UNIVERSITY OF TURKU
Turku School of Economics

Exploring the antecedents of dynamic capabilities in a software product company

Dynamic capabilities enabling adaptation to change

Master's thesis
In Information Systems Science

Author:
Hannes Herrala

Supervisor:
Ph.D. Jouni Similä

13.06.2022
Turku

Master's thesis

Subject: Information Systems Science (ISS)
Author: Hannes Herrala
Title: Exploring the antecedents of dynamic capabilities in a software product company – Dynamic capabilities enabling adaptation to change
Supervisor(s): Ph.D. Jouni Similä
Number of pages: 74 pages
Date: 13.06.2022

Abstract

Firms operating in the software industry are constantly faced with new opportunities and threats that stem from the volatility of the market. Customer needs change and competing solutions come up in high frequency as software development practices have gotten better during the last decade. Companies need to obtain dynamic capabilities to address rapidly changing markets. This study examines how dynamic capabilities and their underlying microfoundations enable adaption to change in the environment. The study was conducted as a qualitative case study of Company X. Data was gathered using semi-structured interviews with employees in the area of product management. Furthermore, data from interviews were complemented with various documentation to understand the phenomenon better.

The findings of this study show that sensing, seizing, and transforming dynamic capabilities deploy their underpinning microfoundations in ways that enable the company to sense opportunities and threats, address those findings, and transform its (intangible) resource base as required for handling change in the environment. The phenomenon results in three distinct outcomes that are represented as principles of agility: agility foundations, agility fostering culture and learning fast.

This thesis contributes to the literature on dynamic capabilities, product management and agility. The dynamic capabilities literature has previously overlooked the software industry, and this study fills this gap.

Key words: Dynamic capabilities, microfoundations, product management, product development, agility, software.

Pro gradu -tutkielma

Oppiaine: Tietojärjestelmätiede
Tekijä: Hannes Herrala
Otsikko: Exploring the antecedents of dynamic capabilities in a software product company – Dynamic capabilities enabling adaptation to change
Ohjaaja(t): FT Jouni Similä
Sivumäärä: 74 sivua
Päivämäärä: 13.06.2022

Tiivistelmä

Ohjelmistoalalla toimivat yritykset kohtaavat jatkuvasti uusia mahdollisuuksia ja uhkia, jotka syntyvät markkinoiden epävakaudesta. Asiakastarpeet muuttuvat ja kilpailevia ratkaisuja syntyy paljon, sillä ohjelmistokehityskäytännöt ovat parantuneet viime vuosikymmenen ajan. Yrityksillä on oltava dynaamisia kyvykkyyksiä, jotta ne voivat vastata nopeasti muuttuviin markkinoihin. Tässä tutkimuksessa tarkastellaan, miten dynaamiset kyvykkyydet ja niiden taustalla olevat mikroperustat mahdollistavat sopeutumisen ympäristön muutoksiin. Tutkimus toteutettiin laadullisena tapaustutkimuksena (Company X). Tiedot kerättiin puolistrukturoiduilla haastatteluilla, joita tehtiin tuotehallinnan työntekijöiden kanssa. Lisäksi haastatteluista saatuja tietoja täydennettiin erilaisilla asiakirjoilla, jotta tutkittava ilmiö ymmärrettäisiin paremmin.

Tutkimuksen tulokset osoittavat, että kolme dynaamista kyvykkyyttä havaitseminen (sensing), hallitseminen (seizing) ja muuntaminen (transforming), käyttävät perustana olevia mikroperustojaan tavoilla, joiden avulla yritys pystyy havaitsemaan mahdollisuuksia ja uhkia, puuttumaan näihin havaintoihin ja muuntamaan (aineetonta) resurssiperustaansa ympäristön muutosten hallinnan edellyttämällä tavalla. Ilmiö johtaa kolmeen eri lopputulokseen, jotka esitetään ketteryyden periaatteina: ketteryyden perusteet, ketteryyttä edistävä kulttuuri ja nopea oppiminen.

Tämä tutkielma täydentää dynaamisia kyvykkyyksiä, tuotehallintaa ja ketteryyttä käsittelevää kirjallisuutta. Dynaamisten kyvykkyyksien kirjallisuudessa on aiemmin jätetty huomiotta ohjelmistoala, ja tämä tutkimus täyttää tämän aukon.

Avainsanat: Dynaamiset kyvykkyydet, ketteryys, tuotekehitys, tuotehallinta, ohjelmisto.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

1.1 Motivation

This thesis focuses on a software product company's ability to adjust to the evolution of the surrounding business environment. The motivation for this dissertation sparked during the covid-19 pandemic while I was working at a software-as-a-service company. The company has a software product that is aimed to provide value for organizations organizing traditional face-to-face events. However, the coronavirus pandemic affected this event technology industry in three ways. First, market turbulence (Jaworski & Kohli 1993) increased due to changing customer preferences from traditional events to virtual events, which also affected the composition of the customers — only those customers remained whose events could be organized virtually. Second, followed by the market turbulence, the industry was faced with technological turbulence (Jaworski & Kohli 1993), which manifested as the challenge of pivoting from on-site events to virtual events due to social distancing restrictions. Third, the digital transformation of the event industry due to covid-19 resulted in a 400% increase in the number of companies that provide event technology solutions (Neves 2020) and as a result, the environment's competitive intensity (Jaworski & Kohli 1993) was amplified. As a result of the pandemic, the software company in question operates in an even more turbulent environment.

1.2 Introduction

A research stream that has focused extensively on the firm's ability to address rapidly changing markets is the theory of dynamic capabilities. Dynamic capabilities theory stems from the strategic management literature and it was made popular by Teece et al. (1997). The focus of the theory is to explain why some organisations can prosper and survive in turbulent environments while aiming to identify the preceding drivers of long-term firm survival and success (Wilden et al. 2016). Firms need to adapt to and exploit changes in their exogenous business landscape while pursuing possibilities to initiate change through technological, organizational, or strategic innovation (Helfat et al. 2007). What eventually makes firms adapt to turbulent environments in the dynamic capability view is the firm's ability to sense and seize opportunities and reconfigure or transform the existing resource base to maintain the firm's evolutionary fitness (Wilden et al. 2016).

Dynamic capabilities have been increasingly studied across different industries varying from manufacturing (Macher & Mowery 2009) to the food industry (Beske et al. 2014) or healthcare (Pablo et al. 2007) and in various settings, such as in the design process of a product (Cautela et al. 2021) or in the circular economy (Khan et al. 2020; 2021). The dynamic capabilities view has gained substantial recognition in the field of information systems, as it is the most utilized strategic theory in information systems leading journals (Talafidaryani 2020; Hajiheydari et al. 2019).

However, prior dynamics capability literature has overlooked companies operating in the software industry, thus existing literature does not provide any answers on the dynamic capabilities phenomenon in the SaaS industry. The characteristics of the software industry make it an intriguing industry for dynamic capabilities research. Firms operating in the software industry are constantly faced with new opportunities and threats that stem from the volatility of the market. Low barriers to entry in many of the software product categories have fulfilled the market with competing solutions making the SaaS market to be worth 170 billion U.S. dollars in 2022 (Statista 2021). Technological and managerial advancements in software development have decreased the time it takes to launch a new product or feature to the market (Fitzgerald & Stol 2017). Increasing competitive intensity, alternative solutions and changing customer needs make it challenging for a software company to develop the right service for its customers and achieve evolutionary fitness in the market (Münch et al. 2019a).

This study addresses the aforementioned gap in the dynamic capabilities literature by focusing on a software company in the SaaS industry. This study aims to investigate how dynamic capabilities and their underlying processes (microfoundations) enable the software company to adapt to changes in the surrounding business environment. The study seeks to provide insights regarding how microfoundations both deploy and develop the organization’s dynamic capabilities.

The research question in this study is *how dynamic capabilities in a software product organization enable adaptation to change in the business environment?*

1.3 Background

1.3.1 Dynamic capabilities

Dynamic capabilities literature is abundant with various higher-level dynamic capabilities arguably due to a lack of standardization on how capabilities can be classified as dynamic capabilities (Sunder et al. 2019). Scholars Sunder et al. (2019) classified 81 distinct dynamic capabilities from 133 articles published in top-tier management journals between 1990 and 2016. Some examples of dynamic capabilities are new product development (Winter 2003), Re-engineering (Zollo & Winter 2002) and absorptive capacity (Zahra & George 2002). This dissertation focuses on the tripartite model of dynamic capabilities by Teece (2007): sensing, seizing and transforming capabilities. According to Schilke et al. (2018), the tripartite capability model is a highly cited framework in the dynamic capabilities literature — yet not studied in the context of a software product company. Furthermore, sensing, seizing, and transforming are essential in facilitating asset orchestration and redesigning of routines in the organization to sustain dynamic capabilities and competitive capabilities over time (Teece 2012; Linde et al. 2021).

Moreover, to further concentrate the area of interest in this research, this study aims to identify the underlying processes, known as microfoundations, of the three dynamic capabilities and investigate how the microfoundations develop and deploy the three macro-level capabilities (figure 1). Focusing on microfoundations is also beneficial for successfully conducting the research as dynamic capabilities themselves are difficult to observe without looking at how they are implemented in processes (Helfat et al. 2007).

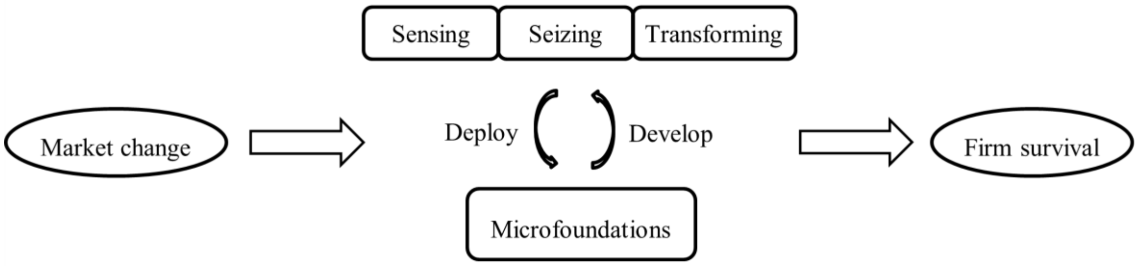


Figure 1 The conceptual framework of this study

The microfoundations concept is known as a movement in strategy and organization theory that helps to break black boxes of macro-level concepts, such as the dynamic capabilities so that their underpinning constructs and the ways that these macro-level constructs emerge, and change can be understood (Felin et al. 2015). Teece (2007, 1319) brought the microfoundations theme into the dynamic capabilities literature, defining them as "the distinct skills, processes, procedures, organizational structures, decision rules and disciplines – which undergird enterprise-level sensing, seizing and reconfiguring capacities". The need for more research related to the microfoundations of dynamic capabilities has been acknowledged, for example, by scholars Schilke et al. (2018) and Wilden et al. (2016).

A common theme in the dynamic capabilities literature is a perspective that insists on dynamic capabilities being explicitly the source of companies' competitive advantage, especially in turbulent environments (Teece et al. 1997; Eisenhardt & Martin 2000; Arend & Bromiley 2009) However, several scholars argue that dynamic capabilities may not always lead to competitive advantage (Schilke 2014; Fainshmidt et al. 2016;) and the role that environmental dynamism has as a moderator for dynamic capabilities and organizations' performance has shown diverse results in the previous literature (Bitencourt et al. 2020). Environmental dynamism's controversial results — being it positive (Menguc & Auh 2006), neutral (Schilke 2014) or negative (Arend 2014) — may be due to several context-dependent factors, such as the competitive intensity of the industry or whether the firm's industry is developed or not (Fainshmidt et al. 2016).

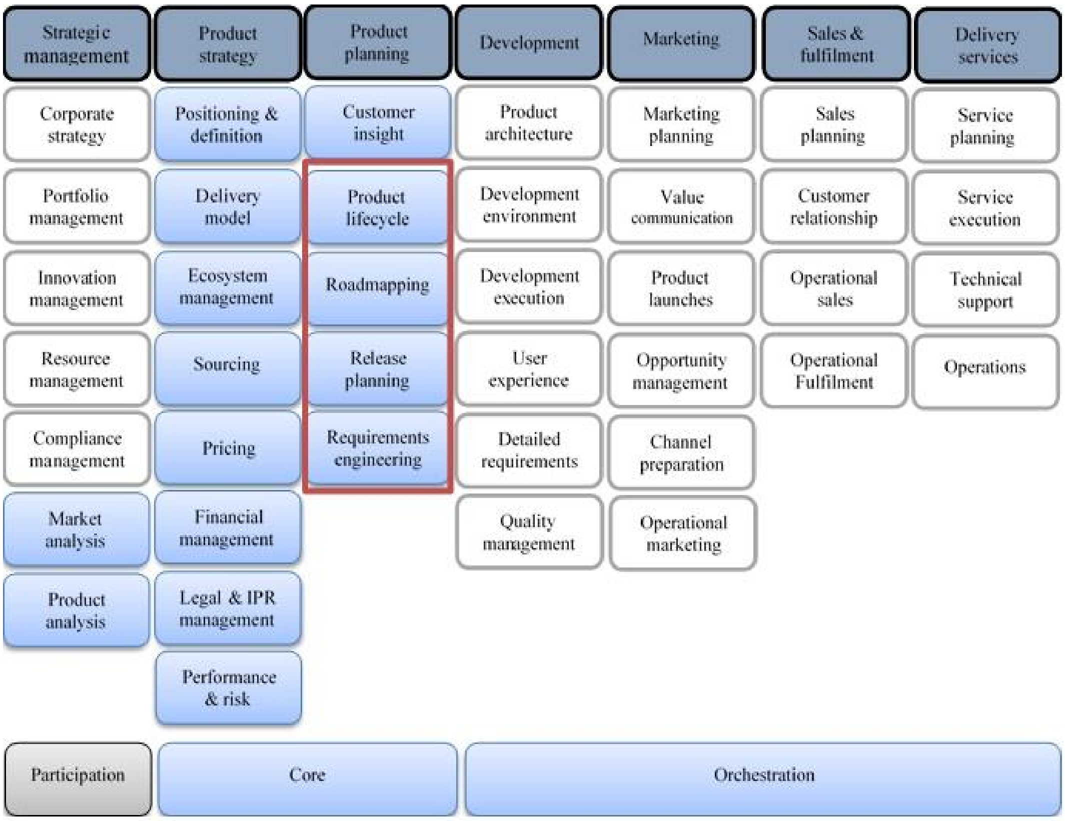
To overcome the conflicting aspects of competitive advantage, firm performance and environmental dynamism in the dynamic capabilities literature, this study will build upon the common understanding of dynamic capabilities' fundamental role in providing changes to the firm's resource base or the external environment (Helfat et al. 2007; Zollo & Winter 2002, Winter 2003; Helfat & Winter 2011) and their ability to "address rapidly changing environments" (Teece et al. 1997, 516) to eventually promote firm survival (Figure 1).

1.3.2 Software product management

To set the context for investigating the phenomenon of dynamic capabilities in a software company, this study will focus on the area of software product management (SPM). The product that a software company provides for the market is arguably the most influential factor for achieving long-term fitness in the surrounding business environment. A set of essential activities take part in the successful development and release of the product to the market. SPM is a socio-technical concept that accounts for all those activities that are related to the product's entire lifecycle from its inception to its evolutionary improvements after it has been launched to the market (Ebert & Brinkkemper 2014; Maglyas et al. 2017).

Software product management in its entirety includes various activities ranging from market analysis to technical support. Figure 2 represents the most recent framework for software product management from the International Software Product Management Association (ISPMA). Software product management as represented in the ISPMA framework would be an excessively large phenomenon to research in its totality, hence this dissertation limits the area of SPM to include core activities that have been investigated in prior literature (see, for example, Weerd et al. 2006; Maglyas et al. 2017):

- Product lifecycle management
- Product roadmapping
- Release planning
- Requirements engineering



Strategic management	Product strategy	Product planning	Development	Marketing	Sales & fulfilment	Delivery services
Corporate strategy	Positioning & definition	Customer insight	Product architecture	Marketing planning	Sales planning	Service planning
Portfolio management	Delivery model	Product lifecycle	Development environment	Value communication	Customer relationship	Service execution
Innovation management	Ecosystem management	Roadmapping	Development execution	Product launches	Operational sales	Technical support
Resource management	Sourcing	Release planning	User experience	Opportunity management	Operational Fulfilment	Operations
Compliance management	Pricing	Requirements engineering	Detailed requirements	Channel preparation		
Market analysis	Financial management		Quality management	Operational marketing		
Product analysis	Legal & IPR management					
	Performance & risk					

Figure 2 Focus area shown in the ISPMA SPM framework (modified from ISPMA 2021).

1.4 Structure of the thesis

Chapter 1 introduced the aim of this study. The theoretical framework of dynamic capabilities was described briefly, and the focus area of software product management was outlined. Chapter 2 considers the literature review of this study. It begins by introducing the theoretical framework of dynamic capabilities and its history. Next, the different definitions of dynamic capabilities are outlined. Then, the three dynamic capabilities that this study focuses on are described. The literature review on dynamic capabilities ends by explaining the notion of microfoundations.

The second part of the literature review consists of software product management. The core product management areas are described in more detail. Next, chapter 3 explains the methodologies used in this study, what are the philosophical foundations of this research, and how the data was collected and analysed. The third chapter ends with an evaluation of the quality of this study. Chapter 4 describes the findings of this study, while chapter 5 provides the discussion and the answer to the research question. Finally, chapter 6 is the conclusion of this thesis.

2 Literature review

2.1 Dynamic capabilities

2.1.1 Antecedents and history

The dynamic capabilities view belongs to the strategic management body of literature. It utilizes many research streams, such as entrepreneurship, the behavioural theory of the firm, behavioural decision theory, organization theory, transaction cost theory, and a bit of evolutionary economics. Strategic management literature focuses on managerial choices regarding the direction of the evolutionary trajectory of the business enterprise. It has a firm grounding in practice and aims to provide frameworks that can aid managers in firms to make better decisions. (Teece 2009). The Five Forces framework by Porter (1979) may be the most known theory in the strategic management field. Porter's framework is insightful for industry selection and building barriers for entry to others coming to the market. However, according to Teece (2009), it has a few limitations; the five forces framework lacks a meaningful conceptualization of the firm, differentiation happens mainly through product choices and little emphasis is given to the firm itself and its capabilities.

