UTRECHT UNIVERSITY

MASTER'S THESIS

Creating Customers: A Go-to-market Approach for Software Startups

Author:
Thomas ALFLEN
4176588

Supervisors:
dr. Slinger JANSEN
prof. dr. Sjaak BRINKKEMPER

A thesis submitted in fulfilment of the requirements for the degree of Master of Science

in

Business Informatics
Department of Information and Computing Sciences

August 8, 2019

UTRECHT UNIVERSITY

Abstract

Faculty of Science
Department of Information and Computing Sciences

Master of Science

Creating Customers: A Go-to-market Approach for Software Startups

by Thomas ALFLEN

Software startups fail more than 90% of the time, and it is frequently due to self-destruction instead of competition. Most startups cannot find a product-market fit and fail to recognise when a pivot is required. This research presents a go-to-market approach for software startups, derived through meta-modelling four integrative startup strategies. We evaluated the go-to-market approach with twelve innovative software startups are active in the Dutch market, three to ten years old, and operating in the final phase of the go-to-market approach. The go-to-market approach presents the crucial activities that are needed to find problem-solution fit and product-market fit. Explicitly formalising methods from strategies defined in literature and evaluating them via domain expert interviews in a multiple-case study, provides substantial scientific contributions. Furthermore, if every software startup would apply a predefined, structured approach, the problem-solution fit is found easier and earlier, which facilitates the product-market fit.

Acknowledgements

The thesis you are going to read is the result of eight months of strenuous but also gratifying work. I have gathered an abundance of knowledge during the execution of this research, and I am delighted with the result. During my time as a Business Informatics master's student, I imagined myself being an IT consultant in the near future. However, my interest in software startups was sparked when I founded a software startup with two other students: Oddity.ai. This experience changed everything for me; notably, it shifted my perspective of the near future.

I want to thank Slinger Jansen for providing me with the opportunity to study software startups and to guide me along the way; I am sure we will keep in touch. You were one of the most inspiring teachers in the Business Informatics Master, especially during the ICT Entrepreneurship course. I got the chance to make the acquaintance with twelve exceptionally interesting and motivating founders. To those founders: keep up the extraordinary work, and thank you for making this research possible. With every interview, I learned new things and got inspired for applying that knowledge in our startup.

My second supervisor, Sjaak Brinkkemper, was always there to provide me with an immediate answer to a (usually) method engineering related question. I want to thank him for being approachable for such, and other, questions. It was also a pleasure to discuss the progress our startups were making through the months. I am also positive that we will remain contact in the future, and I would like wish him the best of luck with his own startup.

A big thanks and token of appreciation would go to my girlfriend for supporting me throughout the thesis process. She brought significant support when needed. I am also grateful for the support of my parents, sister, and close friends. Finally, I would like to end with mentioning and thanking the co-founders of Oddity.ai: Nick and Gerwin. We had our fair share of fierce discussions, but in the end, we always supported each other. The journey we have made in such a short time is remarkable, and it is exciting to think about what the future might bring us.

Contents

Chapter 1

Introduction

Every day, software startups launch worldwide due to an increase of new markets, available venture capital, and accessible technologies (Crowne, 2002a; Smagalla, 2004). Software startups are defined as temporal human institutions that fabricate innovative products or services and, under severe uncertainty, search for sustainable and repeatable business models (Ries, 2011; Blank and Dorf, 2012; Blank, 2007; Bajwa et al., 2017). It is not uncommon that a software startup is built around an ingenious software product. Its creators are convinced that markets for the product exist or might be created if they are nonexistent. However, the future does not look bright: more than 90% of the startups fail (Marmer et al., 2011), and it is rather due to self-destruction than competition (Giardino, Wang, and Abrahamsson, 2014).

1.1 Problem Statement

> Worldwide, few software startups manage to endure the exposure of fierce competition, within an uncertain, rapidly evolving and time-pressured, startup environment. Most software startups tend to fail, especially when bringing a new product idea to the market. Therefore, more research about startup approaches and common pitfalls of startups is required.

Generally, two types of fits are essential to survive as a software startup (Blank, 2007). The first fit is the *problem-solution fit*, where the aim is to implement a solution that tests the trickiest hypotheses of the problem taken into consideration (Giardino, Wang, and Abrahamsson, 2014). The solution is based on a real customer problem that contains a particular need. If the solution holds after testing, the necessary product features need to be built. These features should solve the customer need, and if it does, there is a *product-market fit*.

In 2010, startups were providing the most new jobs in the United States (Kane, 2010) and there are known success stories in the past: *Facebook, Twitter, LinkedIn, Instagram, Dropbox,* and *Pinterest* to name a few. However, many startups fail even before bringing a product to the market, which is a substantial problem. Achieving to pursue such a go-to-market (GTM) approach with the appropriate fits is not easy, especially finding the right customers and providing them relevant products can be a challenging process. Usually, within two years from the startups' creation, it is not competition but rather self-destruction that drives the majority of startups into failure (Crowne, 2002a; Paternoster et al., 2014).

Strategy by Small-Medium Enterprises (SMEs) is defined as *"a fundamental pattern of present and planned activities, resource deployments and interactions of an organisation with markets, competitors and other environmental forces"* (Ian Burke and Jarratt,

2004, p. 129). To reach the goals defined by the startups' strategy, startups can apply a **go-to-market (GTM) approach**, which is a structured set of activities. Software startups are exposed to fierce competition and endure intense time-pressure from the market, within an uncertain context that is rapidly evolving (MacCormack, 2001). To achieve success in this environment, startups should choose the appropriate features to build, and should be able to adapt swiftly to new requests, while operating constrained due to limited resources (Sutton, 2000; Paternoster et al., 2014). Therefore, we hypothesise that software startups can benefit significantly from an effective startup approach. Therefore, we hypothesise that software startups can benefit significantly from a structured GTM approach.

1.1.1 Contributions

This research project provides societal as well as scientific contributions. Startups provide numerous jobs around the world and are being started daily, therefore having a significant *societal contribution*. Furthermore, the failure rate of startups is substantial, and research into this topic could potentially result in lowering this rate. Contemporary studies do provide potential startup strategies and insights about startup failures and pivots. However, this research will also provide *scientific contributions* because there is still a significant knowledge gap and these topics require more insight (Giardino, Wang, and Abrahamsson, 2014; Bajwa et al., 2017). Integrative methods are complementary for startup research: through the combination of conventional and less conventional methods, the chances of achieving problem-solution fit and product-market fit are higher.

1.2 Context

1.2.1 Software Startups

Software startups intend to grow fast and produce state-of-the-art products under the condition of severe business and technology uncertainty (Bajwa et al., 2016). They exhibit different feature constraints compared to established companies, reflecting both business and engineering concerns (Paternoster et al., 2014). Sutton (2000) presents a description of software startups, defined by their faced challenges (Giardino, Wang, and Abrahamsson, 2014):

- **Limited resources:** startups usually focus on taking the product out, promoting it, and establish strategic alliances.
- **Youth and immaturity:** startups have little accumulated experience in organisation management and development processes, compared to established companies.
- **Dynamic markets and technologies:** the novelty of software startups regularly require them to operate with disruptive technologies to gain access to a high-potential (possibly fast-growing) target market.
- **Multiple influences:** pressure from (both actual and potential) customers, investors, competitors, and partners impact the decision-making in a startup. Overall, these pressures might deteriorate clear decision-making.

The classification scheme of Paternoster et al. (2014) lists the most frequently reported contextual characteristics of a software startup. Furthermore, software startups operate with high flexibility, reactiveness, time pressure, and uncertain conditions. Numerous startups are dealing with multiple difficulties continually emerging from different directions.

1.2.2 Software Startup Strategies

The Lean Startup (Ries, 2011) provides a novel approach to innovation and entrepreneurship. It has become increasingly popular in the past decades, demonstrated by the conferences, global Lean Startup meet-ups, and the introduction as the main topic within entrepreneurship education programs (Khanna, Nguyen-Duc, and Wang, 2018). Therefore, the Lean Startup serves a suitable basis for constructing our method. However, The Lean Startup can be complemented by other startup strategies. For instance, as the study by Müller and Thoring (2012) shows, is that other strategies can complement the Lean Startup, which in this particular case was Design Thinking. Several more strategies building upon or lending similarities from the Lean Startup exist in literature and can be utilised for constructing a newly software startup approach, designed as a reference method.

1.3 Research Approach: Design Science

Within this research, a new object (*artefact*) is constructed to improve the problems as mentioned earlier (*context*). The artefact is iteratively *investigated* and *designed* using existing and newly gained knowledge. Therefore, this study is best characterised as *Design Science* (Wohlin and Aurum, 2015), which is proposed by Wieringa (2014). First, the aims and objectives of this research are presented using the template of Wieringa, whereafter the research questions and methods shall be presented.

1.3.1 Aims and objectives

To clarify the aims and objectives of technical research problems, we utilise the template of Wieringa (2014):

> This research aims to *construct* a reference go-to-market (GTM) approach for software startups to structure their process of growth *by* applying new theory and practical insights *that* software startups can use through their starting phase until their scale-up phase *in order to* provide aid to startups within difficult and insecure times.

1.3.2 Research Questions

We structure our main goal by providing the main research question (RQ):

RQ *How can a go-to-market approach be assembled for innovative software startups?*

The main RQ is divided into four subquestions (SQs). By answering these subquestions, we can answer our main RQ. The first SQ paves the way to investigate current startup strategies defined in the literature. Next, SQ2 evaluates the startup approaches identified in this research by discovering the method fragments from the proposed methods, based on the work of van de Weerd and Brinkkemper (2009). SQ3 provides an answer to how a go-to-market approach is assembled from the proposed method fragments. Lastly, with SQ4, we utilise insights gained from SQ1, SQ2, and SQ3 to present a proposed method model within case studies.

SQ1 *Which existing startup approaches are currently described in literature?*
Viewing some of the prominent startup approaches defined in literature provides a nutritious academic background. It became apparent that startup failures and pivots play a prominent role. Therefore, two subquestions are added with this subquestion that investigate the influence of failures and pivots within startups.

1.1 *What insights are gathered from startup failures?*

1.2 *What is the influence of pivots within startups?*

SQ2 *Which method fragments are identified from startup approaches?*
From the startup strategies listed in SQ1, method fragments can be identified: this subquestion answers how we should identify and split these method fragments. Furthermore, these fragments need to be compared using the method comparison method. The first subquestion will answer the differences and similarities between method fragments. The second subquestion ensures that we only extract relevant method fragments within this scope.

2.1 *How are relevant method fragments identified within the scope of this research?*

2.2 *How can multiple strategies be evaluated by method comparison?*

SQ3 *How can a go-to-market approach be assembled using method fragments?*
The relevant method fragments derived from SQ2 are used for the assembly of a go-to-market approach, in the form of a reference method (van de Weerd, de Weerd, and Brinkkemper, 2007). The reference method provides a complete overview of all possible processes and deliverables in a go-to-market startup process. This method shall be used as input for the case studies.

SQ4 *How are go-to-market approaches applied within case studies?*
From the previous subquestions, a new startup approach is derived, which can be evaluated through case studies at various startups. These startups can propose new insights or changes to the existing method.

Research Method	SQ1		SQ2		SQ3	SQ4
	1.1	*1.2*	*2.1*	*2.2*		
Literature Research	✓	✓				
Method Comparison (Method Engineering)			✓	✓		
Reference Method (Method Engineering)					✓	
Holistic Multiple-Case Study						✓

TABLE 1.1: The research methods that are used to answer the SQs.

1.3.3 Research Methods

To answer the subquestions, we use several research methods during the study. Table 1.1 shows the research methods that are used per subquestion. For SQ1 a thorough literature study is performed where several startup strategies are assessed. SQ1.1 and SQ1.2 are also answered during this process by evaluating failures and pivots in literature. After the candidate methods are known, we extract we phases with activities. From these activities, concepts derive. This process is further described in Section 2.1. If the relevant method fragments are known, SQ2 and 2.1 are answered. Next, these method fragments are compared using method comparison, based on the work of (van de Weerd and Brinkkemper, 2009), which answers SQ2, SQ2.1, and SQ2.2. With these method fragments, a new reference method is created (van de Weerd, de Weerd, and Brinkkemper, 2007), which answers SQ3. Both method comparison and the reference method is based on method engineering. Finally, the approach shall be evaluated in the form of a holisitic multiple-case study. While conducting interviews, we construct an answer for SQ4.

1.4 Thesis Structure

Figure 1.1 shows the remaining structure of the thesis. First, the research method is further explained in the following chapter. Next, the literature study provides the foundation of this thesis, consisting of existing software startup strategies, reason for failures, and types of pivots. This chapter also answers SQ1. After the foundation has been presented, the reference method is designed in the method design section, which answers SQ2 and SQ3. The model is evaluated within a multiple-case study, which provides an answer to SQ4. After that, we explain the lessons learned, which presents six key elements entrepreneurs should bear in mind. In the discussion we present the lessons for academia, the limitations that this study presents, and a roadmap for meta-modelling. Finally, the last chapter concludes the thesis and provides directions for future research.

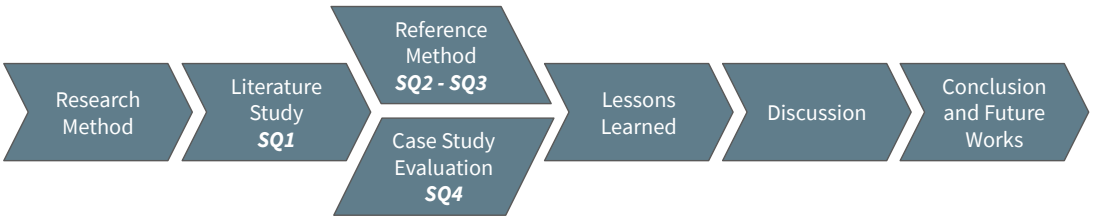


FIGURE 1.1: A visualisation of the thesis outline.

Chapter 2

Research Methods

As presented in Section 1.3, this research is divided into several parts. In the following chapter, a literature study shows four current driving strategies of software startups. That will serve as a theoretical background for the go-to-market approach that is constructed. Four general strategies are discussed, from which approaches are modelled. The related literature will also explain the importance of pivots, and why software startups should recognise and embrace failures. After the literature review, methods fragments are extracted from current software startup strategies. These method fragments are then compared to identify similarities and dissimilarities, and are used as input for the GTM approach. The software startup approach is used as input for the case study, which serves as an instrument during twelve interviews. Finally, this chapter elaborates upon the choices of the research method. In Section 2.1, a description of Process-Deliverable Diagrams is provided, where we also explain the foundations and method comparison. Next, an extensive description of a multiple-case study is provided in Section 2.2. In particular: the definition of a case study, how the data is collected, the case study protocol, and the threats to validity. In Section 2.3 we provide startup inclusion criteria, in Section 2.4 we explain our data analysis protocol, and, lastly, in Section 2.5 we present the research risks of this study.

2.1 Process-Deliverable Diagram

Throughout this study, Process-Deliverable Diagrams (PDDs) are used as meta-models. This section serves as a brief introduction of the modelling method that is introduced by van de Weerd and Brinkkemper (2009). A PDD consists of two parts: processes on the left-hand side, which is based on a UML activity diagram, and deliverables on the right-hand side, which is based on a UML class diagram. In the process-side of the PDD there reside two elements: *activities*, which are either complex or standard, and *transitions* that connect the activities, potentially with conditions. Complex activities contain subactivities and can either be open or closed. If they are open, the subactivities are either shown in the same model or another. When they are closed, the activities are not elaborated upon, either because they are irrelevant or unknown. The deliverable side of the PDD shows the *concepts*, which are again complex or standard. For the complex concepts, the same rule is applied as the activities. Concepts do not have transitions, but they are connected in either of the following *relationships*: association, generalisation, or aggregation. An activity always leads to either a concept, or a *property* of that concept. A visualisation of the PDD elements is visualised in Figure 2.1.

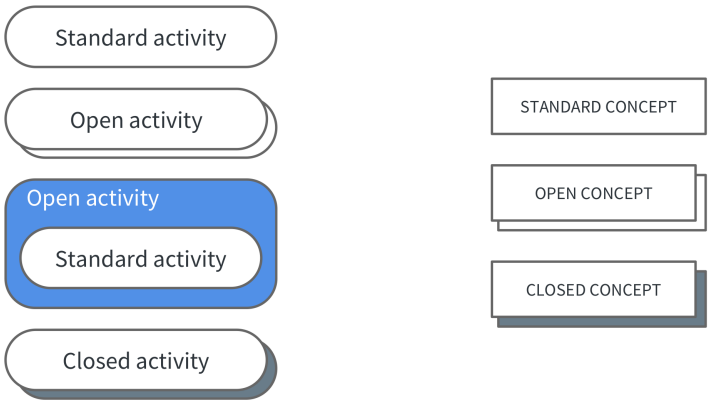


FIGURE 2.1: Key elements of a PDD: displayed are the activity types (left) and concept types (right), adapted from van de Weerd and Brinkkemper (2009).

2.1.1 Method Comparison

The startup approaches defined in literature are modelled as PDDs. One of the advantages of meta-models is that they enable comparison via a method comparison matrix. To accurately and extensively compare the approaches, we use the formal tabular comparison method and notation of Hong, Goor, and Brinkkemper (1993, p. 6). A reference method can be modelled after comparing various approaches, which serves as a complete overview of possible processes and deliverables in a GTM approach. By developing a reference method, similarities and differences between methods are identified in a systematic way (van de Weerd, de Weerd, and Brinkkemper, 2007).

Methods usually consist of phases, which translate to open activities. First, a comparison table is created for the phases, which serves as a high-level overview of the methods' phases. Next, a detailed meta-method is created for each method in the literature. For comparison, the formal notation proposed by Hong, Goor, and Brinkkemper (1993, p. 6) is used, where a *comparison indicator* compares an activity **s** from one method, with an activity **m** from another method:

- **s = m** The activity s is equivalent to the activity m.
- **s > m** The activity s does more than the activity m.
- **s < m** The activity s does less than the activity m.
- **s >< m** A part of the activity s overlaps a part of the activity m, and the other parts of both activities do not overlap.

2.1.2 Method Fragments

During an *activity*, an *action* executes, and when one action finishes a new activity initiates with another action. One specific action within an approach corresponds to an activity. If an action is to be performed within another action, it is classified as a *sub-activity*, thus the *super-activity* is an open activity. In this research, the latter are defined as *phases*.

Method fragments are elements of the PDD. First, we formalise the identification of fragments, whereafter we will provide examples of method fragments. The UML activity diagram (left-hand side) and UML class diagram (right-hand side) are both based on the work of the Object Management Group, which we use for the formalisation (OMG, 2003). During an *activity*, an *action* executes, and when one action finishes, due to objects and data (*concepts*) becoming available, a new activity initiates with another action. Each action in an activity either executes zero, one, or more times when the activity is executed. At the minimum, actions require access to data, and the ability to test and transform data. Therefore, one specific action within an approach corresponds to an activity. If an action is to be performed within another action, it is classified as a sub-activity, thus the super-activity is an open activity. Within this research, the largest activities are also defined as phases. Furthermore, following van de Weerd et al. (2006), when an order of activities is absent, they are presented as unordered activities. This is indicated through the absence of connectors between the (sub)activities. Next, we provide an example method fragment by translating a small part of the Design Thinking (Müller and Thoring, 2012) approach. Figure 2.2 shows the mapping of two phases of Design Thinking: *Understanding the Problem* and *Create a Point Of View*. An activity with more than one specific action is an open activity, which is the case with these aforementioned phases.

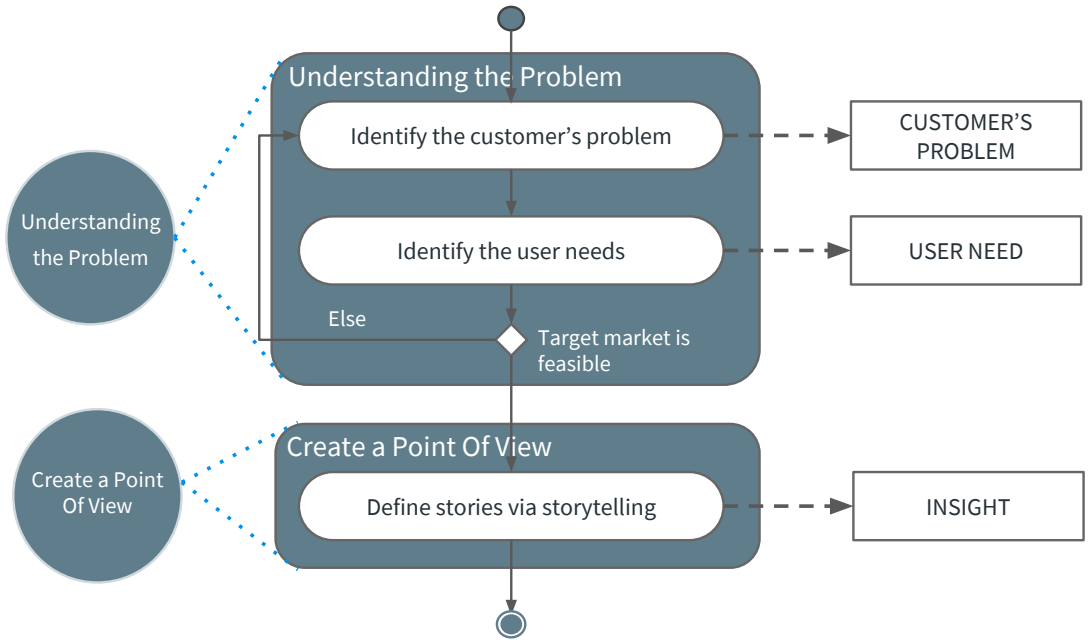


FIGURE 2.2: An example of method fragments that are mapped from two (incomplete) Design Thinking phases (Müller and Thoring, 2012).

Following the research of Müller and Thoring (2012), the full approach of Design Thinking is sequential. Therefore, there are no unordered (sub)activities in the PDD, which is also shown in the partial PDD of Figue 2.2. It also visualises the two sequential subactivities within the phases, which are standard activities (they contain no subactivies with other actions): *Identify the customer's problem* and *Identify the user needs*. The dashed arrow after the first subactivity pointing to the right shows the standard concept: *CUSTOMER'S PROBLEM*. Following our formalisation description, this concept contains data, namely about the customer's problem.

After the second subactivity, a gateway presents the consideration about the target market being feasible. If not, another problem should be uncovered by the startup that resides in an improved target market. If the target market is estimated as adequate, we arrive at a new open activity: *Create a Point of View*. This phase contains another subactivity: *Define stories via storytelling*. The example of Figure 2.2 represents several method fragments: two open activities/phases, three subactivities, one gateway, and three concepts. The actual PDD contains more phases with additional subactivities, as described in Section 3.1.

2.2 Case Study

To assess whether our reference approach is suitable for software startups, we evaluate it with twelve software startups, in the form of a case study. Runeson and Höst (2009, p. 134) define a case study as:

> **Definition 1 — Case study.** Investigating contemporary phenomena in their context, especially when the boundary between the phenomenon and its context is unclear, gathering information from few entities with lack of experimental control.

Furthermore, Runeson and Höst (2009, p. 137) define three critical characteristics of a case study:

1. it is of flexible type, coping with the complex and dynamic characteristics of real-world phenomena, like software engineering,
2. its conclusions are based on a clear chain of evidence, whether qualitative or quantitative, collected from multiple sources in a planned and consistent manner, and
3. it adds to existing knowledge by being based on previously established theory, if such exist, or by building theory.

A case study provides an approach which allows a flexible boundary between the analysed object (the GTM approach), and its environment. Five iterative steps are essential when conducting a case study: (**i**) Case study design, (**ii**) preparation for data collection, (**iii**) collecting evidence, (**iv**) analysis of collected data, and (**v**) reporting (Runeson and Höst, 2009). Steps **(i)** and **(ii)** are defined before conducting the case study, steps **(iii)** and **(iv)** can be conducted incrementally. Following the work of Yin (2011), a case study can either be labeled as a single- or multiple-case study. In this research, we perform a *holistic multiple-case study* because there is a call for multiple cases, without embedded subcases: conditions are tested under which the same findings might be replicated (Yin, 2011). This type of case study is difficult to implement but the resulting data will provide greater confidence in the findings (Yin, 2011). During the case study, we work with first-degree data collection techniques: data is collected in real-time, and we will be in direct contact with the subjects (Lethbridge, Sim, and Singer, 2005). The data collection method that is used within this study is conducting semi-structured interviews. With interviews, the main strengths are that the focus is directly on the object of study and they are insightful, providing explanations and causal inferences. Poor recall of the interviewer is countered by recording and transcribing the interviews. During the research, we watch out for bias in the response of the interviewee or poorly articulated questions.

2.2.1 Protocol

A research protocol, visualised in Figure 2.3, is used to ensure the reliability of the case study. This contains some key elements: the overview of the case study, field procedures, interview questions, and a case study report. It is reused for each case that is conducted, which ensures a consistent and objective manner of data measuring.

