Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Economics and Management
Dept. of Industrial Economics and Technology Management

Sebastian Kaland, Scott Aleksander Bekke,
Marcus Balcon

A study on no-code technology as a resource for competitive advantage

Master's thesis in Entrepreneurship
Supervisor: Øystein Widding
Co-supervisor: Geir Kjetil Hanssen
June 2023

NTNU
Norwegian University of
Science and Technology

SINTEF

Sebastian Kaland, Scott Aleksander Bekke, Marcus Balcon

A study on no-code technology as a resource for competitive advantage

Master's thesis in Entrepreneurship
Supervisor: Øystein Widding
Co-supervisor: Geir Kjetil Hanssen
June 2023

Norwegian University of Science and Technology
Faculty of Economics and Management
Dept. of Industrial Economics and Technology Management

TABLE OF CONTENTS

I Abstract & Preface

1 Introduction 9
- 1.1 Engaged Scholarship Design 9
  - 1.1.1 Real-world problem (P) 11
  - 1.1.2 Literature review and identified gap in the literature (A) 11
  - 1.1.3 Purpose of the study (F) 12
  - 1.1.4 Research questions (RQ) 13
  - 1.1.5 Method (M) 14
  - 1.1.6 Contribution (C) 14
- 1.2 Summary of the Engaged Scholarship Design for our study 15
- 1.3 Structure of the master´s thesis 17

2 No-Code as a Concept 19
- 2.1 The distinction between low-code and no-code 20
- 2.2 Opportunities and challenges of no-code 21
  - 2.2.1 The perceived advantages of no-code 21
  - 2.2.2 Challenges with no-code 22

3 Theoretical Foundation 23
- 3.1 Theoretical framework 23
  - 3.1.1 What is a resource 24
  - 3.1.2 Conditions for competitive advantage 25
  - 3.1.3 Summary of the VRIO-model 27
  - 3.1.4 Combining resources 28
- 3.2 Theoretical framework in relation to no-code 28
  - 3.2.1 No-code as a resource 29
  - 3.2.2 No-code as a valuable resource 29
  - 3.2.3 Combining firm resources and no-code 30
- 3.3 Theoretical framework applied 30

4 Research Method 33
- 4.1 Research design 33
  - 4.1.1 Choice of interview subjects 35
- 4.2 Data collection 37
  - 4.2.1 Interview guide 38
  - 4.2.2 Execution of case interviews 38

4.3 Data analysis 41
4.3.1 Within-case analysis 41
4.3.2 Cross-case analysis 43
4.4 Evaluation of the research questions 46
4.5 Ethical Aspects 46
4.6 Methodological limitations and reflection 47
5 Results and analysis 49
5.1 Within-case analysis 49
5.1.1 Case firm A 49
5.1.2 Case firm B 54
5.1.3 Case firm C 58
5.1.4 Case firm D 62
5.1.5 Case firm E 67
5.2 Cross-case analysis 72
5.2.1 Functionality restrictions and complexity 72
5.2.2 Vendor lock-in 73
5.2.3 No-code effects intangible resources in the firm 74
5.2.4 Technical competency is required to make full use of no-code 77
5.2.5 No-code enables faster time to market 78
5.2.6 Close customer relationships creates hard to replicate resources 79
5.3 Answer to research questions 80
5.3.1 "How does no-code technology interact with a firm's resources to create a competitive advantage?" 80
5.3.2 What are the key considerations and evaluation criteria that early stage firms should employ when considering to adopt no-code technology? 81
6 Discussion 83
6.1 The contribution of key findings to previous literature (C_A) 83
6.2 The contribution to practice (C_P) 84
7 Conclusion 87
8 Further research and implications 89
Bibliography 91

II Appendix
A Interview Guide 99
B Foundation for cross-case analysis 103

Part I

ABSTRACT & PREFACE

Abstract

This master's thesis examines the transformative role of no-code technology in shaping firms' competitive advantage and deconstructs the complex decision-making process behind the adoption of no-code development. Using a qualitative research approach and a multiple-case study design, we investigate how organizations evaluate no-code development platforms prior to adoption and their perceived effects on firm resources and competitive advantage.

According to our research, the evaluation process revolves around factors such as development speed, cost-effectiveness, simplicity of iteration, and reducing dependence on IT specialists. However, the complexity of the project, the need for customized solutions, and concerns regarding flexibility and security play a crucial role in determining whether traditional coding should be considered. We discovered that no-code technology can enhance a firm's competitive standing by lowering resource requirements, optimizing resources, and reducing development cycles.

In addition, we investigate the relationship between no-code technology and a firm's resources by applying resource-based theory in a no-code context. Findings indicate that no-code platforms improve a firm's ability to swiftly adapt and iterate their product, thereby optimizing their use of time and human capital. Furthermore, by democratizing the development process, no-code platforms can empower non-technical team members, thereby enhancing the organization's human capital.

The study offers researchers and practitioners valuable insights into the effects of no-code technology on competitive dynamics and resource allocation. It calls for additional research to investigate the applicability of no-code technology in a wider variety of contexts and its long-term effects on industry structures. This study provides a foundational understanding of no-code technology and its potential to disrupt traditional software development methods and business competitiveness.

Sammendrag

Denne masteroppgaven undersøker hvilken rolle no-code teknologi kan ha i å forme bedrifters konkurransefortrinn og bryter ned den komplekse beslutningsprosessen bak adopsjonen av no-code. Ved hjelp av en kvalitativ forskningsmetode og et multi-case studie design, undersøker vi hvordan organisasjoner til nå har vurdert no-code og forsøker å forklare de underliggende evalueringskriteriene som burde tas hensyn til før implementering.

Ifølge vår forskning dreier evalueringsprosessen seg om faktorer som utviklingshastighet, kostnadseffektivitet, enkelhet ved iterasjon og minsket avhengighet til IT-spesialister. Imidlertid spiller prosjektets kompleksitet, behovet for tilpassede løsninger, og bekymringer angående fleksibilitet og sikkerhet en avgjørende rolle i å bestemme om no-code skal vurderes.

I tillegg undersøker vi forholdet mellom no-code teknologi og bedriftens ressurser. Funnene indikerer at no-code plattformer forbedrer en bedrifts evne til raskt å tilpasse og iterere produktet, og dermed optimalisere bruken av tid og menneskelig kapital. No-code muliggjør at ikke-tekniske teammedlemmer kan bidra i utviklingen, noe som påvirker ressursoptimalisering og kan forbedre organisasjonens menneskelige kapital, med følgende resultat at man kommer raskere og tettere på kunden.

Studien gir forskere og utøvere verdifull innsikt i effektene av no-code teknologi på konkurransedynamikken og ressursallokering. Den oppfordrer til ytterligere forskning for å undersøke anvendeligheten av no-code teknologi i en bredere variasjon av kontekster og dens langsiktige effekter på industrielle strukturer. Denne studien gir en grunnleggende forståelse av no-code teknologi og dens potensial til å utfordre tradisjonelle programvareutviklingsmetoder og forretningskonkurranse.

Acknowledgement

This master's thesis was written at the Department of Industrial Economics and Technology Management (IØT) at the Norwegian University of Science and Technology (NTNU). For the authors, this marks the end of a two-year master's degree at the NTNU School of Entrepreneurship (NSE). The study was carried out from January to June 2023. Preparations for the thesis were conducted in the courses TIØ4530 and TIØ4535 during the autumn of 2022, in the form of a literature review (Balcon, Bekke, & Kaland, 2022). The deep dive into resource theory, competitive advantage, and digitization gave us rewarding insights, and we quickly felt powerless in the face of a preponderance of concepts, frameworks, initiatives, and standards. The goal of this thesis was to explore the potential of leveraging no-code technology alongside other company resources to achieve a competitive advantage.

We would like to express our profound gratitude to Professor Øystein Widding at IØT for excellent guidance, invaluable sparring, and, for the most part, unwavering faith. A big thank you is also directed towards SINTEF, with Geir Kjetil Hanssen, for their contributions and input on the methodological approach and data gathering.

We would also like to thank the five case firms that set aside time for interviews and shared their perspectives, experiences, and expertise. Their contribution to the research has been crucial to the result.

Sebastian Kaland, Scott Aleksander Bekke, Marcus Balcon

June 11, 2023

Acronyms

CA - Competitive advantage

ESD - Engaged scholarship design

IT - Information technology

IØT - Department of Industrial Economics and Technology Management

LCD - Low-code development

MVP - Minimum viable product

NCD - No-code development

NCDP - No-code development platform

NTNU - Norwegian University of Science and Technology

NSE - NTNU School of Entrepreneurship

RBT - Resource-based theory

RPA - Robotic process automation

RQ - Research question

SCA - Sustained competitive advantage

1

INTRODUCTION

No-code technology has become increasingly popular in recent years as a means of streamlining and simplifying the software development process with simple drag-and-drop solutions. The technology provides a layer of abstraction over code, allowing citizen developers to create cutting-edge software and websites without the need for advanced programming skills. In this manner, no-code has the potential to impact the business model of firms across industries as it offers its users agility, reduced development costs, a shorter time to market, and increased efficiency. However, there are shortcomings in the existing no-code literature regarding its social and organizational impact (Käss, Strahringer, & Westner, 2022), such as how the technology can be incorporated with other firm resources to create a lasting competitive advantage. Helfat et al. (2023) also call for more work regarding the impact of digitization on resource literature, especially its effect on the strategies firms pursue and the resources and capabilities they use to compete. With this study, we aim to contribute by offering suggestions to firms on how to leverage no-code technology strategically in terms of its effect on a firm's resources and competitive advantage.

This introductory chapter follows the structured research design of Lars Mathiassen (2017), *Engaged Scholarship Design* (ESD), which seeks to bridge the theory-practice gap. The reasoning behind the adoption of this design method is the fact that the research field is relatively new and intertwined with real-world problems. By implementing the ESD we wish to make the research more transparent and duplicatable, because important decisions and areas of focus become more illuminated.

1.1 ENGAGED SCHOLARSHIP DESIGN

Engaged scholarship is a form of research that aims to address real-world problems (P) by drawing on the perspectives of key stakeholders and devel-

oping knowledge that can help solve these issues (Mathiassen, 2017). A key component of this design is the identification of an area of concern (A) in the literature that relates to the real-world problem (P). This requires a review of the literature to identify gaps or problematic assumptions that can be addressed through engagement with the problem.

The conceptual framing of the argument (F) is crucial in guiding data collection and analysis, serving as the foundation for answering the research questions (RQ) and ultimately developing a contribution (C) (Mathiassen, 2017). Researchers have several options for framing data collection and analysis, including grounded approaches, relying on concepts from the literature, or relying on independent concepts. The chosen analytical approach must allow for leveraging the available data to develop findings that make a contribution (C) to the identified area of concern in the literature (C_A) and the real-world problem (C_P).

A range of research methods (M) are available for engaged scholarship, and the challenge is to select a specific method that can draw on available data to answer the research questions (RQ). Ultimately, the quality and solidity of the contribution (C) component determine whether a study is defendable and can be accepted for publication. All engaged scholarship efforts should involve a contribution to the real-world problem (C_P) and to the area of concern in the literature (C_A). A visual illustration of the design is presented in Figure 1.

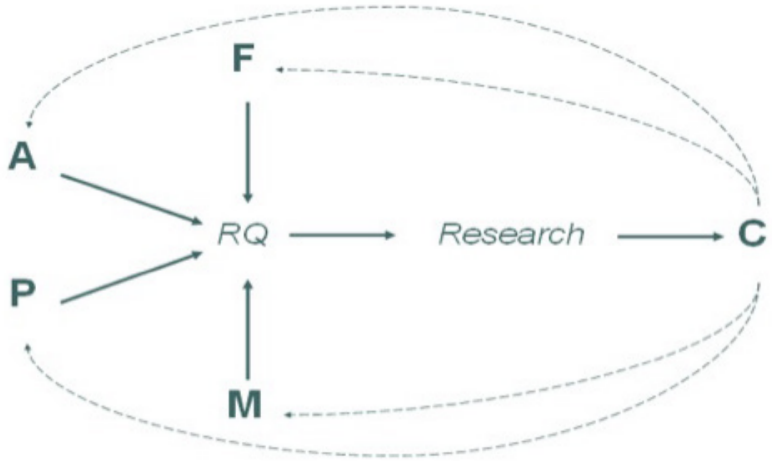


Figure 1: The Engaged scholarship design
(Mathiassen, 2017, p. 19)

1.1.1 *Real-world problem (P)*

The authors are students at NTNU School of Entrepreneurship with backgrounds in economics, media science, psychology, and video production. The master's program is a venture creation program where students start their own businesses alongside the courses, exposing them to the possibility of developing their software solutions using no-code platforms. Due to prior knowledge and experience with Appfarm and Shopify, the authors have gained some understanding of how to use no-code platforms and the potential advantages they may offer.

During a workshop session on the topic, as well as communication with other startups and established firms, the authors discovered a problem related to the evaluation of no-code technology. Even though many firms have adopted no-code technology, there is still a lot of uncertainty regarding how to actually evaluate and make use of it. The discovered real-world problem (P) stoked an interest in the ramifications of no-code technologies for entrepreneurs in a competitive environment and posed as the impetus for the project thesis conducted in the autumn of 2022.

1.1.2 *Literature review and identified gap in the literature (A)*

An understanding of the no-code technology, how the application areas were covered in the literature, and the distribution of publications within designated themes were gained from the literature review (Balcon et al., 2022) conducted prior to this study. The review showed an increase in publications regarding no-code technology from 26 publications in 2017 to 188 publications in 2022, amounting to a 54% average growth rate per year during the period. However, the available literature was technology-focused, addressing its application areas, and little to no research had been conducted on the cross-section with entrepreneurship literature. Searches on the Scopus database for the terms "no-code" OR "low-code" yielded 1086 results, whereas the combination with "competitive advantage" only yielded 2. When the initial searches yielded unsatisfactory results, we continued by searching for related topics to gain an understanding of the concepts influencing both no-code and resource-based theory. We examined research regarding dynamic capabilities, digitization/digital transformation, entrepreneurial orientation, strategic entrepreneurship/strategic management, the lean-startup method, and software

development processes (Balcon et al., 2022).

The work of Käss et al. (2022) also calls for more research regarding the social and organizational impact of no-code technology. Rosin et al. (2020), who looked into the adoption of novel technology, found increases in efficiency for startup firms when implemented correctly. Unsuccessful attempts, however, might lead to the waste of resources, and as a result, they call for more research on how firms can integrate novel technology more strategically.

Furthermore, the literature review showed that the term "competitive advantage" is often used to describe the attributes of no-code technology without actual research backing the claim. To our knowledge, concrete research using case studies is still lacking in order to validate if the adoption of no-code can in fact be linked to an early stage firm's competitiveness. One previous attempt was discovered in "*Strategic Use of Low Code Platforms*" by Cuthbert and Pearse (2021), analyzing whether the adoption of no-code platforms could indeed bring value for firms in terms of increased revenue, proving the value of the technology. However, their research did not fully employ resource-based theory (RBT), and therefore, it did not truly uncover *how* the technology creates value or *if* it can contribute to a lasting competitive advantage.

Therefore, the discussion part of the literature review focused on applying the RBT framework to analyze the potential for sustained competitive advantage (SCA) for no-code as a resource. Our results concluded that no-code is, in fact, value-creating, but only as a part of a resource bundle. Because no-code development platforms are off-the-shelf products homogeneously distributed, the advantage relies on *how* the technology interacts with other firm resources, further strengthening the need for a more comprehensive study on the topic.

Finding that there was a gap in research (A) in this particular area of study, the potential to link no-code technology to resources and competitive advantage formed the motivation for this master's thesis.

1.1.3 *Purpose of the study (F)*

To fully exploit the opportunities proposed by utilizing no-code, one needs to understand how it can pose a strategic advantage. As stated above, the literature still lacks concrete case studies that examine the construction of complementing

resource bundles and no-codes effects on firm resources. We will look into the reasoning for selecting no-code and the value added by the technology in various early stage software firms that are developing internal or external software using no-code development platforms. The following purpose is derived from the uncovered gap in the literature:

"To investigate if no-code, in combination with other resources, can contribute to a firm's competitiveness"

The configuration of the firm's resources, how they are used, and whether or not they serve as a foundation for competitive advantage are considered factors in determining the firm's competitiveness in a market. We will look into the decision-making process of early stage firms using no-code technology to address a market need or solve a problem, as well as the value the technology delivers to the firm. To further emphasize the value no-code adds, we will also examine each case firm and look for commonalities, with the goal of uncovering factors that make no-code a competitive resource. The employed framework (F) is resource-based theory (RBT).

1.1.4 *Research questions (RQ)*

In order to provide an answer to the purpose of the study, we first need to understand how no-code functions as a resource. As a theoretical foundation, we use the definition provided by Barney (1991, p. 3), which states that a resource is "*all assets, capabilities, organizational processes, firm attributes, information, knowledge, etc. controlled by a firm that enable the firm to conceive of and implement strategies that improve its efficiency and effectiveness*". Going forward, the purpose of the study has been divided into two research questions (RQs):

1. *How does no-code technology interact with a firm's resources to create a competitive advantage?*
2. *What are the key considerations and evaluation criteria that early stage firms should employ when considering to adopt no-code technology?*

The first RQ investigates how the technology itself interacts with other firm resources to make it valuable and competitive. The value of no-code as a resource might therefore depend on various resource configurations for various firms, and no-code can also improve competitiveness in different ways. The second RQ seeks to explore the underlying processes of choosing no-code technology. By answering this question, we will examine important aspects of the technology as seen by the case firms. For what purpose is the technology relevant to employ and what problem does it solve, as well as the common traits of no-code as a resource. This will be investigated in order to gain a deeper understanding of how firms should evaluate the technology prior to adoption and how they view its abilities as a source of competitive advantage. The case interviews with different early stage firms using no-code will investigate both research questions.

1.1.5 *Method (M)*

The chosen research method (M) for this study is a qualitative, multiple case-study approach in order to examine the no-code phenomenon. Five case firms were examined using semi-structured interviews. The preparations for the interviews consisted of creating selection criteria and a subsequent interview guide based on the research questions (RQ) and the literature on no-code and resource theory. We adopted an analytical approach and tried to find commonalities in the data. The method will be elaborated on in great detail in chapter 4.

1.1.6 *Contribution (C)*

With the data gathered, we will be better able to understand the potential competitiveness of this novel technology and the variables that should influence firms' decisions to adopt and make use of it (C_P). Furthermore, we wish for this contribution to involve a better decision-making basis for firms seeking to take advantage of the technology by providing a framework for evaluation of the firm's resources in relation to no-code (C_A). This will make it easier to assess the requirements or potential effects of employing no-code technology for a particular purpose in the future. Figure 2 demonstrates a visualization of our final contribution to the literature (C_A). The boxes in the dotted lines show the literature coverage to date as discovered in the literature review (Balcon et al., 2022), with an in-depth evaluation of the technology. Our contribution

investigates (1) the process of choosing no-code and (4) the effects of no-code on firm resources, (5) which could lead to increased competitiveness.

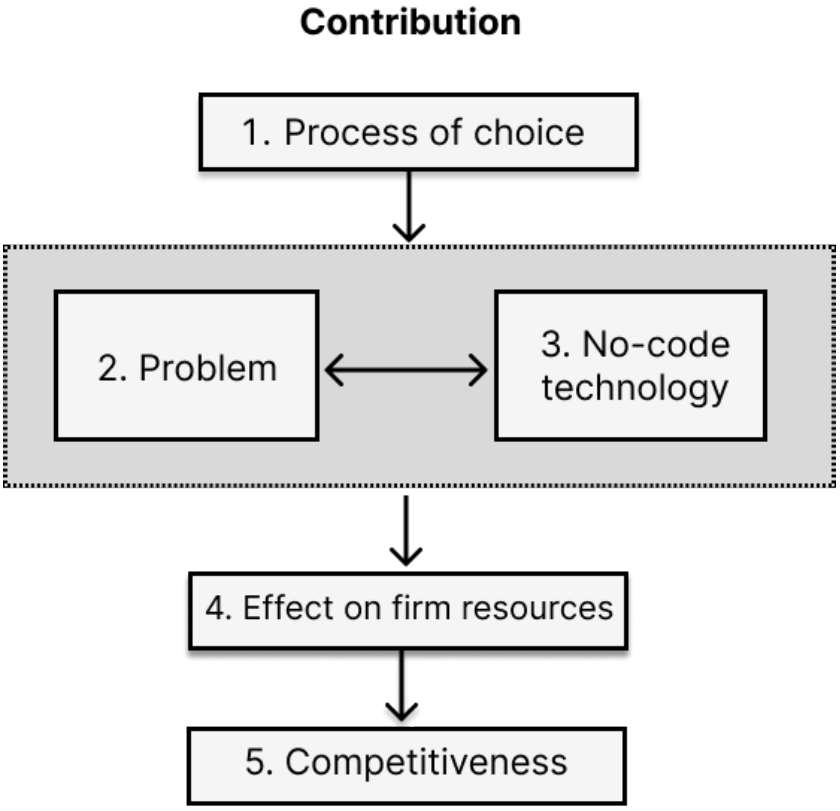


Figure 2: Contribution to the literature (C_A)

1.2 SUMMARY OF THE ENGAGED SCHOLARSHIP DESIGN FOR OUR STUDY

Table 1 gives an overview of Engaged Scholarship Design, its components, and definitions by Mathiassen (2017) and how they apply to our study.

Component	**Definition** (Mathiassen, 2017, p. 20)	**Our study**
P	The problem setting represents people's concerns in a real-world problematic situation.	Companies are currently unable to strategically evaluate their resources in relation to the adoption of no-code technology.
A	The area of concern represents some body of knowledge in the literature that relates to P.	We have examined literature on dynamic capabilities, digitization/digital transformation, entrepreneurial orientation, strategic entrepreneurship/strategic management, the lean-startup method, and software development processes.
F	The conceptual framing helps structure collection and analyses of data from P to answer RQ; F_A draws on concepts from A, whereas F_I draws on concepts independent of A.	We adopt the resource-based theory (Barney, 1995).
M	The method details the approach to empirical inquiry, specifically to data collection and analysis.	Qualitative, multiple-case study approach using semi-structured interviews.
RQ	The research question relates to P, opens for research into A, and helps ensure the research design is coherent and consistent.	RQ1: *How does no-code technology interact with a firm's resources to create a competitive advantage?* RQ2: *What are the key considerations and evaluation criteria that early stage firms should employ when considering to adopt no-code technology?*
C	Contributions influence P and A, and possibly also F and M.	C_P: Lessons for how early stage firms should evaluate the suitability of no-code, and successfully integrate the technology with their resource base. C_A: A detailed empirical account for how no-code interacts with firm resources as part of a resource bundle, including a new framework for evaluating the technology in light of firm-specific cases.