The behavioural theory of the firm provided a set of important foundations for the dynamic capabilities theory. Cyert and March (1963) constructed a theory of organizational behaviour, which has four underlying factors: organizational goals, a bounded rationality view of expectations, choice and control. Organizational goals in the behavioural theory of the firm consider the process in which employees take part in a goal-setting activity. The dynamic capabilities view utilizes this by transforming the goal-setting processes into entrepreneurial actions. (Arndt & Pierce 2018.) The bounded rationality concept introduced by March and Simon (1958) suggests that organizational expectations are affected by a lack of complete information available to make decisions. Limited information processing capacity has a central role in the concept of sensing. It affects firms' sensing by making it heterogeneous, thus some companies can sense new opportunities better than others since they have different information (Arndt & Pierce 2018). Arndt & Pierce (2018) argue that the conceptualization of choice in the behavioural theory of the firm may be the most influencing contribution by the theory to the dynamic capabilities view. Choice and control are reflected in the concept of seizing

(Teece 2007), through which standard operating procedures govern how a firm reacts to its environment's signals. Standard operating procedures are different in each company, and they are also a way of exercising control in the company. (Cyert & March 1963; Arndt & Pierce 2018). The aforementioned four elements of the behavioural theory of the firm have influenced how Teece (1997; 2007) and Eisenhardt and Martin (2000) have conceptualized the dynamic capabilities theory in their research (Arndt & Pierce 2018).

Transaction cost theory acted as an antecedent for the dynamic capabilities view by seeking to explain how companies are organized internally (Teece 2009). Transaction cost theory focuses on answering when value creation activities would occur within the market and when in the company. Based on the theory, a firm would internalize its operations when transactions of goods or services in the market would be high. On the other hand, when transaction costs would be low, a firm would prefer to utilize the market for acquiring a good or service. (Williamson 1991.) The transaction cost theory's main element is governance, and while it is of utmost importance in managing a firm and its internal operations, the theory does not give support on how a firm can govern new assets, learn or leverage opportunities. Thus, the transaction cost theory is about value protection, not value creation (Teece 2009).

The evolutionary view of the firm (Nelson & Wilson 1982) was an exemplar theory for Teece's conceptualization of the dynamic capabilities view, as it understood the firm as an entity that is seeking profits, which needs to utilize organizational learning processes to build and exploit key knowledge assets (Teece 2009). Additionally, the evolutionary view of the firm obtains the conceptualization of routines and competencies, although the routines have a static character in the evolutionary view and resemble standard operating procedures as in the behavioural theory of the firm (Teece 2009; Arndt & Pierce 2018).

Dynamic capabilities theory has its most direct origins in the resource-based view of the firm (RBV) which was made popular by Barney (1991) in his article *Firm Resources and Sustained Competitive Advantage* (Wang & Ahmed 2007). RBV theory views resources as a broad category of assets, capabilities, a firm's processes, attributes and intangible assets, such as knowledge (Barney 1991). These resources are the foundation of competitive advantage in the RBV theory. Resources' fundamental role in providing competitive advantage is built upon the idea that firms' resources must be heterogeneous and immobile. This is due to the implication of an argument that firms cannot hold a competitive advantage when resources are evenly distributed across organizations in the market while utilizing resources' high mobility. (Barney 1991). The heterogeneity of firms' resources is persisted over time since resources are immobile, meaning that organizations cannot effortlessly share them (Wang & Ahmed 2007). However, while resources are the essence of competitive advantage in RBV, not all resources play a role in providing competitive advantage. The important distinction between resources that do not relate to the firm's competitive advantage and those that do are the resources' VRIN characteristics: valuable, rare, inimitable and non-substitutable (Barney 1991; Wang & Ahmed 2007).

Valuable resources utilize opportunities or neutralize threats in a business environment. Resources require rarity in a firm's current and future market. A company can achieve a competitive advantage when it can deploy a strategy that is not simultaneously implemented by many of the competing firms. When a large number of firms obtain a valuable resource and successfully utilize it, the competitive advantage derived from the utilization of that particular resource can no longer exist, as many firms achieve a similar outcome from the resource, thus the firms' strategies converge to common strategies. Therefore, an appropriate degree of rarity by a resource is needed for generating a competitive advantage. Valuable and rare resources can only help an organization maintain its competitive advantage if resources are difficult to obtain by other companies, thus resources should be inimitable. Finally, non-substitutability is the last requirement for firms' resources to provide a sustained competitive advantage, which means that no strategically similar valuable and imitable resource can exist. Resources can be strategically similar if different resources produce the same effect after their exploitation. (Barney 1991).

The first conceptualizing building block for the dynamic capabilities theory took place by Teece et al. (1997) in the seminal article *Dynamic capabilities and strategic management*. The authors were the first to define dynamic capabilities as "the firm's ability to integrate, build, and reconfigure internal and external competencies to address rapidly changing environments" (Teece et al 1997, 516). Teece et al. (1997) addressed three economics-oriented theories in the article; the five forces (Porter 1979), the strategic conflict approach that is built upon game theory (Shapiro 1989) and the RBV (Barney 1991). The three economics frameworks addressed many of the important management themes, such as rationality, competition, and pursuing and exiting markets, however, they left a need for a theory that would aim to explain how firms can achieve and sustain competitive advantage in dynamic environments (Kay et al. 2018). Thus, Teece et al. (1997) promoted the dynamic capabilities theory, which focuses on the firm on a deeper level than its preceding strategic management frameworks, while accounting for entrepreneurial acts, dynamics and knowledge in firms' operations to generate economic rents to maintain long-term growth (Kay et al. 2018).

2.1.2 Definitions and characteristics

Several scholars have provided definitions for the concept of dynamic capabilities. There is much convergence between the different definitions (table 1), as many of the definitions consider dynamic capabilities to have an altering effect on the firm's resource base. According to Schilke et al. (2018) most dynamic capabilities literature refers to the definition provided by Teece et al. (1997), while the definition by Eisenhardt and Martin (2000) is the second most preferred definition. Furthermore, a more recent definition by Helfat et al. (2007) is increasing its popularity among scholars researching dynamic capabilities (Schilke et al. 2018).

Table 1 Dynamic capabilities definitions (adopted from Schilke et al. 2018)

Author	Definition
Teece et al. (1997, 516)	“… define dynamic capabilities as the firm’s ability to integrate, build, and reconfigure internal and external competences to address rapidly changing environments. Dynamic capabilities thus reflect an organization’s ability to achieve new and innovative forms of competitive advantage...”
Eisenhardt and Martin (2000, 1107)	“The firm’s processes that use resources— specifically the processes to integrate, reconfigure, gain and release resources— to match and even create market change. Dynamic capabilities thus are the organizational and strategic routines by which firms achieve new resource configurations as markets emerge, collide, split, evolve, and die.”
Helfat et al. (2007, 1)	“A dynamic capability is the capacity of an organization to purposefully create, extend, or modify its resource base.”
Zollo and Winter (2002, 340)	“A dynamic capability is a learned and stable pattern of collective activity through which the organization systematically generates and modifies its operating routines in pursuit of improved effectiveness.”
Winter (2003, 991)	“… one can define dynamic capabilities as those that operate to extend, modify or create ordinary capabilities.”

Author	Definition
Zahra et al. (2006)	"We define as the abilities to reconfigure a firm's resources and routines in the manner envisioned and deemed appropriate by its principal decision-maker(s)."

The term "dynamic" is used to refer to the changing nature of the firm's exogenous environment that requires adaptive behaviour, such as innovations, by the firm. The term "capability" underlines the importance of strategic management in adapting, integrating and transforming organisational skills, resources and functional competencies accordingly to the dynamic environment. (Teece et al. 1997.) According to Teece and Ali-Aal (2012) competencies belong to the category of organizational resources and they are the result of recurring activities, that enable economic tasks to be conducted. Furthermore, competencies may represent a bundle of organizational routines and processes (Teece & Ali-Aal 2012). A firm's capabilities are built through learning, organizational resources and through an organizational trajectory that a company goes through. The capability that a firm obtains is not exactly the same as its producing abilities since capabilities also conceptualize what a firm could accomplish, not just what it produces currently. (Teece 2014.)

A hierarchy of organizational capabilities is often characterized in the dynamic capabilities literature. Firm capabilities can be generally separated into two categories: ordinary capabilities and dynamic capabilities. (Schilke et al. 2018). Ordinary capabilities have been referred to as zero-level by Winter (2003), first-order (Danneels 2002) and substantive by Zahra et al. (2006). According to Winter (2003), ordinary capabilities are those processes and actions that a firm does to produce revenue. Teece (2014) divides them into administration, operations and governance task categories and argues that ordinary capabilities are intertwined within personnel, physical assets, processes and routines and the administrative guidance that is needed to accomplish operative tasks. In this hierarchical dimension of organizational capabilities, dynamic capabilities can be referred to as higher-order or second-order capabilities (Schilke et al. 2018), which can change ordinary capabilities and the way how the firm makes its living (Winter 2003; Zahra 2006). The meaningfulness of dividing the capabilities into two different categories is criticised by Helfat and Winter (2011), who argue that the separation can be insignificant as some capabilities may be utilized for both operational and dynamic purposes. Furthermore, it is difficult to distinguish whether capabilities supporting major changes are dynamic or ordinary, and change is always present to some extent (Helfat & Winter 2011).

There are distinctions among scholars on which levels of the capability hierarchy are understood to embody dynamic capabilities. The dynamic capabilities literature stream acknowledges two contradictory approaches. (Peteraf et al. 2013; Arndt & Pierce 2018). The first is built upon the work of Teece et al. (1997) and Teece (2007), while the second follows the framework of scholars Eisenhardt and Martin (2000). The two approaches have different assumptions regarding what constitutes dynamic capabilities and in which environments they can exist. Eisendhart and Martin (2000) consider ordinary capabilities, the low-level routines, as they say, to be part of dynamic capabilities, which is an opposing distinction from Teece et al. (1997). Eisenhardt and Martin (2000) state that a firm's capabilities can be referred to as best practices that are similar across firms, but Teece (2014) argues that best practices cannot be considered dynamic capabilities since they are easily imitated. According to Peteraf et al. (2013), as best practices are not required to be unique across companies, the degree to which dynamic capabilities contribute to competitive advantage is lower in the approach of Eisenhardt and Martin (2000) than in the approach by Teece et al. (1997). In the approach created by Teece et al. (1997) the presence of a turbulent environment is fundamental for the existence of dynamic capabilities. Eisenhardt and Martin (2000) have a different understanding of the necessity of rapid changes in the business environment, arguing that dynamic capabilities are also employed in moderately dynamic markets, which have less turbulence.

Dynamic capabilities control the speed and depth to which an organization's distinctive resources and competencies can be realigned constantly to address the opportunities and requirements of the surrounding business environment (Teece & Al-Aali 2012). Dynamic capabilities may be existing in certain change routines, such as in product development (Winter 2003), but most commonly they are found in creative managerial and entrepreneurial acts that rely on curiosity and risk-taking (Teece & Al-Aali 2012). An example of such an action may be pursuing new markets for a software product. Basic competencies, the ordinary capabilities of the organization provide efficient performing of operational activities when the competencies are honed properly. However, dynamic capabilities determine if the company is making the right products for a suitable market segment, or if the firm's forthcoming plans are in line with customer needs, technology push and competitive dynamics. (Teece & Al-Aali 2012.) The intrinsic characteristic of dynamic capabilities is that a firm cannot generally buy them, a firm must invest in building the capabilities. On the other hand, ordinary capabilities are commonly acquired outside the company, for example from consultants and are universal in their nature, providing only support for competitive advantage, not necessarily creating it (Teece 2014). Thus, companies that can create dynamic capabilities themselves have the potential for long term profitability. (Teece & Al-Aali 2012).

2.1.3 Sensing, seizing and transforming capabilities

Many scholars have followed the disaggregation of the dynamic capabilities by Teece (2007) into three distinct capabilities: sensing and shaping of opportunities or threats, seizing of opportunities found with sensing activities through business model modification and resource investments and transforming or reconfiguring of the existing pool of the company's assets (both tangible and intangible) to adhere into the competitive landscape. It is through sensing and seizing of opportunities and reconfiguring the firm’s existing resource base that dynamic capabilities equip the firm with various decisions to make, which obtain the potential to enhance firm performance (Eisenhardt & Martin 2000; Teece 2007). Linde et al. (2021) argue that the three capabilities are needed for companies to stay competitive for long periods and to find ways for exploiting the capabilities interdependently.

The large adoption of the Teecean perspective on dynamic capabilities by other researchers is shown in a comprehensive literature review by Schilke et al. (2018), which concluded that the majority of the articles from the dynamic capabilities literature referred to Teece's (2007) typology or the Teece et al. (1997) typology of coordinating activities, learning and reconfiguring the organization. Table 2 shows a variety of studies that have followed Teece's three-part disaggregation of the dynamic capabilities.

Table 2 A set of studies focusing on sensing, seizing and reconfiguring dynamic capabilities (adopted from Majhi et al. 2021)

Study	Outcome of interest	Study	Outcome of interest
Pavlou and El Sawy (2011)	New product development performance	Babelytė-Labanauskė and Nedzinskas (2017)	Research and development performance and innovation performance

Study	Outcome of interest	Study	Outcome of interest
Wilden et al. (2013)	Financial solvency	Jantunen et al. (2018)	Performance in the media industry
Nedzins kas et al. (2013)	Relative non-financial and relative financial performance	Khan et al. (2021)	Circular economy implementation
Naldi et al. (2014)	Innovative performance	Linde et al. (2021)	Ecosystem innovation in smart cities
Wilden and Gudergan (2015)	Firm performance	Venkatesh et al. (2021)	Enterprise risk management as a DC in crises faced by small and medium enterprises (SMEs
Shafia et al. (2016)	Competitiveness of research and technology organizations		

Table 2. Continues

Sensing capability is derived from the need for information acquisition through constant search, scanning and exploring in a dynamic environment. It is of great importance for a firm that finds itself in a fast-paced, competitive and technology-oriented business environment with changing customer needs. (Teece 2007). Sensing allows both startups and incumbents to take on new opportunities, which is made possible by bounded rationality that limits the information that is available to be exploited by companies (Arndt & Pierce 2018).

Sensing capability provides a firm with new opportunities to take on. These opportunities are then addressed by employing the seizing capability. Seizing capability comprises the activities that are focused on implementing the sensed opportunity. (Teece 2007.) Sensing and seizing capabilities provide the organization with the capacity to find new opportunities and act accordingly to eventually earn more profits. However, organizations need to be able to reconfigure their assets, resources and capabilities to maintain their fit with the surrounding market dynamics and market evolution. This reconfiguring capacity is called the transforming capability and its employment protects a company from becoming too dependent on a single path, which may result if a company has a single channel for revenue. (Teece 2007.)

2.1.4 Microfoundations of the tripartite dynamic capabilities

Approximately a decade after the seminal article by Teece et al. (1997) the dynamic capabilities literature has put an increasing amount of focus to understand the underlying processes and disciplines that undergird dynamic capabilities (Helfat et al. 2007; Teece 2007; Wilden et al. 2016). A research stream that is explicitly focused on the underpinning factors of higher-level constructs is the research of microfoundations. Foss & Pederson (2016) state that microfoundations are an expansive grouping of research heuristics emphasizing inter-level mechanisms while setting the locus of the research to the explanatory primacy of the micro-level. Likewise to the theory of dynamic capabilities, microfoundations research has its background rooted in the strategic management literature (Felin et al. 2012). According to Felin et al. (2012) microfoundations research considers collective phenomena which are required to be inspected at a deeper level of abstraction, mainly to understand better how the collective phenomena are created or developed, and reproduced and managed, such in the case of collective constructs like routines or capabilities. The motivation for studying the microfoundations of routines and capabilities is to better understand the heterogeneities between firms – why some firms thrive and prosper while some don't (Felin et al. 2012).

A simplistic theoretical definition for microfoundations is provided by Felin et al. (2012) advocating that they exist at level *N – 1*, in which the *N* represents the higher-level construct. Level *N* is where the dynamic capabilities generally are ought to be located. While the collective phenomena tend to exist at level *N*, microfoundations located in level *N – 1* may still include collective constructs that together affect the development of the level *N* capability or process. To state it in a differently, Felin et al. (2012) claim that microfoundations consist of *main effects* and *interaction effects*. The *main effects* are individuals, processes and structure; and the *interaction effect*s are the *main effects'* relations which lead to the aggregation and emergence of the *N* level constructs.

Scholars in the dynamic capabilities literature have been following the definition by Teece (2007, 1319):

> the distinct skills, processes, procedures, organizational structures, decision rules, and disciplines – which undergird enterprise-level sensing, seizing, and reconfiguring capacities.

A bit similar to the definition provided by Felin et al. (2012), Teece's definition considers processes and structure however, it provides a broader definition of what microfoundations can be. Another definitive attempt by Ridder et al. (2009) state that intertwined processes constitute bundles of processes which lead to the development and deployment of higher-level dynamic capabilities.

Perhaps due to the broad definitions of what microfoundations may be and the popularity that the Teecean view of dynamic capabilities has, the existing literature on the microfoundations of dynamic capabilities is abundant with various underpinning factors for the Teecean tripartite dynamic capabilities (table 3).

Starting with the sensing capability, the sensing of new opportunities is made possible in various ways. Firms should invest in research activity, pay attention to internal and external advancements in technological developments and closely monitor and address customer needs, both explicit and latent. (Teece 2007.) The list of sensing activities can be further extended to consider the identification of new market segments and utilization of innovations by customers, suppliers and complementors (Conboy et al. 2020). By monitoring the competitive landscape and industry evolution, sensing capability also aids in finding threats that may negatively alter the future of the company (Teece 2007.) According to Kump et al. (2019), a company's high sensing capacity considers acquiring reliable strategically relevant information systematically and continuously.

Table 3 Details of sensing, seizing, and transforming dynamic capabilities (adopted from Conboy et al. 2020).

Definition	**Sensing**	**Seizing**	**Transforming**
	The identification and assessment of opportunities	The mobilisation of resources to address opportunities	The continued renewal of the organisation
Underlying processes	Gathering market intelligence	Building competencies	Achieving recombinations
	Spotting opportunities	Choosing decision-making practices	Re-engineering processes
	Identifying target market segments	Selecting partners and distribution channels	Reconfiguring capabilities
	Spotting changing customer needs and customer innovation	Committing to R&D	Managing knowledge
	Interpreting changes and uncertainties	Mobilising resources to address opportunities	Co-specialising assets

	Creating new business models	Forming alliances and joint ventures	Aligning assets dynamically
Value creation	Positioning for first mover advantage	Leveraging complementary assets	Managing threats
	Determining entry timing		Renewing the business model continuously
			Continued renewal

Some exemplar microfoundational activities for the seizing capability have been listed by Conboy et al. (2020): developing competencies, determining decision-making manners, deciding on suitable partners and distribution channels, mobilising resources for opportunities, establishing alliances and committing to the research and development. A firm's seizing ability is on a high level when it can make proper decisions on what kind of information is so promising in its value, that it is worth pursuing to try and transform it into business opportunities. Pursuing an opportunity requires managerial decisions on where to allocate resources. The nature of seizing decisions has similarities to strategic decision-making since the risk of failure is always present. (Kump et al. 2019.)

Lastly, the microfoundations of transforming activities are represented in table 3, of which process re-engineering was given much emphasis by Teece (2007). Managers should revise the organization's routines to maintain their supporting effect operative tasks. It is not uncommon for established firms to develop rigid processes that neglect creativity and innovation. One way to tackle this issue is to decentralize management and give more responsibility to employees working at the tactical level of the company. (Teece 2007.) A company obtains a high transforming ability when it can consistently take on renewal activities that are made possible through decentralized management, allocated resources and skilled employees (Kump et al. 2019).