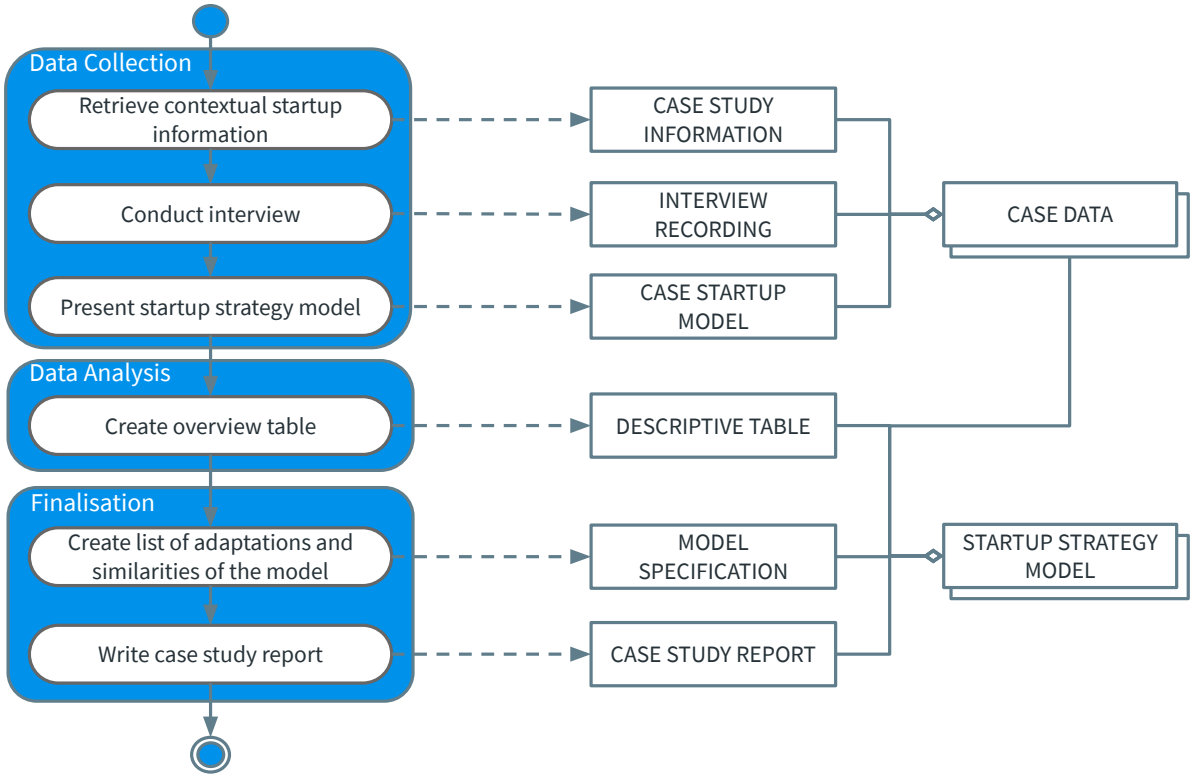


FIGURE 2.3: A PDD visualising the case study protocol in three phases: Data collection, Data analysis, and Finalisation.

The protocol shows three phases: Data collection, Data analysis, and Finalisation. During the first phase, we gather the data of the case, by retrieving contextual startup information. We provide the startup approach to the interviewee during the interview. The case participant can provide comments about the model; the interviewer connects interview results with activities in the model. Together, these elements will be part of the case data. Next, the data analysis phase ensures a descriptive table containing each case. After that, a list is created that contains the adaptations and similarities of the model and the startup case. Finally, a case study report is written, and an evaluated GTM approach is modelled.
Appendix A shows the interview protocol that is followed during the interviews. It is crucial that the questions reflect the approaches that are discussed in Chapter 3 and Chapter 4. The general strategy is to ask the interviewee about the experiences when starting the software company, until where we are now. Furthermore, it is vital to discover what decisions led to their current situation, and what approaches the software startup has taken in that time. After finishing the interview questions, the model shall be shown to the interviewee to see if there are any similarities and dissimilarities between the approach they took. The interviewee can evaluate the model verbally and add, or delete, method fragments.

The data that is put on the printed model is manually assessed afterwards, per case. However, the additional interview data is transcribed and processed in NVivo 11, which is a qualitative data analysis computer software package. Every interview result is derived from the same interview protocol, so the consistency is approximately the same. However, Nvivo allows coding the interviews on important topics for easy reference and improved consistency. These topics are called *nodes*, which together are the *node tree*, which is visualised in Figure 2.4 below:

FIGURE 2.4: The Nvivo node tree that is used for the multiple-case study, consisting of several nodes (topics).

The two main topics are the problem-solution fit and product-market fit, which are explained further in Chapter 3. Concerning the former, the subtopics are identifying the customer problem, acquiring the first customer, developing a solution (Minimum Viable Product), and validating with additional customers. Regarding the product-market fit, it consists of the subtopics converting actual customers, and further developing the solution. Other coded topics are about investments, pivots, employed strategy, and the perceived usability of the approach. Lastly, when other interesting results are found that do not fit these topics, it is coded as other findings. Chapter 5 shows the results of the case studies, which are substantiated with quotes that derive from one of these topics.

2.2.2 Validity

Wohlin et al. (2012) describe four threats to validity. However, in this research the similar scheme of Yin (2017) is used. Runeson and Höst (2009) and Yin (2017) discuss the following validity threats: construct validity, internal validity, external validity, and reliability. The **construct validity** is concerned with the degree to which a test measures what it claims to be measuring. The use of PDDs as a test measure has proven to be effective in multiple researches. However, this is particularly challenging when performing a case study, because the operational measures that are studied during the case, should represent what is investigated following the research questions, which is sometimes difficult. There are several methods to increase the construct validity, where the first strategy is to use multiple sources (cases) of evidence. The second tactic is to establish a chain of evidence: the reader of this study

can follow the derivation of every piece of evidence, from initial research questions, to eventual case study conclusions. The case study protocol contains the interview questions and specific evidentiary sources are cited in the case study report, which results in a clear traceability. With multiple sources of evidence and a clear chain of evidence, the construct validity of this study increases.

The **Internal validity** is of importance when causal relations are examined. This causal relationship should be established correctly; if a factor affects an investigated factor, there should not be a third factor affecting the factor. It could also occur that the the researcher is unaware of the third factor, or that he does not know how much the investigated factor is affected by the third factor. Case participants are treated similar during the case study, and the case protocol is reused for every case participant. Because of the case participants' lack of knowledge regarding PDDs, the use of such models could threaten the internal validity of this research. However, since we mainly focus on the process diagram, it is hypothesised that the case participant has sufficient knowledge to make an assessment about the model. Furthermore, the model is briefly explained beforehand and any questions are answered by the researchers. We also try to increase the internal validity by interviewing in a secured space.

The **External validity** deals with the possibility of generalising the findings, thus relating to the generalisability. The final go-to-market should be applicable for and relevant to other software startups as well. Specifically, software startups that follow the inclusion criteria listed in the next section. However, a common criticism of case studies is the absence of generalisability, usually because a relatively small sample size is involved (Dul and Hak, 2007; Yin, 2017). A case study should be replicated to increase the confidence of the study. In this research, we extensively examine twelve software startups, while following a case study protocol that is reused for each case. We expect these cases to produce homogeneous results because they are of a similar nature, which relates to literal replication. By applying a case study protocol and having a relatively large (homogeneous) sample size, we attempt to increase the external validity (generalisability) of this research. Furthermore, in this research no statistically representative sample is drawn from a population. It deals with *analytical* generalisation: the results are extended to cases with common characteristics.

Finally, the **Reliability** explains to what extent the analysis and the data are dependent on the researcher. We aim to minimise the biases and errors in this study; if another researcher conducts the same study (following the same procedures), the results should be the same. One prerequisite is that the followed procedures should be documented if another researcher conducts the same study. The method engineering process is fully documented in this research, which should result in similar models. The case study protocol is also clearly documented and reused for every software startup. The results of the case studies are transcribed and saved, which increases the reliability. Documenting the procedures and maintaining a clear case study protocol will help to mitigate the threats that come with this aspect of validity.

2.3 Case Study Inclusion Criteria

The sample software startups that is used for this research is gathered through convenience sapling (Morse, 2010), which involves drawing samples that are both willing to participate in a study and easily accessible (Teddlie and Yu, 2007). However, not every software startup will be included as a potential case. This subsection

will provide guidelines and traceability in the selection process of software startups. Studies already provided some requirements as to what qualifies as a software startup, for instance, Ripsas and Tröger (2014, p. 4) use the following requirements that build upon the theory of Blank and Dorf (2012):

- A startup is a young company that is less than ten years old.
- A startup has an innovative business model, deploys innovative technologies, or both.
- A startup shows significant growth either in the number of employees, or in turnover.

To be more specific, Ripsas and Tröger (2014) state that a startup requires to qualify for the first criterion (its age) and one of the last two criteria; having either an innovative business model or technology, or a sign of significant growth. The work of Jansen and van Cann (2012) viewed success stories of Dutch software startups. They worked with the following startup inclusion criteria for their work (Jansen and van Cann, 2012, p. 2)::

- Startups should be Dutch and need to be active in the Dutch market.
- Startups should be at least five years old since this is generally the accepted length of the start-up period of a company (Busenitz, 1999; Nowak and Grantham, 2000; Schutjens and Wever, 2000). Surviving these first five years can be seen as critical for a company to become successful.
- Startups should have at least 50 employees to ensure that we attract the more successful companies. This is also commonly, both by scientific literature as well as practice, seen as the cutoff point between small and medium-sized companies. For instance, see European Commission (2003).
- Startups should add value to their customers, which in practice means we focus on product software for business environments.

On purpose, one condition is omitted, namely: *"They should have been profitable for a number of years."* The main reason for this exclusion is that there are numerous examples of software startups that are on paper not profitable, but still prosperous For instance, Uber (Somerville, 2018), or Twitter, which recently had its first profitable quarter since 12 years (Wagner, 2018). With these lists as input, we create the inclusion list relevant for this research. The conditions can be mandatory (+) or optional (-), where startups should qualify for at least one optional condition.

+ Startups should be Dutch and need to be active in the Dutch market.
+ Startups should be at least three years old, but less than ten years old.
+ Startups require to be in the final phase of the go-to-market approach, which is further described in Chapter 3 and 4.
- A startup has an innovative business model and/or deploys innovative technologies.
- A startup shows significant growth either in the number of employees or in turnover.

The minimum of three years instead of five ensures some more flexibility concerning the inclusion of startups. We also verify if the startup fits the description as mentioned in Section 1.2. Lastly, the final two phases of the GTM approach serve as indicator that the product-market fit has been found, thus a startup should recognise itself being in those phases.

2.4 Data Analysis Protocol

After the case studies have been performed, the data of the cases are analysed within- and cross-case. The records of the interviews are transcribed, and afterwards analysed with the NVivo tool (NVivo, n.d.). By using NVivo, the answers are categorised on the question topics and the GTM approach, to ensure qualitative rigor. First, summarising company data about the sample is presented, for instance showing per case the amount of employees working at that startup. Every interviewee proceeds through the complete GTM approach, and verifies whether the startup took similar steps during their growth. The steps that have and have not been followed by the startup are recorded and analysed. We track this as aggregated data and use this as input for the evaluated GTM approach. It may be the case that several startups state that they did not follow a certain phase, but one other startup did. If that particular company showed a fair reason or justification to contain that activity (which is then quoted), it is not deleted from the final GTM approach. Thus, rationalisation is of importance in this aspect and is used throughout the analysis. Finally, the case study data is presented per phase of the GTM approach, each section providing a short explanation about the activities within the phase. Furthermore, a visualisation, explanation, and occasionally rationalisation are presented, about the activities that are included and excluded. These explanations are substantiated with quotes derived from the interview. The 'new' post-case study GTM approach, with adaptations derived from the case studies, is also presented.

2.5 Research Risks

This research is exposed to certain risks that should be mitigated as thoroughly as possible. The process of performing a case study bears validation risks, which were mentioned in Section 2.2. Below, three research risks are also presented:

- **Insufficient case participants**: It may occur that we do not find enough case study participants for evaluation within the time this research is performed. To mitigate this threat, we actively search within our network, send regular emails, and contact incubators nearby. If too little startups have provided their input, future works will include that additional evaluation is necessary.
- **No unique approach can be extracted**: The approaches pursued in the case studies may be completely dissimilar from each other, from which no unique approach can be extracted. If this is the case, a reference method can still be created, but it requires additional future evaluation.
- **Startups do not follow structured approaches**: There is a possibility that case study participants did not follow a structured approach while building their startup. If this is true, we attempt to extract their choices and strategies, and assess whether a structured reference method could still porvide support to such unstructured startups.

Chapter 3

Background and Related Work

Startups tend to fail because they spend too much money and time, creating the wrong product until realising, too late, what the product actually should have been (Nobel, 2011). One general strategy that takes this problem into account is the Lean Startup, a method that has proven to be successful for numerous young software companies. Therefore, the Lean Startup shall be treated as the base of our method within this study. This section presents background literature and related work concerning software startup strategies. First, several startup strategies are discussed in Section 3.1, from which approaches can be derived. Next, Section 3.2 explains more about the problem-solution fit and product market fit. Section 3.3 presents the importance of failures within software startups, and section 3.4 explains the significance of pivots.

3.1 Software Startup Strategies

This subsection introduces four *user-driven innovation strategies*: they involve potential users, customers, or other stakeholders into the development process, thus maintaining a user-centred approach (Müller and Thoring, 2012). Users comprise a source of inspiration that stimulates innovation; they are a resource in the innovation process (Holmquist, 2004). The following subsections are divided in The Lean Startup (3.1.1), Customer Development (3.1.2), Design Thinking (3.1.3), The Lean Product Playbook (3.1.4), and The Startup Owner's Manual (3.1.5). In the final subsection, software startup evaluation dimensions are introduced.

3.1.1 The Lean Startup

The Lean Startup approach is inspired by the lean principles, which are developed by Toyota manufacturing and production system in Japan (named lean manufacturing), to enhance optimisation in the production processes (Ries, 2011; Womack and Jones, 1997). Essentially, the approach consists of abolishing redundant waste in the company processes during the development phase. The definition of waste is moderately different in this context: the 'biggest' waste is developing a product or service that nobody requires (Müller and Thoring, 2012). Therefore, the entrepreneurs are obliged from day one to *get out of the building* to obtain early feedback and user needs (Blank, 2007). Getting out of the building is important because *"in a startup, no facts exist inside the building, only opinions"* (Blank, 2007, p. 9). When developing an idea within a Lean Startup, it goes through the Build-Measure-Learn (BML) loops. A startup develops an idea, gathers and measures data concerning the customer response, and learns about the outcomes. Subsequently, the loop starts again with building upon the result that has been gathered. Figure 3.1 visualises this loop.

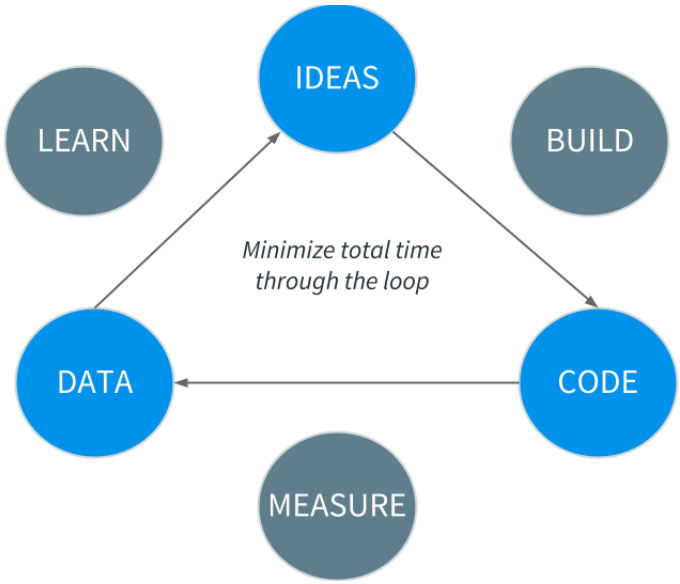


FIGURE 3.1: The Build-Measure-Learn (BML) loop. It starts with an idea, which is further built (coded), data is gathered (measured), and outcomes are learned from.

This loop can be realised through creating a *Minimum Viable Product* (MVP), which is a version of the product that requires the least amount of development time with a minimum amount of effort (Ries, 2011). In essence, this relates to validating a business model through hypotheses, and making a decision about the outcomes: persevere or pivot (Bajwa et al., 2017). This makes the Lean Startup a hypothesis-driven entrepreneurship (Eisenmann, Ries, and Dillard, 2012), which relates to *validated learning*: an empirical procedure of measuring progress concerning present and future business prospects (Ries, 2011). Several studies touch upon the success of the Lean Startup, also when implemented in established firms (Furr and Paul, 2011; Furr and Dyer, 2014; Rasmussen and Tanev, 2015).

3.1.2 Customer Development

Building upon the Lean Startup framework from Blank (2007), Cooper and Vlaskovits (2010) present an objective, straightforward case study, and denominate it Customer Development. It can be useful for any kind of company that desires to launch a new product, which makes it appropriate for software startups to utilise. The framework proceeds through four steps, or phases, to discover and validate if a startup has identified the appropriate market for its product, created the necessary product features that translate to customers' needs, tested methods for acquiring customers, and deployed the correct resources for scaling the startup. Similar to the Lean Startup, it requires entrepreneurs to question their core business by formulating hypotheses about it. The four main phases are described as follows (Cooper and Vlaskovits, 2010):

- **Customer Discovery:** A product should solve a user problem for an identifiable group. The problem-solution fit, proposed MVP, and funnels are of importance during this phase. This phase focuses on the problem-solution fit.

- **Customer Validation:** The market is assessed on its potential, because a feasible business should be built where enough sales can be made. The product-market fit comes into play, together with the business model, and the sales and marketing road map. At the end of this phase, the startup can decide if pivoting is required (thus returning to Customer Discovery).
- **Company Creation:** A repeatable sales and marketing road map should guide the startup to a scalable business. During the Company Creation phase, the upscaling of the company execution is the main priority. The product-market fit should have been found by now.
- **Company Building:** Operational processes and company departments are created to support scaling. Similar to Company Creation, Company Building attempts upscaling the whole organisation.

Within each phase, several activities exist to achieve the outcomes described above. Furthermore, the first two phases focus heavily on searching and validating customers, while the latter two phases focus on executing the business plan. Several recent studies report on the importance of Customer Development within startups (Stampfl, 2015; Järvi, Taajamaa, and Hyrynsalmi, 2015; Ripsas, Schaper, and Tröger, 2018).

3.1.3 Design Thinking

Similar to the Lean Startup, the Design Thinking strategy is also based on a user-centred approach. In the late 90s, design consultancy firm IDEO developed Design Thinking. IDEO was shifting its focus to designing consumer experiences instead of traditional consumer products, due to the requests they obtained (Brown and Wyatt, 2010). When asked to explain what designers do, David Kelley usually answered "thinking", which is where the term Design Thinking derived from. The approach does not refer to Lean principles, however, the main idea behind it is alike: it creates appropriate solutions by attempting to identifying user needs (Müller and Thoring, 2012). The Design Thinking strategy makes use of iteration cycles, feedback loops, and extensive user research, with three main (iterative) phases: inspiration, ideation, and implementation. The inspiration phase tackles the opportunity or problem and initiates the search for solutions. Ideas are generated, developed and tested within the ideation phase and the implementation phase bridges the project to actual consumer products (Brown and Wyatt, 2010). Related to the three phases, Plattner, Meinel, and Weinberg (2009) provided an abstract model of the activities in Design Thinking, which is adapted and visualised in Figure 3.2. The connections between the activities show that Design Thinking is highly iterative. Design Thinking can be applied in many domains and multiple studies emphasise its success (Plattner, Meinel, and Weinberg, 2009; Martin and Martin, 2009; Dorst, 2011). Interestingly, Müller and Thoring (2012) analysed and compared the Lean Startup and Design Thinking. The goal was to uncover any potentials to enrich either strategy, by adapting or merging certain elements. Müller and Thoring effectively combine the strategies, essentially showing that Design Thinking can effectively precede the Lean Startup.

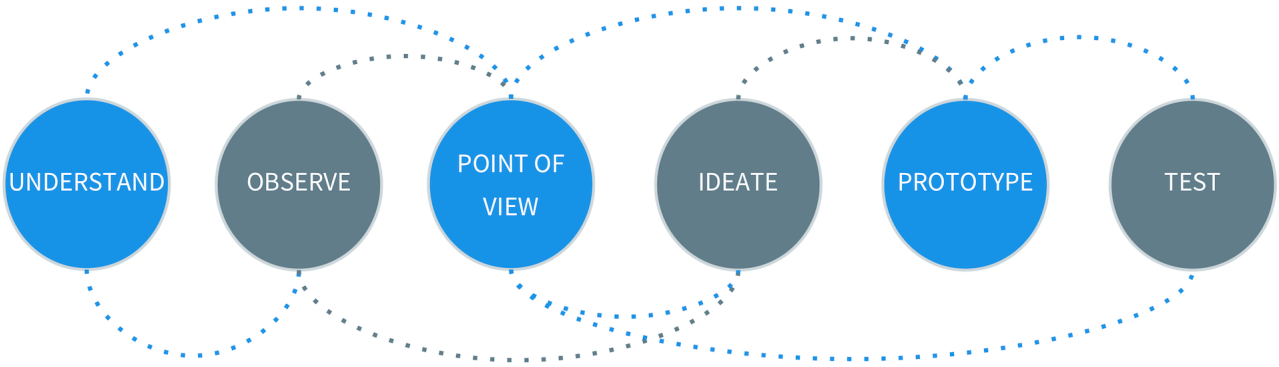


FIGURE 3.2: Abstract model representing Design Thinking. Proceeds through six activities in an iterative manner, which is adapted from Plattner, Meinel, and Weinberg (2009).

3.1.4 The Lean Product Playbook

The Lean Product Playbook by Olsen (2015), builds upon the Lean Startup. It contains a six-phase framework called the Lean Product Process that aims to bring the startup on a level where it has created a product that provides significant customer value. Figure 3.3 visualises this framework, which should be interpreted bottom-up. Similar to the BML loop of the Lean Startup, it follows a hypothesis-based loop: the Hypothesise-Design-Test-Learn cycle. Notably, the product-market fit should be established by the startup after two phases. The fit has to be found before proceeding to the third phase. Within the phases, several key activities should be performed (Olsen, 2015):

- **Determine target customer:** Problems within identifiable groups are discovered because in this group resides the startups' target customer. The groups should be segmented into market segments containing potential customers. Based on this information, the product is refined and tweaked if necessary.
- **Identify underserved customer needs:** The target customer should have underserved needs that the startup requires to fulfil with its product. These needs should also be feasible for the startup. Specific needs are identified that potentially represent a good market opportunity.
- **Define value proposition:** If problem-solution fit is achieved, a value proposition, a plan, is defined, which outlines the essence of the product strategy. The value proposition describes the unique features and why the product serves customer needs better than alternatives.
- **Specify MVP feature set:** In the next phase, an MVP is built that only consists of the required unique features, but not more. During this phase, the MVP is validated with target customers and this is iterated until the MVP properly serves the customers' needs.
- **MVP prototype:** When the necessary features are clear, a prototype is built only consisting of those features. This is then utilised for customer tests.

- **Test with customers:** The prototype is tested with customers from the target market and feedback is integrated within the product and if necessary, a startup can start the process over again (potentially pivoting).

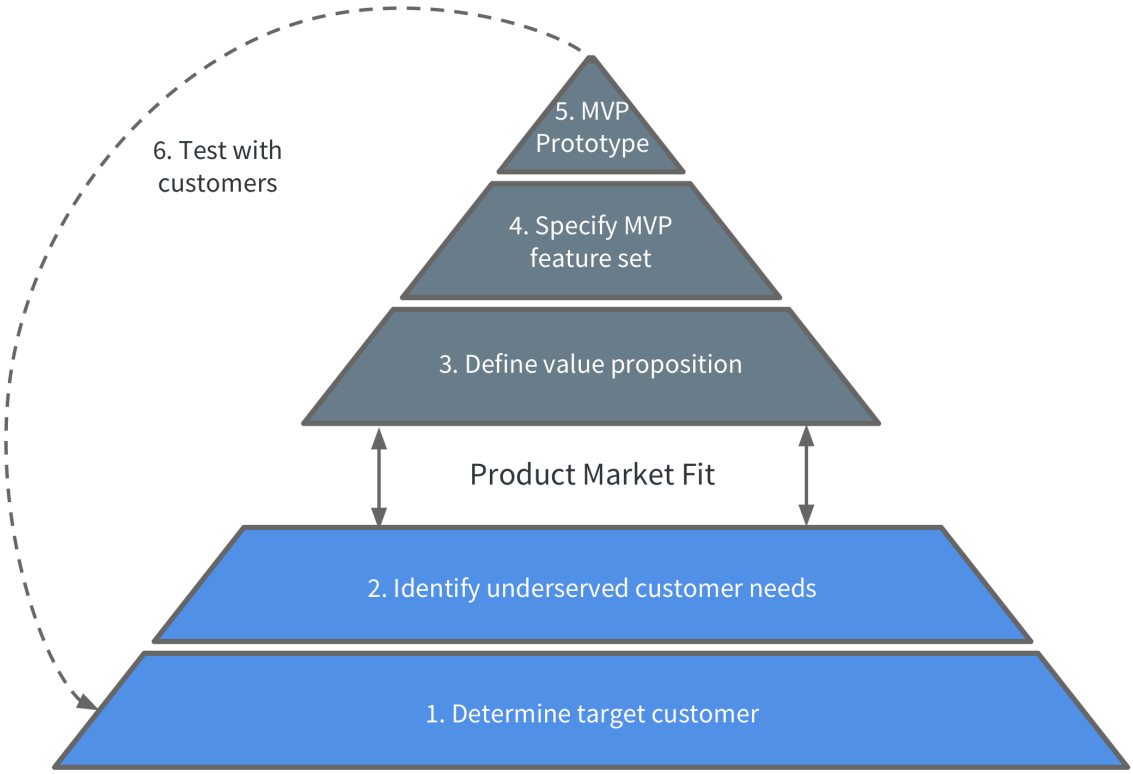


FIGURE 3.3: The Lean Product Process from the Lean Product Playbook, consisting of six phases, adapted from Olsen (2015)

The Lean Product Playbook is not cited often, which could be due to the fact that it is a relatively new strategy, or that it largely builds upon the Lean Startup strategy. It is therefore interesting to include this strategy in the GTM approach.