Table 1: Engaged scholarship design applied

1.3 STRUCTURE OF THE MASTER´S THESIS

The introductory chapter has explained the ESD of the study and its contribution to the real-world problem, the literature on competitive advantage, and no-code. Proceeding, the structure of the study will be as follows: chapter 2 presents a deeper breakdown of no-code as a concept and its opportunities and challenges. In chapter 3, we proceed by outlining the theories of resources and competitive advantage, from which the theoretical framework will be derived. The methodological approach deployed in order to answer the research questions will be described in great detail in chapter 4. We have chosen a qualitative approach consisting of semi-structured interviews and multiple case studies as the research design. chapter 5 contains the analysis and results of the study, while chapter 6 follows with a discussion of key findings and the contribution to the literature. chapter 7 presents the conclusion, and chapter 8 offers recommendations for further research.

2

NO-CODE AS A CONCEPT

The history of software development dates back to Alan Turing in 1936, and in the 87 years since then, the world of software development has undergone significant change. Information technology (IT) is now taught in a variety of academic disciplines, and while the number of individuals with sophisticated programming skills is on the rise, there is still a surplus of demand for these skills (Rosin et al., 2020). In response, modern actors have developed instruments to expedite and enhance the software development process. These technologies fall under the "no-code" umbrella term (Balcon et al., 2022).

No-code technology creates a layer of abstraction over code that transforms coding principles into simple drag-and-drop solutions. This enables developers to create cutting-edge software and websites graphically without the need to know programming languages. No-code technology is a software development methodology that enables non-programming staff members, known as citizen developers, to simply create software or add functionality to existing websites or solutions. No-code development platforms (NCDPs) and no-code tools are intended to be more efficient and cost-effective than conventional software development. Some NCDPs even claim to be 10 times more efficient than traditional software development (Luo, Liang, Wang, Shahin, & Zhan, 2021). The developer visually constructs relations while the NCDP structures the underlying code. This visual approach makes working with databases simpler than ever before (Balcon et al., 2022).

No-code is comparable to creating a structure with prefabricated construction blocks. In contrast to conventional construction methods, construction time is drastically reduced because the building materials are already constructed and entire architectural elements can be rapidly assembled. Where you once had to construct a wall plank by plank, the provider of the prefabricated elements has already completed the task, leaving you only to affix them to the

building's foundation. Similarly, no-code represents prefabricated building elements, allowing software developers to construct application code container by container rather than code line by code line (Balcon et al., 2022).

2.1 THE DISTINCTION BETWEEN LOW-CODE AND NO-CODE

Although no-code has only recently emerged as a prominent catchphrase, low-code has likely been used to describe this sequence the longest. According to an article published by the IEEE Computer Society, the primary difference between low-code and no-code software development is the quantity of knowledge and physical written code required. Unlike no-code environments, low-code environments are said to require programming expertise. This does not mean that custom code cannot be used in no-code environments; it is simply designed so that it is not required (Hurlburt, 2021).

In the academic and business communities, low-code and no-code are used synonymously. As was previously stated, "*no-code*" has recently gained popularity, and it is evident that business actors are shifting from "*low-code*" to "*no-code*". For example, the Norwegian company Genus, which marketed itself as a provider of low-code applications until recently, now refers to itself as a provider of no-code applications (Genus, n.d.). The association with "*low-code*" misrepresents the capabilities of the technology as low quality code and is one of the reasons for the transition. No-code development (NCD) is a closely related concept that differs from low-code development (LCD) in that it precludes hand-coding. Academics are divided over whether NCD is a distinct concept or a subset of LCD (Käss et al., 2022).

The distinction between low-code and no-code is frequently based on usage and may be a matter of perspective. For instance, Microsoft's Excel transitions from no-code to low-code when macros are used, but one could argue that Excel should not be considered no-code due to the use of conditions in formulas (Lethbridge, 2021). No-code and low-code are used interchangeably, both in business and in academia. In recent times, however, no-code has become the most commonly used term in research, so for the purpose of this thesis, we will consolidate both terms and use "*no-code*" as an umbrella term.

2.2 OPPORTUNITIES AND CHALLENGES OF NO-CODE

2.2.1 *The perceived advantages of no-code*

One of the perceived advantages of no-code according to the literature, is the faster build time and rapid development. This is allowed through the employment of pre-built, drag-and-drop components that aim to avoid laborious coding. With this, firms may swiftly prototype, iterate, and deploy applications that would take a long time and a lot of resources to develop traditionally (Rokis & Kirikova, 2022). Updates and adjustments can be made quickly in response to changing user desires or market dynamics due to this acceleration in development speed (Gomes & Brito, 2022; Rokis & Kirikova, 2022).

Another perceived advantage is the flexibility of the NCDP. NCDP enables non-technical users to develop and customize software. This democratization of development can foster creativity and innovation within a company (Hurlburt, 2021). No-code platforms are even more appealing when financial issues are considered. Traditional development costs a lot, including hiring competent programmers and sometimes protracted development cycles. No-code development drastically lowers these costs, as non-programmers can develop powerful software (Käss et al., 2022). Faster development means products reach the market faster, boosting the return on investment and possibly grabbing market share before competitors can react (Hurlburt, 2021; Rokis & Kirikova, 2022).

Finally, while digital security is always a worry, many no-code platforms are developed with strong security safeguards. As platform providers update security measures to address new threats, this built-in protection can reduce data breach and cyberattack concerns (Wang, Feng, Zhang, & Sun, 2021; Hurlburt, 2021; Gomes & Brito, 2022).

No-code literature has identified several benefits, including shorter build times, flexibility, cost savings, and built-in security. These qualities can make it a popular choice for firms looking to improve their digital capabilities efficiently and effectively.

2.2.2 *Challenges with no-code*

When evaluating no-code for software development, it is essential to be aware of the inherent difficulties. The paucity of customization options is one major issue. No-code platforms rely on pre-built templates and modules within the framework of the NCDP, which can hinder the ability to develop unique, complex applications with customized features (Al Alamin et al., 2021; Käss et al., 2022).

Scalability is also a significant concern. Initially, no-code platforms may support the development and deployment of applications, but as an application grows in size and complexity, they may struggle (Lethbridge, 2021; Rokis & Kirikova, 2022). Additionally, there is a notable dependence on platform providers. Businesses rely on these platforms for ongoing maintenance, updates, and new features. This dependency can expose businesses to substantial risk if the platform provider discontinues certain features, fails to keep up with industry developments, or ceases operations. This is also extremely pertinent with regards to vendor lock-in. Since the majority of the application is composed of pre-built blocks, there is little to no portion of the application that can be extracted if you wish to transfer providers. If so, you would have to reconstruct everything from scratch (Luo et al., 2021; Käss et al., 2022; Gomes & Brito, 2022).

Risks related to data security and compliance are additional obstacles of significance. While some no-code platforms may offer security features, precise adherence to security practices and compliance with applicable regulations must still be ensured. It is essential to remember that not all no-code providers meet industry-specific compliance standards. Since this is a relatively new emerging technology, there is a lack of general knowledge regarding keeping no-code applications secure. There may also be performance restrictions. No-code platforms may not be as efficient as custom-coded platforms, especially when managing large amounts of data or performing complex operations. The outcome may be applications that perform below expectations or require additional resources to function properly (Gomes & Brito, 2022; Käss et al., 2022).

We observe that no-code technology is perceived to have a number of positive characteristics. However, we also uncover that there are aspects of the technology that, over time, could create obstacles and restrictions.

3

THEORETICAL FOUNDATION

The theoretical framework used for data gathering and analysis of the findings in chapter 5 is laid forth in the chapter hereafter. As described in subsection 1.1.2, the literature review was conducted on several thematics related to RBT and no-code in order to gain an in-depth knowledge of the surrounding literature. Because the purpose of the study is to research the gap between resource-based theory and no-code technology, this will be the main source of theories used to derive this study's framework. Literature from the other thematic areas will be used as a complement, where appropriate. The framework of choice illustrates how resources and their combinations can enhance a firm's competitiveness. At the end of the chapter, we will use the framework to evaluate the technology after discussing it in a no-code context.

3.1 THEORETICAL FRAMEWORK

Research on competitive advantage has a long rooted history in the field of strategic management. Early theories provided by Porter (1980; 1985) explain how a firm achieves competitiveness by implementing strategies that exploit internal strengths and neutralize external threats. The framework thus explained competitive advantage in light of environmental conditions and relied on a set of assumptions that, in later works by Barney (Barney, 1991), were viewed as unfit to explain sources of competitive advantage. By excluding resource heterogeneity and immobility, Porter's (1980) framework viewed firms in a certain industry as having the exact same strategic resources, effectively eliminating the possibility of sustained competitive advantage (Wernerfelt, 1989; Barney, 1991). RBT develops the literature by substituting these assumptions for the following:

1. The model assumes that firms within an industry may possess heterogeneous strategic resources.

2. The model assumes that these resources may not be perfectly mobile across firms and, consequently, that heterogeneity can be long-lasting.

By allowing for heterogeneity and immobility, the RBT laid out by Barney (1991) explains competitive advantage in terms of how firms are able to perceive external opportunities and respond by constructing unique internal resource bundles. Alvarez and Busenitz (2001) further explain that different agents have different beliefs about the relative value of resources and that opportunities might, as a result, be conceived differently among firms.

3.1.1 *What is a resource*

In this study, we follow the definition of resources provided by Barney (1991, p. 3): "*All assets, capabilities, organizational processes, firm attributes, information, knowledge, etc. controlled by a firm that enable the firm to conceive of and implement strategies that improve its efficiency and effectiveness*". Building on this definition, we further categorize resources into the following three classes:

1. Physical capital
2. Human capital
3. Organizational capital

Additionally, the terms tangible and intangible are used to describe the different types of resources. Tangible assets refer to physical and material resources such as machinery, office space, inventory, and land (Reed, 2005). Intangible assets, on the other hand, refer to resources that have no physical form and are divided into assets or competences (Hall, 1993). Assets typically include intellectual property (trademarks, patents, and trade secrets) or firm contracts, which are included on the balance sheet. Competencies are the skills and know-how of employees, partnerships, or firm culture. Barney (1991), Dierickx & Cool (1989) and Schriber & Löwstedt (2015) argue that intangible assets are more likely to be a source of SCA because it is often difficult for competitors to imitate. Our analysis will focus on the combination of different firm resources in relation to no-code technology.

3.1.2 *Conditions for competitive advantage*

So far, we have discussed what constitutes a firm resource and explained how firms must act upon external market opportunities by creating unique strategies. In order to evaluate the competitiveness of a firm's resources, the VRIO-framework derived from RBT is widely recognized in the strategic management literature. The theory assesses the attributes that must exist to generate SCA, while the environmental models assist in isolating the firm attributes that can become resources. According to the framework, a resource must be (1) *valuable*, (2) *rare*, (3) *inimitable*, and (4) *organized* in order to become a source of SCA (Barney, 1995).

(1) *Valuable:*

For a resource to be deemed valuable, the subsequent firm strategy must enable the firm to achieve effectiveness and efficiency (Barney, 1991). Massey (2016) considered resources valuable if they produced, or helped produce, a significant positive effect on a firm's performance. Peteraf and Barney (2003) contend that this might be related to, but not limited to, direct economic costs or brand image among customers. Digital tools can be valuable resources, with Enterprise Resource Planning (ERP) systems in the early 1990s as a prime example. The introduction of ERP systems was found to reduce the operation costs of the companies, thus giving them an operational advantage over those that did not implement them and making them valuable (Beard & Sumner, 2004). Furthermore, Barney (1995) argues that a resource that is not value-creating when implemented is considered a competitive disadvantage.

(2) *Rare:*

While identifying value-creating strategies, organizations must analyze the rarity of the resource in question. If the resource is widely accessible to other businesses, they will be able to apply the same strategy, essentially eliminating the possibility of a competitive advantage altogether (Barney, 1991). The same holds true when bundling a number of valuable resources. A strategy cannot become a source of SCA if all the resources are accessible to rival firms. This, however, does not mean that common resources should not be acquired and implemented. Common resources could in fact guarantee a firm's survival when used to achieve competitive parity, which is said to be when a firm embraces a valuable but widely accessible resource to prevent rival firms from

getting a competitive advantage from the same resource (Tohãnean, Buzatu, Baba, & Georgescu, 2020; Buzatu, Dinu, Costache, & Tohănean, 2020). Taking it even further, Barney (1991) argues that a common resource can lead to a competitive advantage as long as the number of firms adopting it is smaller than the number of firms required to achieve perfect competition dynamics in the market. Valuable, but common resources could potentially provide a situation described as a first mover advantage.

(3) *Inimitable:*

So far, a resource could become a source of first-mover advantage if the resource in question is valuable and rare. The third requirement of the VRIO-model suggests that a resource cannot be a source of SCA if competitors can obtain it. In order for a resource to be perfectly inimitable, it must fulfill one, or a combination of, three requirements: (1) a firm's ability to acquire a resource is dependent on a specific historical factor; (2) the relationship between the resources a firm has and its SCA is causally ambiguous; or (3) the resource that gives a firm its advantage is socially complex (Barney, 1991). For instance, a historically dependent resource could be a military contract tied to a specific war, making it especially hard to imitate. Causal ambiguity refers to a resource or combination of resources, and subsequent SCA is poorly understood by its competitors, thus being hard to imitate (Lippman & Rumelt, 1982). Lastly, if a resource is socially complex, it is tied to specific social phenomena, such as the overall culture of the firm or relationships with partners and suppliers.

(4) *Organized:*

A specific resource cannot, in and of itself, create a competitive advantage, even if it is valuable, rare, and inimitable. The last requirement for SCA is therefore dependent on the firm's ability to effectively organize the resource in its operations. To do so, the organization's various components—including its management control systems, reporting structures, systems for strategic planning and budgeting, and logistics network—must be effectively integrated and organized. Without the appropriate organizational structure around the resource, a firm will not be able to successfully acquire, use, and monitor it, regardless of how valuable, rare, or inimitable it may be (Barney, 1995). This implies that a business cannot create SCA without an effective organization of its resources.

3.1.3 *Summary of the VRIO-model*

Figure 3 illustrates the relationship between the four evaluation criteria and the resulting degree of advantage using the VRIO-model. If exploited, resources that cannot be perceived as valuable will put a company at a competitive disadvantage. Using the resource may result in competitive parity if it is valuable, widely available, and adopted by a large number of competitors. If a resource is valuable and rare but also easily imitated by rivals, it only serves as a temporary competitive advantage. Resources that are valuable, uncommon, and inimitable have the potential to be sources of SCA, but only if the business is properly organized to make use of them.

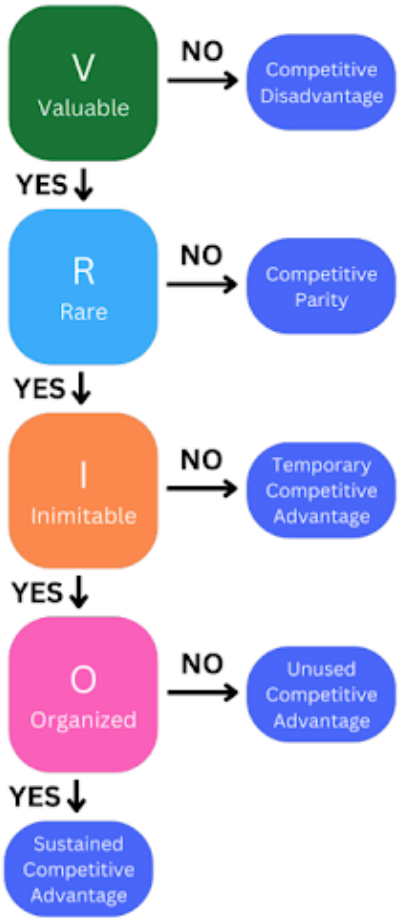


Figure 3: VRIO-model (Barney, 1995)

3.1.4 *Combining resources*

Even though the RBT stresses the need for a firm to acquire distinctive resources, several scholars have suggested that one key resource is not sufficient to create a lasting competitive advantage (Allred, Fawcett, Wallin, & Magnan, 2011). Wernerfelt (1984) points out that substitute resources can depreciate the returns a given resource provides and that firms need a strategy to protect themselves from such depreciations. Eisenhart and Martin (2000) argue that the way in which a firm maximizes its competitive potential is linked to how it develops and configures its resources. Combining resources can thus enhance diversification from other firms possessing the same competitive resource or substitutes for it (Wernerfelt, 1984). Hall (1993) argues that employee know-how is ranked as one of the most important contributing factors to the success of the firm and is backed up by the research of Prahalad and Hamel (1990). Miller and Shamise (1996) also found that human capital has indirect effects on firm performance through interactions with the overall firm strategy. They also contend that human capital initially is less valuable than the costs of acquiring it, but through competency growth, it eventually surpasses the costs, bringing additional value to the firm. To strengthen the advantage giving properties of a specific resource, the firm should combine it with employee know-how.

Teece (1997) contributed to the research field by critiquing the RBT as being static in context. He argues that contexts change, and so the RBT needs to take such environmental disruptions into account. His framework of dynamic capabilities therefore suggests that firms continually need to evaluate their resources to address such changes, by reconstructing their resources into new bundles, which leads to capabilities. The need for combining several resources, and the reconfiguration according to dynamic market changes, ensures SCA over time (Teece, 2014). Digital transformation has increased the speed at which markets change, and so, the need for firms to undergo rapid reconstruction of their resource bundles to create new dynamic capabilities is ever more present (Teece, 2023).

3.2 THEORETICAL FRAMEWORK IN RELATION TO NO-CODE

The subsequent part of this chapter will apply RBT and the VRIO-framework to assess no-code technology as a resource. We assume resource heterogeneity and immobility across firms.

3.2.1 *No-code as a resource*

Firstly, we will examine if no-code technology is a tangible or intangible asset to a firm. Because no-code is purely based on code, it is regarded as a software solution. On the other hand, software, which is an integral part of hardware, can be considered tangible. To answer the question, we have to examine the relation between no-code and the accompanying hardware that is used to utilize it. Because no-code is dependent on the use of computers, hard drives, and servers, one can be quick to assert its tangible properties. This, however, is true for all software solutions and cannot be regarded as a generalization towards tangible assets. It is merely a digital resource upon which you can create industry software applications for firm hardware solutions. We therefore argue that no-code is in fact an intangible asset, but the software solutions created using it might be tangible if integrated into industry hardware. According to Armstrong and Shimizu (2007), intangible resources are typically more strategic than material ones because they are harder to imitate. Going forward, we regard no-code as intangible.

3.2.2 *No-code as a valuable resource*

As described in subsection 3.1.2, a resource that increases the effectiveness or efficiency of a company is deemed valuable. No-code technology enables businesses to develop software fast and easily with little to no manual coding, allowing non-programmers to take part in the process (Käss et al., 2022; Rokis & Kirikova, 2022). By fusing business with technology, this might potentially save time, money, and the necessary human capital while also raising the quality of software products (Bock & Frank, 2021). Firms can gain a lot from the ability to rapidly translate business requirements into applications and make changes without requiring a lot of manual coding. In turn, this can lead to a faster time to market with fewer resource requirements. Thus, it might seem like no-code technology is, in fact, a valuable resource. However, no-code development platforms are widely available as an off-the-shelf service and are therefore not to be considered a source of SCA by themselves (Balcon et al., 2022).

3.2.3 *Combining firm resources and no-code*

No-code technology as a software service cannot contribute to competitive advantage as a stand-alone resource because it is dependent on human capital and tangible resources, such as computers, to be utilized. Furthermore, the availability of NCDPs makes the resource common, which further strengthens the need for other resources to make it competitive. In order for a firm to take advantage of no-code, a combination of resources complementing it is necessary. So far, we have established no-code technology as an intangible resource in a firm's portfolio. As previously mentioned, the no-code literature is vague at best when it comes to the actual empiricism of how the technology, as a resource, contributes to a firm's competitiveness. The need for more research on the missing link between no-code technology and competitive advantage is therefore apparent.

3.3 THEORETICAL FRAMEWORK APPLIED

We decided to approach no-code technology as a resource and developed the following theoretical framework, employing the RBT in a no-code context. The structure, which comprises four levels, was derived from pertinent literature about resources and competitive advantages described in this chapter:

- First layer (Organizational): Describes the competitive landscape and market positioning of the case firms.
- Second layer (Resources): "*all assets, capabilities, organizational processes, firm attributes, information, knowledge, etc. controlled by a firm that enable the firm to conceive of and implement strategies that improve its efficiency and effectiveness*" (Barney, 1991, p. 3).
- Third layer (No-code): How the firm uses no-code to build software for internal or external use as part of their business model.
- Fourth layer (Combining resources): involves the combination of firm resources, how the firm perceives no-code in relation to these resources, and the effect on competitive advantage.

Figure 4 is a visual representation of the applied framework used to analyze *how no-code technology interacts with firm resources and contributes to the competitiveness of a firm*. In order to answer the question, we examine the existing resources of

the case firms and how no-code as a resource interacts with them as part of a bundle. The overlapping area in the figure shows the combination of resources that contributes to no-code being a competitive resource.

