



APPLICATION DEVELOPMENT IN THE ASSET MANAGEMENT STARTUP: CUSTOMIZATION VS BUILDING A CORE PRODUCT

Lappeenranta–Lahti University of Technology LUT

Master`s Programme in GMIT, Master`s thesis

2021

Tatiana Yaldaie

Examiner(s): Professor Ajantha Dahanayake

Professor Ville Ojanen

ABSTRACT

Lappeenranta-Lahti University of Technology LUT
Your school: LUT School of Engineering Science
Degree programme: Industrial Engineering and Management

Tatiana Yaldaie

**TOPIC: Application development in the asset management startup:
customization vs building a core product.**

Master's Thesis

2021

85 pages, 20 figures, 9 tables

Examiners: Professor Ajantha Dahanayake, LUT; Professor Ville Ojanen, LUT

Keywords: Asset Management, Digitalization, Personalization, Profitability, Management System, Mass customization, Quality function deployment

A substantial number of customers are now requesting that companies produce more versatile products in order to meet their needs. Depending on the development stage, companies tend to choose the most suitable way of operation: startup companies usually offer more customized solutions, whereas more established companies try to build a core product that would satisfy the majority of needs. At the same time, companies are aiming to gain and retain customers.

The research is based on a case study. The purpose of this research is to determine which strategy is preferable for the company in question: customization or building a core product. In addition, it is explored what is the optimal strategy to gain and retain customers for the startup in asset management and for a SaaS company operating in the field of asset management. The outcome of this thesis work is the proposition of the most feasible operating strategy for the company in question. Moreover, the author suggests the most optimal way to gain and retain customers for a startup asset management company and for a SaaS company operating in property asset management.

Table of contents

1	INTRODUCTION.....	4
1.1	Research problem and research questions.....	4
1.2	Research method	5
1.3	Research, its scope and structure	6
2	LITERATURE REVIEW	8
2.1	Definitions.....	8
2.2	Role of company departments in gaining and retaining customers.....	26
3	THEORETICAL FRAMEWORK	33
3.1	Profitability assessment.....	33
3.2	Methods utilized in the study	36
3.2.1	Delphi method	36
3.2.2.	Quality function deployment (QFD).....	40
3.2.3	Configuration and Customization Competency Model.....	45
4	METHODOLOGY	48
4.1	Data gathering	48
4.2	Data analysis	48
5	RESULTS.....	50
5.1	Delphi method	50
5.1.1	Background	50
5.1.2	Research questions	50
5.1.3	Method.....	50
5.1.4	The procedure for conducting the questionnaires.	53
5.1.5	Results.....	54
5.2	Quality function deployment.....	64
5.3	Strategy selection and an execution framework	77
6	CONCLUSIONS.....	84
6.1	Main Findings and Implications	84
6.2	Limitations and Future Research	85
	References	86

1 INTRODUCTION

1.1 Research problem and research questions

Historically, there has been a significant difference between software vendors' general-purpose, horizontally focused, mass market developments and the advanced, vertically oriented, custom systems designed to automate unique business operations. A disparity between the diversity of consumer needs and tech industry economies of scale may have created the difference. (Kuo, T. C., 2012) Every consumer has different needs, and only a personalized approach can meet those needs. Software vendors, on the other hand, who are trying to hold production costs down are forced to respond to customers' requests as if they are similar to each other. (Greenfield, J., 2007) Mass customization (MC) has been seen to be an important technique for closing the gap described above. It contributes to streamlined products or services while still allowing for a degree of customization in the end product or service. (Kuo, T. C., 2012)

Software as a Service (SaaS) enables software application providers with a Web-based distribution platform that permits them to support large numbers of clients with multi-tenancy networks and architecture of application sharing, helping them to benefit greatly from economies of scale. Despite the fact that SaaS applications are typically built with highly structured software functionalities in order to support a large number of customers, many customers still request feature variations based on their specific business necessities via easy customization and configuration. Because of the model that is built on the subscriptions, SaaS vendors must devise a well-thought-out plan to enable customers to configure and customize their SaaS applications without modifying the SaaS application source code. (Sun, W., et al., 2008)

Gaining bigger market share is challenging, especially in today's environment. A startup asset management company located in Finland is developing a core product that they are selling to the customers. Furthermore, the startup also adds customized features. The company personalizes the end product to match the customer's requirements. Personalization that aims to tailor the product according to the customer's needs might be time consuming and expensive (Fan, H., & Poole, M. S., 2006) that is why company is setting a goal to assess if it is more feasible to develop a more complete core product that would include many additional options for the users to choose from or it is better to keep offering the basic core product and sell additional features customized according to the customer's needs.

Due to highly competitive environment, the company is aiming to select the most feasible strategy that would allow to gain and retain customers. At the same time, software industry is evolving rapidly and moving towards SaaS business model. Company is positioning itself as a SaaS vendor, thus issues associated with customization and configuration are going to be explored. Two approaches are going to be assessed using different methods, including Delphi and Quality Functional Deployment to conclude which approach is more feasible. Delphi is a method for forming an educated consensus within a group on a complicated topic by employing a series of questionnaires sent in several rounds to collect data from a panel of selected participants on a complex subject (Fink-Hafner, D., et al. 2019). Quality Function Deployment (QFD) is a tool that aids in the translation of customer requirements into new products that actually meet their demands (Jaiswal, E.S., 2012). Finally, a framework that guides the planning and execution of the strategy is to be offered.

As the company researched in this thesis is operating in the asset management sphere and since it is positioning itself as a SaaS vendor, it was suggested by the company to investigate the most optimal scenarios strategy for the startup company, for a SaaS company and for the company in question is.

RQ1. "What is the most feasible strategy to gain and retain customers for a startup asset management company?"

RQ2. "What is the most feasible strategy to gain and retain customers for a SaaS company offering property asset management application?"

RQ3. "What is the most feasible strategy for the company in question: configuration/customization or building a core?"

1.2 Research method

Many SaaS companies are striving to gain new customers and keep the existing ones. This task is quite challenging due to the competition and quickly changing environment. The thesis in question is based on single-case. The research utilizes qualitative approach. Qualitative method helps to understand what is the best way for the company in question to gain and retain customers and what is the most feasible strategy: building a core product or configuration/customization. The case study was selected because the author was given the access to the data required for the research.

The main benefit of qualitative research methods is that they compel the researcher to dive into the

problem's complexity rather than abstracting it away. As a result, the outcomes are richer and more informative. They aid in the resolution of issues involving difficult-to-quantify factors. In general, qualitative analysis is more time-consuming and demanding than quantitative analysis. Qualitative outcomes are frequently seen as "softer" or "fuzzier" than quantitative results and it is more difficult to sum up or simplify them. (Seaman, C.B., 2008)

1.3 Research, its scope and structure

In 2000, Sutton (2000) highlighted a lack of research on the startups in general. Coleman et al (2008) offer further support for this argument. Just a few investigations into software engineering activities with a focus on startups were found in a systematic mapping studies (SMS) conducted in 2013. Furthermore, rather than forming a coherent body of information, the described findings are widely diverse and distributed through various regions. (Paternoster, N., et al., 2014) Recently, the amount of research publications on the startups has increased (Tegegne, E. W. et al, 2018). However, the research on digital asset management startups is scarce. Thus, there is a need for further studies that would be addressing startup and challenges it faces in the process of gaining and retaining customers. Moreover, the research is aiming to assess two approaches: building a core product or customization and provide some conclusions on which one is more profitable for the startup in question.

The content of this thesis paper is organized in the following way. Firstly, the introductory section explains the research problem, method used, research questions and the scope, as well as the justification of the study.

Literature review is started with provision of some definitions necessary for the understanding of the content of the paper and explanation of roles of the departments in the process of gaining and retaining the customers, as well as the concepts of configuration, customization and mass production. Chapter is concluded with the study of the role of company departments in gaining and retaining customers.

Third chapter talks about theoretical background covering the profitability analysis and methods used to assess it. The section on profitability evaluation contains information about the analytical model of three profitability ratios over time (profit margin ratio, cash flow ratio, and return on investment ratio) and gives some insights about the survival of the startups. Delphi method and Quality Function Deployment are

used to evaluate the feasibility of the strategies and specify the customers' requirements. Customization and configuration Competency Model is intended to make strategy formulation and implementation discussions about SaaS setup and customization more efficient.

Methodology would explain what data was utilized, how it was gathered and analyzed. Chapter describing the results of the research summarizes the findings and provides the answers to the set research questions. Limitations and possible future research areas are suggested in the last chapter of this thesis work.

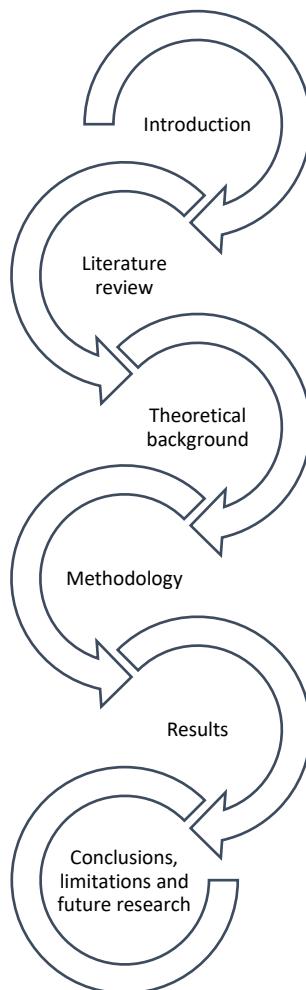


Figure 1 Research structure (Author's own creation)

2 LITERATURE REVIEW

2.1 Definitions

Assets

There are several types of assets, including human, financial, information and intangible assets. Human assets are about knowledge and competence of the employees and what kind of impact they may have on the performance of the physical assets. (ISO, T. & SC, N., 2014) IT workers who regularly solve market challenges and capitalize on business opportunities by information technology are the distinguishing attribute of a valuable human asset. These IT skills, firm-relevant IT expertise and competence are gained through a mixture of advanced preparation, on-the-job experience, and strategic leadership (Ross, J.W., et al., 1996). Investments in infrastructure, operation, maintenance, and materials all require financial assets. Data assets refer to the high-quality data and information needed to design, optimize, and implement an asset management strategy. Intangible assets include the company's reputation that may have a dramatic influence on the investments, operating strategies and associated costs. (ISO, T. & SC, N., 2014)

Asset management

Quite many definitions of asset management exist. Some explain it as a product that accepts asset information. Others may interpret asset management as something obtained from inventory information or fixed-asset systems. The above-mentioned definitions do not really explain what asset management is and how a company can benefit from it. Besides this, these definitions do not give insight into the asset management scope. Asset management can be described as a combination of various tools and process that allow the company to manage the asset base from the point of costs, contracts, support and inventory. (Galusha, C., 2001) In order to achieve its organizational strategic plan, asset management can also be described as structured activities and practices by which an organization efficiently controls its assets and asset systems, their related efficiency, risks and costs over their life cycles (ISO, T. and SC, N., 2014).

Asset management (as opposed to financial asset management) is a systematic, structured process that covers the full life cycle of tangible assets, with the underlying assumption that assets will continue to support the company's execution goals. It needs a particular level of managerial understanding and

abilities from a variety of organizational domains. (Cooperative Research Centre for Integrated Engineering Asset Management, 2008). Therefore, asset management is much more than just asset maintenance; it's a systematic approach to asset management that encompasses policy, risk measurement, sustainability, the environment, and human aspects, among other things. (Amadi-Echendu J.E., 2004)

Successful asset management involves a multidisciplinary strategy requiring synergies between conventional disciplines such as accounting, architecture, economics, arts, procurement, and information systems technologies. Asset management is a critical but challenging business practice. Simulation of asset management systems helps to model the business process and understand the ways to handle the complexities of asset management. (Frolov, V., et al., 2010)

Asset management should be decomposed into a series of procedures to help achieve the best results by handling physical objects. An asset management procedure is a collection of related activities and the chain of these operations that are required to achieve asset management objectives jointly, usually under the limits of an organizational framework and resource constraints. (Ma L., Sun Y. & Mathew J., 2007) The modeling of asset management is suggested as a way to handle the complexities of asset management by business process modeling. In general, business process modelling is a way to depict how business does its work (Davenport T.H., 2005) and is used to improve understanding and overview of business processes, reinterpret organizational complexity, recognize process vulnerabilities, apply best business practices, plan and convey new business models to appropriate stakeholders, and develop and configure information and workflow systems in areas other than asset management. (Bandara W., Gable G.G. & Rosemann M., 2005)

As for the company in question, it operates in the sphere of IT, thus it is necessary to define a concept of digital asset management. Asset management has been addressed at various levels over the years, and it is a broad term used by individuals and companies to describe the need to handle information in the modern era. They all seem to be trying to explain the same thing, from simple asset management to digital asset management, media asset management, and content management. Each of these labels, however, represents a different type of information management. An organization should choose the approach that best fits the current cultural, environmental, and financial situation based on the approach. Digital asset management (DAM) is a term that refers to the electronic management of any kind of digitally stored

data. The management of digital assets is unquestionably strategic. Digital asset management as well media asset management are used to store long-term material for archiving, storing and, most importantly, reusing. (Wager, S., 2005)

The real estate sector changes its focus from the conventional sense of the product to the incorporation of physical properties into the collection of services offered to the customer as a tangible commodity utilized as a method of reaching a primary target (Moretti et al., 2017). This new approach to the real estate brings together material and intangible products and services and shapes an entirely new commodity to be promoted. The results of the real estate development sector turn into a system that encompasses both project elements and additional services. (Baines et al. 2008) In addition, recently the management of the urban environment has displayed the growth in the complexity of physical infrastructure, as well as the large number of stakeholders and the extensive use of ICTs (ICTs) (Centre for Digital Built Britain, 2018). Physical assets can be seen as complex systems that feature tangible and intangible performances. The transition from the conventional paradigm that was seeing building as a product to the contemporary paradigm that sees building as a service makes it possible to incorporate digitization that leads to new complexity by utilizing modeling and knowledge processing and to the presence of a more efficient built environment. Such complexities pose the question of how asset management can accommodate digital-based systems and where are the most effective tools and practices to be used to capture the emerging complexity of the built environment. The study therefore triggers asset management process creativity, using current instruments and practices, merged and reshaped to achieve improved built environment efficiency. (Cecconi, F.R., et al, 2020)

Startup

The company that is used as an illustration of the case study is a Startup. Startup companies are enterprises that are still in their infancy and are struggling for survival. Most of these companies are built on the basis of genius ideas and evolve to be successful. (Salamzadeh, A., & Kawamorita Kesim, H., 2015) Startup companies aim for rapid growth. Not every newly opened company can be considered as a startup. (Graham, P., 2015)

Every day, new software startups pop up around the globe as a result of new opportunities, open technology, and venture capital (Smagalla, D., 2004). The phrase "software startups" refers to companies that are focused on developing high-tech and creative technologies that have little or no corporate

experience, with the aim of rapidly growing their business in highly scalable markets. Being a startup is typically a transient state in which a growing work experience and business domain awareness contributes to an analysis of existing operating processes, reducing intense confusion. (Paternoster, N., et al., 2014) Startups, according to Sutton (2000), are innovative and agile by nature, and are hesitant to adopt method or bureaucratic steps that might lead to inefficient procedures. Due to a lack of funding, product creation takes precedence over the establishment of rigid procedures (Heitlager, I., et al., 2007).

Despite numerous examples of success, a lot of companies collapse within two years of their launch. Failure is cause by self-destruction rather than competition. (Crowne, M., 2002) Tech startups face extreme time pressure from the industry and are subject to constant rivalry (Cusumano, M., et al., 2003; Eisenhardt, K.M. & Brown, S.L., 1998). They operate in a chaotic, constantly changing, and unpredictable setting. In order to survive in this environment, startups must be able to adapt their product to evolving consumer demands when operating with limited capital. (Sutton, S.M., 2000)

From the point of view of an engineer, software development might be difficult because startups exist in a setting where software systems are difficult to follow a prescriptive approach (Sutton, S.M., 2000; Coleman, G., 2005). If startups share certain features with related environments (e.g., small and online companies), the particular app creation background is the result of a mixture of variables (Blank, S.G., 2005).

The value of entrepreneurship can be estimated by looking at the number of small business incubators that have emerged in the last decade (Grimaldi, R. & Grandi, A., 2005). Non-startup businesses have been forced to undergo dramatic corporate and innovational renewals in an effort to act much like entrepreneurs as a result of the tide of change brought about by emerging technology (Christensen, C.M., 1997). Implementing methodologies to organize and manage growth practices in startups, on the other hand, remains a challenge (Coleman, G. & O'Connor, R.V., 2008). Several frameworks have been proposed to guide software development practices in startups, but none have proven to be effective (Coleman, G., 2005).

Startups are more likely to build software services that are clients licensed rather than products that are sold and tailored to a single client (Marmer, M., et al., 2012). This aspect is addressed by market-driven software development (Regnell, B., et al., 2001). Time-to-market is a crucial competitive target for firms

in this field, according to the researchers. Furthermore, specifications are "invented by the software company," "rarely registered," and can only be tested after the product is launched. As a result, "products that do not satisfy consumer needs" are a big source of failed product releases. To solve this problem, startups use product-oriented practices and agile teams, as well as workflows that enable them to rapidly shift their focus to the target market. As a result, many startups value team efficiency rather than presenting workers with strict guidelines. (Giardino, C., et al., 2016)

In the early stages of their growth, tech startups are product-oriented (Heitlager, I., et al., 2007). After early successes, software development and operational management become more complex (Lehman, M.M., 1980), resulting in performance degradation over time. In a nutshell, the need to create initial repeatable and scalable processes cannot be put off indefinitely (Ambler, S.W., 2002). Starting from scratch (Kajko-Mattsson, M. and Nikitina, N., 2008), startups grow over time, producing and stabilizing systems that are only improved until they are fairly mature (Crowne, M., 2002).

Startups, as Sutton (2000) points out, have limited time for skills training, thus the emphasis shifts from prescribed processes to team skills, with individuals who can "hit the ground running". The priority of entrepreneurship has been to empower the team and concentrate on methodological features of processes geared toward prototyping, proof-of-concepts, mock-ups and simulations, and checking simple functionalities. Coherent quality management and long-term planning processes are expected as startups expand. (Yoffie, D.B. & Cusumano, M.A., 1999)

Tingling (2007) explored how a company's maturity impacts method acceptance. He discusses the complexities of implementing Extreme Programming (XP) standards (Beck, K. & Andres, C., 2004) in the development process, and the need for qualified team members to completely incorporate the technique. Similarly, after six months of training the team, da Silva and Kon (2005) were able to start with all of the XP practices in place. And then, customization of activities must be applied, with procedures tailored to the startup environment (Deias, R., et al., 2002).

Lean and Agile methodologies make improvements to the production process's stability and responsiveness. Startups operate in unpredictable environments, which necessitates rapid learning through trial and error, a good client partnership, and the avoidance of spending time on unneeded features and resource fatigue. (Giardino, C., et al., 2016) Yogendra (2002) addresses consumer

engagement with product development as an important factor in promoting early convergence of market needs with technology strategies.

Since “all decisions relating to product production are trade-off situations” (Hilmola, O.P., et al., 2003), startups adapt their workflows to the evolving world in which they operate. Following the “Just do it” credo, startups usually embrace any production model that could fit to support their initial needs (Ries, E., 2011). “Many managers simply opt to adapt what they know, as their practice teaches them it is merely common sense,” Coleman and O’Connor (2008) observe. This does not, however, rule out the prospect of gathering, packaging, and transferring experience in a lightweight manner that allows for the efficient implementation of good technical practices. Startups who do not have access to highly skilled team members, on the other hand, may boost their odds of success by adopting established work practices. (Giardino, C., et al., 2016)

Startups must prioritize and filter in order to produce a product with the appropriate functionality. Most entrepreneurs do not directly use conventional Requirement Engineering (RE) practices to collect and handle specifications from an engineering perspective. Companies can increase the efficacy of requirements elicitation even with largely unfamiliar final users by implementing basic strategies such as Persona and Scenario, minimizing time-to-market. (Aoyama, M., 2005)

Mass customization

The term " Mass customization (MC)" first appeared in the late 1980s, and it can be interpreted in two ways: generally, or narrowly (Silveira, G. D., et al., 2001). Davis (1989) described MC as the ability to deliver individually developed products and services to each consumer through high process agility, durability, modularity, and reusability from a wide perspective. Multiple companies have reshaped their approaches as a result of MC technology, allowing them to detect and respond to latent consumer niches by building technological expertise to satisfy the varied demands of target consumers (Jiao, J., & Tseng, M. M., 2000; Krishnapillai, R., & Zeid, A., 2006). The aim of MC, according to Tseng and Jiao (1996), is to recognize trends of consumer needs associated with product families, as well as components' basic construction blocks, subassemblies, and modules associated with product fulfillment processes. According to Jose and Tollenaere (2005), using platforms allows important family design savings and manufacturing convenience.

Mass customization focuses on how to effectively manufacture and manage many related consumer items, taking into account and making full use of consumer demands' similarities and variations. In the car industry, this is a tried and true approach. The goal here is to create a single assembly line that can manufacture a car model that meets both mass production and customization requirements. (Krueger, C. W., 2001)

In MC, there are two popular approaches to software and product creation. They are, respectively, delayed distinctiveness and adaptable design. Modular design, a form of part design, is a standardization technique (Stevenson, W. L., 2009). Modular may also be used to describe a manufacturing system, which is referred to as a modular production system (Rogers, G. G., & Bottaci, L., 1997). Several other experiments have shown the use of such a component design in the development of software systems (Kotonya, G., et al., 2003; Brown, A., & Wallnau, K., 1998; Crnkovic, L., 2001; Crnkovic, L. & Larsson, M., 2002). Delayed differentiation is a postponement technique that is used to prevent a product from being completed until the final demands are understood. In order to successfully meet consumer demands, product differentiation occurs in the final stage. Delayed differentiation of customer needs not only achieves end-product differentiation, but also aids in lowering production costs. (Kuo, T. C., 2012)

Despite the fact that the MC approach is commonly employed in software development, some important issues have remained unresolved. The first concern is that defining consumer needs remains a difficulty. Customers are still waiting for extremely specialized applications, and software vendors are finding it difficult to satisfy their specific needs. This is a challenge that sometimes entails problems of consistency. The second issue is that reusing software is difficult, if not impossible to achieve. (Kuo, T. C., 2012) Technology software is usually not designed for reusability, according to Felice (1998). Software developers, on the other hand, are actively trying to standardize their technology in order to increase reusability and minimize costs. Finally, costs can pose a problem when it comes to component reusability and efficiency. Higher device reusability is often associated with lower user loyalty and lower prices.