3.1.5 The Startup Owner's Manual

The Startup Owner's Manual is designed by Blank and Dorf (2012) and follows the same phases discussed in Customer Development (Subsection 3.1.2): Customer Discovery, Customer Validation, Company Creation, and Company Building. However, the content of the book focuses mainly on the first two phases, and thus on finding the appropriate target customers for the startups' MVP. It depicts a step-by-step guide to building a profitable and scalable startup in an approximate time frame of 30 months. During the first phase, it presents the importance of discovering customers on a hypothesis-basis, and creating a business model canvas (or lean model canvas (Maurya, 2012b). After this, the Customer Validation phase starts. Essentially, it builds upon the Four Steps to the epiphany (Blank, 2007), providing additional practical tips, guidelines, and checklists for startups to pursue. Therefore, this strategy is an interesting candidate for the GTM approach, combined with the fact that it is not often referenced to in literature.

3.1.6 Software Startup Evaluation Dimensions

We explained the definition, characterisation, and GTM approaches of software startups. Nonetheless, startups experience a dynamic and unpredictable context, and, when resources are scarce, success and survival depend heavily on the people in charge (Giardino, Wang, and Abrahamsson, 2014). The direction of the startups might shift continually, but some startup dimensions remain crucial. MacMillan, Zemann, and Subbanarasimha (1987) studied startup contexts, taking into account four holistic dimensions, the **Team** being the core element. Having the competent people aboard in a software startup is key (Giardino, Wang, and Abrahamsson, 2014) and the type of leadership within a startup can define its success (Ensley, Hmieleski, and Pearce, 2006). Furthermore, Giardino et al. present that passionate behaviour is an important aspect within a team; without passion, barriers are hard to overcome and often used as an excuse for failure. Generally, a software startup aims to build an innovative **Product** (Sutton, 2000) for a known or unknown **Market**, where the product should be destined for a market where customer needs are clearly identified. Lastly, the **Business** sets the startups' growth and its place in the market (Yu et al., 2012; Giardino, Wang, and Abrahamsson, 2014).

3.2 Finding the Fit

Now that it is clear what defines a software startup, what constraints come with it, and what strategies it might follow, the previously mentioned 'fits' are further discussed. As Marc Andreessen puts it: *"The life of any startup can be divided into two parts — before product-market fit and after product-market fit."* The *problem-solution fit* ensures the startup finds and creates a solution to an existing customer problem. With a problem comes a certain customer need that is potentially undiscovered. Discovering and validating that need to achieve and demonstrate a problem-solution fit, is more viable than an unvalidated idea (Hui, 2013; Giardino, Wang, and Abrahamsson, 2014). Moreover, wrongly focusing on perfecting a business model, and attempting to obtain the first paying customer without clearly identifying the target market with its customer needs, can lead to failure. Giardino, Wang, and Abrahamsson (2014) presented through two failed project cases (startups), that an immature problem-solution fit potentially leads to failure. The target customer should be actively involved with the development process, to activate the learning progress. When the problem-solution fit is found, the next step is to find the *product-market fit*: build the unique features (as an MVP) that meets the customers' needs. It can be necessary to *pivot* when the product-market fit is not found. The company can reiterate to the problem-solution fit and try to find new hypotheses that can be tested, which is important: *"Winners recognise their startup is a series of untested hypotheses."* (Blank and Dorf, 2012, p. 38). Pivots can happen at any stage of a software startup and there are multiple triggers and types of pivots, which are further discussed in Section (3.4).

3.3 Embracing Failure in Software Startups

Failure is a reoccurring and important topic within startups; comprehending it is paramount for survival. Software projects in startups tend to fail, which is similar to software projects in established companies (Savolainen, Ahonen, and Richardson, 2012). The major difference is that a failed project from a startup can result in severe consequences for the startup. One project failure can put a startup out of business

because the majority of startups are engaged in a single project at a time (Giardino, Wang, and Abrahamsson, 2014). As described in the previous Section 3.2, many startups neglect or overlook the problem-solution fit or product-market fit, which potentially leads to failure due to lack of customers buying the product. This is confirmed by Blank (2007): few startups fail due to missing technology, but they do struggle to find the customers. A product can be extremely innovative and potentially enter novel markets, but bears no proof of functionality in the real world, which makes it more vulnerable to failure (Giardino, Wang, and Abrahamsson, 2014). Therefore, early customer feedback is crucial for preventing such failures, or undergoing the failure in an early stage (and pivot if necessary). The concept validated learning comes into play here: startups often ignore the validated learning process and circumvent pivoting when required, which leads to a large number of startup failures (Bajwa et al., 2016). Learning from failures is crucial for survival, they should be embraced and occur early and cheap, so final catastrophes are avoided (Ries, 2011). It is difficult to assess when a project is deemed a failure, but entrepreneurs should try to quickly identify them to prevent the business hopelessly pursuing a failed project. What could provide aid is the study of Crowne (2002b), who lists 15 types of common failures and their potential remedies, which are presented in Table 3.1. These are related to either of the following three phases: the *startup* phase, where the first sale has yet to be established, the *stabilisation* phase, which starts when the first product has been bought, and the *growth* phase, which begins when new customers are attracted to the product.

3.4 The Importance of Pivots

Pivots occur occasionally in software startups and are often crucial for pursuing a successful future. For instance, before the photographic success of Flickr, it started as an online role-playing game. Sharing and saving photographs to a web page, which was only a small part of that game, turned out to be the most fun part. This was then further developed to the contemporary Flickr (Basulto, 2015). Another example is Twitter, which started off as Odeo, a podcasting network. Odeo struggled as a company and Twitter bubbled up as a side project, which soon became the main project. Many other companies that are now successful experienced a pivot, examples being Youtube, Instagram, Groupon, Nintendo, and more (Basulto, 2015). A pivot is defined as an unusual type of change intended to test and validate the hypotheses about a product (Ries, 2011) and, whilst not compulsory related to, a business model (Blank, 2007; Maurya, 2012a; Bajwa et al., 2017). Bajwa et al. (2016) also define a type of change relating to one or more elements of a startup: entrepreneurial team, business model, product, or engine of growth (slightly resembling the evaluation dimension in subsection 3.1.6). It does not fall within the definition of a pivot if all four elements are involved in the change; this indicates starting a different and new business.

3.4.1 Types of Pivots

Literature discussing the types of pivots that startups can encounter is scarce (Bajwa et al., 2017). However, Ries (2011) presents ten categorisations of pivots. Table 3.2 presents these types of pivots, followed by other types discovered in literature. Interestingly, there is evidence that a *domino effect* (Terho et al., 2015) can happen: the occurrence of a pivot triggering another pivot (Bajwa et al., 2016). Some types of

Phase	Failure	Solution
Startup	Developers are inexperienced	The leading developer of the company should be highly experienced. This person should also be a technical accomplished leader, to influence less experienced colleagues.
	Product is not really a product	Budget for large expenditure to productise a custom solution.
	Product has no owner	There should be a clear product owner, or ownership about the product, who is preferably a market-oriented engineer that communicates between sales, development, and marketing.
	No strategic plan for product development	A strategic plan should be known for the startup, provided with clear objectives in the short and medium term.
	Product platform is unrecognised	The product platform components should not provide conflicts with the strategic plan (for instance components that have reached the end of life).
Stabilisation	Founders will not let go	Founders must either truly accept a subordinate position, assume a mainstream executive role, or join the board as a non-executive director.
	Development team fails to collaborate	Promote early developers who show leadership and technical potential. Swiftly eliminate any hiring mistakes or other weak links.
	Product is unreliable	Fixing a significant number of defects is hard work and requires experienced management. If it concerns an inexperienced team, skilled people should be brought in to assist.
	Requirements become unmanageable	A certain business process is needed to capture new requirements, prioritise them, and assess their value and feasibility.
	Product expectations are too high	The true state of the product has to be understood by the executive team, plans for the company should be made accordingly.
	Service provision delays development	Sufficient time should be allowed to the development schedule, therefore the full range of services that are expected should be considered.
Growth	Skills shortage delays development	Identify people that posses crucial skills and skills that are required. These should be spread across the team using approaches like documentation, buddy programming, and shadowing.
	Platform creep delays development	A business care should be established for additional platform components before including them in the development plan.
	Product pipeline is empty	Resources should be committed to the invention and development of novel products.
	No process for product introduction	New product introduction should be made as a repeatable process. All stakeholders should be involved in this.

TABLE 3.1: Different types of failures, with their solutions, adapted from Crowne (2002b).

pivots are closely linked and simultaneous pivots also occur. Furthermore, Bajwa et al. (2017) state that in their study the Customer Need Pivot was the most common pivot. This relates to finding the right fit for a startup, discussed in Section 3.2, where understanding the customer problems and needs are crucial.

3.4.2 Pivot Triggers

Every pivot can be related to an event that triggers such a pivot. The exploratory work of Bajwa et al. (2017) describes fourteen pivot trigger categorisations that they discovered within 49 cases, which are shown in Table 3.3. Each trigger was confirmed in at least one of the cases and was thus a cause for a type of pivot as described in Table 3.2. The triggering types are classified as either external or internal. Internal factors relate to the activities or decisions derived from the startup itself, whereas external factors are beyond the control of the startup (Bajwa et al., 2017). The majority of the startups studied encountered external triggers that led to a pivot, where negative customer reaction was the most occurring trigger. Generally, it appears that there is not a clear one-to-one relationship between pivot types and triggers within software startups. Furthermore, the pivot trigger list is potentially incomplete and can be enriched with new types. Lastly, software startups can use these triggers to be more aware of potential pivots in the near future.

3.5 Takeaways

This subsection provides the major takeaways of this chapter. It started by laying a foundation concerning the contemporary software startup approaches that could be relevant for this research: The Lean Startup with Customer Development, Design Thinking, The Lean Product Playbook, and The Startup Owner's Manual. These approaches shall be used as input for our go-to-market approach. Therefore, SQ1 *(Which existing startup approaches are currently described in literature?)* has been answered. Moreover, SQ1.1 *(What insights are gathered from startup failures)* has also been answered. With failures emerges validated learning: pivot in time or else a failure can put the startup out of business. This is why startup failures should be embraced and learned from. Lastly, SQ1.2 *(What is the influence of pivots within startups?)* has been also been answered. The different types of pivots are paramount to understand as a startup since they can be recognised or foreseen via pivot triggers.

Type of Pivot	Description	Example	Source
Zoom-in Pivot	A single feature of the product becomes the entire product.	Sharing photos functionality within a messenger application becomes an application of itself.	Ries (2011)
Zoom-out Pivot	The entire product becomes a single feature of an even larger product (the opposite of zoom-in pivot).	A sentiment analysis becomes a part of a full market analysis dashboard.	Ries (2011)
Customer Segment Pivot	The customer base is different than originally anticipated and the product product now solves a real problem for real customers.	A gaming application designed for younger people is actually only interesting for elderly	Ries (2011)
Customer Need Pivot	Because of customer feedback, a new (related) problem emerged, where the old solution did not solve an important problem. Usually, developing a new product is required.	During the search of improving a modelling language tool, customer needs indicated a new tool will solve a problem.	Ries (2011)
Platform Pivot	Switching platforms, i.e. changing from platform to an application or vice versa.	Changing from a platform that hosts online shops to an online shop itself.	Ries (2011)
Business Architecture Pivot	The business architecture of the startup is changed, choosing to operate within a mass market, or a niche market. Both at the same time is impossible.	Selling medium priced smartphones shifts to selling high-quality, very expensive smartphones.	Ries (2011)
Value Capture Pivot	The company changes its way of capturing value (how it creates revenue).	Software sold as licences is now changed to Software as a Service.	Ries (2011)
Engine of Growth Pivot	A startup can change its growth strategy to improve its upscaling. There are multiple growth engines: *viral* (word-of-mouth), *paid* (buying customers), and *sticky* (retaining customers).	A startup gets picked up and advertised by social influencers, where it first had nothing.	Ries (2011)
Channel Pivot	The startup changes its way of bringing the product to the customer.	Switching from sending out emails to large social media campaigns.	Ries (2011)
Technology Pivot	A novel and different technology is implemented to deliver the same solution.	A startup relying on hard-coding websites now implements a website builder tool.	Ries (2011)
Complete Pivot	All elements of the startup are changed (targeted market, product, and business model), except for the entrepreneurial team.	In this table are multiple examples regarding adaptions in one of these elements.	Bajwa et al. (2016)
Team Pivot	Key members of the entrepreneurial team are changed, or a development team is changed.	One co-founder decided to move forward with the company, letting two other co-founders go.	Bajwa et al. (2016)
Side Project Pivot	Startups have certain ideas of where to go with their product, but they still engage in one or more side projects to ensure cash flow. Sometimes, this becomes their main occupation.	A startup involved in recognising objects in video material might help classifying databases, which then becomes their main occupation.	Bajwa et al. (2017)
Market Zoom-In Pivot	Startups shift from focusing on the entire market to one specific sector.	When testing an application with a market segment, only a small segment part provided a positive response.	Bajwa et al. (2017).

TABLE 3.2: The types of pivots startups can endure, provided with its source.

Grouping	Triggering Factor	Description
External	Negative customer reaction	Relates to slow customer retention, slow customer acquisition, no or negative response from customers, etc.
	Unable to compete with competitor	Multiple competitors outplay the startup by working on a similar idea more effectively.
	Technology challenge	Multiple challenges related to technology. Includes limitation with existing technologies and better technology availability.
	Influence of partner/mentor/investor	Pressure or influence from partners, mentors, or investors to change the direction.
	User appreciation of one particular feature of the product	Users appreciate one specific feature instead of showing interest in the complete product.
	Unanticipated use of the product by users	The product is used in an unexpected manner by the users, which was unforeseen.
	Wrong timing	The market is not ready for the solution the startup provides.
	Positive response from an unforeseen customer segment	One specific customer segment reveals more interest in the product, among different customer segments.
	Running into legal issue	There are legal issues with other companies.
	Side project more successful than main project	There was not enough interest in the main project, but the side project reveals more traction.
	Targeted market narrowing	The initially defined target market becomes smaller, making it more difficult to survive and grow.
Internal	Flawed business model	The revenue model is not working, or the cost of customer acquisition is too high.
	Identification of a bigger customer need through solving an internal problem	While developing an internal solution for the core product, the startup identifies that solution as the real customer need, being larger than the original customer problem.
	Unscalable business	The amount of interest in the solution of the problem is too little, resulting in an unscalable business.

TABLE 3.3: Types of pivot triggers, which lead to pivots, adapted from Bajwa et al. (2017).

Chapter 4

Method Analysis

Several strategies have been defined in Chapter 3: The Lean Startup, Customer Development, Design Thinking, The Lean Product Playbook, and The Startup Owner's Manual. These strategies are used as input for the approach identification in Section 4.1. Next, the approaches are compared via method comparison in Section 4.2. Section 4.3 presents the reference approach draft based on current literature, and Section 4.4 discusses the takeaways of this chapter.

4.1 Approach Identification

Four approaches derived from the strategies defined in Chapter 3. After comparing these approaches, they are used as input for the GTM approach. First, we provide an overview of the four PDDs. Figure 4.1 shows the high-level PDDs of The Lean Startup (Customer Development), Design Thinking, the Lean Product Playbook, and The Startup Owner's Manual. The following explanation refers to contents of Appendix B, which consists of PDDs that are supported by activity and concept tables. These supporting tables contain the thorough explanations of the (sub-)activities and concepts that appear in the PDDs; the statistics are shown in Table 4.1.

The high-level PDD of Customer Development (4.1a) consists of four phases (open activities): Discover Customers, Validate Customers, Create Customers, and Build the Company. From the latter open activity, a closed concept is derived: SCALED ORGANISATION. This concept is closed because the scope of this research focuses on the process to obtaining a scaled organisation. Having a scaled organisation is therefore one of the success indicators of a GTM strategy. The complete PDD, where the subactivities of these phases are visualised, and the activity table and concept table are presented, are found in Appendix B.1. The PDD overview of Design Thinking (4.1b) shows the three main phases of Design Thinking: Understanding the Problem, Create a Point of View, and Ideate. The open concept IDEA is derived from the Ideate activity. The complete PDD is visualised in Appendix B.2, together with its activity and concept tables. The phases of the Lean Product Playbook (4.1c) consist of: Determine Target Customer, Identify Underserved Customer Needs, Define Value Proposition, Specify MVP Feature Set, and Create and Test MVP Prototype. From the latter two open activities, one open concept is derived: the MVP. Appendix B.3 shows the complete PDD, proceeded by the activity and concept tables. Lastly, The Startup Owner's Manual (4.1d) contains completely similar phases compared to the The Lean Startup and Customer Development. However, the main difference is the use of closed activities with Create Customers and Build the Company, instead of using open activities. The PDD of The Startup Owner's Manual is visualised in Appendix B.4, together with the activity and concept tables.

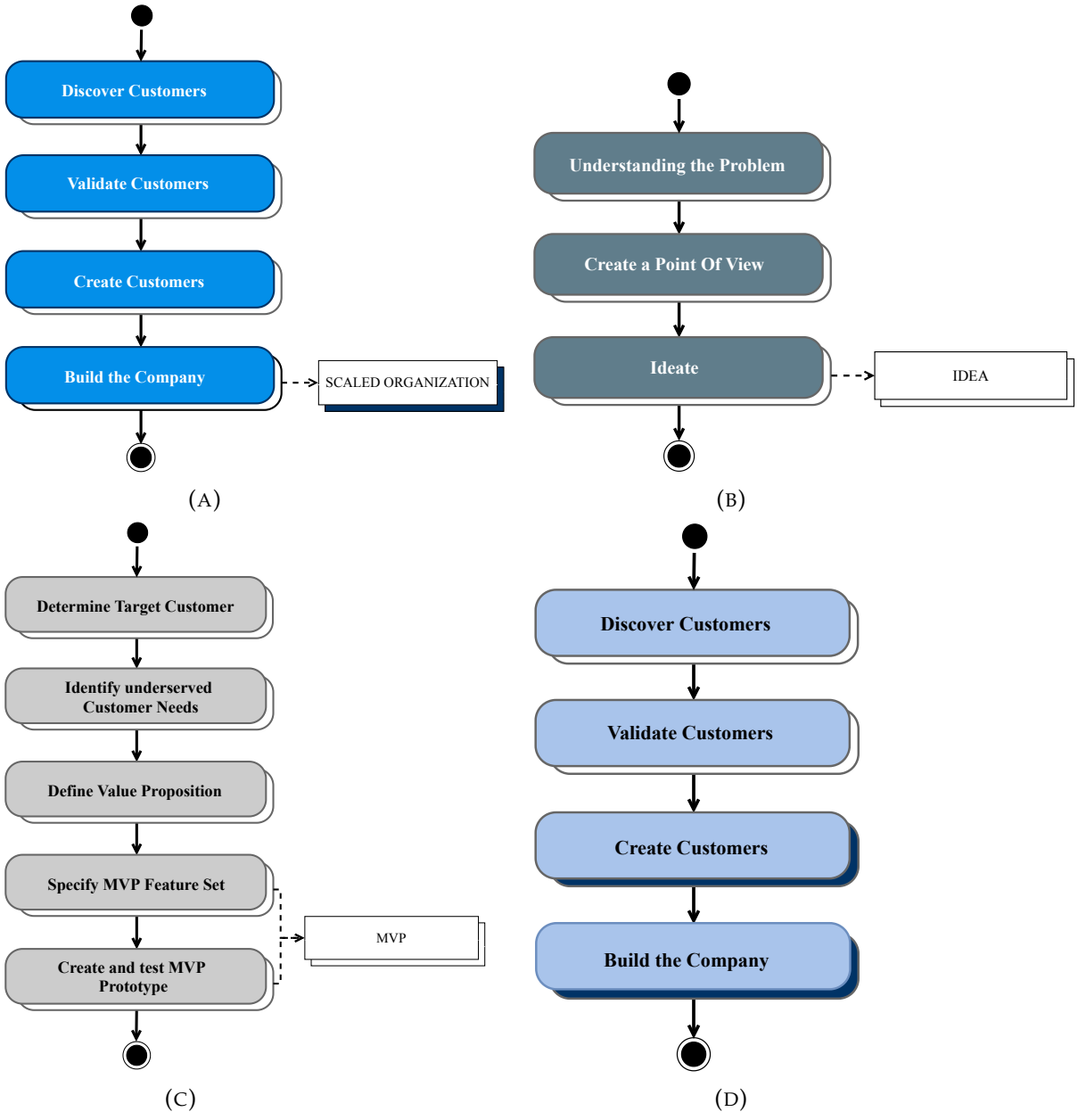


FIGURE 4.1: The high-level PDDs of The Lean Startup (A), Design Thinking (B), The Lean Product Playbook (C), and the Startup Owner's Manual (D).

Method Elements	The Lean Startup	Design Thinking	The Lean Product Playbook	Startup Owner's Manual
Activities	4	3	5	4
Sub-activities	18	10	10	11
Concepts	17	11	11	16

TABLE 4.1: The statistics (activities and concepts) of the approaches.

4.2 Approach Comparison

Following van de Weerd, de Weerd, and Brinkkemper (2007), the next step is to compare the methods by creating a super method, as described in 2.1. First, a super method is created that consists of two tables: one table listing every activity, and another listing the concepts. The symbols '=', '>', '<', and '><' are used to compare the activity fragments with each other. A blank field indicates that the activity is not present in the corresponding method row. A similar approach is used for the concept tables. However, here we use a string, or the '=' symbol, which holds the same representation as in the activity tables. The string represents the same concept, only with a different naming. A blank field indicates that the concept is not present in the corresponding method column. In Tables 4.2 and 4.3 we show excerpts of the activity comparison and concept comparison table respectively. The complete tables are found in Appendix C.1 and C.2. The abbreviations are as follows: The Lean Startup (TLS), Design Thinking (DT), The Lean Product Playbook (TLPP), and The Startup Owner's Manual (TSOM).

(Sub-)Activity	**TLS**	**DT**	**TLPP**	**TSOM**
Understanding the problem				
Identify the customers problem	>	=	<	
Uncover problems within identifiable groups	>	>	=	
Identify the user needs		=	><	
Partition market into market segments		>	=	><
Refine and tweak the product	><		=	><
Understand the needs of the target customer		><	=	
Identify specific needs for a market opportunity		><	=	
Test the customer problem	<	<	<	=
Test the customer solution	<	<	<	=

TABLE 4.2: Activity comparison table (excerpt), the complete table is found in Appendix C.1.

Concept	**TLS**	**DT**	**TLPP**	**TSOM**
Understanding the problem				
CUSTOMER'S PROBLEM	=	=	=	
USER NEED		=	=	
MARKET SEGMENT			=	
TWEAKED PRODUCT			=	
UNDERSTANDING	USER NEED	=	=	
MARKET OPPORTUNITY			=	
CUSTOMER SOLUTION				=
COMPETITION			=	

TABLE 4.3: Concept comparison table (excerpt), the complete table is found in Appendix C.2.

By analysing the comparison tables, we find that the Understanding the Problem, Discover Customers, and Validate Customers are relatively large phases, consisting of more sub-activities. This makes sense, since focusing on the customers and

their problem(s) should be performed as early as possible, instead of prioritising development (Giardino et al., 2015). This increases the focus on the problem-solution fit (Crowne, 2002b; Blank, 2007). Another observation that we make by analysing the comparison tables, is that some startup approaches introduce phases and activities that other approaches do not. For instance, DT introduces three phases that precede TLS. Especially in the second and third phase, DT provides several new activities. Another example is the introduction of multiple new concepts in the first phase by TLPP. These examples show that the approaches complement each other on a few areas. DT attempts to improve insight about your potential target customer, which is why it mostly precedes the other approaches. TLPP is the approach that complements DT on that field by introducing a couple more activities, an example being searching for specific needs for a market opportunity.

The TLS approach provides a solid basis for the PDD starting from the Discover Customers phase. This makes sense, considering it is a popular approach for software startups and other businesses (Furr and Paul, 2011; Furr and Dyer, 2014; Rasmussen and Tanev, 2015). However, we observe in the comparison tables that several elements from TLPP, and a few from TSOM, are an addition to TLS. For instance, some activities related to developing the MVP are more profoundly explained in the TLPP approach. We observe that TSOM provides little addition to the other three approaches. Several activities overlap or are already mentioned in the other approaches. For example, testing the customer problem and solution are already discussed in the other three approaches in a more meaningful manner. However, TSOM does provide some new elements to the final PDD. For instance, during the Validate Customer phase, it lays more focus upon developing a positioning statement, customer focused sales, and marketing materials. These aspects do not appear in the other approaches. Finally, the following and final step is the creation of the reference approach. Based on the aforementioned comparison of the four approaches, the reference approach is an executable method that consists of the best method fragments. The following section presents the reference approach.

4.3 Reference Approach

The resulting draft reference approach consists of seven main phases. Figure 4.2 visualises these phases: Understanding the Problem, Create a Point of View, Ideate, Discover Customers, Validate Customers, Create Customers, and Build the Company. Understanding the problem ensures that the software startup clearly perceived the intended meaning of a (real) customer problem, for which a solution can be created. For this, the user requires to be known to the startup. To increase the understanding in the following phase, the startup takes upon herself the point of view of the potential customer. This is done by, for instance, storytelling and creating visual frameworks. In the Ideate phase, these views are concretised and documented.

