CHALMERS
UNIVERSITY OF TECHNOLOGY

Exploring Customer On-boarding in SaaS Startups

Master of Science Thesis
in the Management and Economics of Innovation Programme

ERIK FRISK

Department of Technology Management and Economics
Division of Innovation Engineering and Management
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2014
Report No. E 2014:098

MASTER'S THESIS E 2014:098

Exploring Customer On-boarding in SaaS Startups

ERIK FRISK

Tutor, Chalmers: Henrik Berglund

Department of Technology Management and Economics
Division of Innovation Engineering and Management
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2014

Exploring Customer On-boarding in SaaS Startups
ERIK FRISK

Master’s Thesis E 2014:098

Department of Technology Management and Economics
Division of Innovation Engineering and Management
Chalmers University of Technology
SE-412 96 Göteborg, Sweden
Telephone: + 46 (0)31-772 1000

Chalmers Reproservice

Göteborg, Sweden 2014

Abstract

More and more software startups are moving to a subscription-based Software as a Service (SaaS) model. It is absolutely critical for these companies to make sure that their customers keep using their software over the long term (Skok, 2010), and one very important aspect of this is helping them get up and running with the software quickly and painlessly. This process is known as customer on-boarding and very little research has been done on it, especially in the case of SaaS startup companies.

The purpose of this study is to explore the field of customer on-boarding in SaaS startups. In order to do this I relay my own first-hand experiences from participating in building a customer on-boarding organisation in a SaaS startup in Sweden. The observations from this case are analysed in the context of existing literature from fields related to customer on-boarding. The result is a set of propositions that are presented as a starting point for further exploratory research and empirical testing.

Acknowledgements

I would like to express my sincerest gratitude towards my academic supervisor, Henrik Berglund, for his support. I would also like to thank everyone at Adco (name changed for the sake of confidentiality) for making this thesis possible. I especially want to thank the company's CEO and COO (you know who you are) for allowing me to pursue this project within the company.

Table of Contents

Introduction

The world of startup companies has experienced a dramatic shift in recent years. The vast amount of information and resources made available to entrepreneurs through the internet and other media has made it easier to start a company than ever before. However, it has not necessarily become easier to build a *successful* company (Maurya, 2010).

Another major shift in recent years has happened in the software industry, with many software businesses moving to a Software as a Service (SaaS) model (Bartels, 2013). This signifies a change in underlying technology as well as a fundamental change in terms of how software businesses make money. Companies are now selling subscriptions to their software instead of receiving money up front for each sold copy. Consequently, it has become absolutely critical for software businesses to make sure that their customers keep using the software over the long term (Skok, 2010), and one very important aspect of this is helping them get up and running with the software quickly and painlessly. Failing to do this could mean that customers end their subscriptions long before they pay enough to cover the costs to acquire them.

The process of helping new customers get up and running with a software or service is known as customer on-boarding. The first 90 days after customers have made their purchase is said by many to be the most important part of the entire customer lifecycle for businesses with subscription models (Costanzo, 2006). It is the period in which a new customer will either decide to become a loyal long-term customer, or leave forever. Very little research has been done on customer on-boarding, especially in the case of SaaS startup companies. There is a large amount of literature on related fields—such as sales, product development, general management of SaaS businesses and account management—but very little on how SaaS startups get new customers started with their service in a way that will make customers keep using the service. More knowledge about this phenomenon would help such companies keep more of their customers and become more profitable.

The purpose of this study is to explore the field of customer on-boarding in SaaS startups. The result is a set of propositions on the dynamics within this field that are presented as a starting point for further exploration and empirical research. The study was done using a participant observation methodology, meaning that I as the author had first-hand experience in building a customer on-boarding organisation in a Business to Business (B2B) SaaS startup company. Because the findings are based on a single case the study did not aim to produce generalisable results, but rather to uncover insights that can later be tested empirically.

In this paper, we use Steve Blank's (2007) definition of the term startup. In other words, a startup is "an organisation formed to search for a repeatable and scalable business model". Furthermore, we focus on the phase in a startup's life after it has found the right product for the right customer, and is now shifting its focus towards scaling the company.

The next chapter of this report contains a review of the literature around customer on-boarding in SaaS startups. Due to the fact that there has been very little academic writing on the topic of customer on-boarding itself, the literature review looks at related fields that each shed light on some piece of the customer on-boarding puzzle. This is followed by an explanation of the research method, including a few words on its limitations. Next is the case itself, where I relay my experiences working with on-boarding in a fast-growing SaaS startup company. The report ends with an analysis of the case—uncovering insights and relating them back to the existing literature.

Literature Review

Part of the reason for this paper is the lack of adequate research in the field of customer on-boarding in B2B SaaS startups. A search of several databases through the Chalmers University of Technology *Summons* search engine yields not a single directly related result. There is tangentially related literature however, which will be outlined below in order to give an idea of the research landscape surrounding the topic of this paper.

As we are exploring customer on-boarding within a startup context, it is important to study literature on the unique challenges faced when managing startup companies. Moreover, we know that customer on-boarding by definition deals both with customer relationships and the practical aspects surrounding the company's product. Therefore we need to study a range of related topics that lie on different points on the continuum from pure product development to pure sales. This gives us a good base of information to which we can relate the findings from the case. Following is a brief outline of the chosen topics, along with the reasons why they were included in this review.

- **Startup management:** How startups should be managed overall. Important in order to understand the unique managerial challenges associated with startups.
- **Sales and account management:** A well studied area that is closely tied to customer on-boarding. The customer on-boarding process begins where the sales process ends and it is reasonable to expect links between the two.
- **Customer relationship management:** Although not traditionally focused on the technical delivery of a product or service, this field deals with how a company should manage its existing customers.
- **Customer on-boarding and delivery:** A look at how customer on-boarding (or similar activities) is handled in other industries and other types of companies.
- **Enterprise software integration:** A look at managing projects where enterprise software is deployed internally in larger firms. Although it does not deal with the specifics of conducting such projects across a supplier-customer relationship, it help to understand the overall dynamics in such projects.

- **Concurrent software development:** Due to the need for rapid iteration and continuous product development in startups, the on-boarding process will not be completely decoupled from product development. Thus it is important to look at the state of the art in software development management.
- **Software testing and quality assurance:** Important to understand in order to ensure quality in the software product being delivered.

Interesting to note is that there is quite a bit of respected practitioner literature on some of these topics. SaaS startup management is particularly well documented in various books, e-books and blogs. Some of these are unorthodox sources to cite in an academic paper, but as the purpose of this thesis is first and foremost to explore a field relatively untouched in academia they are highly relevant.

Startup management

The topic of startup management is very broad and there is a slew of writing in the field, both academic and from practitioners. The writing most closely related to the topic of this paper mainly comes from practitioners. One very relevant subset of the writing on startup management is the field of Customer Development and the Lean Startup Movement. These frameworks for startup management stress the importance of systemising learning processes in order to help the organisation realise early on if it is following the right path or if it needs to make changes to the product or business model (Maurya, 2010). Blank (2007) goes into detail on many vital points for making a startup company succeed, of which some of the most important are:

- **Getting out of the building:** Many startup company founders focus on building their vision without adequate validation from customers. It is crucial to "get out of the building" and talk to prospective customers from day one in order to begin validating and disproving assumptions about the market before it is too late.
- **Using *Minimum Viable Products* to test hypotheses:** Rather than building a full-featured product and attempting to sell it as a linear process, startups should build a product that is as simple as possible while still holding some value. This product, known as a *Minimum Viable Product*, should be put in the hands of customers in order to test whether its value is real or simply imagined by the startup founders and/or

managers. New Minimum Viable Products should be built in order to test new hypotheses based on the aforementioned learning.

- **Knowing your company's current growth phase:** There are four growth phases that a startup company goes through; (1) Customer Discovery, when you are experimenting with customers and minimum viable products in order to find a product that will hold value for some customer segment, (2) Customer Validation, when you begin selling an early version of your product, (3) Customer Creation, where you launch the product at scale, and (4) Company Building, where you rapidly grow the company and try to make the product mainstream.

There is other practitioner literature more directly related to the management of SaaS startups in particular. For example, one blog post from a reputable source (Skok, 2010) outlines the most important KPIs and business dynamics for building a profitable SaaS business. Two guidelines for building a working SaaS business are that the Customer Lifetime Value (LTV) should be greater than three times the Customer Acquisition Cost (CAC), and that it should take no more than 12 months to recover the CAC for an average customer. The formulas for calculating these values (Skok, 2010) are:

- **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.

Relating this literature to customer on-boarding, it would seem there is much to consider when building a customer on-boarding process. Not only does it need to play a part in systemising learning from customers, but it also needs to work to increase the LTV of a customer by maximising the customer's average lifetime and controlling the costs to serve it. One may also find that the goals of the on-boarding process change with the company's current growth stage as outlined by Blank (2007).

Sales and account management

One field that would seem to have a natural connection to customer on-boarding is that of sales. However, the literature on this topic tends to focus solely on activities leading up to the closing of a deal, rather than the process following it. For example, the popular SPIN Selling methodology (Rackham, 1988) focuses on how the sales representative should ask specific questions in order to convince a prospect to sign. SPIN Selling classifies questions asked by the salesperson into four categories, namely (1) Situation Questions, which deal with the facts of the prospect's current situation, (2) Problem Questions, which deal with the prospect's current perceived pain, (3) Implication Questions, which ask about the effects of the aforementioned problems and pains, and finally (4) Need-Payoff Questions, in which the prospect is encouraged to elaborate on their needs and how the product or service being sold might help them. Research found that salespersons who asked more Situation Questions were less successful at closing deals, whereas more successful salespersons tended to ask more Problem Questions and hold off on proposing solutions until very late in the discussion.

Another widespread concept in sales is the "seven steps" (Moncrief and Marshall, 2005) that a sales rep must take the prospective customer though; namely (1) prospecting, (2) preapproach, (3) approach, (4) presentation, (5) overcoming objections, (6) close and (7) follow-up. Only the last of the seven steps, the follow-up, occurs after the closing of the sale and is typically described as no more than making a phone call or sending an email after closing a deal to check with the new customer to see if he or she is happy.

Key Account Management (KAM) is another field in which one might expect to find information related to customer on-boarding. The literature on this topic tends to focus on things such as selecting key accounts, analysing profitability of key accounts, building relationships with customers and obtaining key supplier status. (Woodburn and McDonald, 2011). For example, Cheverton (2012) introduces a six-step process for classifying customers and defining actions that go along with each customer group. The six steps are (1) knowing your objectives for KAM, (2) market segmentation, (3) assembling the classification and selection team, (4) classifying your customers (including identifying the customers' attractiveness and assessing your supplier attractiveness),

(5) applying customer distinction strategies and (6) communicating, aligning and implementing the KAM strategy. More specifically, one should select only a handful of Key Accounts, and they should be aligned with your corporate strategy (Woodburn and McDonald, 2011). Customers that are not aligned with the corporate strategy should be left out of the KAM initiative so as not to dilute it and thus threaten its success.

Literature on KAM identifies different types of customer relationships that each come with their benefits and disadvantages. Woodburn and McDonald (2011) list five such categories:

- **Exploratory relationships:** The supplier and customer are still exploring how they will cooperate. It is up to the supplier to identify potential Key Accounts early and treat them as such. It can take a long time to move into a more productive type of relationship. The supplier must monitor each touchpoint with the customer carefully.
- **Basic relationships:** The supplier's Key Account Manager has built a relationship with a Key Contact in the customer organisation, and simple transactions are carried out. This relationship is typically efficient, but vulnerable to competition and change.
- **Cooperative relationships:** A wider range of people in the supplier and customer organisations are interacting. Clear lines of communication between the organisations have not been established. It is less vulnerable to change than a basic relationship, but is hard to control and not likely to be profitable. It should be regarded as a transitional stage to a more integrated type of relationship.
- **Interdependent relationships:** Close relationship based on structured cooperation between many business functions in the supplier and customer organisations. This type of relationship requires trust leading to relatively free exchange of information, and a deep mutual understanding which allows joint strategic planning. This can in turn lead to substantial business growth for both parties.
- **Integrated relationships:** Boundaries between the two organisations begin to dissolve and they are just short of a merger. This type of relationship is difficult to build and can repel other customers in the same marketplace.