2.2 Software product management

According to Ebert & Brinkkemper (2014), a product can be a mix of systems, solutions, materials or services, which are constituted as a whole to deliver value and experience to the product's users. Weerd et al. (2006) argue that software products differ from traditional tangible products in the fact that the production and distribution of software products are highly cost-efficient, and software products can be updated effortlessly.

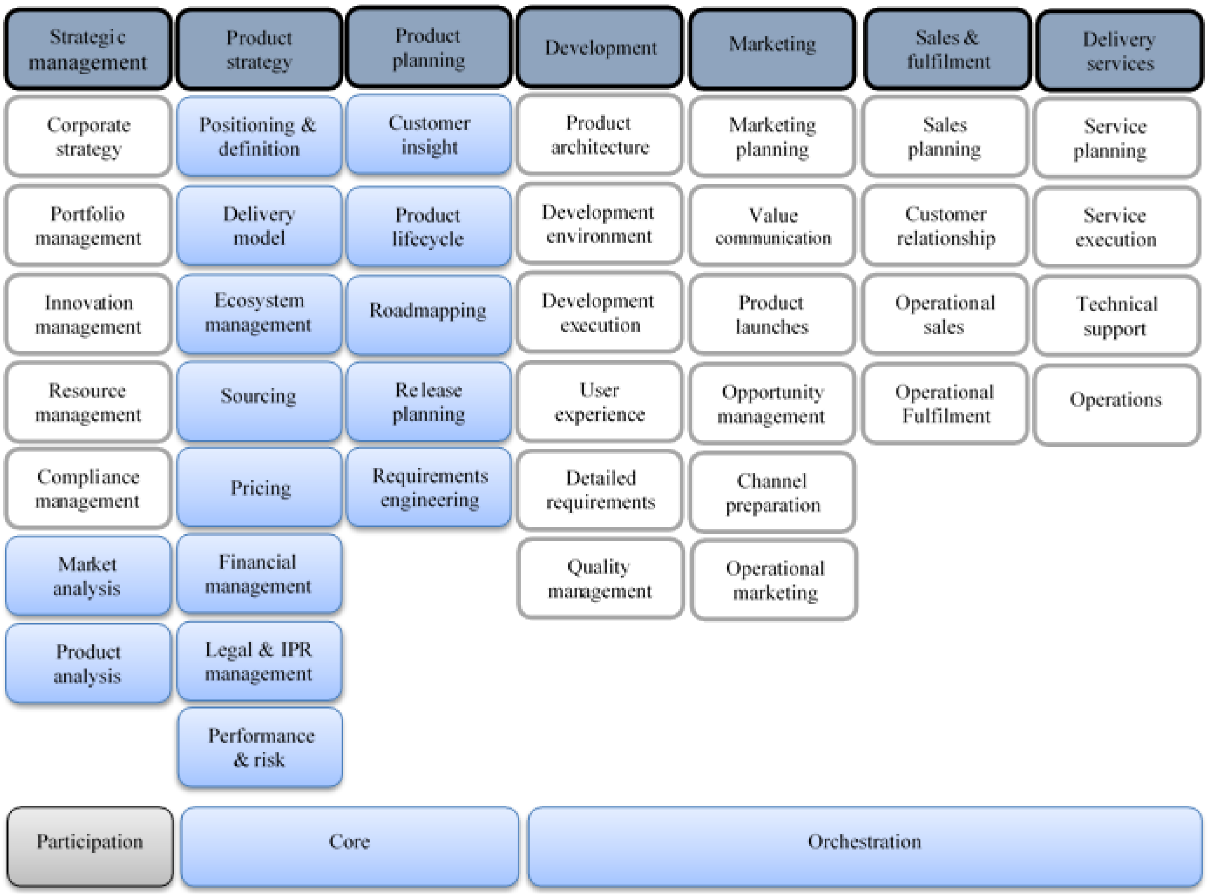
Product management as a concept was first introduced at Procter & Gamble in 1931 when the company assigned product managers to supervise soap products (Kittlaus & Fricker 2017, 1). Later on in the 1960s the marketing mix concept, which is known as 4Ps, was developed by Borden (1964). The 4Ps model considers four areas which are the product (actions related to product development), the price that consumers pay for it, the place where the product is distributed and the promotion that is done to increase awareness of the product (Borden 1964). The marketing mix theory can be considered one of the first theories of product management (Maglyas et al. 2017).

To this date, no generally accepted definition of software product management exists in academia or industry (Maglyas et al. 2017). The lack of general definition stems from the product management's nature – what might be meant with product management is largely dependent on the product involved and on many organizational factors, such as organizational structure, thus the meaning of product management is very company-specific (Kittlaus & Fricker 2017).

Ebert and Brinkkemper (2014) define product management as both a discipline and business process which governs the product's journey from its inception to market delivery to generate as much value as possible for the business. Software product management utilizes technical and business perspectives for managing the software development processes, thus it is a socio-technical phenomenon that requires cross-functional operations (Maglyas et al. 2017). Essentially, the most important aim of software product management is to attain success throughout the life cycle of the software product (Kittlaus & Fricker 2017). A great deal of responsibility in software product management relies on the shoulders of the product manager who is the person that governs and manages the product during its life cycle (Ebert & Brinkkemper 2014).

Software product management in its entirety was represented for the first time by Weerd et al. (2006). In their seminal framework, they argued that software product management consists of four core areas: portfolio management, product roadmapping, release planning and requirements management (Weerd et al. 2006). However, this framework was criticized for not being enough comprehensive (Kittlaus & Fricker 2017). To extend the SPM framework by Weerd et al. (2006), the International Software Product Management

Association (ISPMA) consolidated work from Weerd et al. (2006) and two additional frameworks by Kittlaus and Clough (2009) and Ebert (2007). The ISPMA SPM framework (figure 3) may be considered the state-of-the-art view of software product management, and it is continuously developed by product management experts in ISPMA (Maglyas et al. 2017; Kittlaus & Fricker 2017).



Strategic management	Product strategy	Product planning	Development	Marketing	Sales & fulfilment	Delivery services
Corporate strategy	Positioning & definition	Customer insight	Product architecture	Marketing planning	Sales planning	Service planning
Portfolio management	Delivery model	Product lifecycle	Development environment	Value communication	Customer relationship	Service execution
Innovation management	Ecosystem management	Roadmapping	Development execution	Product launches	Operational sales	Technical support
Resource management	Sourcing	Release planning	User experience	Opportunity management	Operational Fulfilment	Operations
Compliance management	Pricing	Requirements engineering	Detailed requirements	Channel preparation		
Market analysis	Financial management		Quality management	Operational marketing		
Product analysis	Legal & IPR management					
	Performance & risk					

Figure 3 ISPMA software product management framework (modified from ISPMA 2021)

The ISPMA SPM framework represents the holistic view of software product management activities. The activities are shown through functional areas, which are located at the top of each column. The columns obtain a top-down structure, in which strategic activities exist at the top of the column (customer insight) and tactical activities at the bottom (product requirements engineering). Functional areas "product strategy" and "product planning" are marked as "core SPM" to indicate the direct responsibility that a product manager has over these activities. Processes in "Strategic Management" often require participation by software product managers, for example by presenting resource needs. The "Orchestration" area accounts for the activities that must be performed to achieve success with the product, but the activities are outside of product managers' direct responsibility area.

2.2.1 Product lifecycle management

Product management must take care of the software product throughout its lifecycle. The way that product management is conducted is dependent on what kind of lifecycle stage the product is in. First, the product must be created, then published to the market, growth of sales and adoption need to be facilitated, a product may be updated and lastly, the whole product or parts of the product may be withdrawn during the end of its lifecycle. These different phases of the product lifecycle (table 4) require a certain type of focus from the product manager on product planning operations. The first three stages in the product lifecycle require extensive efforts on investing in the product and its growth through development, testing and marketing. The latter three phases aim to keep the product producing a steady cash flow. While the product lifecycle is represented as a linear lifecycle from phase to phase (table 4), in practice the product lifecycle management takes place as iterative work with trial-and-error. (Kittlaus & Fricker 2017.)

Table 4 Product lifecycle (Kittlaus & Fricker 2017).

Phase	Business Aim	Focus Areas
Conception and creation	Investment	Innovate, position product
Market introduction		Launch product, grow market share
Growth		Grow market share, extend functionality
Maturity	Cash Cow	Revitalize product, service product
Decline		Retain customers
Withdrawal		Retain customers, reduce cost

In addition to the product's lifecycle, it is vital for the product management to recognize at which stage of the lifecycle the category of the product is. The product category's lifecycle begins with technology adoption, firstly done by early adopters. At this point, the acquired customers might have different needs than customers coming in during the later phases of the category's lifecycle, a proper product vision and strategy can aid in guiding through the requirements management. The technology adoption stage is followed by the growth phase of the category, in which more customers start to get familiar with the product category in question. After the product's market has achieved extensive growth it starts to mature. When the category is mature, fewer companies are coming to the market, but fierce competition may exist for the remaining addressable market. After the mature phase, the category starts to decline and approach its end of life. During this phase, the product might be found useful in a newly emerged product category with additional development revitalizing of the product. (Kittlaus & Fricker 2017).

2.2.2 Product roadmapping

The product roadmap is a concept that is viewed as a critical element of a company, which lays out the vision and direction of the product offering. The product roadmap represents a journey that the company has to take within the product portfolio to achieve a set of business targets. The product roadmap also defines the required work that must be completed to get to the business objectives. (Münch et al. 2019a). A roadmap is defined by Kerr and Phaal (2021, 8) as a "structured visual chronology of strategic intent". Furthermore, Kerr and Phaal (2021, 13) define the term roadmapping as "the application of a temporal-spatial structured strategic lens". This is derived from the roadmapping processes' role of mapping the problems and opportunities while being a solution that deploys a generic governing framework (Kerr & Phaal 2021). (Kittlaus & Fricker 2017) state that a roadmap in the context of software product management is documentation that shows features or themes of the planned product releases over time. However, the representation of roadmaps described by Kittlaus & Fricker (2017) is criticized for lacking an appropriate blend of "why-what-how-when-who-where" (Kerr & Phaal 2021).

Product roadmapping has its roots in industrial engineering management from the 1960s (Kerr & Phaal 2020). A seminal publication of *Motorola's technology roadmap process* (Willyard & McClees 1987) enormously raised the awareness of roadmapping as a method. Consequently, roadmapping started to spread to a variety of businesses and industries, and it was used for multiple different purposes (Kerr & Phaal 2020. Roadmapping has been utilized in categories such as components, products, systems, supply chains, sectors, regions and nations (Phaal et al. 2010, according to Kerr & Phaal 2021). Roadmapping is also acknowledged as a robust tool for strategic planning and forecasting at the governmental level (Kerr & Phaal 2020). Much of the academic literature on roadmapping obtains the technology and engineering perspective, thus publications are mostly disseminated in three outlets: *Research-Technology Management, Technological Forecasting & Social Change and Portland International*

Conference on Management of Engineering and Technology. Altogether, over 1100 publications are found in the area of roadmapping (Kerr & Phaal 2020). However, despite the numerous articles on the topic of roadmapping, Münch et al. (2019b) argue based on their literature review, that only 23 scientific papers could be recognized as closely linked to the theme of product roadmapping. Additionally, the roadmapping method has yet to be covered extensively in management courses or textbooks on strategy. The industrial roots that roadmapping has, which is greatly reflected in the technology-oriented publications, may lead to the fact that roadmapping is most referred to as *technology roadmapping*. (Kerr & Phaal 2021).

According to (Kittlaus & Fricker 2017) a product roadmap commonly has six basic elements: timescale, releases and versions, release themes and main features, target markets, product dependencies and technology impacts. The roadmap with its constructs is usually visualized for better comprehension, and Phaal et al. (2004) argue for a layered approach for the graphical presentation (figure 4). The layers selected should be based on the product planning needs (Kittlaus & Fricker 2017). Phaal et al. (2004) and Kittlaus and Fricker (2017) suggest that the layers should consist of the market, product and technology layers. The market layer states the objectives that will be pursued in the product's markets and they are essential for the company's financial performance. The milestones share a commercial view, as they may be for example targeted customer segments or increased market share. The product layer comprises the actions taken to develop the product further. The product layer can represent the actions on a higher level by indicating the themes of the software product or by representing planned features on a more detailed level. It is possible to distinguish features further by visualizing essential releases differently than optional ones. Lastly, the technology layer includes crucial enablers that support the development of the software product, such as the technological infrastructure that lies under the product. (Kittlaus & Fricker 2017.)

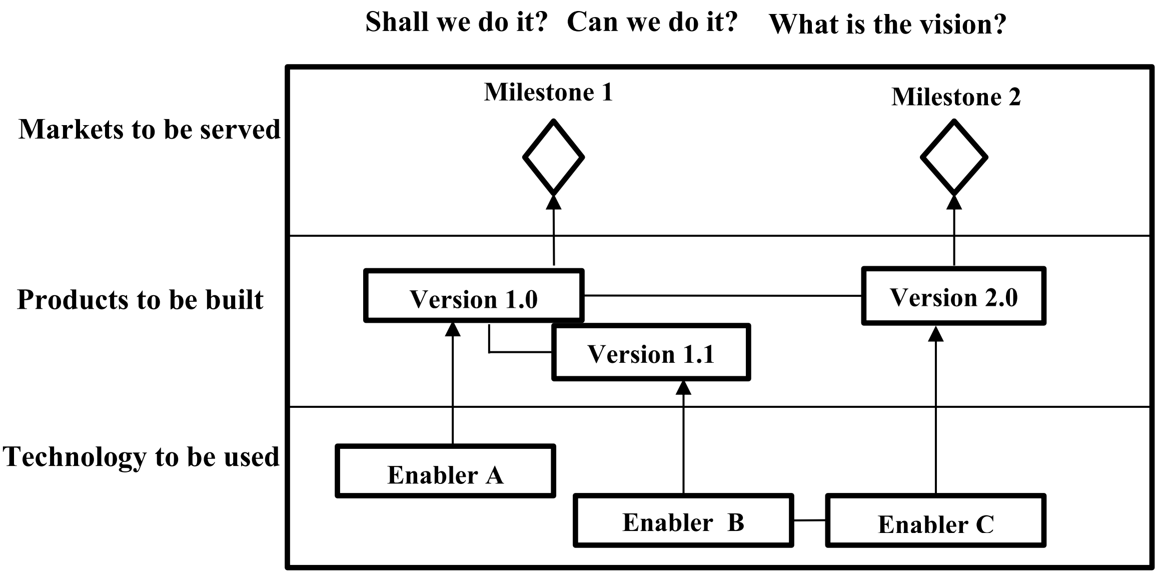


Figure 4 Layered roadmap structure (modified from Phaal et al. 2004; Kittlaus & Fricker 2017)

Product roadmapping is a knowledge-intensive process, that relies on various stakeholders' input. During the process, parties collaborate to achieve a consensus on how to implement the product vision and strategy. Product management and stakeholders also need to agree on the details of the roadmap's implementation. The contributing parties will be selected based on their knowledge and authority (Kittlaus & Fricker 2017). Such contributing stakeholders outside of product management may be marketing, customer representatives and engineering employees. The number of participants in the roadmapping process is highly dependent on the size of the company, meaning that smaller companies might have fewer stakeholders contributing to the roadmapping process. Whilst the number of contributing stakeholders is largely company-specific, the tasks that a roadmapping process comprises are found to be similar across companies. (Suomalainen et al. 2011.)

According to Suomalainen et al. (2011), a usual product roadmapping process consists of capturing features, analysing, and prioritizing features, validating and agreeing with the roadmap and managing changes to the roadmap. Kittlaus and Fricker (2017) advocate for starting with product vision and strategy and then continuing with the capturing of ideas (figure 5). The roadmapping process is continuous and iterative in its manner, meaning that the roadmapping team does not necessarily end the process of capturing features and creating a roadmap as they need to be able to adjust to new opportunities that may arise from market pull or technology push. (Suomalainen et al. 2011; Kittlaus & Fricker 2017.)

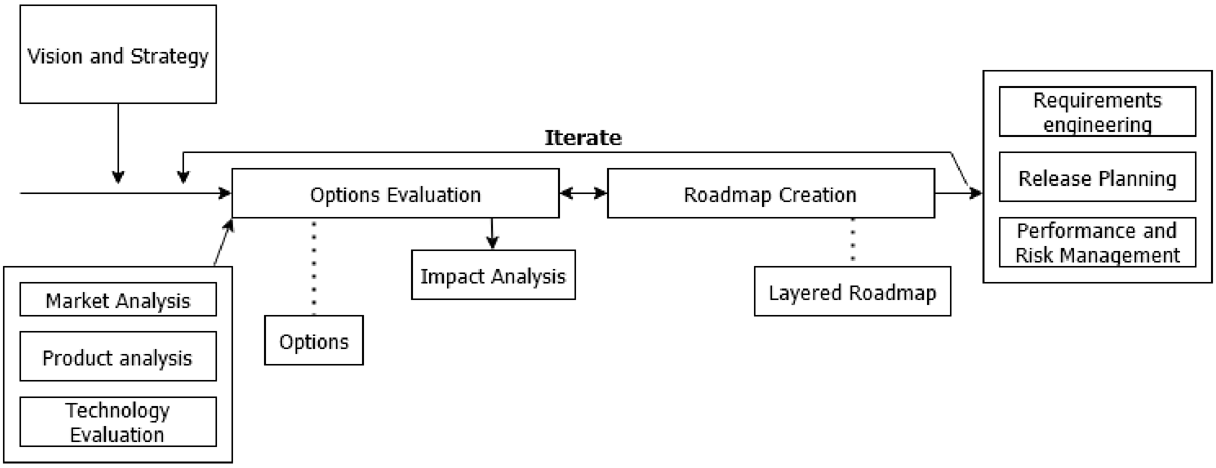


Figure 5 Roadmapping process (Adopted from Kittlaus & Fricker 2017)

2.2.3 Release planning

Release planning is defined by Kittlaus and Fricker (2017, 157) as "the management of the detailed contents and schedule of a forthcoming product release". Ruhe and Saliu (2005) state that release planning accounts for decisions on selecting and assigning features to build a chain of consecutive product releases, which fulfils technical, resource, budget and risk constraints. It is similar to the roadmapping method but distinguishes itself from it by focusing on the short-term releases of new features. Release planning is essential for the operational performance of the product management team since it demands management of the bundles of development resource to guide software development to achieve the product's objectives. (Kittlaus & Fricker 2017). A competent release plan has four characteristics according to Ruhe and Saliu (2005); creating maximum business value by combining the best set of features in an appropriate sequence of releases, fulfilling the needs of the most important stakeholders involved, being worthwhile with available resources, and trace existing dependencies between features. Often companies suffer from a lack of resources in the development team, which leads to challenges for the person responsible for conducting the release project. Often, a release plan is used by the project manager, but in a small company, the product manager may be the primary user of a release plan. (Kittlaus & Fricker 2017.)