When the problem-solution fit is thought to be known (it is still based upon hypotheses at this point), the Discover Customers phase is initiated. The startup actively searches for both problem and solution, and tests the product with customers, assessing if the problem is an actual problem and that the solution solves that problem. The problem-solution fit is found if the startup discovers those customers, maybe even already making its first sale. This is visualised with the dashed (red) line between Discover Customers and Validate Customers. Next, the customers are validated, which essentially comprises taking actions to acquire more than one customer. If the startup manages to find a repeatable business model, and thus succeeds

finding more customers, product-market fit has been found. This is visualised with the second dashed (red) line following Validate Customers. From here on, the Create Customers and Build the Company phases initiate respectively. These phases fall outside the scope of this research, because this research focuses upon the trajectory to the product-market fit.

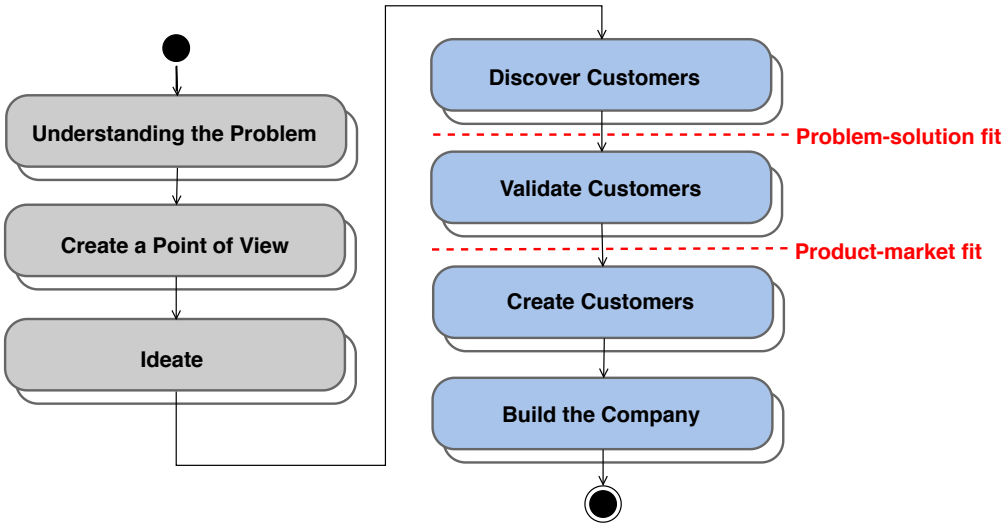


FIGURE 4.2: The reference approach for software startups: Process overview (draft). The gray phases highlight DT, the blue phases refer to TLS and TSOM.

The phases of the process overview are either grey or blue: the grey phases refer to the main phases of the Design Thinking approach, while the blue phases refer to The Lean Startup and The Startup Owner's Manual. Next, we show the draft reference approach, which is visualised in three parts to improve the readability: Figure 4.4, 4.5, and 4.6. The different colours are used to highlight activities from different approaches, which is explained in Figure 4.3. The reference approach is created by using the resulting best activities and concepts of the comparison tables (van de Weerd, de Weerd, and Brinkkemper, 2007).

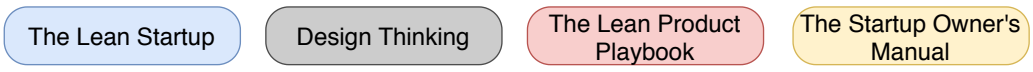


FIGURE 4.3: The used colouring explained: The Lean Startup (blue), Design Thinking (grey), The Lean Product Playbook (red), and The Startup Owner's Manual (yellow).

Understanding the Problem (Figure 4.4) shows that the phase derived from DT have been extended with several activities from TLPP, which are visualised as red activities. *Understand the needs of the target customer* and the consecutive activity ensure a clear understanding of what the customer actually desires. Initially, *Identify the customer's problem* was the first activity. However, this has been replaced with a more meaningful activity: *Uncover problems within identifiable groups*. The nuance in the words "identifiable groups" ensures that the customer's problem is recognised within a group of customers. The if-statement *Target market is feasible* ensures that the startup only proceeds if the target market is feasible. With the information gathered from the previous activities (about the customers' problem, market segments,

and size of the market), a fairly accurate estimation can be made. Create a Point of View and Ideate are completely based on DT, and no activities or concepts from other approaches are added. A better image is painted about your potential customer and the customer's need, by sharpening the point of view through insights, visual frameworks, and theories about user needs. In the Ideate phase, the focus is shifting to fulfilling that need by generating ideas. At the conclusion of the Ideate phase, one idea is followed through voting.

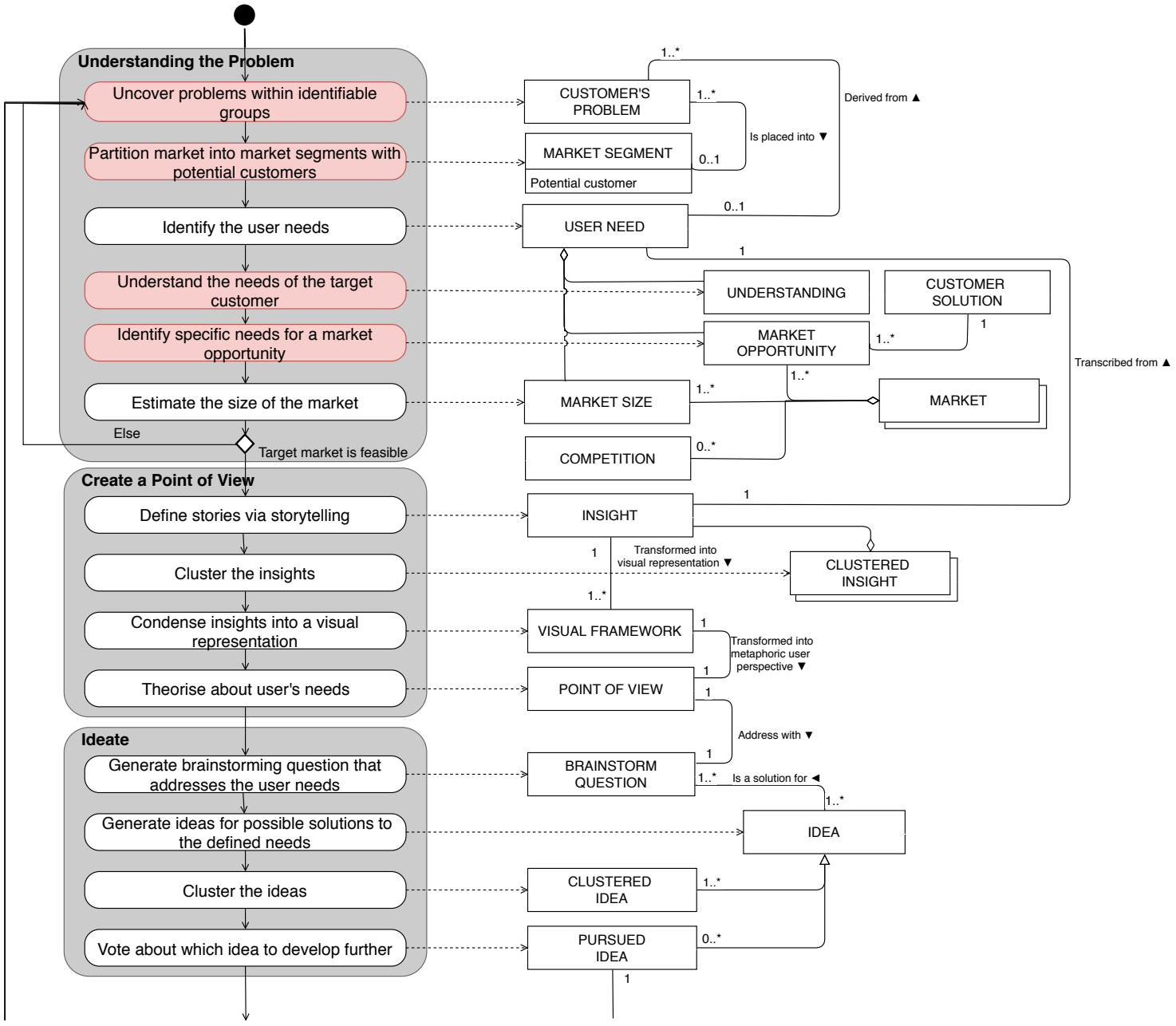


FIGURE 4.4: The complete go-to-market reference approach (draft) for software startups, showing Understanding the Problem, Create a Point of View, and Ideate (part 1/3).

The following phase, Discover Customers (Figure 4.5), focuses on identifying the first customer. It is important during this phase that the hypotheses created in the aforementioned phases are confirmed or falsified. Before engaging into interviews with potential customers, it is important to first *Document the customer-problem-solution hypotheses* and hypothesise about *business, MVP, and funnel assumptions*. These assumptions and hypotheses are tested with prospective customers and from this, we *Elicit market insights, customer-problem-solution feedback, and product-market fit insight*. It is important to talk with potential customers because they are the only ones that can falsify or confirm the hypotheses. Next, *A business plan is created*, which may be in whatever form the startup desires. A business plan holds certain assumptions about the product. The following step is to *Engage in interviews to discover core product functionalities (must haves)* and assess whether the business plan corresponds to those must haves. After several interviews, the data that is gathered should start to look the same, especially if the customer problem resides within identifiable groups that resemble a clear user need. At this point it is wise to consider a pivot if required: it is recommended if the proposed customer solution is not solving any customer problem (Customer Need Pivot in Table 3.2). This can be a complete or partial pivot, as explained in Section 3.4. With the core product functionalities in place, and the business plan confirmed through the interviews, it is time to start *Creating an MVP based on the problem-solution fit*. Afterwards, it is important to *Learn the value of the solution* with customers. The consecutive if-statement ensures that the customer is satisfied with the proposed MVP. The Discover Customers phase is concluded if that is the case, which indicates that the *problem-solution fit* has been found.

Next, the customers need to be reached and acquired (Figure 4.6) in the Validate Customers phase. To initiate this process, the startup *Develops a roadmap for acquiring and converting prospects into customers*. This ensures that the focus is shifted more towards making sales. *A product positioning* clarifies the targeting of the product, reasons why customers would buy it, and highlights competition (and differences compared to it). The following step is to *Convert the MVP into a high-fidelity MVP*, which makes it more attractive for customers to buy the product. TSOM adds four consecutive activities (in yellow) to the reference approach that emphasises the importance of sales and marketing. After these materials and roadmaps are in place, the *Customer acquisition and activation plans are tested by getting out of the building*, to assess if the product positioning is still valid. By validating with new customers, *Feedback is integrated into the product*. If the product positioning still holds and the customers are being acquired, the product-market fit is achieved. If the *Company positioning* does not indicate that a pivot is necessary, the Validate Customers phase is finished.

Finally, the Create Customers phase is initiated, where we explain the importance of *Creating a Business Model Canvas (BMC)*, and, inevitably, approaching your customer segment to start scaling. When additional Customers are Created, you consecutively start Building the Company, where the goal is to move from a startup to a scale-up. As mentioned before, there are more activities concerned with the latter two phases However, these fall outside the scope of this research; they are only briefly touched upon.

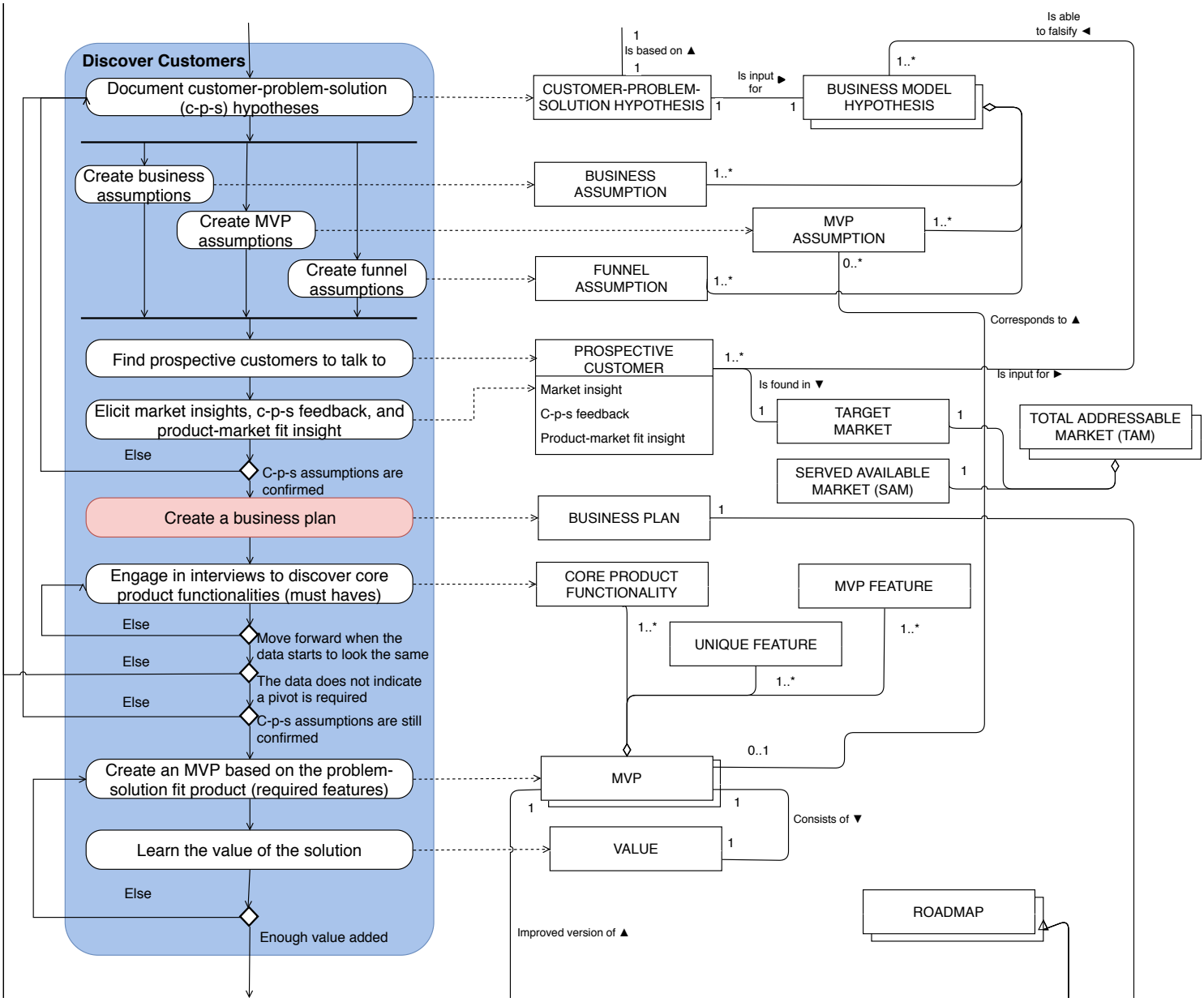


FIGURE 4.5: The complete go-to-market reference approach (draft) for software startups, showing Discover Customers (part 2/3).

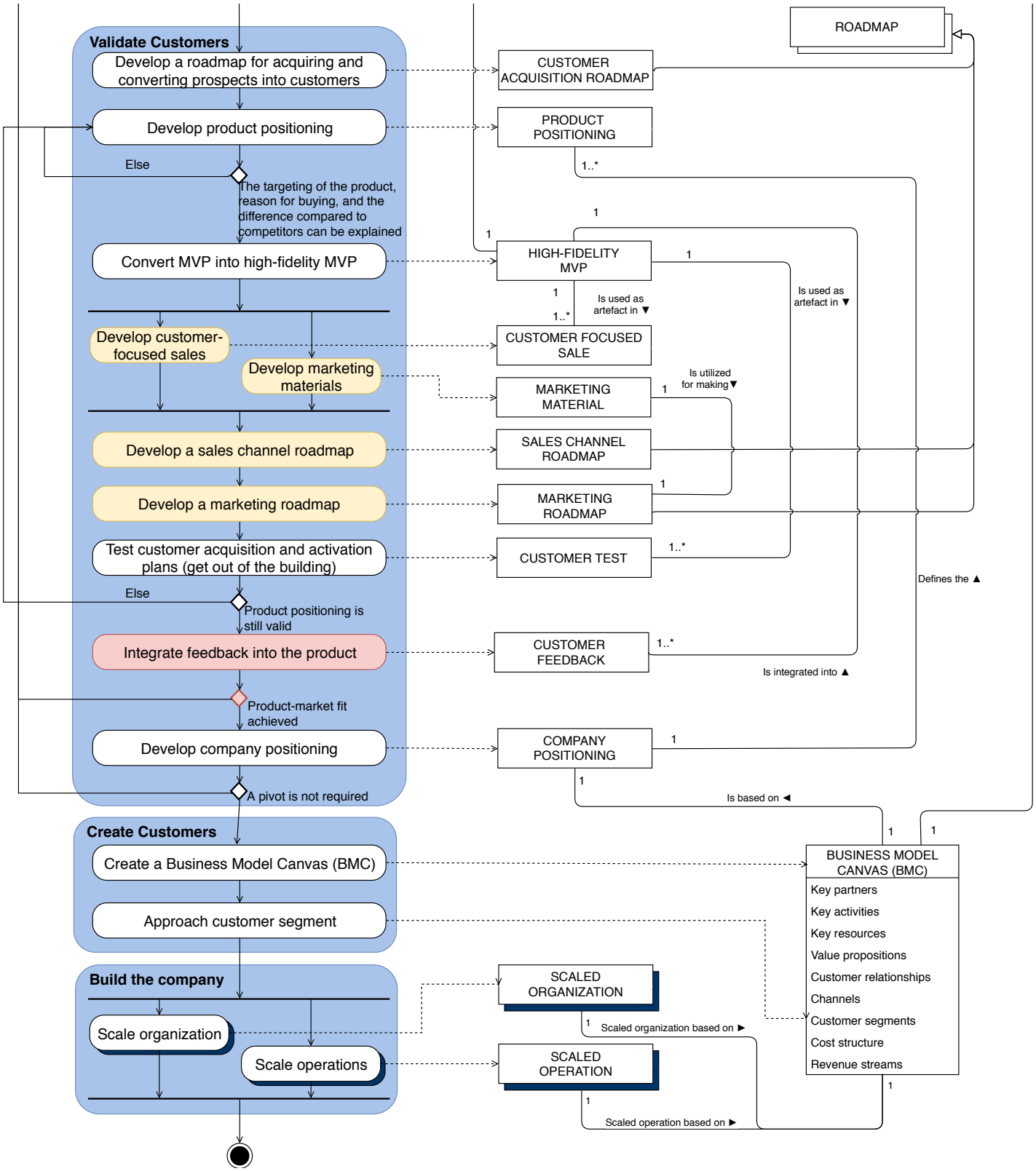


FIGURE 4.6: The complete go-to-market reference approach (draft) for software startups, showing Validate Customers, Create Customers, and Build the Company (part 3/3).

4.4 Takeaways

This subsection highlights the final conclusions of the method analysis. First, the approaches are identified from the strategies discussed in the literature reseach. SQ2.1 (*How are relevant method fragments identified within the scope of this research?*) was already partially answered in Chapter 2. By deriving PDDs of the strategies, we completely answered that research question in Section 4.1. By creating the super approach, comparing the method fragments in comparison table, and choosing the best fragments, we answered SQ2.2 (*How can multiple strategies be evaluated by method comparison?*) in Section 4.2. With these subquestions answered, we inevitably also answered SQ2: *Which method fragments are identified from startup approaches?*

The relevant method fragments that resulted from the method comparison are used as input for the GTM approach, which resembles a complete reference approach. The creation of this approach (also in the form of a PDD) provided a solution to SQ3 (*How can a go-to-market approach be assembled using method fragments?*), where the full draft PDD is visualised in Section 4.3. The GTM approach is now used as input for the case studies with multiple co-founders of software startups that have found the product-market fit. The following chapter provides insight in the *holistic multiple-case study*, where we evaluate the draft go-to-market approach with twelve co-founders of software startups that have found the product-market fit.

Chapter 5

Multiple-Case Study

During the multiple-case study, we let domain experts who have found the product-market fit (founders of software startups), evaluate the reference approach. This information is elicited from a series of semi-structured interviews. We introduce these experts in Section 5.1. Next, we present an overview of the case modifications in Section 5.2, where we take a closer look at those modifications in the subsequent sections (5.3 and 5.4). In Section 5.5, we present the main conclusions of this chapter.

5.1 Introducing the Cases

Table 5.1 provides insight into the software startups of the interviewees (the founders). These cases are anonymised and listed in random order. We provide general information to show the diversity of software startups that participated. These startups are chosen conform the inclusion criteria proposed in Section 2.3.

Company	Type of Software	Year of Establishment	Number of Employees
C1	Product feed management tool	2014	75
C2	Online ticket sales tool	2012	15
C3	AI-driven customer journeys platform	2011	92
C4	Business Intelligence for lawyers	2014	10
C5	Decentralised network for Artificial Intelligence	2017	15
C6	Online supermarket	2015	200*
C7	Structured communication and meeting tool	2014	7
C8	Financial administration	2017	7
C9	Industrial machine manufacturers and users platform	2014	10
C10	Planning tool (ERP)	2016	9
C11	Customer experience and feedback	2012	140
C12	Planning tool (ERP)	2013	15

TABLE 5.1: Multiple-case study details. *200 Employees working in HQ; with logistics included, this number increases to 3000 employees.

The multiple-case study details show the year of establishment and the number of employees currently working at the startup. C6 actually has 3000 employees working there in total because it is an online supermarket. We list 200 employees because these are employed at the headquarters of the company, while the other employees are working in logistics; these are therefore omitted from the table. Furthermore, one case was dropped because it did not meet the inclusion criteria. Specifically, it did not match either of the following criteria:

- A startup has an innovative business model and/or deploys innovative technologies;
- A startup shows significant growth either in the number of employees or in turnover.

The startup did not deploy an innovative business model nor innovative technologies; it provided IT related services to customers. Furthermore, due to its service-oriented nature, it was not showing significant growth in the number of employees nor turnover. This was discovered during that case, and it proves that the GTM approach inclusion criteria were chosen correctly. Lastly, the case study validation risks presented in Section 2.2 were mitigated, and the research risks introduced in Section 2.5 were avoided. To provide an indication of the case study and specifically the evaluation of the GTM approach, we visualise one of the evaluations with a case participant in Figure 5.1.

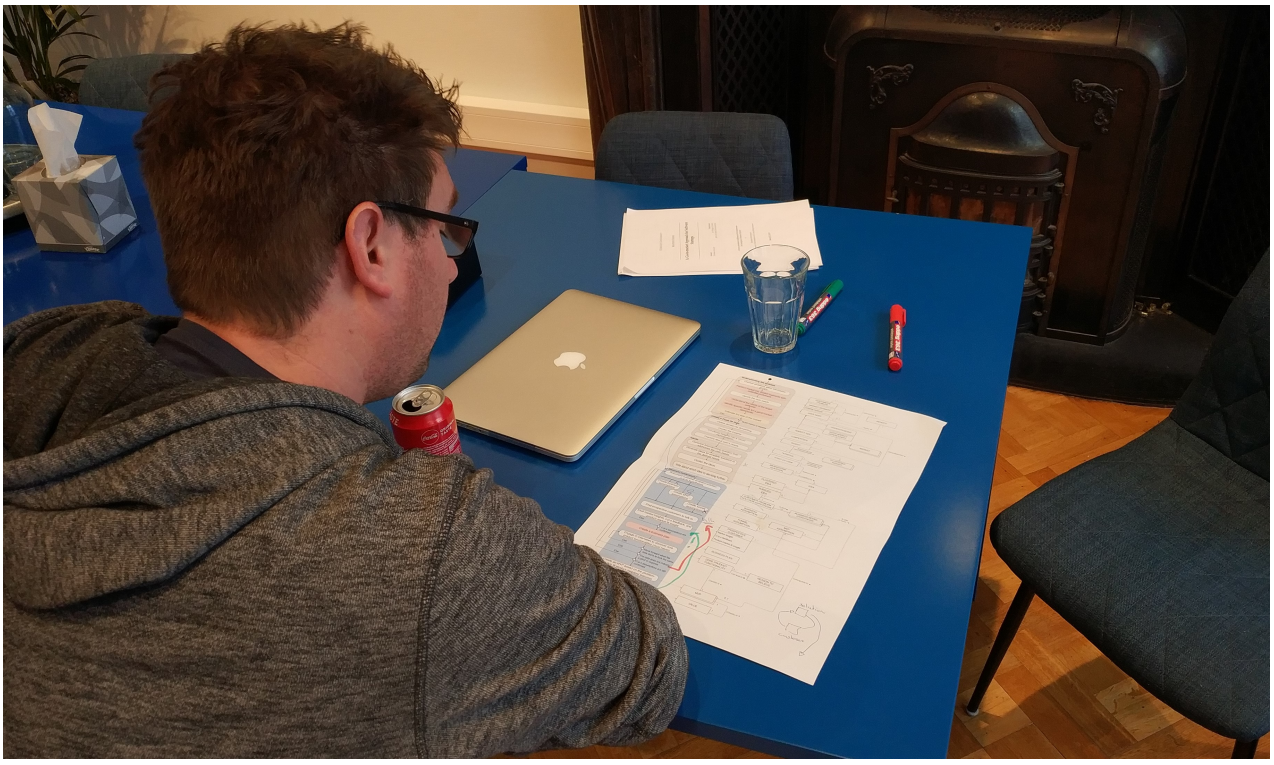


FIGURE 5.1: One of the case participants evaluating the reference approach. The markers and pens are used for notes and comments about the GTM approach.

