Kasper Ilmolahti

Developing a SSSaaS Version of a Software product: Identifying Key Features to Include in an MVP

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Industrial Management and Engineering

Thesis

10 May 2016

Helsinki Metropolia University of Applied Sciences

Author(s) Title Number of Pages Date	Kasper Ilmolahti Developing a SSSaaS Version of a Software Product: Identifying Key Features to Include in a MVP 35 pages 10 May 2016
Degree	Bachelor of Engineering
Degree Programme	Industrial Management and Engineering
Specialisation option	Global ICT
Instructor(s)	Anna Sperryn, Senior Lecturer

This thesis studies an existing, highly customisable software solution and addresses how it should be modified to create a stand-alone, standardised online-based Minimum Viable Product. The study was carried out for a Finnish software company, which specialises in advanced automated billing and invoicing solutions, mainly for clients requiring special operability from their billing systems.

This thesis was established by a need to define a new product for the case company, which has a solution, but wishes to broaden its market. The thesis focuses on the definition phase of the product development in order to create a solid foundation to start planning the new product.

This study forms part of a bigger project to develop a fully operational SaaS product. The main objective is to identify the needed features. The main outcome of this thesis is going to be a list of features to include in the product in the different stages of the development. Additionally, the thesis will document any important questions emerging during the study, which are relevant for creating a product development plan.

On the basis of this thesis, the case company can form a specific plan on the development of the Minimum Viable Product, as well as address the questions arisen. With the research conducted, and the suggestions presented, the company will be provided with a foundation to lean on, as well as find suggestions for possible definition issues.

Keywords	SaaS, IT, Product Development, Minimum Viable Product (MPV)

Tekijä(t) Otsikko Sivumäärä Aika	Kasper Ilmolahti Developing a SSSaaS Version of a Software Product: Identifying Key Features to Include in a MVP 35 sivua 10.05.2016
Tutkinto	Insinööri
Koulutusohjelma	Tuotantotalouden koulutusohjelma
Suuntautumisvaihtoehto	Kansainvälinen ICT-liiketoiminta
Ohjaaja(t)	Lehtori Anna Sperryn

Tämä opinnäytetyö tutkii olemassa olevaa, kustomoitavaa ohjelmistoratkaisua ja ottaa kantaa, kuinka siitä tulisi muokata yksittäinen, standardoitu internetpohjainen minimivaatimukset täyttävä tuote (Minimum Viable Product). Opinnäytetyö on tehty suomalaiselle ohjelmistoyritykselle, joka erikoistuu kehittyneisiin laskutusjärjestelmiin, pääasiallisesti yrityksille, jotka vaativat järjestelmiltään erikoistoimintoja.

Tämä opinnäytetyö on aloitettu uuden tuotteen kehitystarpeesta, yritykselle, jolla on ratkaisu mutta joka haluaisi kasvattaa markkinakyvykkyyttä. Opinnäytetyö keskittyy tuotekehityksen määrittelyvaiheeseen luodakseen pohjan tulevan tuotteen suunnittelulle.

Opinnäytetyö on osa suurempaa projektia, jonka tavoitteena on luoda täysimittainen SaaS-tuote. Työn tavoitteena on tunnistaa tarvittavat toiminnallisuudet tuotteen toteuttamisen kannalta. Työn tuotos on lista ominaisuuksista, jotka tulevat sisältymään tuotteeseen kehityksen eri vaiheissa. Lisäksi työ dokumentoi tuotekehityksen kannalta tärkeitä aiheita, jotka nousevat esiin tutkimuksen edetessä.

Tämän työn pohjalta yritys voi tehdä tarkan suunnitelman minimivaatimukset täyttävän tuotteen toimintojen osalta, sekä kykenee varautumaan mahdollisiin ongelmakohtiin, joita työ esittelee. Tämä työ antaa tuotekehityksen suhteen pohjan, johon nojautua, sekä tarjoaa ehdotuksia, kuinka tuote tulisi toteuttaa.

Avainsanat	SaaS, IT, tuotekehitys, Minimum Viable Product (MPV)

Contents

Acronyms

BSS	Business Support System
C2S	Click to Start
CRM	Customer Relations Management
CTS	Click to Start
ERP	Enterprise Resource Planning
MVP	Minimum Viable Product
OS	Operating System
ROI	Return On Investment
SaaS	Software as a Service
SLA	Service Level Agreement
SQL	Structured Query Language
SSSaaS	Self Service SaaS
UI	User Interface

1 Introduction

1.1 Background

The purpose of this thesis is to study an existing, highly customisable billing software solution and how it should be modified to create a stand-alone, standardised online-based product. The thesis is written in co-operation and for a Finnish software company, which specialises in advanced automated billing and invoicing solutions, mainly for clients requiring special operability from their billing systems. Thus far, the company has built its software in a highly customised fashion for the use of a broad range of companies and industries by working in collaboration with the clients to finalise the solution according to their respective needs.

This study forms part of a bigger project to develop a fully operational Software as a Service (SaaS) product. The main objective is to define the needed features, structure of the product and usability requirements. The outcome of this thesis is a specification needed to start developing the product in question, in the form of lists of features. Additionally, the aim is to document other questions emerging during the study, which are relevant for creating a product development plan. The theoretical framework includes is built on topics such as functionality, usability and connectivity.

There are a few abbreviations repeatedly mentioned throughout the thesis. The working title for the product is going to be Click-to-Start, C2S or CTS, or any combination of the previous. There is going to be continuous mention of Minimum Viable Product, MVP, which is to be the primary objective of the first cycle of the product development – in which this thesis takes place. Additionally, common terms used will be SaaS and SSSaaS. SaaS stands for Software as a Service, and SSSaaS specifies the term to Self Service SaaS. It is worth mentioning that the company has experience in SaaS, as their current solution is available as a web-based user interface, and SSSaaS is the next ultimate objective for the company.

1.2 Business Problem

The company has originally built their solution to meet the needs of an extremely challenging industry, starting with telecommunication operators and virtual operators. Therefore, their solution offers versatile and advanced possibilities to meet their clients needs. However, the main attribute of the existing product obliges tailor-made solutions; in order to implement the solution to the client's system environment, there has to be a long collaboration to fulfil the client's specific needs, in which both parties need to work and build the systems together to configure the system to a working state. Therefore, the company's long-term goal is to create an easily accessible SSSaaS product, which would be easier to sell and would possibly lead to growth of the business.

1.3 Objective & Outcome

The objective of this thesis is to answer the business problem by covering three main areas of defining the product: Features, structure and usability requirements. In this case the goal has been limited to MVP. The main focus is going to be on the functionality section of the research, since the quicker the MVP has been defined, the quicker it can be brought to the construction phase. The goal is to achieve a conceptual product definition, which explains the required functionality, using lists. The thesis can be viewed successful if it is thorough enough to define the features, as well as contemplate on the possible questions derived during the upcoming development process.

1.4 Research Methods

The study starts by defining the methods that enable the most effective background for the study. The most suitable way to approach the research is by best practices, including case study examples of other companies, how they have developed a SaaS product out of existing software, what the major challenges encountered are and how they have conquered them. Additionally, the best practices studies focus on a comparison of the general architecture of SaaS Cloud applications and traditional software, as well as study the elements of SSSaaS. While best practices compose the main part of the theoretical

background, qualitative research is going to take place as well, in the shape of action and exploratory research.

Since the process has been started in the company only recently, the development process has formed to resemble action research by defining, executing, speculating, and improving the plan. In the beginning of the development, brainstorming is an obligatory part of planning the functions and resolving the potentially arising challenges, as well as deciding the methods used to develop the software.

1.5 Structure of the Thesis

This thesis is composed of seven chapters. In chapter one, the business context and objectives of the study are introduced. Chapter two presents the research approach, research design, reliability and validity, governing the subject of the study. Chapter three analyses the current state of the company at the moment of writing this study. The chapter aims to present the current solution, explain the benefits and deliverables of the solution and takes a view of the existing functionalities critical to the objective of the study. In the summary of chapter three, the conceptual framework will be established as a conclusion of the existing practises. Chapter four focuses on establishing a solid theoretical background via case studies and best practises on comparable solutions. Chapter 5 has been solely targeted for propositions which would support the actual construction of the product for the company and chapter 6 presents the actual resolution of the study. The last chapter will summarize the whole study and suggest next steps. The progress of the thesis is presented in Figure 1: Research Design.

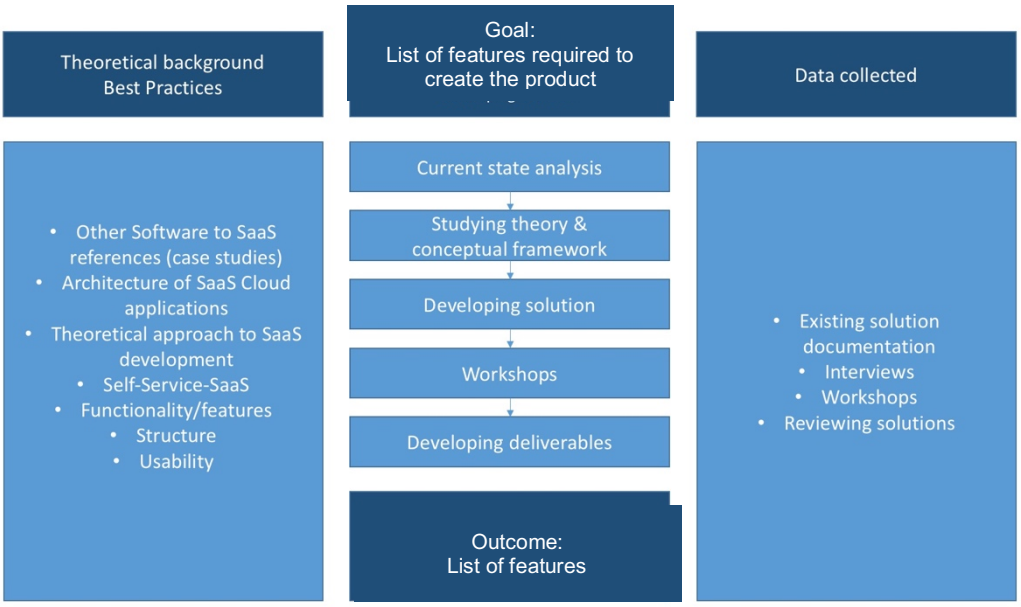


Figure 1: Research Design

Figure 1 has been composed to present the structure of the study, showing the goal, the approach and the outcome in the middle and theoretical background and data on the sides.