Many businesses are attempting to reduce the overall expense of their computer systems in the middle of the global economic crisis (IS). Instead of buying apps, companies are gradually choosing to rent services from web-based platforms. For their sophisticated information systems, most businesses want an inexpensive "absolute investment." Hardware, coaching, implementation costs, and repair fees are all included in the overall investment. Customers, though, want a computing solution that not only meets

their needs but also saves them money, regardless of whether businesses rent or buy software. As a result, designing a software solution that meets consumer expectations while keeping production costs down has proved to be a challenge for companies. Described problem is illustrated in figure 1. (Kuo, T. C., 2012)

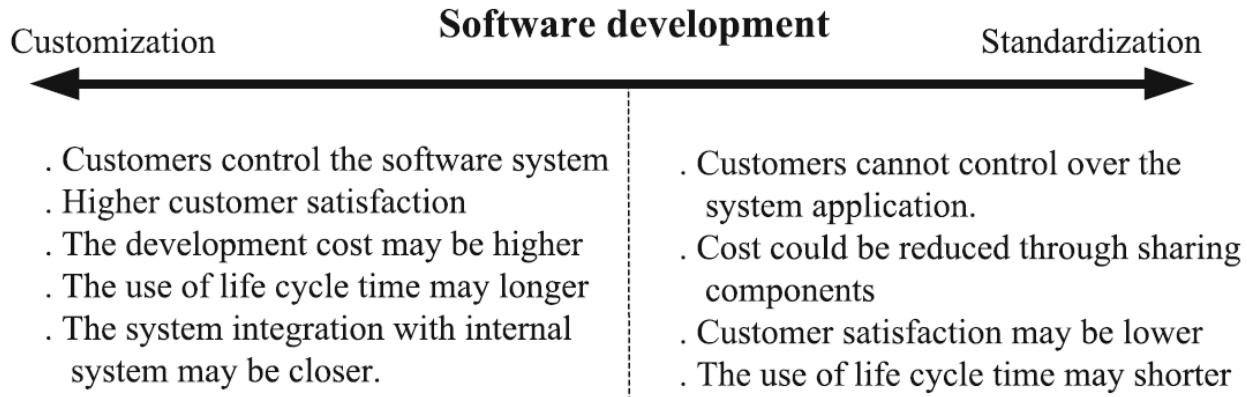


Figure 2 The software development for standardization and customization (Kuo, T. C., 2012)

Consumer markets are evolving at a faster rate than ever before, and customers' demands are more sophisticated than ever in the past (Cox, M. W., & Alm, R., 1998). As a result, in the previous ten years, mass customization has arisen as a strategy for coping with new market realities while still allowing enterprises to profit from mass-production efficiencies (Pine, B. J., 1993). In contrast to mass manufacturing, mass customization is distinguished by a significant level of information intensity (Piller, F. T., 2002). Every deal necessitates consultation and supervision on the client-specific product design, as well as clear interaction between the provider and the customer using appropriate configuration instruments, the features of the provider's solution area are converted into a specific client order. This is referred to as the 'elicitation' of a mass customization approach by Zipkin (2001). In order to discover and convert the client's desires and expectations into a clear product description, the supplier must interact with the consumer to gather the essential data. The elicitation process is usually an act of mutual interaction and co-development, rather than just a data exchange. Customer convergence is the result of bulk customization systems being reduced. Customer integration is a form of industrial value development in which "consumers engage in activities and processes that were historically considered the domain of corporations" (Wikstrom, S., 1996). The customer becomes a 'co-creator,' sometimes known as a 'prosumer' (Toffler, A., 1970). As a result, a co-creation mechanism arises, which is a system of company-consumer contact (social exchange) and adjustment targeted at adding value for both the provider and the client (Milgrom, P. & Roberts, J., 1990). The consumer is viewed as a development factor

by suppliers, as he or she performs tasks that would otherwise be conducted internally in a mass-production system (Ramirez, R., 1999). Co-creation in mass customization, on the other hand, goes beyond traditional approaches like having customers clean up after themselves in a fast-food restaurant or leaving assembly of the product to the customer, like Ikea does. The key part of customer incorporation in a mass customization system occurs during the planning or even design process of a product. The degree of integration can range from simple computer configuration from a collection of predefined options to true product co-design. (Piller, F.T., et al., 2004)

Customization has traditionally been associated with the ability to charge higher premiums due to the added advantage of a personalized solution fulfilling a consumer's individual demands, i.e. the increase in utility a customer receives from a product that meets his or her needs better than the product of the best quality available (Chamberlin, E. H., 1962). It is worth noting that "mass customization" is not the same as "customization." To differentiate mass customization from individual customization, the potential magnitude of the additional desire to pay should be stressed: mass customization would be appealed to only if the prices demanded for the personalized solution do not result in a shift in consumer segments as opposed to delivering the commodity in a mass-production method. The same vast number of consumers can be met as in consumer markets of the modern economy, and they can be handled separately as in personalized pre-industrial economies' marketplaces. (Davis, S., 1987) Traditional customization practices, on the other hand, always result in such significant incremental costs that the resulting prices imply a shift in consumer segments. If it is considered that one of the distinguishing characteristics of mass customization is the need to maintain consumer segments from previous regular (mass) products, managing the further customization expenses has become a strategic component of mass customization. (Piller, F.T., et al., 2004)

Additional costs associated with this method threaten the extra premiums in mass customization (compared to conventional mass-production). In general, higher prices can be seen in both sales and consumer interaction, as well as processing. As previously stated, higher transaction costs result from the elicitation and contact with consumers. This requires not just expenditures in configuration systems and other data-handling equipment, but also methods to lower the costs of customization from the consumer's standpoint. Obviously, a customization approach would not be a strategic benefit if consumers were disappointed or annoyed with the complexity. (Huffmann, C. & Kahn, B. E., 1998) Taking the appropriate steps necessitates extra expenses, such as improvements in customer support centers,

highly skilled personnel, or trust-building promotion programs. In addition, because of the smaller lot sizes in shipping, distribution prices are higher.

Personalization

Personalization is quite an old phenomenon and there is a variety of definitions explaining the concept depending on where the personalization is applied (Merisavo et al., 2002). Concerning Software development personalization means adjusting the product to the requirements and needs of the customer. Personalization is beneficial for the customers as it can help to provide better product or service. (Vesanen, J., 2007) However, personalization also has its costs that include longer waiting time, privacy risks, and extra fees. In case benefits exceed costs, personalization becomes worth applying, otherwise, it might be too early to adopt personalization. (Simonson, 2005) Company may benefit from personalization because it might raise the price of the product or service, impact the customer's loyalty, help to stand out among competitors and increase the overall customer's satisfaction (Vesanen, J., 2007).

Personalization, also known as customization or individualization, approaches and satisfies consumer desires by taking into account individual preferences (Tseng, M. M., et al., 2010). In comparison to customization, which relies on product differentiation for consumer segments, product differentiation focuses on individual consumers. Personalization and customization were studied by Prahalad and Ramaswamy (2000). Personalization, according to their study, refers to the co-creation of the experience, which includes real experiences, while customization refers to choosing from a variety of existing features.

The MC and personalization method for software development concentrates on how to identify and distribute the same key elements through various pieces of software. A software framework, in general, is made up of software elements and software implementations. Software components are self-contained units of architecture, processing, and deployment that communicate and integrate with other components to create a self-contained software framework. (Heineman, G. T., & Councill, W. T., 2001; Szyperski, C., 2002) A component architecture is a well-defined software device with a documented interface that can be used to build bigger systems by combining it with other components. Rodriguez et al. (2004) establish a framework for investigating reusability difficulty in element-based systems. They formulated a product composition cost model based on the premise that the components will be reused, either to construct a new system or to change an existing one. Furthermore, software systems can be customized to meet the demands of customers and users, and their modules can be removed or modified

without compromising the rest of the program (Machiraju, V., et al., 2000). Case trends were used by Issa et al. (2006) to predict reusability of information systems. Reusability was also aided by the definition of product families (Bosch, J., 2000). Deelstra et al. (2005) provided a vocabulary and definition structure for product development. The most crucial concerns are essential in identifying the fundamental characteristics (components) shared by software and deferring component variation until the end of the development process. A good software system, as seen in Fig. 2, should be adaptable to various types of software by part sharing.

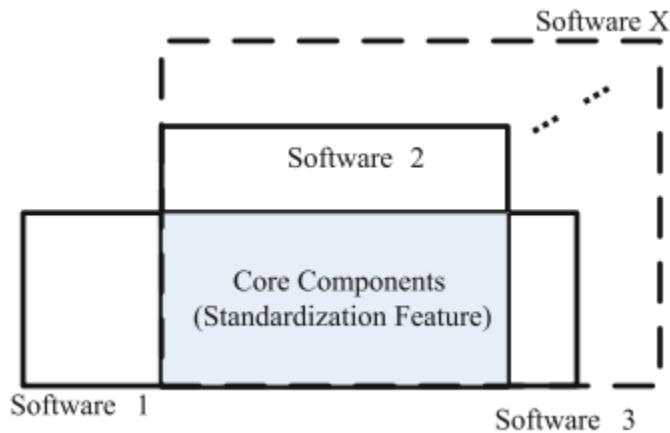


Figure 3 Different software via components sharing (Deelstra, S., et al., 2005)

Customer Satisfaction

By focusing on delivering services, companies offering goods hope to improve their value proposition and therefore their competitiveness (Kindström, D., 2010). A focus on this will have a positive impact, as increased customer loyalty can have a direct influence on a company's reputation, encourage repeat sales, and increase profits (Carroll, A.B. & Buchholtz, A. K., 2015). A potential provider will then turn into a contractor and loyal supplier, forming a relationship with the customer (Schröder, M., 2019). Focusing on a strategic path and ensuring customer loyalty leads to a competitive advantage and long-term success. Recurring orders reduce prices per user exponentially, and manufacturers may also negotiate higher prices. Furthermore, satisfied buyers are more likely to refer the vendor to others, thereby opening up previously untapped possibilities. (Sheth, J., 2001)

Consumer satisfaction is described as a measurement of a customer's overall experience. Customer Success Management, on the other hand, is a metric of net satisfaction generated and experienced by consumers as a result of their use of the seller's goods and services. Customer loyalty rises as the customer

succeeds. (Adams, R., 2020) As a result of CSM, consumers have a positive experience, which leads to increased customer loyalty (Mehta, N., et al., 2016).

Customer Success Management

While sales are always used as the primary point of contact with customers, there will be several other touchpoints with customers in operation. This trend is becoming more and more pronounced as distribution systems get more automated. As a result, using the service as its own distribution outlet is rational. Customer retention, and hence customer loyalty, as well as up- and cross-selling, are both influenced by service. Many businesses are relying on the potential of current buyers as consumer acquisition costs (CAC) continue to rise. This strategy has been used successfully for a long time, especially in the selling of complex applications. (Katzengruber, W. & Pförtner, A., 2017)

The seller's value will be enhanced by not setting retail sales as a priority but instead focusing on the customer's central, financial, and organizational issues. As a result, concentrating on increased problem solving and more consumer success is associated with reduced competitiveness and lower price elasticity. (Küng, P., et al., 2006) Many dynamic solutions are predicted to be offered in service business models in the future as a result of increased digitalization. Increasing the seller's value proposition and thereby increasing their competitive advantage may be as simple as focusing on customer success (Kilian, D. & Mirski, P., 2016).

Subscription models are small starting contracts that may be established over the customer's lifespan. After the first transaction, 75 to 90 percent of the revenue is made, depending on the product or service. Its goal is to keep buyers for a long time and give them extra purchases during that time. (Kilian, D. & Mirski, P., 2016) While attempting to minimize customer turnover, an emphasis on account management often aims to improve sales by enhancements facilitated by preparation, assistance, and thereby improved product use. The aim is to boost Customer Lifetime Value (CLTV). (York, J., 2012) Since acquiring a client is much costlier than keeping one, focusing on customer loyalty is an effective cost-cutting strategy (Kilian, D. & Mirski, P., 2016). CLTV is of concern here, unlike contract value. For this, the expense is compared to the service value over the course of the customer's estimated subscription period. (Nirpaz, G. & Pizarro, F., 2016)

Farming is the method of growing existing clients, as opposed to hunting, which is the process of attracting

new customers. In keeping with the comparison, consumers in farming are nourished and evolve over time rather than being chased down and then left on their own with the product. (Nirpaz, G. & Pizarro, F., 2016) The economic worth of a consumer is realized in subscription models by recurring purchases over time rather than a single charge. This means that IT solution vendors can no longer focus entirely on initial revenues. It is suggested that they turn their emphasis on long-term client partnerships and active customer service. Customers can prolong or even increase their subscriptions in this manner, and suppliers can generate consistent cash flow. (Kilian, D. & Mirski, P., 2016) Farming is a popular method, particularly in subscription-based market models, where consumers must renew their contracts every subscription time. Customers must be happy with the customer service, and customer satisfaction management must be in place, as well as the applied approach must be effective. As a result, this term holds a lot of promise for SaaS. CSM is a toolkit for the farming approach, in which the customer's success must be at the center of all activities. (Nirpaz, G. & Pizarro, F., 2016) A Customer Success Manager's profile is depicted in Table 1.

Table 1 : Profile of a Customer Success Manager (Kilian, D. & Mirski, P., 2016)

Characteristics	Responsibilities
<ul style="list-style-type: none"> • Focuses on establishing long-term client partnerships. • Problem-solving capacity that is proactive • Considering the big picture 	<ul style="list-style-type: none"> • Recognizing and meeting the demands and desires of customers • Promote the customer's implementation of the IT approach and assist with change management. • Customer satisfaction and solution usage to be measured. • Internal coordination and cross-departmental collaboration • Organize the value roadmap to take advantage of up- and cross-selling possibilities.

Adams (2020) created a robust CSM system with the goal of making consumers successful by using the seller's product. This involves assisting the consumer in overcoming challenges in order to maximize the probability that the customer would continue to make more purchases and even increase the rate of

purchases. There are seven steps of this model explained in the table below.

Table 2 Customer Success Management Framework (Adams, R., 2020)

Preparation	This includes collecting more customer knowledge and developing a customer experience policy plan.
Commitment	Organizing and holding the first consumer stakeholder meeting, at which the level of assistance provided by the CSM manager is agreed upon.
Onboarding	Understanding consumer onboarding demands and expectations is one of them. It is developed, implemented, and updated an onboarding milestone schedule that includes tasks and responsibilities.
Adoption Planning	Becoming aware of the demands and uncertainties that can occur during the adoption process. This involves segmenting the types of people who will be impacted by the transition. Furthermore, internal partners must approve an adoption scheme.
Adoption Implementation	Managing the transition that comes with the implementation.
Value Realization	It also entails making sure that the consumer recognizes the value generated and that marketing opportunities are identified and managed.
Engagement Evaluation	Reflecting on the process's insights and lessons learnt in order to enhance prospective CSM projects

CSM is defined in a condensed manner, with four phases for customers listed. The behavior of related sellers is linked and briefly listed in Table 3.

Table 3 Customer Success Cycle (Nirpaz, G. & Pizarro, F., 2016)

Phases for the customer	Actions of a seller
New: The platform must be configured for newly signed customers' users.	Onboarding: ensuring that consumers get an immediate benefit from the solution
Growing: the customer's usual situation. The approach is used by the users.	Nurturing: increase the customer's satisfaction for example, by coaching, the presentation of various business scenarios, and software features.
Renewal/Upsell: customers near or at capacity limit, as well as customers getting towards the end of their subscription period	Harvesting: The customer relationship must be maintained and improved. Need to assist clients in extending (renewing) and expanding (upselling) their contracts or purchasing new solutions (cross-sell)
Cancelled: when consumers are planning to depart (churn)	Saving: identifying and responding to warning signals that a customer is about to leave.

CSM's main aim is to reduce the customer's deployment time so that they can be as efficient as possible and get the most benefit out of the solution. This is particularly problematic in dynamic solutions, since the onboarding process can be lengthy. (Mehta, N., et al., 2016) Fast onboarding and hitting value seems fair, particularly in initial short-term ventures. Although onboarding and nurturing are likely to be the primary target of a pilot, upselling within harvesting seems to be the overarching objective of pilots. The goal is to expand in the enterprise until one or two divisions have accepted the approach and measurable benefit has been shown. (Kilian, D. & Mirski, P., 2016) As a result, the aim is to consistently expand and nourish the consumer. CSM is not restricted to the nature of a project, but rather a long-term relationship with the client. (Nirpaz, G. & Pizarro, F., 2016) As a result, a pilot can be expected to set the tone for what the consumer can expect from a long-term partnership with the SaaS vendor. The following is a summary of CM, which is an integral aspect of CSM. (Adams, R., 2020)

SaaS

More and more IT firms are adopting a subscription-based Software as a Service (SaaS) model. Software as a Service (SaaS) is gaining traction, thanks to a huge increase in the number of suppliers entering the market and the recent success of a number of major companies. (Knorr, E., 2006) SaaS is about providing

technical functionalities to a large number of customers across the Web with one single instance of software application operating on top of a multi-tenancy platform. It was created to take advantage of the advantages of economies of scale. (Guo, C.J., et al., 2007)

Clients are usually not required to buy or update the software kit in their personal computing environment. They use the credentials given by the SaaS provider to log in and access the SaaS service through the Web from whatever Internet browser at whatever time and from any location with Internet connection. In terms of business data management, process control and optimization, and governance, technology is meant to assist a company's activities. Because of the Web-based distribution strategy, SaaS offers considerable difference in terms of cheaper Total Cost of Ownership (TCO) and more adaptability for users. Each customer, on the other hand, is unique, resulting in a wide range of software demands. The major sources of need variance across customers include industry variations, consumer behavior differences, product offering differences, regulation differences, culture differences, and service approach differences. As a result, in order to effectively assist the customer, most software products and services must be customized to some level. (Kingstone, S., 2004)

Comprehensive configuration and optimization features have previously been offered by top SaaS suppliers. For example, Salesfoce.com provides Apex, which uses a multi-tenancy structure to allow for sophisticated device customization and configuration over the Web. Because the SaaS distribution model prevents the SaaS provider from creating and storing a variant of application code for each individual user, the SaaS service's customization and configuration capabilities are essential to its success. (Rohleder, C., et al., 2005)

The basic concept behind SaaS is to support a great number of customers with a single software program. SaaS providers should not create and maintain separate program versions for each customer. The perfect scenario for SaaS vendors is that each customer is comfortable with the standardized offering. However, in the world of commercial software applications, this ideal scenario seldom occurs. (Sun, W., et al., 2008)

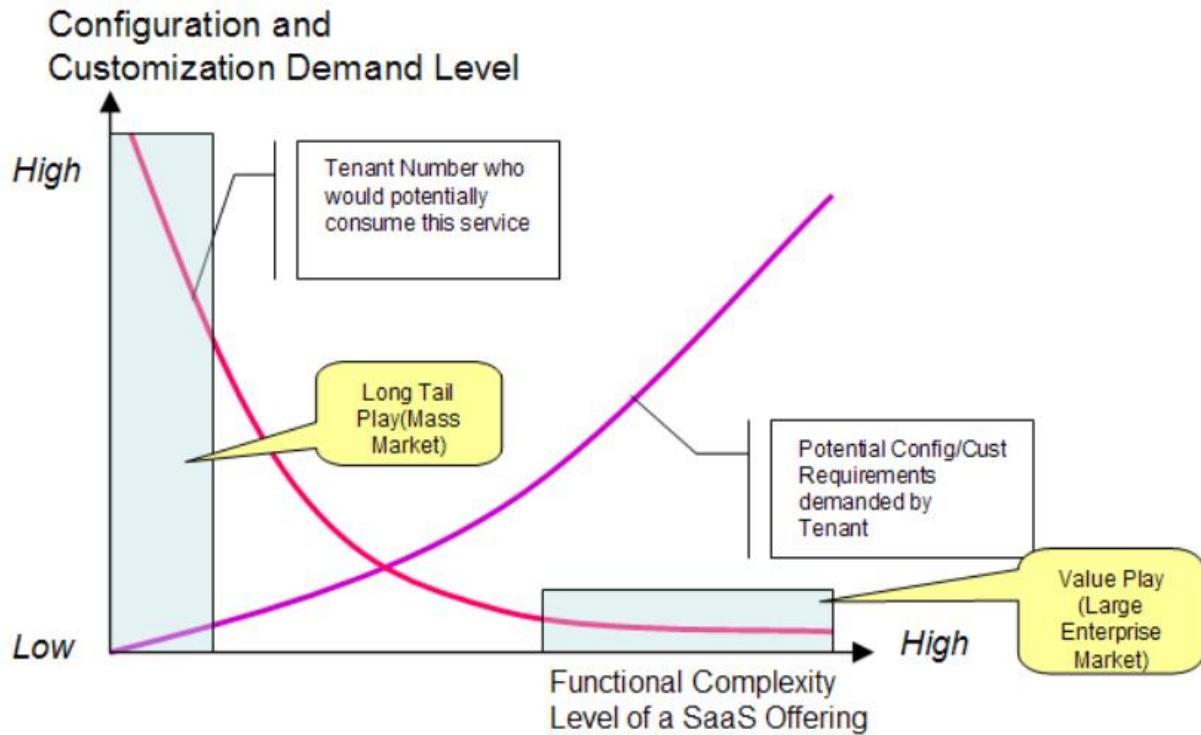


Figure 4 Customization and configuration Demands vs. Functional Complexity of a SaaS (Sun, W., et al., 2008)

As seen in Figure 3, the more the software's functional complexity, the more future tailoring actions are necessary to accommodate a specific client. Clients generally just need to configure Web E-Mail using parameter-based options, such as e-mail box storage capacity and account number, because it is a SaaS software with very simple capabilities. CRM is a service with a medium degree of feature sophistication in the market, which is why many CRM SaaS suppliers offer considerably greater customizing capabilities via customization and configuration tools. Since SaaS capitalizes on the economies of scale of a large number of clients with a long tail strategy, the more sophisticated the program, the less suitable it is to investigate the SaaS model, as clients can request very complicated tailoring specifications that cannot be done efficiently with a Web-based distribution model in a multi-tenancy context. (Sun, W., et al., 2008)

To guarantee that their clients continue to use their software for a long time is vital for these businesses (Skok, 2010), and one of the key aspects is to help them to run the software flawlessly and quickly. In case company is unable to provide a seamless service, it can lead to the loss of a customer that can switch to a more comfortable option. (Frisk, E., 2014)

The process to help clients get up and running is called customer on-boarding. The first 90 days from the

moment the customer has made the purchase is considered to be the most critical time of the entire customer lifespan for enterprise that uses subscription models (Costanzo, C., 2006). This period of time is crucial because the customer will be decided whether to keep using the service or leave. What is more there is quite a limited research made regarding the topic of how to retain the customers, especially in terms of SaaS business. This knowledge is in turns very important because it may help the companies to retain majority of their customers and be more profitable. (Frisk, E., 2014)

Customization and configuration are two main approaches to tailoring a SaaS service. People are often perplexed by the distinctions in these two words. In fact, different SaaS vendors use various words in various contexts. As seen in Figure 2, in order to provide a generic SaaS offering for a single client, we must tailor it into a tenantized offering by meeting this client's specific needs. (Sun, W., et al., 2008)

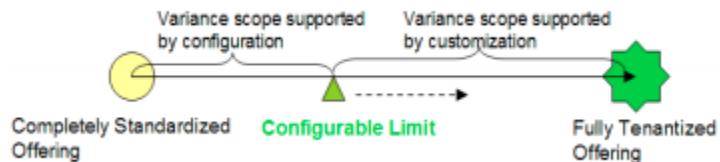


Figure 5 Customization and configuration (Sun, W., et al., 2008)

Both Configuration and Customization can help with this level of customization. The distinction is based on complexity. The SaaS application's source code is not changed during configuration. It normally allows for variation by setting pre-defined criteria or using software to alter program features within a pre-defined frame, such as adding data fields, changing field names, updating drop-down lists, adding buttons, and changing business rules, among other things. Configuration allows for customizing specifications inside a configurable range. Customization entails modifying the source code of a SaaS program to include functionality that goes beyond the configurable maximum. (Sun, W., et al., 2008)

Customization is a way more expensive solution for both SaaS vendors and customers when compared to Configuration. Since customizing SaaS program necessitates modifications to the source code, there are a slew of issues that come with a high cost, for instance: requiring staff with higher qualifications and a higher salary to work on customization; providing resources and infrastructure to deal with different revisions of software code; requiring a much longer lifecycle due to code development/debugging/testing and deployment; and missing market opportunities from customers who cannot tolerate the

customization difficulty and cost[9]. In the case of SaaS, customization is getting even more complicated, as SaaS vendors must manage each piece of customization code tenant by tenant. Any tenant's customization code should not be lost when the SaaS program is upgraded. As a result, SaaS can eliminate customization wherever possible by relying on configuration to satisfy clients' tailoring needs and extending the configurable cap as far as possible to meet clients' specific needs. (Rohledder, C., et al., 2005)

2.2 Role of company departments in gaining and retaining customers

Different departments of the company are contributing to gaining and retaining the customers. Thus, it is important to understand what kind of role startup management, sales and account management, customer relationship management, and development team are playing in the process. As for startup management, it is required to understand what kind of managerial challenges may emerge and how to deal with them. Sales and account management is closely tied to customer on-boarding, as the actual on-boarding process starts once the service or product is sold to the customer. Customer relationship management is related to how existing customers are managed. Software development team is also important because the service or product is undergoing the continuous development and rapid integrations are needed. Software testing and quality assurance is also related to the gaining and retaining of the customers. (Frisk, E., 2014)

Startup management

There are Customer Development and the Lean Startup Movement frameworks that highlight the importance to systemize the learning processes to help the organization to understand at the early stage whether or not it is following the direction or if some changes to the product or service are required (Maurya, A., 2010). Blank (2007) lists some of the most important points that can help a startup company to succeed:

- Validate with the client that the company is making the right product or service. Some companies are building the product or service without sufficient validation from customers. This can result into the outcome that is not accepted by the market.
- Use Minimum Viable Products (MVP) to test hypotheses: Instead of building a product that would be packed with various features and trying to sell it as a linear process, it is advised to build a very simple product that still has some value for the customer, also called MVP, so it can be tested if the value it possesses is enough or not. Later, based on the feedback received from the customers new MVP can be

built taking into account the obtained learning.