5.2 Approach Modifications

In this section, we provide the general results of the holistic multiple-case study. We have created a table that provides insights into the performance of a sub-activity by each case participant. Table 5.3 shows an excerpt of the aggregated data of the process-side of the GTM approach that was presented to the case participants. The complete table is found in Appendix D. From left to right, we visualised the phases, IDs of the sub-activities, the sub-activities, and the cases; the latter are identified from C1 to C12. A case could either performed (Y) or skipped (N) a sub-activity. An example is the first sub-activity of the phase Understanding the Problem (A1): *Uncover Problems within identifiable groups.* Following this example, we conclude that all twelve startups did follow this sub-activity.

	ID		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	
Understanding the Problem	A1	Uncover Problems within identifiable groups	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	A2	Partition market into market segments with potential customers	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	N	N	O
	A3	Identify the User Needs	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	
	A4	Understand the needs of the target customer	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	A5	Identify specific needs for a market opportunity	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	
	A6	Estimate the size of the market	Y	Y	Y	Y	Y	N	Y	Y	Y	N	N	Y	O

TABLE 5.3: Aggregated data of the process-side of the reference approach (excerpt), the full table is in Appendix D.

Visualised in Table 5.3 in the rightmost column, we conclude that fourteen sub-activities are marked as optional (O), and one is excluded (E) in the final reference approach, which resulted in 33 retained sub-activities. In total, there are 48 sub-activities evaluated by the case participants. The inclusion and exclusion criteria, where N indicates a non-performed activity, are as follows:

- N<25% = inclusion;
- 25%≤N<50% = optional activity;
- N≥50% = exclusion.

Figure 5.2 shows the final process overview of the reference approach. Compared to Figure 4.2, it shows that the initial seven phases are still performed subsequently after each other. However, two parallel phases are added in green, namely, Learn and Build. The Validate Customers phase is changed to Convert Customers since *"converting your discovered customers should be the first activity"* (C1, C3). It became noticeable that the word validating created confusion: it potentially insinuates that the startup already has customers, which it needs to validate, while these are potential customers. Furthermore, the second phase is changed to Create a Customer Point of View to emphasise the customer perspective (C3). Lastly, Table 5.3 shows that the phases Create a Customer Point of View and Ideate contain solely optional sub-activities. We found that software startups potentially end up with an apparent customer problem after completing the Understanding the Problem phase. If this was the case, it was unnecessary to get a sharp view concerning the customer and generate ideas, which led them to proceed to Discover Customers. Therefore, we added an if-statement (*Customer problem is clear*) after Understanding the Problem that skips the following two phases if the solution is known to the startup.

The previous PDD was too linear, especially regarding involving customers early and continuously in the startup process. This should be transformed into a more iterative, constant process. The first three phases focused on hypothesising about

customer problems, but when an idea is generated it is vital to *"involve customers as soon as possible"* (C1, C8, C10). This also introduces the creation of the MVP as a parallel process, which is validated by customers (C6, C7). One case participant (C10) acquired customers via partners to immediately validate their created MVP. The fifth case participant stated: *"Obviously, you should understand the problem good enough for yourself, but you should validate with customers as soon as possible."* C8 stressed that *"you should not create a product before you know how to sell it to the customer"*, which is why we focus first on customers before building the MVP in the Learn and Build phase. Similar to the Build-Measure-Learn loop (Blank, 2007) as discussed in Section 3.1, we have added the Learn and Build phase parallel to the first four phases (before achieving problem-solution fit), which ensures customer involvement as soon as possible and encourages validated learning by the startup (Ries, 2011). This was also recognised by C12, who stated that commitment and dedication from (at least) the first customer is critical for a successful start.

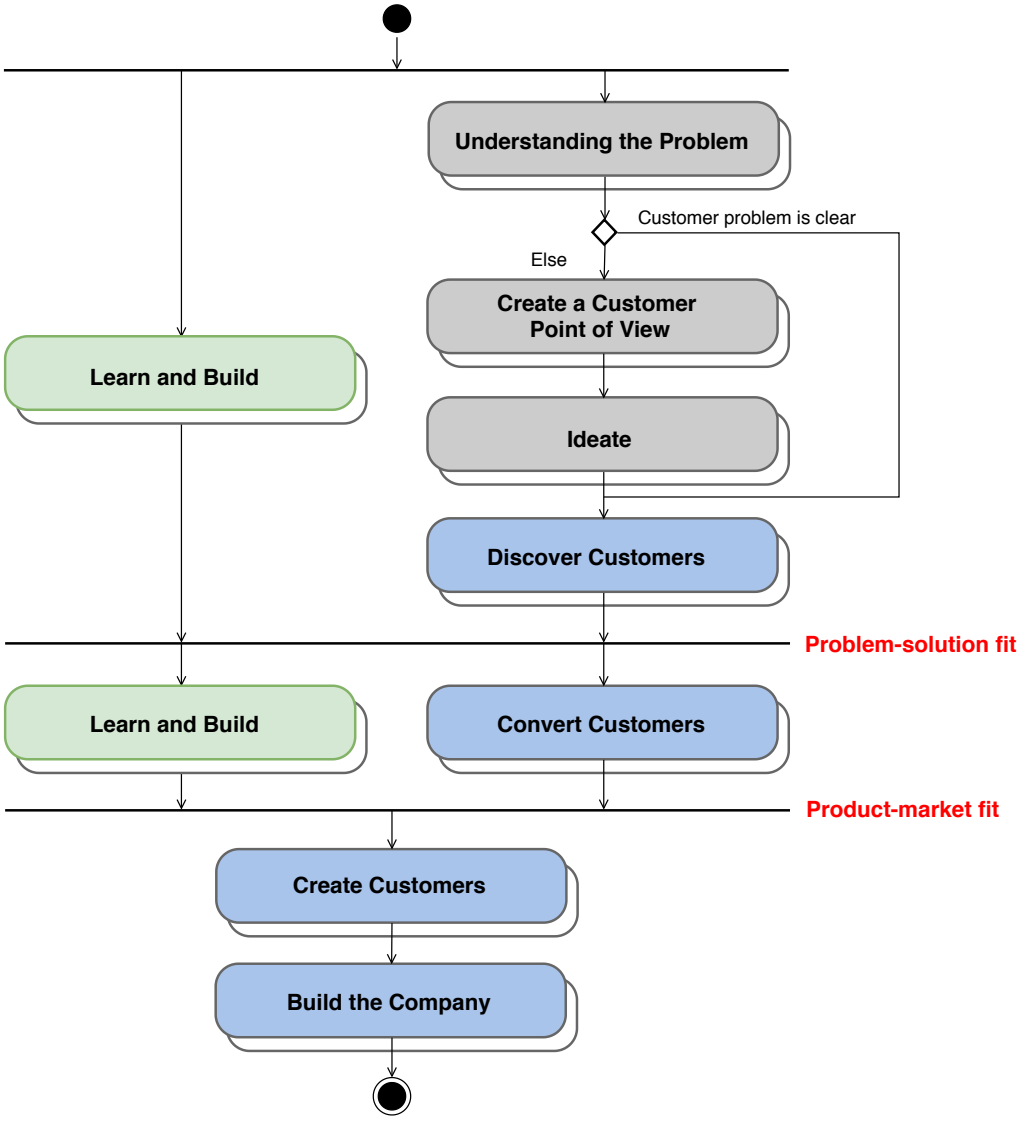


FIGURE 5.2: The reference approach for software startups: Process overview. The green phases are derived from the multiple-case study results.

We added a similar parallel phase after the problem-solution fit because Convert Customers contained sub-activities that should also be in a *"continuous parallel feedback loop"* (C1, C2). Interestingly, C6 emphasised the importance of finding the problem-solution fit before searching for the product-market fit. The startup should work on a problem worth solving, where it is noteworthy that *"some problems are not obvious in early stages, for instance, with Facebook."* The problem should be uncovered during the first few phases (with potential customers), for which a solution should be discovered. When the solution adequately fits the customer problem, and the startup has a clear understanding of why the problem is solved, it is relatively easy to obtain a product-market fit. Each case that was interviewed found himself being in one of the latter two phases, which implicated that they all found product-market fit. Some were still approaching the customer segment to Create Customers, while others were focusing on Building the Company by trying to scale. The next subsections discuss the reference approach in detail, separated into two parts: achieving problem-solution fit and achieving product-market fit.

5.3 Achieving problem-solution fit

Figure 5.3 visualises the complete GTM reference approach, with implemented changes derived from the case studies. The previous section provided an overview of the GTM approach, while this and the subsequent section focus on the details of the GTM approach, by going through each phase. This section, in particular, shows what needs to be performed by the startup to achieve problem-solution fit.

5.3.1 Understanding the Problem

One of the most interesting activities of this phase was A2: *Partition market into market segments with potential customers*. When analysing the aggregated data, this activity should be optional. However, several case participants agreed upon the fact that thinking about partitioning your market beforehand was expected to have a positive outcome. Some stated that they did not do this enough in the beginning, while another case participant (C6) indicated that this usually comes much later because it depends on the solution to your problem. Due to the importance of this activity highlighted by several case participants, we decided to preserve it as a mandatory activity. Furthermore, C3 and C7 explained that *"this is something that can be changed in a later stadium because sometimes you find out what your best market segment is when you first attract some customers."* As C7 further puts it: *"We did approach people that were daring enough to try our solution."* It makes sense to re-partition in a later stadium, which is why we added such an activity in the second Learn and Build phase, as described in Section 5.4.

Another interesting activity was A6: *Estimate the size of the market*. This is the second activity that was registered as an optional activity by the process data. The reason for this is some startups did not explicitly estimate the size of the market, but implicitly did. After all, when a startup partitions into market segments and thinks about user needs, it also makes a rough estimation of the market size. We, therefore, think it is useful to go through this activity as well. Tronstad (2008) shows that if the market is too small, the startup, operating costs, and capital are not covered by the potential sales. However, as C3 puts it: *"This is not rocket science, measure the size relative to the startup, but do this quickly."*

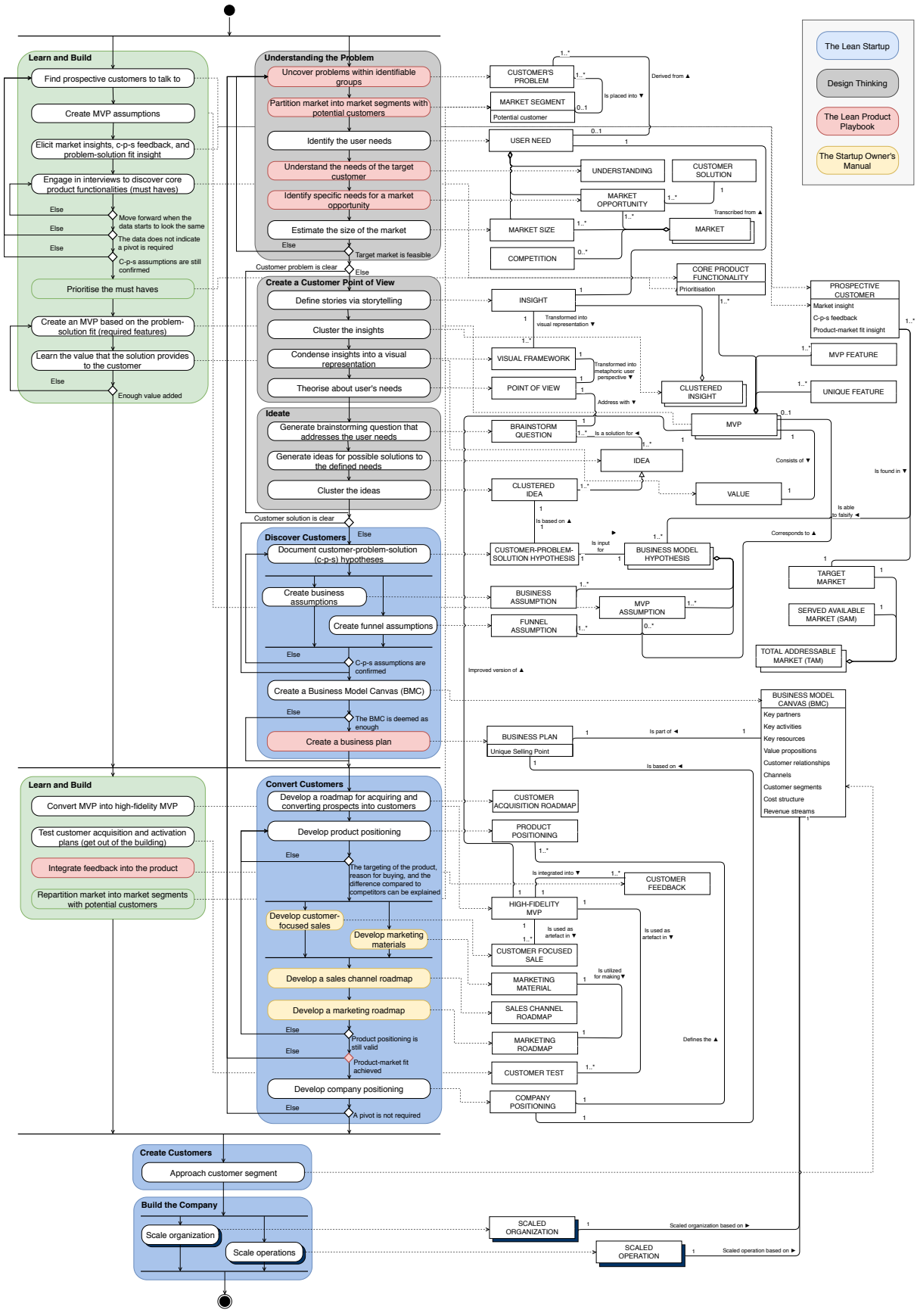


FIGURE 5.3: The complete go-to-market reference approach for software startups.

With regards to stumbling across the actual customer problem, several results were found. Almost every co-founder encountered the problem himself by working in a specific domain where the problem occurred, where we can name a few examples. One of the case participants worked as a lawyer where he encountered problems regarding time writing in the legal profession. Another founder worked in the e-commerce space, where he observed a missing solution for ordering online groceries. However, most founders worked as a consultant where they observed a problem during a consultancy project. They observed that companies were either not solving the problem or *"handling it incorrectly"* (C3), which *"triggered"* (C2) the founders into building a prospective business that solves that problem. In other cases, the founders could stumble upon the customer problems in their personal life by accident or by *"luck"*. Luck was a term that was coined by a few founders, a phenomenon we explain more in Chapter 7.

5.3.2 Create a Point of View & Ideate

The Create a Point of View and Ideate phases are now optional since each sub-activity was flagged as an optional activity. Interestingly, however, C5 found these two phases the most important: *"I would say that if you do not do this, you are definitely going to fail."* Such a quote does emphasise that the phases are relevant for some software startups. The second case participant stated that these phases were intensively used as input for the business plan and the BMC, which corresponds with the current layout of the model. Finally, we deleted A15: *Vote about which idea to develop further*, because multiple case participants explained that they never voted about pursuing ideas.

5.3.3 Discover Customers

After the Ideate phase finishes, the Discover Customers phase initiates. This phase now contains substantially fewer activities compared to the same phase before the case studies. These activities (A18, A20, A21, A24, A28, and A29) have been moved to the Learn and Build phase. The reason for this is that these activities related to approaching customers and creating the MVP, which, as described in Section 5.2, are now appropriate candidates for the Learn and Build phase.

Resulting from the process data, the activities A16, A17, and A19 are now optional. Since these activities focus on making the customer solution more apparent, the following if-statement is added: *Customer solution is clear*. Startups can then start working on the BMC in the subsequent activity. This activity (A45) is moved from the Create Customers phase to this phase because *"it should come before the business plan"* (C1, C7). The final *Create a business plan* activity is now an optional activity.

Interestingly, the opinions about the business plan were diverse. C7 stated that it (the business plan) *"is a living document, as is the BMC"*, while another expressed that such a plan *"is very uncommon these days with startups"* (C6). Following that both these software startups found the product-market fit, it is wise to make it an optional activity, where startups can decide and asses upon their own if it is a complement to the BMC. Furthermore, C6 also declared one thing about business plans: *"you are going to change it later because you work with very high-risk assumptions."* This is in line with, for instance, the works of MacCormack (2001) and Bajwa et al. (2017), where software startups are supposed to work under extreme uncertainty.

5.3.4 Learn and Build

As visualised in Figure 5.3, the first Learn and Build phase is parallel to four other phases. All activities but one are transferred from the Discover Customers phase to this phase. These activities derive from the Lean Startup, which fit well in this Learn and Build phase due to the nature of the underlying BML-loop (Blank, 2007). Case participants also recognised these activities as the first thing a startup should do (A20: *Find prospective customers to talk to*), or something the startup should either perform parallel or repeatedly (A18, A21, A24, A28, and A29). The sub-activity A21 brought confusion to some of the case participants because it stated 'product-market fit insight' instead of 'problem-solution fit insight'. We changed this accordingly to problem-solution fit because this phase is before the achievement of the problem-solution fit (therefore, problem-solution fit insights are elicited instead of product-market).

After the startup learns from the customer and gathers new insights, the startup start building the MVP. Building an MVP is paramount because even the startups that did not consciously start with an MVP recognised its importance, for example, C9: *"We do create MVPs now, but we did not back then. That is one of the precious lessons we have learnt."* C3 expressed that prioritisation of the core product functionalities (must haves) was missing in the model, which was substantiated by other case participants as well. Therefore, we added another activity after discovering the must-haves: *Prioritise the must-haves*. Next, the startup builds the MVP based on the supposed problem-solution fit, and its prioritised must-haves. A29 (*Learn the value of the solution*) was regarded as one of the essential activities by several case participants. This relates to one of the Lean principles: removing every non-value-creating effort through eliminating waste (Ries, 2011; Rasmussen and Tanev, 2015). It is essential to learn the value the solution brings to the customer as early as possible. However, it was not clear that it comprises value it brought *to* the customer, which is why the activity name is now: *Learn the value that the solution provides to the customer*. If the solution solves the customer problem (adds enough value), the Learn and Build phase is finished.

5.3.5 Final Notes

It is difficult to assess and pinpoint when a startup has found the problem-solution fit. In essence, the problem-solution fit is found when a solution is implemented that tests the trickiest hypotheses of the customer problem (Giardino, Wang, and Abrahamsson, 2014). However, several activities were crucial, which we summarise in six elements that are designated for an entrepreneur in Chapter 6) – for example, finding a resourceful initial market segment, and testing with customers as soon as possible. Validated learning with an MVP is also of significant importance for a startup. Learning from customers is proven to be essential for solving the customer problem. *"The key is to understand why it brings that much value for some customers, and why at this moment"* (C3).

5.4 Achieving product-market fit

The startup's new goal is to attract and acquire customers. Figure 5.3 shows four more phases that are discussed below: Convert Customers (previously named Validate Customers), the second Learn and Build, Create Customers, and Build the Company. This section, in particular, focuses on achieving the product-market fit.

5.4.1 Convert Customers

Each case eventually took a user-driven innovation approach, involving potential customers early on into the development process (Müller and Thoring, 2012). Usually, they carried this out by asking feedback regularly or even creating feedback groups or customer panels. C12 pivoted before entering its contemporary startup. They created a solution that did not suit a customer problem: *"We did what you actually should never do: think from the product and then the clients will come by themselves. The product seemed so generic; we thought that we were able to find a good niche."* This demonstrates a clear example of not following a user-driven strategy but thinking from the software product. These findings also emphasise that an immature problem-solution fit, where the customer is not actively involved with the development process, leads to failure (Giardino, Wang, and Abrahamsson, 2014).

Furthermore, following the multiple-case study results, we did not implement many changes in this phase. Another Learn and Build phase has been added, which contains three activities derived from the Convert Customers phase: A34, A39, and A41. These were either focused on improving the MVP or testing customer acquisition, which is suitable for the BML-loop. After all, these were supposed to be in a *"parallel feedback loop"* (C1, C6 C7). Following the method data, A37 was flagged as an optional candidate. However, C2 and C3 explained that this is an essential activity to achieve product-market fit, which resulted in it to stay mandatory in the final model. However, it should be stressed that *"it can be very time expensive"* (C2). It can also be interesting for a startup to create a *"sales analysis and strategy"* (C4), instead of making a roadmap. Finally, it is interesting to note that C6 explained that this activity (together with *Develop a marketing roadmap*) was only a small part of their work. This was mainly because they believed the product should explain itself. If that is the case, it should sell itself.

5.4.2 Learn and Build

The second Learn and Build phase contains sub-activities without arrows to take away its chronological order. Each of the four subactivities was recognised as something that the startup constantly does, or at a random order, parallel to the Convert Customers phase. This is why the startup can choose to randomly to start one of the activities when initiating the second Learn and Build phase. As mentioned before, A34 (*Convert MVP into high-fidelity MVP*), A39 (*Test customer acquisition and activation plans, get out of the building*), and A41 (*Integrate feedback into the product*), were moved from Convert Customers to Learn and Build.

Some case participants highlighted that the MVP should never be too far from the original product idea in this stage. This is difficult because it can be hard to say "no" to specific client wishes, even if it is likely to bring money. It is also essential to bear in mind that the MVP should remain scalable and generic; easy to implement for new customers. Most importantly, new features of the MVP should be continuously tested with multiple customers. On the one hand, C4 found that in hindsight, they *"should have tested the MVP with 100 other offices, not only for a long time with the first client."* However, contradictory, he stated that the product might not be good enough at that stage to test with other offices. Adding to that, C12 explained that they found the product-market fit after implementing with the first eight customers. Before that period, it was a customisation trajectory, whereas, after that point in time, the product was almost generic enough to implement quickly with new customers.

Finally, the following activity was added: *Repartition market into market segments with potential customers*. This was due to some of the comments provided with A2, for instance: *"This is something that can be changed in a later stadium because sometimes you find out what your best market segment is when you first attract some customers"* (C3, C7). This ensures the startup thinks about repartitioning its potential target market. Several case participants, for instance, C3 found an ideal customer after repartitioning in a later stadium, where you already discovered potential customers.

5.4.3 Create Customers & Build the Company

The last two phases almost did not undergo any changes. There was one significant change regarding the Create Customers phase, which was the location of A45: *Create a Business Model Canvas (BMC)*. Several startups mentioned that they did this earlier in their process, for instance, C1 and C7: *"The BMC should come with the business plan."* Therefore, it has been put in the Discover Customers phase before the startup creates a business plan. This is also because a BMC is potentially part of a business plan. Build the Company remained the same, which was also due to the phase not being entirely in the scope of this research. Interestingly, every co-founder interviewed did recognise him or herself in the latter two phases, thus having found the product-market fit.

5.4.4 Final Notes

Similar to the problem-solution fit, it is difficult for a startup to assess if the product-market fit has been achieved. Additional customers are more easily acquired when a startup verifies with them beforehand whether they would buy it. The startup must be wary not to change its course too much, or as C1 puts it: *"If we heard a certain requirement from one customer, we went to another and asked them if they would use it as well."* Several case participants would implement the must-haves if a certain number of customers were going to use it. This process requires constant feedback with the customers utilising the MVP. Again, six key startup elements designated for an entrepreneur are presented in the next chapter.

5.5 Takeaways

This section describes the conclusions of the holistic multiple-case study results. First, we introduced the twelve case participants in Section 5.1. Next, we showed the overall results of the multiple-case study in Section 5.2, and we subsequently described these results in Section 5.3 and 5.4, by showing the details. The subquestion we answered in this chapter was SQ4: *How are go-to-market approaches applied within case studies?* We printed the GTM approach and provided these to the case participants during an interview. The case participants were then able to comment on activities and highlight or change parts of the process where required. This resulted in qualitative data, utilised to adapt the reference approach accordingly. Chapter 6 presents six key lessons we learned from the GTM approach.

Chapter 6

GTM Approach: Lessons Learned

The startup industry taught us that some aspects of the go-to-market approach are of significant value, which we explain in this chapter. The GTM lessons present the most significant elements a startup should take into account to achieve the problem-solution fit and product-market fit. Therefore, this chapter substantiates the GTM approach.

First talk to potential customers

Do not focus on the technology first, assess whether the problem is, in fact, a customer problem by talking to potential customers. Numerous startups overlook this aspect, but the multiple-case study proved that this is the first thing that startups should do. Contemporary literature shows that after building the MVP, you should measure and test it with customers, for instance, via the BML loop (Blank, 2007). However, this implies that startups should first build something while they could have learned beforehand if there exists a customer problem that demands to be solved. One pitfall, especially for innovative startups, that should be noted, is that some inventions are solutions that were not recognised as problems by the potential customers beforehand, a classic example being Facebook. In that case, it is better to work with an MVP to test your hypothesised solution. Other than that, the startup should talk with potential customers and attempt to understand their problems as thoroughly and early as possible. This understanding of the customer problem is paramount for the subsequently devised solution, which should be a prototype as MVP. The Design Thinking (Müller and Thoring, 2012) strategy has proven to be a suitable strategy for this purpose. However, the Value Proposition Design of Osterwalder et al. (2014) also provides practical techniques that can be applied by startups. For instance, it shows how to make a customer profile with customer jobs, which can aid in finding the right customers. This also helps in finding the 'best' market segment, which relates to another vital aspect of this chapter.

Domain expertise aids in understanding the problem

Something we have not seen in literature but does help in finding the problem-solution fit is applying your domain knowledge. The case participants we interviewed usually operated in a specific domain where they eventually found a problem that required to be solved. One founder we interviewed worked as a lawyer at a big firm where he found that time writing by lawyers was being performed inefficiently. He then confirmed at similar firms that they encountered the same problem, which was enough incentive for him to create a solution (which also relates to the previous element). This example, substantiated by other cases, show that domain

knowledge and experience can help in finding or understanding the customer problem. An obvious customer problem also results in designing a fitting MVP without much effort, because you accurately know what the MVP should solve for the customer. Furthermore, it implies that if you are willing to operate in a domain that you do not know much about, it can be an advantage to work with someone in your startup that does.

