LUNDS UNIVERSITET
Ekonomihögskolan

Institutionen för informatik

Maximising Value through Low-Code Technology Assimilation

A Qualitative Study of the Influential Factors during the Post-Adoption Phase

Kandidatuppsats 15 hp, kurs SYSK16 i Informationssystem

Författare: Maria Dalgren
Natali Plazonic

Handledare: Niki Chatzipanagiotou

Rättande lärare: Blerim Emruli
Osama Mansour

ENGLISH TITLE: Maximising Value through Low-Code Technology Assimilation: A Qualitative Study of the Influential Factors during the Post-Adoption Phase

SWEDISH TITLE: Maximering av Värde genom Low-Code Teknologi Införlivande: En Kvalitativ Undersökning av Avgörande Faktorer Post-Adoption

AUTHORS: Maria Dalgren, Natali Plazonic

PUBLISHER: Institutionen för informatik, Ekonomihögskolan, Lunds universitet

EXAMINATOR: Osama Mansour, Docent

SUBMITTED: May, 2024

DOCUMENT TYPE: Bachelor’s thesis

PAGE NUMBERS: 52

KEYWORDS: Low-Code Technologies, Information Technology Assimilation, Post-Adoption, Resource-Based theory.

ABSTRACT:

This bachelor’s thesis explores assimilation of low-code technologies post-adoption within organisations. Through a qualitative approach involving nine semi-structured interviews, which were analysed thematically and discussed with resource-based theory, two distinct uses of low-code technologies are identified: as a technical facilitator for fast and cost-effective software development, and as a digitalisation and innovation enabler. Factors influencing low-code technology assimilation, such as internal teams, solution catalysts, low-code technology readiness, and value realisation, are identified. The findings highlight the strategic importance of aligning low-code technology use with organisational goals and organising internal resources accordingly. Theoretical contributions are made by analysing low-code technology assimilation through the lens of resource-based theory. Practical implications are provided for practitioners to optimise low-code assimilation processes, including alignment of IT strategy based on the organisation’s use of low-code technologies, formation of teams with specific roles and responsibilities, as well as identifying and empowering solution catalysts within the organisation. Future research directions are offered to explore hybrid uses and factors affecting low-code technology assimilation success.

SAMMANFATTNING:

Denna kandidatuppsats utforskar användningen och införlivandet av low-code teknologier i organisationer efter lanseringen (post-adoption). Genom en kvalitativ undersökning som innefattar nio semistrukturerade intervjuer har två huvudsakliga användningsområden för low-code teknologier identifierats; dels för att möjliggöra en snabb och kostnadseffektiv programutveckling, dels som drivkraft för en digital transformation och främjande av innovation. De identifierade faktorerna som påverkar införlivandet av low-code teknologier är: interna team, lösningsfrämjare, organisationens beredskap, och värdeskapande. Forskningsresultaten framhäver den strategiska vikten av att anpassa användningen av low-code teknologier till organisatoriska mål och att organisera interna resurser efter dem därefter. De teoretiska implikationerna görs genom att analysera forskningsresultaten utifrån resource-based teori. De praktiska implikationerna innefattar justering av IT-strategier enligt organisationens användning av low-code teknologier, etablering av interna team med fastställda roller och ansvarsområden, samt identifiering av lösningsfrämjare inom verksamheten. Framtida forskningsinriktningar föreslås även för vidare utforskning av hybridanvändning och faktorer som påverkar framgången i införlivandet av low-code teknologier.

Acknowledgments

We would like to thank our supervisor, Niki Chatzipanagiotou, for guidance and support during the development of this bachelor's thesis. Your expertise and feedback have been instrumental in shaping our work and made this process truly enjoyable. We extend our gratitude to the participants who generously contributed their time and insights. Interviews with you on the topic of low-code technology were both insightful and an important aspect for the completion of this thesis, as well as inspiring and pleasant.

Thank you for enriching the quality of our work.

May, 2024

Natali Plazonic & Maria Dalgren

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Background

Most companies today invest in technology from a strategic point of view – to help the company gain a competitive advantage and generate value (Denning, 2023). While adopting new technology is an important step to reap the benefits of it, information technology assimilation (IT assimilation) is often seen as a goal in today's rapidly changing and competitive market (Martínez-Caro et al. 2020). IT assimilation is a notion describing a phase during which companies succeed at leveraging the potential of the new information technology in their business activities and strategies (Roberts et al. 2023). In other words, these companies not only consider and then adopt the technology but also use the technology to drive growth, innovate and optimise business activities. According to resource-based theory (RBT), when technology is assimilated into an organisation, it becomes a resource, which can only happen when the technology is not only adopted but also used to gain a competitive advantage (Barney & Clark, 2007). However, the effective assimilation of information technology (IT) and information systems (IS) is not only related to technical complexity or cost issues but to social and organisational factors as well (de Mattos & Laurindo, 2017). This is why adopting the technology does not always result in the technology being assimilated, which has significant impacts on the organisational results and the competitive advantages gained from the IT/IS (Martínez-Caro et al. 2020).