The next chapter describes the methods and material used in this study.

2 Method & Material

In this chapter, the methods of delivering the study explained. In order to create the logical structure to the study, the research approach is defined.

2.1 Research Approach

The approach has been selected to be agile and to allow adaptation. Therefore, it is clear to use fast-phased action research, which will let the methods and planning evolve during the study. Action research is a way to explore the idea of a practice deliberately to modify or develop something, and to perform improvements on the plan with quick iterations. It is a process that aims to change things and to develop them even better. Operational development is understood then as a continuous process that does not end, but for example, as a better approach. (Carr, W. et al 1986)

Action research aims to develop new skills or a new approach to a particular issue and to solve problems that have a direct connection to one of the practical activities. As the name implies, it is designed to implement both operational development and research at the same time. It is well suited to situations where the means of action is aimed at changing something, and at the same time to increase the understanding and the knowledge. (Suojanen 1992) Action research is shown in Figure 2.

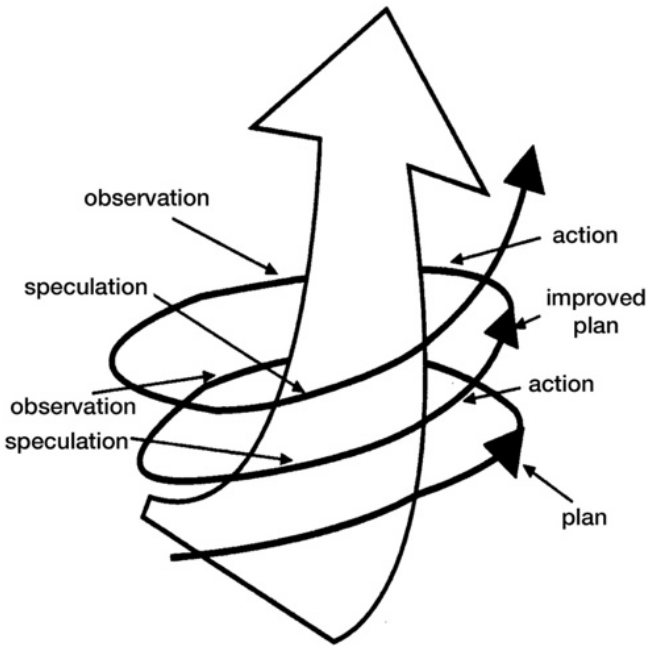


Figure 2: Action research, Heikkinen [1999]

According to Yin (2009), a research process is established by defining the objective for the research. Therefore, the objectives have been set and the business problem has been described. Next the existing knowledge on the subject is analysed which leads to generating the conceptual framework for the measurements. After the generation of the framework, it is customised to be correlative for the business problem of the thesis by matching the needs of the case company. Ultimately the conceptual framework is used to construct suggestions to resolve the business problem.

Exploratory research is going to support action research by targeting challenges that will appear during the research cycle of the development. Even though the company has wide experience of software development, this particular project is relatively uncommon to them. Additionally, case studies will be reviewed to find best practices and to gain contrast for the study. Best practices will include examples of incrementally transforming application to the cloud computing environment, resilient response to possible changes, multi-tenant consideration and vulnerability management of the software.

2.2 Reliability & Validity

Research should be reliable and present justification for the information given. Reliability is needed to reduce the errors and biases in the case study. (Yin 2009:40-45) In this study reliability is covered with precise documentation from the researcher and wide data collection of the research subject. All the statements found in the study are referenced and documented, apart from the information gathered by internal research and materials of the company during the composition of this thesis.

Validity marks the success of the research method compared to the research problem at hand. In this study, validity has been obtained by having workshops and discussions with the people who are related to this particular subject. Moreover, the research literature has been selected to be appropriate and corresponding to the research problem. According to Yin (2009), validity is increased by seeing a clear cause-and-effect, which is the aim of this study. Furthermore, high validity of the research allows making generalisations based on the research. (Yin 2009:40-45). However, in this case study a generalization can be difficult, because the case company has a unique structure and is operating in an exclusive environment.

The study has had major influence by multiple workshops and open interviews. The participants of the workshops include the Chief Operations Officer of the case company, the Director of Solution Development of the case company and several consultants in areas of industrial internet and internet of things.

The next section describes the Current State Analysis performed in the case company in order to identify the features in their current solution in order to build an MVP.

3 Current State Analysis (CSA)

In this chapter, the current state of the company's solution is analysed in order to obtain information on the features on which the MVP solution to be built in this thesis is going to be based on. This is done by describing how the company's solution functions and explaining the detailed features of the system.

3.1 Current solution explanation

The current solution offers near-complete automation for billing. Traditional billing software and Enterprise Resource Planning (ERP) systems offer only limited functionality when it comes to invoicing, and when usage-based and dynamic invoicing needs continue to grow, the current systems of companies usually generate more and more need for manual labour. In business sectors where the volumes are large and single transactions may be small, the more effort the company uses per invoice, the less margin the company will receive.

The company started its journey with tele- and virtual operators where these large volumes were continuous. For example, for the company's biggest client the solution decreased the need for billing personnel by 90 percent, while decreasing the time used for composing bills by 98 percent. Additionally, the client was able to increase their revenue by 10 percent by eliminating manual mistakes.

The company's solution functions by automatically reading the data of the clients' systems and applies certain rules and regulations configured for every specific transaction type to compose the invoice automatically. These rules can be configured by any variable which gives the system flexibility which large ERP and billing software lacks. What makes the product so powerful is the connectivity to clients' existing systems, so it does not need to be an overall standalone program, more likely an accessory, which reads data from the source, and does its magic and uploads the processed data back to the system. Therefore, it does not really compete with large ERP programs such as SAP, but on the contrary, supports them.

The system can be composed to have automatic connections with accounting, banks and collection companies, with which it can automate the money flow as a whole. When

receiving the billing information from the fundamental system, the solution processes the data, composes the invoice, sends it to the collection and accounting companies, connects with the bank, and receives the payment confirmations and sends them to accounting – all automatically. This eliminates the majority of manual labour used for the process, which is still generally speaking the norm in the major part of companies.

3.1.1 Product overview

The company's solution for the user is accessed via an online portal. The portal is basically a single-page UI where the user can access different functions with tabs. Organisational structure is defined to hierarchy, where controllable organisations, groups and services or devices are listed. This portal is going to deliver the basis for the upcoming MVP. While the MVP is going to be built on a portal, specification is going to take place, in order to define the functions to be integrated to the CTS. The user view of the portal is presented in Figure 3.