- Find out what the company's current growth phase is. There are four growth phases that a startup company undergoes:

(1) Customer Discovery. The phase to experiment with customers and minimum viable products to find a product that will be of value for some customer segment.

(2) Customer Validation. The phase when an early version of your product is launched to the market.

(3) Customer Creation. The phase when the product is launched to the market at scale.

(4) Company Building. The phase of rapid growth. (Frisk, E., 2014)

Some practitioner literature, in contrast, is more closely linked to the management of SaaS startups. One authoritative source's blog post (Skok, 2010) discusses the most critical KPIs and market dynamics for creating a successful SaaS business. According to two recommendations for creating a successful SaaS firm, the Customer Lifetime Value (LTV) should be higher than three times the Customer Acquisition Cost (CAC), and it should take no more than 12 months to recoup the CAC for an average customer. The following formula is used to calculate these values:

- LTV - Average monthly revenue per customer x Average lifetime of a customer -

Cost to serve a customer

- CAC - Total cost of Sales & Marketing / Number of deals closed

Customer profitability is equal to LTV - CAC, which underscores the importance of controlling these values.

Concerning the customer onboarding, it appears that there is a lot to consider when creating a customer onboarding process. It must not only contribute to systematizing customer learning, but it must also strive to maximize a customer's LTV by maximizing the customer's total lifetime and controlling the costs of serving it. It can be also discovered that the objectives of the on-boarding process may alter as the company's current growth stage change. (Blank, S. G., 2007)

The research on startup management emphasizes the importance of a startup's ability to learn from and respond to customer behavior as quickly as possible (Blank, S. G., 2007). Client onboarding should be considered to be an essential part of the startup's overall learning process because it is the first time the product that is going to be sold is tested for a specific consumer. A customer onboarding process that is structured to gather knowledge and also deliberately execute tests and reported to the management and

product development teams is likely to help the startup. At the same time, the customer onboarding team in a B2B SaaS company should seek to minimize customer acquisition costs and improve customer loyalty value, according to publications on metrics in SaaS firms. These metrics are important to the profitability of a SaaS organization. (Skok, D., 2010)

Besides this, the background would most likely shift based on the startup's actual stage of growth. The business will turn from experimentation to development as it progresses through the four phases (Customer Discovery, Customer Validation, Customer Development, and Company Building). This would certainly have a similar impact on the client onboarding team's objectives. (Blank, S. G., 2007)

Sales management

Sales management is sometimes considered not having direct relation to the customer on-boarding. Mainly the focus lays within the activities that lead to the closure of the deal and not within the process that follows it. Quite popular methodology called SPIN Selling (Rackham, N., 1988) highlights the importance of how the sales representative should ask the questions to convince the prospect to sign. These questions can be classified in the following way:

- Situational questions dealing with the prospect's situation at the moment
- Problem questions dealing with the current prospect's difficulty
- Implication questions that are aiming to define the effect of the above-mentioned problems
- Need-payoff questions that elaborate regarding the possible solutions to the problems of the prospect.

Research revealed that those sales persons who used to ask too many situation questions were less often closing the deals in comparison to those salespeople who tended to ask questions about the problem of the customer and held the solution until late in the conversation. (Frisk, E., 2014)

Customer relationship management

Customer value, customer retention, customer recruiting, and overall profitability can all benefit from a successful Customer Relationship Management (CRM) program (Ling and Yen, 2001). The modern literature on the subject covers a wide range of topics. For instance, there is a consensus that CRM should involve more than just setting up IT systems to handle customer data. Effective CRM is to help accumulate various information concerning the customers, sales, how effective the marketing is, as well as responsiveness and market trends. A company may reach this by taking into account four key factors:

- Interaction: The events that occur during any interaction between an organization and its customers.
- Communication: Defining each point of contact between the business and its clients.
- Knowledge: Gathering and analyzing data on consumers in order to learn more about them.
- Relating: Developing meaningful connections with customers that contribute to long-term relationships. (Osarenkhoe, A. and Bennani, A.E., 2007)

Apart from the above-mentioned points, it is necessary to implement CRM plan iteratively. Training and empowerment of workers, as well as the implementation of automated processes that assist customers in different ways, should all be included in the CRM strategy. (Ling and Yen, 2001)

Enterprise software integration

It is well known that integrating modern business software into an organization is a challenging task. Because of its scope, the phenomenon has got a lot of attention in academic literature. Best-practices in handling these broad, complex software integration projects are one topic that the literature continues to concentrate on. Lam and Shankararaman (2004), for instance, propose dividing the project into five phases: (1) comprehend the end-to-end business process, (2) map the process onto components, (3) list the specifications, (4) generate the architecture, and (5) schedule the integration. In another article, the importance of using simple Six Sigma tools in such projects is emphasized. More precisely, it recommends that organizations use the DMAIC approach (define, quantify, analyze, improve, and control) for organized problem solving in order to determine which areas of the business can be enhanced using enterprise software. While this field of the literature tends to be well-developed, there does not appear to be a common opinion in the field as to what the best-practice solution is. The significance of top-management support and consideration of the market- and/or people-aspects of the integration, as well as the technological side, is emphasized, but these factors may be considered general best-practices when introducing any new business proposal. (Chau, Liu and Ip, 2009)

Another topic discussed in the literature is why business software integration projects are often unsuccessful. In general, 75% of ERP software implementations are deemed unsuccessful (Rettig, 2007). According to Rettig (2007), this is mostly attributable to managers' perceptions of ERP software's infinite potential in terms of business benefits. They see it as a panacea and massively underestimate the expense and difficulty of introducing ERP program. This complexity also makes upgrading old technologies challenging and expensive, forcing businesses to retain a number of legacy systems that make it difficult

to implement new technology that does not work well with the old.

According to the literature on Enterprise Software Integration, large-scale software integrations are frequently undervalued in terms of difficulty and expense. Simply put, they have a tendency to become shambles. When you combine this with the complexity and high speed with which startup firms work, it seems that preventing customer onboarding mishaps is unlikely. (Frisk, E., 2014)

Software development

Software development can be concurrent or sequential. A high degree of overlap between various operations in the software development process characterizes concurrent software development. In comparison to more conventional planning methods, where various tasks are carried out in a sequential order with little or no overlap, this approach is more innovative. Concurrent software development is more difficult to handle than sequential software development, but the advantages outweigh the difficulties. Simultaneous engineering projects are more competitive in terms of time to market, project development costs, and product quality. (Bhuiyan, N., et al., 2006; Aoyama, T., 1997) They also enable gradual delivery, which ensures that customer input can be gathered earlier in the development phase (Aoyama, T., 1997).

These advantages are primarily due to an improvement in efficiency during the production process's early stages. Risks and tradeoffs can be detected with greater precision, requirements can be more accurate, and future manufacturing problems can be addressed earlier by involving all related functions for the development phase from the start. (Bhuiyan, N., et al., 2006) According to Blackburn, Scudder, and Wassenhove (2000), spending extra time and effort in the early phases of a software project resulted in shorter total cycle times and increased efficiency since less rework was required later. Redesign, recoding, and retesting due to changes in specifications are the leading causes of time delays in software development projects. A temporary lack of resources can lead to a prolonged decline in project performance if the early stages of the project, such as concept design, are downplayed in favor of product design and testing. Poor concept design results into product design rework, which leaves less money for concept design in the next iteration or project, leading to a performance downward spiral. As a result, it becomes critical from a management standpoint to avoid compromising efficiency and capability-building practices during periods of high pressure. (Rahmandad & Weiss, 2009) If this advice is followed, it can save time needed to prepare for the future rather than responding to urgent issues. Concurrent software

development can be difficult to handle, but it can help to achieve shorter cycle times, higher overall efficiency, and higher project quality. (Frisk, E., 2014)

Software testing and quality assurance

Despite the ever-increasing importance of computing in contemporary society, the area of software testing as an engineering activity is still in its infancy. In reality, software testing is more of an art than a science these days, with practitioners mostly using the same techniques as 20-30 years ago. In general, these techniques are neither complex nor effective. (Juristo, Moreno & Strigel, 2006) The sophistication of computing, along with the poor knowledge of its rules, makes software testing very difficult (Choudhary & Kumar, 2011). Testing cannot ensure that a machine is defect-free; it can only expose particular flaws that arise as a result of a specific collection of inputs (Nirpal & Kale, 2011). Nonetheless, it is a critical method for improving software efficiency and reliability (Juristo, Moreno & Strigel, 2006). Good testers may use their imagination, expertise, and instincts to detect flaws in software before it is released (Choudhary & Kumar, 2011).

In respect of lowering the number of issues, software testing and quality assurance research reveals that testing software is essential for reducing the number of defects in the code. It does, however, caution that software testing is not without flaws. While software testing can be improved by modifying the method, it is ultimately dependent on intuition and provides no guarantees. This indicates a need to strike a balance between the services devoted to software testing and those devoted to resolving consumer business issues. It also implies that guaranteeing problem-free software to consumers is impractical, or at least prohibitively expensive, and that the customer onboarding team needs a systematic and careful mechanism for dealing with customer issues. This is particularly true if the program is constantly modified and altered, which is always the case with a startup. (Frisk, E., 2014)

Customers

Retention of customers is very important and customers themselves play a certain role in this process. Loyal customers may bring new clients to the business. Customer plays a key role in terms of increasing sales especially when a complex product is addressed. People tend to refer to the ones they know when deciding upon purchasing new complex product or service. Cost to lose a customer is also high because long-term customer relationships should be taken into account. (DeSouza, G., 1992)

Customer retention should be measured. In case it is not measured, there will be no room for improvement. Business may lose customers due to a number of reasons. Some of them cannot be impacted by the company itself, others can be taken care of early enough that would allow to keep the customer. Reasons why the customers are lost are the following:

- Customers may choose a competitor due to a more attractive price of a product or service. This type of customer can be gained back by offering a better price.
- Competitor's product may be superior to the one offered by the company in question. This kind of customer may be impossible to retain back as
- Customers may select a competitor due to the poor service
- Some customers may choose product or service offered by the company outside of the industry
- Political reasons may also impact the customers choices. (DeSouza, G., 1992)

The actual cost of losing a customer is the amount of money the consumer may have invested into the company over the course of their lifetime. The challenges presently confronting Detroit exemplify the bitter and lasting experiences generated by a poor customer experience. (DeSouza, G., 1992)

3 THEORETICAL FRAMEWORK

3.1 Profitability assessment

One of the most essential statistics to monitor in a SaaS firm is Monthly Recurring Revenue (MRR) which is the revenue that company is going to generate monthly (Saasoptics, n.d.). It is certainly a major factor in profitability. Startups are important for any economy that wants to invest in innovation and development. As a result, the early stages of a startup are widely studied (Davila, Foster, He & Shimizu, 2015). Schumpeter's creative destruction process is related to the death and survival of startups when entrepreneurs with fresh ideas join and replace companies that are old and sluggish (Huynh, Petrunia & Voia, 2012). According to Huynh et al. (2012), one possible impediment to this method is that startups lack the financial capital to carry out their plans. Startups usually have a difficult time raising equity capital, so they have to rely on internal financing and borrowing to set up a business. Internal finance, according to Zingales (1998), is the primary source of financial resources for these companies. There should be enough of Internal finance as it is directly linked to a startup's early-stage development and profitability. As a result, it is important for founders, financiers, and entrepreneurs to understand the early stages of a startup's growth and profitability.

Profitability is commonly calculated using recorded financial ratios, which have a variety of flaws (Murphy, Trailer & Hill, 1996; Losbichler, Hofer, Eisl & Zauner, 2012). As a result, recognizing the early stages of a startup's viability is linked to the features of financial reporting. This is essential since one of the purposes of financial reporting is to make management accountable to investors in order to allocate capital efficiently. Financing should be allocated to the most productive companies, and their valuations should be higher than those of less efficient businesses. (Ak, Dechow, Sun & Wang, 2013) Financial statement analysis is used to assess firm performance, the general approach is applied to quantify ratios that reflect key underlying factors such as profitability. The consumer will then examine the ratios' time-series and cross-sectional patterns. The consumer must have a clear understanding of the content and signals of the ratios in order to get a reliable point of view. Financial ratios demonstrate stable profitability if a company has a steady state, thus the investors may usually agree on its valuation. However, startups tend to face a nonstationary adjustment during the early stages of development, which complicates the forecasting of the future profitability. (Ak et al., 2013)

According to the statistics, in the United States and Europe, only about 45-55 percent of startups survive their first five years (U.S. Bureau of Labor Statistics, 2016; Eurostat, 2016). Thus, in the early stages of a startup, it is critical to monitor the signs of potential failure provided by financial ratios. Financial ratio-based failure prediction methods can provide accurate signals for startups as well. (Laitinen, 1992) The model's reliability is also jeopardized by the presence of extreme ratio values and nonstationary processes of failure (Balcaen & Ooghe, 2006; Moses & Liao, 1987). These models are also hindered by the poor financial data quality that is common among small businesses, especially startups. When it comes to applying failure prediction models to startups, it's critical to understand the nonstationary nature of startup growth and the accuracy of financial data. (Balcaen & Ooghe, 2006)

These models are also hampered by the poor quality of financial data that is common for small businesses, especially startups. Understanding the nonstationary essence of startup growth and the consistency of financial data is therefore important when applying failure prediction models to startups. (Balcaen & Ooghe, 2006) There is a clear empirical observation of the profitability ratios of the startups, Table 1 demonstrates the example of the profitability ratios time series for a number of randomly chosen startups from Finland (the ORBIS database of Bureau Van Dijk). The three first cases in the table depict time series of startups that survived, while the two last columns depict time series of startups that did not survive. These time series are consistent with Garnsey et al. (2006) explanations of early startup growth. Profitability ratios tend to be unchangeable, but they are usually nonstationary, increasing or decreasing rapidly according to the phase of change. It's difficult, if not impossible, for a financial analyst to decipher the meaning of signals conveyed regarding a startup's future, or even what type of profitability these time series represent. These time series are the product of financial statements and the startup growth phase. As a result, only a mathematical model based on growth and accounting principles can clarify the endogenous properties of these time series.

Table 1. Examples of startup profitability ratios of startups during early years. (Laitinen, E., 2017)

Panel 1. Cash flow to net sales ratio (%)

Status of the startup after ten years:

Period	Active	Active	Active	Bankruptcy	Bankruptcy
1	-3,13	5,00	-51,00	4,13	23,17
2	1,30	4,79	-11,58	3,45	16,53
3	3,20	4,40	12,72	0,87	3,25
4	5,14	6,59	13,19	-3,40	-17,00
5	5,78	4,37	16,04	5,32	24,91

Panel 2. Profit margin to net sales ratio (%) (Laitinen, E., 2017)

Status of the startup after ten years:

Period	Active	Active	Active	Bankruptcy	Bankruptcy
1	2,00	-52,77	-22,70	21,44	4,75
2	1,75	-12,62	-3,71	18,21	3,72
3	2,82	12,51	-3,39	3,11	0,36
4	2,09	17,23	3,33	3,35	-4,01
5	0,12	21,27	3,12	3,82	4,89

Panel 3. Return on investment ratio (%) (Laitinen, E., 2017)

Status of the startup after ten years:

Period	Active	Active	Active	Bankruptcy	Bankruptcy
1	25,68	-91,61	-10,20	-32,26	76,51
2	24,23	-91,50	5,40	-85,96	35,68
3	24,78	82,10	13,03	33,59	9,23
4	27,10	65,48	9,61	-8,43	8,12
5	23,69	67,23	6,65	-0,72	7,13

The development of a startup is a quite a complicated process, but for the purpose of research, it will be simplified and assumed that the costs of the startup are growing at a continuous rate and generate equal revenue flows in each period (Laitinen, E., 2017).

The internal rate of return (IRR) and the lag structure of income flows created by periodic spending are assumed to be constant. In addition, each period, the startup is expected to spend a certain amount of periodic expenditure and start of the period assets. Due to continued predictions, ratios of profitability might be nonstationary in the early years and require a substantial amount of change to achieve a stable state. The IRR, growth rate, cost rate, and revenue lag structure all have a major influence on the adjustment process. As a result, it is argued that profitability ratios are unable to correctly depict profitability (IRR) in the early years of a company and might offer misleading signals about the startup's performance. Using a large sample of Finnish startups to evaluate numerical adjustment procedures for steady state estimations offers empirical support for the interpretation of mathematical conclusions. The

parameters of the stable model are computed using financial records from 2608 startups during a nine-year period. (Laitinen, E., 2017)

3.2 Methods utilized in the study

3.2.1 Delphi method

Delphi method can be illustrated by the following example. If the profitability of software has to be evaluated, experts from such fields like software development, marketing, management, etc. can participate in the inquiry. These experts express their opinions regarding the feasibility of the business model and its profitability. All the opinions are collected anonymously. Questionnaires are used to gather the answers. (Lilja, K.K., et al., 2011)

Delphi method should be considered by the organizations that are making significant decisions that may impact the future of the organization. To anticipate management training in healthcare sector, Loo (1996; 1997) utilized a Delphi analysis with a group of human resource experts, as well as a Delphi study combined with a case study. The Delphi findings had direct implications for the creation or modification of corporate strategies, services, and practices in the above-mentioned studies to eliminate or, at the very least, reduce the distressing and ultimately lethal effects of job stress, whereas the Delphi findings had direct implications for the provision of prompt and successful training and development in the other analysis. (Loo, R., 2002)

Delphi method is widely-used in the sphere of technology, education, and others. Kaplan, who led a research initiative at the RAND Corporation after Second world war to strengthen the usage of expert knowledge in policy-making, invented the word "Delphi." According to Creek legend, the oracle at Delphi was asked to predict the future so that accurate and prompt choices could be made before embarking on a significant course of action like war. As a result, Kaplan had the idea that subject-matter experts might be approached for their forecasts or views on the probability of possible events or situations that the RAND Corporation was involved in. The RAND Corporation continued to develop the Delphi method into the 1950s and early 1960s, until material about the Delphi method was first published. (Woudenberg, F., 1991)

The Delphi method, in principle, constructs and encourages group collaboration around a complex problem in a way that, after a number of iterations, a group consensus on a possible path can be found. A consensus-building Delphi, for example, may be used to predict potential fiscal, societal, workforce, and corporate trends in order to assist organizations in implementing human resource programs for the next five years or longer. (Loo, R., 1996) The policy Delphi, which differs from the consensus-building Delphi just discussed, is a significant variant of the Delphi approach in which competing viewpoints are debated on a complex policy problem, and consensus is not always the desired target. Given the complexities of the problem and the failure to meet the competing demands of various stakeholders, the most important outcome of a policy Delphi is the formulation of a successful policy. (Loo, R., 2002)

The main characteristics of the Delphi method are the following:

- The sample is made up of a "panel" of deliberately chosen experts who represent a diverse variety of views on the subject or problem under discussion.
- In most cases, participants are kept secret.
- Over the duration of the Delphi, the "moderator" (i.e. researcher) creates a set of formal questionnaires and feedback notes for the panel.
- It is a multi-step process that usually involves three or four rounds of questionnaires and collection of feedback.
- A study report with the Delphi findings, estimates, policy and program solutions with strengths and disadvantages, advice to senior management, and likely action plans for designing and executing the policies and programs is the typical output. (Loo, R., 2002)

Comparing to other group decision-making methods, Delphi has a number of advantages. First, idea creation in the Delphi is individual-based, confidential, and independent; therefore, unlike Nominal Group Technique (NGT) and Interacting group method (IGM), panel members are not influenced by community demands or outspoken members. (Needham, R. D. & de Loe, R. C., 1990) Second advantage is that since panel participants do not communicate, interpersonal tensions and communication issues do not emerge. Third, unlike NCT and ICW, a Delphi does not require participants to get together, therefore, travel costs and the difficulty of getting everybody to the same place at the same time is absent. Finally, using successive rounds in a Delphi allows the moderator to make conclusions based on previous findings while keeping the analysis on track. (Clayton, W. J., 1997)

As in any similar applied research, special attention should be paid to the planning and execution of the research. Delphi method consists of four activities:

- Problem formulation

Problem formulation is a vital first step in any applied research project to ensure that the type and complexity of the problem or topic to be studied, the study's intended results, and the Delphi method's appropriateness to solve the particular problem are all discussed. (Loo, R., 2002)

- Selection of the plan

The Delphi process, as a community approach to forecasting and decision-making, involves a team of subject-matter experts (SWEs). The requirements for deciding who fits as a SWE can include criteria such as personal interactions or being stakeholders, in addition to the experts. Since there may be few SWEs in certain study areas, panel selection may not be random; therefore, all established SWEs may be selected. If there is a sufficiently large established pool of SWEs, we may use a random sampling technique to eliminate biases that challenge the validity of the results. While the easiest solution is to use a straightforward random collection technique, there might be valid reasons to use stratified random sampling to ensure the representativeness of preferred stakeholder classes or dissenting viewpoints. (Loo, R., 2002)

- Panel size determination

Although there is no set sample size for Delphi research, general guidelines suggest using 15-30 well-chosen SWEs for a varied community and as few as five to ten for a homogenous group. When establishing the panel size, take into account the severity of the problem being studied, the variety of abilities required to address the challenge, and the study's objectives. A reasonably diverse panel is preferred to ensure that the full range of perspectives is represented, especially in a policy Delphi or where sub-panels will be formed, and that the analysis can produce a rich data base for in-depth evaluations and defining alternatives to aid successful decision making. (Martino, J.P., 1972)

Given considerations of calculation error, respondent bias, and the need for predictive strength, those with a clear quantitative experience can find these sample sizes limited in contrast to the hundreds needed for surveys. The Delphi approach, on the other hand, relies heavily on the careful collection of SWEs, which helps a researcher to use a small panel with confidence. This does not mean that big panels are not used. Broad panels of over 100 people have been included in Delphi experiments too. (Chaney, H.S., 1987)

The questionnaire for Round 1 can be developed in the similar manner as any other survey. Based on a consistent identification of the study's goals and a critical literature assessment, among other preparatory research tasks, the researcher would develop question items and response scales (e.g., Likert or rank-order scales) that address the main questions and issue areas to be examined. Open-ended questions can also be utilized to allow panelists to express themselves in their own words, collecting facts and details that aren't specifically asked for in the questionnaire. It is critical to pay great attention to the pre-testing and refinement of the questionnaire. After all, there are only a few SWEs available, and their capacity to participate in several rounds is required for the Delphi analysis to get off to a strong start in terms of methodology. (Loo, R., 2002)

The moderator will set up Round 1 according to some plan, anticipating that the next two or three rounds will be held to reach consensus or other goals. Round 1 might, for example, be used to resolve large problems with the goal of focusing on individual core issues that arise from earlier rounds in subsequent rounds – a funneling effect. This progresses from the general to the more specific in a systemic manner. (Loo, R., 2002)

Because the Delphi method was designed to gather quantitative data for statistical analysis, qualitative techniques (such as open-ended questions and free replies) are recommended to augment the quantitative data. The increasing availability of efficient yet affordable instruments for qualitative research enables using both quantitative and qualitative data, i.e., a triangulation of analysis approaches, both possible and appealing from a philosophical and analytical standpoint. (Denzin, N.K. & Lincoln, Y.S., 1999).