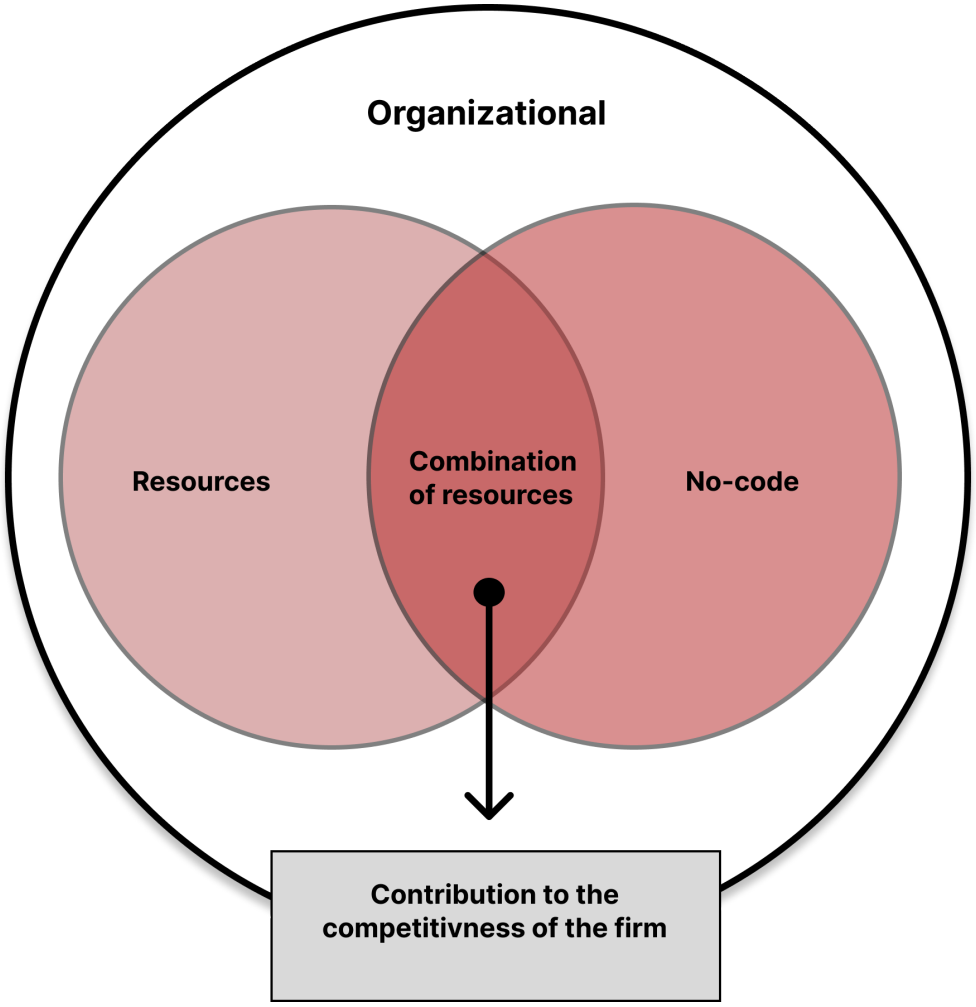


Figure 4: Resource-based theory in a no-code context

4

RESEARCH METHOD

This chapter presents the methodological approach used to answer the research questions and the study's overall goal. Initially, the choice of research design and the underlying rationale for choosing said design are introduced. Followingly, the method for data collection and analysis is presented before finally reflecting on methodological limitations and the method's suitability to contribute to the achievement of the study's main objectives.

4.1 RESEARCH DESIGN

When the primary objective of the research is to define a detailed description of a particular phenomenon, qualitative methods are appropriate for studying fewer cases (Eisenhardt, 1989; Flick, 2015). According to Yin (2018), the use of case studies as methodological tools is appropriate if one has questions like *how* and *why*. This technique is deemed appropriate in light of the study's overall goal to identify "*if no-code, in combination with other resources, can contribute to a firm's competitiveness*", as well as the fact that it enables us to investigate cause and effect in a smaller sample (Krumsvik, 2014). Due to the relative novelty of the research field of no-code in an entrepreneurial context, its maturity is low, and the use of a case study was deemed necessary in order to obtain sufficient depth in the data.

With a multiple-case study, the evidence is oftentimes regarded as more compelling and the study to be more robust than singular case studies (Yin, 2018). Because we wanted to explore how no-code technology may impact businesses' ability to compete with other market players, multiple cases were deemed necessary in order to find compelling evidence. Eisenhardt (1989), who contends that there is no agreement on the ideal number of respondents in such studies, argues that the number of interview subjects is frequently sufficient in a range

between 5 and 10. The selection criteria outlined in subsection 4.1.1 served as a guide when choosing the case firms.

The process was iterative and divided into multiple steps, serving as a systematic plan for the research conducted (Yin, 2018). An illustration of the research process is provided in Figure 5 and follows an abductive research design. According to Saunders et al. (2019), abductive research design begins with the observation of a 'surprising fact' and then tries to work out a plausible theory of how this would have occurred (Sætre & Van de Ven, 2021). The authors observed a rise in adoption of no-code technology amongst startups in the entrepreneurial ecosystem surrounding NSE. However, through conversations with a number of the startups, we uncovered that they did not know how to evaluate the no-code technology and its effect on competitiveness. The following literature review (Balcon et al., 2022) aimed to find answers in the literature, but uncovered a surprising fact regarding the lack of concrete research on no-code technology and competitive advantage. When a phenomenon is not currently, or poorly, understood by existing literature, it is considered an anomaly (Sætre & Van de Ven, 2021), as is the case with no-code in entrepreneurship literature. The next step in abductive research is to work out a plausible theory of how this phenomenon could have occurred (Saunders et al., 2019; Sætre & Van de Ven, 2021). We therefore adopted the RBT as a theoretical framework and subsequently derived research questions. These functioned as foundations for the interview guide used to obtain data through semi-structured interviews.

The following data analysis tried to identify themes, explain the patterns, and subsequently integrate these explanations into the overall conceptual framework (Saunders et al., 2019), in order to build up a plausible theory of how no-code can contribute to a firm's competitiveness. Abductive research is also a more flexible research approach, oftentimes consisting of a "theory-data-theory" where researchers go back and forth between the two (Saunders et al., 2019). As described in greater detail in section 4.4, during the data analysis, we had to evaluate the research questions, resulting in a change of RQ2, effectively creating a non-linear research process. Widding (2006) also makes a similar claim, arguing that the researcher's maturity and comprehension of the phenomenon under study have an impact on the non-linear nature of the research.

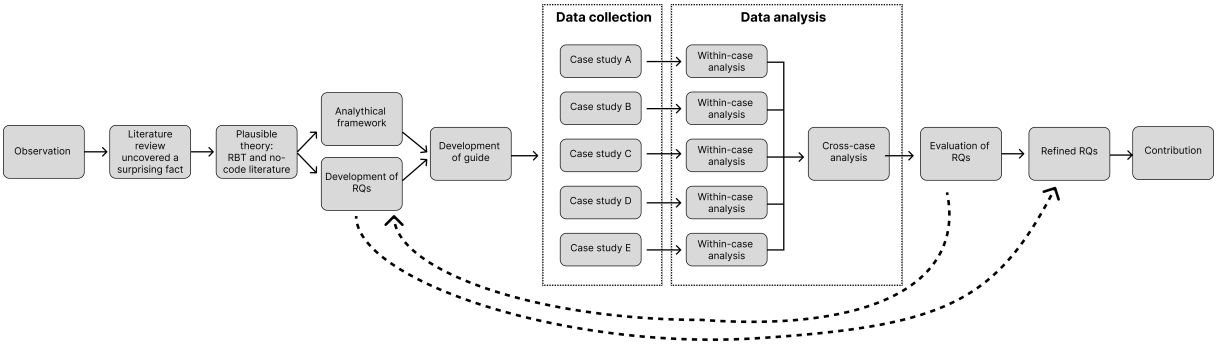


Figure 5: The research process, and the order of analysis which leads to the answer to the RQs

4.1.1 *Choice of interview subjects*

No-code technology encompasses many different solutions, ranging from easy website development platforms to complex industry software development solutions. Thus, the evaluation of relevant case firms was not limited to the no-code development platforms and application areas for which the technology was utilized. However, as the study focuses on the process from entrepreneurial opportunity to market entry, the companies could not have less than twelve months of runtime and needed paying customers. Additionally, we sought to investigate pure software companies, so firms creating hardware with accompanying software were eliminated. Firms utilizing easy website builders such as Wordpress, Webflow, Shopify, Weebly, WIX, Squarespace, or Magento to build static websites for e-commerce stores are excluded because their main product is not the software itself but rather a means to sell hardware products. For the purpose of this study, the firms must build internal or external software as part of their business model.

During a brainstorming session, we determined that the study should be broad in terms of firm sizes to see if there was any correlation between how the technology was utilized and how the different resource pools impacted the results of using no-code. According to Porter (1991), however, the competitiveness of a firm is the sum of all its activities. Large corporations may have activities in different countries, branches, and even subsidiaries with vast amounts of resources. This complexity makes it hard to understand cause and effect because many interrelated activities and strategies may be in effect simultaneously. In order for this study to thoroughly examine the effect on

competitive advantage, the interview subjects were therefore limited to startups and scaleups. Furthermore, the interviewees need a thorough understanding of the technology. The selection criteria are as follows:

Selection Criteria	**Argument**
The firm must have a minimum of twelve months runtime.	The firm must have undergone a development process in order to obtain a thorough understanding of the effects on resources.
The firm must have paying customers.	We want to track the process from the entrepreneurial opportunity up to market entry and customer onboarding.
The firm must deliver software applications as their main service offering.	We want to see the effect of no-code on pure software firms, thus excluding firms developing software as an integral part of a hardware solution.
The firm must utilize some kind of no-code tools in the development of either internal or external software services, as part of their business model.	In order for the interview subjects to have a thorough understanding of the implications of no-code, they must utilize some form of no-code tool for development.
The firm must be a startup or scaleup.	Large enterprises have complex firm structures, vast amounts of resources, and simultaneous strategies, which can affect the analysis of competitive advantage. Because cause and effect in such large companies are hard to uncover, the selected firms must be either startup or scaleup.

Table 2: Selection Criteria

These criteria guided the search for potential interview subjects. We wanted to embrace a wide study across several use cases of the technology as well as examine the effect in different markets to look for commonalities or contrasting viewpoints. The initial search started with an examination of potential cases within our network at NSE. Two cases met the selection criteria. Case firm A had used no-code from the start to build the main solution, whereas Case firm B utilized no-code for internal purposes, thus fitting the motive of multiple use

cases. Searches for other potential cases were conducted using an extended network, which yielded two more options. The relation with these firms came as a result of the researchers own startup activity, in which the two case firms were known for their use of no-code. Case firm C, which has a track record of over three decades, recently started to use no-code in their operations, in parallel with traditional development. Case firm E had adopted no-code from the start and had built their business around the technology, thus posing an interesting viewpoint. Lastly, we researched articles about firms using no-code technology in their ventures. As a result, we discovered a firm that had great success using no-code to create a user interface (Case firm D). Even though they are now considered a scaleup, they adopted no-code from the infancy of the firm.

With an initial pool of five interviews with firms in various markets, use cases, sizes, and times of adoption, the authors felt positive that relevant angles would indeed be explored. An unintentional consequence of using our network is the fact that all case firms are situated in Norway, which strictly limits the ability to generalize the findings across other geographic regions. Table 4.2 shows a description of the chosen case firms. To comply with the desire for privacy, the firms are made anonymous but are described by firm attributes.

Case firm	**Case firm A**	**Case firm B**	**Case firm C**	**Case firm D**	**Case firm E**
Size	Startup	Startup	Scaleup	Scaleup	Scaleup
Established	2021	2021	1988	2019	2016
Industry	Fintech	Energy	Consultancy/startup incubator	Energy	IT-consultancy
Interviewee	CEO	CPO	Designer	CTO	Senior manager
Employees	7	8	99	18	130

Table 3: Case Firms

4.2 DATA COLLECTION

In this section, we present the method for data collection used in the study. The main method for data gathering came from five in-depth interviews - two startups and three scaleups.

4.2.1 *Interview guide*

According to Eisenhardt (1989), it is essential to establish an interview guide before conducting case interviews. An interview guide is a set of predefined questions that the interviewer will use to guide the conversation with the interviewee. It helps to ensure that the interviewer asks the same questions to all subjects, making it easier to compare and analyze the collected data. To create an effective interview guide, it's important to have a clear understanding of the research questions you want to answer through the case study (Eisenhardt, 1989). An interview guide should be tailored to your specific research questions as well as the context of the study and the subjects' experiences. Widding (2006) explains that the guide should be derived from the literature to make it easier for the researchers to extract the relevant empirical evidence from the interviews.

The guide was therefore divided into three thematic areas based on the nature of the research questions and the literature that creates the framework of the study: organizational, no-code, resources and competitive advantage. The three thematics were then divided into categories derived from the corresponding literature. In total, six different categories were developed: (1) organizational, (2) no-code, (3) competitive advantage, (4) firm resources, (5) attributes of no-code, and (6) vendor lock-in. A brainstorming session was then conducted to create a list of open-ended questions for each category, which would help gain a deep understanding of the interviewee´s knowledge of no-code in the specific firm context. We also made sure that overlapping questions were included to build bridges between each category. The session resulted in a first draft of the interview guide, which we brought to an expert in the field from SINTEF for feedback. Adjustments were made regarding the final part of the guide, as described further in subsection 4.2.2. The final guide can be seen in Appendix A.

4.2.2 *Execution of case interviews*

Well-structured and conducted interviews are one of the most popular and important methods of gathering data in qualitative research (Major & Savin-Baden, 2010; Yin, 2018). It was determined in accordance with Bryman's (2016) guidelines that qualitative and semi-structured interviews were suitable in order to acquire a better understanding of how no-code technology might affect organizational operations. Qualitative and semi-structured interviews also allow for the continuation of intriguing viewpoints and nuances that the interviewee

has raised (Widding, 2006; Saunders et al., 2019; Bryman, 2016). This enables deeper comprehension and more insight into the research activity. The aim of the case interviews was to elicit in-depth responses from the participants regarding their decision-making process and outcomes related to the adoption of no-code technology, as well as to explore their usage patterns and the role of no-code in their strategic and operational frameworks.

Eisenhardt (1989) explains two advantages of using multiple investigators: (1) it enhances the creative potential of the study with different perspectives adding richness to the data, and (2) it enhances confidence in the findings. Two interviewers therefore conducted each interview, with one following the guide and the other posing follow-up questions to delve deeper into topics of particular interest to the research objectives, or to encourage elaboration on issues raised in the interview guide. The interviewers were conscious of not asking leading follow-up questions, which could harm the validity and trustworthiness of the findings (Yin, 2018).

The interviews ranged from 40 to 55 minutes, as the open-ended questions allowed the interviewees to express themselves freely (Kvale, 1996). All interviews were recorded, enabling the interviewers to pay close attention to the participants' answers while remaining focused on posing thematic follow-up questions. The final section of the guide focused on reflective questions, such as "*Were there any topics you wished to be asked about*?" or "*Is there anything you would like us to do differently in future interviews?*" This ensured that relevant reflections were not overlooked. The recorded interviews also facilitated the transcription process and allowed for the extraction of valuable data by enabling us to focus on the interviewees' responses.

Cases	Case firm A	Case firm B	Case firm C	Case firm D	Case firm E
Role	CEO	CPO	Designer	CTO	Senior manager
Gender	Male	Male	Male	Male	Male
Age	26	26	26	55	34
Interviewee background	Bachelor's degree in innovation and project management. Master degree from NTNU School of Entrepreneurship.	Masters degree from NTNU School of Entrepreneurship, and a masters degree in industrial economics and technology management.	Master degree in interaction design from NTNU. Work experience from several startups and larger firms.	Two years of education in information technology. Been working 8 years in IT-operations in the banking industry, as well as being developer for a software company.	Masters degree in Economics from the Norwegian School of Economics. Worked as a finance consultant for Ernst Young and a software company.
Interviewee knowledge of no-code	Previous knowledge of no-code from case assignments during his master program at NSE. Most of his knowledge comes from his work with the startup.	Extensive knowledge of no-code technology from previous jobs. The experience stretches across various NCDPs and tasks.	Prior knowledge of no-code comes from several website building platforms such as Wix and Webflow, in addition to more complex app development platforms such as Appfarm.	Deep knowledge of the NCDP Appfarm and have been using it since 2019.	Extensive knowledge of NCDPs. Case firm E uses several no-code platforms from the start in 2016.

Table 4: Interviewees in the case study, their background and knowledge of no-code

4.3 DATA ANALYSIS

The section hereafter describes in detail the data analysis conducted in the study. We divide the analysis into within-case and cross-case analyses, in order to obtain a thorough understanding of each case prior to the presentation of the collective findings.

4.3.1 *Within-case analysis*

Because qualitative data often consists primarily of unstructured textual data, it can be difficult to analyze. Additionally, there aren't many guidelines for how such research should be carried out (Yin, 2018; Bryman & Bell, 2015). To structure the data, we adopted Microsoft Excel to make a matrix consisting of the questions from the interview guide and the corresponding answers from each case firm. This resulted in a matrix of all the textual data from the interview transcripts and functioned as our case study database (Yin, 2018). By keeping the full textual data for the within-case analysis, we ensured that no valuable nuances were lost before coding.

In order to grasp and understand the vast amount of data, we started to "*play with the data*" in order to find patterns, insights, or promising concepts (Yin, 2018). As described in subsection 4.2.1, the interview guide was created using elements from the resource- and no-code literature and the research questions as distinct categories for data analysis. However, while playing with the data, we discovered that some answers would fit better into other categories, and a large amount of it would not fit directly into any of the original ones. Especially answers regarding vendor lock-in and attributes of no-code were more fitting to describe the reason for choosing no-code, or the advantages and disadvantages of the technology. In order to improve the internal validity, generalizability, and theoretical level of the research (Eisenhardt, 1989), we used the proposed framework for the study. We did this to understand the data, and compare it to the body of existing literature, and the theory that emerges during research. By working with the data from the ground up, we divided it into relevant keywords from the literature and the research questions. Each keyword became a node for the coding of the data (see Figure 6).

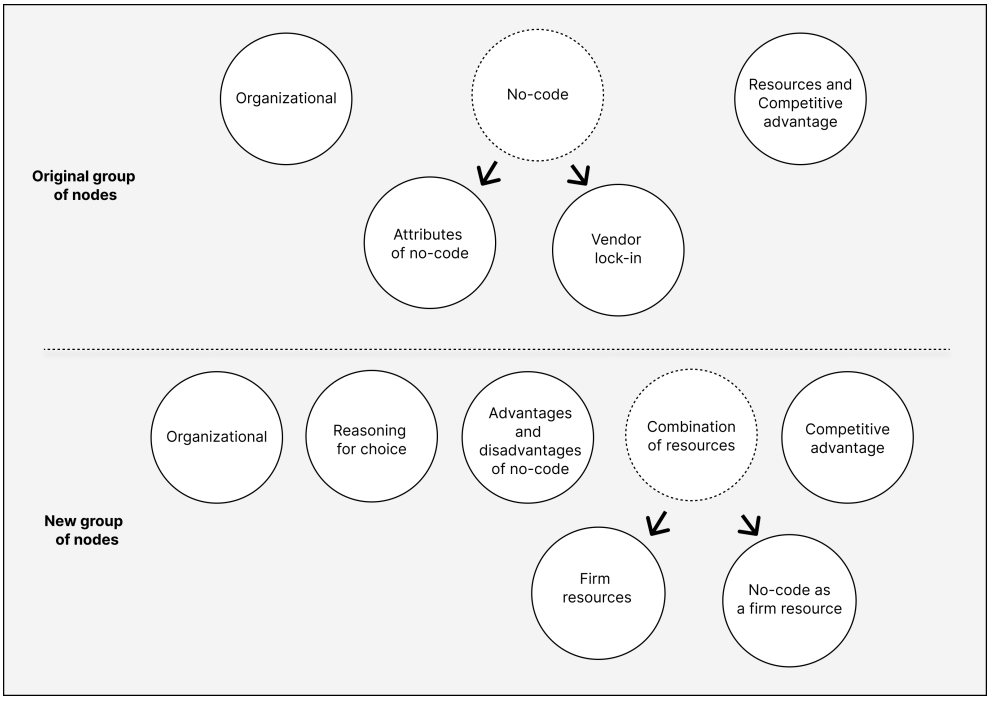


Figure 6: Nodes used for within-Case firm Coding

The nodes "*Combination of resources*" and "*Organizational*" are derived from Figure 4 from section 3.3. In order to answer how no-code contributes to a firm's competitiveness, we examine the firm's resources and the effect of no-code in combination with them. The node "*Combination of resources*" is therefore divided into "*Firm resources*" and "*No-code as a resource*". Proceeding with the within-case analysis, all the textual data from the interviews was broken down and placed into the relevant nodes. Each case firm thus became its own study (Yin, 2018). This helped us identify pertinent data that needed to be addressed in the study's main questions (Yin, 1981). We conducted individual breakdowns and categorizations of the data in order to minimize the subjectivity of the findings. Thereafter, we conducted a collective presentation of the individual breakdowns. All data that was sorted similarly was kept in the node, whereas differences were discussed further before final placement. The results of the within-case analysis will be presented in section 5.1.

4.3.2 *Cross-case analysis*

Proceeding to the cross-case analysis, the researchers constructed a new matrix in Microsoft Excel consisting of the nodes from the within-case analysis and the corresponding answers for each case firm. In order to find similarities or contrasting viewpoints between the cases, we adopted the analytical approach laid forward by Widding (2006), which builds on Grounded Theory. The process is divided into three steps, each with a higher degree of abstraction and theoretical influence: (1) open coding, (2) axial coding and (3) searching for similarities/contrasts.