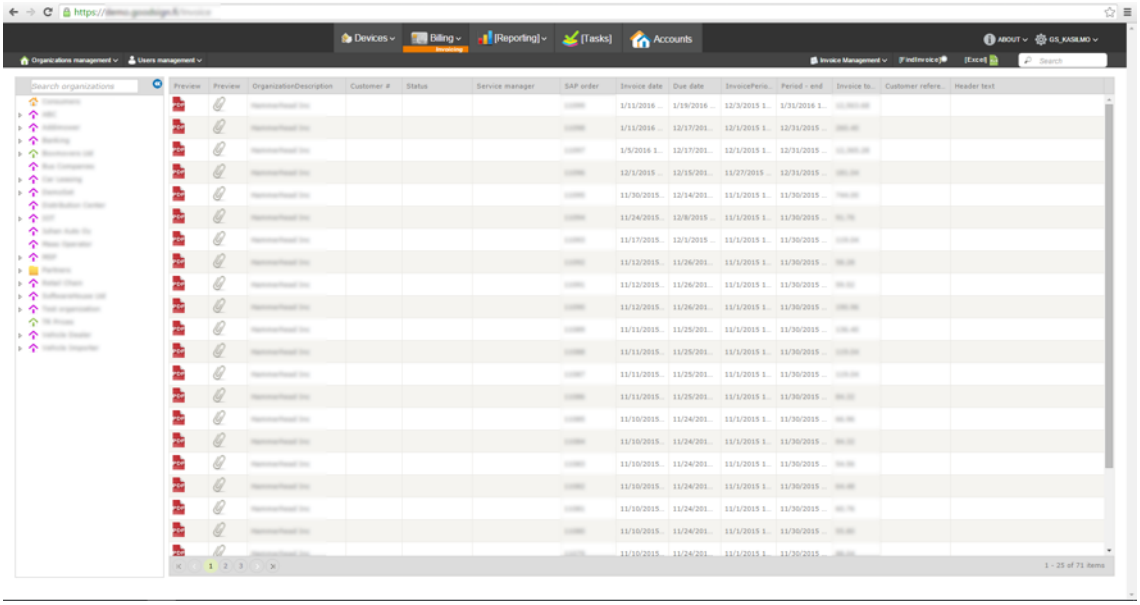


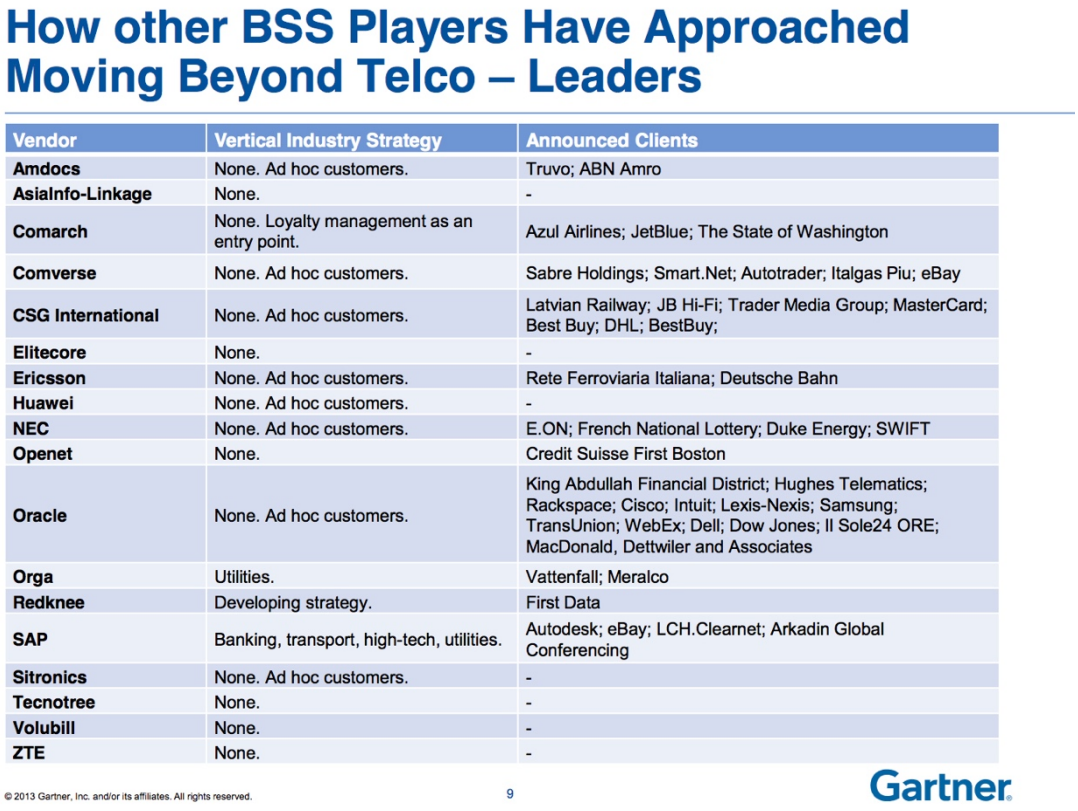
Figure 2: Current solution user interface

In the portal, the user can manage settings, examine invoices before they are sent, view reports and control accounts. All the functions happen behind the curtains as the system does the calculations by itself. However, the user can manage the rules applied for the

billing and preview the changes before sending the complete invoices. The rule configuration has to be done manually but once the rule has been set, it will continue functioning automatically. The rules are basically strips of SQL, which tells the system what to do in the event of a certain situation, and are usually constructed by the company's personnel. When taking the SSSaaS CTS in notice, there can be only little to no custom work.

3.1.2 Existing functionalities

With the background in telecom services, the company has developed a wide variety of functionality. The development has been driven by ad hoc methodology, and therefore whenever there has been a need for extra functionality, it has been created. While constantly broadening the range of services, the line of advancement has not been completely controlled by the company. This particular practice is not uncommon in the field of BSS. According to the study by Gartner (2013), presented in Figure 4, a major part of big telco-originated companies develop their products on an ad hoc basis.



How other BSS Players Have Approached Moving Beyond Telco – Leaders

Vendor	Vertical Industry Strategy	Announced Clients
Amdocs	None. Ad hoc customers.	Truvo; ABN Amro
AsiaInfo-Linkage	None.	-
Comarch	None. Loyalty management as an entry point.	Azul Airlines; JetBlue; The State of Washington
Comverse	None. Ad hoc customers.	Sabre Holdings; Smart.Net; Autotrader; Italgas Piu; eBay
CSG International	None. Ad hoc customers.	Latvian Railway; JB Hi-Fi; Trader Media Group; MasterCard; Best Buy; DHL; BestBuy;
Elitecore	None.	-
Ericsson	None. Ad hoc customers.	Rete Ferroviaria Italiana; Deutsche Bahn
Huawei	None. Ad hoc customers.	-
NEC	None. Ad hoc customers.	E.ON; French National Lottery; Duke Energy; SWIFT
Openet	None.	Credit Suisse First Boston
Oracle	None. Ad hoc customers.	King Abdullah Financial District; Hughes Telematics; Rackspace; Cisco; Intuit; Lexis-Nexis; Samsung; TransUnion; WebEx; Dell; Dow Jones; Il Sole24 ORE; MacDonald, Dettwiler and Associates
Orga	Utilities.	Vattenfall; Meralco
Redknee	Developing strategy.	First Data
SAP	Banking, transport, high-tech, utilities.	Autodesk; eBay; LCH.Clearnet; Arkadin Global Conferencing
Sitronics	None. Ad hoc customers.	-
Tecnotree	None.	-
Volubill	None.	-
ZTE	None.	-

Gartner

Figure 3: BSS companies beyond telco, (Gartner, 2013)

The company's repertoire of services has two major categories: 1) Service data processing and 2) billing. On the side, the company offers services to support these areas as well as the whole billing-supply chain procedure. In Figure 5, the functionality of the product's billing structure and service functionality of the case company is explained.

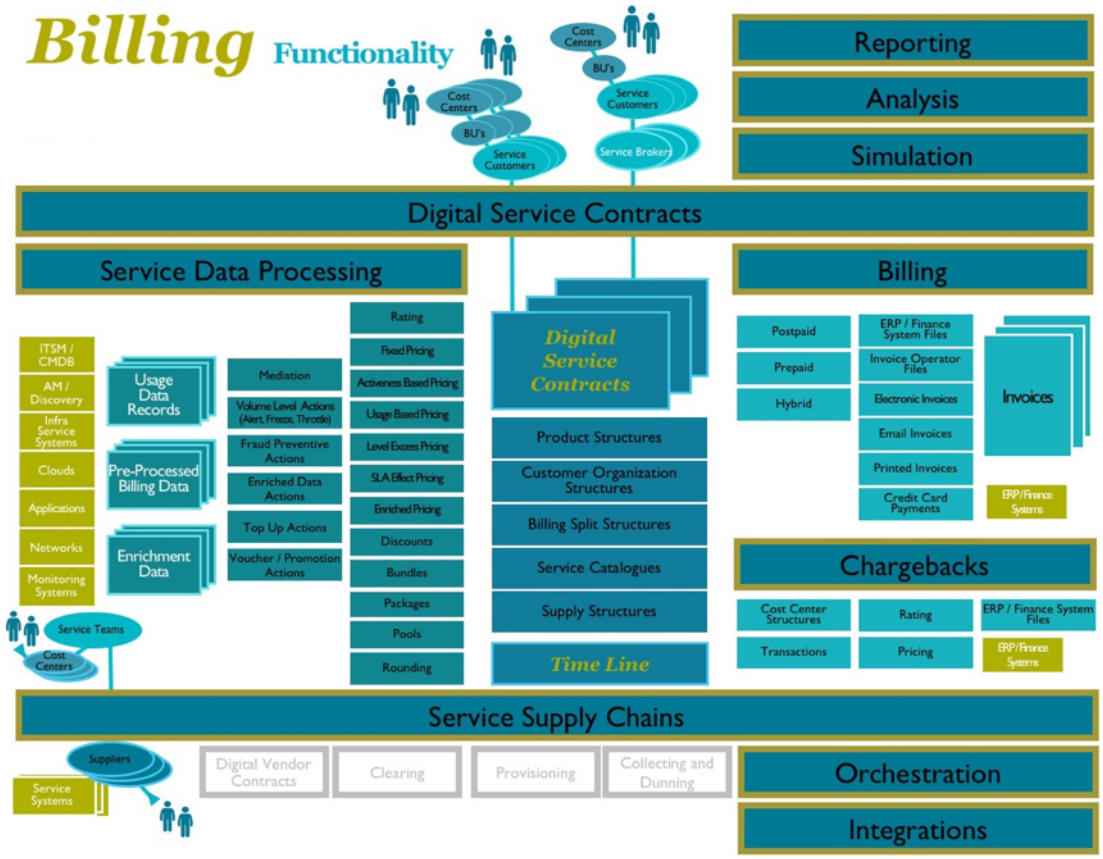


Figure 4: The company's solution functionality

Figure 5 presents the existing features of the solution. In the middle, the core services are split in to single features to showcase the key capabilities of the solution in a deeper level, while in the vertical axis, the additional features have been stated. Even though the Figure has been composed to give a view of the billing functionality, some further related features have been listed as well. For example, provisioning by means of activating services.

Variability and complexity have become more prevalent as more demanding customers expect more from service delivery and improved customer responsiveness. For example,

with the company's solution it is possible to orchestrate service intensive supply chain management and operations supporting variability in user specific requirements, locations, service levels, suppliers and varied technologies. When it comes to service billing automation, most service operating systems or ERPs may not support the required level of detail in order to support complex service contract or service level agreements, even though they manage the flow of information.

3.2 Summary

In order to comprehend the current situation, the best way to present the company's solution is that even though it is inclusive and capable of versatile functions, the implementation process to customers' systems requires plenty of resources. When the implementation process can be expedited, it will open up possibilities in improving the rates of boarding new customers as well as making the company's solution a more appealing option to them.

This current state analysis provides information of the company's solution, as well as of the features, and will be valuable when composing suggestions for the upcoming MVP. As the need is to identify key features to include in the MVP, first an understanding of the existing business is crucial.