Create an MVP and learn its value

Every founder that was interviewed for this study acknowledged the usefulness of an MVP. The MVP should only contain the must-haves; thus, the most essential feature set or functionalities. As early as possible, entrepreneurs should get out of the building to obtain quick feedback, and they need to prevent that a product is being created that nobody desires, i.e. for which there is no customer problem (Müller and Thoring, 2012; Blank, 2007). When the MVP is validated regularly with potential customers, the software startup activates *validated learning* and prevents the creation of 'waste'. This is crucial since startups generally possess limited resources (Giardino, Wang, and Abrahamsson, 2014) and endure intense time pressure from the market (MacCormack, 2001), which emphasises that time and resources should be well spent. A failed product can also put a startup out of business, if not foreseen in time because the majority of startups are engaged in a single project at a time. If the customer need pivot, which is the most common, is initiated in a relatively early stage, less time and resources have been depleted by the startup (Ries, 2011; Bajwa et al., 2017). This pivot is only foreseen when actively learning from the feedback the customer has about the MVP. The Value Proposition Canvas of Osterwalder et al. (2014) provides interesting details into testing methods with prototypes. It presents prototyping possibilities and guides startups step-by-step through the testing process.

Discover the 'best' market segment

This activity was especially important in The Startup Owner's Manual (Blank and Dorf, 2012). However, not every startup actively engages in this activity, even though it became apparent that it helps in finding profitable customers on the long-term. Interestingly, only one of the failures listed by Crowne (2002b) emphasises on the importance of choosing the right market segment, while we found that this is an important activity. Osterwalder et al. (2014) provide techniques to create customer profiles through customer jobs, pains, and gains. Some startups that were interviewed did not actively think about the 'best' market segment. However, in hindsight, they regretted that. The main reason for regret was that they later discovered other more profitable, or suitable, market segments compared to the initial segment they were operating in. Through *validated learning*, some startups can apply a *customer segment pivot* early in the startup phase (Ries, 2011). Edison, Wang, and Abrahamsson (2015) interviewed one case that went through validated learning and successfully applied a market segment pivot, which was crucial for survival. A positive response from an unforeseen customer segment, targeted market narrowing, or merely adverse customer reactions, are pivot triggers that most likely precede such a market segment pivot (Bajwa et al., 2017). Therefore, the startup should actively be aware of these pivot triggers and test its product in multiple customer segments, without losing too much focus. This insinuates that every market segment is easily reachable, which in practice is most often not the case. The cases that were interviewed explained that

they sold to the first customers that are willing to buy, which makes sense. However, a better market segment can still be discovered in a later stage, which is why we added a reflection on the market segment in this chapter.

Test customer acquisition plans

If entrepreneurs follow the aforementioned general elements, they focus on finding problem-solution fit. With the problem-solution fit in place, new customers must be reached by getting out of the building. As Blank (2007, p. 9) explains: *"In a startup, no facts exist inside the building, only opinions."* The next goal for the startup is to find the product-market fit, something that is easier found when an accurate problem-solution fit has been established. After all, if the potential customers are clear enough, you know exactly whom to target and to acquire as new customers. However, it is again important to remain focused and to not add waste to the MVP. Every startup that was interviewed still gathered constant feedback, even after finding the product-market fit; the BML-loop is still relevant in this stage. They also prioritised new requirements and verified with other customers to see if they would also need those requirements, where some would even handle a guaranteed minimum of customers that should use it before actually implementing. Therefore, constant feedback loops with existing and new customers are paramount for building the company. However, in this phase, it might be wise to reflect on the market segment that was initially chosen, which relates to the last element in this chapter.

Reflect on the market segment

There could be other profitable market segments to operate in than initially thought of, which must be continuously considered by the startup. Several startups that we interviewed recognised that they found a better market segment before finding product-market fit, and after finding the problem-solution fit. They initiated a customer segment pivot, as defined by Ries (2011). We have also seen one startup establishing a new business that arose from a side project with another market segment, which essentially is a customer segment and side project pivot combined (Bajwa et al., 2017). The new startup turned out to be far more profitable than the initial startup, and the customer problem was more evident than before. Other founders also agreed on the importance of reflecting on the initially chosen market segment and some implicitly changed their segment through time. Lastly, one case participant had to cut off its segment where he initially operated because it was close to putting the startup out of business. Therefore, this is an essential element, which does not necessarily put you out of business if neglected, but potentially lets you find (an improved) product-market fit. *"Reinvent yourself constantly"* (Osterwalder et al., 2014, p. 266).

Chapter 7

Discussion

This chapter reviews the lessons learned for academia (7.1), as became apparent from the approach creation and the GTM lessons learned. Finally, in Section 7.2, we reflect on the limitations, method engineering process, and threats to validity.

7.1 Lessons for Academia

Startups should focus on achieving the problem-solution fit first and foremost. As substantiated by the work of Giardino et al. (2015), to create something valuable for customers, startups need to comprehend their real problems. Focusing solely on technological solutions does not guarantee success and survival. With our GTM approach, we showed and evaluated an essential process that early-stage startups experience. The Lean Startup and Design Thinking complement each other, with the model proposed by Müller and Thoring (2012). However, activities derived from Design Thinking were only performed when the customer problem was not easy to understand, which in those cases turned out to be useful. Furthermore, we demonstrated the usefulness of the addition of The Lean Product Playbook (Olsen, 2015) and The Startup Owner's Manual (Blank and Dorf, 2012) to the GTM approach. The final approach contained activities that startups used to find the product-market fit, where only one activity was removed. In this work, we integrated the Build and Learn phases with four other integrative approaches. The validated learning is described by Ries (2011) as gathering knowledge through feedback from stakeholders as an evolutionary approach (Giardino et al., 2015). From an academic standpoint, integrative methods are the future of startup research: through the combination of conventional and less conventional methods, the chances of achieving problem-solution fit and product-market fit are higher. In essence, the learning of startups can become a meta-learning experience for academia as well. In this research, we created and applied Process-Deliverable Diagrams as a meta-modelling technique from method engineering. Explicitly formalising methods from strategies defined in the literature and evaluating them via domain expert interviews in a multiple-case study, provides substantial scientific contributions. In Subsection 7.2.1 we further reflect on method engineering and Section 8.1 presents future works.

7.2 Limitations

This section describes the limitations of this research with regards to the GTM approach. We also reflect on using method engineering for this purpose and discuss the threats of validity. First, with a Process-Deliverable Diagram, one is unable to

translate a complete strategy (this research focuses on an *approach*, instead of a *strategy*). There are three aspects relating to a strategy that are noteworthy and also highlighted by some of the cases as important, which we list below:

- **Team:** The GTM approach disregards the startup team or employees, while this is a significantly important aspect of the startup; having competent people aboard and an appropriate type of leadership is key (Ensley, Hmieleski, and Pearce, 2006; Giardino, Wang, and Abrahamsson, 2014). As C7 explained: *"The most difficult aspect of having this startup is to, besides building a great product, also maintain a good team."* Adding to that, C10 states that it is difficult to go from an idea to work with people suitable for the job. This also relates to the five of the fifteen failures mentioned in Table 3.1, based on the work of Crowne (2002b). Therefore, while the team is not reflected in the final approach, it is highlighted as an important aspect for the software startup.

- **Culture:** The approach also neglects the importance of the culture within a startup. As C5 puts it: *"Identify with your co-founders who you really are and what you really stand for."* Substantiated by C2, the startup founders should define the company culture and narrow down what they believe in on a day-to-day basis. This ensures that nobody ends up being confused. A positive culture brings capabilities that are paramount for the first steps of a startup and empowers building of practices on a solid basis (Dande et al., 2014). Passionate behaviour of the team is also required to overcome barriers (Giardino, Wang, and Abrahamsson, 2014).

- **Cash:** The absence of a healthy cash flow makes it difficult to operate for startups. For instance, C11 was multiple times on the verge of terminating the startup due to being unable to generate a healthy cash flow. They operated with monthly subscriptions, which were not enough for them to survive on the long-term. By converting this to enterprise sales, they generated a steady cash flow. C1 added to that: *"The lack of money is always a big problem."* The aspect of sufficient cash is impossible to reflect in the approach but is still paramount for startup survival.

While the team, culture, and cash are not reflected in the final GTM approach, we do stress on the importance of these elements. On an interesting note, a few startups mentioned having *luck* during their early stage startup, on one or several occasions. We hypothesise that it is not luck that made them achieve product-market fit, but hard work and structured persistence, because the cases still presented a structured approach during their early stage startup. Next, we will reflect on method engineering and discuss the threats to validity.

7.2.1 Reflecting on Method Engineering

Utilising method engineering for creating the GTM approach came with advantages and disadvantages. A recurring note from the case participants was the chronological order of the PDD. Case participants would explain that they iterated many times or performed it in a different order. This is partially a disadvantage of the nature of a PDD: its chronological nature. Studies show that startups work in an uncertain context that is rapidly evolving (Crowne, 2002b; Paternoster et al., 2014), whereas the PDD demonstrates structured activities. Therefore, the PDD could be not the ideal manner to translate a GTM approach to startups. However, if the startup finds

itself in a particular phase of the approach, the PDD does represent exemplary steps to take for the startup to proceed. A recommendation would be asking the startup a few 'basic' questions to assess in what phase it is, and subsequently present the 'best practice' activities to proceed with the startup. This also provides interactive feedback to the startup.
This problem sets the stage for future research regarding PDDs, but also meta-modelling as a whole. Future research should involve finding other methods to visualise the GTM approach in an interactive and non-chronological manner. Nowadays, studies are still reporting their results in the form of Process-Deliverable Diagrams (Spruit and Jagesar, 2016; Morschheuser et al., 2018; Ooms, Spruit, and Overbeek, 2019). They also emphasise the understandability for the one that is interpreting the PDD, independent of the domain that is modelled. However, future research of these studies do not call for an interactive manner of modelling. Furthermore, a decline is visible concerning studies published about method engineering (and specifically PDDs). There are some recent web-based meta-modelling platforms and tools defined in literature that can serve as example possibilities for an interactive PDD tool, for instance ADOxx (Bork et al., 2019), WAMS (Elmqvist, Malmheden, and Andreasson, 2019), and Stratus ML (Hamdaqa and Tahvildari, 2015). Future research could investigate the commonalities of these platforms and tools, and assess the strengths and weaknesses to create a novel, interactive meta-modelling platform.

7.2.2 Threats to Validity

In Section 2.2, we mentioned four types of validity: construct validity, internal validity, external validity, and reliability (Runeson and Höst, 2009; Yin, 2017). In this subsection, we reflect on the potential validity threats that came with this research, related to these validity types. **Construct validity** is concerned with the degree to which a test measures what it claims to be measuring. We applied two tactics to increase the construct validity: using multiple sources (cases) of evidence, and establishing a chain of evidence. The second tactic is to establish a chain of evidence: the reader of this study can follow the derivation of every piece of evidence, from initial research questions to eventual case study conclusions. The case study protocol contains the interview questions, and specific evidentiary sources are cited in the case study report, which results in clear traceability. Research related to method engineering has demonstrated the effectiveness of applying PDDs as a test measure. Through method engineering, we attempted to capture startup approaches as PDDs to provide a structured, strategic backbone. Furthermore, we demonstrated evident traceability during the process of creating the GTM approach by applying method engineering, in particular, method comparison and the overall creation of the reference approach.

Internal validity is paramount when causal relations are examined because they should be established correctly. If a factor affects an investigated factor, a third factor should not affect the other factor. During this research, we explicitly followed the protocol, also throughout the case studies, which resulted in the similar treatment of case participants. One threat we identified beforehand was lack of knowledge about PDDs by the case participants. However, we hypothesised that they had sufficient knowledge to evaluate the process diagram, especially after a brief introduction, which turned out to be true. It did occur that a case participant asked specifically about if-statements. However, it did not decrease the internal validity. This, and other questions that arose, were answered immediately, and due to case participants being asked to think aloud, no misunderstandings were noticed. **External validity**

deals with the possibility of generalising the findings that we discovered. We have established that the final model represents the sample startups well. Fourteen activities were marked as optional, and merely one activity was deleted. Therefore, we hypothesise that software startups that follow similar inclusion criteria listed in Section 2.3 can benefit from this approach. However, future research should re-evaluate the new model and especially with startups located outside of The Netherlands. It is interesting to demonstrate if the proposed activities of the GTM approach are bound to geographical constraints. One criticism that was mentioned for the external validity was decreased generalisability due to a small sample size, which is mitigated with twelve cases. The confidence is increased due to the repetition of the case study with a case study protocol reused for each case. By applying a case study protocol and having a relatively large (homogeneous) sample size, we attempt to increase the external validity (generalisability) of this research. Furthermore, in this research, no statistically representative sample is drawn from a population. It deals with an analytical generalisation: the results are extended to cases with common characteristics. Finally, **Reliability** explains to what extent the analysis and the data are dependent on the researcher. If another researcher conducts the same multiple-case study, the results should be the same. Therefore, we attempted to minimise the biases and errors in this study. This validity threat is partially mitigated, but one threat that remains is the coding of the interview results. This has been performed by one researcher, while the reliability increases if more researchers code the same texts, to assess if there are different opinions about constructs. However, we increased the reliability by documenting every followed procedure in this study.

7.2.3 Research Risks

In Section 2.5, we defined three research risks that we wanted to avoid and safeguard throughout the research. First, we considered the possibility of having **insufficient case participants**. We suggested searching actively within our network through, for instance, sending emails and contacting incubators nearby. We found twelve founders that were willing to participate in this research, which we considered sufficient for this research. Secondly, we discussed the risk of **not being able to extract a unique approach**. We would have been exposed to this risk if every founder applied a distinct strategy when building their startup. However, we avoided this risk because each startup that was interviewed applied an approach that was at least reasonably similar to that of other startups that participated. In the final approach, only one activity was excluded, while fourteen were marked as optional, and 33 were retained. This resulted in a unique reference approach that is relevant for software startups in The Netherlands. The final risk we presented was that **startups would not follow a structured approach**. Every software startup we interviewed did follow a structured approach, and it was vital for finding the problem-solution fit and product-market fit. While almost no startup follows a structured, chronological plan, as the GTM approach might suggest, they did perform activities in a particular order and recognised the phases in the model.

Chapter 8

Conclusions

This research proposed a GTM approach for software startups, which visualises guidelines aimed to increase their chance of success. It takes two main gateways for a startup into account: the problem-solution fit, based on the hypothesised customer problem and solution, and the product-market fit, when customers are being acquired. The GTM approach is based on four strategies defined in literature and is evaluated by twelve software startups that have found product-market fit. This chapter presents the conclusions of this study with respect to the research questions.

SQ1 *Which existing startup approaches are currently described in literature?*
- **1.1** *What insights are gathered from startup failures?*
- **1.2** *What is the influence of pivots within startups?*

The first subquestion is answered by laying a foundation with the contemporary software startup approaches that are relevant for this research and found in literature: The Lean Startup (Cooper and Vlaskovits, 2010; Blank, 2007), Design Thinking (Brown and Wyatt, 2010; Müller and Thoring, 2012), The Lean Product Playbook (Olsen, 2015), and The Startup Owner's Manual (Blank and Dorf, 2012). We showed that Lean principles are prominently present in contemporary startup literature. By showing that there are multiple types of failures and pivots startups can endure, we answered SQ1.1 and 1.2. These also play a prominent role in startup approaches: if failures can be foreseen in time, a pivot can be set into motion. The approaches laid a knowledge foundation for the method fragments that are used for the next research question.

SQ2 *Which method fragments are identified from startup approaches?*
- **2.1** *How are relevant method fragments identified within the scope of this research?*
- **2.2** *How can multiple strategies be evaluated by method comparison?*

The second subquestion is answered by applying method engineering to the identified startup approaches. We created Process-Deliverable Diagrams that represented the approaches, based on the method of van de Weerd, de Weerd, and Brinkkemper (2007). Next, we created a super approach that contained every method fragment of the approaches, by which SQ2.1 and SQ2.2 are answered. This approach was used as input for the method comparison.

SQ3 *How can a go-to-market approach be assembled using method fragments?*

Based on the work of van de Weerd, de Weerd, and Brinkkemper (2007), we performed method comparison to create a reference method. The relevant method fragments that resulted from the method comparison were used to build a final PDD: the GTM approach, which answered SQ3. This approach was then ready for a holistic multiple-case study evaluation.

SQ4 *How are go-to-market approaches applied within case studies?*

We introduced twelve case participants (co-founders) that found product-market fit with their software startups. For this, we used contemporary knowledge about performing holistic multiple-case studies (Runeson and Höst, 2009; Yin, 2011; Yin, 2017). Furthermore, the software startups that were included met our inclusion criteria defined in Section 2.3. We interviewed the case participants and provided them the GTM approach for evaluation. After processing the results and approach statistics, we answered SQ4, while mitigating threats to validity (Wohlin et al., 2012; Yin, 2017).

RQ *How can a go-to-market approach be assembled for innovative software startups?*

The main research question of this study has been answered by researching the subquestions mentioned above. From contemporary startup literature, a reference method was created through method comparison. This has been evaluated with twelve case participants, from which an adapted go-to-market approach resulted. The go-to-market approach evaluation showed us that innovative software startups apply **user-driven innovation strategies**: involving potential users, customers, or other stakeholders into the development process. Here we identified two critical gateways for startups to go through: the problem-solution fit and product-market fit. The **problem-solution fit**, which tests hypotheses of the customer problem, has proven to be of significant value, provided that it is tested with potential customers. When one or several customers have been attracted to the software product, the problem-solution fit has been found. Next, the startup needs to find the **product-market fit**, which requires the startup to operate in a pleasant and profitable market with a product that can satisfy that market. This is relatively easy if there is an accurate problem-solution fit in place because the startup knows exactly which potential customers should be approached. However, it might be the case at this point that another market segment proves to be more profitable than the initially chosen market segment. If the last fit has been found, to go-to-market approach is 'completed'. The startup can create new customers and start building the company by scaling the organisation, i.e. become a scale-up. Marc Andreessen describes this point in time as follows: *"The life of any startup can be divided into two parts — before product-market fit and after product-market fit."*

8.1 Future Work

In this section, we make the distinction between future works for academia and the software startup industry because both domains face different types of challenges.

8.1.1 Academia

The approach serves as a starting point for additional method fragments to be included. Future research can identify these method fragments, which should be evaluated with software startups as well. Other future research could focus on translating a complete strategy instead of only an approach. Adding to that, it is interesting to translate it in a non-chronological, interactive manner and visually more appealing, as suggested in Section 7.2.1. This relates to more in-depth research about existing meta-modelling platforms and to assess what the commonalities and differences are. This then paves the way for a novel meta-modelling platform or tool to create interactive PDDs. Specifically related to creating an interactive tool that is relevant for this research (and other meta-modelling research), a suggestion would be asking the startup a few questions to assess in what phase it is, and subsequently present the activities it could best follow to proceed with the startup. Furthermore, it is also interesting to study the *luck*-factor of startups, which some cases brought up during the interviews. Finally, future research could involve applying the GTM approach to new software startups. While the evaluation of the GTM approach proved promising in this study, further empirical validation is required to confirm these exploratory findings. This could also be software startups located outside of The Netherlands to assess if the early stage activities are bound to geographical constraints. Suggestions for future research could be either similar to this research (a multiple-case study), or surveys that measure the perceived ease of use and usefulness via the Method Evaluation Model of Moody (2003), or technical action research. Triangulation of these techniques could result in feedback that is used as input for the GTM approach to refine and improve it. This type of research is also related to future research concerning the software startup industry, which is discussed in the next subsection.

8.1.2 Software Startup Industry

The GTM approach can provide structured guidance to software startups. One specific market where guidance is being delivered to startups is that of incubators and accelerators. Incubators are companies that create a supportive environment that is conducive to the development of new businesses (Bergek and Norrman, 2008), where accelerators are more private-sector versions of incubators (Dempwolf, Auer, and D'Ippolito, 2014). They provide a variety of support and assistance, and they both work with startup businesses that are usually for-profit and contain high-growth potential. For future research, it would be interesting for incubators and accelerators to use the GTM approach during their programs, and for researchers to identify whether there are significant additional activities that should be added to the model. Future research could also include surveying or interviewing startup mentors that are working in this domain to evaluate the model further.

Bibliography

Bajwa, Sohaib Shahid et al. (2016). "How do software startups pivot? empirical results from a multiple case study". In: *International Conference of Software Business*. Springer, pp. 169–176.

Bajwa, Sohaib Shahid et al. (2017). ""Failures" to be celebrated: an analysis of major pivots of software startups". In: *Empirical Software Engineering* 22.5, pp. 2373–2408.

Basulto, Dominic (2015). *The 7 greatest pivots in tech history - The Washington Post*. https://www.washingtonpost.com/news/innovations/wp/2015/07/02/the-7-greatest-pivots-in-tech-history/?noredirect=on&utm_term=.b14612359c3b. (Accessed on 01/14/2019).

Bergek, Anna and Charlotte Norrman (2008). "Incubator best practice: A framework". In: *Technovation* 28.1-2, pp. 20–28.

Blank, Steve (2007). *The four steps to the epiphany: successful strategies for products that win*. Cafepress.com.

Blank, Steve and Bob Dorf (2012). *The startup owner's manual: The step-by-step guide for building a great company*. K&S Ranch.

Bork, Dominik et al. (2019). "An open platform for modeling method conceptualization: The OMiLAB digital ecosystem". In: Association for Information Systems.

Brown, Tim and Jocelyn Wyatt (2010). "Design thinking for social innovation". In: *Development Outreach* 12.1, pp. 29–43.

Busenitz, Lowell W (1999). "Entrepreneurial risk and strategic decision making: It'sa matter of perspective". In: *The Journal of Applied Behavioral Science* 35.3, pp. 325–340.

Cooper, Brant and Patrick Vlaskovits (2010). "Entrepreneur's Guide to Customer Development". In:

Crowne, Mark (2002a). "Why software product startups fail and what to do about it". In: *Proceedings of the international engineering management conference (IEMC)*, pp. 338–343.

— (2002b). "Why software product startups fail and what to do about it". In: *Proceedings of the international engineering management conference (IEMC)*, pp. 338–343.

Dande, Anuradha et al. (2014). "Software startup patterns-an empirical study". In: *Tampereen teknillinen yliopisto. Tietotekniikan laitos. Raportti-Tampere University of Technology. Department of Pervasive Computing. Report; 4*.

Dempwolf, C Scott, Jennifer Auer, and Michelle D'Ippolito (2014). "Innovation accelerators: Defining characteristics among startup assistance organizations". In: *Small Business Administration*, pp. 1–44.

Dorst, Kees (2011). "The core of 'design thinking' and its application". In: *Design studies* 32.6, pp. 521–532.

Dul, Jan and Tony Hak (2007). *Case study methodology in business research*. Routledge.

Edison, Henry, Xiaofeng Wang, and Pekka Abrahamsson (2015). "Lean startup: why large software companies should care". In: *Scientific Workshop Proceedings of the XP2015*. ACM, p. 2.

Eisenmann, Thomas R, Eric Ries, and Sarah Dillard (2012). "Hypothesis-driven entrepreneurship: The lean startup". In:

Elmqvist, Hilding, Martin Malmheden, and Johan Andreasson (2019). "A Web Architecture for Modeling and Simulation". In: *Proceedings of the 2nd Japanese Modelica Conference Tokyo, Japan, May 17-18, 2018*. 148. Linköping University Electronic Press, pp. 255–260.

Ensley, Michael D, Keith M Hmieleski, and Craig L Pearce (2006). "The importance of vertical and shared leadership within new venture top management teams: Implications for the performance of startups". In: *The leadership quarterly* 17.3, pp. 217–231.

European Commission (2003). *What is an SME? | Internal Market, Industry, Entrepreneurship and SMEs*. http://ec.europa.eu/growth/smes/business-friendly-environment/sme-definition_en. (Accessed on 02/01/2019).

Furr, Nathan and Ahlsrom Paul (2011). *Nail it Then Scale it: the Entrepreneur's Guide to Creating and Managing Breakthrough Innovation*. S.I.: NISI Institute.

Furr, Nathan R and Jeff Dyer (2014). *The innovator's method: Bringing the lean startup into your organization*. Harvard Business Press.

Giardino, Carmine, Xiaofeng Wang, and Pekka Abrahamsson (2014). "Why early-stage software startups fail: a behavioral framework". In: *International Conference of Software Business*. Springer, pp. 27–41.

Giardino, Carmine et al. (2015). "Key challenges in early-stage software startups". In: *International Conference on Agile Software Development*. Springer, pp. 52–63.

Hamdaqa, Mohammad and Ladan Tahvildari (2015). "Stratus ML: A layered cloud modeling framework". In: *2015 IEEE International Conference on Cloud Engineering*. IEEE, pp. 96–105.

Holmquist, Lars Erik (2004). "User-driven innovation in the future applications lab". In: *CHI'04 Extended Abstracts on Human Factors in Computing Systems*. ACM, pp. 1091–1092.

Hong, Shuguang, Geert van den Goor, and Sjaak Brinkkemper (1993). "A formal approach to the comparison of object-oriented analysis and design methodologies". In: *System Sciences, 1993, Proceeding of the Twenty-Sixth Hawaii International Conference on*. Vol. 4. IEEE, pp. 689–698.

Hui, Alexis (2013). "Lean change: Enabling agile transformation through lean startup, kotter and kanban: An experience report". In: *2013 Agile Conference (AGILE)*. IEEE, pp. 169–174.