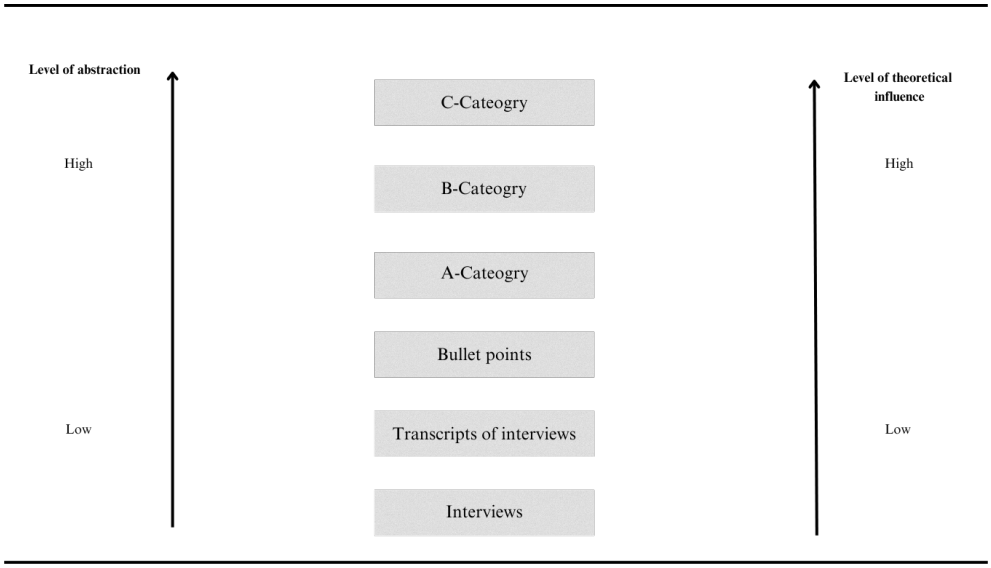


Figure 7: Analytical steps with corresponding degrees of abstraction and theoretical influence (Widding, 2006)

Open coding

Grounded in the empirical material, the purpose of the open coding is to unravel as many promising characteristics and dimensions as possible, in order to construct specific categories and subcategories. This level is referred to as level A and is characterized by a low degree of abstraction and theoretical influence (Widding, 2006). Starting with the aforementioned matrix from the within-case analysis, we began breaking down the answers from each case firm into bullet points of the most important findings. Each case firm was then assigned a number from 1 to 5, in order to make backtracking of the analysis easier. To organize and further analyze, the bullet points were assigned the code "A,"

indicating a level A categorization. The following digit is the numeration of the category, ranging from 1 to 67 (with 67 being the total number of categories extracted from the matrix). In parentheses are the numbers corresponding to the case firms, ranging from 1 to 5. If one or more case firms have the same answer, the category is combined and the numerical value of both, or more, firms is present. This made it easier to evaluate data points that were of particular interest for the main study because A-categories that are more prevalent in the data might pose as stronger evidence.

Example of a bullet point with more than one firm response:
A33 (1, 2, 3, 4, 5): No-code provides you with increased speed of development

Example of a bullet point with only one firm response:
A12 (2): No-code frees up administrative resources

Axial coding
Even though the interview guide is derived from the literature, it is not until axial coding that the connection with the theories is fully employed. The goal is to reorganize the initial A-categories into different B-categories (e.g., B1, B2, B3, etc.), based on their characteristics and dimensions (Widding, 2006). We started grouping together A-categories considered to have similarities within the elements of the theoretical framework of the study. E.g., employees, technical competency, business competency, and industry competency were considered to fall under the umbrella term "Competency is the most valuable firm resource".

B1 Competency is the most valuable firm resource

A1 (1,2,3,4,5) Employees

A2 (1,2,3,4,5) Technical Competency

A3 (4) Industry competency

A4 (5) Business competency

Figure 8: Example of B-categorization

Searching for similarities/contrasts

Following the A- and B-categorizations, the goal of the third step is to find similarities and contrasts within the empirical material. By relating the categories from the open coding and the axial coding against each other in combination with the theoretical framework, one can reach a higher level of abstraction through further categorization. The categories' dimensions are further integrated in this section of the analysis, and the categories that lack adequate illumination are better defined. Connections between the major categories are also strengthened. The analytical integration process could potentially also strengthen the "*systematic validity*" of the research (Zaltman, Pinson, & Angelmar, 1973). Starting out with the A- and B-categories, we grouped together categories that would fit together in light of elements from the literature framework. The C-categories thus illuminate both similarities and contrasts in the data material and function as a foundation for the cross-case analysis and discussion. Figure 9 shows one of the C-categories, "*C1 Intangible Resources*," with corresponding A- and B-categories (full analysis can be seen in Appendix B).

C1 Intangible resources

B1 Competency is the most valuable resource
- A1(1,2,3,4,5) Employees (5/5)
- A2 (1,2,3,4,5) Technical Competency (5/5)
- A3 (4) Industry competency (1/5)
- A4 (5) Business competency (1/5)

B2 Balance sheet resources
- A5 (4) We have substantial financial resources (1/5)
- A6 (1, 2) As a startup we lack financial resources (2/5)

B3 Social relations are important complementary resources
- A7 (1, 3, 4) Culture (3/5)
- A8 (3, 4) Methodology (2/5)
- A9 (1, 2, 4, 5) Partnership with customers (4/5)
- A10 (1) Ties to entrepreneurial ecosystem (1/5)

B8 Mindset
- A23 (3) Entrepreneurial spirit (1/5)
- A24 (1, 3, 4) Willingness to kill-your-darlings (3/5)

B11 Intangible resources as competitive advantage
- A30 (4, 5) Expertise (2/5)
- A31 (2) Switching costs locks in customers(1/5)
- A32 (3) Business model (1/5)
- A33 (3) Methodology is unique in the market(1/5)

Figure 9: C-category with corresponding A- and B-categories

4.4 EVALUATION OF THE RESEARCH QUESTIONS

As a part of the research process, researchers gain more knowledge and comprehension of the topic, which contributes to their maturity. The need to revise or replace the original research questions with more refined, or relevant ones, may arise as a result. This iterative process is inherent in the pursuit of scientific knowledge and reflects the dynamic nature of research (Yin, 2018). Our initial RQ2 was descriptive in nature and intended to provide us with a thorough understanding of the evaluation process of the case firms. During the data analysis, however, our findings indicated that the case firms did not undergo a clear evaluation process of no-code in advance. It seemed that the choice was made rather randomly, solely based on the perceived benefits of the technology. Even though this is an interesting finding in and of itself, it did not provide us with the intended insight into how early stage firms should evaluate the technology in the future. We therefore decided to change the initial RQ2 (as seen in Figure 5) to become more normative, and subsequently allow for more discussion around the collected data. Additionally, the refined RQ was more aligned with the overall purpose of the study. RQ1 remained unchanged as the data required a more comprehensive analysis and discussion. The change in RQ2 can be seen in Figure 10.

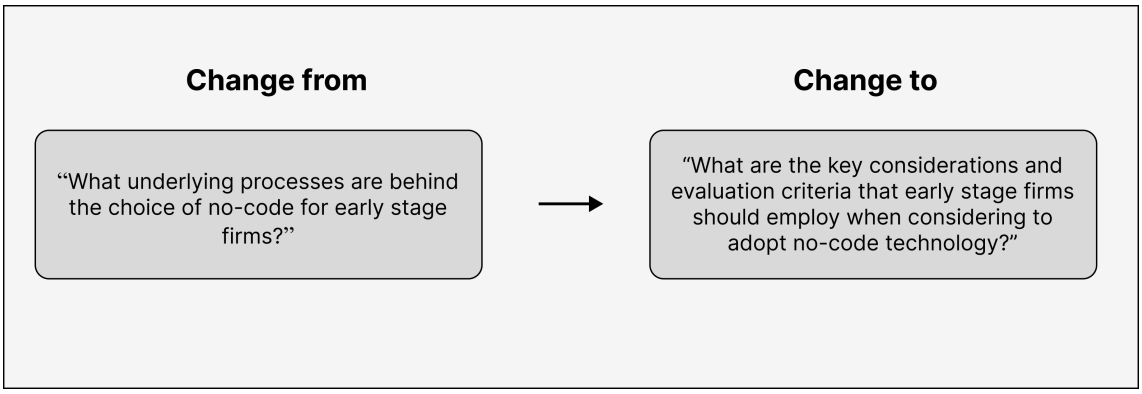


Figure 10: Change of RQ2

4.5 ETHICAL ASPECTS

Before conducting the analysis, the interviewees were given the opportunity to see the transcripts of their interview so they could make corrections and comments on the assertions made. We also requested permission to conduct

audio recordings of the interviews. Participants had to provide their informed consent in order to allow our use of the information that was gathered about them. Before finalizing the study, each case firm was sent a copy to read through. The information regarding each firm's competitive advantage was deemed sensitive, so the firm and personal names were made anonymous (Yin, 2018).

4.6 METHODOLOGICAL LIMITATIONS AND REFLECTION

The no-code term encompasses a large number of different platforms with numerous different problem-solving abilities. Studying such a broad technology with only five case firms can make it hard to obtain enough data to validate the results conclusively. Furthermore, the technology is novel, and there is little to no statistical data available, making it challenging to verify concrete evidence of increased performance and lower resource usage. It is particularly important to interpret and consider data from various sources before drawing any conclusions because the type of information obtained through the interviews could be skewed by the organizations' viewpoints (Yin et al., 2003). The interviewees might also provide the responses they believe the researchers want to hear, which is another consideration. When using the qualitative case method, it may also be challenging to determine which relationships in the findings are most crucial and what is unique to the specific case. The technique does, however, have some advantages, including a high likelihood of producing a novel theory, the likelihood that the results can be tested, and the likelihood that the results are empirically sound (Eisenhardt, 1989).

The quality of a research study is essential to ensuring the validity and reliability of its findings. Lincoln and Guba (1985) suggest the following to determine a study's level of quality: Examining the transferability, confirmability, dependability, and credibility allows for a credible assessment of the study. Study transferability is, in essence, about whether the results are representative of reality and whether a generalization can be applied to different contexts (Halldórsson & Aastrup, 2003). Such generalization, according to Creswell and Poth (2016), makes little sense in qualitative studies, and it's important that the person generalizing the results understands the original context if they're going to extrapolate the results from one context to another. Confirming the validity of the study requires that the collected data reflect the statements made by the respondents neutrally (Lincoln & Guba, 1985). Therefore, we

took necessary precautions when coding and assembling the compiled data so that the outcomes would not be susceptible to subjective interpretations. This was accomplished through individual data compilation and interpretation, as explained in section 4.3.

The consistency of the results and the reproducibility of the findings are prerequisites for the reliability of the study (Bryman, 2016). For qualitative studies, this can be difficult as the context and circumstances change over time, according to Bryman (2016). When studying a technological innovation, this can be especially true due to the rapid development of the underlying no-code platforms and accompanying technologies. In other words, our findings are strictly representative of the status quo and can be susceptible to changes in a relatively short time, due to the dynamic nature of technical progress. This may restrict other researchers from reproducing our findings in the future, and our conclusion is merely a snapshot of the present day.

Furthermore, we would also like to stress that the ill-defined terminology around no-code, from suppliers, users, and academia, means that a qualitative approach has been absolutely decisive for the study's results. Using a qualitative method enables the respondents to express themselves freely on the basis of their own perception of the technology and its impact on the resource base and competitive advantage. The ability to ask follow-up questions has also given us the chance to delve deeper into causal relationships, for which a quantitative study would not have been sufficient, increasing the quality of the data. In retrospect, the qualitative approach is considered suitable for examining the effect of no-code on resources and competitive advantage among the case firms.

5

RESULTS AND ANALYSIS

In the following chapter, we will present the results of the analysis. First, we will analyze each case individually before conducting a cross-case analysis based on the collective findings. Lastly, we answer the RQs.

5.1 WITHIN-CASE ANALYSIS

The overarching aim of the within-case analysis is to build in-depth knowledge of each case firm using the aforementioned nodes from section 4.3. By describing each case individually before conducting the cross-case analysis, unique patterns from each case firm can emerge (Eisenhardt, 1989). There are no formal requirements for how to conduct a within-case analysis, but it usually involves describing the cases in great detail. However, due to the sensitive content provided by the firms, and their absolute desire to remain anonymous, such elaboration is not possible. Individuals with knowledge of the industry could potentially recognize the firms upon the reproduction of specific characteristics. The category of "organizational" will, however, delve somewhat into the characteristics of the firms and their respective markets in order to provide the readers with a context in regards to the results.

5.1.1 *Case firm A*

Organizational
Case firm A was established in 2021 and operates in the fintech space, providing its customers with seamless transactions and workflows. Their competitive landscape is characterized by a large number of existing actors with a long track record and substantial financial resources, providing solutions to the market problem. The competition is also strengthened by the presence of several substitutes. Competition can therefore be described as high. However, the degree

of innovation was described as low because the existing actors have reached a position in the market where their focus is on maintaining the customer base rather than innovating and making new service offerings. The CEO of Case firm A described the firm´s software solution as an incremental innovation in the market, focusing on making existing operations better suited for the end user.

Competitive advantage

When asked about the competitive advantage of the firm, the answer lies in the way the firm develops the solution. With a high degree of customer-focused development, the firm is able to rapidly create the correct functionalities and the right amount of them. Where the competitors focus on creating vast amounts of functionality with a "the more the merrier" mentality, Case firm A focuses on creating the exact needed functions. In doing so, Case firm A is creating a situation where their development processes are in close partnership with their customers. When asked about no-code´s effect on the competitive advantage of rapid development and close relationships with customers, Case firm A continues by replying that no-code is crucial to achieving such partnerships. "*When customers see that their feedback is quickly converted into new functionalities, they feel more included in the development process*".

The interviewee further stated that rival firms are able to replicate the solution entirely, but due to Case firm A's rapid development, they would constantly be months behind. By the time the competitors had blueprinted their solution, Case firm A would have pushed new features and gained even better customer insight. Strengthening this is the fact that the main competitors have a long runtime using large conventional codebases, and in order to imitate Case firm A they have to start from scratch.

Combination of resources

Core firm resources were described as the specific technological competencies, ties to the entrepreneurial ecosystem, and the overall firm culture. However, the most valuable firm resource is described as the aforementioned partnership with its customers, which allows for deeper insight into their problems. As a startup, the financial resources are strictly limited, and the need for an environment where you can decrease the financial spending on development is crucial.

Furthermore, the interviewee highlighted that no-code technology frees up creativity in the firm. With easy-to-build interfaces, the developers can build

new features in a creative manner because the progress is way faster, enabling them to create new aspects in a workday. "*We can use one week to improve our solution, whereas the same changes would take a month with traditional coding*". In addition, the interviewee emphasizes that no-code development enables the firm to test functionalities without imposing substantial resources, mainly time and human capital. Functions that do not provide additional value can be scrapped, making the iteration process both fast and less resource-intensive. The interviewee recognizes no-code as a resource that competitors do not have.

He claims that despite the no-code platforms' lower level of technical competency requirements, the need for such competency is still not completely eliminated. This is contrary to what the no-code vendors advertise. In order to fully take advantage of no-code, you still need a certain level of fundamental development skills. "*It is not true that anyone can use no-code to build complex software solutions; Tove, 80 years old, cannot replicate us*". No-code is less technically demanding, but there is still a need for skilled technicians to make use of it correctly. Areas such as cybersecurity and system architecture are still knowledge-intensive.

Advantages and disadvantages of no-code
The interviewee recognized the main advantages provided by the no-code technology as rapid and fast development, further going into the relationship between the pace of development and the partnership with customers. "*It allows us to get customer feedback and rapidly make changes to our solution, in contrast to building traditionally*". This rapid development and testing creates close ties with customers, laying the groundwork for a unique trust.

A drawback underlined by Case firm A is the fact that no-code technology enables you to create software within the framework provided by the no-code provider. If you want to create complex functionalities that are not currently offered by the NCDP, you are facing a wall that is hard to work around. This is not currently a concern for Case firm A, but the interviewee is clear that the development might be converted to traditional development in the future if the complexity of their software solution increases. For the time being, Case firm A has managed to create workarounds that minimize the restrictions of the no-code platform they use. Another critical drawback is what's called vendor lock-in, where the software created on a specific no-code platform cannot be extracted or converted to another no-code distributor, thus creating a lock-in effect. For Case firm A, this increases operational risk because their survival

might depend on the survival of the no-code vendor. "*If the no-code distributor goes bankrupt today, we would have a fundamental problem*".

Reason for choice
No-code technology was used from the start in 2021. The interviewee is honest when asked about the reasoning behind the choice of no-code and replies that there was no initial rationale other than a good price on the chosen no-code platform. No evaluation regarding the firm's resources was made prior to the adoption of no-code. For Case firm A, the main concern prior to adoption was the vendor lock-in aspect, but the positive attributes of no-code exceeded these drawbacks, and the decision landed on using no-code.

Summary of no-code and Case firm A
Case firm A operates in the fintech space, where competition is described as high, but innovation is low. Their software solution focuses on making existing operations better for the end user, which is achieved through a close partnership with customers. The firm uses no-code technology to rapidly develop and test functionalities without imposing substantial resources, making the iteration process both fast and less resource-intensive. No-code technology enables the firm to obtain a competitive advantage through rapid development, which leads to close relationships with customers. Competing firms can easily replicate their software solution, but would constantly lag behind in pushing new features and gaining customer insight. However, no-code technology limits development within the frame provided by the no-code vendor, and the lock-in effect was therefore a main concern prior to adoption.

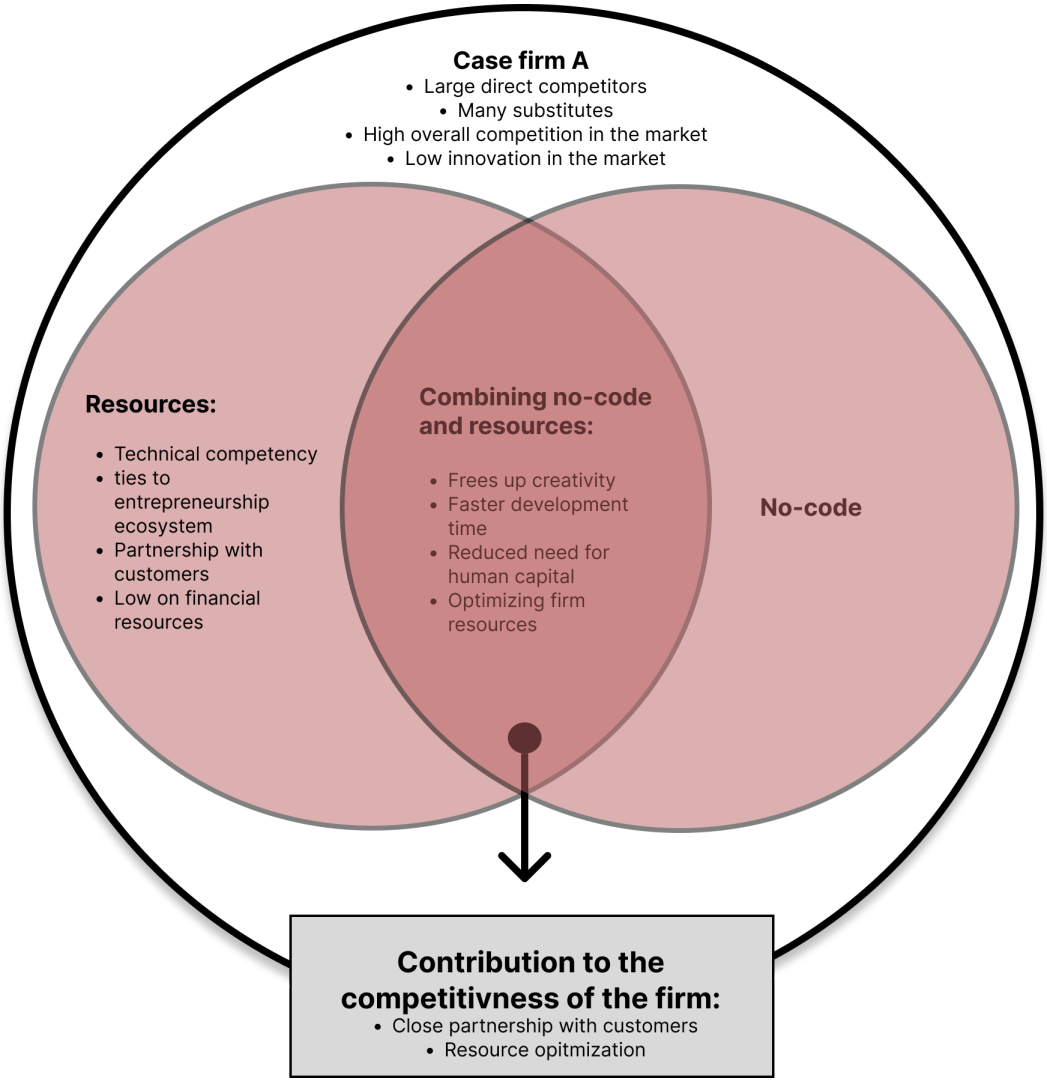


Figure 11: Within-case analysis Case firm A

5.1.2 *Case firm B*

Organizational
Case firm B operates in the renewable energy sector and was established in 2021. When planning to build new wind- or solar panel farms, the screening process of potential properties can be both time-consuming and costly. Case firm B provides a digital platform for seamless and cost-saving screenings. The degree of innovation is deemed high by the interviewee, and the focus is on digitizing operations that are currently manual in the industry. Their core software system is defined by a high degree of complexity and development is carried out using traditional methods. Internal processes and microservices however, are made using no-code tools. The competitive landscape is described as medium for Case firm B, with competitors being a few large enterprises with substantial financial backing. Current solutions are highly customized for each customer due to the nature of the manual work required. The differentiation of Case firm B lies in the digitization of generic tasks, making the solution more flexible for the customer.

Combination of resources
The key resources of Case firm B are the employees who possess deep technical competency. The technical know-how of the founding team also enabled them to start the development of the solution traditionally and acquire the correct people. As a startup the financial resources are scarce, said to be a limiting factor when competing with other market actors. In general, the interviewee states that: "*we have fewer resources than our larger competitors, which makes it very important to use them correctly*".

Case firm B uses no-code to develop internal processes, which frees up administrative resources in the firm. Furthermore, the interviewee states that no-code changes the way resources are utilized. "*The speed of development using no-code is ten times faster than traditional development, meaning that you can use one resource instead of ten to do a task*". No-code is thus less resource-intensive, switching resources from development to other operationally critical tasks. For Case firm B, the freed up resources enable the firm to do tasks that would not otherwise be done, described as "*customer iteration tasks*," such as follow-up emails and communication. In turn, this allows for better customer relations as described by the interviewee.

In order to adopt no-code successfully, the interviewee highlights the impor-

tance of technical competence. The degree of competency is lower than for traditional coding, but the need for somewhat technical know-how is still required. The specific technical competency is linked to a logical understanding and cyber security; "*What processes must be done first, and what happens if I change this process?*". A comparison is made to Microsoft Excel, where the everyday person can make advanced spreadsheets relatively easily and quickly. But, the architecture of the spreadsheets might be horrible, thus making the programs hard to understand, maintain, and operate. In a software environment, bad architecture might add up to problems later, highlighting the need for some technical know-how.

Competitive advantage
Their competitiveness is described as medium, because of their lack in financial resources. With large, well-funded competitors, Case firm B aims to compete on the way they reach customers through development speed, flexibility, and iterations. The strategy of implementing switching costs increases entry barriers in the market and ensures customer loyalty. The competitive advantage of Case firm B is described as the way they develop the solution, and the speed and flexibility are greatly appreciated by the customers. This enables Case firm B to better cater to the needs of the market. "It's all about making what the customer needs the quickest". Even though the development of the main solution is carried out traditionally, the interviewee sees great benefits in the aforementioned enhancement of customer communication made possible by no-code. For Case firm B, no-code is merely a means to strengthen what they already do best.