4 Literature Review

The literature review of this study focuses on gathering relevant information to be used in building the proposition for the case company. By first reviewing a few important concepts, the literature review creates a foundation for the case studies presented in this study. The case studies introduce the structure and infrastructure of a SaaS reimbursed financial system, and take a view on how a Customer Relationship Management (CRM) company has executed a transformation from traditional software to SaaS. After the case studies, a few best practices on building a SaaS solution are discussed.

4.1 Software as a Service

The term SaaS refers to a delivery and licencing model; in which it is used by multiple tenants or customers in a centrally hosted platform. (Benlian, A. et al 2011) The concept of SaaS has been around for a while, originating from the 1990's when outsourced information technology found its way in to companies' mind-sets. Kodak, which is believed to be the first, was a pioneer in this field and quickly other ecosystems followed. And so it happened, SaaS is a subset derived from outsourcing information systems. (Loh, L. 1992) SaaS has been a successful business model because it strips the common challenges and limitations from the user or user organisation. The product consumers' usual responsibilities fade away when they do not need to control or manage the infrastructure of the system underneath, including but not limiting to storage, OS, network or servers. Not to even speak of individual application and configuration capabilities. (Mell, P. et al 2011)

To be able to achieve continuity, reliability, availability, supportability and manageability, all with a cost effective manner, is an advantageous situation. Additionally, it is important to release resources of scale and scope by utilising service providers shared resources infrastructure. (Ramachandran, M. et al 2014)

4.2 Self Service Software as a Service

Self Service Software as a Service (SSSaaS) is a SaaS service model, in which the sales process is fully automated. With the SSSaaS service model the company has planned its product easily accessible for the buyer and additionally, made the product easy to initialise. If reviewing the success of SSSaaS, the main factors are wide user base, maximised visibility, ease-of-use and extensive online support. A wide user base is necessary because generally the Self-Service versions are quite inexpensive for the user. Maximised visibility is obligatory since all of the customer acquisition has been left solely for the marketing department, and without the sales people the rate of conversions is lower. However, the biggest single reason for success or failure is the ease of use. How the customers accustom themselves with the program in the first few moments is critical. If the possible customer starts a free trial, and right from the beginning they encounter a problem or nuance of sort which they do not like – it is game over. The user will probably end up trying some other similar software, and in the Self Service field of SaaS there are little to no areas where competition between providers is minimum or non-existent. ("Saas Startup Strategy | Three Saas Sales Models".)

The service model of SSSaaS is popular among the providers because it is easy to manage, and even with changing circumstances it provides stability. Because the requirements for successful SSSaaS include having an array of material to provide help and additional information for customers, start-ups commonly struggle in this category. All because of the extensive material needed to allow the customers to service themselves.

4.3 Minimum Viable Product

The development progress is planned to start with Minimum Viable Product (MVP). The term has been commonly used in lean development methodology and has been acquainted as a best practice of software development. Frank Robinson, CEO of SyncDev, Inc, first said out loud the words and he also stated: “When I first said ´minimum viable product´ I never had to repeat myself. The words went viral right before my eyes.” (SyncDev, 2009). The term itself means developing a product with the lowest level of functionality required to be useful. Definition by Eric Ries, the man who popularised the term MVP: “The minimum viable product is that version of a new product which allows a

team to collect the maximum amount of validated learning about customers with the least effort.” (Ries, E. 2009) In Figure 6 is the definition by Jussi Pasanen (2014).

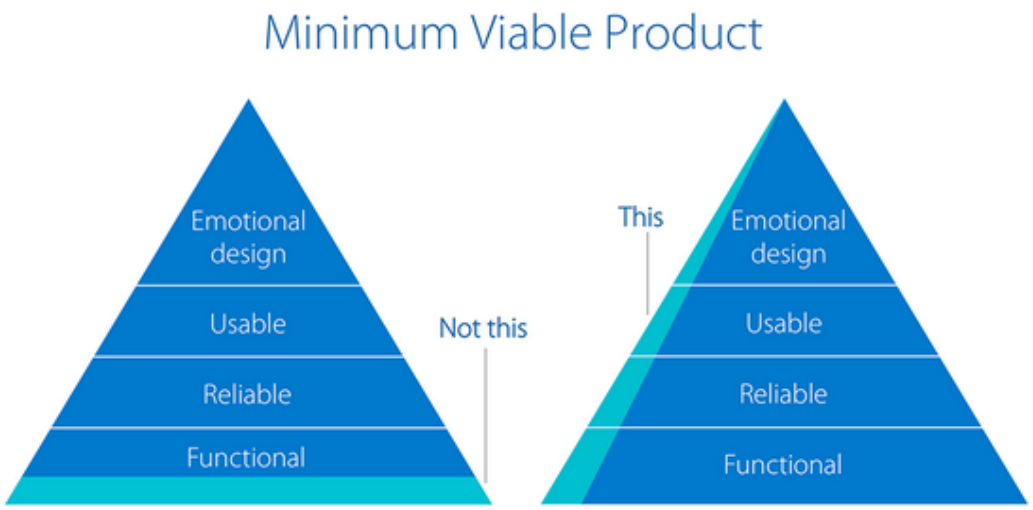


Figure 5: Minimum Viable Product (Pasanen 2014)

The traditional way of software development is to build the whole program at once. After defining the features, there will be a time-consuming phase of programming and debugging, fixing, adding features et cetera. After building the functionalities, the focus will change on usability and design, and after these stages, the program is ready to be tested and reviewed. However, this traditional methodology can be extremely time and effort consuming, and most modern software companies do not have resources to waste on pure development. “IMVU's original MVP took us six months to bring to market. That was a pretty big improvement over a previous company, where we spent almost five years before launching.” (Ries, E. 2009) Figure 7 depicts the definition of the way of MVP development by Jerry Staple. (2014)

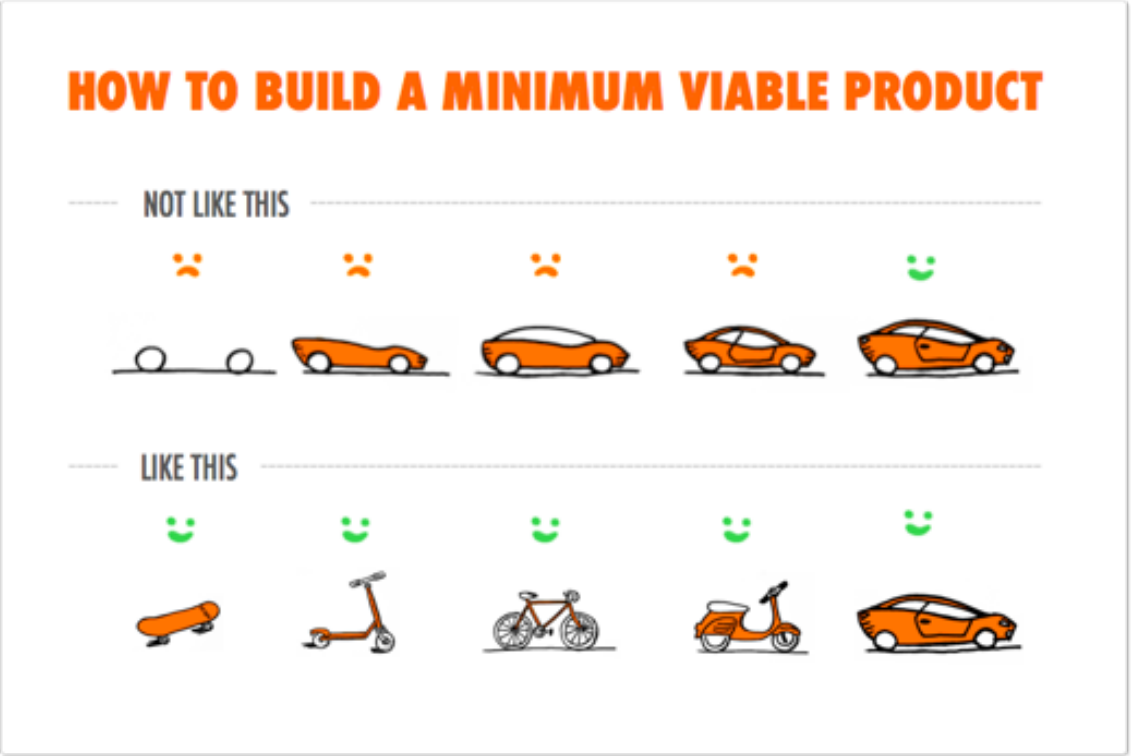


Figure 6: How to build a minimum viable product (Staple, 2014)

In the case of the case company, the objective is to build an MVP of the C2S product, with which it would be enough to start a demo with three companies. The target is to have the MVP ready within 8 months of the start of the project, having only the most crucial features required for the program to deliver value for the user. After getting the demo set up for the clients, the target is to achieve usage data and feedback in order to further improve, and to define the features to be integrated to the final product.

4.4 Case Studies

4.4.1 Case Study: The Design and Research of SaaS-Based Financial Reimbursement System

D. Li, Y. Gong, and N. Shen have composed a system architecture design for Saas-Based Financial Reimbursement system. In their report, they have dismantled the software into layers. They have come to a conclusion in which the system includes four layers: Basic layer, data layer, application layer and presentation layer. The basic layer includes the SaaS basic platform, software and hardware equipment and provides the environmental support for the operation of the system, thus being the physical basis of

the entire system. The data layer is the actual core of the system, while depending on the implementation, it can produce diversified types of data and information; including file management, relational database and other unstructured information processing. In the application layer are located the possible subsystems with various customisation types. This support layer creates basis to match the needs to the functionalities of the system. The layer's structure enables the inclusion of functions connected to task management, budget control, internal processes control, business reimbursement, online finance, etc. In addition, the layer acts as an internal connector to match interfaces among systems like external information system and the business application. These interfaces make integration between departments possible. The last topmost layer, presentation layer, provides a unified login system through a portal and a personalised page display. This is shown in Figure 8 below.

