customization and configuration work and possible custom development in enterprise deployments. This naturally requires quite a lot of effort, resources and time. Therefore, even though SaaS deployments are considered as fast and light projects, I would address the importance of careful planning and reserving enough effort and time for these activities. This might reduce the economic benefits and the time to market of a SaaS deployment, but on the other hand, it ensures that the purchased product truly fits the client's needs.

It was interesting to see that the empirical findings highlighted the high value of the physical meetings. The importance of having physical meetings is related to understanding the SaaS architecture and infrastructure which can be best achieved in face-to-face technical and design workshops. Both project cases were single-tenant cloud solutions where solving connectivity, security and customizations was the key to successful SaaS solution delivery. Face-to-face meetings are important to consider in today's world where everything is being digitalized and remote working has increased.

Another interesting finding was that the project management related literature does not provide exact guidance for specific projects, instead it provides only high level assistance for selecting and using project management tools, methods and practices. This is of course understandable because as presented in the literature review, every project and even every deployment process is a unique project and suitable project management activities need to be selected and modified based on the specific project.

5.2.4 Future research

As a result of this study, recommendations and considerations related to selecting and utilizing a suitable PMM in SaaS projects were given. Additionally, many project management methods, tools, practices, and considerations that could be used in a SaaS project were found. By combining and analyzing the results, a suitable project management framework for SaaS projects could be created. This and the effectiveness of the created PMM would be interesting topics for future studies.

With more subject related research I see that a PMM specifically developed for SaaS deployment projects could be created or existing PMM cloud be modified according to SaaS specific needs. Additionally, more precise general advice to project management activities could be provided how to e.g. form a project team, scope and plan the project, manage the possible risks, manage the changes, and ensure the quality and execution of project deliverables.

Some characteristics of agile methods were identified to fit well the SaaS projects. Additionally, agile being one of the trends in the project management field and one of the author's main project management related interests, it would be highly interesting to study how to use agile methods in SaaS projects. Also, the research of utilizing a mix of traditional and agile project management would be an excellent subject for future research.

Different terms for SaaS (single-tenant, multi-tenant) appeared in the study and different SaaS deployment types were identified (e.g. public, private, or hybrid cloud). It was discovered that most of the cloud related benefits were achieved with the pure SaaS solution. It would be very interesting to study the differences, characteristics, benefits and challenges between these SaaS architectures and deployment types.

The usage of multiple environments in the deployment process was recommended both in the literature and in the empirical findings. From a SaaS perspective more research and recommendations of the usage (e.g. number of needed environments, where to complete testing and data transfer activities, how to maximize the benefits, and how to complete the version upgrades) would be needed.

Personally, I find SaaS as an important topic to study. SaaS has already rushed the software market and its popularity seems to be growing constantly. Occasionally it feels that currently everything is offered as a service. SaaS is being compared to SOA which recently was one of the biggest trends in software business. It is interesting to see how SaaS will evolve as its

popularity will grow and it will need to solve more complex and demanding IT and business problems.

As SaaS is a relatively new technology and a way of thinking, it is very likely that many of SaaS benefits and challenges are yet to be identified. Therefore SaaS will likely remain an interesting study topic also in future.

References

Armbrust, M. et al. 2009. Above the clouds: A Berkeley view of cloud computing. EECS Department, University of California, Berkeley. Tech. Rep. UCB/EECS-2009-28.

Attarzadeh, I. and Ow S. H. 2008. Modern Project Management: Essential Skills and Techniques. Communications of the IBIMA Volume 2 (1-9).

Baran, D et al. 2008.Cloud computing basics. http://www.webguild.org/2008/07/cloud-computing-basics.php .

Benlian, A. and Hess, T. (2011). Opportunities and risks of software-as-a-service: Findings from a survey of IT executives. (41 pages).

Bernstein, David et al. 2009. "Blueprint for the Intercloud – Protocols and Formats for Cloud Computing Interoperability". IEEE Computer Society. (328–336). doi:10.1109/ICIW.2009.55. ISBN 978-1-4244-3851-8.

Buyya, R. et al. 2008. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems (2009), doi:10.1016/j.future.2008.12.001.

Carraro, G. and Chong, F. 2006. Software as a Service (SaaS): An Enterprise Perspective. Microsoft Corporation. Available: https://msdn.microsoft.com/en-us/library/aa905332.aspxx .

Carzaniga. A. 1997. A Characterization of the Software Deployment Process and a Survey of Related Technologies. Dipartimento di Elettronica e Informazione. Politecnio di Milano Italy.

Charvat, J. 2003. Project Management Methodologies. Selecting, Implementing, and Supporting Methodologies and Processes for Projects. John Wiley & Sons, Inc. New Jersey.