Some of the technologies that are consistently growing in popularity and are recognised as ways to gain competitive advantage in various industries are the *low-code technologies*. Low-code technology, a contemporary approach to software development, facilitates application creation by leveraging visual interfaces and abstracting coding complexities behind simple drag-and-drop functionalities (Bhattacharyya & Kumar, 2023; El Kamouchi et al. 2023). This reduces the dependency on IT professionals and increases the ability to concentrate on business logic and streamlining processes, rather than low-level implementation details (Bhattacharyya & Kumar, 2023; El Kamouchi et al. 2023). In practice and in the literature, these technologies are referred to as *low-code development* (LCD) *technologies, low-code development platforms* (LCDP), *low-code software development* (LCSD) or even just *low-code/no-code* (El Kamouchi et al. 2023). In this bachelor's thesis, we adopt the term *low-code technologies* as an umbrella term encompassing various platforms, development tools, and technologies. This term is used to refer to a range of technologies including but not limited to Mendix, Microsoft Power Platform, Salesforce, OutSystems, Appian, and Google AppSheet (El Kamouchi et al. 2023).

According to Rokis and Kirikova (2023), low-code technologies were first mentioned in a Forrester Research paper called "New Development Platforms Emerge For Customer-Facing Applications" from 2014 by Richardson and Rymer. Rokis and Kirikova (2023) found that low-code technologies have been growing in popularity since then and are recognised as an important step for companies in the digital transformation journey, which seek to stay ahead of the curve, enhance innovations and responsiveness to changing market dynamics. For example, to successfully compete, organisations are increasingly investing in low-code technologies as they present an innovative approach to IT solutions, allowing for simplified and faster application development and deployment and, in such way, addressing the growing demands for software solutions amongst the resource constraints (Bucaioni et al, 2022; El Kamouchi et al. 2023). A noticeable increase in interest towards low-code technologies is not only found in practice but also in academic research, from 36 peer-reviewed studies in 2019 to 255 in 2023, with a steady annual increase, as observed in an analysis of articles published in the SCOPUS database. At the same time, despite the widespread adoption of low-code technologies by practitioners, academic research on low-code technologies is still considered young and has begun growing in the past few years. The first peer-reviewed study appeared in 2018, and low-code technologies are still considered a new research topic, with the most common type of research among the peer-reviewed papers being "solution proposal" (Bucaioni et al. 2022, p. 9). In other words, most research papers focused on how these technologies can be used for building specific solutions to given problems, and provide answers to research questions concerning (1) the method of development (43.7%), and (2) feasibility (28.1%) (Bucaioni et al. 2022). However, focus on the assimilation of low-code technologies is scarce.

While it is recognised that IT investments offer significant benefits to companies, many of these investments fail to achieve the expected strategic goals (Denning, 2023). This is also reflected in many research studies over the past 30 years that explored what causes these failures, which factors play a role, and what mechanisms could explain them (Schmidt, 2023). The results of the study by Schmidt (2023) point in many different directions, such as issues with senior management, planning, risk management, and staff, and this seems to be an ongoing challenge with newer technologies as well.

1.2 Problem Identification

Just adopting new technology is not enough to reap the benefits long-term and contribute to the competitiveness of organisations (Denning, 2023). Low-code technology, like many others, needs to be appropriately assimilated into the firm's business processes and used to shape new strategies to realise its potential benefits. However, assimilation presents significant challenges, and assimilation efforts often result in limited success (Peppard & Ward, 2016). Therefore, it is imperative to understand how the assimilation of low-code technologies can be achieved.

This bachelor's thesis aims at addressing the gap between the increasing adoption of low-code technologies by organisations and the limited understanding of the factors influencing successful assimilation. Despite the recognised potential of low-code technologies, the process of successful assimilation and its impact on organisational outcomes is still underexplored.

1.3 Previous Related Studies

With a goal of helping users understand what different leading low-code technologies offer and how they align with organisations' specific requirements, Sahay et al. (2020) conducted a comprehensive examination of features, functionalities, and services on different low-code technologies. The authors analysed eight leading low-code technologies in the market and presented a comparative framework. This included detailed descriptions of the main architectural aspects of such technologies, common features and functionalities shared among the low-code technologies, as well as points where they differ or offer unique capabilities.

To further build on research regarding low-code technology adoption and focus on more than just technological aspects, Käss et al. (2023b) investigated which broader drivers and inhibitors for initial low-code technology adoption are present and how these differ in importance. Through semi-structured interviews with IT experts and a ranking-type Delphi study, the authors identified and ranked 12 drivers and 19 inhibitors. The most influential driver of adoption was found to be the improved efficiency of software development, while the most influential inhibitor was the lack of a low-code technology culture and reluctance to change within organisations. However, while the authors found a general consensus for the most and least important drivers and inhibitors, the consensus was weaker for the factors in between, which, according to the authors, depends on the adoption's context, such as the type of low-code technology adopted. Furthermore, the authors claim that, since practitioners rank technology aspects as less important, further research should focus on factors like people, tasks, and organisational structures.

In a recent multiple mini-case study by Käss et al. (2023a), the authors investigated which factors influence businesses to continue or discontinue using low-code technology after the initial adoption. The authors further categorise combinations of these factors into four archetypes, which illustrate typical examples of low-code technology adoption and non-adoption. The main identified factor hindering the further adoption is the sophistication level of the intended applications, and this factor is found applicable across platforms and types of developers. On the other hand, many factors have been found to have a positive effect on further adoption. One highlighted factor was related to expected efficiency improvements. However, this same factor with the same value and effect was also present in the non-adoption cases, showing that the complexity factor seems to be more significant and enough to lead to non-adoption. The authors conclude that low-code technology adoption does not follow a traditional technology adoption process, and the decision to further adopt or not depends on the combination of multiple factors. Moreover, they conclude that the optimisation of two subsystems in an organisation, the social- and technical subsystems, is of highest importance in the decision to adopt low-code technologies, and that technical, organisational, and environmental factors need to be considered in further research that aims at understanding the low-code technology post-adoption fully.