Advantages and disadvantages of no-code
When asked about the advantages of no-code technology, the interviewee points out that the implementation time is greatly reduced. Development processes can follow a lean method with continuous iterations and testing of the solution. Additionally, the cost and time of developing a functional software solution is, in their opinion, greatly reduced. "*The cost of hiring developers is high, and even small things that take a short time using no-code can take weeks traditionally*".

However, the interviewee explained that the price of no-code initially is lower due to the reduced need for salaries for developers, but scaling often comes with heavy price increases. Eventually, the price level will match that of traditional development. Another aspect considered a limitation is the fact that the fast pace of development can create a false sense of constant delivery to

the customer. "*Once you reach the ceiling for what can be made in the no-code tool, the progress slows down significantly, possibly disappointing the customer*". Going further, no-code development is dependent on the functionalities within the platform. Some degree of tailoring is provided, but it is still far less than what is possible with traditional development.

Reason for choice

No-code was not fit to solve the specific problem Case firm B wanted to solve due to the sheer complexity of the solution. "*We develop a very advanced software system, and the capabilities of no-code are not sufficient for our purpose*". However, internal processes and microservices, which are not considered critical, are still made using different no-code tools. Moving on to the vendor lock-in of no-code, it seems to be a barrier to creating the core system for Case firm B. Administrative and internal processes are less crucial for main operations and can be easily replaced, making it more likely for no-code to be effective for such tasks. Internal mailing services and customer success programs are examples of such processes. However, the filter for no-code stopped early as the limitations regarding functionalities and flexibility made it hard to develop the desired main service. An evaluation of the firm's resources was therefore not made prior to the choice.

Summary of no-code and Case firm B

Case firm B operates in the renewable energy sector and provides a digital platform for cost-effective and seamless screening of potential properties for wind- and solar panel farms. The company focuses on digitizing manual operations in the industry, using traditional development to create a complex software system. While the company employs different no-code tools for internal processes and microservices, the main software service requires traditional development. Case firm B faces competition from large enterprises that provide highly customized solutions for each customer. However, the company differentiates itself by digitizing generic tasks, making the solution more flexible for end-users. Case firm B competes on development speed, flexibility, and iterations, with the technical know-how of the employees being the main key resource. The company implements switching costs to increase entry barriers in the market and ensure customer loyalty. The interviewee pointed out that the advantages of no-code include reduced implementation time and the cost of developing functional software solutions. However, the price level of no-code may eventually match that of traditional development, and the fast pace of development can create a false sense of constant delivery to the customer. The interviewee

also highlighted the importance of technical competence for adopting no-code successfully. While administrative and internal processes can easily be replaced with no-code, the limitations regarding functionality and flexibility makes it challenging to develop the desired main solution.

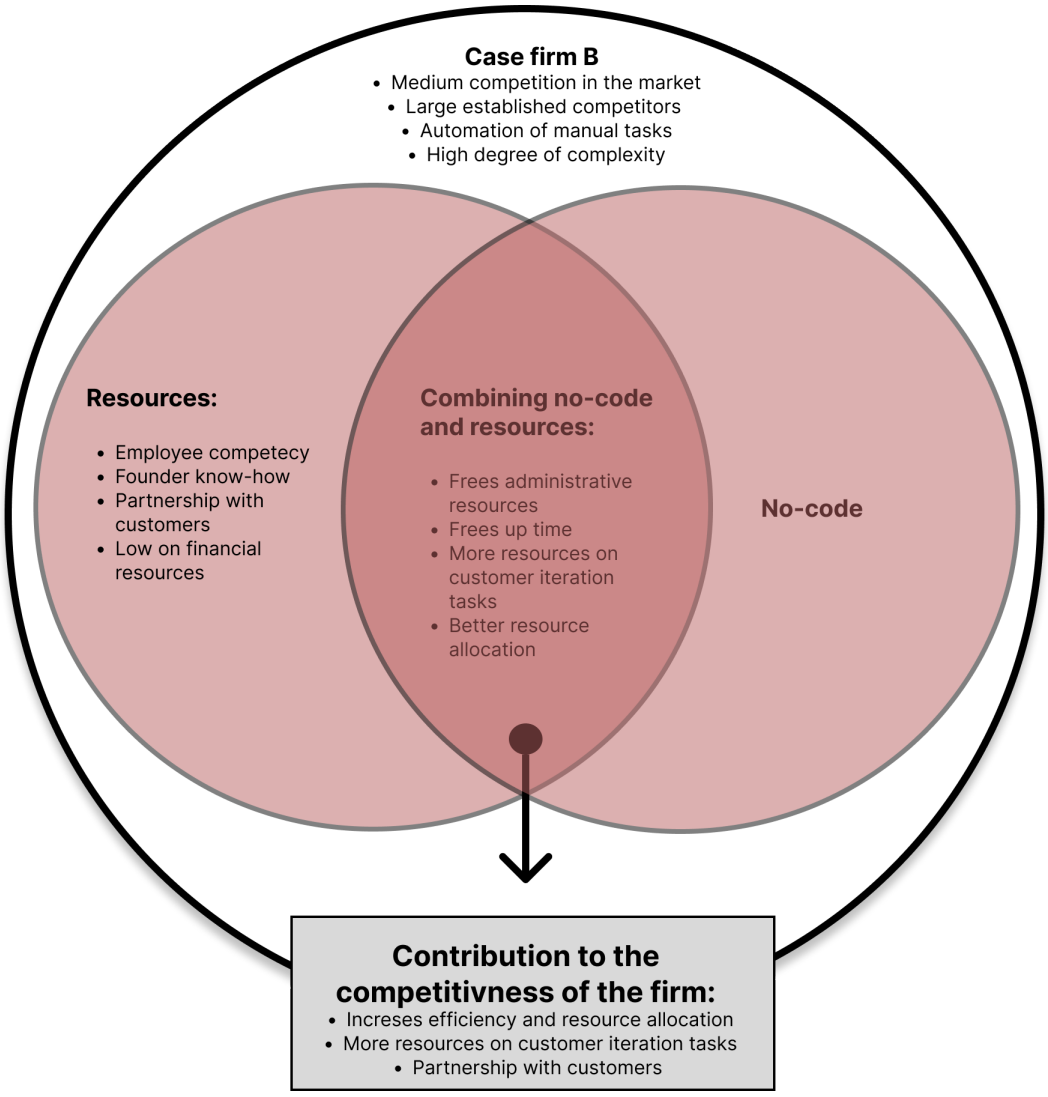


Figure 12: Within-case analysis Case firm B

5.1.3 *Case firm C*

Organizational
Case Firm C is a consultancy service and incubator for startups in Norway, with a runway dating from 1988. Their vision is to help as many startups as possible become viable and guide them from idea to market through their experience and know-how. The competitive landscape consists of other consulting agencies, characterized by high competition in the market. In the cross-section between consultancy and startups, however, the competition is relatively low due to the limited financial resources of startups, which restrict their purchases of more expensive consulting firms. Their business model is "*sweat for equity*", meaning that payment can be done through shares in the startups, allowing them to purchase consultancy without direct monetary payments. This is why the interviewee considers Case firm C to have a competitive edge in the startup market. Their main service is helping entrepreneurs with business ideas to validate their concepts as a tech partner. While working with numerous startups, Case firm C uses both no-code and traditional development.

Combination of resources
Key resources of the firm are described as their methodology, which is deeply rooted in all processes of the firm. For startups, the main concern is to validate the business idea in the market and find a product-market fit. Another concern is the allocation of resources, because startups tend to lack substantial resource bases. The methodology is therefore to continually user-test and validate business concepts in order to minimize the waste of time and resources on faulty ideas. The culture and overall "spirit" of the firm are valued as key resources, focusing on continually iterating and developing. Furthermore, the fact that a lot of their employees have backgrounds from larger companies makes it easier to bring valuable organizational know-how to the startups they work with.

No-code is described as highly complementary to the firm's methodology: "*I would say that there is a direct connection between no-code and our methodology*". The no-code attribute of rapid development makes the validation process of ideas far faster and less resource-intensive for their customers. When combined with the know-how of the firm, the no-code environment is perfect because both Case firm C and the customer startups would use fewer resources to validate the concepts. "*The utilization of resources becomes better when using no-code*".

Competitive advantage

With a high degree of competition in the consultancy industry, Case firm C has found a niche market for startup consulting. Combined with a complementary business model of "*sweat for equity*", they are able to deliver their services in a far more competitive manner than other consulting firms. Building on their entrepreneurial mindset of rapid development and testing, they are better equipped to deliver to their niche market. This, in turn, contributes to the firm having a competitive advantage in the market.

Advantages and disadvantages of no-code

When asked about the advantages of no-code, the interviewee responds by stating that the time and cost of development are greatly reduced. "*Do you want to pay 100 dollars a month, or do you want to pay two developers full salaries to create something that might be thrown away in a few weeks?*". The ability to swiftly and seamlessly create and delete functionalities is also a strong suit of no-code. "*We don't want our developers to work on something that is going to be scrapped shortly after it's done*". Therefore, the use of no-code makes it easier to quickly develop a minimum viable product (MVP) for customer feedback without getting attached to the solution or using large amounts of resources that are unnecessary. Lastly, because no-code is easy to use for non-programmers, and developers are in short supply, the threshold for testing your business idea is reduced.

As per the disadvantages of no-code technology, the interviewee refers to no-code development as a "*happy path*" for those who are comfortable using the tools. The chosen tools provide a fast track for development within the framework and functionalities of the no-code platform. "*If you suddenly need specific functions that the tools do not provide you with, you would find yourself in a difficult situation, and progress might come to a halt*". The worst-case scenario might even include building the entire software from scratch, as explained by Case firm C.

Reason for choice

By working with a number of startup companies, Case firm C has gained insight into both the use of no-code and traditional development. When asked about what would happen if all development was carried out using no-code technology going forward, the interviewee claimed that the traditional developers would be dissatisfied with the decision. "*They like to have control over the code base themselves as well as build lines of code*". This, however, does not seem to be a scenario for which Case firm C can fully employ it because the different startups

have vastly divergent problems to be solved. Some could greatly benefit from implementing no-code, whereas others might be too restricted within the frames provided by the technology. "*It all comes back to the problem you wish to solve*". The choice is therefore dependent on the startup ideas, and an evaluation is made prior to each project.

Going into the vendor lock-in aspect of no-code technology in Case firm C highlights an important issue. Due to the relatively newness of the no-code providers, their lack of long-lasting track records and stable cash flow might pose a problem for anyone using no-code. "*If the vendor suddenly goes bankrupt, your entire software base is lost*". Therefore, it is important to do a thorough risk analysis and assess whether your solution is permanent or temporary before choosing no-code. Furthermore, Case firm C also experienced the transition from one NCDP (Webflow) to another because of restrictions on the functionalities required to build the desired web application. Due to the fact that one cannot extract the code base from one NCDP to another, the entire application had to be built from scratch, strengthening the impression of vendor lock-in. The vendor lock-in aspect is therefore crucial to consider when starting a new project.

Summary of no-code and Case firm C

Case firm C is a Norwegian consultancy service and incubator for startups that helps them from ideation to market through its experienced entrepreneurs and know-how. The firm's key resources are their methodology and overall "spirit", built on continuously testing and validating business ideas, to minimize the waste of time and resources. The interviewee sees great advantages in using no-code technology when working with startup companies, as it provides a tool for exploring creative ideas and obtaining market validation with minimal additional resources. However, the interviewee also notes the disadvantages of no-code technology, including limitations in functionality and the risk of vendor lock-in. The interviewee claims that traditional developers would be dissatisfied with an all-no-code approach, but Case firm C believes it all comes back to the problem you wish to solve.

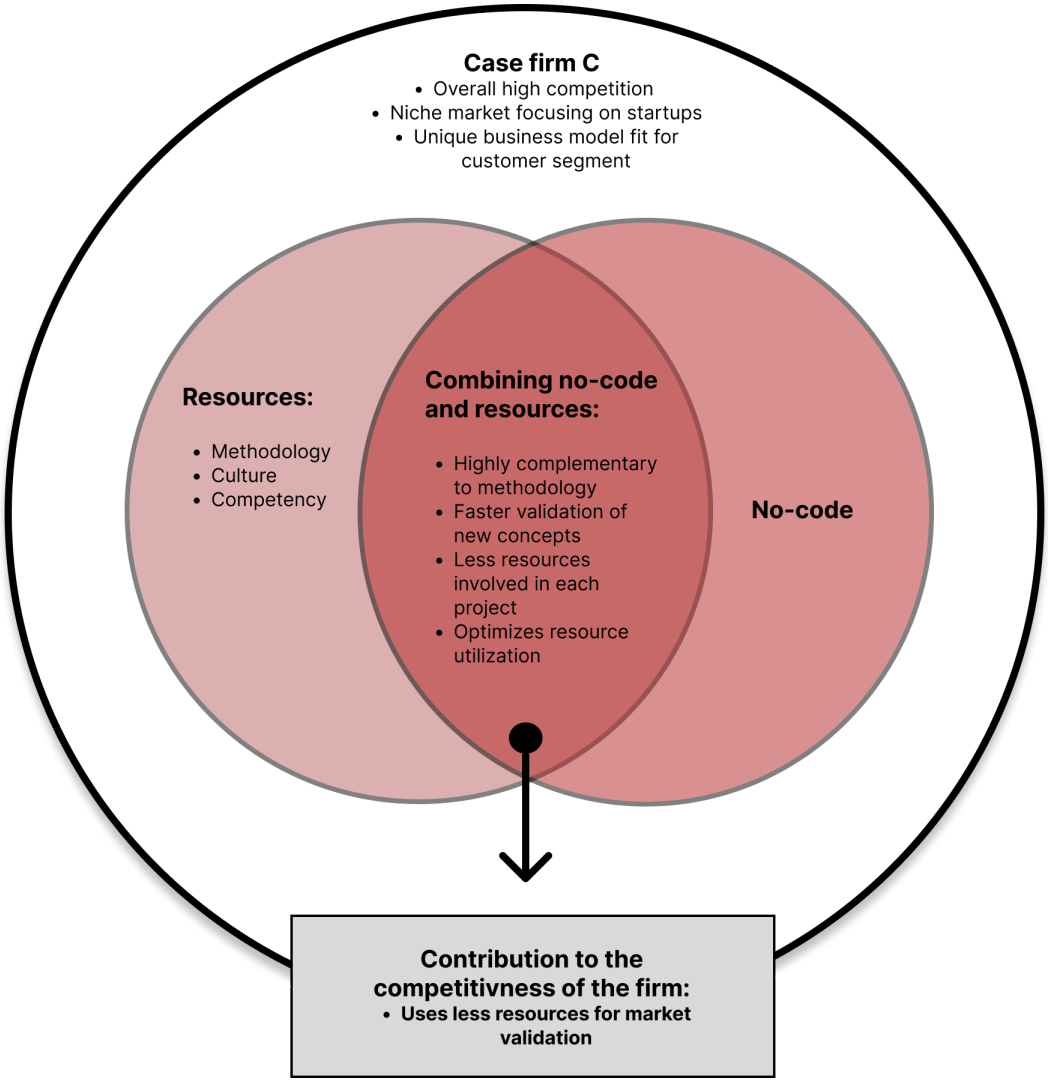


Figure 13: Within-case analysis Case firm C

5.1.4 *Case firm D*

Organizational
Case firm D was established in 2019 as a spin-off firm from a larger energy company. The firm has developed a modern energy platform for easy and streamlined purchases of electricity for Norwegian companies. As an energy provider, Case firm D competes with 140 other companies but recognizes only two as direct competitors in the way they approach the market.

Combination of resources
With a large mother company, Case firm D was created with solid industry competency and financial resources from the start. The firm identifies its key resources as employees, in-depth industry competency, financial resources, and culture. The acquisition of the correct people has been a main priority for Case firm D from the start; "*no employee can be picked off the streets*". Furthermore, the interviewee described the culture of the company as an important complementary resource in which their employees can thrive. "*Without respect for the individual's needs and how they work best, the right people become wrong*". The financial resources also allowed Case firm D to purchase external expertise in the form of eight consultants who focused on customer insight. By using no-code, the interviewee states that the technical resources are freed up in the firm. With the combination of constant dialog with the customers and the speed of development provided by the no-code platform, Case firm D was able to efficiently create their solution, tailored to the customer's needs. "*We came much faster to the customer with a good solution, without imposing extensive resources*".

Case firm D further recognizes some important resources one should have when adopting no-code technology. Firstly, the interviewee specified that one should have an experienced data scientist who understands digital architecture. This becomes apparent in what he called a "*technical debt*". Technical debt is defined as "*technical compromises that can yield short-term benefit but may hurt the long-term health of a software system.*" (Li, Avgeriou, & Liang, 2015, p. 193). When developing software, such technical debt occurs when small errors accumulate over time, creating large problems in the future. Technical competence is therefore needed to make this debt as small as possible. Furthermore, one needs structure in the organization for the tasks to be done.

Competitive advantage

According to a customer survey, Case firm D ranked number one in customer satisfaction, and the interviewee explained the company's market position in terms of how they develop their solution according to the customers. Their development process has been centered around the customer, with a high focus on approaching them on their premises. They have successfully identified the market's pain points through continuous customer communication, which has allowed for the quick development of the solution. The interviewee defines this customer relationship as their main competitive advantage to date. "*The competitive advantage lies in how fast you can reach the customer*". They also adopt a "*straightforward and honest*" approach in an industry of hidden fees, building a brand of trust.

Advantages and disadvantages of no-code

The interviewee, when asked about the advantages of no-code technology, expressed that you have to assess the problem you wish to solve before choosing no-code. For Case Firm D, they wanted to create a user interface for their solution, but did not know how or what functionalities it should include. They needed a tool to rapidly create functionalities based on the feedback from the customer. Furthermore, flexibility was important in order to iterate on the solution based on user testing. The ability to chance and scrap functionalities that were not sufficient or did not create value was also a priority. Therefore, the need for something to create a solution based on a lean startup methodology was the main objective, and no-code had a nice fit with these requirements. "*It's all about the speed of development and the flexibility you get from no-code; it's formidable*". The interviewee also states that the speed of development decreases attachment to what you create and that the problem of "*killing your darlings*" becomes far less prominent using no-code.

As per the disadvantages of no-code, the interviewee elaborated on the effects of no-code on the progression if there was a lack of structure. "*If you lack a structured approach and structural frames for what is being built, the fast pace of no-code development can enhance the speed at which things go wrong*". By experience, Case Firm D started out with a focus on speed of development at the expense of structure. This in turn made the solution slow, not user-friendly, and hard to maintain because of the accumulated technical debt. However, the interviewee assured that this is not a problem with the technology itself but rather with how they "*misused*" it in the beginning. He mentioned that the no-code tools have a set of predefined functionalities as another drawback. "*If you meet a problem*

that you cannot solve using no-code, you are dependent on the vendors to develop the no-code platform further".

Regarding the aspect of vendor lock-in, the interviewee stated that: "*The real value does not lie in the code itself but rather in the customer insight, design, and user experience, which are not tied to the no-code platform*". In other words, vendor lock-in is not considered a drawback by Case Firm D, since the value does not lie in the software itself, but in customer insight and relationships. In the initial phase of the firm, they recognize no-code as a potential advantage, because of the speed at which you can gain customer insight and develop the software. When the firm has reached a satisfactory product, thus not in need of more market validation, the interviewee does not impose no-code with further advantage over traditional software development.

Reason for choice
From the start, they adopted Appfarm´s no-code platform to build their software solution. This enabled the firm to reduce the need for IT consultants to build the software and focus more on customer needs. However, they acknowledge that the firm was not rigged for the no-code environment from the start, but rather was more prone to building traditionally. This was because the in-house technical competence primarily consisted of traditional developers. Even so, the fast pace development environment, as well as the flexibility to rapidly scrap or change functionalities, outweighed this consideration and the choice fell on no-code.

Today, Case firm D has undergone the transition from no-code to traditional development. This came as a result of the developers dissatisfaction with the no-code tools. "*Traditional developers have a certain passion for writing the code themselves, which no-code deprives them of*". During the transition process, Case firm D needed to expand their development team and spent a lot of time looking for the right developers, referred to as "*finding a needle in a haystack*", due to the shortage of such competency.

Furthermore, the interviewee explained that no-code is great for creating software within the framework of the tools. If you want to develop something that is not possible within these frames, the progression stops. After two years of using no-code, the dissatisfaction became dealbreaking for the firm, because they risked losing their developers and thus valuable know-how. "*With no-code we free up some resources, but by continuing we would lose more critical resources,*

namely the developers". The transition was done using parallel processes, both developing the main no-code solution and starting from scratch in Javascript, using the no-code version as a blueprint. This made the progress slower for a short period of time until the new version was complete. After the completion of the Javascript version, operations went back to normal.

Summary of no-code and Case firm D
Case firm D is a spin-off company from a larger energy company in Norway operating in a market with high competition. However, only three are said to be direct competitors due to the way in which they are reaching the customers. Key resources of Case firm D are the employees, the culture, and the financial resources. Their competitive advantage lies in their customer-focused development, which leads to close relationships with their customers. The use of no-code technology enabled the firm to undergo rapid development and decreased the need for technical resources. The perceived advantages of no-code are the speed of development and agility provided by the tools, which shift the resources from technical to more customer-centric, leading to a competitive advantage for the firm. No-code was used from the start to build a user interface, but the firm underwent a transition towards traditional development, because of internal pressure from its developers. The disadvantages of the technology are related to the structural approach of the firm and are not related to the tools themselves. Without a good structure for development and some technical know-how, no-code would accelerate the rate at which you accumulate technical-debt.