The project leader's activity areas can be composed of two phases: strategic planning and operational planning. Strategic planning considers the prioritization and selection of the set of requirements for the upcoming release or for multiple releases in the longer term. Operational planning on the other hand consists of managing the flow of requirements for the development team on a short-term basis. The conceptualization (figure 6) represents the flow from project requirements to a finalized release plan. Requirements engineering is a key concept that precedes release planning, as it produces the input (project requirements), for the release planning's strategic planning and operational planning phases. The outcome of operational planning is the actual release plan, which clearly states who is responsible for what implementation during a certain timeframe. During the release planning process, the project leader must account for multiple aspects, such as social aspects, company policies, client needs, resources, objectives and constraints. (Ameller et al. 2017).

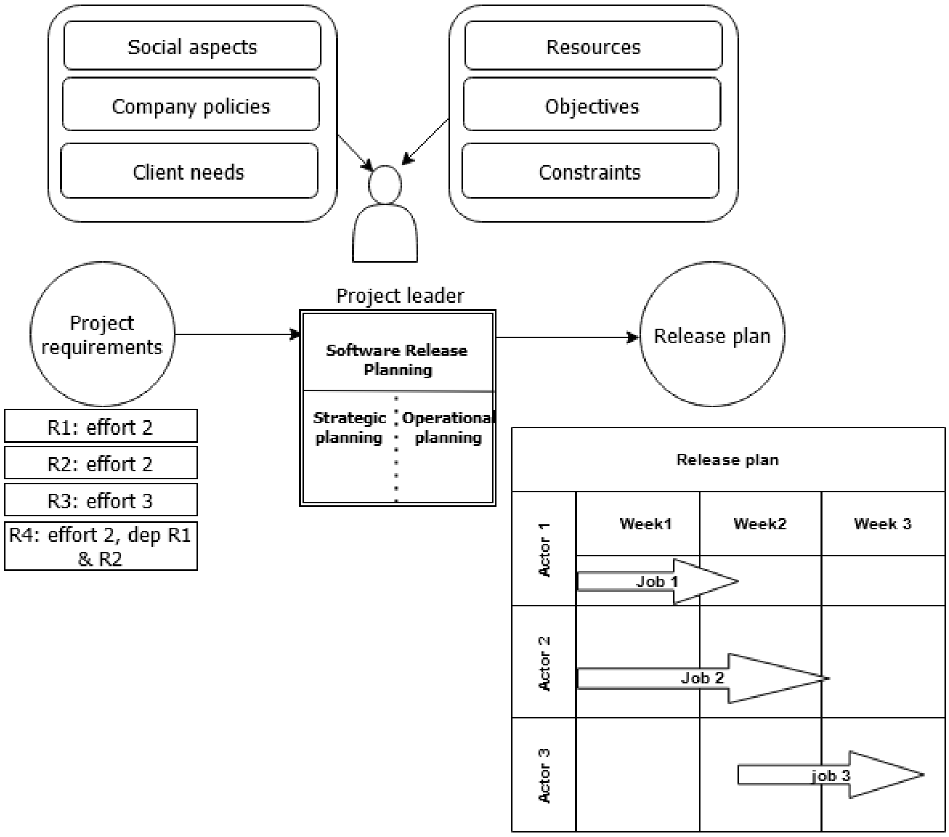


Figure 6 The concept of release planning (adopted from Ameller et al. 2017)

Release types, their definitions and sizes vary to a great degree. Common release types that can be distinguished are pre-release, product release, major release and minor release. A pre-release is an output of early development activity, that is not yet published for the majority of users. A product release is an instance of the existing product that is published for its users and maintained through the product's evolution. Major releases contain an extensive set of new or changed functionality. Minor releases are incremental releases that do not introduce huge changes to the product. The size of the release affects how long it takes to finish the release project. Bigger releases can be done at a lower frequency, but they may be implemented more efficiently than smaller releases. Small releases are developed faster and require less resourcing while promoting greater flexibility to account for changes that may arise from changing market needs. It is a common practice in software-intensive companies to use a combination of frequent small releases and infrequent big releases. (Kittlaus & Fricker 2017.)

2.2.4 Requirements engineering

Requirements engineering comprises actions related to gathering product needs from stakeholders, identifying what has to be done to satisfy needs, and conducting validation on the concepts' appropriateness (Kittlaus & Fricker 2017). A requirement can be a definition of what the user of the product needs or a condition that must be fulfilled to satisfy a need or objective (Suomalainen et al. 2011). In software product management those are characteristics of the product that are required by the stakeholders and agreed with the product management team (Kittlaus & Fricker 2017). In the end, a requirement is a must-have characteristic that a product needs to obtain to provide value for stakeholders.

Every possible feature that the product team could develop is not a requirement. Many ideas are often created in workshops or brainstorming sessions by product management stakeholders. An idea of a new characteristic that the product could instil becomes a requirement when at least one influential stakeholder requires it, or the idea gathers an appropriate degree of stakeholder support. A product manager has to work with great caution during scenarios where the need for an idea cannot be truly confirmed by stakeholders. (Kittlaus & Fricker 2017).

At least three types of requirements can be distinguished: customer requirements, product requirements and project requirements. Customer requirements take place in customer-controlled projects where the software product needs to be tailored for the specific customer, for example by integrating the software product into the customer's other existing information systems. Product requirements are a holistic view of the individual customers' needs. Multiple requirements from different customers are consolidated into a generalizable market view by the product manager. Consolidation of the requirements aims to find similarities in customers' needs that also support the product's strategic objectives. This linking between customer requirements to product requirements is called horizontal traceability. Project requirements comprise a selection of the gathered product requirements for an upcoming product release. Employees responsible for developing the requirements are accountable for translating the product requirements into specifications of a new software release. (Kittlaus & Fricker 2017).

Requirements need to be documented to allow for later use in managing software development, planning the product or revisiting the product roadmap. The way how product requirements are documented is not of importance. No evidence has been found that would support some documentation methods over others for achieving requirements engineering success. Details of the requirements might be saved in documents, spreadsheets, Wikis or requirements database. The documentation method should comply with the needs of the product development team and support the chosen software development cycle by the organization. Agile software development methodology, which started its rise in the early 2000s, instils working in iterations, collaborating with stakeholders during the iterations and having less emphasis on extensive documentation and development planning. In agile development, documentation takes the form of backlogs instead of specification documents. The backlog consists of coarse-grained features that are specified into fine-grained requirements. (Kittlaus & Fricker 2017).

Prioritization of requirements in an agile context occurs by sorting the backlog. The most important requirements are sorted to the top and implemented in the next possible iteration. The entries that are not considered urgent, fall to the bottom of the list and some of those requirements may never be developed. The usage of backlog for requirements gathering does not force any sort of distinction between mandatory and optional requirements. This might induce uncertainty, but the uncertainty of what to develop is mitigated by revisiting and negotiating the backlog at the beginning of a new development iteration. (Kittlaus & Fricker 2017).

An opposing methodology for agile development is a traditional waterfall methodology. The waterfall methodology insists on comprehensive upfront planning and using requirements specification documents that are constructed before the development starts. Requirements documentation has a lot more importance in the context of waterfall development than in agile development, due to the lack of iterations and revisiting of requirements. A successful waterfall implementation requires that correct requirements are defined, thoroughly documented, evaluated and categorized based on their importance. Categorizing may be done by filtering the requirements into optional and mandatory categories. (Kittlaus & Fricker 2017.)

3 Methodology

This chapter aims to explain the methodological foundation of the study. First, the philosophical underpinning of this case study is discussed. Next, the chosen research strategy and its justification are presented. Third, the data collection procedure is reviewed, which is then followed by describing the data analysis process. This chapter ends with evaluating the quality of this research.

The primary goal of this study is to investigate how dynamic capabilities enable adaptation to change in the business environment. Dynamic capabilities phenomenon is a diverse theory with different views that were presented in the literature review, therefore this study focuses, first, on finding the underlying processes (microfoundations) of sensing, seizing, and transforming capabilities in the context of core software product management activities; second, on understanding how the microfoundations are associated with the three dynamic capabilities to promote firm's adaption to change. Reviewed literature on sensing, seizing and transforming capabilities sketched the need for microfoundations research in the software industry, which this study seeks to fill.

3.1 Philosophical underpinnings

All research has some underlying philosophical assumptions that guide the research process and affect which research methodologies are suitable. The philosophical base of the research, known also as epistemology, provides means for the researcher to understand the grounds of his knowledge while providing a reference to the validity of his knowledge. This study falls into the category of qualitative research, therefore making it appropriate to choose from three different epistemological underpinnings: positivism, interpretivism or criticalism. (Myers 2013.)

The positivist approach to research assumes that reality exists objectively, and it can be explained through measurable characteristics, which are not embedded with the research. Positivist research by nature seeks to test theory to improve the predictive understanding of chosen phenomena. On the other hand, interpretivism, often considered as an opposing epistemology to positivism, has a focal assumption that reality can be accessed only through social constructions such as language and collective beliefs. While positivism expects that the data is objective, interpretivism assumes that context determines the correctness of the data. Critical research has similar epistemological assumptions as interpretivism, but the purpose of critical research is to be a social critique. This occurs by challenging the dominant beliefs and assumptions of the research subjects. (Myers 2013.)

This study adopts interpretivism as its philosophical foundation. In interpretivist research, the researcher's focal point tends to be meaning in context. Thus, interpretivist research aims to understand the context of a phenomenon since, in the end, the context provides a medium with certain rules through which the situation will evolve to what it is. (Myers 2013.) In this sense, interpretivism complements the purpose of this study — to gain an understanding of the dynamic capabilities phenomenon in the context of a software product organization. Furthermore, according to Eisenhardt (1989) theory can be exploited at the beginning of the study in guiding the research design and data collection processes. Walsham (1995) continues this idea, describing that in interpretivist case studies, the usage of theory for guiding the latter research processes is done by creating a preliminary theoretical framework that creates a sensible theoretical foundation to which the study's topics and empirical work can be built upon. In a similar sense, Myers (2013) argues that in interpretivist research, the researcher must understand the broader context to correctly inspect a piece of data. In this study, a broad understanding of the context – software product management and dynamic capabilities – has been acquired by the researcher through a comprehensive literature review and complemented with experience from his two-year employment with the case company.

3.2 Research strategy

This study utilizes a qualitative research approach as it provides a profound basis for seeking an understanding of the dynamic capability phenomena, which is embodied in the research question: "how dynamic capabilities in software product organization enable adaptation to change in the business environment?". According to Creswell and Creswell (2018), qualitative research is a suitable approach for researching concepts that are not understood in the existing literature, thus requiring further exploration. Furthermore, it may also be that the topic has not been researched with a certain interest group or context, making it an appropriate case for using a qualitative approach (Creswell & Creswell 2018).

To investigate the nature of the tripartite dynamic capabilities (Teece 2007) and their microfoundations in a software product organization, this study is conducted as a single case study. According to (Yin 2018) case study research methodology explores a current phenomenon in great detail and within its real-world setting, especially in cases where the boundaries between the studied subject and context are not clearly apparent. Furthermore, one of the main benefits of doing a case study is the possibility of using various sources of evidence that contribute to building a holistic understanding of a chosen phenomenon. This case study can be characterized as explanatory since explanatory case studies can investigate a sophistication of activities and events and they seek to explain how some condition – in this case, firm survival – was reached (Yin 2018). Furthermore, this explanatory case study obtains the aforementioned philosophical underpinning of interpretivism, which further promotes the understanding of the studied phenomenon and how it changes over time (Gephart 2018).

Compared to other research methodologies like experimental design, a case-oriented analysis provides a holistic view of the studied subject, rather than picking out variables from the rest of the case. Context-dependency is evident in the case study as a studied variable might lead to a different outcome in a different setting, depending on the causal relations. (Piekkari & Welch 2018.) The importance of context is also accounted for in the dynamic capabilities literature as dynamic capabilities have been acknowledged to be context-dependent concerning the instance that they are developed or employed (Schilke et al. 2018)

According to Piekkari and Welch (2018), the most significant criterion for choosing a case is the provided opportunity for learning and extending the current understanding of a phenomenon. Following that notion, this study focuses on a single software product organization, named Company X, to investigate the dynamic capabilities phenomenon in the context that has been overlooked by previous literature. Company X started its operations in Finland in 2007 but has since expanded its offices to account for Swedish and French markets. Despite having three country offices in Europe, Company X is still serving customers outside its local markets with its software-as-a-service business model. At the time of this study, the company has 70 employees that belong to product, sales, marketing, support or admin functions.

3.3 Data collection

In this study, data is gathered from two main sources: interviews and documentation. Interviews are the primary sources of data while documentation provides secondary data

(Yin 2018). This way multiple sources of evidence are used to support the study's findings, increasing their validity (Yin 2018; Creswell & Creswell 2018). It is also recommended by Walsham (2006) to converge evidence from multiple sources when doing a case study that is built on top of an interpretivist philosophical foundation. Interviews are beneficial for focusing the enquiry directly on study topics with specific people while making it possible for the interviewees to provide deeper explanations or additional insights that could not be planned for in advance. Using documentation allows the researcher to exploit its most significant benefit; to corroborate and enrich evidence from various sources. (Yin 2018.)

This study adopts the semi-structured interviews as its chosen interview type. Semi-structured interviews provide the structural benefit of having some in advance formulated questions, but it does not force the interviewer to obey strict guidelines, thus allowing freedom for emerging questions and conversation during the interview. (Myers 2013.) 7 semi-structured interviews were conducted in total. 6 of the interviewees belong to the company's product team, while one interview was conducted with the CEO of the company. Every interviewed employee had been involved in at least one of the core software product management activities.

Table 5 Interviewee details

Role (abbreviation)	**Time in the company**	**Interview duration**
CEO & founder	15 years	45 minutes
CTO	1 year	50 minutes
Lead developer (LD)	5 years	59 minutes
Product manager 1 (PM1)	1 year	1 hour 7 minutes
Product manager 2 (PM2)	12 years	39 minutes
Product designer (PD)	2 years	40 minutes
Service designer (SD)	1,5 years	32 minutes

Half of the interviews took place on Google Meet. In those cases, the interview was recorded with Meet's recording feature as an mp4-file. Mp4-files were then converted into mp3-files using online converters, and after that, the recordings were uploaded to Microsoft OneDrive service, which allowed for automatic transcription of the Finnish language. The rest of the interviews were recorded with a mobile phone and uploaded to the OneDrive service. All the interviews were conducted in Finnish. The automatic transcription was edited and translated to English. Only those sections of the interviews were transcribed which seemed suitable for microfoundation analysis. The process resulted in 59 pages of transcribed interview data. The interview data was then imported into NVivo 12 software.

Documentation was collected from the company's Google Drive. The collected documentation described the core areas of product management, organization-wide town hall meeting presentations, Company X strategy and vision document, and monthly performance reports. Additionally, product managers' calendars were observed to understand what kind of meetings they conduct.

Table 6 Documentation details

Type	Description	Number
Monthly product information	Monthly product information presentations on the state of product management.	6
Product weekly	Weekly product team information presentations. From the autumn of 2021 and spring 2022.	4
Opportunity portfolio meeting	Meeting presentations regarding opportunities in product management	7
Offering portfolio meeting	Meeting presentations on the current product offerings management	4
Development portfolio meeting	Meeting presentations about the development of upcoming features.	3
Townhall meeting	Company-wide meetings regarding how the company is doing in general.	11
Monthly performance report	Overall performance for the last month. Including financials, product metrics and marketing metrics.	25

3.4 Data analysis

The data analysis was conducted utilizing the Gioia method (Gioia et al. 2013). Gioia method was chosen since it seemed to support the purpose of this research to find the microfoundations of the sensing, seizing and transforming capabilities. Gioia method supports the microfoundations focus of this study, as the analysis process is built on interpretive assumptions (Langley & Abdallah 2011) and the Gioia data structure helps to break the black boxes of dynamic capabilities.

The analysis process with the Gioia method starts with developing 1st-order concepts of the collected data. The process is continued with constructing 2nd-order themes from the groupings of 1st -order concepts. The first-order concepts are put in clusters based on their similarity, and each cluster is linked to one second-order theme. The first iteration of creating the second-order themes resulted in 42 distinct themes. As Gioia et al. (2013) state, it is common to get lost in data in the first part of the analysis. A second iteration was done to cut down the number of first-order concepts and second-order themes to a more manageable amount. The concepts and themes were chosen if they would truly assist in answering the research question. Eventually, 14 second-order themes were constructed from the first-order concepts. The last section of the analysis is to develop aggregated dimensions from the second-order themes, which in this analysis consisted of the tripartite dynamic capabilities (Table 7) (Gioia et al. 2013.)

Table 7 Comprehensive Gioia data structure

Aggregate dimensions (Dynamic capabilities)	2nd order themes (Microfoundations)	1st order concepts
Sensing	**Gathering market intelligence**	• Understanding latent demands by collaborating with customers and different stakeholders • Following industry trends such as customer experience (CX) & agile methods • Understanding competing solutions • Studying innovations outside the main software category
	Strategic foresight	• Weighing on the negative scenarios and benefits of new technologies • Making conscious decisions on when to take technical debt • Planning into the future how different product developments may alter the need for human resources in the support team • Arguing against the board's wishes to widen the product offering amid coronavirus • Understanding where the company should position its offering related to other competing solutions
	Identifying target users and buyers	• Selecting target users & buyer personas • Clarifying the distinction between those two stakeholders together with the marketing function • Considering the different requirements that different regional markets have for the product
	Screening opportunities and threats	• Perception as managerial cognitive capability • Pointing out internal risks • Designers gathering information from product metrics Hotjar and Amplitude • Competitor benchmarking by the design team

		• Multiple employees including the CEO gather customer feedback
Seizing	**Scrum**	• Refinement from sprint to sprint provides a greater degree of agility • Scrum meetings establishing mediums for knowledge sharing • Resources are allocated efficiently as sprints are planned according to the product vision • Iterative development makes it easier to push smaller value-generating functions into development with bigger concepts on an ad-hoc basis
	DevOps	• Pushing developments into production regularly increases the firms' ability to adjust more quickly to uprising needs and trends • Providing means of quality control points which mitigate technical debt in the long term, thus ensuring greater agility in the future • Leveraging infrastructure-as-a-code with automation to construct the base of the product without requiring manual labour for maintaining the technical base
	Outcome-driven roadmap	• Building roadmaps using outcomes that result from creating value for users • Outcome-driven roadmapping with Scrum methodology provides clearer sprint planning • Understanding the context and use case of the requirement • Trying to develop features that create value and not features that no one use
	User-centered design	• Designers working closely with a group of customers • Testing prototypes with customers and collecting feedback • Pushing customer research findings for other stakeholders in product management
	Knowledge integration	• Product requirements are stored in Productboard • Requirements are brought into the scrum backlog from Productboard • Knowledge is created from stored or shared data and information • Knowledge is utilized in decision-making in various sections of product management at various levels • Knowledge flow from different stakeholders • Data and information are triangulated from various sources
	Productization	• Creating a product-led sales in which the users can buy a subscription license within the software platform • Creating multiple ways to productize various features of the software platform through modularity • Motivating users to utilize more functions of the platform without human interaction • Choosing the segments of customers who should receive product marketing messages in the product

	3-portfolio management model	• Holistic perspective for product management through 3 different portfolios • Development resource is allocated in a clear and conscious manner across the offering, development, and opportunity portfolios • Product lifecycle management for the existing product takes place in the offering portfolio
Transforming	**Asset Cospecialization**	• Scrum teams consist of employees with various expertise • Developers, Operations and Designers are working together in each Scrum team • Outsourced experts provide know-how for internal employees • PMs making decisions with the lead developer
	Reconfiguring the product	• Building the product architecture so that it ensures greater modularity enables flexibility in the product offering • Starting the reconfiguration of the technology stack to adjust for the current and future needs of customers • Sunsetting features as the product reaches end of its life
	Agility enabling culture	• Psychological safety facilitates employees' knowledge sharing which is crucial for decision-making by PM & CTO • Top management's support in sharing knowledge • Top management with the board of the company drove innovation amid coronavirus • Organization-wide meetings to share the current state of the product development • Organization-wide free access to Scrum retrospective meetings and product management documentation

3.5 Evaluation of the research

The quality of this research is evaluated by inspecting four aspects that are suitable for qualitative research (Eriksson & Kovalainen 2008):

Table 8 Evaluation of the quality of the research (modified from Eriksson & Kovalainen)

Assessment	**Description**	**Measures taken in this study**
Dependability	The researcher is responsible for giving information to the reader, so that the process has been logical, traceable and documented.	The data collection and data analysis processes are described in detail. Details of the interviewees and documentation was given. Gioia methodology was utilized, and its data structure shown to the reader.
Transferability	The author must show the similarity of the research with prior literature.	Findings were discussed in comparison with prior literature. Findings were inspected through the theoretical lens of dynamic capabilities.