Whereas sales literature understandably tends to focus solely on activities leading up to the sale itself, literature on KAM does shed some light on how to develop the customer relationship after the sale. While it does not offer details as to how to set up the operations of a customer on-boarding process, it does suggest topics which need to be considered in so doing. More specifically, the customer's key account status (or lack thereof) and the depth of the relationship that the supplier has with the customer may decide what activities should be carried out during the on-boarding stage.

Customer relationship management

Customer Relationship Management (CRM) has been thoroughly studied since its emergence in the mid-1990s. It is not a new idea, but has been made practical due to advances made in enterprise software over the past couple of decades (Chen and Popovich, 2003). An effective CRM program can increase customer value, customer retention, customer recruitment and overall profitability (Ling and Yen, 2001). Modern literature on the subject covers various aspects. For example, there is an understanding that CRM should not only be about setting up IT systems for managing customer information. According to Osarenkhoe and Bennani (2007), a CRM strategy should instead be thought of as having a business process orientation that "helps to bring together a variety of information about customers, sales, marketing effectiveness, responsiveness and market trends". A company may accomplish this by managing four key factors (Osarenkhoe and Bennani, 2007):

- **Interaction:** The specific activities that take place in every interaction between the company and its customers.
- **Contact:** Mapping out each point of contact between the company and its customers.
- **Knowledge:** Collecting and analysing information on customers for continuous learning.
- **Relating:** Creating relevant interactions with customers, leading to valuable relationships.

Besides managing the above points, best practices when it comes to implementing a CRM strategy include conducting the implementation in iterative phases to make it manageable, including end users of the CRM system early on to avoid surprises, as well as gaining executive sponsorship to achieve cross functional agreement (Ling and Yen, 2001). It is also important to include training and empowerment of employees in the CRM strategy, as well as the development of automated processes that help customers in various ways (Ling and Yen, 2001).

Literature on CRM tends to focus on two general areas, namely (1) the technical workings and implementation of CRM IT solutions in large organisations and (2) the organisational structure that needs to be put in place around CRM IT solutions to use them effectively. Knowledge in the first area is of limited use in this paper, largely because the rapid development of IT technology in the past few years has rendered much of it obsolete, but also because a growing startup company does not face the same challenges in implementing a new IT solution that a large, already established organisation does. The second area is more applicable. The main lessons learned are to manage all customer touch-points and the activities that take place during said touch-points in a deliberate manner, as well as to collect data on individual customers in a structured way so that it may be used for building meaningful relationships with them.

Customer on-boarding and delivery

As mentioned, surprisingly little literature can be found on the topic of customer on-boarding. What little writing there is mainly comes from the financial services industry. Banks have realised that the first few months of a customer's interaction with the firm is a make-or-break period, in which the customer will either decide to stay with the bank for a long period of time, or move to a competitor. More specifically, the majority of cross sell and up-sell opportunities in the financial sector are said to arise within the first 90 days of a customer opening a new account (Costanzo, 2006). Literature on the subject stresses the importance of putting in place an all-encompassing customer on-boarding process across different communication channels to create a seamless experience for the customer, as well as using the interaction to attempt cross sell and up-sell (Pitney Bowes, 2010). However, little detail is given on how to set up such a process effectively.

Outside of the financial services industry, the literature mainly focuses on the importance of aiding customers in building skills and competencies necessary for gaining maximum value from the product being sold. In this literature the customer on-boarding plan is referred to as the *service delivery strategy*, and the necessary skills are built by providing customer services along with the product in a way that fits with the overall product design strategy of the firm (Kumar and Kumar, 2004). It also stresses the need to continuously evaluate the service delivery strategy so as to iterate and improve it over time. This is especially important in B2B companies as users of industrial products (as opposed to consumers) are "more likely to use the product continuously, or at close to maximum capability" (Kumar and Kumar, 2004), and are therefore typically more demanding when it comes to services surrounding the product. More recently, a framework consisting of a set of operational considerations that are important when designing a service delivery strategy has been proposed (Partha and Roy, 2011). First of all, the framework talks about the degree to which the customer participates in the service delivery process. Some researchers believe that the customer should be encouraged to participate, as this enhances two-way communication, whereas some simply believe that the service delivery strategy should be flexible enough to handle varying degrees of customer participation. Secondly, it stresses the need for successful information flow, both from customer to supplier in terms of expectations and critical operational details and from supplier to customer in terms of the workings of the delivered product. Thirdly, it stipulates the need for incentives put on the supplier to ensure compliance with performance measures. Fourthly, it talks about organisational readiness, stating the importance of creating a culture and a service network that is flexible and agile so as to cope with varying customer demands. Fifth and finally, the framework talks about performance assessment. It suggests that a comprehensive output specification is necessary for the success of a service delivery strategy. However, it also mentions that it can be difficult to create such a specification in a co-production situation because it can be difficult to measure the input from the supplier and the client.

The literature on customer on-boarding tends to be quite abstract. The research within the financial services industry generally states that the first few months with a customer is the most important, but is does not give much tactical advice as to how to take advantage of this time period in the best way, especially when operating outside of financial services. The literature on service delivery strategy is slightly more comprehensive, but still does not give much tactical advice. This may be because the research has been done in the context of large organisations where pure strategy can be of more use than in small startups, where tactics and operations tend to be more critical. However, the service delivery framework put forth by Partha and Roy may very well prove to be applicable to startup companies.

Enterprise software integration

Integrating new enterprise software in an organisation is recognised as something notoriously difficult. The phenomenon has been thoroughly covered in academic literature, perhaps largely due to its complexity. One aspect that the literature tends to focus on is best-practices in managing these large, complex software integration projects. For example, Lam and Shankararaman (2004) introduce a methodology which suggests the project be split into five phases; (1) understand the end-to-end business process, (2) map the process onto components, (3) derive the requirements, (4) produce the architecture, and (5) plan the integration. Another paper emphasises the value of employing basic Six Sigma tools in such projects (Chau, Liu and Ip, 2009). More specifically, it suggests that the DMAIC methodology (define, measure, analyse, improve and control) be used for structured problem solving in organisations in order to pinpoint what areas within the business should be improved using enterprise software. This area of the literature appears very fleshed out, but there does not seem to be a general consensus in the field as to what the best-practices are. There is an emphasis on the importance of top-management support and consideration of the business- and/or people-aspect of the integration as well as the technical aspect, but these considerations could be viewed as general best-practices when implementing any new business initiative.

Another aspect that the literature discusses is why enterprise software integration projects so often fail. In fact, 75% of ERP software implementations are considered failures (Rettig, 2007). Rettig (2007) argues that this is greatly due to the fact that managers consider ERP software to have limitless possibilities when it comes to business benefits. They see it as a silver bullet and tend to grossly underestimate the cost and complexity involved in making an ERP software implementation work. This complexity also makes it very difficult and costly to upgrade old systems, leading to companies having to maintain a number of legacy systems that make it difficult to use new technology that does not interface well with the old.

In summary, it is clear that enterprise software integration is difficult at best. Researchers have attempted to find out exactly why this is, and to create models for how to make enterprise software integration projects successful. Unfortunately, there does not seem to be much consensus around the latter. What researchers seem to agree on is that these projects require a lot of care and consideration in order to stand a chance of success. They should not be taken lightly.

Concurrent software development

Concurrent engineering as a whole is a wide topic that has been thoroughly covered in literature. For the purposes of this paper, we will focus on concurrent engineering as specifically applied to software development. Concurrent software development is characterised by a high level of overlap between different activities in the software development process. This is in contrast to more traditional development strategies where different activities are done in sequence with little or no overlap. Besides a high degree of overlap, concurrent projects are associated with multifunctional teams and/or a high level of two-way communication between functions (Bhuiyan, Thomson & Gerwin, 2006).

Concurrent software development can be more challenging to manage than sequential development, but the benefits are also considerable. Projects using concurrent engineering are more successful in terms of time to market, project development costs and product quality (Bhuiyan, Thomson & Gerwin, 2006) (Aoyama, 1997). They also allow for incremental delivery, which means that customer feedback can be collected to inform the development process at an earlier stage (Aoyama, 1997).

These benefits can be attributed largely to an increase in quality in the early stages of the development process. By engaging all relevant functions for the development process at the project outset risks and tradeoffs can be identified with more precision, specifications will be more correct and potential production issues can be resolved early (Bhuiyan, Thomson & Gerwin, 2006). Blackburn, Scudder & Wassenhove (2000) found that investing more time and effort in the early stages of a software project led to faster overall cycle times and higher productivity due to less need for re-work in the later stages. The top cause of time delays in software development projects is redesign, recoding and retesting stemming from changes in requirements.

If the early stages of the project, such as concept design, are downplayed in favour of product design and testing, a temporary shortage of resources can lead to a sustained decline in project performance. Poor concept design leads to rework in product design, which leaves fewer resources for concept design in the next iteration or project, causing a downward spiral in performance (Rahmandad & Weiss, 2009). Thus it becomes extremely important to avoid sacrificing quality and capability-building activities under times of high pressure from a management perspective (Rahmandad & Weiss, 2009). Not heeding this advice can lead the organisation into a “firefighting” dynamic, where much of the resources are spent reacting to critical issues rather than developing for the future.

In summary, concurrent software development can be challenging from a managerial perspective, but can lead to shorter cycle times, higher overall productivity and higher quality in projects. This is largely because the multifunctional and overlapping nature of these projects allow for better results in the early stages, such as requirements and concept design, which in turn minimises the need for rework in the later stages.

Software testing and quality assurance

The field of software testing as an engineering practice is very immature despite the ever-growing role of software in modern society. In fact, software testing is currently more of an art than a science, and practitioners are still using the same methods as they did 20-30 years ago. These methods are generally neither very sophisticated nor efficient (Juristo, Moreno & Strigel, 2006). The thing which makes software testing so difficult is the complexity of software, coupled with our limited understanding of its principles (Choudhary & Kumar, 2011). Testing cannot prove that a system is free of defects, it can only aim to reveal specific defects that appear as a result of a given set of inputs (Nirpal & Kale, 2011). Even so, it remains a crucial tool for maximising the quality and reliability of software (Juristo, Moreno & Strigel, 2006). Good testers are able to rely on their creativity, experience and intuition to find defects in a piece of software before it ships (Choudhary & Kumar, 2011).

An important distinction that can be made between different software testing methodologies is that of static analysis and dynamic testing (Nirpal & Kale, 2011). In static analysis, a code reviewer reads the source code line-by-line in an attempt to identify potential problem areas. The quality of the analysis depends heavily on the experience and intuition of the reviewer. Dynamic testing, on the other hand, is the practice of executing the code with a given set of input data, and observing the results. The term “software testing” usually refers to just dynamic testing. The difficulty in dynamic testing comes from the fact that the possible set of input data is often near infinite. The problem becomes even more complex if the system needs to accept uncontrolled inputs from the real world, including human interaction and unpredictable environmental factors (Choudhary & Kumar, 2011).

The most common solution to the aforementioned problem with dynamic testing is to select a set of test cases that attempt to represent the complete set of potential inputs as best as possible. A tester probes the system based on her experience in an attempt to find defects, which means that testing is a potentially endless activity. The deciding factor for when to stop testing can be when the system’s estimated reliability meets requirements, but more commonly it is decided by when time, budget or the agreed-upon set of test cases is exhausted (Choudhary & Kumar, 2011).