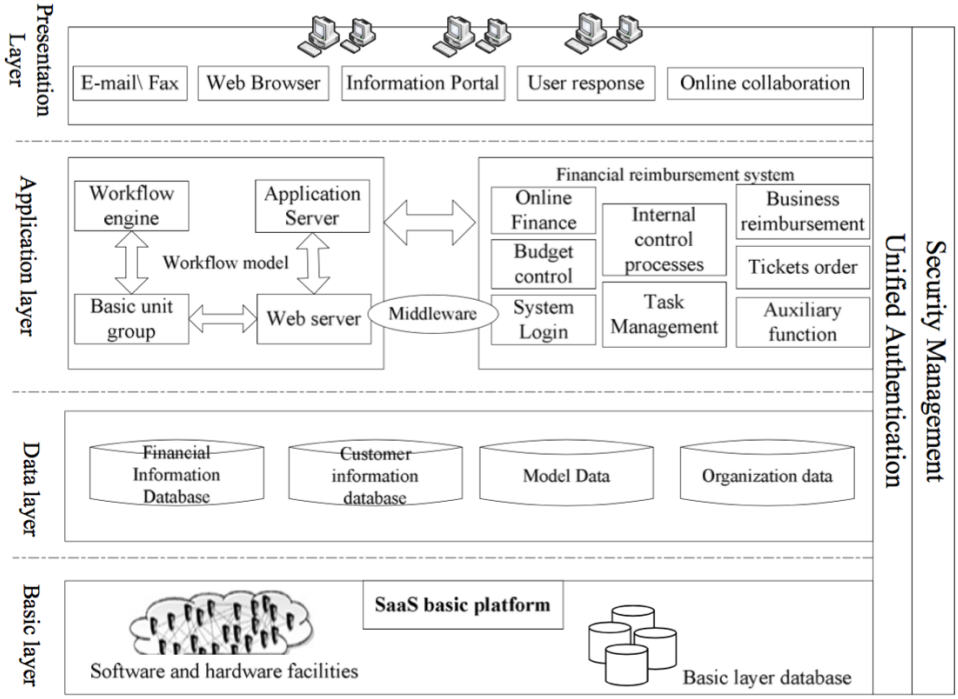


Figure 7: SaaS Layers (Luo, J. 2012)

In the case study D. Li, Y. Gong, and N. Shen have also presented a financial reimbursement subsystem functional block diagram, which they have composed. In order to achieve the functionality, they have listed aspects which are critical for the creation of

the system. The main aspects include online finance, business reimbursement, budgetary control, internal control processes and other functions. The block diagram is presented in Figure 9, which displays the structure.

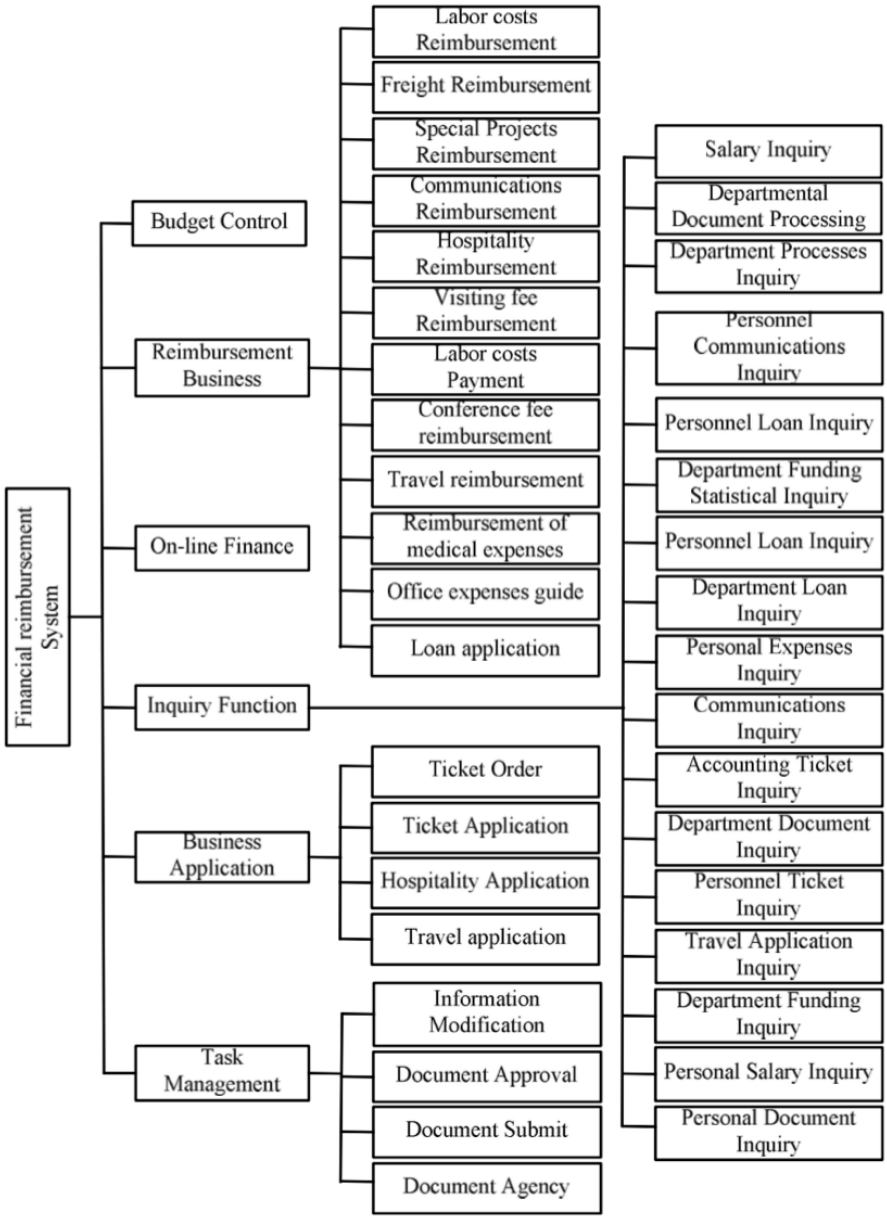


Figure 8: Financial Reimbursement Subsystem Functional Block diagram (Luo, J. 2012)

4.4.2 Case Study: Customer Relationship Management Software Provider

In another case, the company in question is a mid-size Customer Relationship Management (CRM) software company. Previously the software has been available as a web front-end as well as iPhone and iPad applications. The mobile apps originate from a need

for mobility for the customer representatives and salespersons. The industry in which the company operates, is considered to be low-turbulence industry where the provider turnover rate is from low to minimal. Additionally, while already the company has achieved its share of the markets, launching a new product should come naturally, and the optimal level of IT investment required can be quite effortlessly determined.

Because the company's software had evolved to a state in which it fulfils the main characteristics of online access with scalability, multiple devices, and no local data storage, the continuum to SaaS was defined quite naturally. (Lazaros, G. et al 2016) However, there is a risk involved. The medium-sized software development company might not be able to cover the financial investment in a case of failure, a point of which to take utterly serious. In a case of analogous event happening, the risk assessment will have to be omnipresent. (Woodard, C.J. 2012) "…there is not only the question whether it is worth to invest into a cloud solution, but also whether it is possible for the company to survive in the long run without a cloud solution…" (Excerpt from company's email memo). When contemplating the strategy for the upcoming investment, the company recognised that an aggressive move against the big providers in the industry would be onerous at least, and most likely would lead to a price war. Henceforth the company constituted a scheme with which they could exploit the leverage brought by the limited ability of imitation of competing companies. Their selling propositions specifically saw regional advantages, legal advantages, and a know-how advantage (involving a specific security algorithm). The branch of unique selling propositions stem from the fact of security being a significant element in high-security-loss environments in which clients would face considerable economic loss in case of a security attack. (August, T. et al 2014)

With the speculated propositions in mind the company decided against a public cloud offering, as it foresaw constraints probable of disturbing the current business model, respectively with probable effect on the customer base. Thus the company opted instead for a private cloud solution, as it would be predominantly compatible with its existing customised services. The company saw a probable market slot in serving companies, which do not want to operate the system themselves but do want their own private application. As a constraint in this case was the ability to host the application in a lack of data centres, which the company solved by outsourcing the hosting to a well known provider (Lazaros, G. 2016).

As the case company of this study has and targets customers, which prefer similar qualities of their systems, this case study provides an aspect which is to be taken seriously.

4.5 Best Practises & Challenges of SaaS internalisation

4.5.1 Best Practises

In the following chapter, four different best practices on challenges of environmental changes are discussed that were extracted from a survey. The subjects are going to cover: Incrementally transforming application to the cloud computing environment, resilient response to possible changes, multi-tenant consideration and vulnerability management of the software. (Park, S. 2015)

When it comes to changing the governing environment of the software, one must have clear vision of risk mitigation and the factors of success. Regarding the risks, the best way to prepare is to have sufficient enterprise support and service maturity, as well as readiness to address return on investment concerns. The key success factors are going to include having and/or building the organic and sufficient cooperative relationships with the related sources of data and institutions. Furthermore, it is vital to secure continuous accumulation and management technology, established to serve the stream of information.

In case of a transition to cloud-based environment, high costs are usually involved. Thus it is worthwhile to consider securing the funding before the transit, and further, contemplating the level of the enterprise as well as the maturity of the service. When achieving the state of analysis where multiple viewpoints on the governing matter have been discussed, a plan must be introduced in order to minimize the risks. The precondition for this phase sets in software engineering aspect via the support to enable the possible re-engineering methods for the legacy application, support tools and the planned approach for the cloud solution.

The second best practice is going to address change response resilience. Because cloud services are quite extensively connected to stakeholders and associated systems more closely than application based services, frequent requests regarding the software are going to become prevalent. Hence, such matter needs to be taken into account in order

to be able to swiftly respond to external requests concerning changes. Requirements from the software engineering aspect include service oriented traceability management and portfolio management process to integrate legacy application to the cloud lifecycle.