Chin, C. M. M. et al. 2012. Project Management Methodologies: A Comparative Analysis. Journal for the Advancement of Performance Information and Value Vol.4 No.1 (106-118).

Chorafas, D. N. 2010. Cloud Computing Strategies. Auerbach Publications (354 pages) Citation. ISBN: 9781439834534.

Cockburn, A.2000. Selecting a Project´s Methodology. IEEE Software July/August. 64-71.

Columbus, L. 2015. Roundup of Cloud Computing Forecasts and Market Estimates, 2015. Forbes 1/24/2015.
Available: http://www.forbes.com/sites/louiscolumbus/2015/01/24/roundup-of-cloud-computing-forecasts-and-market-estimates-2015/

Competitive Insight. 2006. " Software as a Service Predictions". April 2006.

Cunsolo, V. D. et al. 2009.Volunteer Computing and Desktop Cloud: The Cloud@Home Paradigm. In IEEE International Symposium on Network Computing and Applications, NCA 2009.134-139.

Dearle, A. 2007. Software Deployment, Past, Present and Future. Future of Software Engineering (FOSE´07). IEEE Computer Society.

Dinsmore, P. C. et al. 2005. The right projects done right! John Wiley and Sons, 2005. ISBN 0-7879-7113-8.

Elliot, S. 2008. Agile Project Management. Seminar paper in Seminar on Current Trends in Software Industry. University of Helsinki.

Foley, J. 2008. Private Clouds Take Shape. Information Week 8/7/2008. (IW 2008) Available: http://www.informationweek.com/private-clouds-take-shape/d/d-id/1070793 .

Furht, B. & Escalante, A. (eds). 2010. "Handbook of Cloud Computing" Springer 2010 (637 pages) Citation. ISBN:9781441965233.

Gens, F. 2008. Defining "Cloud Services" and "Cloud Computing". IDC Exchange September 23rd 2008. Available: http://blogs.idc.com/ie/?p=190 .

Goff, S. A. 2008. What is a PM Methodology? A Search for Efficiency, Consistency, and Performance. Available: http://www.maxwideman.com/guests/pm_methodology/what_is.htm .

Hai, H. & Sakoda, S.2009. SaaS and Integration Best Practices. Fujitsu Sci. Tech. J. Vol 45:3. 258-264.

Hill, R. et al. 2010.Guide to Cloud Computing. Principles and Practice. Springer-Verlag. London.

Howes, N. R.2001. Modern Project Management. Successfully Integrating Project Management. Knowledge Areas and Processes. American Management Association Press, New York.

Kanaracus, C.2014.Saas predictions for 2014. PCWorld Dec 23, 2014. Available: http://www.pcworld.com/article/2082820/saas-predictions-for-2014.html.

King, R. 2014. "Pivotal's head of products: We're moving to a multi-cloud world". Between the Lines June 19, 2014. Available: http://www.zdnet.com/article/pivotals-head-of-products-were-moving-to-a-multi-cloud-world/

Lester, A. 2014. Project Management, Planning and Control: Managing Engineering, Construction, and Manufacturing Projects to PMI, APM, and BSI Standards, Sixth Edition Butterworth-Heinemann (593 pages). ISBN:9780080983240

Mc Connell, E. 2010. Project Management Framework: Definition And Basic Elements. Available: http://www.mymanagementguide.com/project-management-framework-definition-and-elements.

Mell, P. & Grance, T. 2011. The NIST Definition of Cloud Computing. National Institute of Standards and Technology. Information Technology Laboratory. Computer Security Division. Gaithersburg, MD 20899-8930.

Meyer, C. B. 2001. A Case in Case Study Methodology. Field Methodology 13: 329. Available: http://fmx.sagepub.com/content/13/4/329.

Minick, E. et al.2006. Application Release and Deployment For Dummies now Available! https://www.ibm.com/developerworks/community/blogs/greenelk/entry/application_release_and_deployment_for_dummies_now_available?lang=en

Morrison, J. & Brown, C. 2004. Project management effectiveness as a construct: a conceptual study. South African Journal of Business Management 35:4. 73-94.

Murray, A. C. 2009. There´s No Such Thing As A Private Cloud. Network Computing 1/9/2009. Available: http://www.networkcomputing.com/cloud-infrastructure/theres-no-such-thing-as-a-private-cloud/a/d-id/1075465 .

Mäkilä, et al. 2010. How to Define Software-as-a-Service – An Empirical Study of Finnish SaaS Providers. DOI: 10.1007/978-3-642-13633-7_10 Conference: Software Business - First International Conference, ICSOB 2010, Jyväskylä, Finland, June 21-23, 2010.