Another distinction which can be made between different types of testing is that of unit testing vs system testing (Nirpal & Kale, 2011). Unit testing is the practice of testing small pieces of the code to ensure that they perform their intended individual functions. System testing, on the other hand, is the practice of testing the entire system as a whole, which is important in making sure that everything works as intended once all parts have been assembled.

Research on formally “proving” the correctness of code has been conducted in an attempt to overcome some of the limitations with software testing. Although promising, this approach is only feasible for very simple programs (Choudhary & Kumar, 2011). Furthermore, some have suggested that testing software may not be the most effective way of ensuring correctness at all. It may be better to improve the design process than to improve the testing process (Choudhary & Kumar, 2011).

Generally speaking, the literature on software testing highlights the limitations of the practice as it is done today. Testing is currently more of an art than an engineering practice, and the complexity of software means that there are no simple solution to this. Despite the limitations, many agree that software testing is critical in minimising the number of defects in software, but some believe that this energy is better spent in improving the design process, although there is no clear consensus as to how it should be improved.

Summary and Synthesis

The purpose of this literature review is to see what may be inferred about customer on-boarding in B2B SaaS startups from literature on tangentially related topics. As mentioned in the beginning of this chapter, the chosen topics can be organised on a continuum from technical product development activities to pure sales activities. This is important since customer on-boarding effectually bridges the gap between sales and the practical aspects of how the product is used. The topic of general startup management does not fall on this continuum, but is nevertheless important in order to understand the overall context. A representation of the continuum and where the topics fall on it can be found in Figure 1 below.

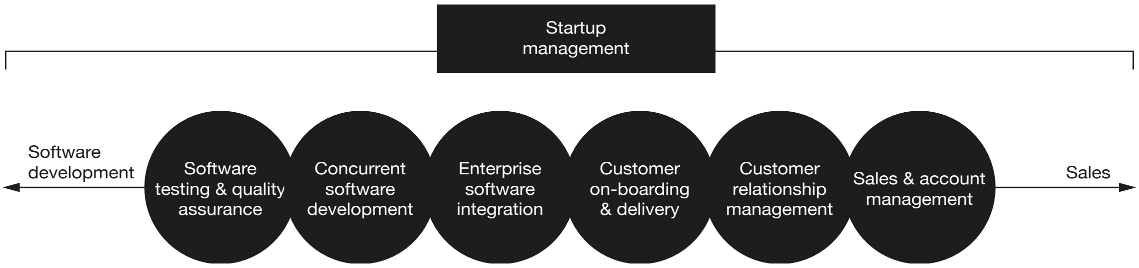
Startup management
Software development
Software testing & quality assurance
Concurrent software development
Enterprise software integration
Customer on-boarding & delivery
Customer relationship management
Sales & account management
Sales

Fig 1 — Theoretical framework illustrating the literature topics' relation to each other

The startup context

The literature on startup management stresses that it is crucial for a startup to maximise the pace at which it learns from, and adapts to, customer behaviour (Blank, 2007). Since the customer on-boarding process is the first point at which the product being sold is truly put to the test with a given customer, we can expect that customer on-boarding should be an integral part of the startup's overall learning process. It is likely that the startup will benefit from having a customer on-boarding process that is designed to collect learnings, and even actively run experiments, that are communicated to the management and product development teams. At the same time, the literature on metrics in SaaS businesses (Skok, 2010) suggests that the customer on-boarding team in a B2B SaaS startup should strive to minimise customer acquisition costs and maximise customer lifetime value, as these metrics are pivotal for a SaaS business's profitability.

Furthermore, the context will likely change depending on the startup company's current growth phase (Blank 2007). As the company transitions through the four phases (Customer Discovery, Customer Validation, Customer Creation and Company Building) it will shift from focusing on experimentation to focusing on growth. This will undoubtedly affect the goals of the customer on-boarding team in a similar way.

Software development in customer on-boarding

Looking at the topics to the left on the continuum in Figure 1, there are several things we can infer about customer on-boarding from the literature. First of all, the literature on Enterprise Software Integration, stresses the fact that large scale software integrations are often underestimated in terms of complexity and cost. Put plainly, they tend to get messy. Combining this with the uncertainty and high pace that startup companies operate in suggests that avoiding mishaps in customer on-boarding is impossible.

In terms of reducing the number of incidents, the literature on software testing and quality assurance indicates that the practice of testing software is crucial for minimising the number of defects in the code itself. However, it also warns that software testing is far from perfect. Software testing can be made more effective though tweaking the process, but in the end it is largely based on intuition and gives no guarantees. This suggests a need to balance the resources allocated to testing software and the resources allocated to fixing software problems experienced by customers. It also suggests that it is impossible, or at least extremely costly, to guarantee problem-free software to customers, and thus the customer on-boarding team needs a structured and deliberate process for dealing with problems experienced by customers. This is especially true if the software is updated and changed continuously, which may often be the case in a startup company.

The literature on concurrent engineering mentions a negative spiral that can be initiated by poor concept design in software development. Poor concept design leads to re-work, which leads to fewer resources available for the next concept design phase, leading to even more re-work, and so on. Thus, the quality of the learnings collected from customer on-boarding is very important for the organisation as a whole. Failure in collecting and communicating relevant learnings could contribute to re-work in the development process, and potentially tip the organisation into a firefighting dynamic. Furthermore, the complexity of large software integration projects mentioned previously suggests a need for the customer on-boarding team to work actively with the product development team to design the product in a way that minimises the technical complexity of the integration.

Customer relationships in customer on-boarding

Looking at the right side of the continuum in Figure 1, the literature on customer on-boarding in the financial services industry talks about the importance of making the first 90 days of a customer's experience with the product flawless. This requires a certain degree of stability and control that does not come naturally to a startup operating in an environment of high uncertainty, which suggests a need to actively work to make the customer on-boarding process more efficient and manageable. In terms of how to attain said efficiency, the literature on Key Account Management suggests that an important factor to consider when managing the customer on-boarding process may be which customers to focus more time and resources on, rather than just assuming that all customers should be given their equal share. It may be beneficial to assess which customers are most aligned to the corporate strategy and focus resources on them. The literature on CRM also highlights the importance of setting up automated processes and empowering employees in dealing with customers. Another important dynamic to consider is Blank's startup growth phases. As the company transitions into the later growth phases, i.e. Customer Creation and Company Building, the need for efficiency in customer on-boarding will likely increase.

The literature on Customer Relationship Management stresses the need to deliberately map out and manage all touch-points with customers. This should be done for the entire customer lifecycle, including the customer on-boarding process. Each touch-point should be designed to strengthen the relationship between the company and the customer. Furthermore, the literature on service delivery strategy suggests that it is necessary to carefully design the on-boarding process to help customers build the capabilities necessary to fully utilise the product or service being sold. Success factors include

(1) encouraging the customer to actively participate in the delivery process, (2) facilitating two-way information flow, (3) incentivising the supplier to comply with performance goals, (4) creating organisational readiness on the supplier side (meaning the supplier can quickly adapt to changing customer demands), and (5) continuously assessing and improving the customer on-boarding process.

The role of this framework

The literature on the topics mentioned above does shed some light on the field of customer on-boarding. There are no major contradictions in the learnings, but there is a question as to which of the points mentioned above are most important for building a successful customer on-boarding process. There may also be dynamics and considerations on the topic of customer on-boarding in B2B SaaS startups that can not be inferred from tangentially related literature. In either case, the general framework illustrated in Figure 1 can be used as a point of reference when analysing the real case study for insights, as will be shown in the coming chapters.

Method

The lack of adequate scientific writing on the subject of customer on-boarding in B2B SaaS startups suggests that there is a need for exploratory research in the field. The aim of this research is to produce more clearly defined problems and propositions for future study. Due to the exploratory nature of this research, the method is based on single-case, overt participant observation. This methodology is often used in fields such as cultural and social anthropology, where it is sometimes not enough to be a passive observer. Instead one must find an active role in the phenomenon being studied. Rather than being concerned with ”knowledge for understanding”, participatory research often seeks “knowledge for action” (Clark et al, 2009) which makes it suitable for this study. The focus is on uncovering insights with practical implications as well as insights for theoretical understanding.

The case to study was chosen largely by convenience. I had already been offered a role in building the customer on-boarding process in a certain B2B SaaS startup company, which meant extensive and rare access to data.

Data gathering

All data stems from my own direct participation in developing a customer on-boarding department at a venture-capital-backed B2B SaaS startup company from June 2011 up until May 2012. I kept a daily journal where I recorded the most important events of each day. The aim of the journal was to act as a form of scaffold, allowing me to remember the exact dates on which certain events happened so that I would later be able to puzzle together various important chains of events. I typically focused on noting factual events rather than feelings and opinions, although these were also noted when they appeared relevant. A large amount of data was also generated in the form of email communications, meeting notes, calendar records and documents created as part of the day-to-day work.

Data presentation

Due to the exploratory nature of the research question and the complexity of the data, the results are presented in the form of a full-bodied case study text. At the end of the data gathering period, the aforementioned daily journal was broken into short sentences or paragraphs describing small individual events. These events were then grouped and organised chronologically into parallel chains of events, or *sub-plots*, that occurred throughout the development of the customer on-boarding department at Adco. Each sub-plot's timeline was supplemented by data from email communications, meeting notes, calendar records and other working documents. Each of these sub-plots were then fleshed out and organised into a full-bodied text, documenting the case.

Data analysis

Once the case text had been produced, it was analysed in order to identify insights concepts and dynamics suitable for further scientific study. While processing the case, three overarching themes became prevalent with regard to the development of the customer on-boarding department at Adco. These themes were used as a basic structure for the data analysis. Concrete examples of events within each theme were identified and related back to the theory in accordance with the framework in Figure 1. Examples tying the themes to both the software development side and the customer relationship side of the framework were highlighted. Whenever evidence from the case study and evidence from theory produced an interesting general insight, a formal proposition was formulated that can be used to inform future research.

Validity, reliability and generalisability of the method

As previously mentioned, the method was not primarily designed to produce statistically generalisable results regarding customer on-boarding in B2B SaaS startups. The participant observation methodology is often used to illustrate propositions rather than prove them (Bell 1969). In other words, participant observation as a method allows you to uncover examples of a certain phenomenon, but may not allow you to make accurate generalisations. This is acceptable for the purposes of this study as the aim is to explore a novel phenomenon. The method was chosen to produce rich, qualitative results from a complex and unpredictable situation.

It is also important to address the risk of personal bias. According to Bell (1969), participant observation introduces a risk of bias towards verification if the author approaches the research with some degree of prejudice. This risk was mitigated by keeping the case text primarily focused on the events that transpired, rather than my own reflections in the field. It is important to note that the case text is based primarily on a daily journal rather than memory. The purpose of the journal was to record factual dates and events, which helped me analyse the case with some degree of separation and objectivity, even though there is inevitably a degree of subjectivity to the findings. Again, this is acceptable as long as the aim of the study is to explore rather than make reliable predictions. According to Flyvbjerg (2006), a full fledged narrative with complexities and contradictions does not necessarily detract from the case's validity. Instead, it can be a sign that the case study has uncovered an especially rich and undigested area of study.

Results

This section includes a detailed telling of the events that occurred throughout the project, from June 2011 until May 2012. The case has been structured into a collection of chapters that each convey a certain story or theme. The case is centered around the evolution of the customer on-boarding process at a B2B SaaS startup company. In order to provide necessary context, the case also touches upon related areas such as the company's product development and sales activities. The stories in each of the chapters are told chronologically in their own right, but there is significant overlap in time between the different chapters. Many of the stories told take place over the course of several months, making it more practical to organise the entirety of the case into overlapping stories and themes rather than one holistic chronological telling.