Some moderators might start Round 1 with a scenario, finding it more beneficial. A scenario that describes the current state or condition as well as different potential future states can be useful in orienting panel participants to the analysis and ensuring that everybody starts from the same place. (Bijl, R., 1992)

In a Delphi analysis, the cover letter is particularly relevant because participants must be notified and encouraged to participate in all rounds and return their completed questionnaires on schedule so that evaluations can be performed and the input report and next questionnaire can be constructed and circulated for subsequent rounds. Finally, although Delphi analyses have traditionally been performed via mail, the access to e-mail and the Internet opens new contact opportunities, potentially reducing the time

it takes to complete all rounds from months to weeks. (Loo, R., 2002)

- Delphi rounds conclusion

In order to create a feedback report for the panel and to assist the moderator in organizing resources for the next round, the returned questionnaires are subjected to quantitative (e.g. dissemination estimates, scores, and ranks) and qualitative (e.g. extraction of themes) assessments. The questionnaire items grow increasingly detailed or exact in subsequent rounds, focusing on areas where consensus has yet to be reached. (Loo, R., 2002)

While three to four rounds are typical, the moderator has the authority to call a halt to the rounds if the consensus requirements are satisfied, the outcomes become repetitive, or an impasse is reached. It should be kept in mind that if too many rounds are required or the analysis is extended for too long, panel members may lose interest. (Loo, R., 2002)

Following the conclusion of the final round, the moderator produces a brief summary and delivers it to all participants, either in its entirety or in condensed form. If time allows, the moderator could suggest assembling the panel for a post-Delphi debriefing. This might be utilized for a variety of purposes, such as assessing the research, recognizing everyone's interest in a successful study, or holding a panel discussion that builds on the Delphi's work. (Loo, R., 2002)

3.2.2. Quality function deployment (QFD)

System requirement analysis, architecture, coding, testing, deployment, maintenance, and enhancement are all typical, interconnected phases in the development of a software system. Understanding consumer requirements takes time because it necessitates doing system reviews, creating software designs, and finalizing programs. (Hoch, D. J., et al. 2000; Sommerville, I., 2007)

Quality function deployment (QFD) is a practice that originated in Japan as a method for ensuring that new goods are of high quality. QFD was first used in 1972 by Kobe Shipyard of Mitsubishi Heavy Industries Ltd. Although the use of QFD has grown in Japan over time, its spread to the rest of the world has been gradual. The earliest cases of using QFD in the United States were after 1986, when Ford and Rank Xerox were among the first to do so. Following that, other firms such as AT&T Bell Labs, Digital Equipment, Procter & Gamble, and Hewlett-Packard began to use it. (Zairi, M., & Youssef, M. A., 1995)

Construction agencies, restaurants, and airlines have also used QFD in non-manufacturing contexts. While it is most often associated with new product development (NPD), QFD may also be applied to current products, services, and procedures. Many sectors, including software development, use QFD. (Zairi, M., & Youssef, M. A., 1995)

QFD is an excellent chance to move from the approach that company knows better what the consumer needs and adopt the approach to hear the customer's voice. In certain ways, it allows the company to be more vigilant about product issues rather than reacting to them and asking for consumer complaints. QFD also allows businesses to equate their product quality levels to those of their rivals, assisting them in determining ways to gain a comparative advantage. QFD has three main goals: identifying who the customer is, determining what the customer wants, and determining how to meet the customer's needs. (Zairi, M., & Youssef, M. A., 1995)

The Japanese place a high value on the term deployment. In Japan, deployment entails increasing one's role in or broadening one's scope of operations. In Japan, QFD is regarded as the most effective tool for raising awareness of the importance of focusing on consumer needs and encouraging company-wide accountability and dedication to meeting quality levels that are compliant with customer demands and the company's own goals. (Zairi, M., & Youssef, M. A., 1995)

QFD means that consumer preferences are incorporated into new goods as early as the design stage, whether it is perceived as a mechanism, a procedure, a framework, or even a theory. As a result, it is important to conduct a consumer survey and address these requirements. Customers are taken in and engage in the design of the product with teams of design and engineering engineers in several of the Japanese businesses that have adopted QFD. The strength of QFD lies in its ability to re-examine customer-defined hows in order to determine true customer whats. (Selecman, W.H., 1990)

Quality function deployment (QFD) can have a number of advantages, including:

- Established product parameters that satisfy the needs of the consumer while keeping an eye on the competition
- The customer's needs and the product's observable features are ensured to be in synchronization
- Ensured continuity between the preparation and manufacturing processes
- Educating and persuading all individuals in charge of various phases of the process about the link

between the quality of each phase's performance and the quality of the end output.

- Enabling things to move more smoothly when preparation occurs early in the process, reducing the likelihood of erroneous conceptions of goals and targets. (Adams, R.M. & Gavoor, M.D., 1990)

A survey regarding application of QFD in Japanese industry that was conducted by Adao states that companies used QFD in their business processes reported certain advantages and it helped the organizations to reach the following:

- At each point of the growth and manufacturing processes, translate consumer requirements into practical (technical) requirements.
- Provide a systematic framework for tapping into all of an organization's NPD expertise and facilitating its monitoring and influence.
- Brings staff from different disciplines together which makes it easier to form teams that will fulfill consumer needs. (Zucchelli, F., 1992)

Other advantages identified by Japanese companies using QFD include:

- a 50% reduction in the number of technical changes
- a 50% reduction in design cycle time
- a 50% decrease in the number of engineering modifications
- and improved consumer loyalty and less warranty requests (up to 50 per cent).

QFD appears to have a greater quality, lower cost, quicker lead time, and a substantial marketing advantage as its bottom line. (Zairi, M., & Youssef, M. A., 1995)

Quality function deployment (QFD) has been used and proved to be useful in identifying key roles of all users in order to efficiently and truly consider consumer needs. The aim of QFD is to improve the consistency of both the software development process and the end goods. (Shin, J. S., & Kim, K.-J., 2000; Shen, X. X., 2001) It should be seen as a method for negotiating cost requirements. Owing to the ambiguity and confusion surrounding the significance assigned to customer requirement judgment in QFD, it is crucial to figure out how important client requirements are. To estimate the value weights of user needs, Kwong and Bai (2002) proposed an analytic hierarchy process (AHP) based on fuzzy scales which aims to specify the importance weights of customer requirements. Figure 3 depicts the QFD process of software development. The QFD matrix defines the relationship between consumer needs and functions of software. Below is the illustration of how the matrix is used (Haag, S., et al. 1996; Liu, X. F., 2000):

- (1) User requirements are collected to be used by appropriate stakeholders and placed on the left;
- (2) the requirements are translated to technical criteria with the aid of stakeholders and placed on the top;
- (3) the stakeholders are then asked to fill in the matrix with their perceived correlations;
- (4) a list of requirements priorities is established.
- (5) A priority list of technical requirements is created.

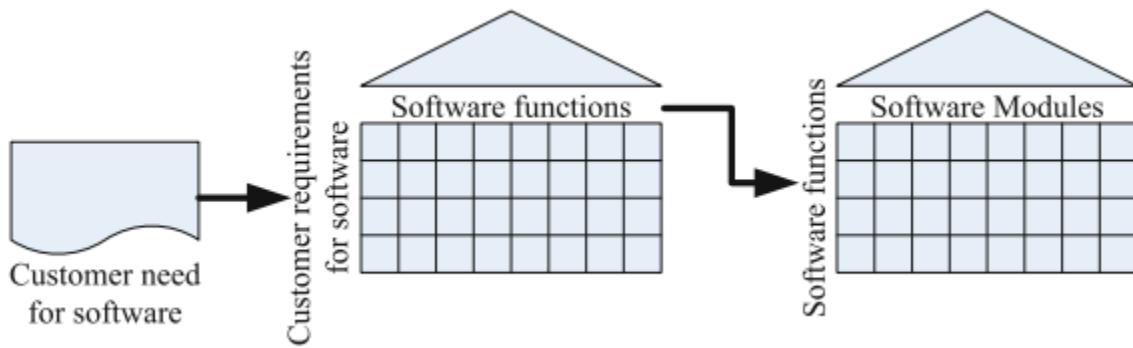


Figure 6 QFD in software (Haag, S., et al. 1996; Liu, X. F., 2000)

Customer needs ("whats"), product requirements ("hows"), customer preferences, design requirements' priority, correlations between hows, and correlations between hows and whats are all stored in a QFD matrix, often known as the "house of quality tables." The six aspects of the house of quality are depicted in Figure 6. (Sun, H., Zhao, Y. and Yau, H.K., 2009)

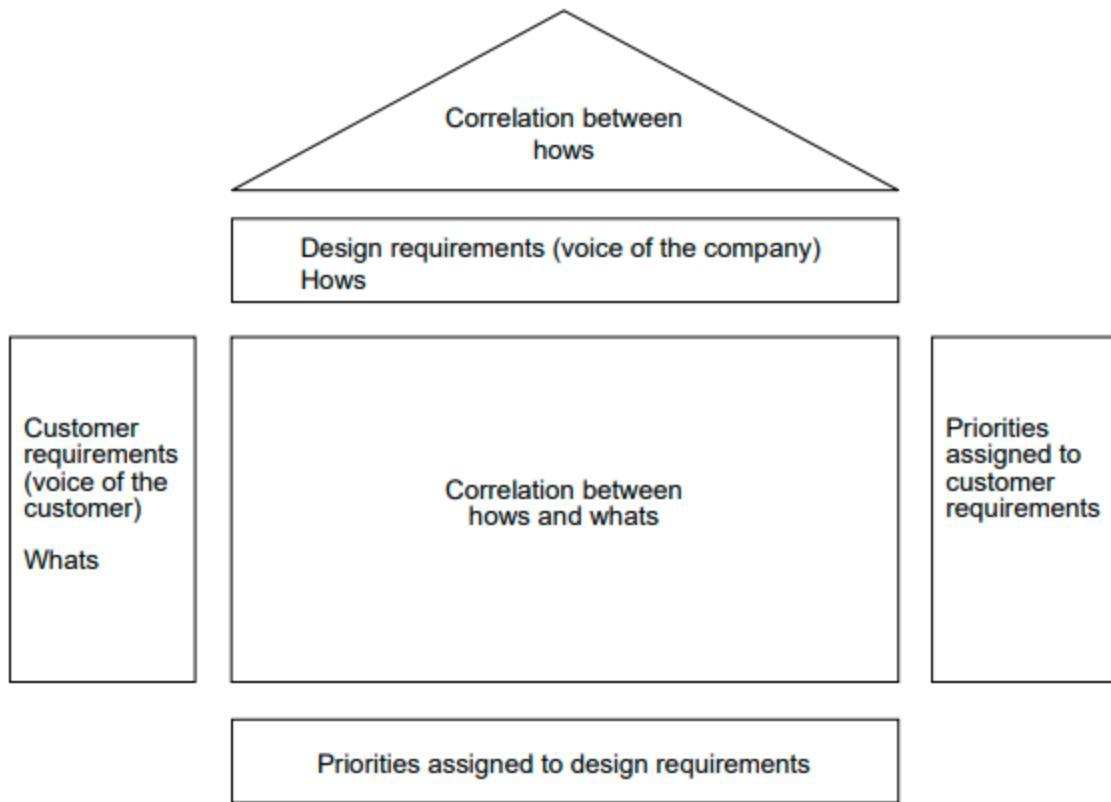


Figure 7 Basic components of QFD (Sun, H., et al., 2009)

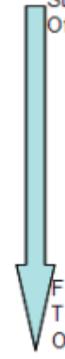
QFD is a good way to do parallel engineering jobs. Its core philosophy is to put people from various functions together to collaborate on key product, operation, and manufacturing parameters. Simultaneous engineering was a complex idea to implement in the past because it appeared to be adopted by a small number of individuals without broadening participation or making a significant effort to transform the mindset of "business as normal." Essentially, QFD eliminates the "kick it over the wall" strategy to establishing the customer-supplier chain. Simultaneous innovation enhances QFD to ensure that the customer's input is heard in the architecture of the manufacturing process. Its goal is to improve process design consistency while also increasing design performance. (Zairi, M., & Youssef, M. A., 1995)

The only way to consistently produce high-quality innovation at the right pace, to the right client, and in the right amounts is to handle key processes using QFD and parallel technical activities. Since the aforementioned procedures are continually reviewed and updated, managing key processes in this format is perhaps the perfect way to instigate the "never-ending change" principle. (Zairi, M., & Youssef, M. A., 1995)

3.2.3 Configuration and Customization Competency Model

Configuration and customization Competency Model is suggested to help with strategy specification and execution discussions around SaaS configuration and customization. From Five degrees of competences are identified, ranging from "Entry" "to "World Class". These degrees are explained in the figure 7. This approach may be used in the evaluation of SaaS applications to establish improvement targets for configuration and customization by comparing the competency level of industry leaders. Various levels of expertise can permit various levels of variability using a range of technological approaches supported by a variety of SaaS services, ranging from a completely standard offer across all tenants to a fully tenantized offering for each individual tenant. The greater the skill level, the more clients and complicated variation needs the SaaS service can handle in principle. Various SaaS suppliers, on the other hand, may have different tactics in terms of client categories targeted, supported scope of variance, and so on. Even if the SaaS service's configuration and customization competency is only at the "Entry" or "Aware" level, if the SaaS strategy is effectively established, the SaaS service can be successful on the market. (Sun, W., et al., 2008)

Level of Competency	Description	Approach	Variance Level Supported
Entry	Highly standardized offering without any configuration and customization support	Well design the functionalities as standardized offering to cover targeted customers	None 
Aware	Relatively standardized offering with pre-defined variance points	Offer parameterized configuration	Low 
Capable	Relatively standardized offering with user defined configuration	Offer self serve configuration tool to empower customers	Medium 
Mature	Base offering with programmable environment to enable user preferred customization	Offer scripting based programming for very flexible customization	High 
World Class	Offer a platform supported by programming model and tools to enable extremely strong customization or even new application development	Offer well defined programming model and tools to enable extensive customization and new application development	Extremely High 



Completely Standardized Offering

Fully Tenantized Offering

Figure 8 SaaS Configuration Competency Model (Sun, W., et al., 2008)

A SaaS service's configuration and modification can be done in a variety of ways. Summit Strategies Inc investigated SaaS configuration and customization options across three software implementation layers: Presentation Logic, Application Logic, and Database Logic. This issue is analyzed from the perspective of the client's requirements. (Sun, W., et al., 2008)

As shown in Figure 8, SaaS tenants may have a variety of configuration and customization requirements. Each tenant might present the SaaS vendor with the following objective: "I need extra fields to describe my business papers"; "Our manager needs a new report/dashboard to evaluate sales data"; "Our company does not have a procurement manager"; "Our company's process differs from what you can provide" Any of these issues may be broken down into consequences for many aspects of the SaaS service, such as data, user interface, organizational structure, processing logic, workflow, and business rules. Consider the following scenario: When a tenant wishes to modify the SaaS vendor's default data structures, the configuration and customization tools should allow them to do so. (Sun, W., et al., 2008)

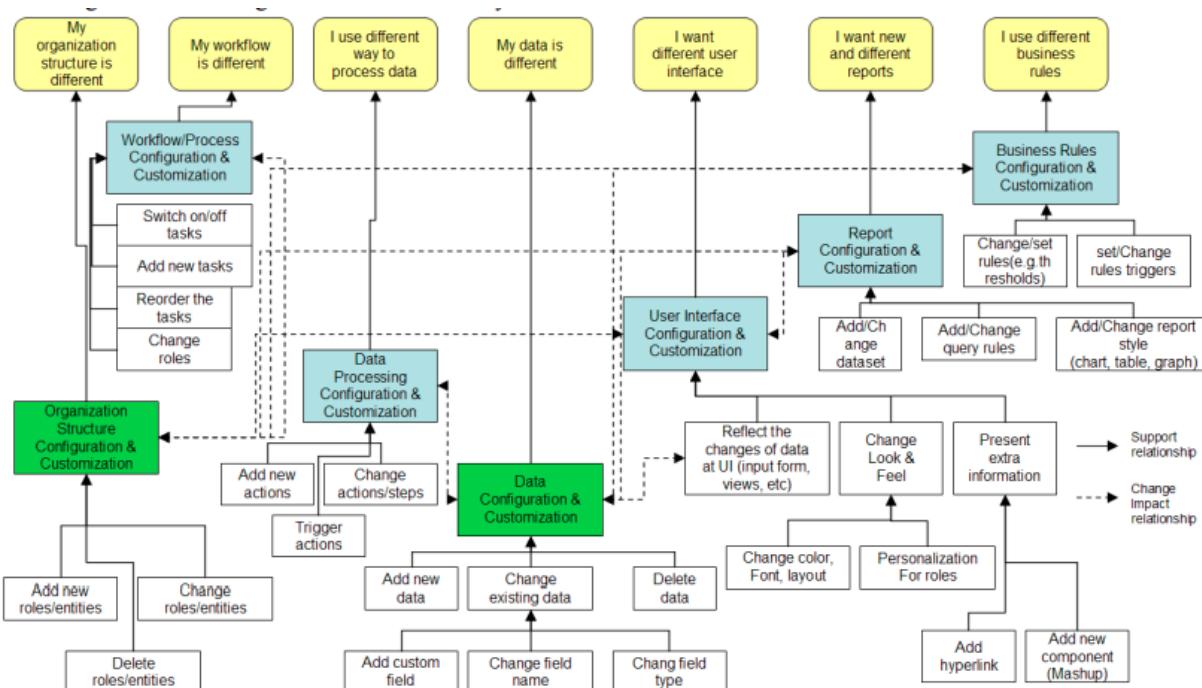


Figure 9 Perspectives of Configuration and Customization Requirements (Sun, W., et al., 2008)

Adding custom field, changing field name and type are all options. Whenever a tenant wishes to alter a SaaS vendor-built workflow, the configuration and customization tools should allow them to switch on/off Tasks, add or reorder new tasks, and change roles for a task. The "change impact relationship" lines on the diagram demonstrates that "Data Configuration and Customization" and "Organization Structure Configuration and Customization" are the two most significant perspectives; any modification to these two perspectives could have a significant impact on many other perspectives, such as User Interface, Workflow, and Business Rules. (Sun, W., et al., 2008)

For instance, if a tenant modifies a data structure, the user interface supporting data input and viewing should be updated as well; if a tenant changes the roles' description, the workflow for tasks performed by those roles should be updated as well. As a result, SaaS suppliers should devote a lot more time and effort to establishing well-designed Data and Organization Structure layers that allow for easy setup and modification. It is crucial to think about the consequences and build connections between other software products and changes in data and organizational structure. (Sun, W., et al., 2008)

4 METHODOLOGY

There is a lack of research on startup companies operating in asset management (Frisk, E., 2014). Startup company in question identified a need to research which option is more feasible in their case: configuration/personalization or mass production. What is more, there is a need to analyze what strategy is more suitable for gaining and retaining customers. The method is based on a single-case.

The case company was selected due to the author's interest in the asset management and the author is able to get access to the required data to conduct the research. What is more, having screened google scholar and google trends, there is not much research available regarding startups in asset management. In addition, there is a growing interest towards SaaS startups. (Frisk, E., 2014)

4.1 Data gathering

Data is going to be provided by the case company in question. In order to conduct the research, there is a need to conduct series of interviews to obtain sufficient information for Delphi method. QFD method implies gathering users' requirements in order for the QFD matrix to be built. The above-mentioned methods allow to understand which is the better option: customization or mass production. A Framework helps to plan and execute configuration and customization strategy. Profitability estimates are built upon the results obtained in the above-mentioned research and data. Data is generated via email, phone and online communications.

4.2 Data analysis

Firstly, the case is described and the objectives for the research are set. Data collected during Delphi method would be analyzed to understand the insights of the experts regarding the research questions of the thesis in question and identify the features that are on demand by the customers. QFD enables to understand the customers' preferences and needs better. Based on the findings of the above-mentioned methods, a Framework to plan and execute the chosen strategy is to be defined.

Based on the methods used it is possible to make some conclusions regarding the most feasible way to deliver to the customer, whether solutions should be personalized for each client or there should be some core product that the company is going to offer to the customers. What is more, the research answers to

the questions regarding the best strategy for gaining and retaining customers for a startup asset management company and for a SaaS company offering property asset management application.

5 RESULTS

The obtained results are described in this chapter. The research embarks with the Delphi method. Series of interviews are conducted to collect the data required for the research. The Delphi method is followed with Quality Function Deployment to identify the customer needs and to evaluate the necessity of customization. Finally, the research is concluded with insights regarding the configuration and customization competency model and provision of a framework to plan and execute the strategy.