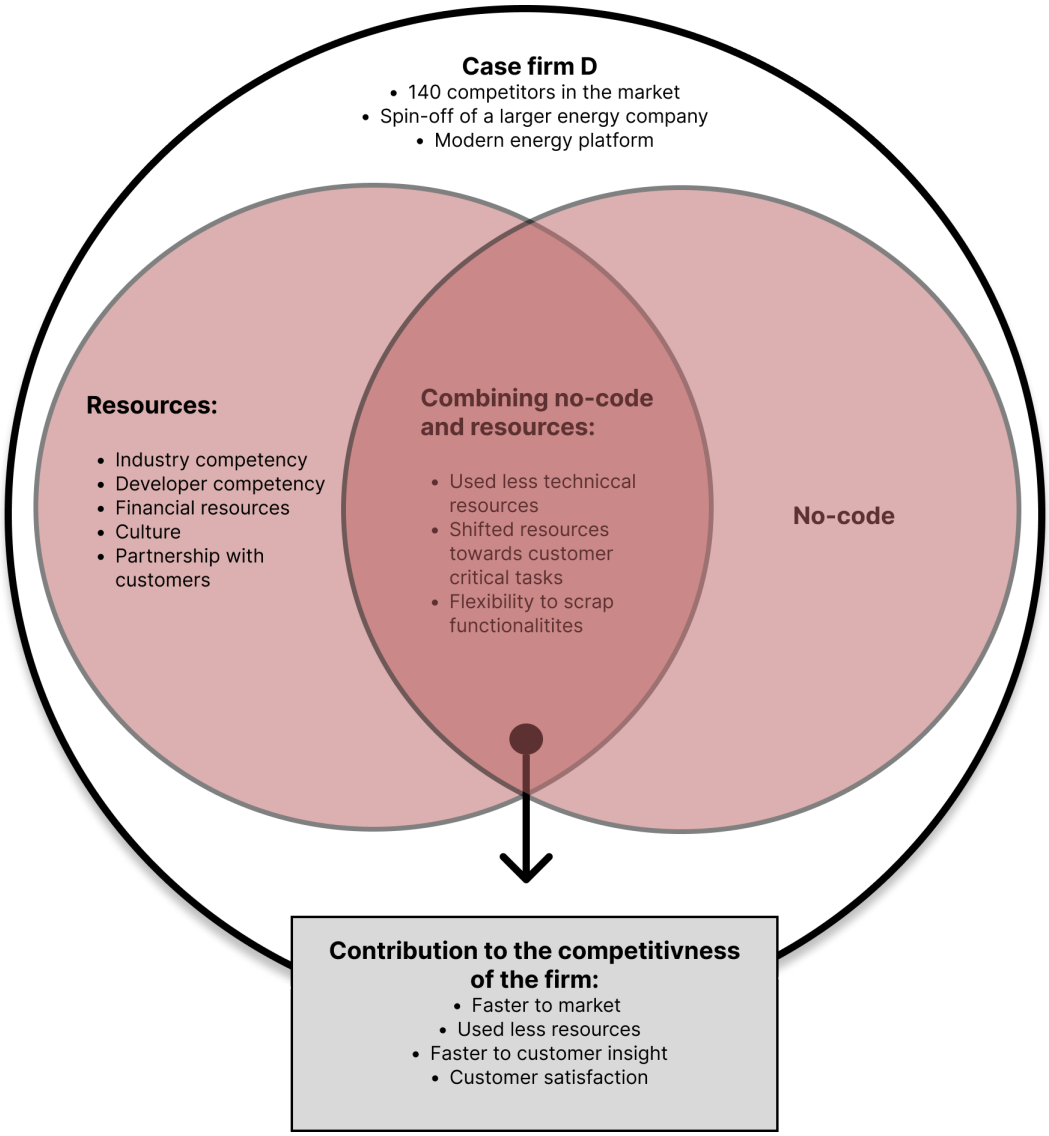


Figure 14: Within-case analysis Case firm D

5.1.5 *Case firm E*

Organizational
As a "born digital" IT consulting firm, Case firm E was founded in 2016 around the use of no-code technology. The competition in the market is considered high, with the largest competitors providing the same service using traditional development. Case firm E focuses on a niche market, Robotic Process Automation (RPA), in which they consider themselves market leaders on a global scale. Their competitiveness is also related to the in-house development of certain expertise that is not easily replicable. This expertise, in combination with the firm's focus on interdisciplinarity and close partnership with technology providers and customers, creates an environment in which Case firm E is able to differentiate.

Combination of resources
Key resources of Case firm E are described as 95% the employees of the firm, the collective know-how. By focusing on different academic disciplines, they are able to create a more wholesome understanding of the customers' problems. In contrast to other IT consulting firms, Case firm E hires economists and business developers to work as developers of software solutions using no-code technology. This combination of business understanding and the use of no-code makes Case firm E able to thoroughly examine customer problems from a more complete perspective, and deliver more value than its competitors. The methodological approach is a complementary resource, but the interviewee is aware that this approach does not differ from the competitors. However, the composition of the know-how with the method and use of no-code creates a bundle that is hard to replicate. "*When working with the no-code platforms, the business people become technologists as well*", further strengthening the inimitability of the core resource.

Technical competency is a resource the interviewee underlines as important when utilizing no-code. The need for employees who understand the layers of application development as well as the business perspective is crucial. Every new employee therefore has to undergo a training program in the use of the different no-code platforms. "*When employees have the combined knowledge of business and development, the results are better*". The interviewee adds that this does not differ from the use of traditional development. However, no-code enables Case firm E to do more with fewer resources, which they can use to focus more on the problems of the customers. "*The technology frees up time by up to 20–30% through providing you with pre-built components, making the development*

of generic tasks more streamlined". Time spent on tasks such as cyber security, architecture, and code review can be spent on the unique problems for the specific customer. The interviewee also explains that the need for human capital is greatly reduced using no-code. "*Instead of needing a team of twelve people, you can have a team of three, as long as they have the technical competency and business understanding*".

Competitive advantage

The competitive advantage of the firm is said to be how they develop their in-house competencies, by leveraging no-code technology to create employees with both business and development know-how. This allows for better resource utilization as well as creating hard-to-replicate competencies. Whereas the competitors still have some business developers, the fact that they use traditional development methods makes the teams larger; more people are involved and there is more room for errors and miscommunication. The mindset of "*we are going to own the problem with the customer*", is also said to have a profound impact on the way in which they differentiate. Case firm E has built strong relations with both customers and technology providers, such as no-code vendors. "*They can partner with everyone, but they like to work with us the best*".

Advantages and disadvantages of no-code

A perceived advantage of using no-code technology is its ability to bridge the gap between the business perspective and the development of software solutions, as claimed by the interviewee. The technology is easily comprehensible for citizen developers and therefore enables business people to gain more control over the development process. This in turn makes the development less resource-intensive by reducing the need for human capital, due to the reduced need for deep IT-expertise. "*When working with your customers, you can show them the code visually, and they can even participate in the development. Changes can happen right away, shortening the feedback loop*". Additionally, the speed of development is an advantage when working with innovation projects, because it allows for faster workflows and a faster time to market with functional prototypes.

A disadvantage highlighted by the interviewee is the fact that no no-code platform is sufficient to solve all problems due to the restrictions of each platform. "*It is an illusion that a single no-code platform can solve all your problems*". The flexibility of no-code is also perceived as lower than traditional development, because you are bound by the functionalities within the given platform. The lower flexibility is therefore a disadvantage of no-code, as stated by the

interviewee.

When asked about vendor lock-in, he acknowledges it as a problem one needs to be aware of. Even if the tools provided you with the ability to extract the code base, the code would be useless because it's tied to the specific framework of the tool. You can extract the "*blueprint*" and overall logic of the solution, but there is still a large job of converting it in another tool. "*When deciding to use no-code, it is important to evaluate the benefits and risks associated with each new project, and vendor lock-in must be addressed*". To add to the topic, the interviewee explained that traditional development also incurs a degree of vendor lock-in, because the different layers of the software cannot easily be transitioned to other programming environments, and that the switching costs for doing so are also high. For Case firm E, the difference in vendor lock-in in practical terms is indistinguishable from traditional development, but must be addressed regardless when starting on a new project.

Reason for choice
Since the start in 2016, the use of no-code has been central to the development of the firm. The first RPA project required Case firm E to develop a user interface, an application, for which they lacked sufficient traditional developers. The restricted resource pool of traditional developers led to the adoption of a no-code platform, and Case firm E found that the specific problem could in fact be solved in this manner. Case firm E thus started to build the company around no-code technology, targeting business people for hiring and developing their competencies to create a unique market position.

If Case firm E was to transition it's operations from no-code to traditional development, the interviewee´s main concern would be the misalignment of the human capital. Because the company's key resource is citizen developers, the transition would shave away it's competitiveness in the niche market. Rebuilding the company would require fundamental rehiring because the existing human capital would not be of value. If the competitors converted to no-code today, the interviewee believes they would have great difficulty convincing their traditional developers to adopt and use it. However, he also believes that the competitors will ultimately be pulled towards the use of no-code, which he explains as "*a gravitational force*". For the time being, Case firm E views no-code as a part of their competitiveness in the consulting market.

Summary of no-code and Case firm E

Case firm E is a born-digital IT consulting firm founded in 2016, focusing on the niche market of RPA. By combining employees with diverse academic backgrounds and leveraging no-code platforms, the company differentiates itself from competitors. The firm's key resources are its employees, who possess both business understanding and technical competency due to no-code, allowing for a comprehensive perspective on customer problems. In combination with the faster pace of development, and subsequently shorter feedback loops, no-code enables Case firm E to pay more attention to the needs of each specific customer. Case firm E's competitive advantage lies in its ability to develop in-house competencies by leveraging no-code technology, which is a hard to replicate resource. However, no-code is not fit to solve all problems due to functionality restrictions, limited flexibility and vendor lock-in. For each project, a risk analysis is therefore conducted in order to evaluate its applicability.

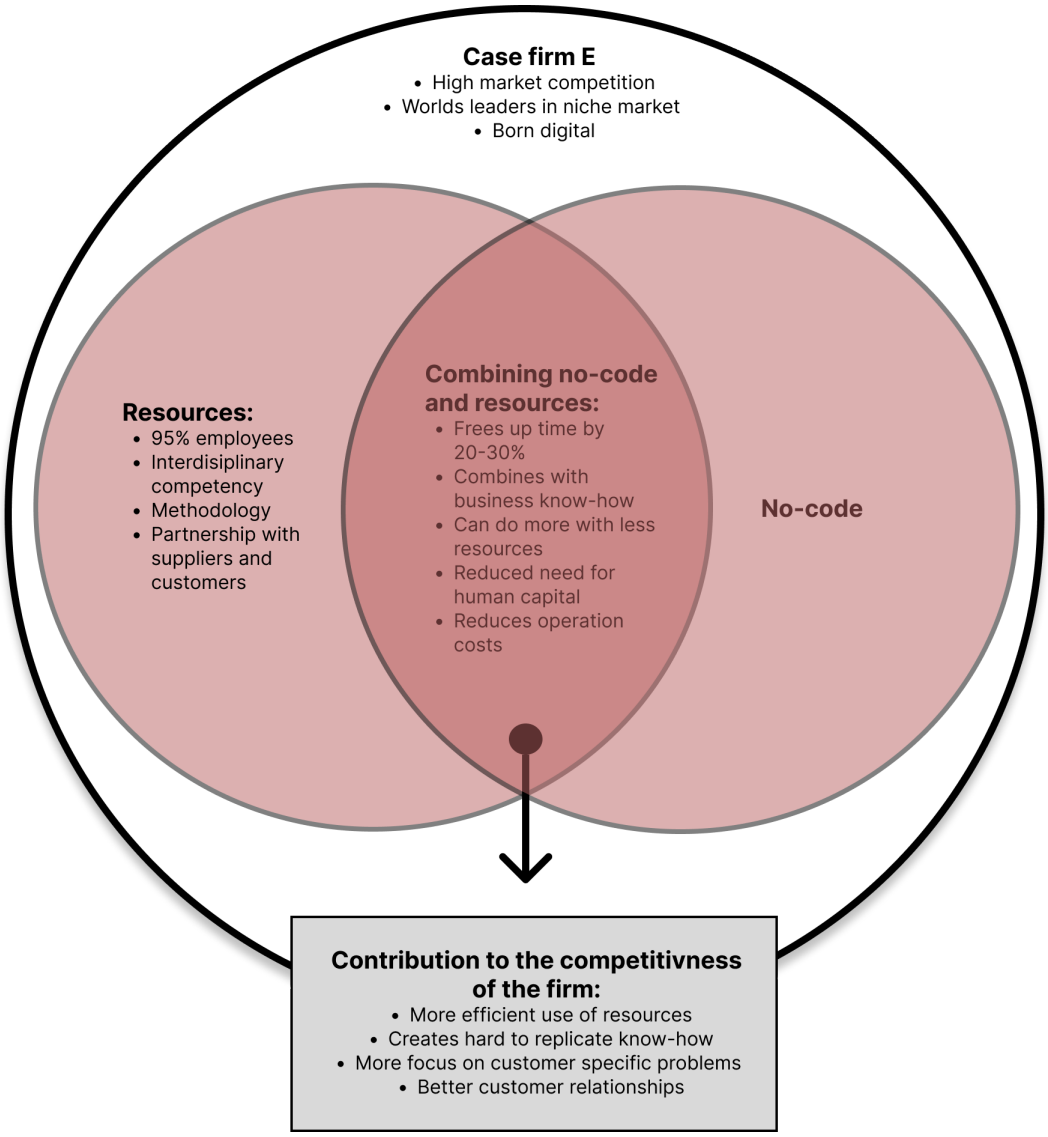


Figure 15: Within-case analysis Case firm E

5.2 CROSS-CASE ANALYSIS

The main objective of the cross-case analysis is to reach meaningful conclusions that go beyond the confines of the individual cases. By systematically comparing the multiple cases, researchers can uncover similarities and differences in the data material, which can contribute to theory development (Yin, 2018). It enables researchers to investigate various aspects, including context, processes, outcomes, and other influencing factors at play. The methodological approach for the cross-case analysis was described in subsection 4.3.2 and consisted of synthesizing A-, B- and C-categories, each with a higher level of abstraction and theoretical influence (Widding, 2006). The most prominent findings across the five cases will be laid forth hereafter.

5.2.1 *Functionality restrictions and complexity*

In accordance with existing literature on the topic of no-code, we find that the functionality restrictions of the NCDPs are one of the key inhibitors of utilizing no-code technology. This is highlighted in all five cases in our study. Both the literature and our findings indicate that there are challenges related to customization, limited development possibilities, and scalability. The cases in our study further elaborate on specific drawbacks and concerns.

The functionality restrictions often cause challenges in customizing user interfaces or services provided by vendors, limiting the degree of complexity obtainable when developing software within the framework (Al Alamin et al., 2021; Käss et al., 2022). This is reinforced by all the case firms, indicating that the scope of customization within no-code platforms is restricted to the framework provided by the vendor. They also emphasize the difficulty of working around the limitations when attempting to create complex functionalities not yet offered by the platform. Case firm B mentions that some degree of tailoring is provided, but it is still far less than what is possible with traditional development, hence the decision to not use no-code for their main software solution. In addition, the NCDPs face a challenge in supporting the limited experience of citizen developers, as well as meet the needs of more experienced developers (Käss et al., 2022). While supporting user-friendliness towards citizen developers, the lack of functionality required for complex applications can, in some cases, lead to frustration among experienced developers (Käss et al., 2022). The interviewees stress that when faced with problems beyond the capabilities of the platform, progress may come to a standstill, and they may have to seek alternative solu-

tions or even resort to traditional development. Case firm D, as an example, had to undergo a transition from no-code to traditional development due to a growing dissatisfaction among their developers regarding the limitations of the technology. Though it is important to mention that in addition to the limitation factor, a part of the reason for transitioning was also the developers passion for writing code, which was deprived by using no-code.

We also see in the literature and our findings that scalability is a significant inhibitor to the adoption of NCDPs (Käss et al., 2022). Our findings from Case firm C echo this, noting that no-code development is a "happy path" as long as it remains within the limitations of the chosen tools. However, once specific functions beyond the platform's capabilities are required, difficulties arise and development may come to a halt. The need to rely on platform vendors for updates and additional functionalities is also emphasized by Case firm D. Case firm E elaborates on this by adding that there is no single no-code provider that is sufficient to solve all your problems, meaning you might have to utilize several platforms to meet your functionality needs.

Comparing the literature findings with the insights from the case firms reveal a consistent picture regarding the functional restrictions of no-code platforms. The limitations of customization, functionality, scalability, and lack of flexibility are recurring themes in all cases. Both Käss et al. (2022), Al Alamin et al. (2021), and our findings indicate that no-code platforms excel in simpler applications, but challenges arise when dealing with complexity, unique requirements, and the need for extensive customization.

As NCDPs are a relatively new phenomenon, you are at all times restricted to the level of complexity that the vendor offers. The aforementioned restrictions are linked to the current state of NCDPs. As the field of no-code development is evolving, advancements may address some of these limitations in the future.

5.2.2 *Vendor lock-in*

Another critical inhibitor to the adaptation of NCDPs are the concerns regarding vendor lock-in and the effect it might have on the organization's operations. Several researchers (Luo et al., 2021; Käss et al., 2022; Rokis & Kirikova, 2022) highlight that the fear of lock-in to a specific vendor is a significant concern. The inability to access the generated source code limits the transferability of

applications to other NCDPs, making organizations highly dependent on the decisions and actions of the vendor. This finding aligns with the concerns expressed by Case Firm A, which emphasizes the fundamental operational risk associated with dependence on a specific no-code vendor.

Further, Käss et al. (2022) discuss the challenges in estimating the total cost of adopting NCDPs due to varying pricing schemes and factors such as user development, application deployment, and data storage. This finding is shared by Case Firm C, which recognizes the difficulty in estimating costs associated with using no-code platforms due to the lack of standardized pricing models. It emphasizes the need for a thorough risk analysis before choosing no-code to assess the permanence or temporariness of the solution. Getting locked into a vendor that might prove more expensive than first perceived, can have severe consequences for the firm. Though there might be a certain lock-in effect when choosing no-code, Case firm D also draws a parallel between no-code and traditional development, suggesting that vendor lock-in exists to some extent in both approaches.

Contradicting the concern regarding vendor lock-in, Case firm E has a different perspective on the perceived lock-in effect, and does not regard it as a critical drawback. They argue that the value is not the software itself, but the customer insight, design, and user experience, which is not tied to the specific no-code platform. For Case firm E, the ability to extract the code base is less important, because they prioritize the non-technical aspects of their solution. While the literature emphasizes the fear of lock-in inhibiting adoption, Case firm E perceives no-code as advantageous primarily in the initial phases of software development for rapid customer insight and validation. Once a satisfactory product is achieved, Case firm E suggests that traditional software development becomes a viable alternative, indicating a diminishing advantage of no-code platforms over time. In other words, they do not regard vendor lock-in as an issue due to their perspective on NCDPs as a go-to-market strategy, working as a blueprint for traditional development, rather than a long term solution.

5.2.3 *No-code effects intangible resources in the firm*

For a resource to be considered valuable, strategies enacted upon it must result in the firm's increased effectiveness or efficiency (Barney, 1991), or pose a substantial beneficial effect (Massey, 2016). All case firms except Case firm B are

utilizing no-code technology to create their main solution, whereas Case firm B develops internal solutions for customer success tasks. In order to evaluate how no-code affects the resources of firms, one needs to understand the resource bundles it interacts with.

Our study has examined the main resources from the perspectives of the case firms. As described in subsection 3.1.1, intangible resources refers to resources that have no physical form, such as assets and competencies (Hall, 1993). When studying the resources reported by the case firms, we see that they are predominantly intangible, with competency (employee know-how) being the most prevalent key resource. Building on the importance of employees as the most valuable resource is the fact that social relations are described as complementary resources that strengthen the value of employees. Such social resources are reported to be firm culture, methodology, ties to the entrepreneurial ecosystem, and partnerships with customers. Combined with the employees, this creates resources that are hard for competitors to imitate, making them more likely to become a source of SCA (Dierickx & Cool, 1989; Barney, 1991). When combined with no-code, our findings suggest a twofold effect on firm resources: (1) no-code increases resource optimization, and (2) no-code is less resource-intensive. The first suggestion regards doing more with the existing resources currently inherited by the firm, whereas the second contends that firms need to acquire fewer resources to make use of no-code.

Overall, in the majority of cases, firms have experienced no-code to optimize the use of existing firm resources. The main impact reported is the shortening of development tasks due to the technology's quick development environment. This, in turn, frees up human capital to focus on more customer-critical tasks. When the main key resource of the firm is the employees, the fact that no-code provides you with increased speed of development means that one can utilize the human capital better (Bock & Frank, 2021). This is contradictory to the findings of Buzatu et al. (2020), who found that implementation of digital tools oftentimes required the firms to acquire new technical competency or develop it themselves, which drives away the human capital from main operations. Case firm A also reports a higher degree of creativity being freed up because the time saved on generic development can be utilized to create new functionalities. Furthermore, Case firm C contends that their methodology of rapid development and testing is strengthened by the use of no-code, and that the technology is thus highly complementary to the way in which they provide their services. In this manner, no-code in fact interacts with the methodology of the firm.

The second effect of no-code on firm resources is said to be the resources needed to utilize it. In particular, the need for human capital is reported by all case firms to be lower than with traditional development. Case firm E explains that projects previously requiring 12 team members can be done using only 3 in no-code. For startup firms, Case firm C contends that the reduced need for human capital makes it easier to test and evaluate new business concepts, before considering putting more time and resources into them. Faulty concepts could therefore be discarded quickly, without a substantial waste of resources.

There is, however, a disagreement between the case firms as to the reduced need for financial resources when using no-code. As the literature on no-code often describes the technology as being less costly than traditional development methods (Käss et al., 2022), this may not be the case for all firm phases. An argument made for the reduced cost of no-code is the fact that paying salaries for developers can accumulate large expenses for a firm, and that no-code reduces the costs by also reducing the need for skilled developers (Käss et al., 2022; Rokis & Kirikova, 2022). For case firms A and B, the perceived cost of no-code is high. Startups lack the financial resources to pay a monthly licensing fee but may resort to a sweat-for-equity payment to developers in order to minimize direct spending. Case firm E is in agreement that no-code is in fact expensive for startup companies, but states that it all depends on the problem you wish to solve and what tools are needed for the job - with more powerful no-code tools being more costly. If the problem is fundamental to a firm's operation, the costs become peanuts (Case Firm E). In contrast, Case Firm D expresses that the cost of no-code itself is not high, but that the wrong implementation and usage can make it more expensive over time.