The third best practice concerning cloudization of an application is to consider a multi-tenant system. Because cloud environment has brought some major changes to the original infrastructure, many devices, which base on virtualization technology, compose another “infra-environment”. Accordingly, the software must be prepared to comprise services to correspond to varying contract terms depending on the tenant, even when the application functionality remains the same. It is not easy to achieve the same level of independency and performance of a multi-tenant system that were present in the case of provider being subordinate to the customer. With these concerns in mind, the way of mitigating risks is to pay attention to security and privacy, and possible decline of performance.

The fourth best practice is to pay attention to vulnerability management of the software. In order to succeed, the following factors are critical: Establishing a broad strategy to insure potential customers and ensuring high quality of service and high reliability. Because of tenants sharing the same service, an issue of a tenant can quickly escalate to be the issue of every tenant. Therefore, a solution must be prepared to control vulnerability in order to provide security for the tenants’ personal information scattered in the cloud. It is worth to mention that software vulnerability has not been significantly highlighted in traditional software development as a best practice, as risks of cloud-based software have not been present but a while. For these reasons it is worth paying attention to the engineering point of view to safety guarantee guidelines throughout the software lifecycle, verification technology and most importantly, having an analysis method for software vulnerability. (Park, S. 2015)

The communications of the ACM has approached the best practices by analysing three case companies in various areas of software industry, and composed a list of general “building blocks” for SaaS business strategy. The seven attributes are listed under three characteristics, which include industry environment, digital capital and customer requirements. The first three attributes are grouped as characteristics of the industry. These attributes are governed by the industry in which the company operates. The blocks are: degree of industry turbulence, concentration and growth rate of the industry. The turbu-

lence defines the turnover rate of the industry, in means of companies entering and leaving the industry. Concentration demonstrates the amount of competition in the market and growth is to address the change in the size of the market. (Mithas, S. et al 2013) The following two building blocks refer to the design capital of the company, in terms of internal processes and systems. For example, if a company has a low option value and high technical debt, it can be considered to be in a low-quality design capital state. However, by maybe accessing technical capabilities of other providers, it can escape the low-quality state. The last two remaining blocks – criticality of optimisation security and demand for software customisation, are categorised under customer requirements. While emerging from the obvious reasons, these attributes appear to complement the other categories as well as respects to the client's needs in cloud-based SaaS offering. The blocks are visualised in Figure 10.

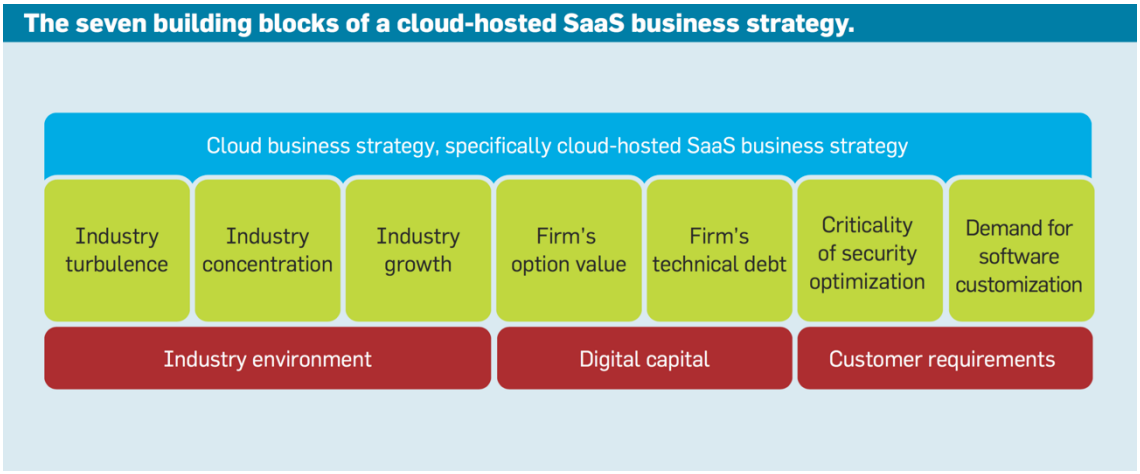


Figure 9: The seven building blocks of a cloud-hosted SaaS business strategy (Lazaros, G. 2016)

In order to create a successful product, it is important to identify the key features having an impact on the environment the product will be based in. The building blocks presented in the Figure 10 provide guidelines, which are to be kept in mind while profiling the matters leading to the listing of the features.

4.5.2 Challenges

Rather than recognising the IT environment transition toward SaaS-Cloud computing as a minor technology change, it would be more suitable to consider it as a paradigm shift.

(Cisco 2014) Correspondingly, regarding the development and operation of the software, many aspects are required to change. In the following list, Soojin Park, Sangeun Lee, and Young B. Park point out different challenges:

- **Challenges in Service Management:** In order to provide resilient and flexible service, one of the benefits of cloud-based system, a new service must be able to be created through compounding applications in the cloud.
- **Challenges in Service Level Management:** Cloud computing-based services must provide different services depending on the amount of their users pay.
- **Challenges in Incident and Problem Management:** A developed form of virtualization has a significant impact on the impact analysis of incidents and error. It results in changes in traditional 1st-2nd-3rd support organisation structure.
- **Challenges in Change Management & Release/Deployment Management:** Due to dynamics of cloud computing, risk and impact analysis becomes more complicated. It requires constructing mature change management and automatised release & deployment process.
- **Challenges in Capacity Management:** Dynamics of Cloud computing may be the factor to complicate the process of predicting and provisioning the total service capacity. Therefore, the model predicts capacity itself needs to be changed to reflect cloud computing property.
- **Challenges in IT Financial Management:** Due to flexible pricing policy of Cloud computing, more strategic decision making is necessary while pricing IT service.
- **Challenges in Capacity Management:** Dynamics of cloud computing may be the factor to complicate the process of predicting and provisioning the total service capacity. Therefore, the model predicts capacity itself needs to be changed to reflect cloud computing property.
- **Challenges in Supplier Management:** Whether the IT organisation is combined with IT service inside and out or not, it is highly necessary to integrate different IT services according to defined Service Level Agreement (SLA).
- **Challenges in Supporting Common Operations:** As the provisioning needs to be done much quicker than before, software service development process must be clearly defined and every process must be approved beforehand.
- **Challenges in Service Portfolio Management:** Service portfolio management process is a new mechanism demonstrating the new business opportunity proposed by cloud computing. Presence of service portfolio is related to every risk. It means the service system must be able to react to every possible situation.

- **Challenges in Service Validation and Testing:** It requires service validation and testing system that can support dynamics of cloud computing. However, prompt provisioning and flexible scale-up have made things easier.

Since the case company's way of product development has been mainly ad-hoc, it is important to create plans according to general best practices on cloud-transition.

When considering transition to SaaS-cloud computing environment, certain factors of determination can be demonstrated with Return of Invest (ROI). There are expected benefits and risks when transitioning to cloud computing. The following can be considered absolute benefits: the reduced service costs, preserved mass storage without additional facility investment and increased speed of service due increased flexibility, agility and adaptability of the service. However, as with all kinds of transitions, risks will be present and they can include but may not be limited to:

- possible performance decline,
- security and privacy vulnerabilities,
- enterprise-level support and service maturity, and
- most importantly – return on investment concerns.

(Figliola et al 2014)

4.6 Summary

The literature review of this study was composed to gather relevant information to be used together with the results from the Current State Analysis in building the proposition. By first reviewing a few important concepts, the literature review created a foundation for the case studies, which were presented. The case studies introduced the structure and infrastructure of a SaaS reimbursed financial system, and took a view on how a Customer Relationship Management (CRM) company has executed a transformation from traditional software to SaaS. After the case studies, a few best practices on building a SaaS solution were discussed. On this basis, the study can be continued to the identification of the key features needed to be included in a MVP. This is done in the next chapter of this study.

5 Developing the solution

5.1 Introduction

The approach towards the execution of the next chapter is straightforward, addressing one point of view at a time. Although the main subjects concerning the development of the software can be conveniently defined, the subjects will inevitably cross. However, the next part is designed to be linear, leading from one subject to another efficiently.

The execution is composed by first describing the challenge at hand, and continuing to the definition of the target group. When the target group has been defined, the MVP functionalities are defined to serve the needs of the primary target group. Afterwards, the matter at hand will cover a comparison between the existing solution and the upcoming MVP, and a more specific view to the functionalities. When the functionality aspect has been covered, the next step is to pay attention to the usability factors. Even though not widely covered by this study, it is worth mentioning that this specific part is going to be one of the major aspects in the creation of the MVP, because while the company's current solution is a SaaS service, it does not really let the user to have total control over the system. Some functionalities even require programming language to use, whilst other functionalities can be hidden or locked. Therefore, it is vital for the success of the MVP to be able to define the required development needed to achieve the real SSSaaS.

The next part is continuously parallelised with the best practices provided by the literature review in order to find a best possible solution for the challenges of the development. Accordingly, if new information or challenges arise during the execution, the matters are dealt with. Because the project in the company is ongoing, it is improbable that no new information would surface during the planning phase.

This chapter includes the key features defined by the objective of this study and therefore includes the conclusions, which will also be summarised in the last chapter.

5.2 Target group

The process will start by defining the MVP target group. Even though the company's solution can be versatile and adapted to a wide area of businesses, in order to build a viable MVP, the target group will have to be defined to offer the maximum usability for the demo clients while of course minimizing the effort needed for the development. While the functionalities within the company's solution exist to cover the majority of the cases, the strengths of the system are covering large volumes and small transactions. Therefore, the MVP target group could be found in the software industry. In the software industry it could be also viewed as a major improvement to current systems because the needs of the billing systems can be versatile: Continuous charges, devices, usage-based charges, consulting, tickets and what else. If there would be a need to start from something simpler, other possibilities could be leasing and rental companies, personnel leasing, storage, or even assisted living.