5.1 Delphi method

5.1.1 Background

Delphi method is a wide-spread technique that is also used in software engineering. Startup company in question is positioning itself as a SaaS company and it operates in asset management. The company operates both in Finland and abroad. It serves clients in Germany, UK, Sweden, Denmark, Netherland, etc. Company supports the following languages for its application: English, Swedish, Norwegian, Finnish, Danish, Dutch and so on. What is more, users can select a number of various currencies to work with, including euro, krone, pound, etc.

According to the company's management, they are currently aiming to make a decision regarding the most feasible strategy to gain and retain customers.

5.1.2 Research questions

The Delphi method is aiming to investigate the following research questions:

RQ1. "What is the most feasible strategy to gain and retain customers for a startup asset management company?"

RQ2. "What is the most feasible strategy to gain and retain customers for a SaaS company offering property asset management application?"

5.1.3 Method

All the participants are kept anonymous to avoid any possible arguments. Information flow is structured by eliminating content that is not relevant. All the participants are asked to give their feedback regarding the answers of other participants which leads to further improvement.

The author acts as a facilitator monitoring the correct performance of Delphi method. The author distributes the questionnaires and keeps track of the experts' opinions. Gathered answers are analyzed to find common views. Several rounds are organized until the final decision is reached.

Experts are chosen from the company itself and there are two external experts as well. Experts are selected based on their knowledge in Software Engineering and their competence in Asset Management. All together there are five experts participating in the analysis. It was decided to have five experts due to the complexity of the project and availability of professionals.

The procedure described below was conducted to select the experts. Delbecq et al. (1975) gave specific instructions about how to find qualified experts for a nominal community methodology analysis, implying that the same method could be used for a Delphi study. They identified a stringent protocol that ensured the identity of appropriate experts and enabled them to engage in the research. A Delphi analysis does not rely on a random survey that is said to be descriptive of the whole population. It's a group decision-making process that necessitates trained professionals with a thorough knowledge of the issues. As a result, selecting trained experts is one of the most important criteria. (Okoli, C., & Pawlowski, S. D., 2004)

Panel structure

Experts will be divided into panels. Their size and composition are determined by the essence of the study issue as well as the measurements along which the experts are likely to differ. Professionals in customer services and marketing and in software engineering are two groups of experts in this case who may contribute to the research. These two parties will most likely have opposing viewpoints. It will be better to include committees that split these categories when the aim is to achieve a fair degree of consensus. This design also allows for comparison of different stakeholder groups' viewpoints. Each panel will consist of 2 to 3 members. The target is for at least half of the members of each panel to work for the company for which the research is conducted. This structure would collect a fair amount of views from the "within," and it will be possible to do comparisons to see if there are any gaps in viewpoints between respondents

on the inside and those on the outside. (Okoli, C., & Pawlowski, S. D., 2004) The stages of the expert selection process are depicted in Figure 7.

Stage 1: Create a knowledge resource nomination worksheet (KRNW). The Knowledge Resource Nomination Worksheet's goal is to better categorize experts before naming them, so that no relevant class of experts is overlooked. The study's first Worksheet as seen in Table 5. Delbecq et al. (1975) stressed the importance of not writing down any real specialist names at this time. It is important to maintain a high degree of abstraction by first defining expert groups.

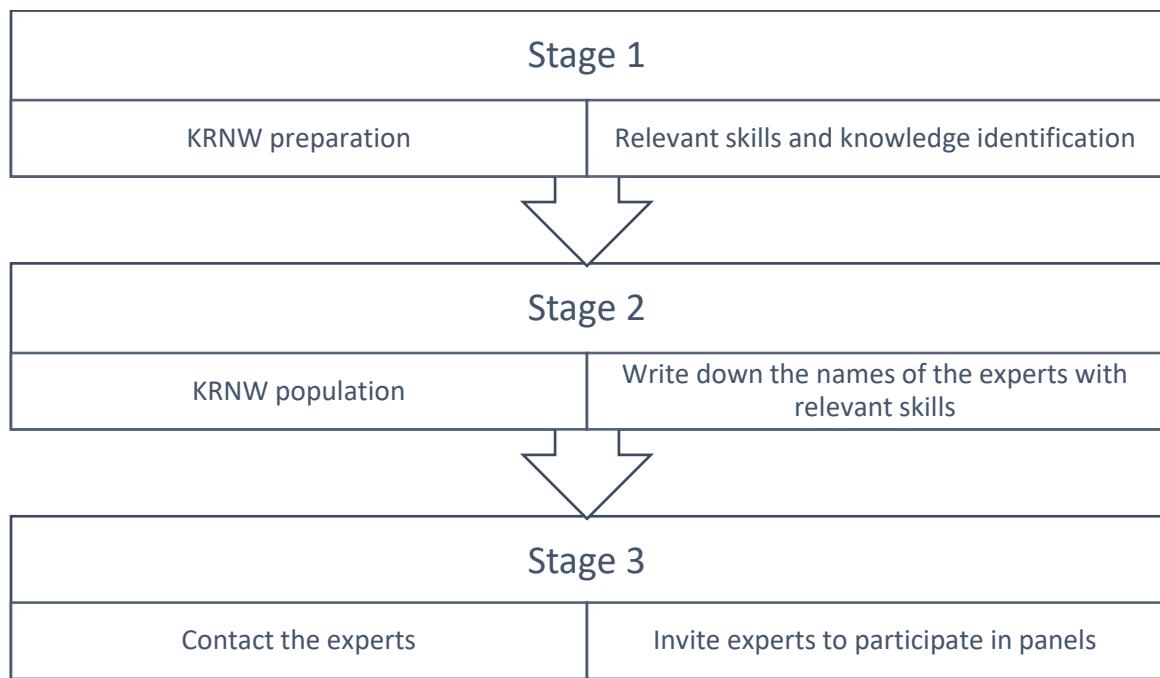


Figure 10 The stages of the expert selection process.

Table 4 Experts and their field of competence (Author's own creation)

Experts	Role/Position	Domain Expertise	Organization
Expert 1	Senior expert	Software Development	Asset management company
Expert 2	Senior expert	Customer Services and Marketing	Asset management company
Expert 3	Senior expert	Customer Services	Asset management company

Expert 4	Senior expert	Sales	Asset management company
----------	---------------	-------	--------------------------

Stage 2: Adding names to the KRNW. Following the completion of the KRNW, the following iterative process would be used to fill the groups with individual names of possible Delphi experts. Each heading (domain expertise, organizations) reflects a different approach to naming and evaluating experts, and it is anticipated that expert titles will vary significantly.

Author's personal connections are limited and skewed to her personal networks. As a result, Delbecq et al (1975) protocol is to be followed to ensure that the most competent experts are identified. After starting with personal connections, it will be proceeded over each heading in the KRNW and the gaps in the lists based on the categories will be filled.

Stage 3. Experts are to be contacted and rounds of interviews are to start.

5.1.4 The procedure for conducting the questionnaires.

E-mail is used to distribute the Delphi questionnaires. The experts participating in the interviews are located in three different countries, thus it may take few days to collect all the answers. Given that the researcher cannot send out the next questionnaire until all of the findings for a panel have been received, there might be a delay.

The interview starts with providing the brief description and setting objectives of the analysis. Timelines are also set. Each expert is asked to respond to the questionnaire, feedback on the assumptions is recorded. Collected feedback is then shared with the rest of experts and everyone is asked to comment on the new assumptions that emerged within the round interview. Rounds are repeated three times and final conclusions are drawn. Rounds are depicted in the figure 8. Interview rounds are arranged online.

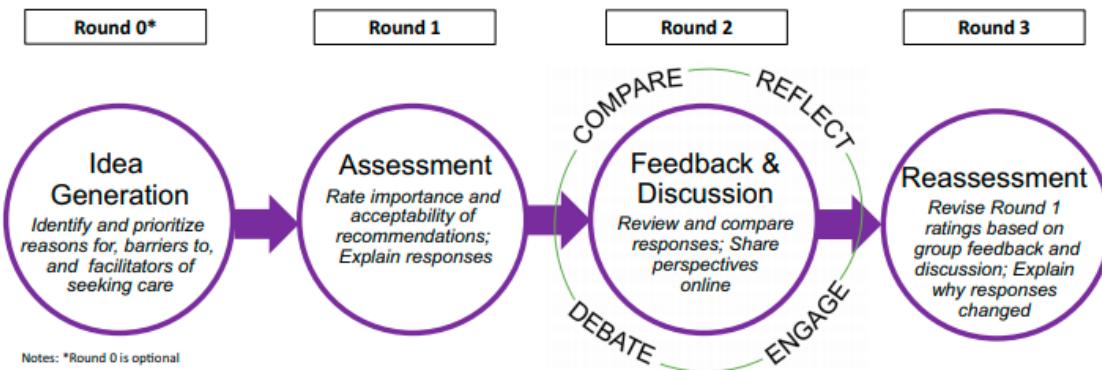


Figure 11 Delphi method rounds (Khodyakov, D. et al., 2019)

Time allocated for the Delphi analysis is one week because experts are familiar with the problem and it was discussed previously within the company. Delphi method was selected due to its simplicity to conduct and it is quick to obtain results. Delphi method allows to get initial conclusions regarding the problem the company is aiming to solve. Delphi analysis is to be conducted in three stages. Stage one is brainstorming. The main aim is to firstly treat experts as individuals and ask for their estimates and assumptions regarding the feasibility and identify possible complications. Stage two is narrowing down the initial list to the most important assumptions and stage three is rating the list of important assumptions. (Okoli, C., & Pawlowski, S. D., 2004)

Issues with questionnaire construction.

Delbecq et al. (1975) and Dillman's recommendations are to be followed for the majority of the design considerations. First, since a Delphi analysis requires far more time from respondents than a conventional sample due to the various steps and variations, one goal would be to ensure that no single questionnaire takes more than 30 minutes to complete.

5.1.5 Results

Stage one: brainstorming

Questionnaire 1: Initial list of considerations (3.2.4.1). We'll submit the first questionnaire the same day an expert decides to serve on a Delphi team, via e-mail, fax, or the Web, depending on the expert's choice. Since a Delphi survey is an open-ended solicitation of proposals, the initial questionnaire is quite easy.

Three simple questions are to be asked in the questionnaire, each of which corresponds to one of the research questions.

The questionnaire would ask experts to mention at least six relevant factors (see Schmidt) that may impact the process of gaining and retaining the customers that may be present in a startup company (RQ1). The aim of this query is to come up with all the possible complications that may become an obstacle to gain and retain the clients. Second question would tackle the same issue but from the point of view of Asset management company (RQ2). The third question would ask to list at least five complications associated with personalization and configuration (RQ3).

RQ1. "What is the most feasible strategy to gain and retain customers for a startup asset management company?"

- Mention at least 6 factors that impact the process of gaining and retaining customers in a startup company.

RQ2. "What is the most feasible strategy to gain and retain customers for a SaaS company offering property asset management application?"

- Mention at least 3-6 factors that impact the process of gaining and retaining customers in an Asset management company and are common for this specific field.

RQ3. "What are the possible complications of personalization and configuration approaches?"

- List at least 5-10 complications associated with personalization and configuration.

The questionnaire is to be submitted to all of the experts without taking into account their panel at this time. Diverse groups are more innovative than homogeneous groups in group decision-making, according to various reports. The study's nature groups experts together based on their similarity. The risk of stifled creativity may be avoided by splitting the research into three phases. The Delphi study's creative stage involves soliciting factors from participants, while the rating and weighting stages mostly require judgmental opinion. (Okoli, C., & Pawlowski, S. D., 2004)

Similar responses will be removed from the first questionnaire's responses. At this point, it would be reported how many panelists who initially listed each item there were and this will be reflected on the combined lists, and then these items will be conceptually organized into groups to make it easier for panelists to understand each list when it is used for the next phase. (Okoli, C., & Pawlowski, S. D., 2004)

Since the researcher would consolidate the lists and organize them into groups rather than the experts, the list would be sent to the professionals for validation. To define and clarify each aspect, an explanatory lexicon will be utilized, in addition to a short, one-sentence explanation provided by the experts in the initial questionnaire. The experts will get a copy of the responses to the first questionnaire. The second questionnaire will ask experts to verify and improve the factor categorizations by (a) confirming that the answers were correctly understood and allocated to the relevant category; and (b) confirming and refining the factor categorizations. According to Schmidt (1997), there is no reason to believe that a real, aggregated list has been created without this step. Experts would be able to add additional factors that they would not have considered before.

Stage one results:

List one (factors that impact the process of gaining and retaining customers in a startup company):

- Fast response to customer requests.
- Adequate requirement gathering.
- Data privacy control.
- Quality assurance.
- Gathering feedback.
- Update the customer about the upcoming potential changes in the product.

List two (factors that impact the process of gaining and retaining customers in an asset management company that might be common only for an Asset management industry):

- Understanding the industry.
- Researching about the industry related laws. (sometimes different laws in different countries).
- Understanding the different practices of the players in the Asset management industry.

List tree (complications associated with personalization and configuration):

- Different practices of the players in the Asset management industry.
- Difficulty gathering the requirements.
- Understanding the unique terms in the industry.
- Data gathering (sometimes data is in unmaintainable form)
- Most of the players need a more or less unique solution.

Stage two: narrowing down

The experts are divided into four groups in the next two stages. In a nutshell, panels will narrow down factors that represent the viewpoints (stage 2) before facilitating consensus (stage 3). The target in the second stage, which narrows down lists of factors, would be to understand the relevance ratings of the factors based on the different stakeholder groups' perspectives. Certain groups of experts could have different perspectives, and these disagreements may have significant impact on the decision-making process. As a result, rather than attempting to accommodate widely divergent viewpoints, the approach is to make groups of people who think alike decide within themselves which considerations are most appropriate. (Okoli, C., & Pawlowski, S. D., 2004)

Questionnaire 2: determining the most important considerations

The complete consolidated lists of items will then be shown to each specialist on each panel. The second questionnaire will be organized at random to eliminate bias in the order in which the elements are mentioned. Each panelist will be asked to choose at least 10 factors from each list that are relevant to:

- gaining and retaining customers in a startup company
- gaining and retaining customers in an Asset management company and are common for this specific field
- complications associated with personalization and configuration

Experts are asked to only select ten most relevant factors instead of ranking them. When all of the panelists have returned their comments, each panel will be analyzed individually and factors that were chosen by more than half of the experts will be identified. The lists would be reduced to a manageable size as a result of this step. The number of objects to rank will be limited to 10. (Okoli, C., & Pawlowski, S. D., 2004)

According to the results obtained from the second phase the following factors are pointed out by the experts:

- Gaining and retaining customers in a startup company
 1. Realistic goals
 2. Clear communication
 3. Keeping the given promises

- 4. Building trust
 - 5. Identify the customers, to whom your SaaS is suited for
 - 6. Fast response to customer requests
 - 7. Quality assurance
 - 8. Customer service
 - 9. Understanding the customer need and reacting to the need
 - 10. Fulfill the customer expectations
 - 11. Agile pilots. To have a small test if some business case works, if not, then leave that one
 - 12. Long-term commitment from customer organization and strong change management from the supplier
-
- Gaining and retaining customers in an Asset management company and are common for this specific field
 - 1. Understanding the industry
 - 2. Researching about the industry related laws
 - 3. Understanding the different practices of the players in the Asset management industry
 - 4. High quality DATA
 - 5. Right KPIs
 - 6. Reporting
 - 7. Clear rent rolls
 - 8. Visual approach
 - 9. Lack of digitalized tools, customers' reluctance to leave old fashioned tools
 - 10. Unrealistic customers' expectations about the capabilities of the system
-
- Complications associated with personalization and configuration
 - 1. Unscalability
 - 2. Different practices of the players in the Asset management industry
 - 3. Data quality is poor
 - 4. Many systems involved
 - 5. Difficulty gathering the requirements.
 - 6. Understanding the unique terms in the industry.
 - 7. Data gathering (sometimes data is in unmaintainable form)

8. Most of the players need a more or less unique solution.
9. Too little time reserved for the system deployment
10. Many stakeholders involved in the decision making

Stage 3: ranking relevant factors

The final stage's aim is to find a consensus within each panel about how to rate the related factors. Delphi groups are more difficult to find consensus in than those that require direct contact with members, according to studies. However, since the panel participants are chosen for their homogeneity, achieving consensus is easier with a panel configuration. (Okoli, C., & Pawlowski, S. D., 2004)

Questionnaire 3: ranking the selected factors

The final step is to come to an agreement on how to rate the factors. Delphi groups are more difficult to find consensus in than groups of direct contact between members. The researchers purposefully chose panel participants for their homogeneity, making it easier to reach consensus. (Okoli, C., & Pawlowski, S. D., 2004)

During this stage of the process, each panel will rate the factors on their respective pared-down lists separately. The priority order for each panel will be reflected in each ranked list. Each expert will each of three lists. Experts will also be asked to add some remarks describing or defending their rankings as part of the questionnaire.

While Schmidt (1997) proposes collecting helpful feedback, an analysis of the literature found no such clarification of rankings in any Delphi analysis. Rohrbaugh (1979) linked the Delphi approach to a social judgement analysis (SJA)-based collective decision method, which uses a structured graphical method to present the logic behind a panelist's decisions to the other members of the panel. While the findings obtained for SJA were not dramatically more reliable, the groups were able to achieve a higher level of agreement because the participants were able to understand each other's logic. According to Rohrbaugh's research, if the committees are given input on the panelists' reasonings, they may be able to reach an agreement faster.

Table 5 Remaining CSF after consolidation. Factors that impact the process of gaining and retaining customers in a Startup company (Author's own creation)

Success factor	Expert 1	Expert 2	Expert 3	Expert 4	Expert 5	Σ	%
Fast response to customer				x	x	2	40
Adequate requirement gathering	x			x		2	40
Data privacy control			x	x		2	40
Quality assurance		x	x			2	40
Gathering feedback	x		x			2	40
Update the customer about the upcoming potential changes in the product	x			x		2	40
Clear communication	x	x		x		3	60
Unambiguous pricing			x	x		2	40
Realistic goals	x	x		x	x	4	80
Team work	x	x				2	40
Customer service		x		x		2	40
Contact persons			x	x		2	40
Keeping the given promises	x	x			x	3	60
Building trust		x	x		x	3	60
Process of learning from failures	x		x			2	40
Making product "necessary" for customer	x	x				2	40
Understanding the customer need and reacting to the need			x	x		2	40
Fulfill the customer expectations	x				x	2	40
Not enough money for sales	x	x				2	40
Long-term commitment from customer organization and strong change management from the supplier		x	x			2	40
Identify the customers, to whom your SaaS is suited for	x	x	x			3	60
The right kind of partnerships. The resources are scarce thus partners have to bring clear addition value, which support the core focus		x				1	20
Agile pilots. To have a small test if some business case works, if not, then leave that one.			x		x	2	40
Not enough references				x		1	20
Unsure how long the company will exist			x		x	2	40
Not enough money for marketing				x		1	20

Total	12	12	12	12	12		
-------	----	----	----	----	----	--	--

The following factors have the biggest impact on gaining and retaining customers in a startup company and they are ranked in the following order:

1. Realistic goals
2. Clear communication
3. Keeping the given promises
4. Building trust
5. Identify the customers, to whom your SaaS is suited for

Table 6 Remaining CSF after consolidation. Complications associated with personalization and configuration in Asset management company (Author's own creation)

Success factor	Expert 1	Expert 2	Expert 3	Expert 4	Expert 5	Σ	%
Understanding the industry	x	x	x	x	x	5	100
Researching about the industry related laws	x	x	x	x	x	5	100
Understanding the different practices of the players in the Asset management industry	x	x	x	x	x	5	100
High quality DATA	x	x	x	x	x	5	100
Right KPIs	x	x	x	x	x	5	100
Reporting	x	x	x	x	x	5	100
Clear rent rolls reluctance to leave old fashioned tools	x	x	x	x	x	4	80
Reacting fast			x	x		3	60
Co-operation with big companies						0	0
Visual approach		x		x	x	4	80
Lack of digitalized tools	x	x	x		x	4	80
Unrealistic customers' expectations about the capabilities of the system	x	x	x	x	x	5	100
Total	10	10	10	10	10		

The following factors have the biggest impact on gaining and retaining customers in an Asset management company and are common for this specific field:

1. Understanding the industry
2. Researching about the industry related laws
3. Understanding the different practices of the players in the Asset management industry

4. High quality DATA
5. Right KPIs
6. Reporting
7. Unrealistic customers' expectations about the capabilities of the system

Table 7 Remaining CSF after consolidation. Factors that impact the process of gaining and retaining customers. (Author's own creation)

Success factor	Expert 1	Expert 2	Expert 3	Expert 4	Expert 5	Σ	%
Different practices of the players in the Asset management industry	x		x	x	x	4	80
Difficulty gathering the requirements		x		x		2	40
Understanding the unique terms in the industry	x	x				2	40
Data gathering (sometimes data is in unmaintainable form)		x			x	2	40
Most of the players need a more or less unique solution				x	x	2	40
Misunderstandings in the communication						0	0
Data quality is poor	x	x	x	x	x	5	100
If contact persons are not committed enough and available, deployment might get delayed			x	x		2	40
Customers' expectations are too high		x	x			2	40
Getting out of your own scope	x			x		2	40
Getting too complex offering	x	x				2	40
Product management difficulties	x		x			2	40
Unscalability	x	x	x	x	x	5	100
Focus is too narrow					x	1	20
Not getting commercialised fast enough				x		1	20

What customer wants, is not necessarily what customer needs		x	x			2	40
Too little time reserved for the system deployment	x	x				2	40
Locked to one use case - unable to other possibilities with the system				x	x	2	40
Many stakeholders involved in the decision making	x		x		x	3	60
Many systems involved	x	x	x	x	x	5	100
Country specific differences					x	1	20
Total	10	10	10		10	10	

The following factors have the biggest impact on complications associated with personalization and configuration:

1. Unscalability
2. Different practices of the players in the Asset management industry
3. Data quality is poor
4. Many systems involved

Schmidt offered a great and extensive guideline of rules to follow when it comes to quantitatively establishing the rankings of the items in the lists, which is going to be utilized as the foundation of the research. Kendall's W coefficient of concordance is widely acknowledged as the finest measure for quantifying non-parametric ranks. (Siegel, S., 1956) W is a score that goes from 0 to 1, with 0 indicating no agreement and 1 representing complete agreement between lists. Schmidt produced a table that explains how to interpret various W values, with 0.7 representing good agreement. The W value advises how to proceed in the ranking after computing the concordance within each panel. A W value of 0.7 or higher indicates acceptable agreement, and the ranking step is considered complete. To compute the final rating for a completed panel, mean ranks for each item might be utilized. (Okoli, C., & Pawlowski, S. D., 2004)

5.2 Quality function deployment

Customization levels are often specified during the product design process to describe which components, pieces, or modules, also known as configuration items, can be changed and chosen according to client expectations. Any customized product is created based on the requirements of the client and must fulfill the diverse needs of the product's users. To satisfy the demand for producing customized goods with a cost-effective solution, product structure and design approaches, such as a product configured from modules, are required. The objective of product variation management is to provide as many product versions to the client as feasible while keeping internal variety to a minimum. Modularity and dependability, as well as cycle time and the implementation of manufacturing process reconfiguration, should all be included in product design criteria. Cost, product quality, and market responsiveness are the three major aims of each production system. (Kutschchenreiter-Praszkiewicz, I., 2018)

The reconfigurable manufacturing system (RMS) is a new type of production system that was recently suggested. RMS is defined by modularity and the capacity to update itself in order to respond to dynamic requirements or unforeseen failures. Modules are built with interfaces for component integration, and all key components are modular. (Kutschchenreiter-Praszkiewicz, I., 2018)

Quality function deployment (QFD) is one of the ways for customizing products that takes into account client needs as well as product and manufacturing process features. QFD is being developed as a way for translating qualitative user demands into quantitative parameters, introducing functions that create quality, and implementing techniques for achieving design quality in subsystems and component components, and lastly in specific manufacturing process elements. (Kutschchenreiter-Praszkiewicz, I., 2018)

Quality Function Deployment (QFD) is a strategy for identifying customer needs and translating them into accurate production plans. The "voice of the customer" refers to these explicit and unspoken client expectations and requirements. The customer's voice is captured through direct discussion, interviews, surveys, focus groups, customer needs, observation, warranty data, field reports, and other means. Following that, a product planning matrix, or "house of quality," is created based on the client's needs. These matrices translate higher-level "what's" or requests into lower-level "how's" - product specifications. These matrices are applied to translate higher-level "what's" or requests into lower-level

"how's" - product specifications or technical characteristics that will satisfy those requirements. (Kiran, D.R., 2017)

In order to identify what features should be included into the core application, quality function deployment method is applied. In order to construct the QFD, it is required to conduct some interviews with the company staff who possess knowledge regarding customer needs. QFD is applied in software engineering in order to reach customers' satisfaction. Besides this, QFD might also give structural principles for re-designing services and generating new products, while previous published work on service quality has been focused on describing and measuring service quality and customer satisfaction. (Shahin, A., n.d.)