No-code seems to have increased the efficiency and resource utilization of the case firms, thus making it a valuable resource (Barney, 1991). As discussed in subsection 3.2.2, no-code is uniformly available for all firms to use and cannot be considered a source for SCA by itself. In order to become a source of SCA, no-code needs to be integrated and organized with the human capital, overall culture and methodology of the firm. Zahra (2021) argues that startups have the ability to implement new digital tools relatively seamlessly due to their restrictions on resources and even explains how such implementation can happen from the firm's very infancy - building the resources around the technology. In our study, this seems to hold true, with Case Firm E as an example of a firm building its human resources from the start, using no-code. Our findings

suggest that no-code interacts with the firm's intangible resources, with the main effect being on the utilization and reduced need for human capital.

5.2.4 *Technical competency is required to make full use of no-code*

Even though a lot of no-code vendors appraise no-code's ability to eliminate the need for highly skilled developers, our study finds that all case firms assert technical competency as a must-have resource in order to fully make use of the technology. This is not to say that no-code does not reduce the need for developers, as the case firms also highlight that the technology is in fact less technically demanding. Its ability to visualize the code and create a layer of abstraction makes it far easier and more comprehensible for citizen developers to use and learn (Käss et al., 2022). However, this does not fully eliminate the need for some technical competency amongst the firm's employees. Important competencies regarding application development, such as architecture, structure, cyber security, algorithmic logic, and layers of application development, are still very much needed in order to minimize the accumulation of technical debt. Our findings are in agreement with previous research conducted on no-code technology, stating that the ready-to-use components provide you with a less technically demanding way of developing (Luo et al., 2021; Lethbridge, 2021).

According to Lethbridge (2021), it is common for such citizen developers to write far more complex code than what the no-code platforms are actually intended for. Without an understanding of proper software architecture, the result is what he calls "mountains of technical debt", where the code becomes exceptionally hard to understand, maintain, and modify. Furthermore, an intimate understanding of the practical limitations and underlying logic is a prerequisite for cyber security (Hurlburt, 2021). Even though no-code vendors have some built-in security in their platforms (case firms A and E), the firm must still acquire the needed technical competency to prevent algorithmic bias or false data (Hurlburt, 2021). This specialization still falls within the realm of IT professionals (Hurlburt, 2021; Lethbridge, 2021).

Whereas it seems to hold true that no-code development platforms can easily be adopted by citizen developers to create functional applications, we argue that a firm must in fact have some technical competency in its resource bundle to fully make it a competitive advantage. One can also argue based on the findings of

this study that no-code, when used without adequate technical competency, can contribute to the accumulation of technical debt, in turn making it more of a competitive disadvantage (Barney, 1995). For firms wanting to make use of the technology, the need for technical competency is still not fully eliminated.

5.2.5 *No-code enables faster time to market*

A firm's agility is its ability to respond swiftly to changes in the environment and, for digital businesses, to effectively develop service offerings in alignment with customer needs (Xu & Koivumäki, 2019). Firms utilizing no-code experience rapid development as a strong suit provided by the technology. Previous research has found that this rapid development in turn allows firms to reach the desired customers faster with their solutions and rapidly bring applications to market (Louw & Nieuwenhuizen, 2020; Luo et al., 2021; Pfister & Lehmann, 2021). Our findings are in alignment with previous research, as all case firms assign no-code with superior speed of development over traditional methods. This in turn makes the iteration and testing of new functionalities faster and increases the firm's agility in the market (Olsson, Bosch, & Alahyari, 2013; Tohãnean et al., 2020), where functions that do not create additional value for the customer can be easily scrapped or changed. The development cycles decrease as a result, making the time to market shorter (Xu & Koivumäki, 2019; Käss et al., 2022). Overall, the increased agility of the firm in turn allows for a faster reaction to changing markets, strengthening the firm's dynamic capabilities (Teece, 2023).

Another way in which development cycles and, subsequently, time to market decrease is through the interaction of human capital. Käss et al. (2022) explained that the use of no-code enables a common language between the IT-department and business. When working in cross-functional teams, there is a risk of miscommunication and misunderstanding between departments regarding functionality requirements, which in turn can result in slower progress and lower quality of the software. Because no-code is easily comprehensible for citizen developers, it enables the business competency of the team to take a deeper involvement in the development process. As discussed in subsection 5.2.3, no-code also reduces the human capital involved in the projects, further decreasing the risk of miscommunication.

Case firm E provides a good example of this, where the main employees have business backgrounds. By educating business people in the use of no-code

tools, they have created a situation where fewer employees are involved in each project. In addition, the hybrid competency creates a more wholesome understanding of the entire process, from business requirements to fully developed functionalities. In effect, the business people become developers, which also creates a competency that is hard to replicate. The development cycles are thus greatly reduced, because business requirements can be swiftly translated into functionalities without imposing the risk of miscommunication across larger cross functional teams. By utilizing no-code technology, Case firm E has effectively increased the firm's responsiveness and agility.

Similarities can also be found for Case firm A, Case firm C, and Case firm D, who all remark that the increased speed of development enabled them to reach the market faster with their solutions. One can rapidly create an MVP for which potential customers can test and provide feedback. The feedback is then quickly converted to new functionalities, which again are user-tested shortly after, making the iteration process faster (Xu & Koivumäki, 2019; Louw & Nieuwenhuizen, 2020).

5.2.6 *Close customer relationships creates hard to replicate resources*

Building on the principle of faster time to market established in the previous section, the way in which this can be converted into a SCA still remains. Little is written about the way in which rapid development actually contributes to the competitiveness of the firm. However, during this study, a possible contribution may have been illuminated.

When creating a new market offering, firms aim to create solutions that solve a problem for the customer. In order to create solutions of value, the firm needs to understand these problems thoroughly, and customer insight is therefore a prerequisite for product-market fit. Through the rapid development of MVPs in no-code, firms are able to test hypotheses regarding functionalities and receive feedback from customers (Edison, Wang, Jabangwe, & Abrahamsson, 2018). The reduced development cycles enable consistent delivery in shorter time spans and create close relationships with customers (Pfister & Lehmann, 2021). This in turn creates a situation in which the firm gains substantial customer insights faster than their competitors. According to Case firm D, this is what creates value for the firm. No-code thus functions as a means to gain customer insight through shorter development cycles. You are able to quickly convert customer

feedback into functionalities and constantly deliver updated versions of the software, building a unique trust.

This is also backed up by the statements of Case firm A, which contends that the firm can stay ahead of competitors that seek to blueprint their solution. Because the firm rapidly gains more customer insight, competitors will lag behind when new features are pushed. To date, customer relationships are the most valuable resource in their portfolio, regarded as a result of using no-code technology. For Case firm B, which does not build its main solution in no-code, the same can seem to be true. The administrative resources and time freed up by no-code are used on internal processes regarding customer iteration tasks. The follow-up with customers becomes more streamlined, enabling them to retain relationships better. Case firm E also states that the increased speed and ease of development contribute to the firm using more resources to solve problems specific to each customer. The problems are therefore gaining more focus, and the right solutions are being built as a result.

With a customer-centric development process, in combination with speed of development, we argue that the use of no-code technology functions as an enabler to build closer relationships with customers. These relationships are hard to replicate for competitors, because they are causally ambiguous and socially complex (Barney, 1991). Based on our findings, the relationship with customers is what may contribute to a SCA when using no-code technology.

5.3 ANSWER TO RESEARCH QUESTIONS

5.3.1 *"How does no-code technology interact with a firm's resources to create a competitive advantage?"*

The findings show that no-code has several ways of creating value for a firm as part of a resource bundle. Firstly, the use of no-code is less resource-intensive in terms of the human capital needed to create functional applications, with a lower threshold for citizen developers to take part in the development process. Our findings, however, have uncovered that no-code technology still does not eliminate the need for skilled developers to take part in the process. Especially competency regarding the architecture, cyber security, and algorithmical logic of software development is needed in order to minimize the accumulation of technical debt. Lack of sufficient technical competency may, in fact, put a firm

at a competitive disadvantage with software of low quality and performance. Some disagreement in the findings also concerns the resource intensity of costs and the financial burden, especially for startup firms.

Secondly, there is the effect of no-code on resource optimization, where firms uniformly experience increased utilization of human capital and time. The effect is linked to the speed of development, which frees up time, creativity, and administrative resources to do other tasks using the existing human capital. No-code thus seems to be a valuable resource due to its effect on resource optimization and reduced development cycles. However, because no-code is off-the-shelf software that is evenly distributed across all market participants, it needs to be an integrated part of the firm's overall culture, methodology and structure in order to create lasting competitive advantages.

Furthermore, the real contribution to a firm's SCA is, according to our findings, its enabling effect on creating close customer relationships. When used properly, the benefits of decreased development cycles are found to increase a firm's ability to respond to customer needs and convert them rapidly into new functionalities through iterations and user testing. This increase of firm agility, in combination with better resource allocation to focus on customer needs, is the underlying enabler for good customer relationships. The customer relationships further enable better and faster customer insight, effectively creating a new resource that is hard to replicate, because of its causal ambiguity (Barney, 1995). Customer relationships are also a valuable firm resource in and of themselves due to their social complexity (Barney, 1995).

5.3.2 *What are the key considerations and evaluation criteria that early stage firms should employ when considering to adopt no-code technology?*

Even though no-code seems to have a substantially beneficial effect, our findings suggest that there are several underlying considerations that early stage firms should examine before potentially choosing it for their software development. A thorough evaluation process can help assess the feasibility, suitability, and potential risks associated with adopting no-code technology. Our study does not propose a clear structure for the evaluation process, but rather highlights the aspects that should be considered.

The first key aspects that should be considered are the functionality restrictions

and obtainable complexity with the use of no-code. The specific requirements and complexity of the intended software needs to be assessed against the capabilities of the chosen tools and vendors. Furthermore, the limitations of no-code platforms in terms of customization and development possibilities should therefore be part of the evaluation. Only then is it possible to assess whether the desired functionality can be achieved within the framework provided by the vendor or if complex functionalities beyond the platform's capabilities are required. According to our findings, no-code platforms currently seem to be best suited for smaller and simpler applications, although this might change as the platforms evolve. Still, the challenges of relying on platform vendors for updates and additional functionalities might be hindering, and could potentially halt the progress as projects increase in complexity. If you see no-code merely as a means for customer insight, validation, and a go-to-market strategy, scalability does not necessarily need to be a part of the evaluation.

Another aspect to assess is the potential risks and consequences of vendor lock-in. Exploring different NCDP vendors available on the market and evaluating their reputation, customer reviews, support options, and track record of delivering updates and new features can be a good starting point. Furthermore, evaluating the ability to access and transfer the generated source code to other no-code platforms might be important to avoid a lock-in effect. According to our findings, there are certain operational risks associated with being dependent on the decisions and actions of a specific vendor. Such risks include, but are not limited to, system downtime, vendor stability, data security, and compliance with regulatory requirements. Overall, the impact of these risks should be considered prior to adoption.

Additionally, the total cost of ownership utilizing no-code platforms needs to be addressed. Factors such as licensing fees, additional functionalities or plugins, training costs, ongoing support expenses, and scaling costs should be taken into account. Furthermore, as the need for skilled developers is not fully eliminated, one should evaluate the technical competencies required for tasks such as code architecture and structure, cyber security, and algorithmic logic. Therefore, assess the availability of training resources, community support, and the expertise needed within the team to effectively utilize the chosen platform. Our findings also suggest that teams consisting of traditional developers should pay extra attention to the possibility of developer dissatisfaction.

6

DISCUSSION

The literature review conducted prior to this study uncovered a distinct rise in publications in the field of no-code technology in recent years (Balcon et al., 2022). Several papers highlight the perceived attributes of no-code and connect them to competitive advantage, none of which doing so by utilizing the framework of RBT. Furthermore, there were shortcomings regarding *how* no-code affects firm resources, and *if* the technology could be incorporated create a lasting competitive advantage. Below, we address our contribution to no-code technology as a research field by bridging it with the RBT. The findings are discussed in light of *Engaged Scholarship Design* by Mathiassen (2017), introduced in chapter 1 of this thesis. Thereby, we specifically discuss our contribution to research - the literature (C_A), and our contribution to practice - the real world problem (C_P).

6.1 THE CONTRIBUTION OF KEY FINDINGS TO PREVIOUS LITERATURE (C_A)

Through our research, we were able to confirm and expand upon findings from existing literature that shed light on the benefits and limitations of no-code technology, as well as its effects on firm resources and competitiveness. Previous studies address the efficiency of no-code platforms in facilitating rapid application development and deployment, which aligns with our findings. Specifically, we expand the understanding of no-code by examining it in a firm context.

Our literature review (Balcon et al., 2022) concluded that no-code could be a valuable firm resource, but that the competitiveness is determined by its interactions as part of a resource bundle. We believe to have found compelling evidence in this study of how the technology in fact can lead to a SCA. This becomes a key contribution to previous research by addressing the call by Käss

et al. (2022) for further investigation into the social and organizational impact of no-code technology. By exploring the relationship between no-code technology and firm resources, we demonstrate how no-code technology interacts with other resources as part of a bundle. The main effect being optimization of the firm's intangible resources such as human capital and time.

The ability of no-code platforms to enable closer partnerships with customers through rapid iterations and responsiveness enhances the firm's competitive advantage. These insights contribute to the resource literature by also emphasizing the role of customer relationships as a valuable and hard-to-replicate resource. By also providing insight into the different aspects for which firms should evaluate the applicability of no-code, we believe to have contributed to no-code literature with practical guidance for organizations considering its implementation.

Lastly, we might also have been contributing to Helfat et al.'s (2023) call for more work regarding the impact of digitalization on resource literature. Specifically, as our study looked at the effects of a novel technology on firm resources in relation to competitiveness. During this work we experience the RBT to be an invaluable framework, providing us with the ability to understand important relations and contexts. We also believe that RBT and the dynamic capabilities framework will be more intertwined in the years to come, as the age of digitization rapidly changes business environments.

6.2 THE CONTRIBUTION TO PRACTICE (C_P)

Looking into our contribution to practice (C_P), our thesis provides practical guidance for early stage firms considering the adoption of no-code technology. Our observation of a real-world problem in the entrepreneurial ecosystem at NSE, regarded such firms' inability to evaluate and make use of no-code. This study has contributed to this real-world problem by extension of the contribution towards the literature.

Our findings have illuminated important factors that should be included in the evaluation process for early stage firms, concerning the technologies applicability, suitability and risks. Especially the limitations regarding functionality restrictions may pose significant restraints, and followingly should the complexity of the intended software be discussed. Vendor lock-in might also impose

operational risks, due to the inability to extract the underlying code to other development environments. Combined with the functionality restrictions, this might lead to long-term complications as the firms are dependent on vendors to further develop their NCDP.

Furthermore, we highlight the significance of integrating technical competency within the resource bundle surrounding no-code adoption. By also recognizing the need for some technical expertise to avoid the accumulation of technical debt and ensure the development of high-quality software, we provide practical recommendations for firms seeking to leverage no-code effectively and circumvent a situation of competitive disadvantage. This, however, is also dependent on the technical competency already existing in the firm, as no-code might lead to developers dissatisfaction.

Additionally, we add to the importance of considering how no-code technology impacts firm resources, such as employee know-how and culture. This highlights the need for alignment with the overall development methodology and the evaluation of potential effects on existing resources, thereby introducing novel insights on practical implications. As a result, we believe that our study offers actionable recommendations for early stage firms considering the adoption of no-code platforms in the future. To summarize, our thesis contributes to the literature (C_A) by (1) examining important factors influencing the feasibility and suitability of choosing no-code, (2) its effect on firm resources and (3) the subsequent effect on firm competitiveness. With this contribution we have made a fertile ground for further research on the gap between no-code and resource literature. Additionally, our research contributes to practice (C_P) by offering guidance on strategic integration, providing the means for a better evaluation process. These contributions enhance the understanding of no-code technology and its implications for organizations, facilitating informed decision-making and effective utilization of the technology.

7

CONCLUSION

In this master's thesis we looked into how no-code, in combination with other resources, affects a firm's ability to compete. Our study's findings demonstrated how closely related no-code technology is to other intangible resources, and how selecting a technology reflects how competitive a resource is.

Our findings indicate that no-code itself is a valuable resource in its ability to bring effectiveness and resource optimization to a firm. For early stage firms, the technology is closely integrated with intangible resources. Specifically, the reduced need for human capital involved in a project, as well as the freeing of time due to the rapid development abilities the technology provides, have substantial positive effects. Our study, however, discovered that in order to make optimal use of no-code technology, the firm needs some technical competency to avoid the accumulation of technical debt during development. Lack of such competency might even put a firm at a competitive disadvantage as a result of low quality software, leading to solutions that are slow, hard to maintain, and difficult to understand. We therefore suggest that a resource bundle around no-code should include a certain level of technical competency.

The agility of the firm also seems to be greatly enhanced by the use of no-code. Reduced development cycles contribute to firms gaining customer insight faster, which is considered a valuable resource in and of itself. When faced with new and conflicting market information, firms utilizing no-code can swiftly respond by creating, or scrapping, functionalities that align with the changes. The totality of the aforementioned effects brings about another unique and hard to replicate resource which is close customer relationships. We believe that this is how no-code can become a source of SCA, because relations amongst customers are valuable, unique, socially complex, and organized (when customer insight is effectively integrated into new functionalities).

The findings of our study also suggest that companies often adopt no-code technology for software development without a thorough evaluation process. The decision seems to be based more on perceived benefits than on structured assessments. However, we emphasize the importance of evaluating the feasibility, suitability, and potential risks associated with using no-code before making a decision. One crucial aspect to consider is the functionality restrictions and complexity achievable with no-code platforms, as they may not be suitable for complex applications. Evaluating vendor reputation, support options, and the ability to transfer source code should also be done to mitigate the risks of vendor lock-in. Assessing the potential risks, total cost of ownership, and technical competencies required is essential. Furthermore, the impact of no-code on the firm's resources, such as employee know-how and culture, should be considered. It is important to ensure that the use of no-code aligns with the overall methodology and does not lead to dissatisfaction among developers.

The contribution extends the research on no-code technology and its effects on firm resources that can lead to a SCA, in addition to examining the criteria by which firms should evaluate the technology prior to adoption. By contributing to the literature, we also contribute to the real-world problem identified in the entrepreneurial ecosystem at NTNU, in which firms have had a hard time evaluating no-code strategically. We hope that this study can be of value to firms when entering a new entrepreneurial venture by illuminating how they can go about implementing no-code effectively. To the best of our knowledge, our study is the first to explicitly investigate the actual impact of no-code on competitive advantage.

8

FURTHER RESEARCH AND IMPLICATIONS

Our research into the effects of no-code technology on a firm's resources and competitive advantage has yielded valuable insights. However, it has also revealed limitations that pave the way for future research into unexplored areas of the field. One of our studies' key limitations is its limited geographical span, as we only look at Norwegian firms. Future research could expand the investigation to include a wider range of geographic locations and investigate how cultural and institutional factors may influence the adoption and impact of no-code technology in various contexts.

Depending on the particular tools or platforms used and the industries in which they are implemented, the impact of no-code technology can vary significantly. In-depth examinations of the impact of various NCDPs on different industries could yield granular insights into best practices and strategic considerations for adopting no-code.

While our research primarily focused on startups and scaleups, no-code technology may also benefit large enterprises. Further research could investigate how large companies are utilizing NCDPs, identify the challenges they face, and investigate the potential benefits of adopting this technology. Case studies involving large enterprises or cross-sectional studies comparing the adoption and use of no-code technology by small and large firms could provide valuable insights. In addition, examining the interaction between no-code and traditional coding in larger corporations could reveal how these methodologies can coexist and/or complement one another in a hybrid development environment. Future research should investigate the dynamics of this hybrid paradigm and its influence on resources and competitive advantage.

During our interviews with the case firms, the interviewees also brought up

several research-worthy topics. One of the topics that arose was conducting research that spans across the entire value chain. This could involve interviewing customers of companies that use no-code development to gain insight into their experience with the development process. We can gain a comprehensive understanding of the value chain and identify areas for improvement and optimization by also examining the customer's perspective. Another emerging topic was the exploration of building exit cases utilizing no-code technology. Understanding how the adoption of no-code affects exit strategies and whether it is an obstacle or a driver can be a topic worthy of further study.

While our study has laid the foundation for understanding the impact of no-code on firm resources and competitive advantage, it also serves as a starting point for future research in the cross-section between no-code technology and competitive advantage. Future studies can advance our understanding of the potential of no-code technology and its implications for firms by addressing the identified limitations, exploring new research avenues, and taking into account the perspectives of both firms and customers.

BIBLIOGRAPHY

Al Alamin, M. A., Malakar, S., Uddin, G., Afroz, S., Haider, T. B., & Iqbal, A. (2021). An empirical study of developer discussions on low-code software development challenges. In *2021 ieee/acm 18th international conference on mining software repositories (msr)* (pp. 46–57).

Allred, C. R., Fawcett, S. E., Wallin, C., & Magnan, G. M. (2011). A dynamic collaboration capability as a source of competitive advantage. *Decision sciences*, *42*(1), 129–161.

Alvarez, S. A., & Busenitz, L. W. (2001). The entrepreneurship of resource-based theory. *Journal of management*, *27*(6), 755–775.

Armstrong, C. E., & Shimizu, K. (2007). A review of approaches to empirical research on the resource-based view of the firm. *Journal of management*, *33*(6), 959–986.

Balcon, M., Bekke, S. A., & Kaland, S. B. (2022). What does the literature say about startups using no-code as a resource for competitive advantage? Retrieved from https://drive.google.com/file/d/1mp1kPS4yntzW0FXnTjNJIQkIbFJw1xie/view?usp=sharing

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

Barney, J. (1995). Looking inside for competitive advantage. *Academy of Management Perspectives*, *9*(4), 49–61.

Beard, J. W., & Sumner, M. (2004). Seeking strategic advantage in the post-net era: viewing erp systems from the resource-based perspective. *The Journal of Strategic Information Systems*, *13*(2), 129–150.

Bock, A. C., & Frank, U. (2021). Low-code platform. *Business & Information Systems Engineering*, *63*(6), 733–740.

Bryman, A. (2016). *Social research methods*. Oxford university press.

Bryman, A., & Bell, E. (2015). Business research methods (vol. 4th). *Glasgow: Bell & Bain Ltd*.