Ian Burke, G and Denise G Jarratt (2004). "The influence of information and advice on competitive strategy definition in small-and medium-sized enterprises". In: *Qualitative Market Research: An International Journal* 7.2, pp. 126–138.

Jansen, S and Roderick van Cann (2012). *Software business start-up memories: Key decisions in success stories*. Springer.

Järvi, Antero, Ville Taajamaa, and Sami Hyrynsalmi (2015). "Lean software startup–An experience report from an entrepreneurial software business course". In: *International Conference of Software Business*. Springer, pp. 230–244.

Kane, Tim J (2010). "The importance of startups in job creation and job destruction". In:

Khanna, Dron, Anh Nguyen-Duc, and Xiaofeng Wang (2018). "From MVPs to pivots: a hypothesis-driven journey of two software startups". In: *International Conference of Software Business*. Springer, pp. 172–186.

Lethbridge, Timothy C, Susan Elliott Sim, and Janice Singer (2005). "Studying software engineers: Data collection techniques for software field studies". In: *Empirical software engineering* 10.3, pp. 311–341.

MacCormack, Alan (2001). "How internet companies build software". In: *MIT Sloan Management Review* 42.2, pp. 75–84.

MacMillan, Ian C, Lauriann Zemann, and PN Subbanarasimha (1987). "Criteria distinguishing successful from unsuccessful ventures in the venture screening process". In: *Journal of business venturing* 2.2, pp. 123–137.

Marmer, Max et al. (2011). "Startup genome report extra: Premature scaling". In: *Startup Genome* 10, pp. 1–56.

Martin, Roger and Roger L Martin (2009). *The design of business: Why design thinking is the next competitive advantage*. Harvard Business Press.

Maurya, Ash (2012a). *Running lean: iterate from plan A to a plan that works*. " O'Reilly Media, Inc."

— (2012b). "Why lean canvas vs business model canvas". In: *Available in: http://practicetrumpstheory.com/why-lean-canvas*.

Moody, Daniel L (2003). "The method evaluation model: a theoretical model for validating information systems design methods". In: *ECIS 2003 proceedings*, p. 79.

Morschheuser, Benedikt et al. (2018). "How to design gamification? A method for engineering gamified software". In: *Information and Software Technology* 95, pp. 219–237.

Morse, Janice M (2010). "Sampling in grounded theory". In: *The SAGE handbook of grounded theory*, pp. 229–244.

Müller, Roland M and Katja Thoring (2012). "Design thinking vs. lean startup: A comparison of two user-driven innovation strategies". In: *Leading through design* 151.

Nobel, Carmen (2011). "Teaching a 'Lean Startup'Strategy". In: *HBS Working Knowledge*, pp. 1–2.

Nowak, Michael J and Charles E Grantham (2000). "The virtual incubator: managing human capital in the software industry". In: *Research Policy* 29.2, pp. 125–134.

NVivo (n.d.). *What is NVivo? | NVivo*. https://www.qsrinternational.com/nvivo/what-is-nvivo. (Accessed on 04/03/2019).

Olsen, Dan (2015). *The lean product playbook: How to innovate with minimum viable products and rapid customer feedback*. John Wiley & Sons.

OMG (2003). *United Modeling Language 2.0 Proposal*. https://www.omg.org/spec/UML/2.0/About-UML/. (Accessed on 03/26/2019).

Ooms, Richard, Marco R Spruit, and Sietse Overbeek (2019). "3PM Revisited: Dissecting the Three Phases Method for Outsourcing Knowledge Discovery". In: *International Journal of Business Intelligence Research (IJBIR)* 10.1, pp. 80–93.

Osterwalder, Alexander et al. (2014). *Value proposition design: How to create products and services customers want*. John Wiley & Sons.

Paternoster, Nicolò et al. (2014). "Software development in startup companies: A systematic mapping study". In: *Information and Software Technology* 56.10, pp. 1200–1218.

Plattner, Hasso, Christoph Meinel, and Ulrich Weinberg (2009). *Design thinking*. Springer.

Rasmussen, Erik Stavnsager and Stoyan Tanev (2015). "The emergence of the lean global startup as a new type of firm". In: *Technology Innovation Management Review* 5.11.

Ries, Eric (2011). *The Lean Startup: How Constant Innovation Creates Radically Successful Businesses. Kindle Edition*. Penguin Books Limited. New York, NY.

Ripsas, Sven, Birte Schaper, and Steffen Tröger (2018). "A startup cockpit for the Proof-of-Concept". In: *Handbuch Entrepreneurship*, pp. 1–17.

Ripsas, Sven and Steffen Tröger (2014). *Deutscher Startup Monitor# DSM*. KPMG AG.

Runeson, Per and Martin Höst (2009). "Guidelines for conducting and reporting case study research in software engineering". In: *Empirical software engineering* 14.2, p. 131.

Savolainen, Paula, Jarmo J Ahonen, and Ita Richardson (2012). "Software development project success and failure from the supplier's perspective: A systematic literature review". In: *International Journal of Project Management* 30.4, pp. 458–469.

Schutjens, Veronique AJM and Egbert Wever (2000). "Determinants of new firm success". In: *Papers in Regional Science* 79.2, pp. 135–153.

Smagalla, David (2004). "The truth about software startups". In: *MIT Sloan Management Review* 45.2, p. 7.

Somerville, Heather (2018). *Uber narrows loss but still a long way from profitability | Reuters*. https://reut.rs/2H8ctPh. (Accessed on 02/04/2019).

Spruit, Marco and Raj Jagesar (2016). "Power to the People!" In: *Proceedings of the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*. SCITEPRESS-Science and Technology Publications, Lda, pp. 400–406.

Stampfl, Georg (2015). *The process of business model innovation: an empirical exploration*. Springer.

Sutton, Stanley M. (2000). "The Role of Process in a Software Start-up". In: *IEEE Softw.* 17.4, pp. 33–39. ISSN: 0740-7459. DOI: 10.1109/52.854066. URL: https://doi.org/10.1109/52.854066.

Teddlie, Charles and Fen Yu (2007). "Mixed methods sampling: A typology with examples". In: *Journal of mixed methods research* 1.1, pp. 77–100.

Terho, Henri et al. (2015). "Ways to cross the rubicon: pivoting in software startups". In: *International Conference on Product-Focused Software Process Improvement*. Springer, pp. 555–568.

Thoring, Katja, Roland M Müller, et al. (2011). "Understanding design thinking: A process model based on method engineering". In: *DS 69: Proceedings of E&PDE 2011, the 13th International Conference on Engineering and Product Design Education, London, UK, 08.-09.09. 2011*, pp. 493–498.

Tronstad, Russell (2008). "Evaluating market size". In: *Niche Markets: Assessment and Strategy Development for Agriculture*, pp. 2–08.

van de Weerd, Inge and Sjaak Brinkkemper (2009). "Meta-modeling for situational analysis and design methods". In: *Handbook of research on modern systems analysis and design technologies and applications*. IGI Global, pp. 35–54.

van de Weerd, Inge, Stefan de Weerd, and Sjaak Brinkkemper (2007). "Developing a reference method for game production by method comparison". In: *Situational method engineering: Fundamentals and experiences*. Springer, pp. 313–327.

van de Weerd, Inge et al. (2006). "A situational implementation method for web-based content management system-applications: method engineering and validation in practice". In: *Software process: improvement and practice* 11.5, pp. 521–538.

Wagner, Kurt (2018). *Twitter just reported its first profitable quarter ever, but didn't add any new users in Q4 - Recode*. https://bit.ly/2DtV4Qe. (Accessed on 02/04/2019).

Wieringa, Roel J (2014). *Design science methodology for information systems and software engineering*. Springer.

Wohlin, Claes and Aybüke Aurum (2015). "Towards a decision-making structure for selecting a research design in empirical software engineering". In: *Empirical Software Engineering* 20.6, pp. 1427–1455.

Wohlin, Claes et al. (2012). *Experimentation in software engineering*. Springer Science & Business Media.

Womack, James P and Daniel T Jones (1997). "Lean thinking-banish waste and create wealth in your corporation". In: *Journal of the Operational Research Society* 48.11, pp. 1148–1148.

Yin, Robert K (2011). *Applications of case study research*. Sage Publishing.

— (2017). *Case study research and applications: Design and methods*. Sage Publishing.

Yu, Ya-Wen et al. (2012). "Entrepreneurial Success for High-Tech Start-Ups–Case Study of Taiwan High-Tech Companies". In: *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*. IEEE, pp. 933–937.

Appendix A

Interview Protocol

A.1 Informed Consent

We have asked you to participate in a research about software startup approaches, specifically about how **[Company]** applied certain approaches in the past, sharing their experiences. This research is conducted by Thomas Alflen, under the supervision of Slinger Jansen and Sjaak Brinkkemper.

The interview will take approximately one hour and shall be recorded. Furthermore, during the interview, I (Thomas Alflen) will take extra notes if required. The recording will only be used within this research in order to transcribe and analyse the interview.

For any questions afterwards, you are able to contact Thomas Alflen by emailing (t.alflen@uu.nl).

I have read this informed consent and I agree,

Name:

Date:

Signature:

A.2 Questions

[Interviewer starts the recording]

Introduction

I am a student from Utrecht University, enrolled in the Master of Business Informatics. For my thesis, I would like to gather data regarding software startup approaches. I have proposed a software startup approach that struggling software startups can utilise in the future to guide them where needed. However, this model requires to be evaluated by real case studies, which is why I contacted **[Company]**. In particular, I would like to know exactly which approaches have been taken in the past until where you are now. After that, I shall propose my model which you can evaluate; we can discuss any adaptations required in your opinion. Furthermore, we shall try to couple certain actions you have taken in the past to fragments of my model. By doing this, you shall provide an expert opinion upon my model, which is used as an evaluation.

The results are completely anonymous and they are only analysed by me and the aforementioned supervisors. As stated before, the complete interview will be approximately one hour.

Questions

When was the company founded by its owners?
This information might be used for exploratory temporal analysis.

Please tell me about the route that you have taken since then.
To let the interviewee speak freely about their journey since starting the company. Important moments:

- Discovering problem-solution fit (customer problems, needs)
- Discovering product-market fit
- First sale
- First returns
- First profit
- Occurrence of investments

Can you tell me about your Minimum Viable Product at that time, if there was any?
Let the interviewee talk about the MVP and what was needed for it:

- Value proposition
- MVP feature set
- MVP (first) prototype
- Tests with customers; testing hypotheses?
- How did it relate to the target customer?

At the time, did you pursue any approach or strategy that you know of?

Provides a general picture of the approaches of the startup. Ask about the following strategies:

- The Lean Startup
- Design Thinking
- The Lean Product Playbook
- The Startup Owner's Manual

How beneficial would pursuing a structured strategy or method be for a startup?

Try to assess the usefulness.

If a certain approach or strategy was/is pursued: Can you tell me about it as detailed as possible?

Provides the basis of the approach the startup has taken in the past. Important topics:

- What steps have you taken
- Where there any hurdles along the way
- What important choices did you encounter
- Where there any significant changes over time that you can think of

How and when did you start upscaling/building the company?

To clearly get into view what happens after the product-market fit. Important:

- Any strategies/approaches used?

Were there any failures along the road?

Mainly to document failed projects or ideas.

- Failed ideas
- Failed projects
- Connect this to pivots in the next question

Are you aware of any pivots your company had to pursue?

Try to capture the pivots as clearly as possible.

- Try to connect each pivot to a type in Table 3.2
- Try to identify pivot triggers for each pivot.

Providing the Go-to-market Approach

[Shows a physical model of the go-to-market approach] For the second part of this interview, I propose to you [the interviewee] the go-to-market approach that we have designed. In an interactive manner, then I will attempt to put the answers you [the interviewee] provided into the context of the go-to-market approach. We shall go over every phase and subactivities to see if we can obtain some resemblance, or see if there are activities that are deemed obsolete. You [the interviewee] are free to make any adjustments on the physical model using a pen.

Do you think such a method would have been beneficial for you to use?

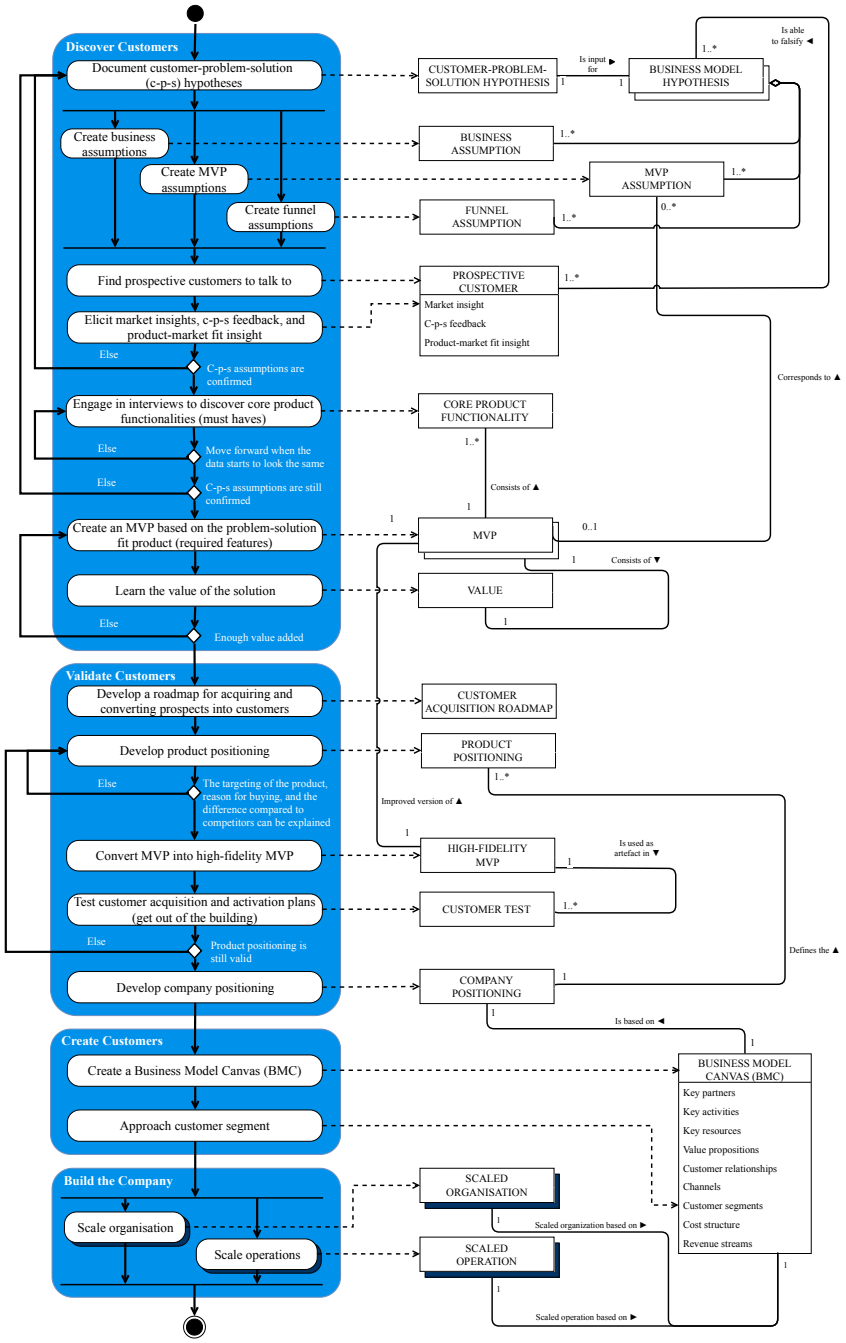
When we are done, the interview is over and you are free to ask any additional questions.

[Interviewer stops the recording]

Appendix B

Process-Deliverable Diagrams

B.1 The Lean Startup and Customer Development



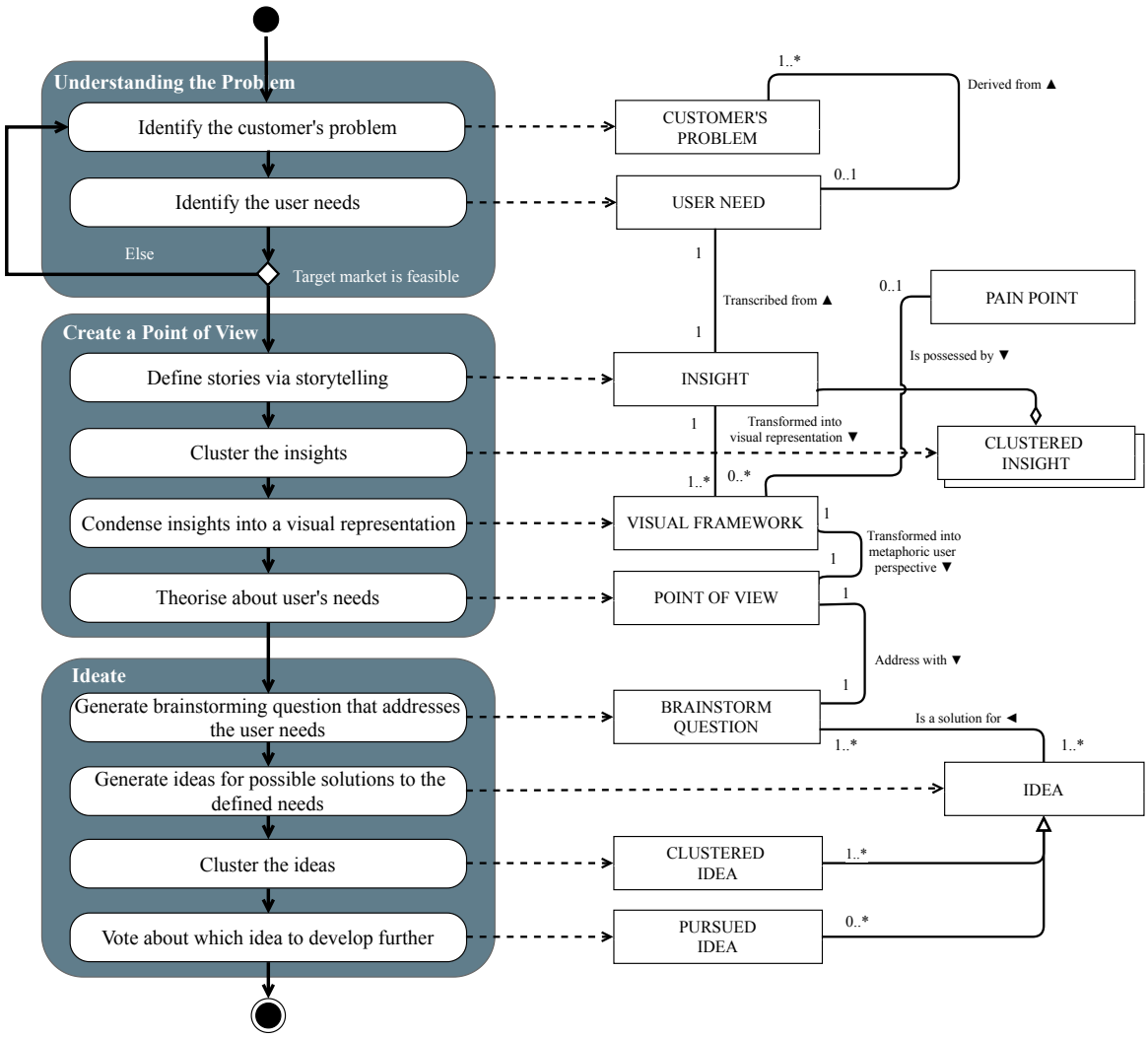
Activity	Sub-Activity	Description
Discover Customers	Document customer-problem-solution (c-p-s) hypotheses	At the start of the process, hypothesise about CUSTOMER-PROBLEM-SOLUTION HYPOTHESIS and document these. This solution should solve a particular customer problem.
	Create business assumptions	Make explicit what kind of BUSINESS ASSUMPTIONS are going to come with this solution.
	Create MVP assumptions	Think about what kind of MVP you are making for the customer, which is an MVP ASSUMPTION.
	Create funnel assumptions	Make assumptions about which funnels you are going to use to reach your PROSPECTIVE CUSTOMERs.
	Find prospective customers to talk to	One of the most important parts is to talk to PROSPECTIVE CUSTOMERs. These can affirm or falsify your hypotheses and assumptions.
	Elicit market insights, c-p-s feedback, and product-market fit insight	This is an activity that is performed while talking to the PROSPECTIVE CUSTOMER. Move forward if c-p-s assumptions are confirmed.
	Engage in interviews to discover core product functionalities (must haves)	One should focus on only building the must haves: the CORE PRODUCT FUNCTIONALITY. Repeat if new data is gathered, else move forward. If the c-p-s assumptions are falsified, document new customer-problem-solution hypotheses.
	Create an MVP based on the problem-solution fit (required features)	The MVP should only consist of the previously gathered CORE PRODUCT FUNCTIONALITY, thus the required features.
	Learn the value of the solution	Assess whether the MVP brings enough VALUE to the PROSPECTIVE CUSTOMER. If this is not the case, the MVP should be adapted in accordance with the required features.
Validate Customers	Develop a road map for acquiring and converting prospects into customers.	Previously, you have gathered PROSPECTIVE CUSTOMERs, which you should now convert to real customers, for which you should make a CUSTOMER ACQUISITION ROAD MAP.
	Develop product positioning	It is essential to be able to explain the targeting of the product, the reason for buying, and the difference compared to competitors (Unique Selling Points). This comprises the PRODUCT POSITIONING.
	Convert MVP into high-fidelity MVP	On the sideline, the MVP should be converted to a HIGH-FIDELITY MVP, which can be used for CUSTOMER TESTs and sales.
	Test customer acquisition and activation plans (get out of the building)	In this stage, it is wise to get completely out of the building and to test and maybe sell your HIGH-FIDELITY MVP in CUSTOMER TESTs.
	Develop company positioning	Only when the PRODUCT POSITIONING has proven to be valid still, you should develop the COMPANY POSITIONING.
Create Customers	Create a Business Model Canvas (BMC)	A BUSINESS MODEL CANVAS (BMC) describes key elements of your company and to create new customers, it is good to list those elements.
	Approach customer segment	One of the properties of a BMC is the customer segment, which you now have defined. The HIGH-FIDELITY MVP works as expected, the hypotheses are correct, and it is time to approach the CUSTOMER SEGMENT and sell the product.
Build the Company	Scale organisation	Building the company is outside the scope of this work and consists of new sets of subactivities that ultimately lead to a SCALED ORGANISATION.
	Scale operations	Parallel to obtaining a SCALED ORGANISATION, you also would want to work to SCALED OPERATIONs, which is as well outside of the scope of this research.

TABLE B.1: The activity table of The Lean Startup and Customer Development.

Concept	**Description**
CUSTOMER-PROBLEM-SOLUTION HYPOTHESIS	A certain solution that has been made up for a certain customer problem, in the form of an hypothesis.
BUSINESS MODEL HYPOTHESIS	This open concept has several underlying assumptions, which make up for the hypothesized business model.
BUSINESS ASSUMPTION	Basic assumptions that you make about your business, for instance about finance, customers, and profit.
MVP ASSUMPTION	Assumptions about your first MVP, based on your hypothesised customer-problem-solution.
FUNNEL ASSUMPTION	Assumptions about how to reach your customer base, via which funnel(s).
PROSPECTIVE CUSTOMER	A customer that fits your hypothesis and provides feedback upon it. It also ensures insights about your prospective market and the product market fit.
CORE PRODUCT FUNCTIONALITY	During interviews you should discover the core functionalities of the product (MVP), thus the must haves.
MVP	The MVP only consists of those features that the customer must have: the core functionalities. This should resemble the MVP assumption.
VALUE	The MVP brings a certain value to the prospective customers which you should learn and should be significant to be able to sell it.
CUSTOMER ACQUISITION ROADMAP	A roadmap that defines how to convert prospecting customers into real paying customers.
PRODUCT POSITIONING	This defines the targeting of the product, the reason for buying, and the difference compared to competitors.
HIGH-FIDELITY MVP	The MVP is transformed into a fully working prototype, a more detailed, sellable, end product.
CUSTOMER TEST	The high-fidelity MVP is used for tests with customers, to assess whether they are willing to buy it and to ensure the product-positioning is still valid.
COMPANY POSITIONING	The company positioning should clearly outline what the business should do to market its service or product to its customers.
BUSINESS MODEL CANVAS (BMC)	A canvas that consists of key partners, activities, and resources, value propositions, customer relationships, channels, customer segments, cost structure, and revenue streams.
SCALED ORGANISATION	Several sub-concepts define a scaled organisation, that is not within the scope of this research. Implicitly, the product-market fit at this point has already been found.
SCALED OPERATION	Several sub-concepts define the scaled operations, that is also not within the scope of this research. It is, however, mentioned here since it plays a crucial role into building the company.

TABLE B.2: The concept table of The Lean Startup and Customer Development.