Credibility	The author should be familiar with the topic and the data must be sufficient to back up claims. Other researchers should come relatively close to the same interpretations.	The author has done a comprehensive literature review on dynamic capabilities and software product management, gaining familiarity with the topics.
Conformability	Findings are linked to the data in easily understood manner.	Findings were discussed with quotations. The Gioia data structure aids the reader to understand the link between findings and data.

4 Findings

This section outlines the narrative of the microfoundations of dynamic capabilities in the product management area, that facilitate the case company's ability to adapt to change in the surrounding environment, using Teece's (2007) disaggregation of sensing, seizing and transforming capabilities. The sensing dynamic capability is a means of identification and assessment of opportunities and threats. The seizing dynamic capability accounts for mobilisation of resources to address opportunities. The transforming dynamic capability is embodied in continued renewal of the organisation. Microfoundations, the *N–1* level constructs, were developed as the 2nd-order concepts utilizing the Gioia method. The aggregated dimensions represent the *N* level dimensions.

4.1 Sensing

Four microfoundations were revealed from the data analysis that deploy and develop the sensing dynamic capability in product management in Company X (figure 7). These outlined underpinning phenomena were gathering market intelligence, strategic foresight, selecting target markets and screening opportunities and threats.

Gathering market intelligence took place by various employees in product management. Information and data were collected considering not only the customer's wishes regarding the product of Company X, but also intelligence on possible opportunities or threats that may arise from technological developments, or market and SaaS-industry trends.

> Even though there may be something really cool about some new technology…That there's going to be a lot of those new technologies, like in six years, but how many of them have been buried and perhaps you have realized that this was just garbage, can't do anything with this… We checked that if PHP, for example, has enough kick to withstand like a load. An infographic was found, where it was stated, like: 'we have this tech stack in use and this much load is on it'. – LD
>
> … but then it turned out to be so that there is such a [fragmented] user base in so many softwares that even if we chose just Zoom, then we would exclude quite a lot of other [Company X's] customers from it. – PM2
>
> Most people [in the event management market] understand that the kind of measurement that can be done, like actively implementing and doing it and so that it covers much of the events, must be super simple, that it is like, I think Happyornot is a good example… We have found out that there's no single software that covers everything – CEO

These activities that fall under the gathering market intelligence microfoundation are important for sourcing exogenous information into the firm through deploying the sensing dynamic capability (Teece 2007). The case company also created a structure for competitor benchmarking which also improved further the company's existing sensing capability and made it easier for designers to find competitors for benchmarking.

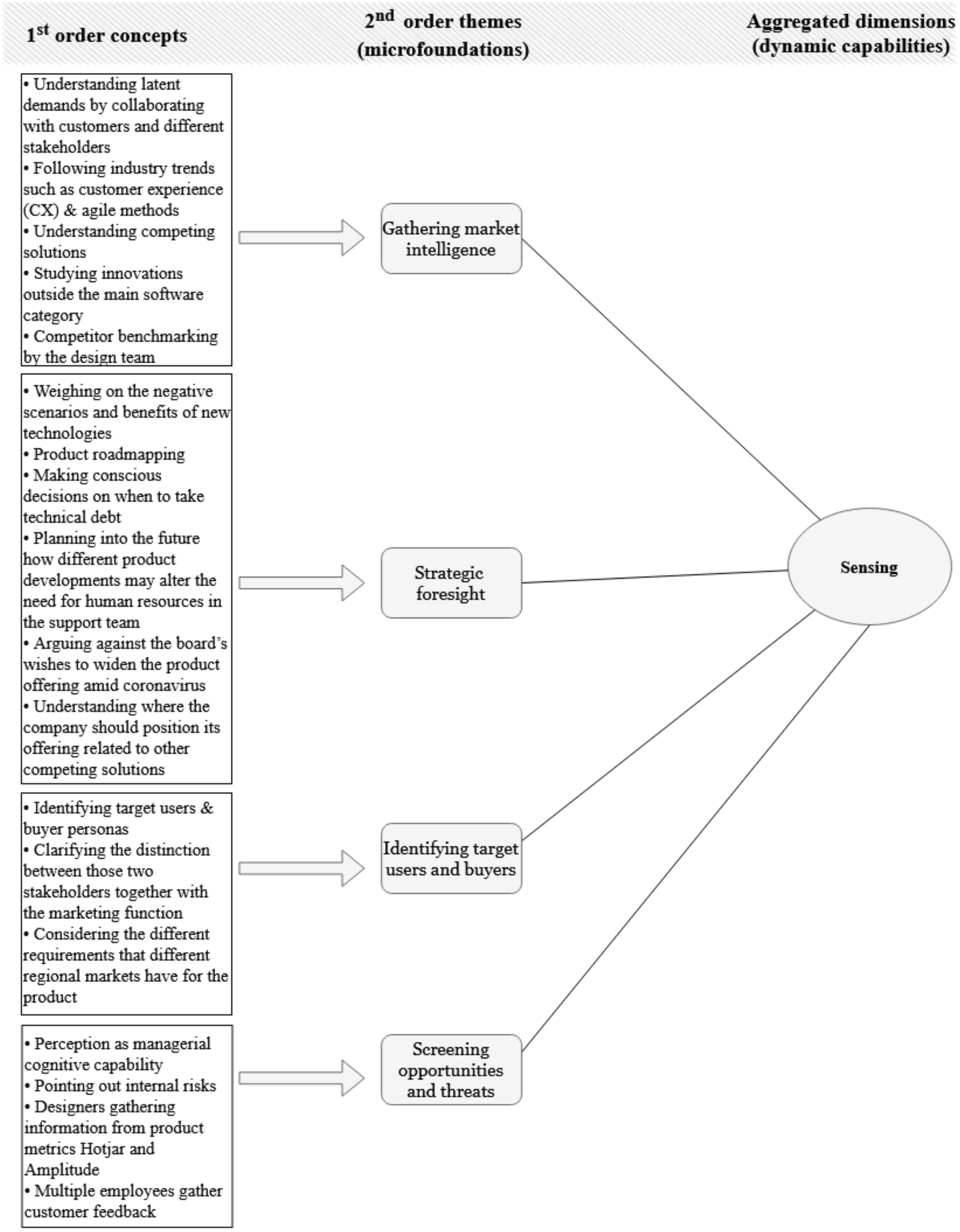

1st order concepts
2nd order themes (microfoundations)
Aggregated dimensions (dynamic capabilities)
• Understanding latent demands by collaborating with customers and different stakeholders
• Following industry trends such as customer experience (CX) & agile methods
• Understanding competing solutions
• Studying innovations outside the main software category
• Competitor benchmarking by the design team
Gathering market intelligence
• Weighing on the negative scenarios and benefits of new technologies
• Product roadmapping
• Making conscious decisions on when to take technical debt
• Planning into the future how different product developments may alter the need for human resources in the support team
• Arguing against the board's wishes to widen the product offering amid coronavirus
• Understanding where the company should position its offering related to other competing solutions
Strategic foresight
• Identifying target users & buyer personas
• Clarifying the distinction between those two stakeholders together with the marketing function
• Considering the different requirements that different regional markets have for the product
Identifying target users and buyers
• Perception as managerial cognitive capability
• Pointing out internal risks
• Designers gathering information from product metrics Hotjar and Amplitude
• Multiple employees gather customer feedback
Screening opportunities and threats
Sensing

Figure 7 Sensing microfoundations

Strategic foresight considers 1st-order concepts such as forward planning, scenario thinking and product roadmapping. It became evident from the interviews with the CTO and the LD that they had put a lot of conscious future-centric thinking on what kind of technologies to utilize in the creation of an improved product platform.

> … it was like there was criteria, like wondering, whether there is know-how [available on the market]. But like the know-how, that you know how it's supposed to work but then it doesn't work like that, and you just need to know [that]. It's like kind of a bug in this language itself. And what the limits of that language are? Where is kind of the borders of the system? It is good that if we have a need beyond its borders, then we know that this is not [suitable]. – LD

> Infra-tech-stack so perhaps the most important thing is that when making technology choices, then we choose technologies with the longest possible life cycle. History is full of stories about technologies that have been hot stuff at a certain moment of time and then buried.. [The choices] are difficult because technology stack choices last from 5 to 10 years from now. The pace of technological development is so amazing that it's really hard to tell what the technologies are valid from now in 5 years… Open source technologies give you a certain kind of security for the choice– CTO

Additionally, the CEO and other top-level managers exhibited strong strategic foresight, as they were arguing against the board's recommendations to widen the product offering to new categories.

> The board had the opinion that it was [invest into new category] just like a must to do, but we were not doing it, there was no business case whatsoever. We have like such a comprehensive cut of accounts that there's no such hidden segment screaming in that market – CEO

Strategic foresight has been stated to significantly influence a firm's decision-making rationality and strategic flexibility (Haarhaus & Liening 2020). Product management in the case company included plenty of investment decisions, both in the short and long term, and the greater the investment decision, the more it seemed to include rationality and forward-thinking. Technology roadmapping has been explicitly described as one of the techniques that embody strategic foresight (Haarhaus & Liening 2020; Marinković et al. 2022), and in Company X, product roadmapping activities deployed the sensing dynamic capability as the employees accounted for the opportunities and threats that may "lie ahead on the road".

Identifying target users and buyers as a microfoundation refers to the actions and processes that resulted in clear plans to where and to whom the product should be sold.

B2B sales has a characteristic of being arguably more complex than B2C market; one reason being the distinction between the user of the product and the buyer of the product in larger enterprises. It was brought up in the interviews and documentation that the value proposition should be different for the user and the buyer.

> We were just talking about it this morning about customer groups and sort of target groups and how the sales were thinking about and that they kind of define those "needs-groups" from there. There have been identified 2 buyer personas, event planner and marketing manager and now there have been thinking whether we should add event success manager as third buyer persona. Then we have user personas which are a bit different. We think about it so that there are the actual users and therefore also think about them [other] stakeholders, for example, like the marketing manager. The set [we offer] is a completely different version based on the user and then in a way the [buying] person who makes the decision who needs the data, she's not interested in the user interface, she's not so much interested in the features, she's interested in whatever the data she gets, how does it benefit her – PM1
>
> Currently we are developing the customer segmentation. – SD

Teece (2007) outlines that when the microfoundation is about identifying, rather than selecting, then it underpins the sensing dynamic capability, and similarly Conboy et al. (2020) argue that identifying target market segments undergirds the sensing capability. In this study, *identifying target users and buyers* manifests the sensing capability, since it is not about mobilizing resources to address opportunities, instead the microfoundation was about acquiring information and making a committing preliminary decision about future. It was evident in product roadmapping processes and also in more short-term work in sprint planning as it must be clear to whom the feature is going to be built.

Lastly*, screening opportunities and threats* is at the core of the sensing dynamic capability, and it monitors both opportunities and threats within and outside the company's borders. In Company X, pointing out threats, especially internal ones, was more present by employees in positions of management or leadership, perhaps due to greater amount of human capital that is often required in such positions.

> Most often, technical debt is taken unknowingly, and it is taken a bit by bit. Which, is because those things that accumulate that technical debt, is not visible… As architecturally, there had been perhaps some such things left behind in today's [Company X's] product. I don't know if you can say them about casting faults, but in a way, solutions that have been right in that moment of time. But as business has moved on, the needs have grown. There have been different kinds of customers, then it has become, in a way, like taking on the technical debt… In the database, there are such structures that

> are very inefficient and poorly scalable... As it is code-wise, there's been kind of inefficiency. So, we must do the same thing, like at worst three times [due to technical debt]. These are like the results of long-term product development, in which that debt has been taken out either consciously or unconsciously. – CTO

> It's definitely a handicap and in a way if you think about the technical debt, in my opinion it has been one of the biggest drivers for the next gen project… We have made there manual processes because we have not had the opportunity to use developers' time for certain things, which then makes it so that we have a lot of time-consuming manual processes, which again slows down our overall growth – PM1

> We do it once a year. An outsider does penetration testing, that is, let's put up the same environment. Then we say hack it. and tell us what you found. Some test it with different tools. Then they give us a list of fixes that should be made, kind of like car inspection.- LD

Based on the interviews, most of the time, internal threats are not collected systematically over a structured process. Instead, recognizing threats was an individual-level phenomenon in which the employees' cognitive abilities, such as perception and sense-making, were essential for acknowledging threats. Perception is a managerial cognitive capability which embodies dynamic managerial-level capabilities, and prior experience accumulates greater perception (Helfat & Peteraf 2015). This was seen especially in the interviews with the newly hired CTO and PM1, as they came to Company X with several years of experience in the industry. They sensed complex problems that resulted from teams working in silos and utilizing a waterfall-based project development model.

On tactical level, designers perceived both threats and opportunities from information systems that provide product metrics (Hotjar & Amplitude), which digitalize the users journey within the software product. Such information provides more background releases or making decisions regarding product lifecycle management. Product metric data helps identifying areas that can be *sunsetted* (feature reaches the end of its lifecycle and is removed).

4.2 Seizing

In total 7 microfoundations of the seizing dynamic capability were found from the data analysis (figures 8 & 9). Four of the microfoundations can be viewed as a cluster of microfoundations that largely facilitate the agile software development which took place in Company X (figure 8).

Scrum consist of 1st -order concepts that stem from the Company X’s development model, Scrum methodology. Scrum as a microfoundation deploys the seizing dynamic capability through its sprint planning, refinement of requirements, iterative development, and resource allocation.

> Scrum helps you like to do things more agile. We can get the value-generating thing out, like quickly… It will make it possible for us to be much more committed to the task that is going on. We will have fewer side balls, which will then cause the promised or planned workers' tasks not to be true or to be completed as those would in roughly in the time frame they should be – PM2

> Good agile software development involves clearly more planning than such a traditional waterfall or some kind of project style where most of the planning work always takes place at that beginning and then there is a long phase when it’s implemented. In the agile world, the design and implementation phases alternate smoothly all the time and then, like from the perspective of a roadmap, its effect is that its [roadmap’s] steering capability is much better. – CTO

Scrum as an agile development framework creates a strong organising structure for short-term release planning and development work in the case company. Activities that occur during Scrum sprints and meetings are ultimately those that put most of the planned investments and resource allocations into action with DevOps, to address opportunities.

DevOps as a microfoundation both deploys and develops Company X’s capability to seize opportunities. Concepts that belong in the DevOps microfoundation create a metaphorical automated factory line, which allows product management to push developments into life, leveraging continuous integration and continuous development (CI/CD) (Fitzgerald & Stol 2017). This happens by bringing the two teams, Developers and Operations, together, as the term DevOps hints. DevOps is a software process that enlarges the agile practices, by focusing on the software development and delivery (Mishra & Otaiwi 2020). The CTO explained why Company X saw investing in adopting DevOps as necessary:

> Well, it is not the goal of Scrum to put things into production. It is rather to have a coherent concept after a sprint that is ready for production. It is a different process to put it into production [through DevOps]. There is this CI/CD pipeline. Continuous integration/continuous deployment which means that we have a kind of software in development…there is such like a production line in a factory where things go into the production line in a controlled manner, moving forward. Second key thing in DevOps world is the infrastructure of how to run these types of systems… Instead of this infrastructure being like this, like a hand-built and documented one, it's written as code. Then you can put version control in there, automated testing there and code review checks. On the infrastructure side, these are like quality

> control points… Manually doing it and in a way, like maintaining it, it's really hard, laborious and slow. When it’s written as a code, it’s suddenly a lot easier to manage and reproduce. – CTO