Although researchers have made much progress in understanding low-code technologies (El Kamouchi et al. 2023; Pinho et al. 2023), there are still many gaps in the literature. Firstly, many studies have investigated technological aspects, benefits and factors affecting low-code consideration and initial adoption, yet the research on low-code assimilation is scarce. Secondly, while IT assimilation has been researched during the past decade in various contexts of technology and management including e-commerce, ERP and supply chain management (Wang, 2019), to our knowledge, understanding of low-code technology assimilation is limited.

1.4 Research Purpose and Research Questions

Based on the aforementioned identified research gap, the purpose of this bachelor's thesis is to explore how organisations use low-code technologies post-adoption, with the aim of contributing to an improved understanding of low-code technology assimilation and also yielding practical implications for practitioners to reap more benefits from low-code adoptions. To address this research purpose, our research questions (RQ) are as follows:

- RQ1: How do organisations use low-code technologies post-adoption?
- RQ2: What factors affect the assimilation of low-code technologies?

1.5 Delimitations

In this bachelor's thesis, we have chosen to maintain a focused scope of research. More specifically, we do not focus on the technical specifics of low-code technologies nor does our focus extend to aspects of IT assimilation that are not directly related to low-code implementations. However, we do not differentiate between different types of low-code technologies when analysing and discussing our findings.

While our interview participants all have insights into low-code technology implementation within the specific organisation being discussed, their roles within these organisations vary. Additionally, we only explore this topic in a Nordic context. Finally, the organisations discussed may not represent the general situation, and as such, the results do not enable any type of statistical generalisation to be made regarding low-code technology assimilation.

1.6 Thesis Organisation

This bachelor's thesis is structured as follows. After the introduction, we provide a detailed literature review. Then, we briefly describe our methodology. Next, we present our empirical findings. The discussion that follows emphasises how our findings relate to the previous research, as well as present recommendations and lessons that can be learned. Finally, we provide a concluding summary, identify theoretical and practical contributions, as well as possible future research directions.

2 Literature Review

2.1 Search Procedure

The search for the material included in this bachelor's thesis literature review was made in the following scientific databases: *ACM Digital Library, arXiv.org e-Print archive, Business Source Complete, Emerald, IEEE, Inspec, Springer eBook Collections, and Scopus*. The search was conducted by using the following keywords: low-code, low code, LCDS, LCDP, in combination with *technology*, *adoption*, *implementation*, *assimilation*, and *resource-based*, by using the logical operator AND to join the primary terms and OR to combine the alternate ones. For example, "low-code" OR "low code" AND adoption.

The selection process for the literature review was conducted by a set of following inclusion criteria:

- Papers published between January 2019 and March 2024, with the exception of material related to theoretical concepts.
- Papers available as a full-text journal article, a conference proceedings paper, or a book chapter.
- Papers that have been peer-reviewed.
- Papers that have been published and written in English.

Thus, the material that did not fall into the inclusion criteria was ultimately excluded (For an overview of the material included, see Table 1).

Table 1. Overview of the material included in the literature review

Article	Source	Motivation
Pinho, D., Aguiar, A. and Amaral, V. (2023). What about the usability in low-code platforms? A systematic literature review, *Journal of Computer Languages*, 74.	ScienceDirect	The article was used to define low-code technology and describe common characteristics
El Kamouchi, H., Kissi, M. and El Beggar, O. (2023). Low-code/No-code Development: A systematic literature review, 2023 14th International Conference on Intelligent Systems: Theories and Applications (SITA), pp. 1–8.	IEEE	The article was used to understand the perceived benefits and challenges with low-code technologies
Rokis, K. and Kirikova, M. (2023). Exploring Low-Code Development: A Comprehensive Literature Review, *Complex Systems Informatics and Modeling Quarterly*, 2023(36), pp. 68–86.	Scopus	The article was used to explore the present research on low-code technologies
Aveiro, D. et al. (2023). Traditional vs. low-code development: comparing needed effort and system complexity in the NexusBRaNT experiment, 2023 IEEE 25th Conference on Business Informatics (CBI), pp. 1–10.	IEEE	The article was used to support views on effort- and time-related benefits
Chuanjian, C., Shuze, G. and Hua, W. (2023). Research on Software Development Based on Low-Code Technology, 2023 2nd International Conference on Artificial Intelligence and Autonomous Robot Systems (AIARS), pp. 210–213.	IEEE	The article was used to showcase time-related benefits and to explore perceived concerns about their use
Ciucan-Rusu, L., Timus, M., Stefan, D., and Calin-Adrian, C. (2023). Low-code solutions integration for digital process automation - application in management, 2023 22nd RoEduNet Conference: Networking in Education and Research (RoEduNet), pp. 1–5.	IEEE	The article was used to support views on effort- and time-related benefits