When starting to develop the MVP, the objective is to first build a fully operational product but with limited functionality. To start with an extremely basic product is advantageous in many aspects, for example, if starting with ambitious goals, the time required to build the complete product will be manifold compared to the MVP. With the basic product, which can be launched as beta, user information can be gathered at a much earlier state. Additionally, the received feedback can help to further develop the product and will support the decision-making when figuring out the additional features.

Therefore, the primary target group for the MVP is an audience with minor requirements for their billing systems, but who would still benefit from the product in question. After getting the feedback from the first target group and getting confirmation that the product is useful, a second round of definition will have to take place. The product will have another development cycle and with the added features, it is supposed to serve the needs of a more complicated industry.

When approaching the question of a primary target group via the aspect of service deliverability and coverage, it is clear that the target group should be selected according to who would benefit the most of the strengths of the case company's product. As having a background in telecommunications, the company is extremely efficient in adapting to a

high volume of transactions and complex rules surrounding the billing automation. Accordingly, the scope of potential customers has been defined to include, but not limited only to:

- Data Centers
- Cloud Operators
- Software Houses
- Service Contract Billing –based business

In order to argue the advantages a customer candidate could gain by using the case company's solution, data centre operations are explained. Because of the administrational requirements of data centres, it is a natural pick as a possible customer segment candidate. Data centres have a wide range of operations, which make them challenging to administrate, at least with a single solution. Data centre operations include aspects of infrastructure operations, security, power and cooling and management. Infrastructure covers installing, maintaining and monitoring and updating network, server and storage resources. Additionally, on behalf of management, tasks will be introduced relating to creation, enforcement and monitoring of policies and procedures within data centre processes. Besides these operations, data centres have exceedingly high security standards tied to tools and technologies that ensure physical and logical security – on and off premises. ("What Are Data Center Operations? - Definition From Techopedia")

5.3 Functionalities

When defining the functionalities of the MVP, it is important to examine the functionality which is ready or near-ready to be implemented in order to maintain the budget and maximise the return of invest in the product development. This following subchapter will introduce a comparison between the current solution and the upcoming product as well as a list competitive advantages. Finally, before moving to usability factors, connectivity aspects discussed.

5.3.1 CTS vs. original solution

It is clear that the CTS, or in this case, MVP needs to be constructed with limited resources – which will mean limited functionality compared to the original solution. One of the challenges will include connectivity, which will be covered in Chapter 5.3.4.

The current solution has been composed to work as an efficient and comprehensive tool to manage and automate complex functionality. In the MVP the functionality will be greatly limited while still carrying out complete tasks. One could see the MVP as a so-called “funnel” which would restrict the operability to ease the integration process while still performing the governing actions in the solution-based system. The current solution functionality is presented in Figure 11. The Figure has appeared in this study earlier, in chapter 3.1.2, where it has been explained.

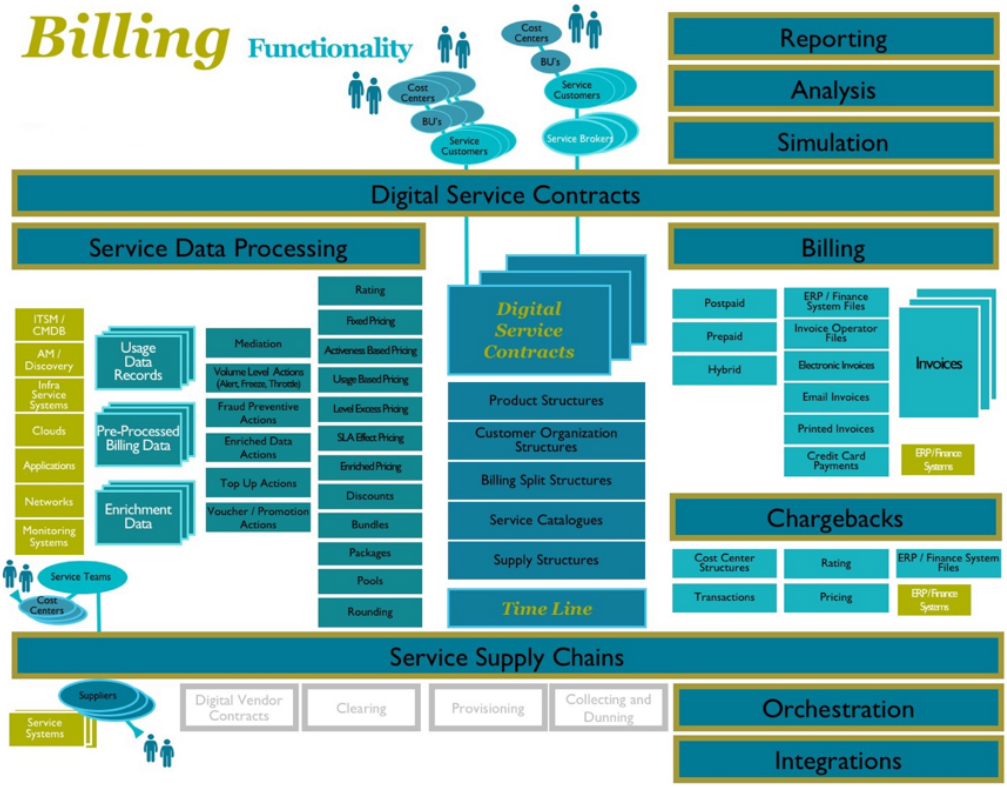


Figure 11: Current solution functionality

Although the current solution has been constructed to support a range of features, inevitable limitations will have to be made.

5.3.2 Features

In the next Figure 12, the list of features have been composed, as well as possible way to compose the various stages of the MVP presented. The information for the Figure was gathered by the author of this study. It is founded on the literature review and an analysis of the existing solution functionality, and composed in collaboration with a colleague in the case company.

Figure 12: C2S List of features

In Figure 12, a preliminary confinement regarding the features is presented. The standalone features have been listed under categories (dark blue) in order to keep track of the aspects of the product. The green colour represents the first development cycle of the MVP and the yellow represents the additional features to be implemented in the second development cycle. As the main objective remains to create a functional standalone product right from the first version, the preliminary features have been categorized by the governing functions. The information defining the functionality has been composed out of internal market and competition analyses and the resolution has been aimed to support the existing functionality combined with the available resources, which can be used in product development.

5.3.3 Competitive advantages

One of the company's major advantages is a flexible rule engine, which allows creating sophisticated automated functionality. With this particular rule engine, it is possible to perform complex operations in a fraction of the time compared to manual labour. The basic idea of the rule engine is explained in Figure 13.

The rule engine delivers profitability in administrational challenges, where tasks that require computing and/or manual labour. The engine uses data delivered from the source systems connected, creates association between the data and the corresponding rule, and executes a function based on the rule. But what makes the engine so powerful, is that it can be defined to deliver different outcomes or reports to different interest groups.

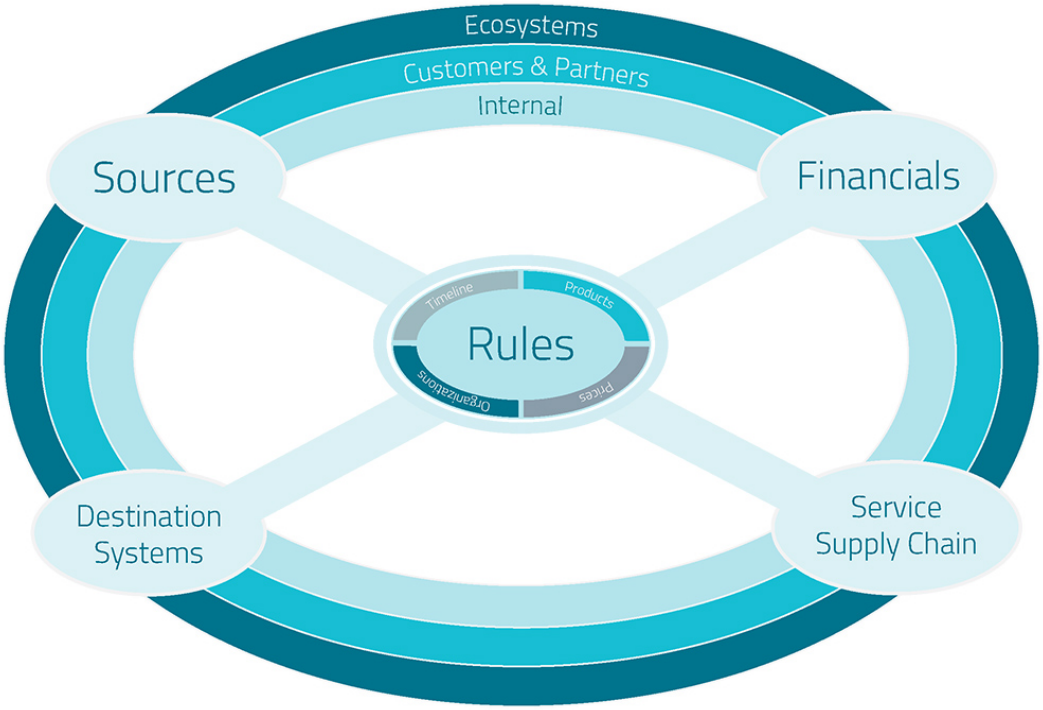


Figure 13: Rule engine basic idea

Another advantageous feature of the company's solution is mobility. The data inputs could be made with for example SMS or via mail. This could broaden the range to cover networks of employees, including companies in construction, delivery, cleaning, maintenance, assisted living and real estate management. For example, if a cleaning company with 20 employees does jobs around the city, they will all have to report their work to an administrative person. Then the person lists the jobs done, writes invoices, and sends them. What if the employees on the job could just send a simple SMS to the system,

defining the client, work done and traveling distance for example? The system would keep record of the work done and automatically create and send the invoices. However, this task management workflow and service call functionality needs to be adjusted to suit the MVP target customer group demands.