Buzatu, A. I., Dinu, A. C., Costache, C. I., & Tohănean, D. (2020). Innovative entrepreneurial companies in the digital era: The impact of business automation on international development and competitive advantage. In *European conference on innovation and entrepreneurship* (pp. 750–XV).

Creswell, J. W., & Poth, C. N. (2016). *Qualitative inquiry and research design: Choosing among five approaches*. Sage publications.

Cuthbert, C., & Pearse, N. (2021). Strategic use of low code platforms. *Proceedings of the 15th International Multi-Conference on Society, Cybernetics and Informatics*.

Dierickx, I., & Cool, K. (1989). Asset stock accumulation and the sustainability of competitive advantage: Reply. *Management Science*, *35*(12), 1514–1514.

Edison, H., Wang, X., Jabangwe, R., & Abrahamsson, P. (2018). Innovation initiatives in large software companies: a systematic mapping study. *Information and Software Technology*, *95*, 1–14.

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

Eisenhardt, K. M., & Martin, J. A. (2000). Dynamic capabilities: what are they? *Strategic management journal*, *21*(10-11), 1105–1121.

Flick, U. (2015). *Introducing research methodology: A beginner's guide to doing a research project*. Sage.

Gomes, P. M., & Brito, M. A. (2022). Low-code development platforms: A descriptive study. In *2022 17th iberian conference on information systems and technologies (cisti)* (pp. 1–4).

Hall, R. (1993). A framework linking intangible resources and capabiliites to sustainable competitive advantage. *Strategic management journal*, *14*(8), 607–618.

Halldórsson, Á., & Aastrup, J. (2003). Quality criteria for qualitative inquiries in logistics. *European journal of operational research*, *144*(2), 321–332.

Helfat, C. E., Kaul, A., Ketchen Jr, D. J., Barney, J. B., Chatain, O., & Singh, H. (2023). Renewing the resource-based view: New contexts, new concepts, and new methods. *Strategic Management Journal*.

Hurlburt, G. F. (2021). Low-code, no-code, what's under the hood? *IT Professional*, *23*(6), 4-7. doi: 10.1109/MITP.2021.3123415

Krumsvik, R. J. (2014). Forskingsdesign og kvalitativ metode: ei innføring.

Kvale, S. (1996). Interviews: an introduction to qualitive research interviewing: Sage.

Käss, S., Strahringer, S., & Westner, M. (2022, June). Drivers and inhibitors of low code development platform adoption.. doi: 10.1109/CBI54897.2022.00028

Lethbridge, T. C. (2021). Low-code is often high-code, so we must design low-code platforms to enable proper software engineering. In *International symposium on leveraging applications of formal methods* (pp. 202–212).

Li, Z., Avgeriou, P., & Liang, P. (2015). A systematic mapping study on technical debt and its management. *Journal of Systems and Software*, *101*, 193–220.

Lincoln, Y. S., & Guba, E. G. (1985). *Naturalistic inquiry*. sage.

Lippman, S. A., & Rumelt, R. P. (1982). Uncertain imitability: An analysis of interfirm differences in efficiency under competition. *The bell journal of Economics*, 418–438.

Louw, C., & Nieuwenhuizen, C. (2020). Digitalization strategies for smes: A cost vs. skill approach for website development. *African Journal of Science, Technology, Innovation and Development*, *12*(2), 195–202.

Luo, Y., Liang, P., Wang, C., Shahin, M., & Zhan, J. (2021). Characteristics and challenges of low-code development: The practitioners' perspective. In *Proceedings of the 15th acm/ieee international symposium on empirical software engineering and measurement (esem)* (pp. 1–11).

Major, C. H., & Savin-Baden, M. (2010). *An introduction to qualitative research synthesis: Managing the information explosion in social science research*. Routledge.

Massey, B. L. (2016). Resource-based analysis of the survival of independent web-native news ventures. *Journalism & Mass Communication Quarterly*, *93*(4), 770–788.

Mathiassen, L. (2017). Designing engaged scholarship: From real-world problems to research publications. *Engaged Management Review*, *1*(1), 2.

Miller, D., & Shamsie, J. (1996). The resource-based view of the firm in two environments: The hollywood film studios from 1936 to 1965. *Academy of management journal*, *39*(3), 519–543.

Olsson, H. H., Bosch, J., & Alahyari, H. (2013). Towards r&d as innovation experiment systems: A framework for moving beyond agile software development. In *Proceedings of the iasted* (pp. 798–805).

Peteraf, M. A., & Barney, J. B. (2003). Unraveling the resource-based tangle. *Managerial and decision economics*, *24*(4), 309–323.

Pfister, P., & Lehmann, C. (2021). Returns on digitisation in smes—a systematic literature review. *Journal of Small Business & Entrepreneurship*, 1–25.

Porter, M. (1980). Competitive strategy, new york free press. *PorterCompetitive Strategy1980*.

Porter, M. E. (1985). *Competitive advantage: Creating and sustaining superior performance*. New York and London: Free Press.

Porter, M. E. (1991). Towards a dynamic theory of strategy. *Strategic management journal*, *12*(S2), 95–117.

Prahalad, C. K., & Hamel, G. (1990). The core competence of the corporation. *Harvard business review*, *68*(3), 79–91.

Reed, M. (2005). Reflections on the 'realist turn'in organization and management studies. *Journal of Management studies*, *42*(8), 1621–1644.

Rokis, K., & Kirikova, M. (2022, September). Challenges of low-code/no-code software development: A literature review. In (p. 3-17). doi: 10.1007/978-3-031-16947-2_1

Rosin, A. F., Proksch, D., Stubner, S., & Pinkwart, A. (2020). Digital new

ventures: Assessing the benefits of digitalization in entrepreneurship. *Journal of Small Business Strategy*, *30*(2), 59–71.

Sætre, A. S., & Van de Ven, A. (2021). Generating theory by abduction. *Academy of Management Review*, *46*(4), 684–701.

Saunders, M., Lewis, P., Thornhill, A., & Bristow, A. (2019, 03). "research methods for business students" chapter 4: Understanding research philosophy and approaches to theory development. In (p. 128-171).

Schriber, S., & Löwstedt, J. (2015). Tangible resources and the development of organizational capabilities. *Scandinavian Journal of Management*, *31*(1), 54–68.

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

Teece, D. J. (2023). The evolution of the dynamic capabilities framework. In *Artificiality and sustainability in entrepreneurship* (pp. 113–129). Springer.

Teece, D. J., Pisano, G., & Shuen, A. (1997). (1997):"dynamic capabilities and strategic management.". *Resources, Firms, and Strategies: A Reader in the Resource-Based Perspective*, 268.

Tohãnean, D., Buzatu, A. I., Baba, C.-A., & Georgescu, B. (2020). Business model innovation through the use of digital technologies: Managing risks and creating sustainability. *Amfiteatru Economic*, *22*(55), 758–774.

Wang, Y., Feng, Y., Zhang, M., & Sun, P. (2021). The necessity of low-code engineering for industrial software development: A case study and reflections. In *2021 ieee international symposium on software reliability engineering workshops (issrew)* (pp. 415–420).

Wernerfelt, B. (1984). A resource-based view of the firm. *Strategic management journal*, *5*(2), 171–180.

Wernerfelt, B. (1989). From critical resources to corporate strategy. *Journal of General Management*, *14*(3), 4-12. Retrieved from https://doi.org/10.1177/030630708901400301 doi: 10.1177/030630708901400301

Widding, L. (2006). Teorigenerering basert på case-intervjuer-analysemetode inspirert av grounded theory. *Bodø: HHB*.

Xu, Y., & Koivumäki, T. (2019). Digital business model effectuation: An agile approach. *Computers in Human Behavior*, *95*, 307–314.

Yin, R. K. (1981). The case study crisis: Some answers. *Administrative science quarterly*, *26*(1), 58–65.

Yin, R. K. (2018). *Case study research and applications: Design and methods*. Sage Books.

Yin, R. K., et al. (2003). Design and methods. *Case study research*, *3*(9.2), 84.

Zahra, S. A. (2021). The resource-based view, resourcefulness, and resource management in startup firms: A proposed research agenda. *Journal of Management*, *47*(7), 1841–1860.

Zaltman, G., Pinson, C. R., & Angelmar, R. (1973). *Metatheory and consumer research*. Holt, Reinhart and Winston.

Part II

APPENDIX

A

INTERVIEW GUIDE

Vi er et team på 3 masterstudenter fra NTNUs Entreprenørskole, som holder på å skrive en masteroppgave rundt no-code teknologi i programvareutvikling. Vi skal hovedsakelig finne ut av to ting: hvilke underliggende prosesser som er bak avgjørelsen om å bruke no-code i forhold til tradisjonell utvikling, og hvilke ressurser et selskap må ha for å kunne ta i bruk no-code som et konkurransefortrinn.

Vi vil også informere deg om at intervjuet blir tatt opp. Selskapet ditt og deg vil bli anonymisert, men blir beskrevet ved hjelp av attributter.

Introduksjon

1. Hva er din stilling og hvor lenge har du jobbet i *selskapet*?
2. Hva slags utdanning og/eller erfaring har du fra før av?

Vi har lest litt forskning som tyder på at det kan finnes forskjeller mellom kjønn og aldersgrupper i forhold til implementering av ny teknologi. Kan du fortelle hvor gammel du er, og hvilket kjønn du identifiserer deg som?

Selskapet

1. Hvordan vil du kategorisere *selskapet* i form av størrelse? (eks. startup, scaleup, SMB, konsern etc.)
2. Hvordan vil du beskrive *selskapet* når det kommer til innovasjon og digitalisering?
    a. Hvordan da?

No-code/low-code

1. Har du hørt om no-code/low-code før? I såfall hva?
2. Hva kjenner du til av no-code/low-code verktøy fra før av?
3. Hvilke fordeler mener du er ved no-code/low-code?
4. Hvilke ulemper mener du er ved no-code/low-code?
5. Bruker *selskapet* no-code/low-code verktøy?

Hvis ja:

- Hvorfor bruker dere no-code/low-code verktøy?
- Hvilke(t) no-code/low-code verktøy bruker dere?
- Hvorfor valgte dere akkurat dette/disse?
- Hva bruker dere no-code/low-code til?
- Hvilke bruksområde har no-code/low-code for dere sammenliknet med tradisjonell utvikling?

Hvis nei:

- Hvorfor ikke?
- Har dere vurdert å bruke no-code/low-code?
- Hva er det som hindrer dere fra å bruke no-code/low-code?
- Hvorfor foretrekker dere tradisjonell utvikling sammenliknet med no-code/low-code?

  a. *Mulig oppfølgingsspørsmål:* Forskning tyder på at no-code/low-code kan være egnet til rask utvikling av MVP for tidlig markedstesting. Hva tror du om det?

Ressurser og konkurransefortrinn

1. Hvordan ser konkurransebildet deres ut? Har dere mange konkurrenter/få konkurrenter?
2. Hvordan vil du beskrive *selskapet* sin konkurransekraft sammenliknet med konkurrentene/substituttene?
3. Anser du *selskapet* for å ha et konkurransefortrinn i markedet?
    a. Hvorfor/hvorfor ikke?
4. Hva vil du si er kjerneressursene til *selskapet*?
5. Har dere gjort en vurdering av kjerneressursene deres i valg av no-code/low-code kontra tradisjonell utvikling?
6. Er det noe dere har av ressurser som ikke konkurrentene har?
7. Er det noe dere gjør med ressursene deres som ikke konkurrentene gjør?
8. Er konkurransekraften deres knyttet til hvordan dere bruker ressursene deres? Hvordan?
9. Er det noen ressurser du synes man burde ha for å dra nytte av no-code/low-code?
10. Hva tror du dere har gjort/kan gjøre for å få et konkurransefortrinn gjennom bruk av no-code/low-code?
11. Er det noen ressurser hos dere som du tror blir forsterket eller frigjort ved bruk av no-code/low-code verktøy?

Attributter ved no-code/low-code
Hva tror du om bruk av no-code/low-code når det kommer til:

- Hastighet
- Pris
- Ressurser
- Fleksbilitet
- Tilpasninger (skreddersøm)

- Sikkerhet
- Kompetanse

Er det noen flere attributter ved no-code/low-code du føler mangler som jeg ikke har nevnt?

Vendor lock-in
Det er veldig lite forskning på no-code/low-code, men den forskningen som er fremhever et negativt attributt med no-code/low-code som kalles for "vendor lock-in". I prinsippet så går dette ut på at hver enkelt no-code/low-code utviklingsplattform lar deg bygge programvare gjennom utviklingsplattformen sine helt egne komponenter i et drag & drop interface. Hvis man ønsker å gå over til tradisjonell utvikling eller en annen utviklingsplattform så er det minimalt man kan ta med seg videre fra plattformen, altså at man blir låst til leverandøren.

1. Hva tenker du om "vendor lock-in" i forhold til bruk av no-code/low-code teknologi for deres del?
2. Har dere tenkt på dette med vendor lock-in i avgjørelsen om no-code/low-code kontra tradisjonell utvikling?

Scenario

1. Hvordan ville du reagert om det ble bestemt at hovedvekten av utviklingen i *selskapet* skal skje gjennom no-code/low-code verktøy fremover?
2. Hvilken omstilling måtte dere evt. gjort for å tilpasse seg til dette?
3. Hvordan ville dere agert dersom deres hovedkonkurrenter begynte å bruke no-code/low-code?

Avsluttende spørsmål

- Har du noe du ønsker å legge til i forbindelse med temaene vi har snakket om?
- Er det noe du synes vi burde ha spurt deg om?
- Er det noe du synes vi burde vært tydeligere på eller gjøre annerledes til neste intervju vi skal ha?

B

FOUNDATION FOR CROSS-CASE ANALYSIS

C1 Intangible resources

- B1 Competency is the most valuable resource
  - A1(1,2,3,4,5) Employees (5/5)
  - A2 (1,2,3,4,5) Technical Competency (5/5)
  - A3 (4) Industry competency (1/5)
  - A4 (5) Business competency (1/5)
- B2 Balance sheet resources
  - A5 (4) We have substantial financial resources (1/5)
  - A6 (1, 2) As a startup we lack financial resources (2/5)
- B3 Social relations are important complementary resources
  - A7 (1, 3, 4) Culture (3/5)
  - A8 (3, 4) Methodology (2/5)
  - A9 (1, 2, 4, 5) Partnership with customers (4/5)
  - A10 (1) Ties to entrepreneurial ecosystem (1/5)
- B8 Mindset
  - A23 (3) Entrepreneurial spirit (1/5)
  - A24 (1, 3, 4) Willingness to kill-your-darlings (3/5)
- B11 Intangible resources as competitive advantage
  - A30 (4, 5) Expertise (2/5)
  - A31 (2) Switching costs locks in customers(1/5)
  - A32 (3) Business model (1/5)
  - A33 (3) Methodology is unique in the market(1/5)

C2 No-code effects firm resource utilization

- B4 No-code optimizes resources
  - A11 (1,2,3,4,5) No-code optimizes our resource usage (5/5)
  - A12 (1) Frees up creativity (1/5)
  - A13 (2) Frees up administrative resources (1/5)
  - A14 (1, 2, 3, 4, 5) Frees up time (5/5)
  - A15 (1,2,3,4,5) Frees up resources to focus on customer problems (5/5)
- B5 No-code is less resource intensive
  - A16 (1,2 ,3,4,5) Less resource intensive (5/5)
  - A17 (1, 3, 4,5) Reduced need for human capital (4/5)
  - A18 (1, 3, 4, 5) Use less financial resources (4/5)
- B6 Social resources and no-code integration
  - A19 (3) Highly complementary to methodology(1/5)
  - A20 (1, 5) Developers did not want to use it (2/5)
- B14 Less resource intensive
  - A45 (1,3,4,5) No-code is less resource intensive (4/5)
  - A46 (3) Lower price than paying full salaries for developers(1/5)
- B17 Resource intensive
  - A56 (2) Might incur heavy price increase (1/5)
  - A57 (1) Highly priced (1/5)

C3 Citizen developers

- B7 Competency is the most important resource for no-code utilization

  A21 (1,2,3, 4,5) Technical competency is required (5/5)

  A22 (4) You need good structure (1/5)

- B13 No-code can be used by many

  A40 (5) Code becomes visual (1/5)

  A41 (3, 5) Easily comprehensible for non-programmers (2/5)

  A42 (5) No-code bridges the gap between business and development (1/5)

  A43 (1, 3, 4) Killing your darlings becomes easier(3/5)

C4 Time to market and no-code

- B10 Development in no-code

  A29 (1, 2, 4) Development speed is greatly increased (3/5)

- B12 Reduced time to market

  A34 (1,2,3,4,5) Development speed (5/5)

  A35 (1, 2, 3, 4) Faster iteration and testing (4/5)

  A36 (1, 3, 5) Shorter feedback loop (2/5)

  A37 (4) Faster to user testing (1/5)

  A38 (4) Flexibility is increased(1/5)

  A39 (5) Faster to market with functional prototypes (1/5)

  A44 (4) Development follows lean startup method (1/5)

C5 Customer focused development

- B9 The customer in the center

  A25 (1,2,3,4,5) Customer focused development (5/5)

  A26 (1,2,4,5) Better customer relationships as a result (4/5)

  A27 (4) Good reputation amongst customers (1/5)

  A28 (2) High flexibility for customer (1/5)

- B15 Customer related

  A47 (4) Faster to customer insight (1/5)

C6 No-code and evaluation process

- B16 Platform specific disadvantages

  A48 (1,2,3,4,5) Functionality restrictions (5/5)

  A49 (2) Less tailoring possible (1/5)

  A50 (1, 3, 5) Vendor lock-in (3/5)

  A51 (1) No-code vendors with short runtime creates operational risk(1/5)

  A52 (5) No no-code platform can solve all your problems (1/5)

  A53 (3) Complexity is an issue (1/5)

  A54 (5) Flexibility is an issue (1/5)

  A55 (2) Speed of development can create a false feeling of delivery (1/5)

- B18 Evaluation of no-code and firm resources prior to adoption

  A58 (3, 5) Did evaluate resources (2/5)

  A59 (1) Good price was main criteria (1/5)

  A60 (5) Lack of traditional developers (1/5)

  A62 (1) Adoption was random (1/5)

  A63 (3, 5) Each project is evaluated for the use on no-code (2/5)

- B19 Technology specific evaluation

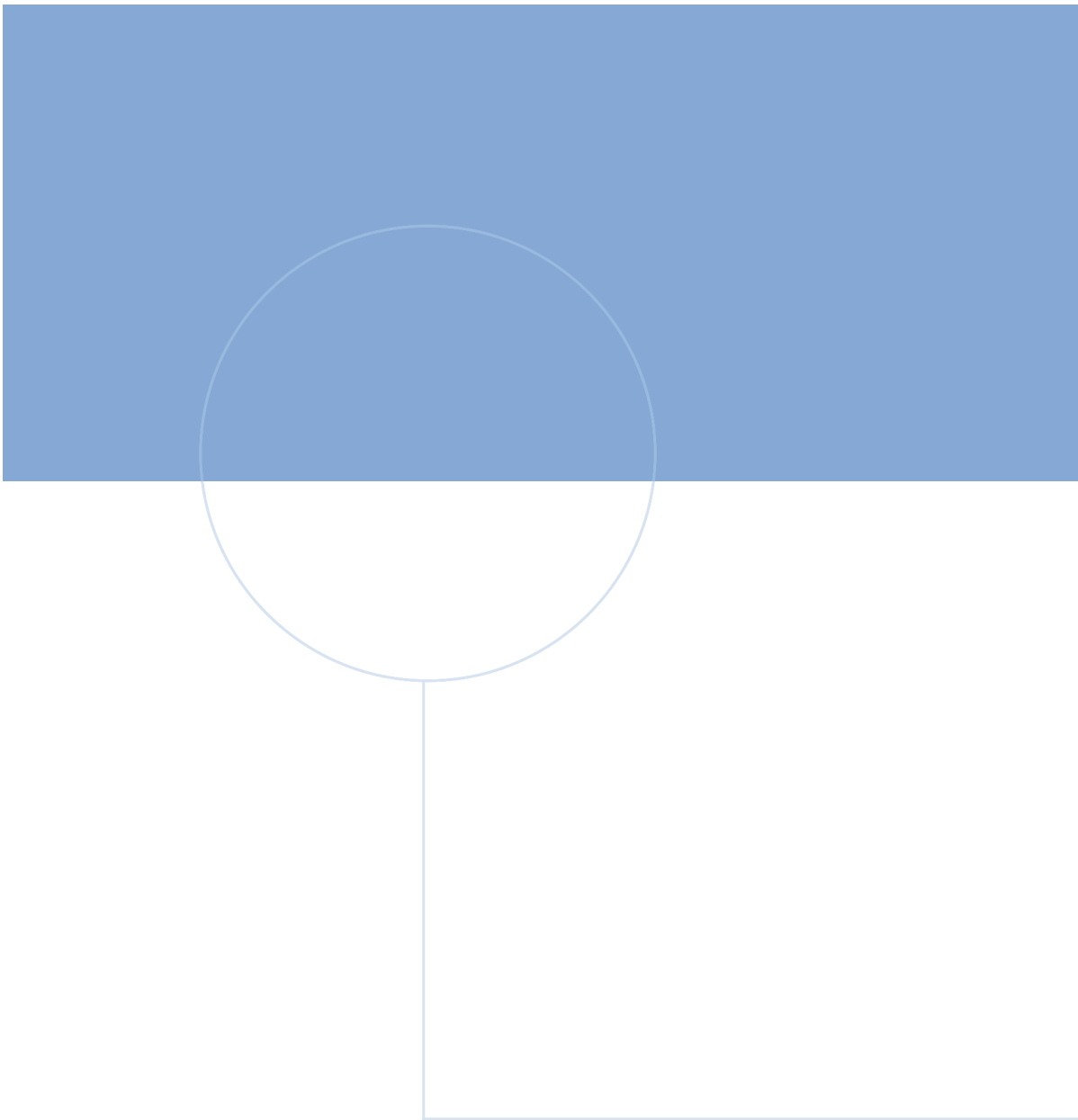
  A64 (1, 2, 3) Vendor lock-in pose a threat (3/5)

  A65 (2) Complexity was an issue (1/5)

  A66 (2) Less operation critical tasks are made in no-code(1/5)

  A67 (2) Internal processes (1/5)

  A61 (1, 4) Developers becomes dissatisfied (2/5)



NTNU
Norwegian University of
Science and Technology

SINTEF