Insights will later be identified from the case study, as described in the section on Method above. Names of all people and organisations have been changed for the sake of confidentiality.

Preamble

Adco was founded in 2009 on the premise that the online advertising industry needed to be disrupted. Most online advertising campaigns did not take advantage of the endless possibilities that the digital format provided. Instead, ad agencies and advertisers simply took the same approach to the online space as they did to traditional media. Adco wanted to build tools that made it easy for ad agencies to take advantage of the power of digital media, and so they built their first product WriterTool.

WriterTool was a text editor that allowed copywriters to write dynamic text for online advertising messages. The text would change depending on the user viewing it, so that it would target that specific person in a way that was not possible before. The product received a lot of buzz in the industry, but when it came down to it there were very few buyers. It turned out that ad agencies found it difficult to build ads that incorporated this dynamic text. The technical platform simply was not in place.

Adco decided to pivot. They began building the technical platform necessary for ad agencies to build ads that would change dynamically depending on who viewed it. Again the product received buzz, and a few ad campaigns built using Adco's technology won prestigious awards, but the platform did not take off. Adco found that the majority of ad agencies thought that building dynamic ads was more trouble than it was worth. In order to prove that dynamic ads were much better than traditional static ads, Adco asked ad agencies to measure the performance of dynamic ad campaigns versus static ones. They found, however, that the ad agencies had no good way of measuring said performance. Here was another opportunity, and so in 2010 Adco decided to pivot again...

1. The origins of *Studio*

In June of 2011, Adco was in the midst of its first major growth phase. The company had recently closed a venture capital round of 20 million Swedish kronor (approximately 3 million US dollars). They had grown from seven employees in the beginning of the year to 15 in the beginning of June, and there was no sign of slowing down. Over the same period of time the company had worked on building a new product; Studio for Media. Studio was an analytics tool for online advertising. It was originally designed to allow advertising agencies to track the online banner ads[1] they created, in order to give them insight into how their different banners performed. The grand idea was to catalyse a shift to a new era of "Agile Advertising", whereby advertising agencies would create better ads by experimenting with different designs and collecting data rather than simply basing campaigns on instinct and gut feeling. In other words, Studio was a software that helped advertising agencies get facts about the performance of their internet advertising campaigns, allowing them to become more data-driven.

What Adco quickly discovered when they began working with live customers was that Studio didn't solve the real problem in the online advertising industry. When advertisers saw the data that Studio provided they learned how poorly their online advertising performed compared to their advertising efforts in traditional media, and thus they

[1] A banner ad is a type of online advertising where the ad, usually in the form of an image of Adobe Flash animation is embedded onto a webpage.

dramatically cut their online advertising budgets. This would not make for a sustainable business on Adco's part.

The real problem was that ad placements[2] provided for banners on typical websites were too small, poorly placed on the site, and always competing with website content and other ads for the user's attention. Simply put, most websites did not provide ad placements that gave banner ads a fair chance to make an impact. This was the root cause, and a problem that needed to be solved in order to move more of the world's advertising spend into the online space.

Studio was split into two separate products. Studio for Advertisers (SFA) was geared at advertisers and ad agencies, and the new product, Studio for Media (SFM), would be geared at the publishers who managed the sites where banner ads were displayed. Instead of tracking the ads per se, Studio for Media would track the ad placements and provide publishers with information on how to create placements that would give banners a chance to make an impact on the user. Development on the media product began in January 2011, and by June of 2011 a first version was live with two pilot customers. In concrete terms, Studio for Media was a web application that publishers' analytics teams could login into in order to get data on how the ad placements on their website performed. The hope was that this data would inform changes to the site that would increase the value of the ad placements.

2. Early evolution of the customer on-boarding process

In June of 2011, Adco had two large pilot customers testing SFM; namely Lokali and EVB. The product was being used to track ad placements on a few large websites in Sweden, most notably lokalnytt.se. Up until that point the development team had taken care of customer delivery and support themselves, partly due to the small size of the team, but also due to the fact that the products and their integration solutions were still in development. However, this would not scale. Sometime in the not too distant future the organisation would need the capacity to on-board new customers in a consistent and efficient way, and someone needed to be responsible for building that process. As a

[2] An ad placement is a specific space on a website that is dedicated for displaying advertising.

student of high-tech entrepreneurship with a background in web development, I was recruited for the job.

I spent the first month getting to know the organisation and working on several business development tasks. It was not until July of 2011 that I officially took on the role of “Head of Delivery”. My first initiative was to map out the existing delivery process and take inventory of the knowledge that already existed in the organisation. Up until that point, the person who had handled most coordination with customers was the company’s Technical Lead, Peter Martinsson. Peter and I sat down for an interview on the current process.

The result was a basic map of the current state of the technical delivery process, which mainly involved having the development team release a stable version of Adco’s “tracking script” as well as testing that script before sending it to customers to put up on their websites. Adco’s data platform team also needed to be informed and ready when the script went live so they could monitor the effect of the data traffic from the new site. In its current form, the process required a lot of hands on work by the development team. It was also dominated by a relatively extensive testing process. The final component was the work involved in coordinating a time for deployment together with the customer. This could either be a piece of cake or a huge pain depending on the bureaucracy and people in the customer’s organisation.

In the beginning, the standard process was to coordinate a date and time for deployment together with the customer, in parallel with the internal process of releasing and testing a new version of the tracking script. The initial hypothesis was that this would save time as the processes could be run in parallel, but Adco quickly ran into a problem with the current setup; the releases were not always finished in time for the deployments. This would not be a problem if it was easy to change the deployment date, but for some customers this was a hassle. It also gave a somewhat unprofessional appearance.

One such occurrence was when it came time to deliver a new script to EVB to deploy on their website affäreridag.se. Adco had already tried deploying on affäreridag.se three times in the past, but various technical problems had appeared that forced EVB to take down the script. On one of these occasions, the website actually stopped working in certain web browsers for a day. It had been a couple of months since the last attempt when the delivery process was started for affäreridag.se in July of 2011. Internal acceptance testing revealed bugs in the script, and it took several round trips to produce a script that worked. At the same time, the customer had trouble committing to a deployment date, especially as it had to be changed during the process due to the bugs found during testing. In the end, the deployment happened two weeks later than initially intended. When interviewing the customer after the script was live, they were happy with the process, which they called "smooth". Internally at Adco it had not seemed that way and it had taken up too many man-hours to be scalable .

Similar experiences were had when coordinating the delivery process for an updated version of the script on lokalnytt.se, and deploying for the first time on EVB's website nyhetsdagen.se. Lokalnytt's script could not be delivered until Adco managed to fix a bug that lokalnytt.se had found. In order to solve the issue, Adco needed information from a third party ad server[3] provider. The process of gaining the information required took several weeks in lead-time.

While the delivery process for lokalnytt.se was hampered by problems with external parties, the process for nyhetsdagen.se was slowed down by internal technical challenges at Adco. First, the software used for final acceptance testing on the script was having problems working with nyhetsdagen.se in particular. This made it very difficult to assure the quality of the script. A workaround was finally found, but when it came time to book a deployment it turned out the data platform would not be able to handle the extra data load from the new site. The platform team would need to make preparations in the next sprint[4] which would lead to a one month's delay for the deployment. After a couple of days of internal negotiations a workaround was found for this as well, and the deployment was finally given the go-ahead.

While all of this was going on with the deployment process for SFM, SFA was also being tested on pilot customers. The difference here was that there was no previous

[3] An ad server is a software solution that allows companies to administrate their online advertising. The company uploads their banner ads into the ad server, and the ad server serves the ads to visitors of a website.

[4] A "sprint" is a concept used in the popular software development methodology *Scrum*. In Scrum, work is estimated and divided into two-to-four week periods known as sprints.

process in place. I created one from scratch, figuring it out as I went along. Rather than delivering a script to be placed on the customers website (such as was the case for SFM), SFA involved tracking customer's online banner ads wherever they might appear on the internet. Thus, a piece of tracking code had to be included in the banner ad itself. The company had previously built a self-service version of SFA that allowed customers to include tracking code in their banner ads themselves, which worked for some types of banners but not others. Adco had recently created a method of tracking the remaining types of banner ads, but there was no self-service interface for this in the product. This meant that the process to get customers who were using certain types of banner ads started with SFA was for them to email the banners to me, whereby I would convert them manually using internal tools and email them back. This process was laborious and not particularly scalable, but necessary since the tracking method was so new. Converting the banner ads myself, I could identify issues early and report them back to the development team so that they could improve the tracking code for the future.

The first pilot customer using the new tracking method for SFA was an advertising evaluation consultancy. They were going to use Studio to evaluate a list of five campaigns for a large consumer food company in Scandinavia. The first of the online campaigns with Adco's tracking code in the banner ads was set to begin in August of 2011. The banners were converted without too much hassle, but problems appeared when they were sent out to the websites set to display them. The converted banner ads were making it difficult for the website owners to upload to their ad servers, as their ad servers did not recognise the format of the banner ads containing Adco tracking code. Adco's tech lead, Peter Martinsson, and I collaborated on a solution and managed to find one by coordinating directly with the individual media companies, but it became clear that launching SFA campaigns would require some level of manual work, at least until Adco became large enough that ad server providers began updating their systems to support banner ads tracked with Adco code.

3. Building Studio 2

In September of 2011, Johan Björk (Adco's CEO and Co-founder) called for an all-hands meeting where he stated that Adco would form a new sub-team to build the next version of Studio, including both a new SFA and a new SFM. By then the company had increased it's number of employees to over twenty, expanding the development team with more programmers, as well as the product team with one more user interface designer and one interaction designer. The new Studio 2 Team would include the product team, approximately half of the development team, as well as other key persons. As the company's Head of Delivery, I was tasked with helping the product designers make Studio 2 completely self-service for new customers. There were some design challenges, as well as some fundamental technical challenges with this notion, but if it could be achieved it would make Studio infinitely more scalable and deliverable than it was at the time. Customised enterprise versions of the product would still be available, which would require resources from the delivery team when on-boarding a new customer. The directive from management was that the delivery team should be able to activate a new customer without involvement from the development team in order for the product to be considered "finished".

I began working with the product team to design a self-service flow for Studio 2. For SFA this was not too difficult, but for SFM there were certain technical challenges that meant that a script could not be created automatically for all sites. A script could be generated if it was only to pick up certain data points, but for it to pick up every data point required for the full version of Studio, a developer would still need to write website-specific configurations in the code. The solution would be to limit the functionality in the most basic version of the product to the point where a tracking script could be generated automatically, whereas more advanced versions of the product would still require a manually configured script.

Before long there was a specification ready to be implemented and tested, but while the product team was working productively to create wireframes[5] and mockups[6] for Studio 2, the development team was having problems. They had chosen to build the product on certain cutting edge technologies that would make for a very responsive experience for the user once the app was finished, but also caused a lot of unplanned

[5] A wireframe is a simple line drawing representing the layout of a Graphical User Interface. It is used as a part of the user interface design process.

[6] A mockup is an image that shows exactly what a Graphical User Interface will look like once implemented.

work due to their immaturity. Studio 2 was lagging behind. It was initially meant to be finished in just six weeks, but now it was looking as if it would take much longer.

After two months had passed and there was still no Studio 2, Peter Martinsson was brought in to work more hands on with the Studio 2 team in order to get the development back on track. The team was restructured, different people were put in charge of the various areas of the development, and the technical foundation of the new app was rethought from the ground up. Things were starting to get back on track. At the same time Adco was beginning to set up its first proper sales force, with a dedicated office in Stockholm. Even though the Studio 2 development process was beginning to show promise, no one was prepared for what was to happen in the coming months.