The minimization of management risk is an essential aspect of software quality. Delay in delivery, going beyond the budget, poor product performance, and a short product lifecycle are all management risks that must be managed in order to achieve software quality. These management risks are related to the software development process rather than the software product itself. Quality Function Deployment (QFD) is a powerful technique for quality analysis and improvement. The benefits of QFD are as follows:

- Design modifications are made less frequently and earlier
- The time it takes to develop a product is cut in half
- There are less issues at the beginning
- Documentation is simpler
- There are fewer field issues
- Reduction in warranty claims
- Cross-functional teamwork is being developed
- Enhanced design dependability
- Customer satisfaction (Dillibabu, R., et al., 2011)

The intangible character of service processes, which are generally associated with customer engagement and interaction, is one of the challenges with applying QFD to services (Eringa, K. & Boer, I.L.J., 1998). Customer expectations are typically trade-offs, as is the requirement to establish the degree of interrelationships between engineering features and set exact target values for them. Many of these judgments must be made on the basis of subjective judgment. It's also worth noting that most QFD users quit after creating their own customized version of HoQ. (House of Quality). This is partly due to a lack of

clarity in the literature on how to use the charts used in the QFD process's later phases. (Bouchereau, V. & Rowland, H., 1998)

Because of the specific characteristics of software development, traditional QFD must be modified for use in software deployment. The first distinguishes software development from other manufacturing processes in that it is not a repeatable process that must be customized for each product. QFD was created with the intention of implementing client needs across the product development process, from design through manufacture. A set of design specifications is followed while developing software parts. This would be an element of product rollout in QFD. (Dillibabu, R., Krishnaiah, K. & Baskaran, R., 2011)

The second key distinction between manufacturing and software development is that technical specifications do not immediately meet consumer needs. Software or information systems, unlike produced items, are generally designed to offer support infrastructure or function as an enabler for an organizational activity. Customer needs are converted into technical specifications for a specific product in QFD. Customer demands are addressed in the creation of software or information systems by offering particular system functions. One or more software systems may be required to perform these system functions. A customer who needs "precise inventory information," for example, will want inventory monitoring software, purchasing and receiving software, and cost management software. (Dillibabu, R., Krishnaiah, K. & Baskaran, R., 2011)

QFD has four phases modified in order to fit for the purposes of software product development. These phases are depicted in the figure below.

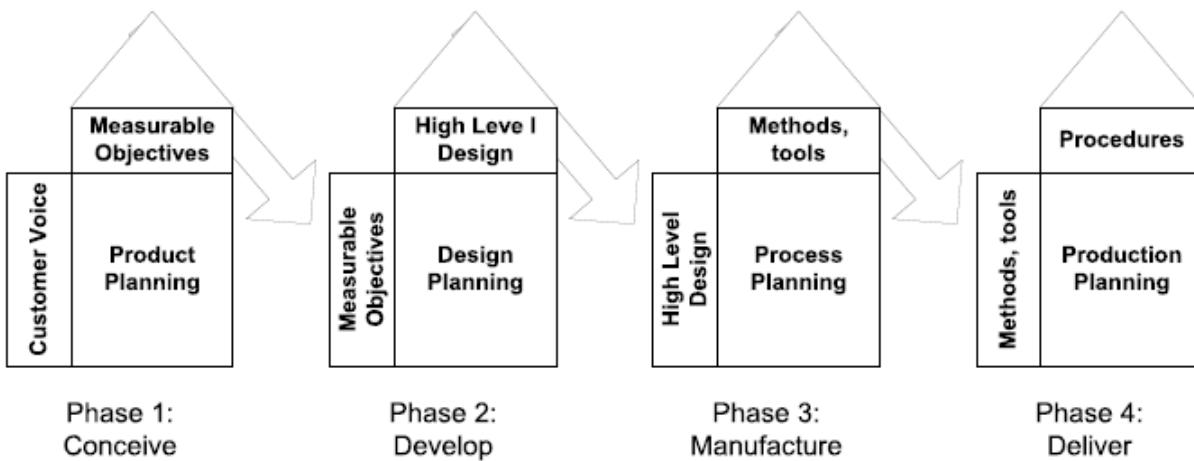


Figure 12 Adapting QFD to Software (Cohen, L., 1995 & Hauser, J.R. and Clausing, D., 1996)

According to the data obtained from the interviews, the following requirements ("what's") are collected:

- Accurate user-friendly reporting (fund reporting, rent-related reporting, KPIs, asset reporting)
 - Meaning that calculations in the report should be accurate and repeatable. Customer should be aware of the formulas used to calculate the values. The report should be accessible from different devices and with a variety of tools which are standard in the industry.
- Automatic report creation (fund reporting, rent-related reporting, KPIs, asset reporting)
 - Meaning that the user does not have to do any manual work to generate the reports. The system generates them automatically and deliver them to the stakeholders via email or by other means of communication which are in use (example, SMS)
- Low cost of data ownership
 - Means that the service provider should create storage places which are not very costly for the users. This can be done by compressing the data and avoiding data duplication.
- High availability
 - Sharing the data between interested parties is very important. This can be accomplished by using Application Programming Interface (API) which means that any 3rd party developer with the right permissions can utilize the data and build their own interfaces.
- High performance
 - Meaning providing enough memory and processing power. This can be dynamic which means as the demand grows, processing power increases.

- Balance in system design
 - System should be designed in a way that the code can be reused and the system is less prone to bugs, errors and security risks by following the system development standards
 - Code simplification (dividing the code into different modules and following modern architecture like Model View Controller (MVC))
- Rapid modular system expansion
- Redundancy
 - Meaning that there are additional or duplicated resources available to support the system (having extra servers on stand-by to go live in case problems in the main server occur)
 - Redirect the logged in users to the other servers in case of a problem in the main server
- Compatibility with existing systems
 - There are different data transfer models in the industry like plain text, databases, APIs, graphics and paper. Integrating all these different data sources is a challenge that should be done in the interest of the data accessibility.
 - Integration means converting these different data sources to each other based on users' requirements.
 - This will require: processing different data formats, converting data formats to each other, API.
- Easy data management
 - This will require provision of sufficient memory.
 - It means that reports should be delivered automatically by means of email, for instance.
 - Duplication should be avoided.
- No special environment requirements
 - This means that user does not have to install any special or paid tools to access the system.

The below listed encounters ("how's") are obtained:

- Export reports in formats like pdf, excel
- Create a secure link for the report to access from anywhere
- Follow industry standards when creating the report
- Create automatic periodical reports for users
- Delivering report automatically (email)
- Compress the data

- Avoid data duplication
- Provide low cost data storage
- API
- Provide access levels
- Provide enough memory
- Sufficient processing power
- Code sustainability (reuse code)
- Code simplification
- Additional servers on stand-by
- Users redirection to other servers

Requirement and encounters are organized into the matrix according to the example below. Relationships are identified.

		Service encounters						
		Encounter 1	Encounter 2	Encounter 3	Encounter 4	Encounter 5	Encounter 6	Encounter 7
Common customers' Requirements (CR)	CR 1	■	■				□	
	CR 2	■	■					
	CR 3	□		□				
	CR 4			■	□		■	■
	CR 5	■		□		■		
	CR 6							■
	Added CR (for encounter no. 1)	■		■	□			■

Figure 13 An example of integrative service QFD (Shahin, A., n.d.)

Through brainstorming with client representatives, the links between customer requirements and technological requirements are formed. Every what and every how are linked in these connections. All connections are divided into three categories: strong, medium, and weak. Various symbols are used to

represent various levels of relationship strength (circle with dot in the center signifies a strong relationship, blank circle signifies a medium relationship and triangle signifies a weak relationship). The allocation and classification of the relationships is done with great care and attention, and it is double-checked several times. (Dillibabu, R., Krishnaiah, K. & Baskaran, R., 2011)

Property management software Yardi Voyager and DoorLoop, n.d. are listed as competitors of the product in question. These companies were selected because Yardi is an identified competitor of the company in question and DoorLoop, n.d. is a promising start up that can become an actual competitor in future. Yardi has established itself as the industry's top provider of high-performance software solutions. With a mix of responsiveness and technical innovation, Yardi has set the bar for real estate software solutions. Yardi has the technical knowledge and vision to convert the understanding of the industry's diverse segments' specific demands into strong solutions that meet and surpass those expectations. (Yardi, n.d.)

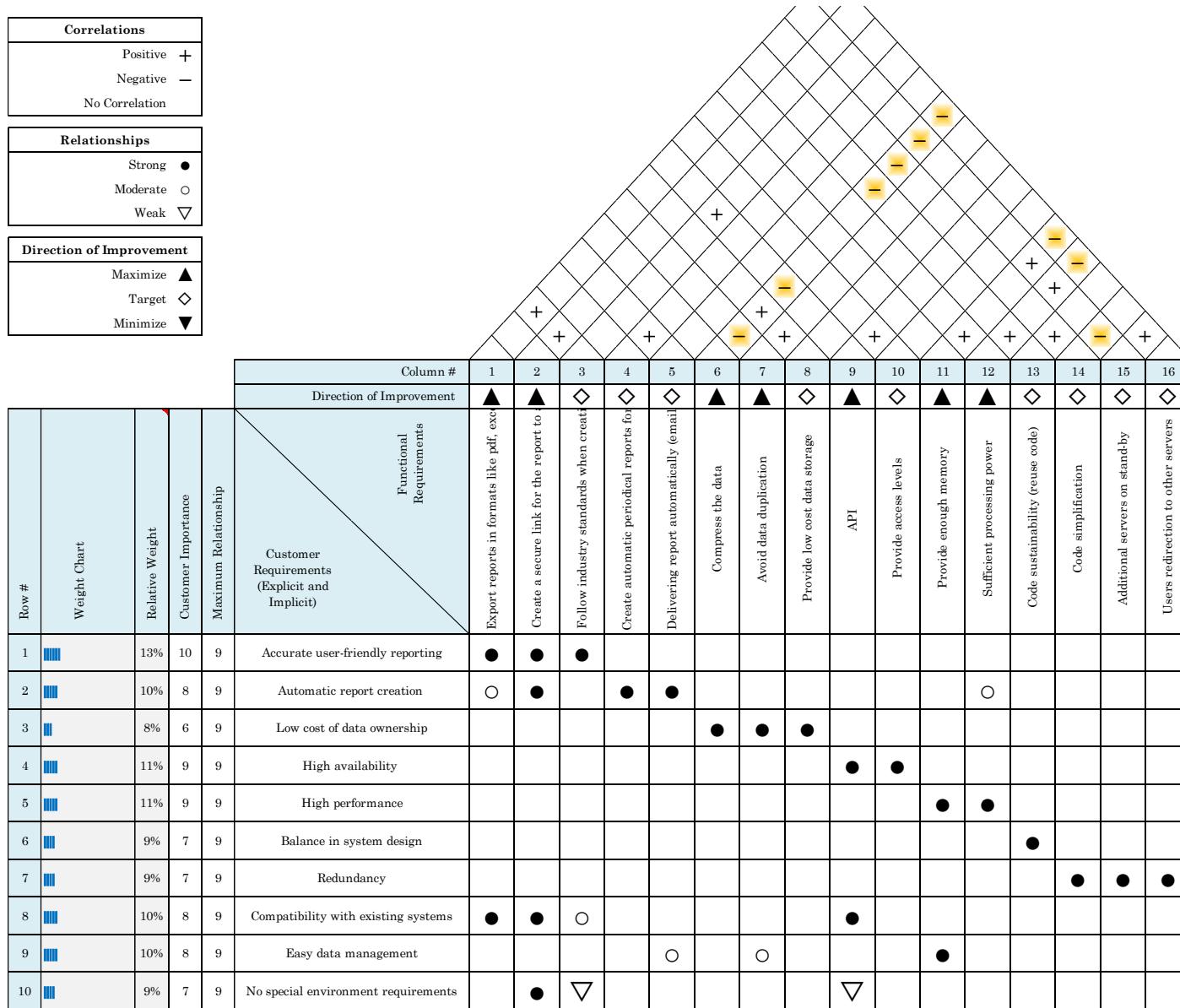


Figure 14 QFD Matrix (Author's own creation)

Property managers, property management firms, owners, landlords, investors, and renters may all benefit from DoorLoop, n.d.'s property management software. It is praised for its adaptability and ease of usage. The platform is so adaptable that it can be used by anybody, from a single property manager to a large corporation with thousands of properties all over the world. (Gilbert, N., n.d.) Below are pros and cons of each system concerning a particular characteristic:

Table 8 Comparison table of the product in question and its competitors (Author's own creation)

	Product in question	Yardi Voyager	DoorLoop, n.d.
Accurate user-friendly reporting	<p>Pros</p> <ul style="list-style-type: none"> • Export reports in formats like pdf, excel • Create a secure link for the report to access from anywhere • Follow industry and EU standards when creating the report • Easy to use platform <p>Cons</p> <ul style="list-style-type: none"> • Absence of customizable field names (ex. Lease, rent roll) • Absence of hiding/showing fields in the report 	<p>Pros</p> <ul style="list-style-type: none"> • Diverse custom reporting with several editable user fields available (TrustRadius, n.d.) • State and Federal Compliance, works all around the world (TrustRadius, n.d.) • Easily switch from property to property (TrustRadius, n.d.) <p>Cons</p> <ul style="list-style-type: none"> • Challenging to learn for a new user (TrustRadius, n.d.) • Some features can be hard to find (TrustRadius, n.d.) • Confusing layout (TrustRadius, n.d.) 	<p>Pros</p> <ul style="list-style-type: none"> • Intuitive design (Getapp) • Easy to use (Capterra, n.d.) <p>Cons</p> <ul style="list-style-type: none"> • Zapier integration is still not out
Automatic report creation	<p>Pros</p> <ul style="list-style-type: none"> • Export reports in formats like pdf, excel • Create a secure link for the report to access from anywhere • Create automatic periodical reports for users (time-period of 	<p>Pros</p> <ul style="list-style-type: none"> • Create automatic periodical reports for users (TrustRadius, n.d.) • Strong reporting, especially financial reporting (TrustRadius, n.d.) 	<p>Pros</p> <ul style="list-style-type: none"> • Many automated features that save time (Trustpilot) • Custom reporting (Softwareadvice, n.d.)

	<p>the report should be set, then source of data is selected, client should choose what kind of report is needed and then the report is downloaded manually)</p> <ul style="list-style-type: none"> • Delivering report automatically (email) 	<ul style="list-style-type: none"> • Automated workflow (Financesonline, n.d.) • Easy scheduled reporting (Saxonypartners, n.d.) <p>Cons</p> <ul style="list-style-type: none"> • Adhoc report creation is not easy (TrustRadius, n.d.) • Custom resident email creation (TrustRadius, n.d.) 	
Low cost of data ownership/Overall cost	<p>Pros</p> <ul style="list-style-type: none"> • Compress the data • Avoid data duplication • Provide low cost data storage • Fee starts from 60e/month 	<p>Cons</p> <ul style="list-style-type: none"> • High cost of customer service and help (TrustRadius, n.d.) • Fee starts from 100e/month (Yardi website) 	<p>Pros</p> <ul style="list-style-type: none"> • Affordable price starting from 49 dollars (DoorLoop, n.d.)
High availability	<p>Pros</p> <ul style="list-style-type: none"> • API • Provide access levels • Available on a mobile devices 	<p>Pros</p> <ul style="list-style-type: none"> • API • Easy access availability (Softwareadvice, n.d.) • Platform can be extended (Financesonline, n.d.) • Available on a wide amount of devices (Financesonline, n.d.) 	<p>Pros</p> <ul style="list-style-type: none"> • Available on a wide amount of platforms including Mac, Windows, Linux, etc. (Slashdot) • API
High performance	<p>Pros</p> <ul style="list-style-type: none"> • Provide enough memory 	n/d	n/d

	<ul style="list-style-type: none"> Sufficient processing power 		
Balance in system design	<p>Pros</p> <ul style="list-style-type: none"> Code sustainability (reuse code) 	n/d	n/d
Redundancy	<p>Pros</p> <ul style="list-style-type: none"> Code simplification Additional servers on stand-by Users redirection to other servers 	n/d	n/d
Compatibility with existing systems	<p>Pros</p> <ul style="list-style-type: none"> Export reports in formats like pdf, excel Create a secure link for the report to access from anywhere Follow industry standards when creating the report Create automatic periodical reports for users API 	<p>Pros</p> <ul style="list-style-type: none"> Easy customization with API (Financesonline, n.d.) <p>Cons</p> <ul style="list-style-type: none"> Can be challenging to export data from excel (TrustRadius, n.d.) 	<p>Pros</p> <ul style="list-style-type: none"> API Open API infrastructure (Financesonline, n.d.) <ul style="list-style-type: none"> DoorLoop, n.d. has a team that migrates data from, for example, excel to the platform (Financesonline, n.d.)
Easy data management	<p>Pros</p> <ul style="list-style-type: none"> Delivering report automatically (email) Avoid data duplication Provide enough memory 	<p>Pros</p> <ul style="list-style-type: none"> Access via mobile devices (Financesonline, n.d.) <ul style="list-style-type: none"> Flexible management platform (Financesonline, n.d.) 	<p>Pros</p> <ul style="list-style-type: none"> Access via mobile devices (Financesonline, n.d.)
No special environment requirements	<p>Pros</p> <ul style="list-style-type: none"> Create a secure link for the report to access from anywhere API 	<p>Pros</p> <ul style="list-style-type: none"> Running product in certain browsers is not possible (TrustRadius, n.d.) 	<p>Pros</p> <ul style="list-style-type: none"> Web-based interface (Financesonline, n.d.) <p>Cons</p>

	<ul style="list-style-type: none">• Does not work in old versions of browsers (Firefox and Chrome are recommended)		<ul style="list-style-type: none">• Does not have native applications for iOS and Android devices (Financesonline, n.d.)
--	--	--	--

Based on the pros and cons listed in the table above the product and competitors were assigned a number between 0 to 5 depicting how well it fulfills the users' requirement. The grading is shown in the figure below.

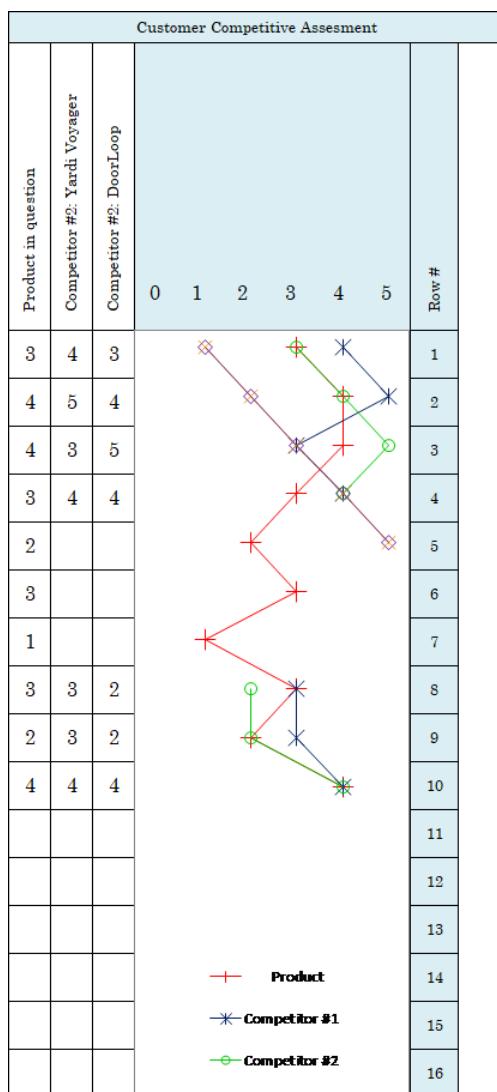
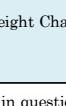
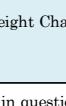
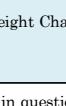
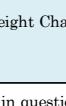
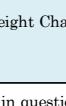
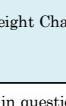
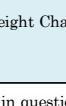
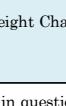
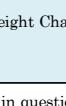
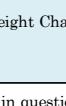
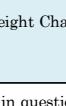
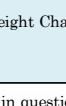
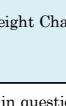
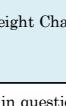
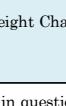
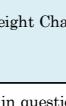
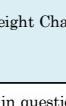
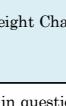
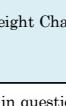


Figure 15 Product in question and its competitors' assessment (Author's own creation)

According to the reviews studied it can be concluded that Yardi is more established product that offers a wide range of features to its customers. Company was established in 1982. DoorLoop, n.d. is a startup launched in 2019 and it is providing more customized-solutions.

Below is the table that depicts the customer preferences regarding the product in question. Based on this rating, it can be concluded that the company in question has to work on creating a secure link for the report to access from anywhere, delivering report automatically (email), additional servers on stand-by and users redirection to other servers. These requirements are not currently fulfilled. Creating automatic periodical reports for users and API have to be improved.