5.3.4 Connectivity

The current solution has generally been implemented in a way that through different pathways it can fetch the data required for the execution of the function at hand. Yet, that will not be possible in the case of the MVP. This is a crucial issue, which needs to be comprehensively considered. Currently, the main suggestion is to use certain pathways to upload the data from the source systems, where the MVP can read the data in and utilise it in an appropriate fashion.

The connectivity raises a challenge in a way that when there is a system that requires a cluster of data in order to perform, further specified to numerous amounts of separate lines, manual input will be out of the question. The suggestion at hand is to create a data input wizard, which would accept a range of formats and with which it would be possible to generate an automatic format switcher. The data input wizard would read the material at hand, from which the user would have to specify the corresponding fields for the system. While the data output of source systems remains somewhat static, it would be possible to read the future data inputs without the need for human interaction.

5.4 Usability

When building a self-service product, it is essential to achieve a stance of usability with which the user can feel comfortable. As mentioned at the start of the study, the solution has been mainly built ad-hoc, adding features in response to demand. Because of this development, some of the functionality can be hidden or separated over the UI and menus. Therefore, it is crucial to build the product so that the user can personally create the required rules, objects and functions and connect them easily to each other, without the need for technical support from the company. However, currently there are undergoing development projects, which are aimed to resolve these issues with the solution. Therefore, the features can be easily integrated to the upcoming product as well.

6 Conclusions

This chapter concludes the overall findings of the thesis. Afterwards, further suggestions for the case company will be made in the form of next steps.

6.1 Summary

The study started by analysing the current state of the case company as well as its current solution. The result was that while the solution is versatile and can perform a wide variety of functionality, the company's current problem lies within the challenges of implementation. Therefore, the company is inclined to create a product which would not only allow the company to enter new markets but to improve the existing solution, as well.

Next, by defining the methods used to approach the business problem, the structure of the thesis was composed. The primary research method was selected to be action research, due to its applicability for the business problem at hand. The methods of action research made it possible to further specify the business problem and objective as well as take notice of the arising issues. From that conclusion the study started to approach the business problem by first examining the current solution and its features and continuing to explore existing relevant literature. Additionally, the literature review explained some terminology crucial for the study.

The literature review together with the results of the CSA composed a foundation on which the suggestions found in Chapter 5 were built on. In Chapter 5.3.2, the features have been listed, as well as suggestions for the functionality had been established and layered to support the agile development methods. Additionally, other subjects were addressed which need to be considered before the actual construction phase of the upcoming SSSaaS product. The subjects included deliberation in functionality, connectivity and usability.

In the end of this chapter, in Figure 15, the suggested product development structure is presented, from planning to the completed first version of MVP.

6.2 Next Steps

On the basis of this thesis, the case company can form a specific plan on the development of the MVP, as well as address the questions arisen. With the research conducted, and the suggestions presented, the company is provided with a foundation to lean on, as well as find suggestions for possible definition issues.

The next physical step for the company is to form a plan covering the development, followed by defining the actual target group for the product. This thesis has presented possible options for the target group, which will hopefully help in making of that choice. After the definition of the target groups, the actual functionality factors can be contemplated. If these factors have been carefully taken into notice, the development itself can be straightforward, and follow the usual resourcing customs of the company.

When starting to develop the MVP, the objective is to first build a fully operational product but with limited functionality. To start with an extremely basic product is advantageous in many aspects, for example, if starting with ambitious goals, the time required to build the product will be manifold compared to an MVP. With the basic product, which can be launched as beta, user information can be gathered at a much earlier state. Additionally, the received feedback can help to further develop the product and will support the decision-making when figuring out the additional features.

Therefore, the primary target group for the MVP will be an audience with minor requirements for their billing systems, but who would still benefit from the product in question. After getting the feedback from the first target group and getting confirmation that the product is useful, a second round of definition will have to take place. The product would have another development cycle and with the added features, it is supposed to serve the needs of a more complicated industry, such as software business. In this second phase, the program will have added features to be used in for example licensing, periodical invoicing, SaaS and consultation services.

At this point, after configuring the product to match the needs of more complicated industries, the final phase is ready to take place. The final functionality is then decided for the use of the wider target group, the service packets are defined, the monetizing plan

is done and configurability aspects are considered. Figure 15 shows the progress of product development.

Defining the MVP	Executing the MVP	First Usage Testing & Development	Second Incarnation	Final Definition	Complete Product
- Simple	- "Usable Product"	- Learning From Phase 1	- Better Product	- Fixing the Faults	- Launch
- Limited Functionality	- Learning	- Fixing the Faults	- More Challenging Industry	- Adding Missing Functions	- Marketing
- For Basic Industry	- Collaborative test users	- Adding Missing Functionality	- Increased Data Flow Through System	- Widening the Target Group	- Continuous Improvement
- Functionality/ Usability/ Effort	- Usage Data	- More Demanding Target Group	- More Critical Test Users	- Final Functionality	
	- Next Improvements	- More Features	- "Pressure Test"	- Preparation	
	- Next Additions		- More Data		

Figure 10: Progress of the Product Development

The information for Figure 15 was gathered by the author of this study, with a foundation built by the literature review and analysis of the existing solution, and composed in collaboration with a colleague.

6.3 Reliability and Validity

Finally, in this last subchapter, based on methods provided by Yin (2009) the reliability and validity of the study is tested.

The material used to conduct this study consists of data from multiple sources. Sources included encase discussions and interviews with persons in operative positions of the case company. The data used has been gathered out of trusted sources and from internal material of the company. Additionally, the material used was inspected to provide relevance and reliability.

The findings of the study have been presented to the case company personnel in order to test the validity. Though the evaluation may not be perfect, due to the lacking measurements of the information flow, the findings can be relied on. A critical examination of the data provides validity, and by continuously comparing the data gathered to the business problem, the relevance was demonstrated. Furthermore, it is necessary to mention that the person conducting the study was an employee in the case company during the research period, which might have affected some of the results.

As a summary, the thesis project has been a fascinating way not only to get acquainted with the company and its procedures, but to learn about conducting a research as well. Despite this being the first major study done by the writer, learning the ways to produce research will be valuable in the future, without a doubt.

References

Carr, W. & Kemmis, S. 1986. Becoming critical: Education, knowledge and action research. London: Falmer

Suojanen, U. 1992. Toimintatutkimus koulutuksen ja ammatillisen kehittymisen välineenä. Loimaa: Finn Lectura Ab.

Heikkinen, Hannu L.T. ja Jyrkämä, Jyrki 1999. Mitä on toimintatutkimus? Toimintatutkimuksen perusteita ja näköaloja. Jyväskylä: Atena

Yin, R. K. (2009). Case study Research; Design and methods. Sage Publications, Inc. Fourth edition.

Figliola, P.M., Fischer, E.A.: Overview and Issues for Implementation of the Federal Cloud Computing Initiative: Implications for Federal Information Technology Reform Management (February 3, 2014), http://fas.org/sgp/crs/misc/R42887.pdf

"Lessons Learned: Minimum Viable Product: A Guide". Startuplessonslearned.com. N.p., 2009. Web. 31 Mar. 2016.

Benlian, A., Hess, T. 2011: Opportunities and risks of software as a service: Findings from a survey of IT executives. Decision Support Systems 52(1), 232–246

Loh, L., Venkataraman, N. 1992: Determinants of information technology outsourcing: A cross sectional analysis. Journal of Management Information Systems 9(1), 7–24

Mell, P., Grance, T. 2011: The NIST definition of cloud computing. NIST special publication. 800-145, 1–3

Ramachandran, M., Chang, V. 2014: Modelling financial SaaS as service components. In: Chang, V., Wills, G., Walters, R. (eds.) Emerging Software as a Service and Analytics, pp. 13–20. SCITE Press, Portugal

"What Is A Literature Review? - In Depth Evaluation Of Previous Papers". Explorable.com. N.p., 2016. Web. 31 Mar. 2016.

"Saas Startup Strategy | Three Saas Sales Models". Chaotic-flow.com. N.p., 2016. Web. 22 Feb. 2016.

Acharjya, D. P, Satchidananda Dehuri, Sugata Sanyal 2015. Computational Intelligence For Big Data Analysis. Springer: Singapore

J. Luo (Ed.) 2012: Affective Computing and Intelligent Interaction, AISC 137, pp. 861–868. Springer-Verlag: Berlin Heidelberg

Communications of the ACM, Lazaros Goutas, Juliana Sutanto, Hassan Aldarbesti, January 2016

Woodard, C.J., Ramasubbu, N., Tschang, F., and Sambamurthy, June 2012: V. Design capital and design moves: The logic of digital business strategy. MIS Quarterly 23, 2, 537–564.

August, T., Niculescu, M.F., and Shin, H. Sept. 2014: Cloud implications on software network structure and security risks. Information Systems Research 25, 3, 489–510.

Soojin Park, Sangeun Lee, and Young B. Park 2015: Best Practices in Software Engineering for SaaS-Cloud Era, Springer-Verlag: Berlin Heidelberg

Mithas, S., Tafti, A., and Mitchell, W. June 2013: How a firm's competitive environment and digital strategic posture influence digital business strategy. MIS Quarterly 37, 2, 511–536.

Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013-2018, http://www.forbes.com/sites/louiscolumbus/2014/03/14/roundup-of-cloud-computing-forecasts-and-market-estimates-2014/ (February 5, 2014)

What Are Data Center Operations? - Definition From Techopedia". Techopedia.com. N.p., 2016. Web. 28 Mar. 2016.