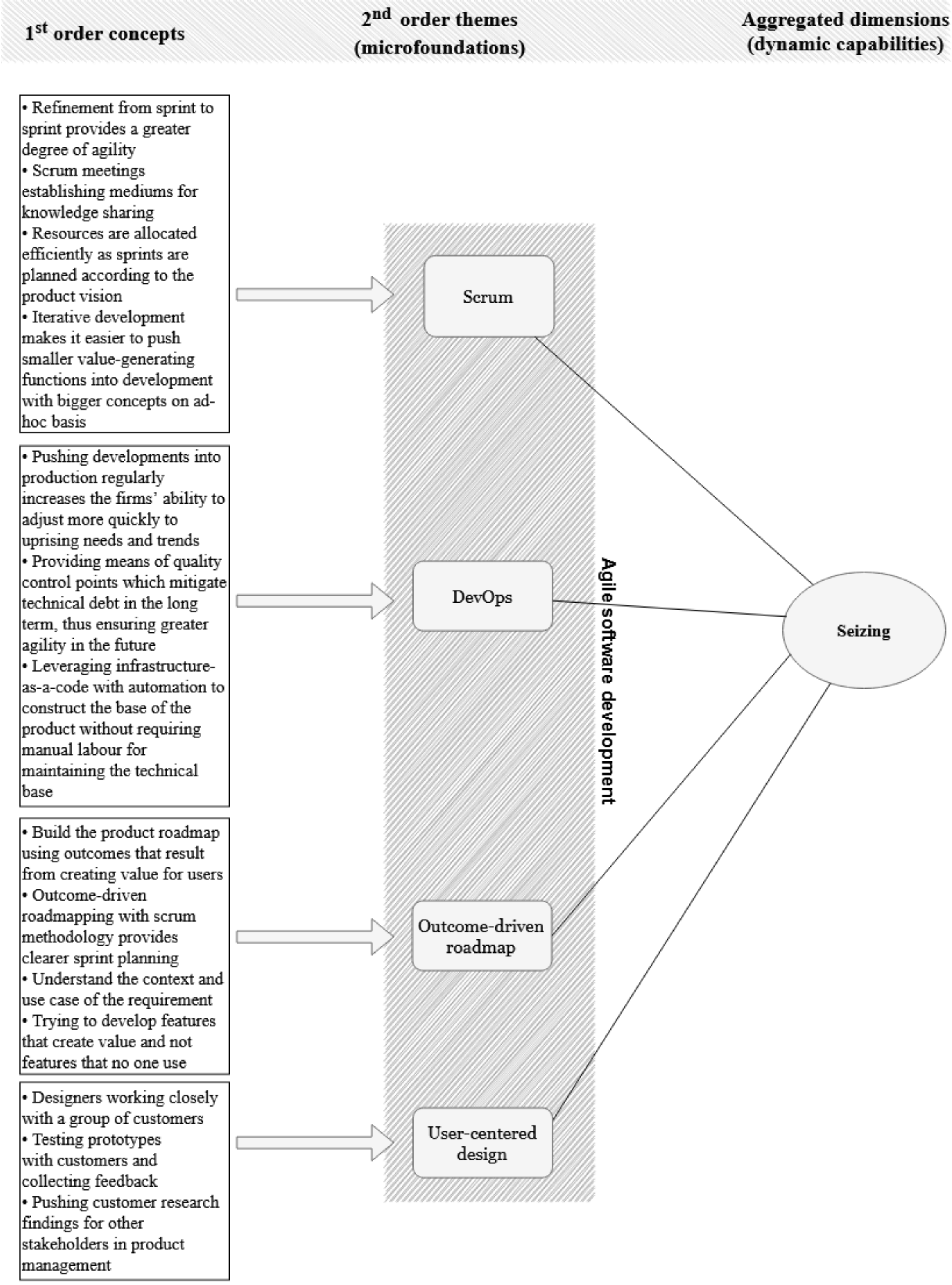

1st order concepts
2nd order themes (microfoundations)
Aggregated dimensions (dynamic capabilities)
• Refinement from sprint to sprint provides a greater degree of agility
• Scrum meetings establishing mediums for knowledge sharing
• Resources are allocated efficiently as sprints are planned according to the product vision
• Iterative development makes it easier to push smaller value-generating functions into development with bigger concepts on ad-hoc basis
• Pushing developments into production regularly increases the firms’ ability to adjust more quickly to uprising needs and trends
• Providing means of quality control points which mitigate technical debt in the long term, thus ensuring greater agility in the future
• Leveraging infrastructure-as-a-code with automation to construct the base of the product without requiring manual labour for maintaining the technical base
• Build the product roadmap using outcomes that result from creating value for users
• Outcome-driven roadmapping with scrum methodology provides clearer sprint planning
• Understand the context and use case of the requirement
• Trying to develop features that create value and not features that no one use
• Designers working closely with a group of customers
• Testing prototypes with customers and collecting feedback
• Pushing customer research findings for other stakeholders in product management
Scrum
DevOps
Outcome-driven roadmap
User-centered design
Agile software development
Seizing

Figure 8 Seizing microfoundations (1/2)

Adopting DevOps practices enhances Company X's seizing capability by providing faster feature deployment cycles, which eventually result in addressing opportunities sooner. In terms of product management, it is seen as improved time-to-market when launching new features or a bigger software release. Another thing that DevOps brings to Company X is that its quality control routines help to keep the iterative development's wheels spinning, by mitigating technical debt. Rubert and Farias (2022) found that after the implementation of DevOps's continuous delivery practice, customers reported fewer defects and bugs in the product, indicating that the implemented practices can improve the product quality. Releasing good quality products is essential for creating customer value and satisfaction, and arguably in the SaaS context, a bad quality product may lead to customer churn. As Mishra & Otaiwi (2020) outline, the foundations for success in the software industry are speed for developing a product and software quality, and DevOps may help to build those foundations by increasing deployment frequency as much as 40 times higher compared to non-DevOps practices.

Outcome-based roadmap undergirds the seizing capability by insisting that the development goals are closely related to fixing customer's problems and creating value. It brings clarification and clearer path for addressing sensed opportunities, by focusing on what matters – customer value:

> Then, in a way, if customers have even more reasoned why they want something, then that is extremely important information, that we understand that background wish, because it is they who may be hoping for a button for something, but still that wish may be someone else. – PM2

> With the next gen, we have now like a whole new approach to our work, that the product vision was, for example that [chief people office] can do a town hall event for employees. A good and such a clear goal… Before it was like a variety of roadmap milestones and this was like, a concrete goal with all the features and functions for it… What is the value that the customer gets form this job and then we always have in sprints like it is such an outcomes-based… But the product vision helped a lot of everyone get together and map out what we need to do. Nothing else was in it, it made it easier to cut things off. – PD

> There has been quite justified feedback about the product vision, what is it? It is a bit unclear what the use case is… Now it's more like we want quite strongly that the things bring value to the customer. – CEO

Outcome-driven roadmap may be viewed as a discipline to do roadmapping practice using outcomes as milestones on the roadmap, instead of doing the more common feature list roadmaps, which have a bad reputation for guiding development when times get turbulent (Münch & Trieflinger 2019b). Outcomes are closely tied to the value that the development should create for the product's user. Together with *Scrum* and *user-centered design* microfoundations, they create value-centric thinking for creating the roadmaps and planning the shorter-term releases, which aid Company X in addressing the opportunities.

User-centered design is the last microfoundation that belongs to the agile software development cluster (figure 8). It represents the 1st-order concepts that consist of the actual customer-centric development actions that took place in Company X, such as customer prototyping and research. Compared to other microfoundations, this underpinning phenomenon is largely executed by the designers in the product team without the immediate presence of a manager. These actions try to make sense of whether the visioned outcomes are correct – did the user of the product really benefit from this development?

> We first had a closed beta that we took in certain customers. Then there's prototypes, and like, such as solutions, solution ideas. How to solve those customers' problems and then we test them. 'Would this be a good solution?' and then iterate them that it's a cycle like that. And on the basis of them, we make such solution proposals and test them so that we are now trying as much to test with the real customers. To get a realistic picture of those people's thoughts. All the different things and concepts are tested, but that you have to have a precise understanding of what we want to test… In Scrum it is quite important that all tickets are done from the customer's point of view – PD

> We have been taking customers to develop with us. We involve them in the alpha to some extent, the alpha testing and beta testing phases and all the time because when you get involved you get to influence, and you know what's coming out of it, so you think I want to use it. – PM1

In the case company, *user-centered design* and *outcome-based roadmap* microfoundations drive the seizing dynamic capability's "designing mechanisms to capture value" (Teece 2007, 1334). *Outcome-based roadmap* accounts for higher-level planning and identification of the value capture, while *user-centered design* governs on the tactical level that the product development is steered toward the right direction. If the product designers identify misalignment between the planned feature and users' perceived value during *user-centered design* practices, it acts as a catalyst for microfoundations such as *Knowledge integration* and *Scrum*.

Knowledge integration represents those knowledge management activities that take place in product management: documenting information, sharing it with others and utilizing that gained knowledge in decision-making. Product management is arguably a very knowledge-intensive discipline; the product manager responsible for giving the orders of what developments to pursue needs to have various information from diverse sources. Based on the interviews and product managers' calendars, the product managers in the case company organize meetings with different internal stakeholders on a monthly basis.

> We have then again meetings in the whole. If you think about, that our product is guided by them. And we have meetings with sales, with support. We have with marketing and in a way, the fact that we get feedback also from different countries constantly, so all the data that we collect guides product management… Then at the end of the day, it's just that you listen to the stuff and then you make decisions as you understand that 'let's actually forget this. or let's add these in here' [upcoming sprint]… They bring in the know-how and the decisions. So, more sensible decisions, because they were the ones that I would decide basically what the professionals thought, rather than me doing the spec on my own and finding that this is what I wanted. – PM1

> In a way we're nothing like the kind of rulers of the only real information as product managers, that it's like the important thing about the whole expert organization that's around it – PM2

Knowledge integration microfoundation is essential for addressing opportunities in a successful manner, since both Company X's product managers do not have technical background per se, given that they come from the business side of operations. They complement their knowledge with all the professionals expertise that is available in the product team. Additionally, the lead developer works closely with the two product managers as the authority to make the product management decisions. Both product managers and product designers seem to triangulate the data from different sources. In the interviews with the product designer, it was mentioned that they try to seek for data from information systems and ask opinions from couple of different stakeholders, such as the customer support team or customers of the company.

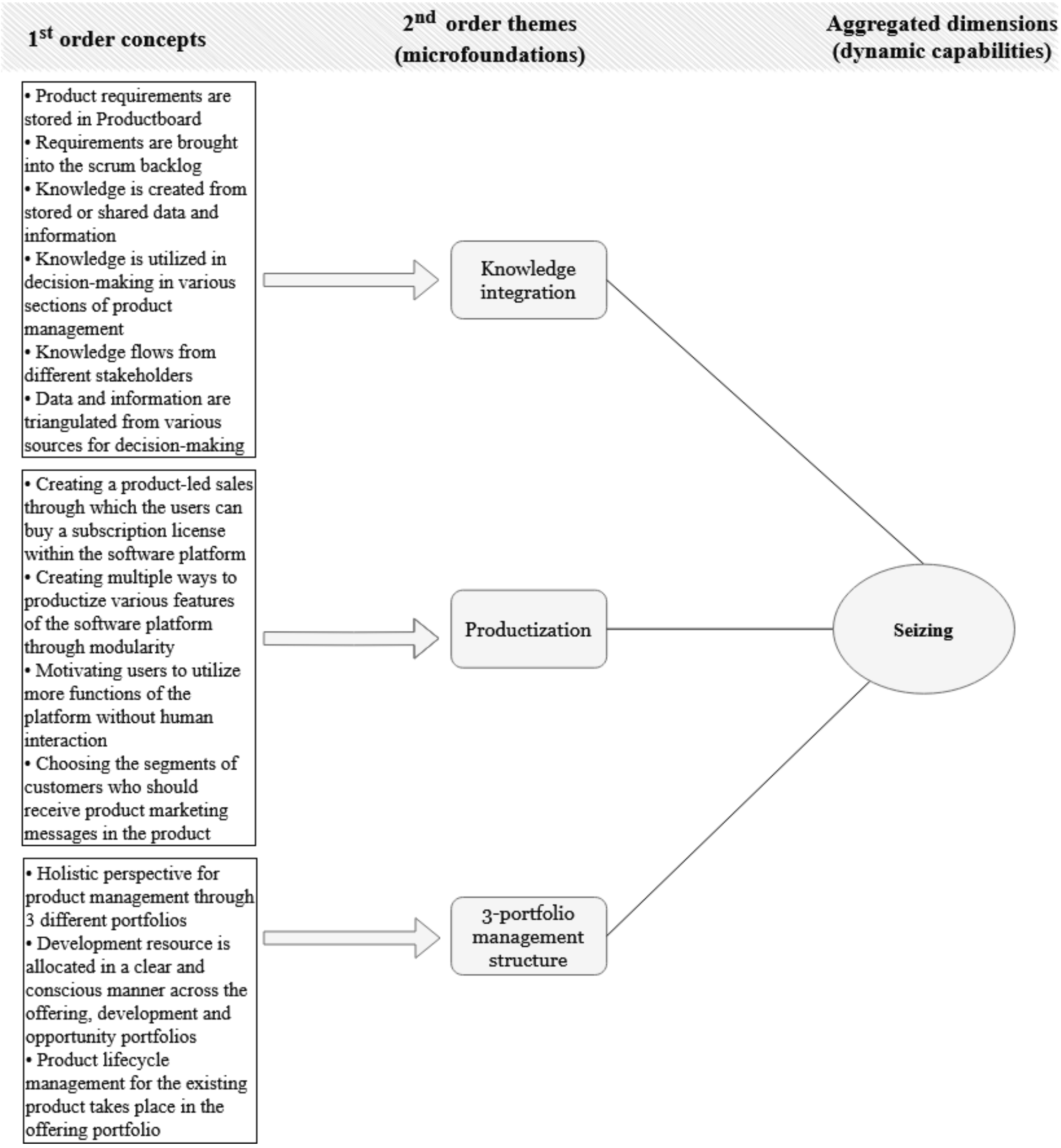

1st order concepts
2nd order themes (microfoundations)
Aggregated dimensions (dynamic capabilities)
• Product requirements are stored in Productboard
• Requirements are brought into the scrum backlog
• Knowledge is created from stored or shared data and information
• Knowledge is utilized in decision-making in various sections of product management
• Knowledge flows from different stakeholders
• Data and information are triangulated from various sources for decision-making
Knowledge integration
• Creating a product-led sales through which the users can buy a subscription license within the software platform
• Creating multiple ways to productize various features of the software platform through modularity
• Motivating users to utilize more functions of the platform without human interaction
• Choosing the segments of customers who should receive product marketing messages in the product
Productization
• Holistic perspective for product management through 3 different portfolios
• Development resource is allocated in a clear and conscious manner across the offering, development and opportunity portfolios
• Product lifecycle management for the existing product takes place in the offering portfolio
3-portfolio management structure
Seizing

Figure 9 Seizing microfoundations (2/2)

Productization microfoundation considers how Company X executes the process of translating, assembling and constructing a proper mix of elements into a product (Harkonen et al. 2015). It is the act of putting the product's different features into a coherent construct that is the most suitable for generating sales with a certain prospect type, such as a small business or large enterprise. This way, customers can choose the feature offering that they prefer and not buy the whole software product. A recent trend in the SaaS industry is known as product-led growth, meaning that the sales and growth happens through the product, and not through outbound sales. Company X is also planning to implement product-led sales to its platform:

> We are going to make sure that the product leads the user to the fact that there is an option of this kind [of feature], ‘but your license package itself does not have this’, so at some point user wants to pay a certain amount to take a certain add-on in, and we go through it that way… I have to say that if you think about that next gen, then we can't do it on any other basis [than product-led sales]. We have to get “something under the line" and that means that we have to go in the direction that we are thinking about the product-led [sales] and thinking about add-ons. – PM1

> If you think that now the next gen is starting to build module, that customers are not going to just take a few modules, but like have the courage to utilize our platform in diverse manner. they can choose something from the modules and whether we can identify the kind of customers that we can most effectively sell at a specific point – SD

Productization and its 1st -order concept product-led sales deploy the seizing dynamic capability’s characteristic of creating mechanisms to capture value (Teece 2007). Additionally, the ability to productize the software product grew exponentially, as the Company X started reconfiguring the technical base of the software, enabling greater modularity and scalability. This modularity and scalability create more ways to productize the value offering, since every single feature can be combined with a set of different value-creating features. However, this added commercial flexibility isn’t inherently a good thing, as it makes more difficult to find the right ways of productization, since too many options to choose from for the customer may alter negatively the customers’ willingness to buy a subscription. In the end, modular software base, its positive effect on *productization* and product-led sales, develop the case company’s seizing capability, as the firm can seek to create economic rents from every feature that provides value for the users.

The last identified microfoundation of the seizing dynamic capability, *3-portfolio management structure* considers the product management decision-making activities which are mostly done by the managers of the product team. This microfoundation is similar to Teece’s (2007) thoughts on seizing’s foundation of selecting decision-making protocols. The newly hired CTO brought the 3-portfolio management framework to guide the product management in a holistic manner.

> The 3-portfolio model brings more to that area, as it does, if it works well, it produces such a thing as it does, holistic, three hundred and sixty degrees of visibility to our product for doing business, that there's like the opportunity [portfolio] side, there is what product development does when it's established in an opportunity that ‘now that's such a good opportunity that it's going to be

> done'. It then goes to the development portfolio. Then when it's done in production, it becomes part of offering area. – CTO

In the history of Company X, there had been, for example, a one-time project on productization and other acts of product management taking place as a project or as ad-hoc process. However, one of the reasons why this kind of management framework was implemented in the case company was that these kind of projects and ad-hoc acts should be replaced with continuous and systematic management of product management, which this 3-portfolio model should facilitate. Additionally, the *3-portfolio management structure* acts as a catalyst for *productization* microfoundation:

> Offering area, how do we price, how do we productize, what kind of product packages do we have? It improves our ability to maximise our existing intellectual property's value creation. –CTO

4.3 Transforming

The last aggregated dimension consists of three developed microfoundations that manifest the transforming dynamic capability in Company X's: asset *cospecialization*, *reconfiguring the product* and *agility enabling culture*.

Asset cospecialization (figure 10) represent team structures, activities and processes that enable cospecialization. According to Teece (2007), cospecialization can take place from asset to asset, meaning that the assets are used in conjunction creating more value than the asset would do without the integrative usage. In the case company, *asset cospecialization* occurred from integrating professionals from various functional areas to work together.