Brunninghaus, M. and Rocker, C. (2022). Low-Code Development in Worker Assistance Systems: Improving Flexibility and Adaptability, 2022 IEEE 20th International Conference on Industrial Informatics (INDIN), pp. 366–373.	IEEE	The article was used to explore how low-code technologies are used for addressing unique needs
Käss, S., Strahringer, S. and Westner, M. (2022). Drivers and Inhibitors of Low Code Development Platform Adoption, IEEE 24th Conference on Business Informatics (CBI), pp. 196–205.	IEEE	The article was used to describe factors seen as barriers or inhibitors during low-code technology initiation
Roberts, N., Jeyaraj, A. and Pullin, J.E. (2023). Assessing the Connections among Top Management Support, IT Assimilation, and the Business Value of IT: A Meta-Analysis, *Journal of the Association for Information Systems*, 24(1), pp. 107–135.	Business Source Complete	The article was used as an introduction to IT assimilation and the importance of top management support
Martínez-Caro, E., Cepeda-Carrión, G., Cegarra-Navarro, J.G. and Garcia-Perez, A. (2020). The effect of information technology assimilation on firm performance in B2B scenarios, *Industrial Management & Data Systems*, 120(12), pp. 2269–2296.	Emerald	The article was used to understand the importance of IT assimilation
Denning, S. (2023). Recognizing and outmaneuvering the resistance to digital transformation, *Strategy & Leadership*, 51(2), pp. 10–16.	Emerald	The article was used to understand current views on IT assimilation
Gao, P., Gong, Y., Zhang, J., Mao, H. and Liu, S. (2019). The joint effects of IT resources and CEO support in IT assimilation: Evidence from large-sized enterprises, *Industrial Management & Data Systems*, 119(6), pp. 1321–1338.	Emerald	The article was used to describe factors influencing IT assimilation
Wang, N., Liang, H., Ge, S., Xue, Y. and Ma, J. (2019). Enablers and inhibitors of cloud computing assimilation: an empirical study, *Internet Research*, 29(6), pp. 1344–1369. doi:10.1108/INTR-03-2018-0126.	Emerald	The article was used to explore enablers and inhibitors of IT assimilation
Razak, S.F.A., Ernn, Y.P., Yussoff, F.I., Bukar, U.A. and Yogarayan, S. (2024). Enhancing Business Efficiency through Low-Code/No-Code Technology Adoption: Insights from an Extended UTAUT Model, *Journal of Human, Earth, and Future*, 5(1), pp. 85–99.	Scopus	The article was used to describe theoretical understanding of low-code technology initiation
Alamin, M.A.A., Uddin, G., Malakar, S., Afroz, S., Haider, T. and Iqbal, A. (2023). Developer discussion topics on the adoption and barriers of low code software development platforms, *Empirical Software Engineering*, 28(1).	arXiv	The article was used to understand factors seen as challenges during low-code technology adoption
Käss, S., Westner, M. and Strahringer, S. (2023b). Practitioners' Perceptions on the Adoption of Low Code Development Platforms, *IEEE Access, Access, IEEE*, vol. 11, pp, 29009–29034	IEEE	The article was used to describe factors seen as barriers or inhibitors during low-code technology adoption
Pacheco, J., Garbatov, S. and Goulao, M. (2021). Improving collaboration efficiency between ux/ui designers and developers in a low-code platform, ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), pp. 138–147.	arXiv	The article was used to show an example of successful low-code technology assimilation
Redchuk, A., Mateo, F.W., Pascal, G., and Tornillo, J.E. (2023). Adoption Case of IIoT and Machine Learning to Improve Energy Consumption at a Process Manufacturing Firm, under Industry 5.0 Model, Big Data and Cognitive Computing, 7(1).	Scopus	The article was used to show an example of successful low-code technology assimilation
Käss, S., Westner, M. and Strahringer, S. (2023a). A Multiple Mini Case Study on the Adoption of Low Code Development Platforms in Work Systems, IEEE Access, Access, IEEE, vol. 11, pp. 118762–118786.	IEEE	The article was used to describe factors influencing further (non-)adoption of low-code technologies
Barney, J. (1991). Firm Resources and Sustained Competitive Advantage. *Journal of Management*, 17(1), p. 99.	Business Source Complete	The article was used for theoretical background (RBT)

Barney, J.B. and Clark, D.N. (2007). *Resource-based theory: creating and sustaining competitive advantage*. Oxford University Press.	Oxford University Press	The article was used for theoretical background (RBT)
Peppard, J. and Ward, J. (2016). *The strategic management of information systems: Building a digital strategy.* 4. ed. Wiley.	Wiley	The article was used for exploring present knowledge behind successful digital strategies
Mata, F.J., Fuerst, W.L. and Barney, J.B. (1995). Information Technology and Sustained Competitive Advantage: A Resource-Based Analysis, MIS Quarterly, 19(4), pp. 487–505.	Business Source Complete	The article was used for exploring RBT in the context of IT/IS
Caldeira, M.M. and Ward, J.M. (2003). Using resource-based theory to interpret the successful adoption and use of information systems and technology in manufacturing small and medium-sized enterprises, *European Journal of Information Systems*, 12(2), pp. 127–141.	Inspec	The article was used to understand how RBT was previously adapted within an IS/IT context

2.2 Low-Code Technology

While there has been research on low-code technologies for a few years, the literature did not seem to have come to a consensus regarding a formal definition. The first scientific article to our understanding that has provided a formal definition is done by Pinho et al. (2023). To use a definition that stems from the literature and can guide future work on the topic in a more consistent way, the definition adopted in this bachelor's thesis is based on Pinho et al. (2023, p. 12) and is as follows:

> *Low-code development is a set of approaches, technologies, and tools that enable rapid application development through techniques that reduce the amount of code written. These approaches can make it possible for end-user developers to program software and use techniques and tools often including but not limited to model-driven engineering, domain-specific languages, and drag-and-drop mechanisms*.