Mäntylä M. V. & Vanhanen J. 2011. "Software Deployment Activities and Challenges – An Industrial Case Study of Four Software Product Companies", in Proceedings of the 16th European Conference on Software Maintenance and Reengineering, (CSMR), s. 131-140.

Nokes, S. 2007. The Definitive Guide to Project Management. 2nd Ed. London (Financial Times / Prentice Hall). ISBN 978-0-273-71097-4.

Normand, A. 2015. Software Development Lifecycle. The University of Melbourne. Australia.

Pallis, G. 2010. Cloud computing: the new frontier of internet computing. IEEE Internet Computing 5. 70-73.

Papke-Shields, K. et al. 2010. Do project managers practice what they preach, and does it matter to project success? International Journal of Project Management 28. 650–662.

Pendharkhar, P. C. et al. 2008. An empirical study of the Cobb–Douglas production function properties of software development effort

PMBOK. 2010. A Guide to the Project Management Body of Knowledge. 27-35. Available: http://www.pmi.org/PMBOK-Guide-and-Standards.aspx .

PMI. 2010. "What is Project Management? Project Management Institute". Available: http://www.pmi.org/ . Retrieved 2014-06-04.

Rao et al. (2011). Implementation of SaaS in CC environment. International Journal of Computer Science and Telecommunications Vol. 2 Issue 8 November. 98-101.

Saugatuck Technology. 2013. Saugatukck´s Cloud Survey Data Reports: Hang On for Total Cloud Impact. Available: http://blog.saugatucktechnology.com/saugatucks-cloud-survey-data-report-hang-total-cloud-impact/

Schwaber et al. 2007. ´The Truth About Agile Processes`. Forrester Research. 5-6.

Subashini, S. & Kavitha, V. 2011. Review: A survey .n security issues in service delivery models of cloud computing. Journal of Network and Computer Applications Vol 34 Issue 1, January. 1-11.

Taylor, C. W. 2004. Improving grid behavior. IEEE Spectrum. 40-45.

Tekrati. 2006. "SaaS Soars, Challenges Status Quo in Asian Software Markets, Says Springboard Research", May 16, 2006, Springboard Research.

Tellis, W. 1997. Introduction to case study [68 paragraphs]. The Qualitative Report [On-line serial], 3(2) July. Available: http://www.nova.edu/ssss/QR/QR3-2/tellis1.html

Tellis, W.1997. Application of a Case Study Methodology, The Qualitative Report, 3(3) September. Available: http://www.nova.edu/ssss/QR/QR3-3/tellis2.html?ref=dizinler.com.

Tapscott, D. 2008. Grown Up Digital. McGraw-Hill. 384 pages.

Tyrväinen, P. et al. (eds.).2010. Software Business. Proceeding in First International Conference, ICSOB 2010, Jyväskylä, Finland June 2010.

UCISA ITIL. 2015. A guide to release and deployment management. Universities and Colleges Information Systems Association. https://www.ucisa.ac.uk/

Velte, A. T. et al. 2010. Cloud Computing: A Practical Approach. Mc Graw-Hill/Osborne. 2010 Citation. 354 s. ISBN-9781439834534.

Wateridge, J. 1995. IT Project Business: a basis for success. International Journal of Project Management, Vol. 13 No.3. 169-172.

Weinhardt, C. et al. 2009. Cloud Computing - A Classification, Business Modesls, and Research Directions. Business & Information Systems Engineering 5. 391-399.

Wysocki, R. 2014. Effective Project Management: Traditional, Agile, Extreme, 7th Edition. John Wiley & Sons (768 pages) ISBN:9781118729168

Yang, C. et al. 2013. Spatial Cloud Computing: A Practical Approach. CRC Press. 357 pages.

Youseff, L. et al. 2008. Understanding the Cloud Computing Landscape. Cloud Computing and Software Services. (16 pages).
Available: http://www.cs.bu.edu/~jappavoo/Resources/Papers/EBK1439803158-c1.pdf

Yin, R.1993. Applications of case study research. Newbury Park, CA: Sage Publishing.

Yin, R. 1994. Case study research: Design and methods (2nd ed.). Thousand Oaks, CA: Sage Publishing.

Xiaoyu Yang and Lu Liu (eds). 2013. "Principles, Methodologies, and Service-Oriented Approaches for Cloud Computing". IGI Global 2013 (452 pages) Citation. ISBN: 9781466628540.

West, Mike (2010-05-28). "SaaS Integration Platforms: The Looming SaaS Deployment and Support Dilemma". Saugatuck Technology.

Appendix 1: List of Figures