> Well, of course, it [scrum] has brought it in a way, like getting the expertise of the teams more to planning, that we have as we see that we have a business need for this, then someone with the team can refine it more carefully, that whatever the need is. I feel like taking advantage of [PM1]'s and [LD]'s expertise. It works really well for us that we have such different perspectives from where we come from that I have strong product expertise, and then again [PM1] has more of higher-level, kind of business know-how and [LD] has that technical architecture side, so our discussion are very fruitful. – PM2

> We try to work together a lot. That at least in my job it makes it easier to have as many in it. A pair of eyes in it. Maybe half and half of the work is like independent and then about half just [other designer] or others like product team members. – PD

> But in the future, I see that it's like the operations team and these Scrum teams so they're getting closer to each other that it can be that it might be like devops know-how might be in the future Part of that Scrum team without it being like a separate operations team. – CTO

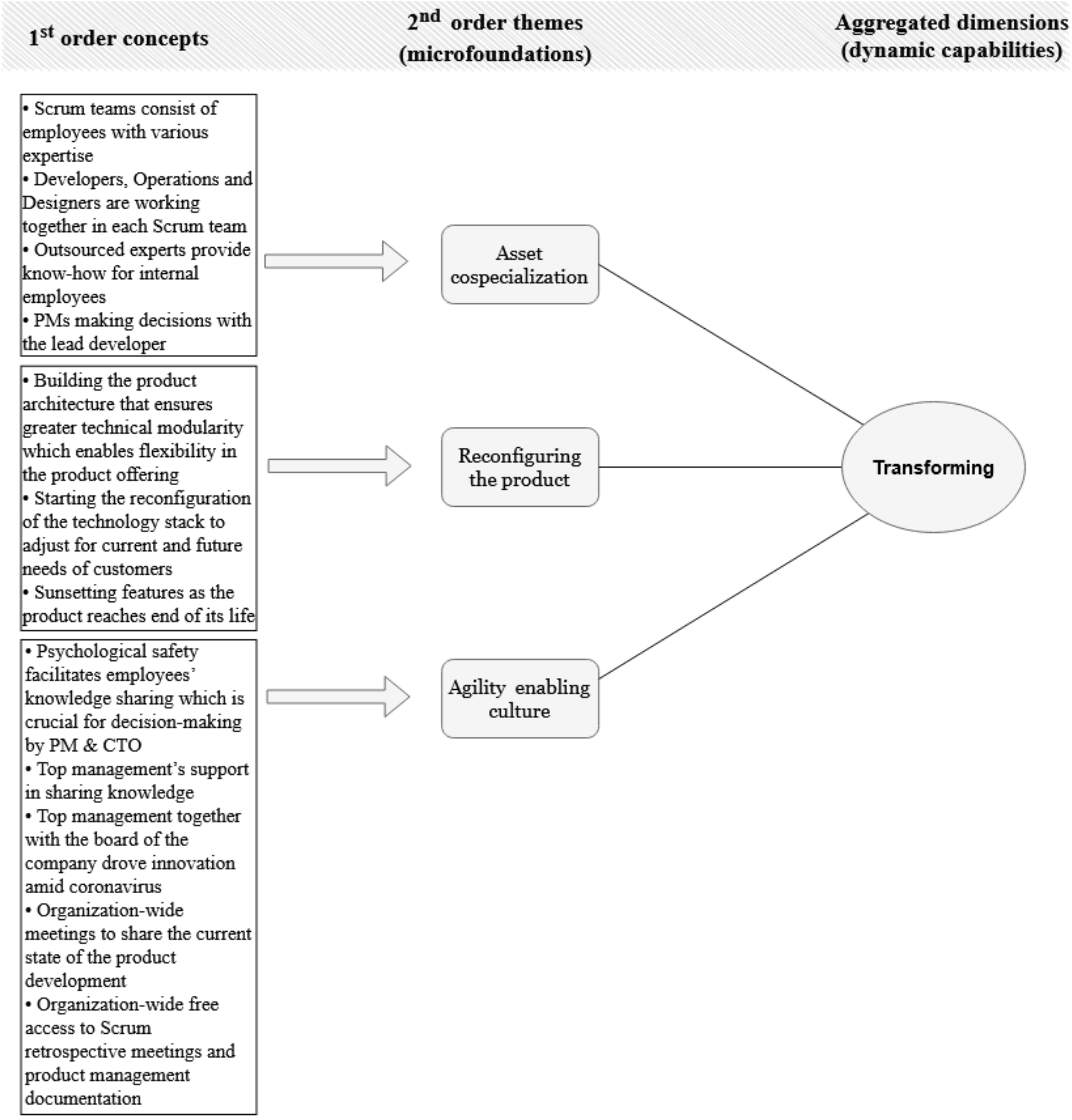

1st order concepts
2nd order themes (microfoundations)
Aggregated dimensions (dynamic capabilities)
• Scrum teams consist of employees with various expertise
• Developers, Operations and Designers are working together in each Scrum team
• Outsourced experts provide know-how for internal employees
• PMs making decisions with the lead developer
• Building the product architecture that ensures greater technical modularity which enables flexibility in the product offering
• Starting the reconfiguration of the technology stack to adjust for current and future needs of customers
• Sunsetting features as the product reaches end of its life
• Psychological safety facilitates employees' knowledge sharing which is crucial for decision-making by PM & CTO
• Top management's support in sharing knowledge
• Top management together with the board of the company drove innovation amid coronavirus
• Organization-wide meetings to share the current state of the product development
• Organization-wide free access to Scrum retrospective meetings and product management documentation
Asset cospecialization
Reconfiguring the product
Agility enabling culture
Transforming

Figure 10 Transforming microfoundations

Asset cospecialization took place in many areas of product management. Firstly, the value that results from the product managers working in conjunction with the lead developer is surely higher than it would be if the product managers were to work either alone or together without anyone providing technical expertise. The resulting value manifests in better decision-making in product management, especially when doing the specification for the upcoming sprints. Additionally, *asset cospecialization* occurred in the Scrum teams; the operations team is brought into the Scrum teams, and as a result, the Scrum teams consist of employees with development, operations, security and business expertise. The knowledge of the different kinds of professionals is integrated within the Scrum sprints, creating idiosyncratic and hard to imitate intangible assets that are built on know-how for Company X. Lastly, Company X's product managers worked closely with the product designers, mainly to influence designers with business understanding regarding efficient resource utilization. Close collaboration with the aforementioned counterparts is also recommended by Zorzetti et al. (2022).

Reconfiguring the product consists of the activities that the product team does for restructuring the software product. The CTO and PM1 were explicitly hired to guide the team towards transformation. Company X had been running its business for almost 15 years at the time of the interviews, and it became clear that the customer needs and the business itself had grown as the event management software market evolved. Some changes that acted as facilitators for the reconfiguration were the larger number of end-users (event participants) and customer requirements, such as different features and security aspects. The company felt unable to adapt to these changes without reconfiguring the product since technical debt had risen to levels that made development work either impossible or very inefficient. Thus, the older version of the product is reaching the end of its life, and that initiated the continuous restructuring towards an improved platform.

> We will make the decision that we sunset the old one at some point, so that there is quite clear [process]. In a way, we turn it so that customers use the new and then sunset the old implementation away. So, there will be something that comes with new developments, so we remove the old away constantly… We have to resource the classic also, so we have a classic's resourcing separately. We think all the time, how we sunset classic. So, let's think about what we can clean up from there, and let's think about how old stuff can be cleaned up and what we do to get it out. It is not so much of cleaning the code, but rather sunsetting removes stuff, removes data, which makes the structure more lightweight, which makes it run better... Instead of customer support team teaching the users, it [guidance] comes from the software, so it will be done directly in the system. – PM1

The product managers are tasked with allocating the available resource in the product team to sunset the classic product while orchestrating the development of the new product at the same time. This reconfiguration is at the core of the transforming dynamic capability, it is deployed in the continuous alignment and realignment of the intellectual property to better address the changing customer needs. *Reconfiguring the product* as an underpinning process for the transforming dynamic capability is also strongly intertwined with the *Scrum* microfoundation of the seizing dynamic capability. The agile methodology facilitates the reconfiguration during specification and release planning:

> They could say there, ‘this is what was hoped for, this is completely impossible to implement, this is not possible, instead, something like this could be done’. If we would do this, then we change the specification and say ‘in fact, well, that's a bit better way to do it’.

The technical transformation of the product also benefitted from the sensing dynamic capability and particularly its *gathering market intelligence* microfoundation. For example, the product team had become aware of the open-source Laravel framework and technical developments in database services, such as the Google Cloud platform and Aiven, which they started utilizing amid the reconfiguration. The utilization of this open-source framework resulted in faster development time and saved development resource since the framework provided various ready-made foundations for the features to be developed.

> Of course, when you modernize tools, they automatically come with like safety features included. With this newer next gen, the development framework is being implemented where, a lot of these issues have already been solved. There's nothing like that in original code [older platform], nothing already solved issues as It's done starting from zero inside the house. It hasn't necessarily occurred during the development that something like this or that should be considered. The new framework consists of thousands of hours of research already, for free. It’s like ‘here is already those 100 things solved, just use this framework how it is supposed to be used’. – LD

> GCP or these other clouds, you get them like a ready-made service with its parent engine. Very far, like that setup and maintaining that it is, It's a pretty big deal because it's like operations from a type of traditional doing, so databases Installation, care, maintenance. That's it's a pretty demanding job. The fact that you're buying that service, you're saving quite a lot then in the cost of personal work in the long run. – CTO

> Think about the Laravel framework what it brings there, it brings, like how fast you can develop it on top of that framework. – PM1

Lastly, perhaps the most important effect that this deployment of transformation dynamic capability has, is that the technical infrastructure and the product’s architecture are built on a robust foundation. These technical foundations are critical for ensuring future modularity and agility, and those changes also come with cost savings and more efficient resource allocation.

> Think like product development, that the efficiency of product development depends on the tech stack and processes, and it is precisely the set, if those two things are not in order, then your product development cannot be in any way. – PM1

Agility enabling culture represents the 1st-order concepts that function as agility enabling factors in the case company's culture. Agility here refers to the capacity of the case company to efficiently and effectively redeploy resources to create value as circumstances require (Teece et al. 2016). It became evident from the interviews that the employees in the product team are encouraged to share their opinions and expertise with a low threshold. Culture, more specifically psychological safety, facilitates knowledge sharing within the scrum teams and in product management in general.

> For example, I've been for a year in the house so that I base my decision, if necessary, that I must be consulted. They must challenge what I'm thinking. If I suggest something, then I want to hear that 'no, here's this and there's that and what about this and what about that and what works when there's this' and then when I get it all, then I'll know… And then when they get used to the fact that it gets noticed, that it affects, so they give them [more]. I think it's really important, nobody wants to work if it doesn't show up anywhere and doesn't affect anything and your opinion does not matter – PM1

> I usually share all of these in general [channel in Slack communication tool] and then also that whenever I've done an interview like already in a series with customers or with [other employees], I always try to share those results – SD

> Well, it has taken much like repetition of the message. Long-term repetition in practice that, like we have this kind of a problem. And like try to convince the management team, CEO and colleagues. – LD

The hiring of the new CTO and PM1 and the implementation of scrum methodology also improved the culture in the product team:

> In a way, this is so influenced by the whole that how the product is managed from here, as well as from my point of view, if you consider that in the past it was very much like the personalised for the CTO, CTO is leading the decision-making… I didn't feel like I could really be a PM anyway, that when it came to that change of person, now I can definitely do it, and then I feel like taking advantage of [PM1]'s and [LD]'s expertise. It works really well for us that we have such different perspectives from where we come from that.

Company X had also top management collaborating on the day-to-day operations, gathering knowledge and market intelligence. The CEO himself was driving innovation with the board of the company when the coronavirus pandemic started. This is leadership by example, which aids in developing that culture, in which the employees are free to share their ideas for better addressing the needs of the market. *Agility enabling culture* is foundational for timely adaptation to the opportunities and threats that the company senses. It is critical for managers to be on top of the situations, therefore employees at all levels of the hierarchy must feel safe to share their opinions. In case the leadership is perceived as tough and too authoritative, it may decrease the psychological safety of the employees, which may result in not reconfiguring the specifications of the features, the release plans, or the roadmaps if the critical information is not shared to the responsible decision-makers. Thus, *agility enabling culture* employs the transforming dynamic capability by enabling change with low threshold knowledge sharing.

5 Discussion

The aim of this study was to investigate how dynamic capabilities and their underpinning factors, known as microfoundations, enable the software company to adapt to change in the surrounding business environment. The study focused on the Teecean typology of the disaggregation of dynamic capabilities: sensing, seizing, and transforming capabilities (Teece 2007; Teece et al. 1997). A qualitative case study was adopted to conduct the empirical research, providing a deep understanding of the phenomena in the case company. In total 14 microfoundations were found within the area of product management: 4 for sensing, 7 for seizing, and 3 for transforming (table 7).

5.1 Dynamic capabilities and agile software product organization

RQ: *how dynamic capabilities in a software product organization enable adaptation to change in the business environment?*

The findings of this study show that sensing, seizing, and transforming dynamic capabilities deploy their underpinning microfoundations in ways that enable the company to sense opportunities and threats, address those findings, and transform its (intangible) resource base as required for adapting to change in the environment. To explain how the adaptation to change may take place, it is proposed here to view the phenomenon's outcomes through three distinct layers of agility (figure 11) (cf. Teece et al. 2016). These principles of agility occur when the microfoundations and their corresponding dynamic capabilities are manifested:

- Agility foundations
- Agility fostering culture
- Learning fast

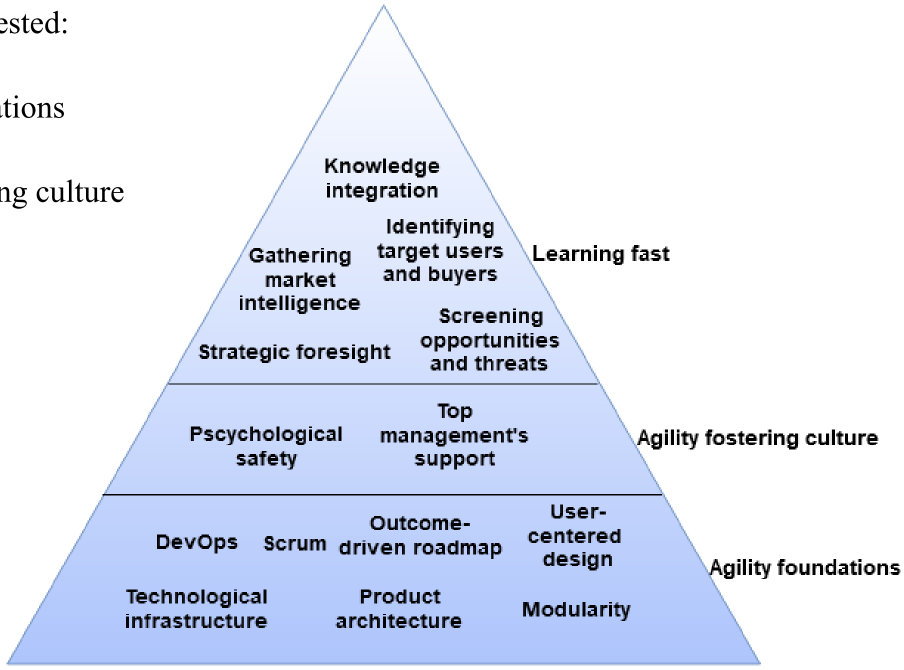


Figure 11 The principles of agility in a software product organisation

Agility foundations (figure 11) provide a platform to express the firm's seizing dynamic capability in addressing opportunities and threats promptly. This foundation is constructed with technological concepts, such as technological infrastructure and product architecture, and agile development concepts. Nowadays, scalable cloud infrastructure and modular architecture, especially with microservices, create a robust foundation, on which the product management can operate on through different agile development concepts, like Scrum and DevOps. This study shows that agile development that creates value for users and organizational agility cannot truly exist if the development environment does not cultivate it. It requires the dynamic capabilities of sensing and transforming to overcome such an inefficient situation.

Additionally, *agility foundations* help the company to revert from unfavourable path dependencies, which means the function of a firm's existing position and the possible paths forward (Teece et al. 1997). A software product organization is inherently path dependent, at least on the product level due to the characterization of roadmaps — they are a plan of a firm's future directions (Suomalainen et al. 2011). Therefore, investing in creating a robust agility base, both in the technical and methodological sense, creates a stronger transforming dynamic capability for the firm. In the event of radical change in the market that forces the firm to reconfigure its value offering, the agile foundations, especially the modularity (Teece et al. 2016; Teece 2007), can help the company address the change by making it easier to develop a product offering for a new category. The benefits of modularity are also seen in the number of ways to productize the software product.

In a similar sense to the foundations of agility stated here, Baskerville and Pries-Heje (2021) suggest that IT agility can enhance a firm's resilience, and its ability to bounce back from challenges, thus enabling adaptation to change in the surrounding environment. Furthermore, highly agile and resilient firms prosper in response to competitive threat, they deliver higher financial performance (ROI 150%, ROE 500%) than companies that don't obtain such agile foundations, while being dynamic to reconfigure the course of operations (Pulakos et al. 2019).

Agility fostering culture is the following layer after agile foundations. It acts as a facilitator and mediator between the bottom and top of the pyramid. Know-how and opinions won't be shared and utilized in decision-making if the culture doesn't foster the open communication of opportunities and threats. For the software company to be dynamic, to reconfigure its development operations, the release plans and roadmaps, it must obtain an agility fostering culture, in which the professionals feel safe to share their know-how and sensed threats. Mishra and Otaiwi (2020) acknowledged the importance that a suitable culture has on the successful implementation of DevOps. A suitable culture promotes sharing of knowledge and shared responsibility for delivering a quality product (Mishra & Otaiwi 2020). Similarly, Teece et al. (2016) advocate for the importance that values, culture and collective ability have on the ability deploy the dynamic capabilities. Agility fostering culture is arguably vital for extracting value of the sensing dynamic capability.

Learning fast comprises the top of the pyramid and is the final outcome of the dynamic capabilities phenomena that support adaptation to change. Knowledge integration to address opportunities and threats cannot take place successfully in an agile manner if the first two layers are not in place. The successful utilization of knowledge in product management results in less waste developed, efficient resource utilization and created customer value, by developing products that customers truly want to use. Previously Teece (2007) outlined knowledge management activities to underpin the transforming dynamic capability. However, this study shows that in a software company, knowledge integration deploys the seizing dynamic capability within the agile development concepts, such as scrum sprints and outcome-based roadmapping, which are the microfoundations to address opportunities or threats.

Learning fast is largely the result of a strong sensing capability that cultivates knowledge flows that function as catalysts for seizing and transforming dynamic capabilities and their underlying microfoundations. Theoretically, these are distinct concepts, but empirically the dynamic capabilities and their microfoundations are intertwined, and their deployment is often consequent. Consistent with recent research by Zorzetti et al. (2022), *agile foundations* are essential for *learning fast* as the successful combination of agile development concepts improves customer centricity and (strategic) planning.

To conclude, dynamic capabilities and their undergirding microfoundations are helpful for a firm to promote agility, meaning its capability to reconfigure its resource allocation to value creating and protecting activities in the face of adverse change in the environment (Teece et al. 2016).

5.2 Implications

This study complements the dynamic capabilities literature by researching the previously overlooked software industry and focusing on the microfoundations of the sensing, seizing and transforming dynamic capabilities. Furthermore, this study's findings build on the research by Teece et al. (2016) which presented the dynamic capabilities framework for approaching agility. The findings conceptualize how the three dynamic capabilities and their microfoundations are manifested to enable agility.

The findings of this research demonstrate how a software product organization can adapt to change in the surrounding business environment by starting to build an agility foundation for the company. Managers must be aware of the importance that a suitable, agility fostering culture has for integrating knowledge for effective decision-making. It is of utmost importance for the managers to create a psychologically safe culture for the professionals to share their opinions on sensed opportunities and threats. Otherwise, a company may end up too far on an unfavourable path, and the upcoming reconfiguration of operations will end up being costly. To construct the agility principles for a software product company, it must obtain the sensing, seizing and transforming dynamic capabilities both at the organizational and managerial level, especially in the area of product management.

5.3 Limitations and future research

This study focused on a single software product organization operating in the event management industry. The amount of data collected would have been greater by either focusing on more companies or by conducting a longitudinal case study. Spending a long time with the case companies would help in analysing the dynamic capabilities since the phenomena take place for long periods in the case of reconfiguring the resource base and processes of a company. Second, the study was conducted as a qualitative case study on interpretivist philosophical foundations, thus the findings are subject to the interpretation of the researcher.

Further studies should investigate different industries and sizes of companies because product management is orchestrated in alignment with the needs of the specific context. Thus, both product management and dynamic capabilities are context-dependent. Companies with multiple software products should have different microfoundations and deploy dynamic capabilities in different ways. Additionally, since technical debt may negatively affect the agile software development and perhaps overall agility of the company, it should be inspected how dynamic capabilities occur in such high-tech companies.