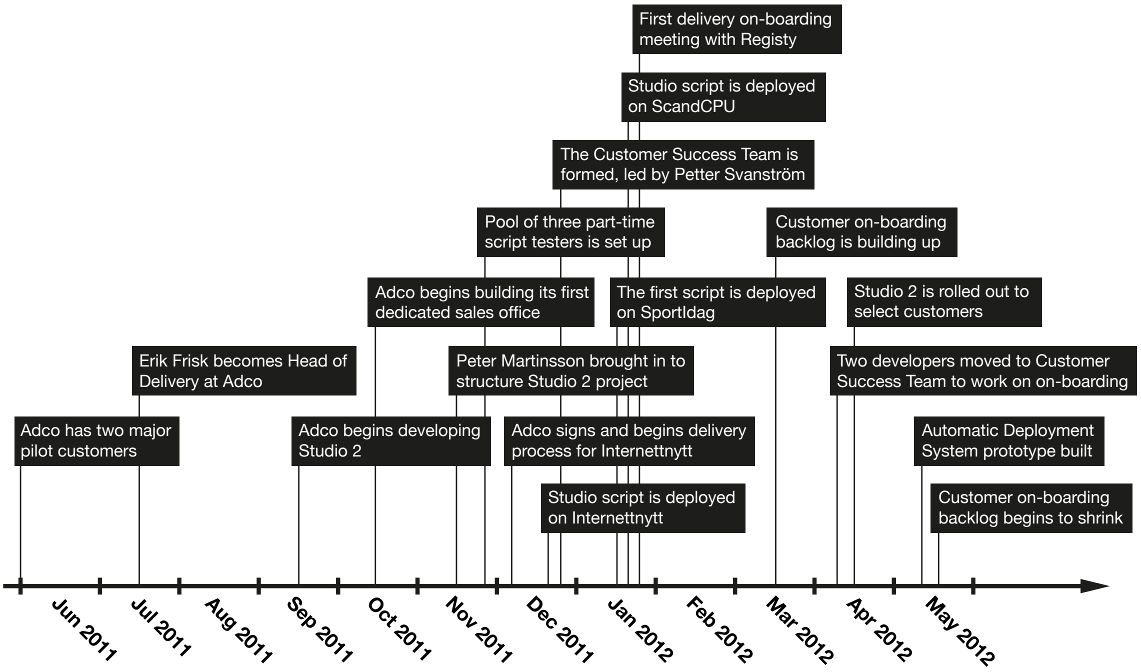
First delivery on-boarding meeting with Registy
Studio script is deployed on ScandCPU
The Customer Success Team is formed, led by Petter Svanström
Pool of three part-time script testers is set up
Customer on-boarding backlog is building up
Adco begins building its first dedicated sales office
The first script is deployed on SportIdag
Studio 2 is rolled out to select customers
Erik Frisk becomes Head of Delivery at Adco
Peter Martinsson brought in to structure Studio 2 project
Two developers moved to Customer Success Team to work on on-boarding
Adco has two major pilot customers
Adco begins developing Studio 2
Adco signs and begins delivery process for Internettnytt
Automatic Deployment System prototype built
Studio script is deployed on Internettnytt
Customer on-boarding backlog begins to shrink
Jun 2011
Jul 2011
Aug 2011
Sep 2011
Oct 2011
Nov 2011
Dec 2011
Jan 2012
Feb 2012
Mar 2012
Apr 2012
May 2012

Fig 2 — Timeline showing some of the major milestones throughout the case.

4. The great wave approaches

Throughout fall of 2011, many suspected that there would come a “singularity” in terms of customer on-boarding and delivery. Adco had only had a few pilot customers to work with, and for a long period of time the company had not had the experience of bringing on board a completely new customer. My job as Head of Delivery was not only to take care of the technical delivery of updates and new features to existing customers, but also to design the on-boarding process to be able to handle a throughput of customers that was orders of magnitude higher than the current one. I did what I could to predict the future scenario and build for that, but I encountered several challenges. For one, it was difficult to build processes and tools meant to support an entire team of people when there was currently only one team member. Any processes put in place for organising the team were purely theoretical, and could not be tested in a real environment until more delivery engineers were hired. Secondly, it was difficult to prioritise technical development meant to make the product more "deliverable" ahead of time. Even though it was clear that certain technical development needed to be made to make the product deliverable on scale in the future, there were always more immediate technical problems that the development team needed to prioritise. Finally, it was difficult to anticipate what technical problems would occur in terms of integration with new customers' systems. For example, since the script for SFM required non-trivial manual configuration depending on how the customer's website was built, one could imagine encountering problems if new customers had unanticipated solutions on their websites in terms of their advertising. Very little could be done to prepare for this as there was simply too much uncertainty. The only way to tackle the problem in a relatively lean way would be to face it head on and solve problems as they were encountered.

Adco's first dedicated sales office was created in October and November of 2011. The first hire was an established sales manager by the name of Stefan Brunn, who had previous experience in leading sales teams in fast-growing startup companies. Him and Adco's COO, Anders Ström, spent time in Stockholm recruiting sales representatives. Adco signed three ambitious sales reps in just a few weeks, and they soon began making a dent in the Swedish and Norwegian markets. When the office was first set up, it was believed that the sales cycle would be somewhere around nine months, which was the case for many other enterprise solutions in the advertising technology space. However, Adco soon discovered that the cycle for Studio was closer to three months on average. This was a very pleasant surprise, but it also meant that customers waiting to be on-boarded began piling up quickly. The rest of the company was effectively six months behind in terms of product development and organisation due to the sales cycles being shorter than expected. The organisation was unprepared for the unexpectedly fast growth.

5. The first new customers

Mega Media, the UK

In November of 2011, Petter Svanström, who had been hired as a business developer during the summer, came to me and explained that Adco had signed a new SFM customer. The company was a large media house in the UK that had many websites under their umbrella. They wanted to give Studio a try, beginning with three of their smaller websites and with the understanding that they would expand it to larger sites if everything worked well. During the sales process the customer, Mega Media, had explained that they could not go ahead with the trial if they would need to do any development on the websites. It would simply be too difficult for them to get their IT departments to make any changes, and so they wanted everything to be handled through their ad serving software. They had asked Petter if this was possible, to which he had answered yes. Mega Media signed the contract.

Petter told me of the concern Mega Media had had as he handed over the project to me, and that we would need to handle everything through the ad server. There was no solution in place for this, and no one in the organisation had the necessary understanding of ad servers to know how this could be done. However, since the only "development" necessary on the site was to copy and paste a small snippet of code, I decided to talk to Mega Media and see if this could not be done. After all, they might have feared a much more complex integration on the site when they said they wanted the tracking to be done completely through the ad server.

Lars Nilsson, lead developer for the Studio tracking script, and I organised a telephone meeting with the representatives from Mega Media. We explained the technical setup process for Studio, including adding the tracking code to the sites. Mega Media was not pleased. For them, even copy and pasting a few lines of code into the website source code was considered "development", and would signify a complete showstopper. Lars and I promised to look into possible solutions. We suggested to get Mega Media started with the advertiser product, SFA. That way Mega Media could test Studio without adding code to the sites. Even though the reports in SFA would not be as relevant as those in SFM, the test would give the representatives from Mega Media a stronger case to approach their IT team and tell them they wanted proper SFM code on the sites. Mega Media agreed.

In order to get Mega Media started with SFA, all they would have to do was to email banner ads they wanted to track on the websites to us, and we would convert the banners to include Adco tracking code. When Mega Media sent over the banners, we discovered that they were in a format that we had not seen before, and which were not supported. Developers were brought into the loop to build support for the new type of banner, but the process proved to be more complex than initially thought. After several round trips between Adco and Mega Media, patience and time ran out on both sides and communication seized.

Internettnytt, Norway

While all of this was going on with Mega Media, one of Adco's sales representatives, Robert Karlsson, was beginning to close deals in Norway. In December of 2011, Robert closed a deal with Internettnytt; a very large publisher on the Norwegian market. Adco began to scurry to make preparations. Norway was showing signs of being a very attractive market for Adco, with great opportunities to spread quickly due to online publishers on the market working closely together and tending to adopt new technologies on mass. This was both a blessing and a curse. If the first big customer on the Norwegian market, namely Internettnytt, was satisfied it would set a good precedent and help spur Adco's growth tremendously, but if they were unsatisfied all other big players on the market would know, and it would make things significantly more difficult. An internal task force was created at Adco to ensure that Internettnytt's on-boarding went smoothly. The members were Robert Karlsson, Carina Sten (a highly skilled representative from Adco's analytics and R&D department), William Persson (who had recently left his job at Adco's customer EVB to come work with Customer Success at Adco), and me.

Internettnytt signed on the 5th of December, and the deadline for going live on their site was the 19th, just two weeks later. Perhaps the biggest concern was that deploying on Internettnytt would more than double the amount of load on Adco's data processing platform, which was continually being developed to increase its capacity. In fact, one of the largest technical challenges for Adco up until that point had been building a data platform that could cope with the huge amount of data needed to make Studio work. I spoke to our data platform developers and our Head of IT Operations regarding Internettnytt, and there was a possible solution. The only caveat was that they would need to do some work to prepare the platform for the huge amount of new traffic, which would take longer than the deadline set with Internettnytt. We agreed on sampling the data, only collecting information on 10% of the users on Internettnytt's site for the first two or three weeks. That would give the platform team enough time to prepare for collecting 100% of the data.

When this information was brought back to the internal task force, there were loud protests from Robert and William. They were concerned that Internettnytt would not accept this, and may make demands to postpone payment until 100% of the data could be collected. Carina and I both argued that the sampling would not matter much as the extrapolated data from 10% of users would give extremely good accuracy for everything besides a few data points. In the end we made a compromise and decided to go ahead with the temporary data sampling, but to not tell Internettnytt about it.

On Tuesday the 13th of December—six days before Internettnytt was scheduled to go live—the configured script was finished, tested and ready to be sent to Internettnytt for their own quality assurance. I emailed our technical contact at Internettnytt in the morning with the script. He responded in the afternoon and his answer came with a surprise; he asked if the included script was only for their main site, internettnytt.no, or for all the the sites they were supposed to go live on. It turned out they had a whole network of approximately twenty websites, and were expecting all of them to be tracked with Studio. No one at Adco knew about this. Robert (who had sold the solution to Internettnytt) had not thought to ask, and neither had anyone else later in the process.

Adco's Internettnytt task force went into crisis mode. The company had a total of four websites running Studio to date, so none of the processes were designed for the enormous new throughput. Nevertheless, the customer had to be made happy. William, Robert and I created a plan for deploying on all of the websites the following week, meaning Internettnytt would not be up and running on the same day as promised, but at least in the same week. The development team slaved away to finish all of the configured scripts on time.

The main site, internettnytt.no, went live on the 19th as promised. Many of the medium sized sites went live in the following couple of days. Some of the smaller sites could not be delivered the same week due to roadblocks on the customer's side, and so were delivered after Christmas instead. When asking Internettnytt after the delivery process about whether they were satisfied, it turned out they had not cared too much about being live on the 19th. They were just tremendously satisfied with the quick and reliable service they had received. This was a big learning for the organization; several misunderstandings had come up between the customer and Adco throughout the process. In light of this, Adco created a checklist with information for sales reps to collect in combination with each sale. This information would then serve as an order from Sales to Delivery when a new customer was to be on-boarded.