2.2.1 Perceived benefits

Literature reviews on low-code technologies have consistently identified multiple benefits. For example, El Kamouchi et al. (2023) identified minimising the cost and time as the most commonly discussed benefits in academic research, followed by maintainability, simplicity and easy use by non-technical individuals. Other benefits include improved productivity, reusability and agility (El Kamouchi et al. 2023). Similarly, Rokis and Kirikova (2023) provided a review of benefits as well, highlighting the same benefits, in addition to improved collaboration among the development team and business, promoted digital innovation, and mitigation of shadow IT (i.e. the use of technologies without central IT governance or support).

The time related benefits were researched in several studies. Aveiro et al. (2023) conducted an experiment and compared traditional development approach and low-code approach and found that, in their particular case, the low-code approach delivered a high-quality and functional application with a 94.63% reduction in the needed effort, and a reduction of 86% in complexity (in terms of lines of code). Similarly, Chuanjian et al. (2023) conducted a study where they, using low-code technologies, built an information management system for the elderly. Their findings showed that this approach reduced the development by 16 days compared to the traditional method, while still meeting the requirements and standards (Chuanjian et al. 2023).

Another category of benefits is related to the empowerment of organisations to use low-code technologies for quickly addressing their own unique needs, without the need of long and complicated development processes. In a case study by Ciucan-Rusu et al. (2023), Microsoft's low-code platform (Power Platform) was used as a toolkit to develop a digital solution that supports project implementation. One of the most important benefits recognised by the authors is the ability to simply automate processes within a project, and therefore allowing more time to focus on other important aspects (Ciucan-Rusu et al. 2023). Another example of an organisation using low-code technologies to quickly address their unique needs was done by Brunninghaus and Rocker (2022) in the context of worker assistance systems – systems that support workers in complex production processes or unfamiliar manual activities. Empirical evidence showed that using low-code technology with a simple editing tool was faster, more flexible, and offered better usability than the existing solutions (Brunninghaus & Rocker, 2022). According to both studies, low-code technologies can be a highly promising and useful option in scenarios where individuals without coding knowledge can benefit from quick app development, as well as in scenarios where new applications have a shorter product lifecycle and high process variability that requires a lot of time and effort (Brunninghaus & Rocker, 2022; Ciucan-Rusu et al. 2023).

2.2.2 Perceived challenges

While recognising the benefits of low-code development, multiple authors mention concerns as well, especially regarding data privacy and stability (Chuanjian et al. 2023; El Kamouchi et al. 2023). Additionally, challenges recognised in the literature include being locked to a particular vendor's environment, concerns about performance, and a low level of scalability (El Kamouchi et al. 2023; Pinho et al. 2023). Furthermore, potential high costs that come with hosting and licensing, as well as weak documentation, low customisation possibilities, and unfriendly user experience are mentioned as challenges as well (El Kamouchi et al. 2023; Pinho et al. 2023).

A few benefits are recognised as being a challenge as well. More specifically, despite the concept of empowering citizen development (i.e. development done by non-technical individuals) being mentioned repeatedly as a benefit, low-code technologies are often seen as too complicated for many and as having a high learning curve, since they still require some IT knowledge or learning a lot about how the tools work before fulfilling software needs (El Kamouchi et al. 2023; Pinho et al. 2023).

Similarly, while the pre-made components generally integrated in low-code technologies are seen as a benefit since they support simpler and easier development, they do come with recognised challenges (Käss et al. 2022). For example, their customisation often requires handling of bugs in the code (El Kamouchi et al. 2023). However, due to the novelty of the approach and the frequent changes based on IT trends, low-code technologies often came with low bug resolution and weak documentation (El Kamouchi et al. 2023; Pinho et al. 2023). Additionally, this simplicity can hamper compatibility, which is considered crucial when adopting low-code technologies (Käss et al. 2022). For example, the simplicity of low-code technologies can be an inhibitor when it comes to ensuring interoperability between other systems, integration with existing systems, and allowing creation of a wide range of functionalities (Käss et al. 2022).

2.3 IT Assimilation

IT assimilation within organisations has gained significant attention due to its potential to improve decision-making processes, ways of running operations and overall competitiveness (Roberts et al. 2023). Broadly speaking, IT assimilation refers to the process by which organisations integrate new technologies, such as enterprise resource planning (ERP) systems or Software as a Service (SaaS) platforms, into their existing workflows and business processes (Martínez-Caro et al. 2020). This process involves not only the technical implementation, but also the organisation's capacity to use the new technology to support, shape, and enable its business activities in new and improved ways. This is different from the mere IT adoption, where an organisation only starts using new technology, without fully internalising it throughout the organisation. In other words, IT assimilation can be understood as a step further from IT adoption. As stated by Martínez-Caro et al. (2020), if done properly, bringing in new technology can spark new ideas, transform ways of doing business and potentially lead to new business models.

The importance of IT assimilation is further highlighted by its role in shaping organisational performance and success (Martínez-Caro et al. 2020). Research by Martínez-Caro et al. (2020) suggests that organisations with strong IT assimilation capabilities are better positioned to capitalise on the benefits of technology investments and achieve superior financial and operational outcomes. Moreover, effective IT assimilation is seen as an enabler. As argued by Denning (2023), IT assimilation enables organisations to better adapt to changing market conditions, seize new business opportunities, and stay ahead of competitors in rapidly changing industries.