Table 9 Customer preferences regarding the product in question (Author's own creation)

Target	Target for Functional Requirement #1	Target for Functional Requirement #2	Target for Functional Requirement #3	Target for Functional Requirement #4	Target for Functional Requirement #5	Target for Functional Requirement #6	Target for Functional Requirement #7	Target for Functional Requirement #8	Target for Functional Requirement #9	Target for Functional Requirement #10	Target for Functional Requirement #11	Target for Functional Requirement #12	Target for Functional Requirement #13	Target for Functional Requirement #14	Target for Functional Requirement #15	Target for Functional Requirement #16			
Max Relationship	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9			
Technical Importance Rating	235.44	375.95	153.16	91.139	121.52	68.354	98.734	68.354	202.53	102.53	193.67	132.91	79.747	79.747	79.747	79.747			
Relative Weight	11%	17%	7%	4%	6%	3%	5%	3%	9%	5%	9%	6%	4%	4%	4%	4%			
Weight Chart																			
Product in question	5	0	4	2	0	3	3	3	4	5	5	5	3	3	0	0			

According to the importance weighting, exporting reports in formats like pdf, excel, creating a secure link for the report to access from anywhere, API and providing enough memory are ranked as the most important requirements.

QFD methods assists in determining the relevance of a product features and comparing product features. In turn, the evaluation of product features aids in the selection of the appropriate version that requires redesign. The technique provided here is beneficial for assisting decisions made during product setup. Product customization necessitates the redesign and manufacture of the product. One of the most essential factors that contribute to the attractiveness of an offer is the amount of time spent on certain activities connected to product development and manufacture. The time it takes to prepare product technical documentation and manufacturing standards might be used to estimate delivery time.

(Kutschchenreiter-Praszkiewicz, I., 2018)

QFD helps to answer to the research question three (RQ3) and conclude that the requirements identified are quite common for the majority of customers and based on that it can be concluded that some sort of core product can be built that would encompass all the mentioned features.

5.3 Strategy selection and an execution framework

It is critical to specify the software functional scope that will be delivered as a SaaS. For SaaS suppliers, having the proper strategy and software architecture to allow configuration and customization is important. This is the foundation for a SaaS service that wants to grow at a low cost. (Sun, W., et al., 2008)

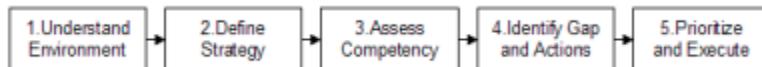


Figure 16 A Framework to Plan and Execute Configuration and customization strategy (Sun, W., et al., 2008)

Diagram depicted in the figure 15 presents a framework to assist the planning and implementation of SaaS configuration and customization strategy. This framework includes a methodology as well as analysis tools.

"Understand Environment": The methodology's first stage is to do the essential research to comprehend the environment around the SaaS service's configuration and customization. Client requirements and market leader expertise levels are the two primary topics that must be evaluated. Analyzing client needs has the goal of identifying the intended customer group as well as the needed variance scope. (Sun, W., et al., 2008)

A customer segmentation study may be carried out by segmenting the entire market into four quadrants, each separated by two key dimensions: uniqueness of requirements and capacity to obtain alternative solutions (see Figure 16). Customers in quadrant III, in general, should be the primary target client group for SaaS services since they have a low degree of variation needs and a limited capacity to buy other solutions (e.g., invest in a custom created application) to the SaaS service. Quadrant I is typically a challenging category for SaaS services to win since each of the clients in this segment has highly specific

needs and the capacity to investigate other options. (Sun, W., et al., 2008)

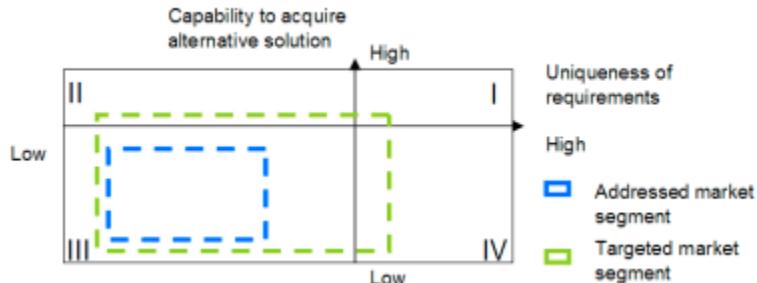


Figure 17 Customer Segment Analysis (Sun, W., 2008)

Customers in quadrants II and IV are often in a marginal position. SaaS suppliers might gain more clients in these two areas if they could provide powerful, simple, and low-cost configuration and customization options. To produce such a customer segment study, the SaaS vendor can use a survey and interviews with chosen potential clients. This analysis has two key components. The first is to clearly define the SaaS application's function scope and complexity level so that as many clients as feasible fit within quadrant III. The second step is to select the targeted market segment and addressed market segment using increased configuration and customization based on the variance needs that have been researched. From a configuration and customization standpoint, a market leader's competency level inquiry may assist a SaaS vendor position itself advantageously in the competition environment. (Sun, W., et al., 2008)

"Define Strategy": SaaS vendors should determine how they will serve the intended client segment's configuration and customizations needs. To make the debate easier, the possible methods for Configuration and Customization are broken down into four models, as shown in the table below. (Sun, W., et al., 2008)

Table 10 Different Configuration and Customization Approaches (Sun, W., et al., 2008)

	Model A: Native	Design Model B: Smooth Evolvement	Model C: Pulse Evolvement	Model D: Failure Management
Description	Thoroughly analyze the common	SaaS providers must make an attempt to meet	A group of customers provides SaaS	Individual customer configuration and

	<p>configuration and customization requirements before building the SaaS application; design the application to allow for extensive configuration and customization; offer powerful web-based tools for customers to configure and customize the SaaS service themselves or through another system integrator vendor.</p>	<p>each customer's configuration and customization needs. However, SaaS suppliers may control costs by utilizing tools and assets and gradually reducing the amount of money spent on configuration and customization for each customer.</p>	<p>suppliers with configuration and modification requests. When the return on investment can be justified by the prospective advantages given by the effort, upgrade the application to meet the standards needed by a large group of renters.</p>	<p>customization are supported by SaaS suppliers. They do not keep costs under control in the way that a thriving firm requires.</p>
Approach	<p>Provide customers with a programming paradigm, web-based tools, and APIs to undertake self-service setup and modification. SaaS suppliers will not alter the</p>	<p>SaaS providers alter application codes to meet the needs of their customers. They use a management tool and a procedure to keep track of how much each</p>	<p>When the configuration and customization needs are stated and justified by a large number of customers, SaaS suppliers modify application codes.</p>	<p>Application codes are changed by SaaS suppliers according to the needs of each customer. They do not have the right tools or processes in place to keep track of the</p>

	application code for a single customer.	renter costs.		money spent on each renter.
Scalability	Very high	Medium-to-low	Medium-to-high	Very low
Application complexity	Medium-to-low	Medium	Medium	High

The four techniques, "Native Design" (Model A), "Smooth Evolvement" (Model B), "Pulse Evolvement" (Model C), and "Failure Management" (Model D), configuration and change have varied degrees of impact on the SaaS service delivery cost paid on each client, as illustrated in Figure 17 (Sun, W., et al., 2008).

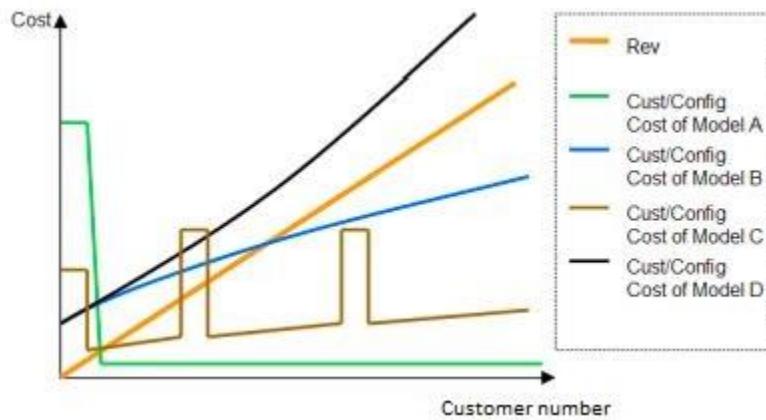


Figure 18 The Impact on Configuration and Customization (Sun, W., 2008)

Model D is definitely a poor one with which no SaaS firm should be engaged. All three of the remaining models may sustain long-term SaaS service firms with varied profit margins. They may be a suitable fit for a certain SaaS business environment, such as application complexity, scalability goals, vendor market knowledge, budget constraints, and so on. Generally, Model B is better suited to SaaS with a small number of clients, as addressing each individual customer's specific requirements is a costly method. Model A is the optimum strategy for SaaS suppliers that wish to explore extremely high scalability to exploit very large economical scale (Long tail play). Model C is the most reasonable approach for a SaaS vendor to take to start with; they learn about the market along the way and can eventually evolve into Model A when they clearly define and build the configuration and customization capabilities required by the large number of customers they want to acquire and serve. Model A can only be developed with a thorough

understanding of the SaaS service's prospective configuration and customization requirements. It uses a specially built software architecture and web-based tools to allow for easy and comprehensive setup and modification without having to change the SaaS application source code. (Sun, W., et al., 2008)

For the company in question the strategy of creating a core product was selected. Features chosen for the core product should be customized based on the requirements of the majority of the industry. The company in question is advised to follow the model A as the company has enough resources to build the application and sell to the customers. The core product is able to attract the customers and in order to retain the customers, customization can be used in case there are many requests for a specific feature. The objective of the proposed model is to provide customers with a programming paradigm, web-based tools, and APIs to undertake self-service setup and modification. SaaS suppliers will not alter the application code for a single customer. Modifications may be done in case there is a big demand from multiple customers. Having the core will also allow the company to attract the investors and get the financing for the further development. What is more, well-made core product can be suitable to many clients and this option may be more profitable for the company itself as implies less investments in the development, unlike personalized solutions which are very costly for the developing company.

Based on the factors that have the biggest impact on gaining and retaining customers in a startup company, it is suggested to choose to propose individually customized solutions to the customers in order to gain the market share and attract clients. Model B is recommended for the startup company as long as they have a small number of clients, because addressing every individual customer's specific needs is a quite costly.

As for the case of an Asset management company, based on the factors that have the biggest impact on gaining and retaining customers and are common for this specific field, it is suggested to use mass customization. Mass customization means creating a clear line of communication with consumers, allowing asset managers to reduce information cycle times, increase market share, and increase customer satisfaction. Model C is the recommended option for the startup Asset management company as it is the easiest approach for a SaaS vendor to start with. Following model C allows to study the market as they proceed and helps to finally evolve into Model A when they clearly identify and build the configuration and customization capabilities needed by the majority of customers they want to gain and whose needs they want to meet.

"Assess Competency": For conventional software application suppliers interested in exploring SaaS as a new delivery model, this stage is critical. For application providers that have successfully addressed the consumer and small medium company (SMB) markets, configuration and customization may not be a significant problem. Typically, these providers follow a volume play strategy and do not cater to specific client variation requirements. They may get a head start on SaaS by converting their apps to multi-tenancy with a Web interface. However, for those application providers that cater to the medium to large corporate market, the problem of setup and customization poses a significant barrier. Despite having a well-packaged program as a basis, these vendors are typically compensated by their clients for using a bespoke development strategy to meet each customer's specific variation requirements. If the program lacks a well-defined configuration framework, source code alterations are frequently used. However, under the conventional application delivery model, the vendor can afford to do so because the end customer pays for it. This method of delivery cannot be duplicated in the SaaS model. The use of SaaS is based on a subscription model. The customer's tiny initial payment and monthly subscription price are insufficient to cover the overall cost of source code level modification. As a result, before deciding to transfer their program to the SaaS distribution model, these application suppliers should exercise caution and do a thorough review of their configuration and customization capabilities. (Sun, W., et al., 2008)

The configuration and customization competency model, as well as several critical views to consider for a SaaS service, are described in the section devoted to the framework. The six viewpoints that may be classified are data structure and processing, organizational structure, user interface, workflow, business rule, and reporting. This approach may be used to evaluate the configuration and modification capabilities of a current software program. As illustrated in Figure 18, a benchmark study may be conducted to compare competency with that of market leaders in order to clearly identify competency improvement objectives. This research does not imply that every SaaS seller should raise their expertise to a higher degree from all angles. Higher-level configuration and customization skills might help a vendor achieve a competitive advantage if their application is functionally equivalent to other current SaaS services on the market. (Sun, W., et al., 2008)

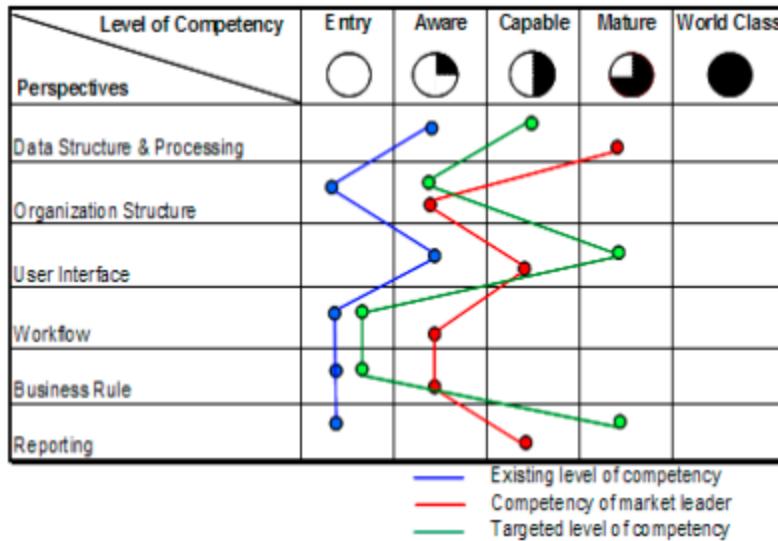


Figure 19 Competency Assessment and Gap Analysis (Sun, W., et al., 2008)

"Identify Gaps and Actions": The competence benchmark study may be used to identify competency gaps, which can then be used to define actions. Figure 18 illustrates the gaps and improvement objectives, particularly in terms of user interface and reporting. The competency enhancement objective may be further expanded into specific activities using the analysis in section 5.3. (Sun, W., et al., 2008)

There is a variety of techniques that may be used to meet a variety of skill level criteria. Parameterized configuration allows the end user to establish pre-defined parameters and choices in the runtime environment to enable "Aware" level variance. Users may build bespoke user interfaces and alter the structure of the data model and the application's business logic using the Self-Serve configuration tool, which uses an application variance metadata framework and a series of easy point-and-click wizards (workflow, business rules, etc). The metadata structure, however, limits the extent of customization. Scripting-based programming, a type of end-user programming, allows for a broader range of customization by the end user by expanding the tool's functionality with a constraint scripting language, which ensures security and prevents script-related harm to the main application. World-class SaaS services integrate their applications with a development environment and a formal programming paradigm that users may use to create new application code or alter existing code to meet their needs. Different SaaS suppliers use different approaches to deployment and build their own. (Sun, W., et al., 2008)

6 CONCLUSIONS

6.1 Main Findings and Implications

Having conducted the research, it became possible to answer to the posed research questions. Customization can be seen as the stage of the company to obtain the customers.

The factors that have the biggest impact on gaining and retaining customers, in both, a startup and an startup asset management company, were identified. Based on this knowledge, a strategy was suggested for each option.

QFD helps to answer to the research question three (RQ3) regarding the selection of the most feasible strategy of the company in question, and conclude that the requirements identified are quite common for the majority of customers and based on that it can be suggested that some sort of core product can be built that would encompass all the mentioned features.

The approach of developing a core product was chosen for the company in question. The core product's features should be modified to meet the needs of the majority of the industry. The firm in question should adopt model A since it has the resources to develop the application and sell it to consumers. Customers are attracted to the core product. The suggested model's goal is to provide consumers with a programming paradigm, web-based tools, and APIs for self-service configuration and change. For a single customer, SaaS providers will not change the application code. If there is a high demand from many consumers, changes may be made.

In order to increase market share and attract clients, it is recommended to select to offer individually personalized solutions to consumers based on the factors that have the greatest influence on acquiring and retaining customers in a new firm. Model B is advised for a starting firm with a small number of clients, as addressing each individual customer's unique demands is extremely expensive.

In the case of an asset management firm, mass customization is recommended based on the factors that have the greatest influence on gaining and retaining clients and are prevalent in this industry. Mass customization entails establishing a direct channel of communication with customers, allowing asset managers to shorten information cycle times, expand market share, and improve customer satisfaction.

Model C is the best option for a new Asset management firm since it is the most straightforward method for a SaaS vendor to begin with. Model C allows them to explore the market as they go, eventually leading to Model A when they can clearly identify and create the configuration and customization capabilities required by the bulk of consumers they want to acquire and whose demands they want to fulfill.

SaaS companies must consider configuration and customization while designing their products. However, the most essential thing is to effectively build the SaaS application functionalities to meet as many customers' needs as possible in the intended customer group and application area, the configuration and competency expertise suggests the major competitive advantages on the market. In many situations, robust configuration and customization capabilities will become the main success element if the SaaS service is difficult to develop as a uniform solution for most customers.

6.2 Limitations and Future Research

Some scarcity of the real numbers and data can be seen as one of the limitations of the research. Another limitation is the unwillingness of the competitive companies to share some data that could facilitate the research in question.

Possible area for research may include a more profound study of mass customization application in the asset management field. To those players willing and able to allocate resources to establishing true customer closeness and move beyond supplying local distribution relationships with global products, mass customized asset management products offer intriguing possibilities for new growth and profitability in the asset management industry. Currently application of mass customization is also studied in various fields and researching those other fields and how usage of mass customization may be useful is another subject for the research.

References

- Adams, R. (2020). Practical customer success management: a best practice framework for rapid generation of customer success. New York, NY: Routledge.
- Adams, R.M. and Gavoor, M.D. (1990) May. Quality function deployment: Its promise and reality. In 1990 ASQC Quality Congress Transactions, San Francisco, USA (pp. 33-38).
- Ak, B. K., Dechow, P.M., Sun, Y. & Wang, A.Y. (2013). The use of financial ratio models to help investors predict and interpret significant corporate events. Australian Journal of Management. 38(3). 553-598.
- Amadi-Echendu J.E. (2004). Managing physical assets is a paradigm shift from maintenance. 2004 IEEE International Engineering Management Conference. pp. 1156–1160.
- Ambler, S.W. (2002). Lessons in agility from Internet-based development. IEEE software, 19(2), pp.66-73.
- Aoyama, M., 1997. Managing the concurrent development of large-scale software systems. Int. J. Technology Management, 14, pp.739-765.
- Aoyama, M. (2005). August. Persona-and-scenario based requirements engineering for software embedded in digital consumer products. In 13th IEEE International Conference on Requirements Engineering (RE'05) (pp. 85-94). IEEE.
- Baines TS, Lightfoot HW, Benedettini O, Kay JM (2008) The servitization of manufacturing: a review of literature and reflection on future challenges. J Manuf Technol Manag 20(5):547–567
- Balcaen, S. & Ooghe, H. (2006). 35 years of studies on business failure: an overview of the classic statistical methodologies and their related problems. The British Accounting Review. 38(1). 63–93.
- Bandara W., Gable G.G. & Rosemann M. (2005). Factors and measures of business process modelling: model building through a multiple case study. European Journal of Information Systems, 14(4), 347–360.

Beck, K. and Andres, C. (2004). Extreme programming explained: Embrace change. 2-nd edition.

Bijl, R. (1992). Delphi in a future scenario study on mental health and mental health care. *Futures*, Vol. S No. 3, pp. 232-50.

Bhuiyan, N., Thomson V. & Gerwin, D., 2006. Implementing concurrent engineering. *Research Technology Management*, 49(1), pp.38-43.

Blackburn, J., Scudder, G. & Wassenhove, L., 2000. Concurrent software development. *Communications of the ACM*, 43(11), pp.200-214.

Blank, S.G. (2005). The four steps to the epiphany: Successful Strategies for Products that Win CafePress. com.

Blank, S. G., 2007. The Four Steps to the Epiphany: Successful Strategies for Products that Win. (Self-published).

Bosch, J. (2000) Design and use of software architectures: Adopting and evolving a product line approach. New York, NY: Pearson Education Addison-Wesley & ACM press.

Bouchereau, V. and Rowland, H. (1998). "An intelligent systems approach to quality function deployment (QFD)", *Transactions from The Tenth Symposium on Quality Function Deployment (QFD)*, Michigan: Novi, 14-16 June, pp. 535-552.

Brown, A., & Wallnau, K. (1998). The current state of CBSE. *IEEE Software*, 1998, 37–46.

Carroll, A.B. & Buchholtz, A. K. (2015). Business & society: ethics, sustainability, and stakeholder management. Ninth edition. Stamford, CT, USA: Cengage Learning.

Cecconi, F.R., Dejaco, M.C., Moretti, N., Mannino, A. and Cadena, J.D.B., 2020. Digital asset management. In *Digital Transformation of the Design, Construction and Management Processes of the Built Environment* (pp. 243-253). Springer, Cham.

Centre for Digital Built Britain (2018) Year one report. Towards a digital built Britain.

Chamberlin, E. H. (1962). The Theory of Monopolistic Competition (Cambridge, MA: Harvard University Press).

Chaney, H.S. (1987). Needs assessment: a Delphi approach. *Journal of Nursing Staff Research*, Vol. 3 No. S, pp. 373-86.

Chau K., Liu S. and Ip W., 2009. Enhancing enterprise information integration using Six Sigma. *Total Quality Management*, 20(5), pp.537-546.

Christensen, C.M. (1997). The Innovator's Dilemma. Harvard Business School Press. Boston, MA.

Clayton, W.J. (1997). Delphi: a technique to harness expert opinion for critical decision-making tasks in education. *Educational Psychology*, Vol. 17 No.4, pp. 373-86

Cohen, L. (1995). Quality function deployment: How to make QFD work for you, USA: Addison Wesley Longman Inc.

Coleman, G. (2005) January. An empirical study of software process in practice. In Proceedings of the 38th Annual Hawaii International Conference on System Sciences (pp. 315c-315c). IEEE.

Coleman, G. and O'Connor, R.V. (2008). An investigation into software development process formation in software start-ups. *Journal of Enterprise Information Management*.

Cooperative Research Centre for Integrated Engineering Asset Management. (2008) What CIEAM do. Model [www document]. [Accessed 5 April 2021]. Available: http://www.cieam.com/aboutus.html#what_is_eam.

Costanzo, C., 2006. Case Studies: Three approaches to 'On-Boarding?'. *American Banker*, 171(117), pp.4-11.

Cox, M. W., and Alm, R. (1998). *The Right Stuff. America's Move to Mass Customization* (Dallas: Federal Reserve Bank of Dallas).

Crnkovic, I. (2001). Component-based software engineering: New challenges in software development. *Software Focus*, 2(4), 127–133.

Crnkovic, I., & Larsson, M. (2002). Challenges of component-based development. *Journal of Systems and Software*, 61, 201–212.

Crowne, M. (2002) August. Why software product startups fail and what to do about it. Evolution of software product development in startup companies. In *IEEE International Engineering Management Conference* (Vol. 1, pp. 338-343). IEEE.

Cusumano, M., MacCormack, A., Kemerer, C.F. and Crandall, B. (2003). Software development worldwide: The state of the practice. *IEEE software*, 20(6), pp.28-34.