B.2 Design Thinking



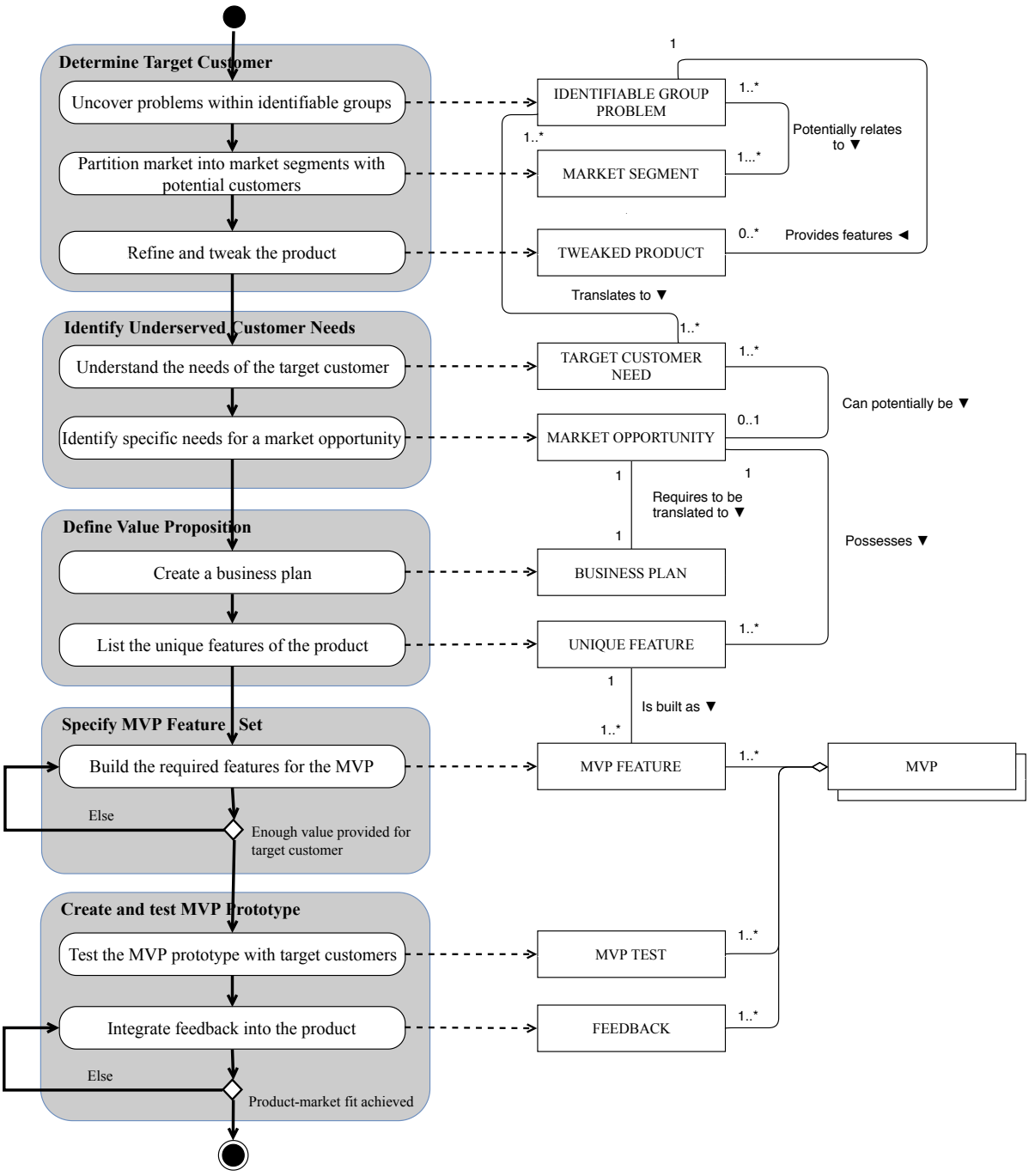
Activity	**Sub-Activity**	**Description**
Understanding the Problem	Identify the customer's problem	Identify an existing CUSTOMER PROBLEM and try to fully understand this problem.
	Identify the user needs	Clearly understand the USER NEED that comes with the CUSTOMER'S PROBLEM via qualitative research (interviews and observations).
Create a Point of View	Define stories via storytelling	Bring everyone on the same level by exchanging research results with the use of storytelling, which leads to INSIGHTs about the USER NEEDs
	Cluster the insights	By structuring the INSIGHTs into similar groups, you get CLUSTERED INSIGHTs.
	Condense into a visual representation	To fully understand the USER NEEDs, Condense INSIGHTs into a VISUAL FRAMEWORK (venn diagram, persona, or something else). This framework contains certain PAIN POINTs which means there is room for improvement on that point.
	Theorise about user's needs	Search for metaphors and analogies that concern the USER NEEDs, which results in a POINT OF VIEW.
Ideate	Generate brainstorming question that addresses the user needs	The BRAINSTORM QUESTION surrounds solving the USER NEED, it might start with: "How might we...".
	Generate ideas for possible solutions to the defined needs	When answering the BRAINSTORM QUESTIONs, you obtain certain IDEAs that provide a solution to the aforementioned USER NEED.
	Cluster the ideas	Structure the IDEAs to, for instance, their usefulness or feasibility. You then obtain CLUSTERED IDEAs.
	Vote about which idea to develop further	Only IDEAs that meet certain criteria become PURSUED IDEAs and the other IDEAs are discarded.

TABLE B.3: The activity table of Design Thinking, based on the work of Thoring and Müller (2011).

Concept	**Description**
CUSTOMER'S PROBLEM	Become an expert of the proposed customer's problem by acquiring knowledge about it via newspapers, internet, and other sources.
USER NEED	A customer's problem translates to a user need, based on observations and interviews. Users might not know that they have this problem need beforehand.
INSIGHT	By sharing knowledge via storytelling you obtain one or more insights about the user needs.
CLUSTERED INSIGHT	Insights are clustered according to specific themes. This is to identify certain patterns derived from the insights.
VISUAL FRAMEWORK	Condensed insights, which could result in, for instance, Venn diagrams, a causal map, or persona.
PAIN POINT	Certain points derived from the visual framework, which denotes room for improvement (translated from the user need).
POINT OF VIEW	The point of view is a (verbalised) micro-theory about the identified problem and user needs.
BRAINSTORM QUESTION	Building upon the point of view, the brainstorming question addresses the user need with questions like "How might we ..."?
IDEA	Ideas are generated during brainstorming sessions that address the brainstorming questions.
CLUSTERED IDEA	Ideas are clustered upon their feasibility and usefulness.
PURSUED IDEA	The ideas that are not discarded are further pursued.

TABLE B.4: The concept table of Design Thinking, based on the work of Thoring and Müller (2011).

B.3 The Lean Product Playbook



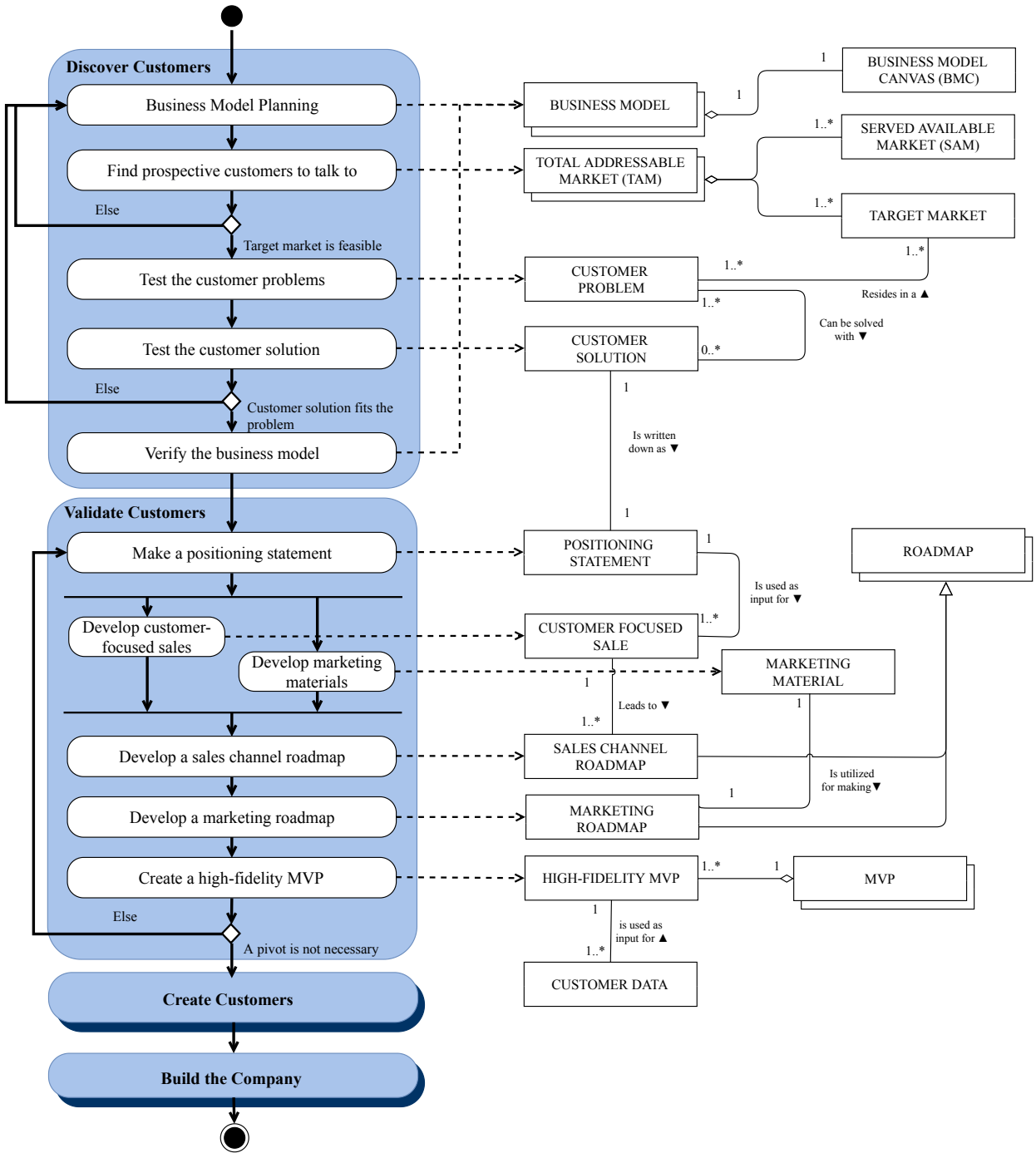
Activity	Sub-Activity	Description
Determine Target Customer	Uncover problems within identifiable groups	To first determine the target customer, there should be a clear problem that relates to a group of people, which translates to an IDENTIFIABLE GROUP PROBLEM.
	Partition market into market segments with potential customers	This step ensures that MARKET SEGMENTs are created from the problem(s) the group encounters. This refines the idea of your potential customer.
	Refine and tweak the product	According to the knowledge you gathered about your potential customer, you should refine the MVP you created and creat a TWEAKED PRODUCT.
Identify Underserved Customer Needs	Understand the needs of the target customer	The IDENTIFIABLE GROUP PROBLEM translates to a certain TARGET CUSTOMER NEED which you are attempting to satisfy. For this, you need to thoroughly understand that need.
	Identify specific needs for a market opportunity	Not every TARGET CUSTOMER NEED translates to a MARKET OPPORTUNITY. Therefore, this step ensures you identify a need that corresponds to one or several MARKET OPPORTUNITIES.
Define Value Proposition	Create a business plan	Create a BUSINESS PLAN that at least describes the benefits you are providing to customers and what you are doing better compared to competitors.
	List the unique features of the product	The UNIQUE FEATUREs are features that make your product unique compared to other similar products and might relate to your unique selling point(s).
Specify MVP Feature Set	Build the required features for the MVP	Create an MVP with the FEATURES that are 'must haves'. Only the necessary features that result in a product that is functional, reliable, usable, and delightful.
Create and Test MVP Prototype	Test the MVP prototype with target customers	TEST the MVP with customers to assess whether your product satisfies the TARGET CUSTOMER NEED.
	Integrate feedback into the product	Integrate FEEDBACK into the MVP and confirm if product-market fit is achieved after the integration; learn from the TEST and iterate if required.

TABLE B.5: The activity table of The Lean Product Playbook, based on the work of Olsen (2015).

Concept	**Description**
IDENTIFIABLE GROUP PROBLEM	A problem (or multiple problems) that is relatable to a group (or multiple groups) of people. This sets the stage for a target customer and a need that is to be satisfied.
MARKET SEGMENT	The problem can live in multiple market segments and not every segment has to be interesting for you to serve. Therefore, market segments need to be defined.
TWEAKED PRODUCT	According to the definition of your target customer, the product requires to be tweaked and refined.
TARGET CUSTOMER NEED	A customer problem translates to a need that requires to be fulfilled by you. To satisfy this, you need to fully understand the need.
MARKET OPPORTUNITY	It may be the case that the need translates to a market opportunity, which is beneficial. However, this does not have to be the case.
BUSINESS PLAN	A plan that describes how you are performing your business, usually a 'living' document. At the very least, it describes the benefits you provide to customers and what you are doing better compared to competitors.
UNIQUE FEATURE	Before developing the MVP, the unique features it requires need to be clear and listed. These are usually also the 'must haves' and could correspond to your unique selling point(s).
MVP FEATURE	A part (feature) of the MVP; usually a 'must have'. These are derived from the needs of the target customer segment.
MVP	The Minimum Viable Product that has in this case features, tests, and feedback.
MVP TEST	A certain test that shows if your MVP satisfies the target customer need.
FEEDBACK	The MVP tests will always result in feedback which can be used to refine the MVP accordingly. If the feedback indicates that the target market will use the product, product-market fit has been achieved.

TABLE B.6: The concept table of The Lean Product Playbook, based on the work of Olsen (2015).

B.4 The Startup Owner's Manual



Activity	**Sub-Activity**	**Description**
Discover Customers	Business model planning	BUSINESS MODEL planning lets the entrepreneurs think about how to make money. The authors recommend using the BUSINESS MODEL CANVAS (BMC) tool to do so.
	Find prospective customers to talk to	To assess your TOTAL ADDRESSABLE MARKET (TAM) and to see if your hypothesised BUSINESS MODEL is actually going to work. Within the TAM your TARGET MARKET is especially important to discover and address.
	Test the customer problems	Within the TARGET MARKET there is a CUSTOMER PROBLEM. You need to test if this is really an existing CUSTOMER PROBLEM and for this you need to go outside and test with the TARGET MARKET.
	Test the customer solution	The CUSTOMER'S SOLUTION is based on the existing problem and should also been tested on its effectiveness. Does it really fit the CUSTOMER PROBLEM and is it solved with your solution?
	Verify the business model	If the solution fits, it might still be questionable if you found a product-market fit. Furthermore, is it possible and feasible to make money and scale?
Validate customer	Make a positioning statement	The POSITIONING STATEMENT should be catchy and refer to why customers should buy your product.
	Develop customer-focused sales	Acquire customers by creating awareness, consideration, interest, and eventually a CUSTOMER FOCUSED SALE. The authors recommend bringing sales people on board for this.
	Develop marketing materials	The MARKETING MATERIALS should be largely derived from the hypothesised information that you generated during the customer discovery phase.
	Develop a sales channel roadmap	The SALES CHANNEL ROADMAP covers the relationships in the distribution channel, businesses in the organisational food chain, and how money moves and is going to move in the channels.
	Develop a marketing roadmap	Parallel on developing your SALES ROADMAP you develop your MARKETING ROADMAP and collect necessary marketing materials to sell your product.
	Create a high-fidelity MVP	The HIGH-FIDELITY MVP will have additional features compared to the previous MVP. Not being complete nor finished, it is more polished. For this, as much CUSTOMER DATA should be gathered and used as input.
Create Customers		If you arrive in the Create Customer phase you build upon the company's initial sales success. The product-market fit has already been achieved and it is time for scaling. Therefore, this is a closed activity.
Build the Company		The startup has found a scalable and repeatable business model when it has arrived in this phase and it has in fact become a company. Therefore, this is also a closed activity.

TABLE B.7: The activity table of The Startup Owner's Manual, based on the work of Blank and Dorf (2012).

Concept	Description
BUSINESS MODEL	Describes how you are going to make money. This is with the aid of making a Business Model Canvas (BMC).
BUSINESS MODEL CANVAS	A canvas that consists of key partners, activities, and resources, value propositions, customer relationships, channels, customer segments, cost structure, and revenue streams.
TOTAL ADDRESSABLE MARKET (TAM)	The market that is approachable by you and consists of the Served Available Market and the Target Market.
SERVED AVAILABLE MARKET (SAM)	The share of the total addressable market that can be served or fulfilled by all businesses that offer this particular product or service.
TARGET MARKET	A part of the TAM where your customer problem resides in. This is also the market that you are going to approach and sell your product or service to.
CUSTOMER PROBLEM	This clearly resides in your target market and is a problem that is not yet solved or not enough. You are going to test if this problem is really a customer problem and needs solving.
CUSTOMER SOLUTION	To solve the customer problem a certain customer solution requires to be created and tested with those same customers.
POSITIONING STATEMENT	A message that should be compelling and explaining why people should buy your service or product.
CUSTOMER FOCUSED SALE	You as a seller should take the viewpoint of the customer with the aid of the previously gathered information, thus putting yourself in the shoes of the customer.
MARKETING MATERIAL	This includes any item that is used for communicating your marketing to (potential) customers.
SALES CHANNEL ROADMAP	To align the relationships in the distribution channel, the business model and sales strategy must associate with the motivations, needs, and goals of the channel partners.
MARKETING ROADMAP	This outlines your marketing strategy and efforts. It is essentially a blueprint that provides an overview of marketing initiatives to stakeholders.
ROADMAP	This parent-concept defines two different types of child-concepts, namely the sales channel roadmap and the marketing roadmap.
HIGH-FIDELITY MVP	This has more features than the MVP and is an upgraded version, although still not being the complete end product.
MVP	This is the minimum viable product which only contains the essential features, the must haves.
CUSTOMER DATA	Is used as input for the high-fidelity MVP, which is gathered during the whole process. Feedback from the customer is essential to create an MVP that solves the customer problem.

TABLE B.8: The concept table of The Startup Owner's Manual, based on the work of Blank and Dorf (2012).

Appendix C

Method Comparison

C.1 Activity comparison table

(Sub-)Activity	**TLS**	**DT**	**TLPP**	**TSOM**
Understanding the problem				
Identify the customers problem	>	=	<	
Uncover problems within identifiable groups	>	>	=	
Identify the user needs		=	><	
Partition market into market segments		>	=	><
Refine and tweak the product	><		=	><
Understand the needs of the target customer		><	=	
Identify specific needs for a market opportunity		><	=	
Test the customer problem	<	<	<	=
Test the customer solution	<	<	<	=
Create a Point of View				
Define stories via storytelling		=		
Cluster the insights		=		
Condense insights into a visual representation		=		
Theorise about user's needs		=	><	
Ideate				
Generate brainstorming question that addresses the user needs		=	>	
Generate ideas for possible solutions to the defined needs			=	
Cluster the ideas				
Vote about which idea to develop further		=		

TABLE C.1: The activity comparison table.

(Sub-)Activity	**TLS**	**DT**	**TLPP**	**TSOM**
Discover Customers				
Document customer-problem-solution (c-p-s) hypotheses	=	><	><	><
Create business assumptions	=			
Create MVP assumptions	=			
Create funnel assumptions	=			
Find prospective customers to talk to	=		=	=
Elicit market insights, customer-problem-solution feedback, and product-market fit insight	=		>	
Create a business plan	>		=	
Engage in interviews to discover core product functionalities (must haves)	=		><	
Create an MVP based on the problem-solution fit (required features)	=		>	
List the unique features	<		=	
Build the required features for the MVP	<		=	
Test the MVP prototype with target customers	<		=	
Learn the value of the solution	=			
Integrate feedback into the product	<		=	
Verify the business model	<		<	=
Validate Customers				
Create a positioning statement				=
Develop customer focused sales	>			=
Develop marketing materials				=
Develop a roadmap for acquiring and converting prospects into customers	=			
Develop a sales channel roadmap				=
Develop a marketing roadmap				=
Develop a product positioning	=			
Convert MVP into high-fidelity MVP	=		><	=
Test customer acquisition and activation plans (get out of the building)	=		>	=
Develop company positioning	=			
Create Customers				
Create a Business Model Canvas (BMC)	=		<	><
Business model planning	<		><	=
Approach customer segment	=			
Build the Company				
Scale organisation	=			
Scale operations	=			

TABLE C.2: The activity comparison table (continued).

C.2 Concept comparison table

Concept	**TLS**	**DT**	**TLPP**	**TSOM**
Understanding the problem				
CUSTOMER'S PROBLEM	=	=	=	
USER NEED		=	=	
MARKET SEGMENT			=	
TWEAKED PRODUCT			=	
UNDERSTANDING	USER NEED	=	=	
MARKET OPPORTUNITY			=	
CUSTOMER SOLUTION				=
COMPETITION			=	
Create a Point of View				
INSIGHT		=		
CLUSTERED INSIGHT		=		
VISUAL FRAMEWORK		=		
POINT OF VIEW		=	TAR. CUST. NEED	
Ideate				
BRAINSTORM QUESTION		=		
IDEA		=		
CLUSTERED IDEA		=		
PURSUED IDEA		=		
Discover Customers				
CUSTOMER-PROBLEM-SOLUTION HYPOTHESIS	=	INSIGHT	TRG CUST NEED	CUST SOL
BUSINESS ASSUMPTION	=			
MVP ASSUMPTION	=			
FUNNEL ASSUMPTION	=			
PROSPECTIVE CUSTOMER	=		=	=
BUSINESS PLAN	BMC		=	BMC
CORE PRODUCT FUNCTIONALITY	=		MVP FEATURE	
MVP	=		=	
UNIQUE FEATURE			=	
MVP FEATURE			=	
MVP TEST	VALUE		=	
VALUE	=			
FEEDBACK	VALUE		=	
BUSINESS MODEL HYPOTHESIS	=			
BUSINESS MODEL CANVAS	=		BUSN PLAN	=
SERVED AVAILABLE MARKET (SAM)				=
BUSINESS MODEL HYPOTHESIS	=			
TARGET MARKET		MKT SEGM		=

TABLE C.3: The concept comparison table.

Concept	**TLS**	**DT**	**TLPP**	**TSOM**
Validate Customers				
POSITIONING STATEMENT				=
CUSTOMER FOCUSED SALE	CUST ACQ RDMP			=
MARKETING MATERIAL				=
CUSTOMER ACQUISITION ROADMAP	=			CUST FOC SALE
SALES CHANNEL ROADMAP				=
MARKETING ROADMAP				=
PRODUCT POSITIONING	=			
HIGH-FIDELITY MVP	=		MVP	=
CUSTOMER TEST	=		MVP TEST	=
COMPANY POSITIONING	=			
Create Customers				
Build the Company				
SCALED ORGANISATION	=			
SCALED OPERATION	=			

TABLE C.4: The concept comparison table (continued).

Appendix D

Aggregated Data (Process-side)

	ID		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	
Understanding the Problem	A1	Uncover Problems within identifiable groups	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	A2	Partition market into market segments with potential customers	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	N	N	O
	A3	Identify the User Needs	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	
	A4	Understand the needs of the target customer	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	A5	Identify specific needs for a market opportunity	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	
	A6	Estimate the size of the market	Y	Y	Y	Y	Y	N	Y	Y	Y	N	N	Y	O
	A7	*If/Else: Target market is feasible*	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
Create a Point Of View	A8	Define stories via storytelling	N	Y	Y	N	Y	Y	Y	N	Y	N	N	Y	O
	A9	Cluster the insights	N	Y	Y	N	Y	Y	Y	N	N	Y	N	Y	O
	A10	Condense insights into a visual representation	N	Y	Y	N	Y	Y	Y	Y	Y	Y	N	N	O
	A11	Theorize about user's needs	N	Y	Y	N	Y	Y	Y	Y	Y	Y	N	N	O
Ideate	A12	Generate brainstorming question that addresses the user needs	N	N	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	O
	A13	Generate ideas for possible solutions to the defined needs	N	N	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	O
	A14	Cluster the ideas	N	N	Y	N	Y	Y	Y	Y	N	Y	Y	Y	O
	A15	Vote about which idea to develop further	N	N	N	Y	N	Y	N	N	N	Y	N	N	E
Discover Customers	A16	Document customer-problem-solution (c-p-s) hypothesis	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	O
	A17	Create business assumptions	Y	Y	Y	N	Y	Y	Y	Y	N	N	Y	N	O
	A18	Create MVP assumptions	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	A19	Create funnel assumptions	N	Y	Y	N	Y	Y	Y	Y	N	Y	Y	Y	O
	A20	Find prospective customers to talk to	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	A21	Elicit market insights, c-p-s feedback, and product-market fit insight	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	A22	*C-p-s assumptions are confirmed*	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	A23	Create a business plan	Y	Y	Y	Y	N	N	Y	Y	Y	Y	N	Y	O
	A24	Engage in interviews to discover core product functionalities (must haves)	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	
	A25	*Move forward when the data starts to look the same*	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	A26	*The data does not indicate a pivot is required*	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	A27	*C-p-s assumptions are still confirmed*	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	A28	Create an MVP based on the problem-solution fit product (required features)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	A29	Learn the value of the solution	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	A30	*Enough value added*	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
Validate Customers	A31	Develop a roadmap for acquiring and converting prospects into customers	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	
	A32	Develop product positioning	N	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	
	A33	*The targeting of the product, reason for buying, and the difference compared to competitors can be explained*	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	A34	Convert MVP into high-fidelity MVP	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	A35	Develop customer-focused sales	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	
	A36	Develop marketing materials	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	
	A37	Develop a sales channel roadmap	Y	Y	Y	Y	N	N	N	Y	Y	Y	Y	Y	O
	A38	Develop a marketing roadmap	Y	Y	Y	Y	Y	Y	N	Y	N	Y	Y	Y	
	A39	Test customer acquisition and activation plans (get out of the building	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	
	A40	*Product positioning is still valid*	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	A41	Integrate feedback into the product	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	A42	*Product-market fit achieved*	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	A43	Develop company positioning	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	
	A44	*A pivot is not required*	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
Create Customers	A45	Create a Business Model Canvas (BMC)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	
	A46	Approach customer segment	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
Build the company	A47	Scale organisation	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	A48	Scale operations	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	