6 Conclusions

This study offers answers to the question of how a software product organization can adapt to change in the environment. The question is inspected through the lens of dynamic capabilities and their underlying microfoundations, as they enable companies to address rapidly changing markets (Teece et al. 1997; Teece 2007).

Based on the findings of this study, the capacity of a software product organization to adapt to change is dependent on its ability to deploy the sensing, seizing and transforming dynamic capabilities through their underpinning microfoundations. A manifestation of this phenomenon improves the firm's agility through the embodiment of agility principles (figure 11). The resulting principles that foster agility are:

- Agility foundations: Essential for the utilization of the seizing dynamic capability.
- Agility fostering culture: A psychologically safe culture ensures that professionals feel safe to share their opinions of sensed threats and opportunities.
- Learning fast: The knowledge flows started by the sensing dynamic capability are put into use in effective decision-making within product management. The three dynamic capabilities are all deployed for achieving fast learning that occurs through knowledge integration.

References

Ameller, D. – Farre, C. – Franch, X. – Valerio, D. – Cassarino, A. (2017) Towards continuous software release planning. 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER), 402-406.

Arend, R. J. (2014) Entrepreneurship and dynamic capabilities: how firm age and size affect the 'capability enhancement–sme performance'relationship. *Small Business Economics*, Vol. 42 (1), 33-57.

Arend, R. J. – Bromiley, P. (2009) Assessing the dynamic capabilities view: spare change, everyone? *Strategic organization*, Vol. 7 (1), 75-90.

Arndt, F. – Pierce, L. (2018) The behavioral and evolutionary roots of dynamic capabilities. *Industrial and corporate change*, Vol. 27 (2), 413-424.

Babelytė-Labanauskė, K. – Nedzinskas, Š. (2017) Dynamic capabilities and their impact on research organizations' r&d and innovation performance. *Journal of Modelling in Management*, Vol. 12 (4), 603-630.

Barney, J. (1991) Firm resources and sustained competitive advantage. *Journal of management*, Vol. 17 (1), 99-120.

Baskerville, R. – Pries-Heje, J. (2021) Achieving Resilience through Agility. ICIS 2021 Proceedings. 8. 1-18.

Beske, P. – Land, A. – Seuring, S. (2014) Sustainable supply chain management practices and dynamic capabilities in the food industry: a critical analysis of the literature. *International journal of production economics*, Vol. 152, 131-143.

Bitencourt, C. C. – de Oliveira Santini, F. – Ladeira, W. J. – Santos, A. C. – Teixeira, E. K. (2020) The extended dynamic capabilities model: a meta-analysis. *European Management Journal*, Vol 38 (1), 108-120.

Borden, N. H. (1964) The concept of marketing mix. *Journal of Advertising Research*, Vol. 4 (2), 2–7.

Cautela, C. – Simoni, M. – Moran, P. (2021) Microfoundations of dynamic design capabilities: an empirical analysis of "excellent" Italian design firms. *Journal of Product Innovation Management,* 1-22.

Conboy, K. – Mikalef, P. – Dennehy, D. – Krogstie, J. (2020) Using business analytics to enhance dynamic capabilities in operations research: a case analysis and research agenda. *European Journal of Operational Research*, Vol. 281 (3), 656-672.

Creswell, J. W. - Creswell, J. D. (2018) Research design : qualitative, quantitative, and mixed methods approaches. Fifth edition. Sage, Los Angeles.

Cyert, R. M. – J. G. March (1963) A behavioral theory of the firm. Prentice-Hall, Englewood Cliffs, New Jersey.

Danneels, E. (2002) The dynamics of product innovation and firm competences. *Strategic management journal*, Vol. 23 (12), 1095-1121.

Ebert, C. (2007) The impacts of software product management. *Journal of systems and software*, Vol. 80 (6), 850-861.

Ebert, C – Brinkkemper, S. (2014) Software product management - an industry evaluation. *Journal of Systems and Software*, Vol. 95, 10–18.

Eisenhardt, K. M. (1989) Building theories from case study research. *Academy of management review*, Vol. 14 (4), 532-550.

Eisenhardt, K. M. – Martin, J. A. (2000) Dynamic capabilities: what are they? *Strategic Management Journal*. Vol. 21 (10/11), 1105–1121.

Eriksson, P – Kovalainen, A. (2008) Qualitative research evaluation. In: Qualitative methods in business research. 290-297. SAGE Publications Ltd.

Fainshmidt, S. – Pezeshkan, A. – Lance Frazier, M. – Nair, A. – Markowski, E. (2016) Dynamic capabilities and organizational performance: a meta-analytic evaluation and extension. *Journal of Management Studies*, Vol. 53 (8), 1348-1380.

Felin, T. – Foss, N. J. – Heimeriks, K. H. – Madsen, T. L. (2012) Microfoundations of routines and capabilities: individuals, processes, and structure. *Journal of Management Studies*, Vol. 49 (8), 1351-1374.

Felin, T. – Foss, N. J. – Ployhart, R. E. (2015) The microfoundations movement in strategy and organization theory. *Academy of Management Annals*, Vol. 9 (1), 575-632.

Fitzgerald, B. – Stol, K. J. (2017) Continuous software engineering: a roadmap and agenda. *Journal of Systems and Software*, Vol. 123, 176-189.

Gioia, D. A. – Corley, K. G. – Hamilton, A. L. (2013) Seeking qualitative rigor in inductive research: notes on the Gioia methodology. *Organizational research methods*, Vol. 16 (1), 15-31.

Haarhaus, T. – Liening, A. (2020) Building dynamic capabilities to cope with environmental uncertainty: The role of strategic foresight. Technological Forecasting and Social Change, 155, 120033. 1-15

Hajiheydari, N. – Delgosha, M. S. – Talafidaryani, M. (2019) Is research theoretical foundation: theories used and the future path. In International Conference on Research and Practical Issues of Enterprise Information Systems, 24-39.

Helfat, C. E. – Finkelstein, S. – Mitchell, W. – Peteraf, M. A. – Singh, H. – Teece, D. J. – Winter, S. G. (2007) Dynamic capabilities: understanding strategic change in organizations. Blackwell, Malden.

Helfat, C. E. – Winter, S. G. (2011) Untangling dynamic and operational capabilities: strategy for the (n) ever-changing world. *Strategic management journal*, Vol. 32 (11), 1243-1250.

Helfat, C. E. – Peteraf, M. A. (2015) Managerial cognitive capabilities and the microfoundations of dynamic capabilities. *Strategic management journal*, Vol. 36 (6), 831-850.

ISPMA (2021) <https://ispma.org/framework/>, retrieved 18.10.2021

Jantunen, A. – Tarkiainen, A. – Chari, S. – Oghazi, P. (2018) Dynamic capabilities, operational changes, and performance outcomes in the media industry. *Journal of Business Research*, Vol. 89, 251-257.

Jaworski, B. J. – Kohli, A. K. (1993) Market Orientation: Antecedents and Consequences. *Journal of Marketing*, Vol. 57 (3), 53–70.

Kay, N. M. – Leih, S. – Teece, D. J. The role of emergence in dynamic capabilities: a restatement of the framework and some possibilities for future research, *Industrial and Corporate Change*, Vol. 27 (4), 623–638.

Khan, O. – Daddi, T. – Iraldo, F. (2020) Microfoundations of dynamic capabilities: Insights from circular economy business cases. *Business Strategy and the Environment*, Vol 29 (3), 1479–1493.

Khan, O. – Daddi, T. – Iraldo, F. (2021) Sensing, seizing, and reconfiguring: key capabilities and organizational routines for circular economy implementation. *Journal of Cleaner Production*, Vol. 287. 1-12.

Kerr, C. – Phaal, R. (2020) Technology roadmapping: industrial roots, forgotten history and unknown origins. *Technological Forecasting and Social Change*, Vol. 155, 1-16.

Kerr, C. – Phaal, R. (2021). Roadmapping and Roadmaps: Definition and Underpinning Concepts. IEEE Transactions on Engineering Management, Vol. 69 (1), 6-16.

Kittlaus, H. – Clough, P. N. (2009) Software product management and pricing: key success factors for software organizations (pp. 250-265). Springer, Berlin, Kittlaus, H.– Fricker, S. A. (2017) Software Product Management. Springer, Berlin.

Kump, B. – Engelmann, A. – Kessler, A. – Schweiger, C. (2019) Toward a dynamic capabilities scale: measuring organizational sensing, seizing, and transforming capacities. *Industrial and Corporate Change*, Vol 28 (5), 1149-1172.

Langley, A. – Abdallah, C. (2011) Templates and turns in qualitative studies of strategy and management. In Building methodological bridges. Emerald Group Publishing Limited.

Linde, L. – Sjödin, D. – Parida, V. – Wincent, J. (2021) Dynamic capabilities for ecosystem orchestration a capability-based framework for smart city innovation initiatives. Technological Forecasting and Social Change, Vol. 166, 1-12.

Macher, J. T. – Mowery, D. C. (2009) Measuring dynamic capabilities: practices and performance in semiconductor manufacturing. *British Journal of Management*, Vol. 20, 41-62.

Maglyas, A. – Nikula, U. – Smolander, K. – Fricker, S. A. (2017) Core software product management activities. *Journal of Advances in Management Research*, Vol. 14 (1), 23–45.

Majhi, S. G., Mukherjee, A., & Anand, A. (2021). Business value of cognitive analytics technology: a dynamic capabilities perspective. *VINE Journal of Information and Knowledge Management Systems,* 1-19.

March, J. G. – H. A. Simon (1958) Organizations. Wiley, New York.

Marinković, M. – Al-Tabbaa, O. – Khan, Z. – Wu, J. (2022) Corporate foresight: a systematic literature review and future research trajectories. *Journal of Business Research*, 144, 289-311.

Menguc, B. – Auh, S. (2006) Creating a firm-level dynamic capability through capitalizing on market orientation and innovativeness. *Journal of the academy of marketing science*, Vol. 34 (1), 63-73.

Myers, M. D. (2013) Qualitative research in business & management. 2nd ed. Sage, Thousand Oaks (CA).

Münch, J. – Trieflinger, S. – Lang, D. (2019a) What's hot in product roadmapping? Key practices and success factors. In International Conference on Product-Focused Software Process Improvement. 401-416, Springer.

Münch, J. – Trieflinger, S. – Lang, D. (2019b) Product roadmap – from vision to reality: a systematic literature review. In 2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), 1-8.

Naldi, L. – Wikström, P. – Von Rimscha, M.B. (2014) Dynamic capabilities and performance. *International Studies of Management and Organization*, Vol. 44 (4), 63-82.

Nedzinskas, Š. – Pundzienė, A. – Buožiūtė-Rafanavičienė, S. – Pilkienė, M. (2013) The impact of dynamic capabilities on sme performance in a volatile environment as moderated by organizational inertia. *Baltic Journal of Management*, Vol. 8 (4), 376-396.

Nelson, R. – S. G. Winter (1982) An evolutionary theory of economic change. Belknap Press, Cambridge.

Neves, M. (2020) Event tech vendors grow 4 fold in past year: here's all 800. <https://www.eventmanagerblog.com/event-tech-vendors-800>, retrieved 18.10.2021.

Pablo, A. L. – Reay, T. – Dewald, J. R. – Casebeer, A. L. (2007) Identifying, enabling and managing dynamic capabilities in the public sector. *Journal of management studies*, Vol. 44 (5), 687-708.

Pavlou, P.A. – El Sawy, O.A. (2011) Understanding the elusive black box of dynamic capabilities. *Decision Sciences*, Vol. 42 (1), 239-273.

Peteraf, M. – Di Stefano, G. – Verona, G. (2013) The elephant in the room of dynamic capabilities: bringing two diverging conversations together. *Strategic management journal*, Vol. 34 (12), 1389-1410.

Phaal, R. – Farrukh, C. J. – Probert, D. R. (2004) Technology roadmapping — a planning framework for evolution and revolution. *Technological forecasting and social change*, Vol. 71 (1-2), 5-26.

Phaal, R. – Farrukh, C. J. – Probert, D. R. (2010) Roadmapping for strategy and innovation: aligning technology and markets in a dynamic world. Institute for Manufacturing.

Piekkari, R., & Welch, C. (2018) The case study in management research: beyond the positivist legacy of Eisenhardt and Yin. In: The SAGE handbook of qualitative business and management research methods, 345-358. Sage Publications Ltd, London.

Pulakos, E. D. – Kantrowitz, T. – Schneider, B. (2019) What leads to organizational agility: it's not what you think. Consulting Psychology Journal: Practice and Research, Vol. 71 (4), 305-320.

Porter, M. E. (1979) How competitive forces shape strategy. *Harvard Business Review*.

Gephart, R. P. (2018). Qualitative research as interpretive social science. In: The SAGE handbook of qualitative business and management research methods, 33-53. Sage Publications Ltd, London.

Ridder, H. – Hoon, C. – McCandless. A. (2009) The 'hows' of dynamic capabilities: underlying processes and their drivers. Academy of Management proceedings. 1-8.

Rubert, M. – Farias, K. (2022) On the effects of continuous delivery on code quality: a case study in industry. Computer Standards & Interfaces, Vol. 81, 103588. 1-20.

Ruhe, G. – Saliu, M. O. (2005) The art and science of software release planning. IEEE software, Vol. 22 (6), 47-53.

Schilke, O. (2014) On the contingent value of dynamic capabilities for competitive advantage: the nonlinear moderating effect of environmental dynamism. *Strategic management journal*, Vol. 35 (2), 179-203.

Schilke, O. – Hu, S. – Helfat, C. E. (2018) Quo vadis, dynamic capabilities? A content-analytic review of the current state of knowledge and recommendations for future research. *Academy of Management Annals*, Vol. 12 (1), 390–439.

Shafia, M.A. – Shavvalpour, S. – Hosseini, M. – Hosseini, R. (2016) Mediating effect of technological innovation capabilities between dynamic capabilities and competitiveness of research and technology organisations. *Technology Analysis and Strategic Management*, Vol. 28 (7), 811-826.

Shapiro, C. (1989) The theory of business strategy. *The Rand journal of economics*, Vol. 20 (1), 125-137.

Statista (2021) <https://www.statista.com/statistics/505243/worldwide-software-as-a-service-revenue/>, retrieved 1.11.2021.

Sunder, M. V. – Ganesh, L. S. – Marathe, R. R. (2019) Dynamic capabilities: a morphological analysis framework and agenda for future research. *European Business Review*. Vol. 31 (1), 25-63.

Suomalainen, T. – Salo, O. – Abrahamsson, P. – Similä, J. (2011) Software product roadmapping in a volatile business environment. *Journal of Systems and Software*, Vol. 84 (6), 958-975.

Talafidaryani, M. (2021) A text mining-based review of the literature on dynamic capabilities perspective in information systems research. *Management Research Review*, Vol. 44 (2), 236-267.

Teece, D. J. (2007) Explicating dynamic capabilities: the nature and microfoundations of (sustainable) enterprise performance. *Strategic Management Journal*. Vol 28, 1319–1350.

Teece, D. J. (2009) Dynamic capabilities and strategic management organizing for innovation and growth. Oxford University Press, Oxford.

Teece, D.J (2012) Dynamic capabilities: routines versus entrepreneurial action*. Journal of management studies*, Vol. 49 (8), 1395-1401.

Teece, D. J. (2014) The foundations of enterprise performance: dynamic and ordinary capabilities in an (economic) theory of firms. *Academy of management perspectives*, Vol. 28 (4), 328-352.

Teece, D. J. – Pisano, G. – Shuen, A. (1997) Dynamic capabilities and strategic management. *Strategic Management Journal*, Vol 18, 509–533.

Teece. D. J. – Ali-Aali, A. (2012) Knowledge assets, capabilities and the theory of the firm. In: *Handbook of Organizational Learning and Knowledge Management,* ed. by Easterby-Smith, M. – Lyles, M. A, 505-534. John Wiley & Sons, West Sussex.

Teece. D. J. – Peteraf, M. – Leih, S. (2016) Dynamic capabilities and organizational agility: risk, uncertainty, and strategy in the innovation economy. *California management review*, Vol. 58 (4), 13-35.

Venkatesh, V.C. – Dasgupta, M. – Prashar, A. – Andersen, T.J. (2021) Dealing with surprise attacks: decomposing erm as a dynamic capability to handle crises. *Journal of Small Business and Enterprise Development*, Vol. 28 (4), 515-536.

Walsham, G. (1995) Interpretive case studies in is research: nature and method. *European Journal of information systems*, Vol. 4 (2), 74-81.

Walsham, G. (2006) Doing interpretive research. *European journal of information systems*, Vol. 15 (3), 320-330.

Wang, C. L. – Ahmed, P. K. (2007). Dynamic capabilities: a review and research agenda. *International journal of management reviews*, Vol. 9 (1), 31-51.

Weerd, I, van de – Brinkkemper, S. – Nieuwenhuis, R. – Versendaal, J. – Bijlsma, L. (2006) On the creation of a reference framework for software product management: validation and tool support. 2006 International Workshop on Software Product Management (IWSPM'06-RE'06 Workshop), 3-12.

Wilden, R. – Gudergan, S.P. – Nielsen, B.B. – Lings, I. (2013) Dynamic capabilities and performance: strategy, structure and environment. *Long Range Planning*, Vol. 46 (1/2), 72-96.

Wilden, R. – Gudergan, S.P. (2015) The impact of dynamic capabilities on operational marketing and technological capabilities: investigating the role of environmental turbulence. *Journal of the Academy of Marketing Science*, Vol. 43 (2), 181-199.

Wilden, R., Devinney, T. M., & Dowling, G. R. (2016). The Architecture of Dynamic Capability Research Identifying the Building Blocks of a Configurational Approach. Academy of Management Annals, Vol. 10 (1), 997–1076.

Williamson, O. E. (1991) Comparative economic organization: The analysis of discrete structural alternatives. *Administrative science quarterly*, Vol. 36 (2), 269-296.

Willyard, C. H. ,– McClees, C. W. (1987) Motorola's technology roadmap process. *Research management*, Vol. 30 (5), 13-19.

Winter, S. G. (2003) Understanding dynamic capabilities. *Strategic Management Journal*, Vol. 24 (10), 991–995.

Yin, R. K. (2018). Case study research and applications. (6th ed.). Sage publications inc, Thousand Oaks.

Zahra, S. A. – George, G. (2002) Absorptive capacity: a review, reconceptualization, and extension. *Academy of Management Review*, Vol. 27 (2), 185-203.

Zahra, S. A. – Sapienza, H. J. – Davidsson, P. (2006) Entrepreneurship and dynamic capabilities: a review, model and research agenda. *Journal of Management studies*, Vol. 43 (4), 917-955.

Zollo, M. – Winter, S. G. (2002) Deliberate learning and the evolution of dynamic capabilities. *Organization Science*, Vol. 13 (3), 339–351.

Zorzetti, M. – Signoretti, I. – Salerno, L. – Marczak, S. – Bastos, R. (2022) Improving agile software development using user-centered design and lean startup. *Information and Software Technology*, Vol. 141, 106718. 1-14.