Da Silva, A.F., Kon, F. and Torteli, C. (2005). June. Xp south of the equator: An experience implementing xp in brazil. In *International Conference on Extreme Programming and Agile Processes in Software Engineering* (pp. 10-18). Springer, Berlin, Heidelberg.

Davenport T.H. (2005). The coming commoditization of processes. *Harvard Business Review*, 83(6), 100–108.

Davila, A., Foster, G., He, X. & Shimizu, C. (2015). The rise and fall of startups: Creation and destruction of revenue and jobs by young companies. *Australian Journal of Management*. 40(1). 6–35.

Davis, S. (1987) *Future Perfect* (Reading, MA: Addison-Wesley).

Davis, S. (1989). From future perfect: Mass customizing. *Planning Review*, 17(2), 16–21.

Deias, R., Mugheddu, G. and Murru, O. (2002). Introducing XP in a start-up. In *Proc. 3rd International*

Conference on eXtreme Programming and Agile Processes in Software Engineering-XP (pp. 62-65).

Deelstra, S., Sinnema, M., & Bosch, J. (2005). Product derivation in software product families: A case study. *Journal of Systems and Software*, 74(2), 173–194.

Delbecq, A.L., Van de Ven, A.H. and Gustafson, D.H. (1975). *Group Techniques for Program Planning: A Guide to Nominal Group and Delphi Processes*, Scott, Foresman, and Company, Glenview, Illinois.

Denzin, N.K. and Lincoln, Y.S. (1994). *Handbook of Qualitative Research*, Sage, Thousand Oaks, CA.

DeSouza, G., 1992. Designing a customer retention plan. *Journal of Business Strategy*.

Dillibabu, R., Krishnaiah, K. and Baskaran, R., 2011. Development and Application of SQFD Model for an Innovative Embedded Software Product—A Case Study. *International Journal of Industrial Engineering Research and Development (IJIERD)*, 2(1), pp.15-45.

DoorLoop, n.d.. (n.d.) [www document]. [Accessed 5 August 20121]. <https://www.DoorLoop, n.d..com/pricing>

Eisenhardt, K.M. and Brown, S.L. (1998). Time pacing: Competing in markets that won't stand still. *Harvard business review*, 76(2), pp.59-70.

Eringa, K. and Boer, I.L.J. (1998). "Integrating Quality Function Deployment and service blueprinting", Michigan: Novi: Transactions from The Tenth Symposium on Quality Function Deployment (QFD), 14-16 June, pp. 289-304.

Eurostat (2016). Business demography by size class (from 2004 onwards, NACE Rev. 2).

Fan, H., & Poole, M. S. (2006). What is personalization? Perspectives on the design and implementation of personalization in information systems. *Journal of Organizational Computing and Electronic Commerce*, 16(3-4), 179-202.

Felice, P. D. (1998). Why engineering software is not reusable: Empirical data from an experiment. *Advances in Engineering Software*, 29(2), 151–163.

Financesonline, n.d.. (2021). Compare Yardi Voyager vs Building Engines. [www document]. [Accessed 5 August 20121]. <https://comparisons.Financesonline, n.d..com/yardi-voyager-vs-building-engines>

Fink-Hafner, D., Dagen, T., Doušak, M., Novak, M. and Hafner-Fink, M. (2019). Delphi Method: strengths and weaknesses. *Metodoloski Zv*, 16, pp.1-9.

Frisk, E. (2014). Exploring Customer On-boarding in SaaS Startups.

Frolov, V., Ma, L., Sun, Y. and Bandara, W. (2010). Identifying core functions of asset management. In Definitions, concepts and scope of engineering asset management (pp. 19-30). Springer, London.

Galusha, C. (2001). Getting started with IT asset management. *IT Professional*, 3(3), 37–40. doi:10.1109/6294.939973

Garnsey, E., Stam, E. & Heffernan, P. (2006). New firm growth: exploring processes and paths. *Industry and Innovation*. 13(1). 1-20.

Geeksforgeeks. Software Engineering | COCOMO Model [www document]. [Accessed 5 February 20121]. Available: <https://www.geeksforgeeks.org/software-engineering-cocomo-model/>

Getapp. (n.d.). [www document]. [Accessed 5 August 20121]. <https://www.getapp.com/real-estate-property-software/a/DoorLoop, n.d./reviews/>

Giardino, C., Paternoster, N., Unterkalmsteiner, M., Gorschek, T., & Abrahamsson, P. (2016). Software Development in Startup Companies: The Greenfield Startup Model. *IEEE Transactions on Software Engineering*, 42(6), 585–604.

Gilbert, N. (n.d). Pros & Cons of DoorLoop, n.d.: Analysis of a Popular Property Management Software [www document]. [Accessed 5 August 20121]. <https://Financesonline, n.d..com/pros-cons-of-DoorLoop, n.d./>

Graham, P. (2015). Startup = growth.

Greenfield, J. (2007). Mass customizing solutions. *Methods & Tools*, Fall, 15(3), 27–39.

Grimaldi, R. and Grandi, A. (2005). Business incubators and new venture creation: an assessment of incubating models. *Technovation*, 25(2), pp.111-121.

Guo, C.J., Sun, W., Huang, Y., Wang, Z.H. and Gao, B. (2007). A framework for native multi-tenancy application development and management. In The 9th IEEE International Conference on E-Commerce Technology and The 4th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services (CEC-EEE 2007) (pp. 551-558). IEEE.

Haag, S., Raja, M., & Schkade, L. (1996). Quality function deployment usage in software development. *Communications of the ACM*, 39, 41–49.

Hauser, J.R. and Clausing, D. (1996). "The house of quality", *IEEE Engineering Management Review*, Vol.24, No.1, Spring, pp. 24-32.

Heemstra, F. J. (1992). Software cost estimation. *Information and Software Technology*, 34(10), 627–639.

Heineman, G. T., & Councill, W. T. (2001). Component-based software engineering: Putting the pieces together. Boston, MA: AddisonWesley.

Heitlager, I., Helms, R. and Brinkkemper, S. (2007) October. A tentative technique for the study and planning of co-evolution in product. In Third International IEEE Workshop on Software Evolvability 2007 (pp. 42-47). IEEE.

Hilmola, O.P., Helo, P. and Ojala, L. (2003). The value of product development lead time in software startup. *System Dynamics Review*, 19(1), pp.75-82.

Hoch, D. J., Roeding, C. R., Purkert, G., Kindner, S. K., Ralph, M. (2000). Secrets of software success:

Management insights from 100 software firms around the world. Boston

Huffmann, C., and Kahn, B. E. (1998). Variety for sale: mass customization or mass confusion? *Journal of Retailing*, 74(4), 491–513.

Huynh, K.P., Petrunia, R.J. & Voia, M. (2012). Duration of new firms: The role of startup financial conditions, industry and aggregate factors. *Structural Change and Economic Dynamics*. 23. 354– 362.

ISO, T. and SC, N. (2014). Asset management—Overview, principles and terminology.

Jaiswal, E.S., 2012. A case study on quality function deployment (QFD). *Journal of mechanical and civil engineering*, 3(6), pp.27-35.

Jiao, J., & Tseng, M. M. (2000). Understanding product family for mass customization by developing commonality indices. *Journal of Engineering Design*, 11, 225–243.

Jose, A., & Tollenaere, M. (2005). Modular and platform methods for product family design: Literature analysis. *Journal of Intelligent Manufacturing*, 16(3), 371–390.

Kajko-Mattsson, M. and Nikitina, N. (2008). December. From knowing nothing to knowing a little: experiences gained from process improvement in a start-up company. In 2008 International Conference on Computer Science and Software Engineering (Vol. 2, pp. 617-621). IEEE.

Katzengruber, W. & Pförtner, A. (2017). Sales 4.0: Strategien und Konzepte für die Zukunft im Vertrieb. 1. Auflage. Weinheim: Wiley-VCH Verlag GmbH & Co. KGaA.

Khodyakov, D., Grant, S., Denger, B., Kinnett, K., Martin, A., Peay, H., & Coulter, I. (2019). Practical Considerations in Using Online Modified-Delphi Approaches to Engage Patients and Other Stakeholders in Clinical Practice Guideline Development. *The Patient - Patient-Centered Outcomes Research*.

Kilian, D. & Mirski, P. (2016). Digital Selling: erfolgreiche Strategien und Werkzeuge für B2B-Marketing und Vertrieb. Wien: Linde international.

Kindström, D. (2010). Towards a service-based business model – Key aspects for future competitive advantage.' In: European Management Journal, 28 (2010), 6, p. 479–490.

Kingstone, S. (2004). Understanding total cost of ownership of a hosted vs. premises-based crm solution. Yankee Group Report June.

Knorr, E. (2006). Software as a service: The next big thing. InfoWorld, 3, pp.2-4.

Kotonya, G., Sommerville, I., & Hall, S. (2003). Towards a classification model for component-based software engineering. In: Proceedings of the 29th Euromicro Conference (Vol. 43). Washington, DC: IEEE computer society.

Krishnapillai, R., & Zeid, A. (2006). Mapping product design specification for mass customization. Journal of Intelligent Manufacturing, 17(1), 29–43.

Krueger, C. W. (2001). Software mass customization. Austin: Big Lever Software, Inc.

Küng, P., Schillig, B. & Toscano-Ruffilli, R.(2006). Key Account Management: Praxistipps - Beispiele - Werkzeuge. p.40. 3., erw. Aufl. St. Gallen: Midas Management Verl (= Midas Management).

Kuo, T. C. (2012). Mass customization and personalization software development: a case study eco-design product service system. Journal of Intelligent Manufacturing, 24(5), 1019–1031.

Kutschchenreiter-Praszkiewicz, I. (2018). Configuration of a Customized Product. In Product Lifecycle Management-Terminology and Applications. IntechOpen.

Kwong, C. K., & Bai, H. (2002). A fuzzy AHP approach to the determination of importance weights of customer requirements in quality function deployment. Journal of Intelligent Manufacturing, 13(5), 367–377.

Laitinen, E.K. (1992). Prediction of failure of a newly founded firm. Journal of Business Venturing. 11(4).

323- 340.

Laitinen, E.K., 2017. Profitability ratios in the early stages of a startup. *The Journal of Entrepreneurial Finance*, 19(2), pp.1-28.

Lehman, M.M. (1980). Programs, life cycles, and laws of software evolution. Proceedings of the IEEE, 68(9), pp.1060-1076.

Lilja, K.K., Laakso, K. and Palomäki, J., 2011, July. Using the Delphi method. In 2011 Proceedings of PICMET'11: Technology Management in the Energy Smart World (PICMET) (pp. 1-10). IEEE.

Ling, R. and Yen, D., 2001. Customer relationship management: An analysis framework and implementation strategies. *The Journal of Computer Information Systems*, 41(3), pp. 82-97.

Liu, X. F. (2000). Software quality function deployment. *IEEE Potentials*, 19(5), 14–16.

Loo, R. (1996). Managing workplace stress: a Canadian Delphi study among human resource managers. *Work & stress*, Vol. 10 No. S, pp. 183-9.

Loo, R. (1997). The future of management training in Canadian healthcare organizations. *Journal of Management Development*, Vol. 16 No. 9, pp. 680-9.

Loo, R. (2002). The Delphi method: a powerful tool for strategic management. *Policing: An International Journal of Police Strategies & Management*, 25(4), 762–769.

Losbichler, H., Hofer, P., Eisl, C. & Zauner, B. (2012). An Investigation of ROCE and Its Drivers: Empirical Analysis of European Companies. In Jodlbauer, H., Olhager, J. & Schonberger, R.J. (Eds.). *Modelling Value. Series: Contributions to Management Science. Selected Papers of the 1st International Conference on Value Chain Management*. Physica-Verlag GmbH Co. 119-148.

Ma L., Sun Y. & Mathew J. (2007) Asset management processes and their representation. 2nd World

Congress on Engineering Asset Management, Harrogate, UK.

Machiraju, V., Dkkhil, M., Griss, M., & Wurste, K. (2000). E-service, management requirements. Alto, CA: HP Laboratories, Palo.

Marmer, M., Herrmann, B.L., Dogrultan, E., Berman, R., Eesley, C. and Blank, S. (2012). The startup ecosystem report 2012. Startup Genome.

Martino, J.P. (1972). Technological Forecasting for Decision Making, Elsevier Publishing, New York, NY.

Maurya, A., 2010. Running Lean: A systematic process for iterating your web application from Plan A to a plan that works. Model [www document]. [Accessed 5 February 2021]. Available: http://www.runningleanhq.com/runninglean_preview.pdf

Mehta, N. Steinman, D.& Murphy, L.(2016). Customer success: how innovative companies are reducing churn and growing recurring revenue. Hoboken: Wiley.

Merisavo, M., Vesanen, J. and Raulas, M. (2002), Suomalaisten asiantuntijoiden kokemukset ja näkemykset digitaalisten kanavien käytösta markkinoinnissa, LTT, Helsinki.

Milgrom, P., and Roberts, J. (1990). The economics of modern manufacturing: technology, strategy, and organization. *American Economic Review*, 80(6), 511–528.

Moretti N, Dejaco MC, Maltese S, Re Cecconi F (2017) The maintenance paradox. IStEA 2017—Re-shaping. Constr Ind 234–242

Moses, D. & Liao, S.S. (1987). On developing models for failure prediction. *Journal of Commercial Bank Lending*. 69. 27–38.

Murphy, G., Trailer, J., & Hill, R. (1996). Measuring performance in entrepreneurship research. *Journal of Business Research*. 36(1). 15–23.

Needham, R.D. and de Loe, R.C. (1990). The policy Delphi: purpose, structure, and application.

The Canadian Geographer, Vol. 34 No. 2, pp. 133-42.

Nirpaz, G. & Pizarro, F. (2016). Farm don't hunt: the definitive guide to customer success.

Okoli, C., & Pawlowski, S. D. (2004). The Delphi method as a research tool: an example, design considerations and applications. *Information & Management*, 42(1), 15–29.

Osarenkhoe, A. and Bennani, A., 2007. An exploratory study of implementation of customer relationship management strategy. *Business Process Management Journal*, 13(1), pp. 139-164.

Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T. and Abrahamsson, P. (2014). Software development in startup companies: A systematic mapping study. *Information and Software Technology*, 56(10), pp.1200-1218.

Piller, F.T., Moeslein, K. and Stotko, C.M. (2004.) Does mass customization pay? An economic approach to evaluate customer integration. *Production planning & control*, 15(4), pp.435-444.

Rackham, N. (1988). *SPIN Selling*. McGraw-Hill.

Prahalad, C. K., & Ramaswamy, V. (2000). Co-opting customer competence. *Harvard Business Review*, 78(1), 79–87.

Rahmandad, H. & Weiss, D., 2009. Dynamics of concurrent software development. *System Dynamics Review*, 25(3), pp.224-249.

Regnell, B., Höst, M., och Dag, J.N., Beremark, P. and Hjelm, T. (2001). An industrial case study on distributed prioritisation in market-driven requirements engineering for packaged software. *Requirements Engineering*, 6(1), pp.51-62.

Rettig, C., 2007. The trouble with enterprise software. *MIT Sloan Management Review*, 49(1), pp.20-27.

Ries, E. (2011). How today's entrepreneurs use continuous innovation to create radically successful

businesses. The lean startup.

Piller, F. T. (2002). Customer interaction and digitizability: a structural approach. In C. Rautenstrauch et al. (eds) *Moving Towards Mass Customization* (Berlin: Springer), pp. 119–138.

Pine, B. J. (1993). *Mass Customization* (Boston, MA: Harvard Business School Press).

Ramirez, R. (1999). Value co-production: intellectual origins and implications for practice and research. *Strategic Management Journal*, 20(1), 49–67.

Rodríguez, I., Núñez, M., & Rubio, F. (2004). A formal framework for analyzing reusability complexity in component-based systems. *Information and Software Technology*, 46(12), 791–804.

Rogers, G. G., & Bottaci, L. (1997). Modular production systems: A new manufacturing paradigm. *Journal of Intelligent Manufacturing*, 8(2), 147–156.

Rohleder, C., Davis, S. and Günther, H. (2005). Software customization with XML. *Issues in Information Systems VI* (2).

Rohrbaugh, J. (1979). Improving the quality of group judgment: Social judgment analysis and the Delphi technique. *Organizational Behavior and Human Performance*, 24(1), pp.73-92.

Ross, J.W., Beath, C.M. and Goodhue, D.L. (1996). Develop long-term competitiveness through IT assets. *Sloan management review*, 38(1), pp.31-42.

Saasoptics. (n.d.). MRR Definition. [www document]. [Accessed 8 November 2021]. Available: <https://saasoptics.com/saaspedia/monthly-recurring-revenue-defined-for-a-saaS-business/>

Salamzadeh, A., & Kawamorita Kesim, H. (2015). Startup Companies: Life Cycle and Challenges. *SSRN Electronic Journal*. doi:10.2139/ssrn.2628861

Schmidt, R.C. (1997). Managing Delphi surveys using nonparametric statistical techniques. *decision*

Sciences, 28(3), pp.763-774.

Schröder, M. (2019). Solution Selling: Betriebssystem für den Vertrieb von erklärungsbedürftigen Lösungen. München: Haufe Lexware Verlag.

Seaman, C.B., 2008. Qualitative methods. In Guide to advanced empirical software engineering (pp. 35-62). Springer, London.

Selecman, W.H. (1990). June. Quality improvement starts at the beginning with QFD. In Transactions from the Second Symposium on Quality Function Deployment (No. 18-19 June).

Shahin, A., SQFD: Designing a Comprehensive Quality Function Deployment (QFD) for Service Applications.

Shen, X. X., Tan, K. C., & Xie, M. (2001). implementation of quality function deployment based on linguistic data. Journal of Intelligent Manufacturing, 12(1), 65–75.

Sheth, J. (2001). Jagdish Sheth. Competitive Advantages through Customer Satisfaction. [www document]. [Accessed 8 April 2021]. Available: <https://www.jagsheth.com/relationship-marketing/competitive-advantages-through-customer-satisfaction>

Shin, J. S., & Kim, K.-J. (2000). Complexity reduction of a design problem in QFD using decomposition. Journal of Intelligent Manufacturing, 11(4), 339–354.

Siegel, S., 1956. Nonparametric statistics for the behavioral sciences.

Silveira, G. D., Borenstein, D., & Fogliatto, F. S. (2001). Mass customization: Literature review and research directions. International Journal of Production Economics, 72, 1–13.

Simonson, I. (2005), “Determinants of customers’ responses to customized offers: conceptual framework and research propositions”, Journal of Marketing, Vol. 69 No. 1, pp. 32-45.

Skok, D., 2010. SaaS Metrics – A Guide to Measuring and Improving What Matters. For Entrepreneurs [www document]. [Accessed 5 February 2021]. Available: <https://www.forentrepreneurs.com/saas-metrics>

Slashdot. (n.d.). [www document]. [Accessed 5 August 20121].
<https://slashdot.org/software/p/DoorLoop>, n.d./

Smagalla, D. (2004). The truth about software startups: it's not the size of the budget but how it is used that determines success or failure of the enterprise. MIT Sloan Management Review, 45(2), pp.7-8.

Softwareadvice, n.d.. (2019) Yardi Voyager Software. [www document]. [Accessed 5 August 20121].
<https://www.Softwareadvice.com/property/yardi-voyager-profile/>

Sommerville, I. (2007). Software engineering, 8th edn. New York, NY: Adison-Wesley Longman publishing CO.

Stevenson, W. L. (2009). Operations management. Boston: McGrawHill Irwin.

Sun, W., Zhang, X., Guo, C. J., Sun, P., & Su, H. (2008). Software as a Service: Configuration and Customization Perspectives. 2008 IEEE Congress on Services Part II (services-2 2008).

Sun, H., Zhao, Y. and Yau, H.K. (2009). The relationship between quality management and the speed of new product development. The TQM Journal.

Sutton, S.M. (2000). The role of process in software start-up. IEEE software, 17(4), pp.33-39.

Szyperski, C. (2002). Component software: Beyond object-oriented programming, 2nd edn. Boston, MA: Addison-Wesley Professional.

Tegegne, E. W., Seppänen, P. , Ahmad, M. O. (2018). Software Development Methodologies and Practices in Startups. M3S Research Unit, University of Oulu.

Tingling, P. and Saeed, A. (2007). July. Extreme programming in action: a longitudinal case study. In International Conference on Human-Computer Interaction (pp. 242-251). Springer, Berlin, Heidelberg.

Toffler, A. (1970). Future Shock (London: Bantam).

Trustpilot. (n.d.). [www document]. [Accessed 5 August 20121].
<https://www.trustpilot.com/review/DoorLoop>, n.d..com

TrustRadius, n.d.. (2020). Yardi Voyager Reviews. [www document]. [Accessed 5 August 20121].
<https://www.TrustRadius, n.d..com/products/yardi-voyager/reviews?qs=pros-and-cons>

Tseng, M., & Jiao, J. (1996). Design for mass customization. CIRP Annals-Manufacturing Technology, 45(1), 153–156

Tseng, M. M., Jiao, R.-J., & Wang, C. (2010). Design for mass personalization. CIRP Annals-Manufacturing Technology, 59, 175–178.

U.S. Bureau of Labor Statistics (2016). Business Employment Dynamics. Entrepreneurship and the U.S. Economy.

Vesanen, J. (2007). What is personalization? A conceptual framework. European Journal of Marketing, 41(5/6), 409–418. doi:10.1108/03090560710737534

Wager, S. (2005). Digital asset management, media asset management, and content management: From confusion to clarity. Journal of Digital Asset Management, 1(1), 40–45.

Wells, J. (2021). Saxonypartners, n.d.. Yardi Q&A: How do I schedule monthly reports in Yardi? [www document]. [Accessed 5 August 20121]. <https://www.Saxonypartners, n.d..com/yardi-qa-how-do-i-schedule-monthly-reports-in-yardi/>

Wikstrom, S. (1996). Value creation by company–consumer interaction. Journal of Marketing Management, 12(5), 359–374.

Woudenberg, F. (1991). An evaluation of Delphi. *Technological Forecasting and Social Change*, Vol. 40, pp. 131-50.

Yardi. (n.d.). About us. [www document]. [Accessed 5 August 20121]. <https://www.yardi.com/about-us/>

Yoffie, D.B. and Cusumano, M.A. (1999). Building a company on Internet time: Lessons from netscape. *California Management Review*, 41(3), pp.8-28.

Yogendra, S. and Sengupta, S. (2002). August. Aligning business and technology strategies: a comparison of established and start-up business contexts. In *IEEE International Engineering Management Conference* (Vol. 1, pp. 2-7). IEEE.

York, J. (2012). SaaS Sales Models - Strategic and Organizational Choices. [www document]. [Accessed 8 April 2021]. Available: <http://chaotic-flow.com/media/saas-sales-models.pdf>

Zingales, L. (1998). Survival of the fittest or the fattest? Exit and financing in the trucking industry. *Journal of Finance*. 53(3). 905–938.

Zipkin, P. (2001). The limits of mass customization. *Sloan Management Review*, 42(3), 81–87.

Zucchelli, F. (1992). March. Total quality and QFD. In 1st European Conference on QFD, Milan.