6. Experimenting with quality assurance

Throughout the history of Adco's customer on-boarding process, there had always been a need to test the site-configured tracking script before delivering it to the customer. Part of the testing was to make sure that the script tracked all ad placements on the site and collected data in the intended way, but the most important thing to test for was the potential of the script destroying some functionality on the customers' sites. Failing to catch such errors could potentially lead to catastrophic damages to Adco's relationship with customers. Towards the beginning of Studio's life, in spring of 2011, this had been done by random non-developers in the team. The reason the final acceptance testing was done by non-developers was to avoid bias in the testing—the developers themselves would be less eager to find mistakes in the code.

This process worked for a while, but as the company grew and more organisational structure had to be put in place, it was no longer practical to ask random individuals for help with testing every time a new script was to be deployed. In response, Adco tried hiring a tester full time. After a few weeks, it was clear that this strategy was not proving cost-effective. The tester could not deal with the variable work load and was not enough of a self starter that he could spend his downtime improving the overall process.

As the workload varied heavily depending on how many customers were set to be deployed at any given time, the next experiment was to hire part-time workers to come in and do testing whenever there was a need. I contacted a staffing company who specialised in finding part-time jobs for local university students. The idea was to create a pool of five or six students who could be contacted on relatively short notice whenever testers were needed. Having many testers in the pool would ensure that at least two would be available at any given time. In the end the pool only required three students to nearly guarantee sufficient availability, and the experiment was a success. The pool of three university student testers was set up in late November 2011, and from there on out nearly all final acceptance tests were done by them. The solution proved to be scalable and flexible enough to deal with the turmoil in the customer on-boarding process.

7. Inito and Northern Air

In November of 2011, Johan Björk (CEO and Co-founder of Adco) came to me with a special assignment. The company had been discussing a new product, which would extend the functionality of Studio to reach new heights. Using the detailed information that Adco had about users on websites all over the world, the new product, Inito, would allow advertisers to do extremely precise targeting of their ad campaigns. They would be able to specifically target users with the right online behaviour, and do it only on websites that provided the proper media environment for that specific message or brand. This would be completely unprecedented on the market, and Adco was beginning to reach a level of technology from which the product was possible.

In order to build and test a proof-of-concept of Inito, Johan had spoken to Roger Kamp, Director of Global Marketing at Northern Air, who had been one of the first customers of SFA. The idea was to do a proof-of-concept live with Northern Air.

Even though there was some development work to be done, it made sense to begin the delivery process with Northern Air right away, as they would need to work through a lot of bureaucracy in order to get all the technology in place. Roger wanted documentation on how the product would work, and so some high-level descriptions were created. Roger also needed several other people in his organisation to sign off on the idea, and so a meeting was scheduled between, Peter Martinsson (then Director of Engineering at Adco), Lars Nilsson (lead developer for the Studio tracking script), Roger Kamp, Roger's colleagues, and myself. The purpose of the meeting was mainly to reassure Northern Air's IT department that the proof-of-concept was safe to try, and they gave the thumbs up after clear descriptions from Lars Nilsson regarding how everything would work. Roger explained that the next step would be for Northern Air to begin their approval process for getting the code onto the Northern Air website, which would "probably take some time".

While Northern Air was working through their approval process, Adco was working on building the proof-of-concept. This was no simple task, especially given the focus Adco was putting on finishing up the work on Studio 2 at the time. When Roger Kamp got back to me in the end of January 2012 and explained that they were nearing the end of the approval process, we had hardly begun building the proof-of-concept. I managed to buy some time with Roger and the development team prioritised the work on Inito. After a few days there was a basic piece of code to deliver to Northern Air, which would get the project started. Finally, Northern Air put up a tag on their website that allowed Adco to deploy a script. The only problem was that there was still a lot of work to be done in Adco's data processing platform in order to make sense of the collected data. This work was done in parallel with coordinating with Northern Air's marketing department for how they would like to use the new technology. This particular work was done mainly by Johan Björk himself and Anton Stark (Adco's Director of R&D) over the course of several months, which bought the development team enough time to get the backend technology in place. At the time of writing, in May 2012, the proof-of-concept is still being explored together with Northern Air.

8. Building the on-boarding team

Throughout the fall of 2011, I felt that customer on-boarding was gearing up to become a bottleneck in the organisation. With the blessing of Adco's COO, Anders Ström, I began interviewing candidates for another Delivery Engineer role. The theory was that the Delivery department would need another couple of people sometime early 2012 to deal with the initial wave of new customers, and that it was necessary to bring these people on board a month or two in advance.

I interviewed several candidates and a couple went on the short list. The only thing preventing them from being hired was the fact that Adco was beginning to run short of money. Current investors were planning to do a bridge investment which would keep the company afloat, but it was not the proper time to increase the burn rate by hiring new people. In December of 2011, I spoke to Johan regarding the need to increase the number of people working with delivery until new people could be brought on board. A few weeks prior, the company had made a new recruitment in William Persson. William had previously worked for one of Adco's most important pilot customers, EVB. In fact, he had been the person managing the Adco integration and development project on EVB's side. He had decided to quit in order to look for new opportunities, at which point Adco snapped him up. Now Johan suggested that William help out with delivery until more people could be brought into the team.

William and I began working together throughout the rest of December of 2011, but it was not enough of a change to completely deal with the stream of new customers. Johan and Anders took William and me aside and explained that they would try a new way of organising customer on-boarding and relationship management in the company. Adco was to create a Customer Success Team lead by Petter Svanström, then Key Account Manager. William and I would become Customer Success Managers in the new team.

The newly formed Customer Success Team began meeting to define a work structure and an appropriate division of roles. The initial idea was that Petter would have the long-term responsibility for each new customer, and that William and I would come in earlier in the process. William would work with the sales representatives to gather all information required to do the technical delivery, I would take over the project when it came to executing the technical delivery plan, and Petter would take over once the customer was up and running to take care of long term satisfaction and up-sales. After attempting to follow this framework for a few weeks it became clear that it wasn't working. In practice, there was still too much uncertainty in the delivery process to be able to do an effective pre-study phase and execution phase for each new customer. Instead, each project presented its own unique challenges, often because of the specific quirks in different customer's implementations of 3rd party systems that Adco needed to be compatible with. The result was that it was more effective for one Customer Success rep to handle the entire delivery process for a given customer from start to finish. A more fitting division of roles evolved naturally in the Customer Success Team, and after a few weeks the decision was made to make the new framework the official one. In the new way of working, William took care of SFA customers (who were beginning to come in more and more often), and I took care of SFM customers. Petter handled overall contact with customers to ensure nothing fell between the cracks, and later helped with on-boarding SFM customer once the throughput was too high for just one person to handle.

In the beginning of the life of the Customer Success Team, it had been organised around the idea of a linear and predictable process. When it turned out that the reality of delivering Studio was quite different, the team was reorganised with more of a problem solving mentality. Each new customer was assigned a Customer Success Manager who was tasked with getting them up and running using whatever means necessary. Building on this idea, Petter and I created the concept of a "technical advisory workshop". This would be a workshop held with customers' technical representatives, preferably prior to signing them, where the technical delivery could be discussed and any potential challenges could be identified as early as possible in the process. The first of these was held with a very large new customer in Norway; Registy.no. It was at this workshop and subsequent similar workshops that the real value of the technical advisory workshop became clear. Identifying technical issues early on was one advantage, but the most helpful thing was that the workshop allowed Adco to establish a somewhat informal relationship with the customers' technical stakeholders, which made it a lot easier to communicate around challenges in the delivery process later on.

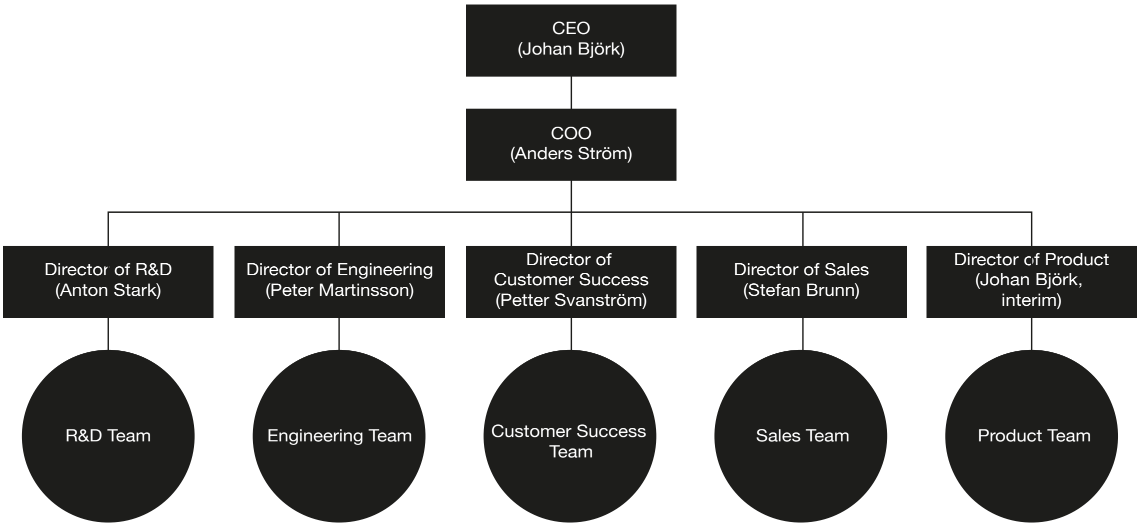

Fig 3 — Adco's organisational structure in the beginning of 2012.

9. Ups and downs with new customers

During the last quarter of 2011, the newly formed sales team had been hard at work selling Studio; mainly in Sweden and in Norway. By the end of the year many customers were getting ready to give Studio a shot. There are too many stories to tell them all, but a representative sample can be found below, including a couple of the more interesting cases.

SportIdag, Sweden

SportIdag.com was an online publisher that signed up to try Studio in the end of 2011, with the understanding that they would go live in January 2012. It was not a particularly large site compared to Adco's existing customers, but there was one thing that made the case different; they were not using an ad server from a recognised ad server provider to administrate their advertising. Instead, they had built their own ad server. This was no problem for Adco in and of itself, but it did introduce a new degree of uncertainty.

By mid January 2012, Adco had a script ready to deploy on SportIdag. Some of the script's fundamental workings had to be changed to accommodate the way SportIdag had built their ad server. The deployment happened on Monday the 16th of January. Adco's testers commented that the site seemed unusually slow right after the deployment but that seemed to be the case both with the script on and with the script blocked, meaning that it was most likely just SportIdag having unrelated temporary problems with their servers. Seemingly, everything had gone well.

But unfortunately it was not so. The day after the deployment, SportIdag contacted me and said that they had been getting complaints from some users of the site that it had become very slow since the deployment. SportIdag had investigated and found the cause to be the Studio tracking script, so they took it down. They asked us to fix the script and told us that they could try deploying again once the script was optimised.

Lars Nilsson set about optimising the script, and managed to get quite far. However, the new script required changes in SportIdag' custom-built ad server. They had built the ad server in a way that fundamentally prevented Studio from being able to track certain data points, which had to be fixed. All technical changes were coordinated with Adco's technical contact person at SportIdag. The changes were made but rejected by Adco because they were either incorrect or incomplete. At the same time, the total number of customers was increasing more and more and so it was increasingly difficult to prioritise working on a custom solution for SportIdag. Even though the customer relationship was kept relatively positive, time dragged on. It was not until May of 2012 that SportIdag set a plan in motion to begin using a known ad server from an ad server provider after the summer, and it was decided that the integration work with Studio would wait until after they had made the switch.

ScandCPU, Sweden

Whereas SportIdag was an example of a customer where the technical conditions for success simply weren't there, ScandCPU was an example of a delivery project that went relatively smoothly. The customer was using an established ad server provider, TechServe, who Adco had good contacts with. A few questions regarding the workings of the ad server came up while writing the site-configured tracking script, and Adco was able to ask these directly to contacts within TechServe's support team. Issues were resolved promptly and ScandCPU went live on time in January 2012.