2.3.1 IT assimilation stages

Although different variations of the IT assimilation stages have been proposed, according to our understanding of the literature, research consistently shows at least three stages of organisational IT assimilation: (1) initiation, during which a newly discovered technology is evaluated for its usefulness, (2) adoption, in which the decision to adopt the technology is made and the technology starts being used, and (3) assimilation, during which the technology becomes widely used and is seen as an integral part of business strategies.

While each stage presents unique challenges and opportunities and is required for complete assimilation, the third stage is recognised as a crucial step if an organisation aims to accomplish valuable positive effects on organisational performance (Roberts et al. 2023; Martínez-Caro et al. 2020). This bachelor's thesis primarily focuses on the third stage – the assimilation stage. Accordingly, the following section describes previous research on the IT assimilation stage. Following this, to enhance understanding of low-code technologies, we cover previous studies on all three stages in connection with low-code technologies specifically.

A study by Gao et al. (2019) focused on understanding how large businesses can effectively use IT to support their business goals and strategies. The researchers pinpointed IT resources and top management support as two key factors contributing to the successful assimilation of IT within large organisations. Furthermore, positive synergy was observed between various IT resources and top management support. For example, the IT resources present in the organisation should be managed effectively to contribute further to IT assimilation. However, authors also conclude that some IT resources, such as infrastructure and human resources, need further examination to understand how they influence IT assimilation.

Similarly, Martínez-Caro et al. (2020) researched Spanish companies and found that for successful IT assimilation, it’s not enough to just invest in IT infrastructure and resources like IT training. The authors provide empirical evidence indicating that active involvement and support from managers is essential for ensuring employees’ perception of IT as useful in their work processes. However, they argue that getting the new technology to be routinely used doesn’t guarantee the transformation and exploitation of knowledge within the organisation. The researchers suggest that management practices should be driven by a strategy that goes beyond the implementation of IT and even the application of IT-related knowledge, to also foster an organisational culture inclined towards innovation.

This is in line with a study by Wang et al. (2019) that explored IT assimilation in the context of cloud computing. The authors claim that support and participation of top managers should always be encouraged and is especially beneficial within organisations that have a low level of organisational inertia (OI) (low OI can be simply described as low resistance to change regarding how things are done). However, the results of the study indicate that, for organisations with a high level of OI, just support from top management is insufficient and additional strategies must be implemented to prevent assimilation efforts from failing. The authors propose specific forms of support, such as help desks or training programs, along with policies for smooth transitions. According to the authors, when these measures are combined with top management support, they serve as potential strategies that ensure assimilation, at least within the cloud computing context.

2.4 IT Assimilation within the Low-Code Technology Context

As mentioned in previous paragraphs, the process to IT assimilation can be divided into three stages: the initiation stage, the adoption stage, and the assimilation stage. In the following paragraphs, we provide an overview of the literature regarding low-code technologies within organisations at different stages.

2.4.1 Low-code technology initiation

Razak et al. (2024) conducted research on Malaysian businesses implementing low-code technology, focusing on users' behavioural intentions toward the adoption of the low-code technology platform. By using The Extended Unified Theory of Acceptance and Use of Technology (UTAUT) model as the theoretical framework, the authors identified components that influence potential adopters' decision-making processes. More specifically, the results of the study show a positive relationship between the following five factors with the behavioural intention to adopt low-code technology: Performance Expectation, Effort Expectation, Social Influence, Perceived Risk, and Perceived Cost.

According to the authors, these components are essential and work together to shape decision-making related to low-code technology adoption.

In a literature review conducted by Käss et al. (2022), additional and more specific factors influencing the initiation stage of low-code technology were identified. Seven factors that drive the adoption and 13 factors that inhibit adoption were consistently found in the literature as significant. The most frequently discussed drivers for adopting low-code technologies are the technologies' ability to enhance software development efficiently, lower the entry barriers for non-skilled developers, and the overall simplification of the development processes. On the other side, among the 13 inhibitors found in the literature, limited flexibility and customisation, difficulties with operability with other systems and their integration, together with the concern of vendor lock-in, are found.

2.4.2 Low-code technology adoption

Alamin et. al (2023) explored the online developer forum Stack Overflow (SO) to better understand the most common issues when actively using low-code technologies. By analysing more than 33 000 SO posts related to the topic, the authors identified the parts of low-code technologies that were found most challenging by users. The authors considered the common issues as those having the largest number of questions related to them, which showed to be application customisation, database and file management, and third-party API integration. These results suggest which aspects are likely to be found as the most challenging once the low-code technology is adopted and starts being used. Interestingly, the authors noticed that the topics of low-code technology adoption and low-code technology maintenance have gained considerable attention in recent years and that all topic categories are evolving rapidly.

Similar to the third-party API integration challenge mentioned, aspects often brought up by practitioners are integration and interoperability complexities. According to Käss et al. (2023b), integrating low-code technologies with existing systems and processes seems to be a frequently identified challenge in the literature, especially if an organisation has legacy systems that are not compatible with low-code technologies. This emphasises the complexity involved in the adoption stage, when the organisation transitions to low-code solutions within established IT infrastructures (Käss et al. 2023b).

Additionally, during the adoption stage, organisations that have the support of employees who have previously worked with and possess knowledge about low-code technologies – often referred to as champions – are well-positioned to encourage further adoption and usage (Käss et al. 2023a). Availability of vendor support and training, for example through tutorial-based documentation, has also been found important to facilitate the adoption process by providing necessary guidance and resources (Alamin, et al. 2023).

2.4.3 Low-code technology assimilation

According to Pacheco et al. (2021), efficiency between designers and front-end developers is a crucial factor in today's market characterised by complex users' experience needs and scarcity of available individuals with relevant expertise. In their study, the authors utilised OutSystems as a low-code technology and tested if and how it can be used to improve the development of customer-facing applications. A solution that automates the generation of structured application screens and takes advantage of low-code reusable UI components was implemented. The results indicate that such a solution can improve the value these teams can deliver to their customers. More specifically, despite investing the same amount of effort, the team increased the number of application screens created between 150% and 400% and reduced time by 56%. While this study does not specifically explore low-code technology assimilation, it does provide an example of successful assimilation of low-code technology in business processes, crucial for a company's success and adaptability to changing demands.

As another example of successful low-code technology assimilation, a study by Redchuk et al. (2023) presented a case where a low-code AI platform was adopted to improve energy efficiency and lower the environmental footprint of a food industry company. The low-code solution was not only adopted but also integrated into the existing organisational infrastructure. In this way, it both contributed to energy efficiency, which was one of the main goals, but also empowered employees, giving them a central role in the improvement of operations (Redchuk et al. 2023).

Käss et al. (2023a) were interested in understanding what drives further adoption of low-code technology after the initial adoption. They analysed 36 case studies, of which 26 included continued adoption of the technology, and 10 discontinued. Three archetypes of continued low-code use emerged from their analysis. The first archetype, the Application Development Democratiser, aims at empowering and increasing the number of individuals without advanced coding skills who can develop applications. The second archetype, the Synergy Realiser, aims to achieve efficiency by leveraging already existing systems or licences. The third archetype, IT Resource Shortage Mitigators, addresses the challenge of developing these applications when there's limited traditional IT resources and aims at building applications efficiently despite these constraints. Authors found that all three archetypes see the expected efficiency improvements, compatibility, internal IT capabilities, and sophistication of the application to be developed as key reasons for further low-code technology use. They argue that these results indicate that, for further adoption to happen, at least one of these factors needs to be present. On the other hand, the authors identified one archetype that leads to non-adoption – the Intricacy Adversary. This archetype is characterised by the perception of the application that should be developed as too complex and sophisticated. Furthermore, the results show that this factor is so strong that it is sufficient for discontinuation of low-code technology use, despite the positive factors being present.

2.5 Resource-Based Theory

The resource-based theory by Barney (1991) views an organisation as a system of firm resources. Firm resources are all assets, capabilities, and knowledge that are controlled by an organisation, such as information, knowledge, and human resources, that can be used to gain competitive advantage (Barney & Clark, 2007). According to Barney and Clark (2007), when these resources are used in a way that supports the achievement of organisational goals, they create value. However, for these resources to be a source of competitive advantage, value creation is not enough, and four main criteria need to be met: (1) resources should be valuable in terms of revenue generation or cost savings, (2) resources should be rare within the industry, (3) resources should be difficult to copy by the competitors, and (4) no different resource can deliver the same result.

2.5.1 IT from a resource-based theory perspective

Mata et al. (1995) have acknowledged that IT can add value to organisations, for example, by reducing costs or increasing revenues. However, when adapting the resource-based theory in the context of IT, the authors argue that adding value is not the same as IT being a source of sustained competitive advantage. According to the authors, IT is not likely to be a source of sustained competitive advantage since that would go against the main business models of most IT providers, whose main goal is to provide the technology to as many businesses as possible. Therefore, while it can be valuable, it is not likely that the technology will be rare, inimitable or not substitutable. The authors argue that gaining sustained competitive advantage through IT is not likely to be easy, but it can be possible: "For IT to be a source of sustained competitive advantage, organisations must focus less on IT, per se, and more on the process of organising and managing IT within a firm" (Mata et al. 1995, p. 500). However, according to the authors, this is difficult to achieve. According to the authors' perspective on IT/IS in relation to the resource-based theory, IT/IS can become an organisation's source of competitive advantage only when the organisation is able to leverage IT/IS resources in combination with other organisational capabilities. In this way, the focus is not on the technical aspects, but on building a capability for identifying business and organisational opportunities provided by those technical aspects.

2.5.2 A model of the IS/IT capability

Calderia and Ward (2003) conducted a study in which the resource-based theory was applied to explain the long-term success of IS/IT adoption. They concluded that sustained success is mainly determined by two internal organisational factors: top management involvement in the IS/IT used, and the IS/IT knowledge present internally or through a very close external partnership. The authors developed a model based on the resource-based theory, containing three levels: individual level, organisational level, and business level (see Figure 1). The individual level emphasises internal technical skills, IS/IT managerial skills, and general business and managerial skills. These together create the organisational level competences through organising and adjusting processes and structures. More specifically, processes that specify how to work with the selected IT/IS successfully as well as in a way that creates actual benefits to the business. Finally, to build a capability from IS/IT on a business level, the model emphasises prioritisation from top management and investment allocation.