Registy, Norway

Even though Internettnytt had previously caused a lot of excitement within Adco, Registy was an even bigger haul. Registy was the largest website in Norway in terms of traffic. Robert Karlsson had been trying to get them as a customer for a while, and the only thing blocking a signature was that Registy needed buy-in from their IT team to go ahead with the integration project. Robert Karlsson, Anton Stark (Adco's Director of R&D) and I went to see Registy in January of 2012. Registy had gathered all stakeholders that needed to give their consent to go ahead, and after having all of their questions answered Registy signed on the spot.

I created a rough time plan for when things would go live together with Registy. It was designed so that the final deployment on the website would be in March. However, during February and March the development team was so overburdened with finishing the Studio 2 app that it was very difficult to get script configurations done. The consensus in the company was that delivering to customers was the first priority, but in reality small, urgent tasks kept coming up and distracting the script development team. The lead time for creating a configured script was beginning to become absurdly long; approaching several weeks rather than a couple of days as before. It was not until March 2012 that the script developers were able to get to creating a script for Registy, and during the work they discovered that Registy had some unusual solutions for loading ads on different sections on their website. After we discussed this with Registy it turned out Registy was moving all of their ad placement to a technical solution that would require a more complex integration with Adco's tracking script. Not only that; when learning more about how Adco's tracking script worked they didn't like some of the things that the Adco code did on the website. I had been exploring a more stable way of integrating with Registy's ad server provider, TechServe, which would allow the tracking script to be less invasive. Registy explained that stability was more important for them than getting up and running soon, and so they opted for the latter solution, which would require some research and development work together with TechServe. At the time of writing, in May 2012, the new solution is still being investigated together with TechServe.

10. The Automatic Deployment System

Throughout the end of 2011, and continuing into 2012, Adco had brought in a consultant with substantial experience in the advertising technology industry to review and help develop the company; both from a business perspective and a technical perspective. In February 2012 the consultant, Pieter, had a workshop with Lars Nilsson, Jakob Varg (a developer in the tracking script team), and me regarding how Adco's tracking script worked. After Lars, Jakob and I had explained the script configuration process, and the workings of the tracking script, Pieter had some major reservations. He was concerned that the technical configuration was much too complex, and that the process would not scale. Also, it would become more and more difficult to maintain the scripts already live on websites as the number of existing customers grew. His strong advice was to begin right away with building an automatic deployment system, where scripts would be managed through a graphical user interface rather than with extensive manual work. There were some serious technical challenges with this, but nevertheless Lars, Jakob, and I created a roadmap for what would need to be solved in order to build the automatic deployment system. It was clear that Pieter was right in the fact that the current process was unsustainable, and that Adco would hit a wall in terms of customer on-boarding sooner rather than later if nothing was changed. Script configuration was already a clear bottleneck in the overall customer on-boarding process.

The work in creating an automatic deployment system could potentially be made much easier if certain operations of the tracking script could be done in the customer's ad server. A roadblock that Adco experienced was that there was a limited understanding of what capabilities and features different ad servers had. A deep understanding of the systems that Adco needed to integrate with would have been a valuable asset.

Throughout the coming weeks, I pushed to have the automatic deployment system prioritised in the development work. It was not easy though. There was still substantial work to be done on finishing Studio for Media 2, and work on Studio for Advertisers 2 had barely begun. I explained the likely consequences of not prioritising the automatic deployment system to Johan and Anton, saying that it would seriously limit the speed at which new customers could be brought on board. They understood the concern, but argued that the work would have to wait in favour of finishing Studio 2.

11. Hitting the wall... and breaking through

Customers were coming in at an increasing pace and creating configured scripts was taking an absurdly long time by March of 2012. The development team was running late on finishing work on Studio 2, and this was making it very difficult to prioritise work on investigating solutions for customers who required some level of custom integration. The problem was that nearly all customers needed some degree of customisation in the tracking script since their systems would not allow the standard code to be used. The Customer Success Team was quickly becoming overwhelmed by having to deal with the delays and explaining them to customers.

In April of 2012, it was beginning to dawn on the rest of the organisation that something needed to be done to streamline the on-boarding process and work off the backlog of customers that was quickly building up. Anders Ström and I took a walk, where I explained the situation and suggested that the problem would not be solved by only making changes in the Customer Success Team. Instead, the problem had more to do with the interface between Customer Success and the development team. It was a problem that could only be solved with actions outside of Customer Success's reach in the organisation. In the subsequent days, there were serious discussions in Adco's management team regarding what needed to be done. The result of the discussion was that two script developers, Jakob Varg and Nicklas Sundqvist, were officially moved over to the Customer Success Team and tasked with working on customer delivery full time. This was also made possible by the fact that SFM 2 was finally ready to be rolled out to customers. Lars Nilsson was also temporarily moved over to the Customer Success Team in order to help work off the backlog.

A week or two after the reorganisation, things were starting to move again. In order to help visualise the work, I set up a Kanban[7] board for the technical delivery process with post-it notes representing each script that needed to be configured and deployed. At that time the number of websites in the pipeline was up to approximately thirty, and the board had a positive effect in communicating to everyone in the company how heavy the workload was. It also made it easier for the Customer Success Team to com-

[7] Kanban is a tool whereby a process is mapped out on a board, and tasks moving through that process are placed on cards underneath the step they are in. The tool helps visualise work currently in progress, as well as pending work.

municate around each site and get a feel for how far off different websites were from being delivered.

Throughout the end of April and beginning of May of 2012, websites were beginning to get deployed for the first time in several weeks. Lars Nilsson used some of his time working in the Customer Success Team to begin building the automatic deployment system that Pieter had previously pushed for. The system would not be able to handle every customer case, but the estimate was that it would be able to handle the majority of customers with the push of a button. Other customers might need some degree of custom coding, but the deployment system would take care of script management even for these customers once the custom code was built. At the time of writing, in May 2012, there is a prototype of the automatic deployment system which has not yet been tested live on customers. The hurdles in Adco's customer on-boarding process were beginning to be overcome. By mid May 2012, the customer backlog that had seemed overwhelming just a few weeks earlier was beginning to melt away.

Discussion

Looking at the case study, three major themes are prevalent throughout the development of the customer on-boarding organisation at Adco:

- dealing with unforeseen challenges
- a gradual transition towards efficiency and automation
- interfacing with customers and other parts of the organisation

It is worth noting that Blank's (2007) startup growth phases can be related to the first two themes in a very elegant way. There is a high degree of uncertainty around every aspect of the business in the early growth phases, leading to unforeseen challenges that must be dealt with practically by the customer on-boarding team. As the company transitions into the later growth phases the overall level of uncertainty decreases, which leads to fewer unforeseen challenges in the on-boarding team. Transitioning into the later growth phases also means that the company begins focusing on growing the size of the business. Thus, the customer on-boarding team takes advantage of the decreasing level of uncertainty to set up more efficient processes that will aid in the startup company's expansion.

In the following chapter, the case is looked at from the perspective of each of the major themes, tying back to the theoretical framework put forth in the literature review.

Theme 1: Dealing with unforeseen challenges

The case highlights many unforeseen problems that appeared throughout the development of the customer on-boarding process at Adco. Looking at the literature, it is reasonable to expect many unforeseen challenges to appear. First of all, the literature on startup management emphasises the need for learning through trial-and-error, implying that mistakes are an essential part of the development of a young startup company (Blank 2007).

Organising for flexibility

The literature leaning towards the side of software development in Figure 1 further explains the predominance of unforeseen technical challenges. When customer on-boarding requires complex software integration as is the case with Adco, the customer on-boarding process is subject to the challenges mentioned in the literature on enterprise software integration. In other words, people will tend to underestimate the cost and complexity involved in such projects, often leading to failure (Rettig 2007). On top of all this, one can expect the software being delivered to have bugs and issues that will slip past the company's test routines. Software testing is largely based on experience and intuition (Choudhary & Kumar, 2011). It may be very difficult to find software testers with the right experience in the case of a startup that operates with constant change, high degrees of uncertainty and that builds innovative software. The number of software issues experienced by the customer can be reduced with good testing routines, but it will be extremely costly and time consuming to prevent them all (Choudhary & Kumar, 2011). Examples of this in the case include the technical issues experienced when trying to deploy on affäreridag.se and sportidag.com. In both cases the issues were not caught by Adco's already quite extensive testing routines.

The literature on the right side of the continuum framework (i.e. the literature leaning towards the side of managing customer relationships) also helps shed some light on the theme of dealing with unforeseen challenges. Partha and Roy (2011) stress the need for *organisational readiness*, which refers to creating an organisation that is agile and can cope with varying customer needs.

The solution for Adco was to organise the customer on-boarding team for flexibility and problem solving. When the Customer Success team was first formed, its members were organised around a standardised process where one person took care of a customer during certain steps and then handed the customer over to the next person. This quickly proved to be inefficient, as each customer required its own set of special considerations and the hand-over between team members became very difficult. In the end the team was reorganised so that one person would lead the project of on-boarding a given customer from start to finish, allowing that person to improvise and deal with unique challenges as they appeared.

Another example of organising for flexibility was the creation of the pool of part-time software testers, which allowed Adco to deal with varying workload on short notice in a way that employing a single full-time software tester could not. All of this leads us to the following proposition:

> *Proposition 1: During a B2B SaaS startup's early growth phase, the customer on-boarding team should primarily be organised for flexibility and problem solving.*

A higher focus on efficiency and repetition may be appropriate as the organisation transitions into the later growth phases and learns how to automate and standardise larger parts of the on-boarding process.

The danger of firefighting

An important consideration that is illustrated in the case is the effect that the organisation falling into a firefighting dynamic has on customer on-boarding. In chapter 11 of the case, much of the development team's resources were spent on fixing small but urgent issues. This meant that there was little time left over to create robust integrations with customers, which in turn contributed to more issues down the line. It was not until the organisation transferred more resources to customer on-boarding, and internal automation tools were built, that the organisation began making progress on the customer backlog.

Theme 2: Transition towards efficiency and automation

In the beginning of the case, Adco did not have a formal customer on-boarding process. Tying this back to Blank's (2007) company growth stages, Adco transitioned from the Customer Validation stage to the Customer Creation stage, and later to the Company Building stage. As the company transitioned into the Company Building stage, the requirement on the customer on-boarding organisation became to activate new customers at a higher and higher pace using a fixed amount of resources—essentially building a more and more efficient process. The efficiency of the on-boarding process directly affects the company's Customer Acquisition Costs, which is one of the central KPIs for a SaaS business (Skok 2010).

Learning by trial and error

A large amount of trial and error was required in order to design an efficient process. At the outset, the organisation's knowledge on how to efficiently on-board a customer was low. Unforeseen problems appeared as new customers were on-boarded (as illustrated in *Theme 1*), and different tactics were employed to deal with these issues. The organisation learned what worked and what didn't over time, which gradually led to a more efficient on-boarding process.

Tying back to the customer relationship side of the theoretical framework in Figure 1, an example of learning from trial and error is the use of customer distinction strategies (Cheverton 2012) to pinpoint which customers should be given extra attention and which customers have a low probability of being profitable. It takes a certain amount of experimentation and learning from mistakes to determine how key customers can be distinguished from unprofitable ones. In the case, SportIdag could not be successfully on-boarded due to their home-built ad serving solution. A lot of time and resources were wasted pursuing the project. In contrast, ScandCPU's technical setup allowed for a very smooth on-boarding. The process would be made more efficient by distinguishing between such cases earlier on, but the organisation needs to try and fail before learning how to identify them.

Another example, this time related to the software development side of the theoretical framework, is the gradual improvement of the software test routines at Adco. The number of software errors that slipped past the test routines decreased over time, even though they never completely went away. By observing errors in the software over time—both those that were caught by the test routines and those that slipped past—the software developers and testers gained experience that allowed them to intuitively recognise potential errors or areas of high risk. Since good testers must rely partly on experience and intuition (Choudhary & Kumar, 2011), the overall software test at Adco became more reliable and efficient over time. All of this leads us to the following proposition:

> *Proposition 2: As a company transitions into the Company Building stage, the customer on-boarding process gradually transitions from experimentation to efficiency, and the experimentation is required to attain said efficiency.*

Budgeting for support tools

In the final chapter of the case, the development of the automatic deployment system greatly helped work off the backlog of customers that had been building up over the previous months. Ling and Yen (2001) state the importance of automated processes in a successful CRM strategy. The results from the case suggest that this holds true for customer on-boarding as well. However, building automated systems and internal support tools require time and resources which could otherwise be used to develop functionality in the product that is more directly valuable to customers. Furthermore, chapter 3 in the case *(Building Studio 2)* suggests that full automation may require certain functionality to be sacrificed completely. In other words, certain functionality may require a delivery that is so complex that the increase in CAC may simply not be worth it. All of this pertains to both the software development side and the sales side of the theoretical framework and it suggests that a company needs to find the right balance between functionality in the product and automaton in the customer on-boarding process, which leads us to the following proposition:

> *Proposition 3: It is essential for an organisation to actively consider the balance between efficiency in customer on-boarding and functionality in the product, and subsequently budget a certain amount of time and resources for building automated processes and internal support tools for customer on-boarding.*

Building automated processes and internal support tools are not obvious activities, and may therefore be neglected in favour of adding functionality to the product. The results from the case suggests that this may be a mistake in the long run. This is supported by the literature on concurrent engineering. Rahmandad and Weiss (2009) highlight the importance of not sacrificing capability-building activities under times of high pressure, as this may tip the organisation into a firefighting dynamic.

Theme 3: Interfacing with customers and other parts of the organisation

A major portion of the customer on-boarding team's daily activities at Adco involved working together with other departments in the company as well as with customers.

Some forms of collaboration proved more productive than others, as can be seen in the case and further validated by theory.

Involving all stakeholders early

The customer on-boarding team at Adco depended heavily on the company's engineering team, who needed to be involved when customers experienced technical issues with the product or when customers with special technical requirements were to be on-boarded. Chapter 8 of the case describes the concept of the *technical advisory workshop*, which was introduced for the purpose of involving all stakeholders early in the on-boarding process so that potential challenges could be identified and overcome. This included the customer's technical stakeholders as well as Adco's customer on-boarding and engineering teams. The initiative proved successful at Adco, and is further validated by theory. The literature on concurrent engineering on the software development side of the theoretical framework states that involving all stakeholders in the beginning of the project can lead to faster cycle times and higher productivity due to a reduction in re-work in the later stages of the project (Rahmandad & Weiss, 2009).

Another positive effect of introducing the technical advisory workshop at Adco was that it allowed Adco's team members to meet the customer's technical stakeholders in person early on. This enabled a relationship to be built that made it easier to communicate and coordinate with the customer in later stages of the on-boarding process. The importance of this information flow is supported by the literature on service delivery strategy (Partha and Roy, 2011) on the customer relationship side of the theoretical framework. All of this leads to the following proposition:

> *Proposition 4: The customer on-boarding team should aim to involve all stakeholders early in a given on-boarding project in order to quickly identify potential issues and facilitate information flow.*

Collaborating with the sales team

Another insight from the case has to do with the customer on-boarding team's interaction with the sales team at Adco. The misunderstandings that happened when on-boarding Internettnytt, Mega Media and Registy are all examples that shed light on this dynamic. In the case of Internettnytt, the sales organisation had not been aware of what questions to ask the customer in preparation for the on-boarding process. In Mega Media's case, sales had misunderstood a fundamental technical limitation in the product. Finally, Registy was simply an example of a customer where sales needed the on-boarding team to help convince the customer to sign by explaining how the technical integration would work. In other words, the on-boarding team supported the sales team in the fifth step of the sales process; *overcoming objections* (Moncrief and Marshall, 2005). This all results in the following proposition:

> Proposition *5: It is necessary to organise a deliberate collaboration between the on-boarding team and the sales team—including information sharing and cross-training—in order to increase the efficiency of the customer on-boarding process.*

Working with sales to increase the on-boarding team's understanding of customer expectations may also be a key factor, as supported by Partha and Roy's (2011) framework of operational considerations for a successful service delivery strategy.

Collecting customer insights

Something that was not prevalent in the case, but that is very much worth mentioning, is the opportunity of using the customer on-boarding process to systematically collect customer insights that can be used to inform the development of the startup company as a whole. One of the main themes in the Customer Development and Lean Startup philosophies is the systematic collection of customer insights, with the goal of learning what works and what does not work as quickly as possible (Maurya, 2010) (Blank 2007). The customer on-boarding process is arguably the first point at which the product is put to the test with real customers on an ongoing basis, meaning that it is a treasure trove of insights for the product development and overall management teams.

Conclusion

The aim of this study was to explore the field of customer on-boarding in B2B SaaS startups, and subsequently identify insights that may be suitable for further study. By doing a deep and open-ended analysis of the evolution of the customer on-boarding process at one such company we were able to create a list of propositions that can be refined and tested empirically:

- ***Proposition 1:*** *During a B2B SaaS startup's early growth phase, the customer on-boarding team should primarily be organised for flexibility and problem solving.*
- ***Proposition 2:*** *As a company transitions into the Company Building stage, the customer on-boarding process gradually transitions from experimentation to efficiency, and the experimentation is required to attain said efficiency.*
- ***Proposition 3:*** *It is essential for an organisation to actively consider the balance between efficiency in customer on-boarding and functionality in the product, and subsequently budget a certain amount of time and resources for building automated processes and internal support tools for customer on-boarding.*
- ***Proposition 4:*** *The customer on-boarding team should aim to involve all stakeholders early in a given on-boarding project in order to quickly identify potential issues and facilitate information flow.*
- ***Proposition 5:*** *It is necessary to organise a deliberate collaboration between the on-boarding team and the sales team—including information sharing and cross-training—in order to increase the efficiency of the customer on-boarding process.*

The results of this study are likely to be applicable to other B2B SaaS startups than Adco, especially as they take existing literature into account. Many insights may even be applicable to startups outside of B2B or SaaS. However, the boundaries between what is generalisable and what is specific needs to be tested in future research. Besides the five propositions, an important contribution of this paper is the raw telling of the case itself, which may allow others to draw conclusions besides those mentioned here. Hopefully this will lead to a better understanding of customer on-boarding in B2B SaaS startups, making such companies more scaleable, profitable and successful.

References

Ang, L. and Buttle F., 2009. *Customer development strategies for exceeding expectations - An exploratory study*. Database Marketing & Customer Strategy Management, 16(4), pp. 267-275.

Aoyama, M., 1997. Managing the concurrent development of large-scale software systems. *Int. J. Technology Management*, 14, pp.739-765.

Armour, P., 2011. The business of software testing: Failing to succeed. *Communications of the ACM*, 54(10), pp.30-31.

Bartels, A., 2013. *Global Tech Market Outlook 2013 To 2014*. Forrester Research [online] Available at <http://www.forrester.com/Global+Tech+Market+Outlook+2013+To+2014/fulltext/-/E-RES82921> [Accessed 24 March 2013].

Bell, C., 1969. *A note on participant observation*. SAGE Sociology, 3, pp.417-418.

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

Braganza, A., 2002. *Enterprise integration: Creating competitive capabilities*. Journal of Manufacturing Technology Management, 13(8), pp.562-572.

Butler, T. and Murphy, C., 2008. *An exploratory study on IS capabilities and assets in a small-to-medium software enterprise*. Journal of Information Technology, 23, pp. 330-344.

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

Chen, I. J. and Popovich K., 2003. *Understanding customer relationship management (CRM)*. Business Process Management Journal, 9(5), pp.672-688.

Cheverton, P., 2012. *Key Account Management: Tools and Techniques for Achieving Profitable Key Supplier Status*. 5th Ed. London: Kogan Page.

Choudhary, D. & Kumar, V., 2011. Software testing. *Journal of Computational Simulation and Modeling*, 1(1), pp.1-9.

Clark, A. et al., 2009. *Learning to see: lessons from a participatory observation research project in public spaces*. International Journal of Social Research Methodology, 12(4), pp.345-360.

Cooper, B. and Vlaskovits, P., 2010. *The Entrepreneur's Guide to Customer Development*. [e-book] Available at <http://custdev.com> [Accessed 30 December 2011].

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

Day, G., 2002. *Managing the market learning process*. The Journal of Business & Industrial Marketing, 17(4), pp.240-252.

Ezeh, P., 2003. *Integration and its challenges in participant observation*. SAGE Qualitative Research, 3(2), pp.191-205.

Flyvbjerg, B., 2006. *Five misunderstandings about case-study research*. Sage Qualitative Inquiry, 12(2), pp.219-245.

Juristo, N., Moreno A. & Strigel, W., 2006. Software testing practices in industry. *IEEE Software*, 23(4), pp.19-21.

Kumar, R. and Kumar, U., 2004. *A conceptual framework for the development of a service delivery strategy for industrial systems and products*. The Journal of Business & Industrial Marketing, 19(4/5), pp.310-319.

Lam, W. and Shankararaman, V., 2004. *An Enterprise Integration Methodology*. IT Professional, 6(2), pp.40-48.

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.

Maurya, A., 2010. *Running Lean: A systematic process for iterating your web application from Plan A to a plan that works*. [pdf] Available at <http://www.runningleanhq.com/runninglean_preview.pdf> [Accessed 13 November 2011].

Miller, R. & Collins C., 2001. Acceptance Testing. [online] Available at <http://www.christophertcollins.com/papers/Testing05.pdf> [Accessed 22 December 2012].

Moncrief, W. and Marshall, G., 2005. *The evolution of the seven steps of selling*. Industrial Marketing Management, 34, pp.13-22.

Nirpal, P. & Kale, K., 2011. An overview of software testing methodology. *International Journal of of Knowledge Engineering*, 2(1), pp.15-20.

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.

Ozanne, J. and Saatcioglу, B., 2008. *Participatory action research*. Journal of Consumer Research, 35(3), pp.423-439.

Partha, P. D. and Roy, R., 2011. *Operations strategy for the effective delivery of integrated industrial product-service offerings*. International Journal of Operations & Production Management, 31(5), pp. 579-603.

Pitney Bowes, 2010. *Next Generation Customer Onboarding* [pdf] Available at: <http://www.pbinsight.com/files/resource-library/resource-files/pbbi-next-gen-cust-on-boarding.pdf> [Accessed 14 January 2012].

Plouffe, C. and Barclay, D., 2007. *Salesperson navigation: The intraorganisational dimension of the sales role*. Industrial Marketing Management, 36, pp.528-539.

Poppendieck, M. & Cusumano, M., 2012. Lean software development: A tutorial. *IEEE Software*, 29(5), pp.26-32.

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

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

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

Rowley, J. and Haynes L., 2005. *Customer Relationship Management: The Matalan Way*. The Marketing Review, 5, pp.175-187.

Skok, D., 2010. *SaaS Metrics – A Guide to Measuring and Improving What Matters*. For Entrepreneurs [online] Available at <http://www.forentrepreneurs.com/saas-metrics/> [Accessed 15 January 2012].

Vanhaverbeke, W. and Torremans, H., 1999. *Organisational structure in process-based organisations*. Knowledge and Process Management, 6(1), pp.41-52.

Woodburn, D. and McDonald, M., 2011. *Key Account Management: The Definitive Guide*. 3rd Ed. Chichester: John Wiley & Sons.