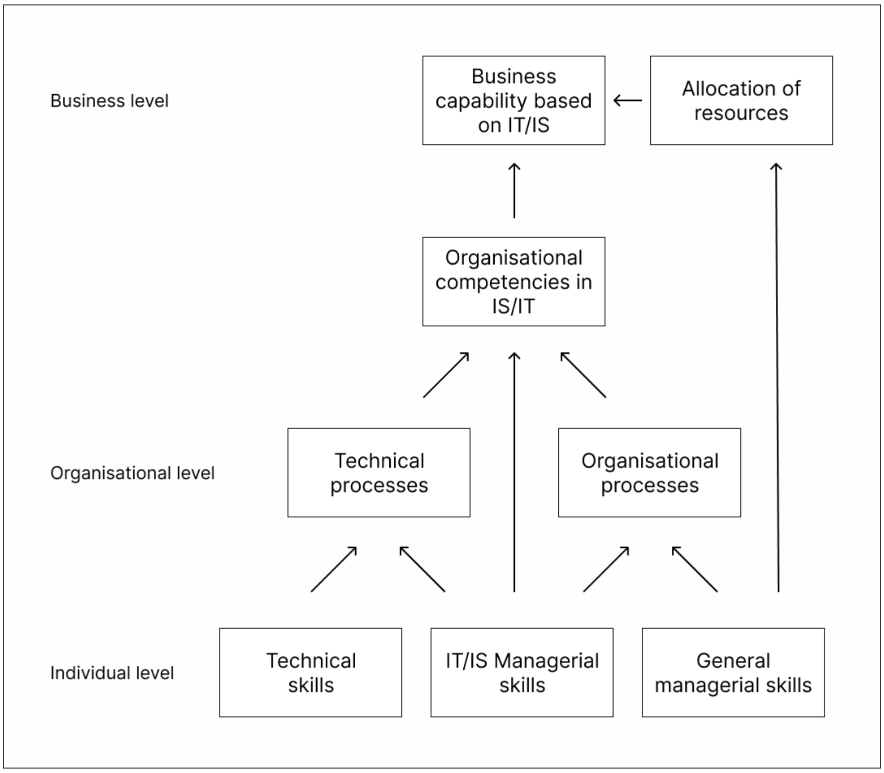

Figure 1. A resource-based model to understand the long-term success of IS/IT adoption (adapted from Calderia and Ward, 2003, p. 138)

This bachelor’s thesis focuses on IT assimilation stemming from the resource-based theory, as the assimilation of IT into business operations has become recognised as central to organisations' efforts to improve business activities, innovation and adaptability. This theory and its related key concepts are therefore further used in a more detailed manner for interpreting and discussing the research findings of the bachelor’s thesis.

3 Methodology

3.1 Research Approach

As described by Oates (2006), academic research typically uses one of the three methodological approaches: qualitative, quantitative, or mixed methods. According to the author, the quantitative approach, characterised by the collection and analysis of numerical data, is traditionally favoured by positivist researchers for its ability to quantify observations in a more objective manner. On the other hand, the author describes the qualitative approach as useful for studies requiring an in-depth understanding of human behaviour and social processes, that delve into non-numeric data and aim to uncover patterns, themes, and meanings. Lastly, it is possible to employ a mixed methods approach, combining the previously described methodologies.

Given the focus of our bachelor's thesis, the qualitative approach is considered more suitable and therefore chosen as a methodological approach based on the following main reasons. Firstly, the topic of low-code technology is considered new, and while IT assimilation has been researched in the past, previous studies on low-code technologies indicate that low-code technology does not follow a traditional technology adoption process (Käss et al. 2023a). Therefore, a qualitative method is considered more suitable for exploring this emergent challenge in depth, and gaining a nuanced understanding of complex phenomena that is less understood. Secondly, the goal is to collect detailed perspectives of participants and explore their experiences in a way that quantitative methods may not fully capture. Thirdly, a qualitative method allows us to place a strong emphasis on the lived experience of individuals within socio-technical contexts who are actively involved in using low-code technologies post-adoption.

3.2 Method of Data Collection

As described by Oates (2006), there are several methods of data collection in qualitative research, including interviews, focus groups and case studies. Each method has its own set of advantages and disadvantages, as well as characteristics suitable for various research objectives. Interviews, in particular, can be categorised into three main types: structured, semi-structured, and unstructured (Oates, 2006).

Structured interviews are based on predetermined questions, mirroring the format of a questionnaire but are conducted in person, which does not allow much deviation from the script to ensure consistency throughout all interviews (Oates, 2006). Semi-structured interviews are guided by a predefined set of questions but leave space for more flexibility, as they allow researchers to get a deeper understanding with follow-up questions and in such way, enable participants to speak their minds more freely (Oates, 2006). Finally, unstructured interviews allow most flexibility as they leave opportunity for open-ended dialogues, but also challenges when analysing the data (Oates, 2006).

Due to semi-structured interviews being considered as an approach more manageable for young researchers, as well as providing a good balance between structure and rich data, this bachelor’s thesis employs semi-structured interviews as a method of qualitative data collection. The interview guide used can be found in Appendix A.

3.2.1 Interview procedure

Each interview lasted 30 minutes, which was selected to both respect our participants’ time and fulfil the need for a thorough understanding of the topic while still maintaining a manageable scope of analysis. According to Patton (2015), shorter interviews can be effective when the research questions are clear and focused, and the questions asked are relevant. By designing the interview guide, preparing for the interview, and ensuring each question aligns with the research objectives, we aimed to achieve higher participation rates, respect participants' time, and gather meaningful data within a 30-minute period.

The interviews conducted in this bachelor’s thesis took place in multiple locations. Seven interviews were carried out online, using either Teams or Google Meet, and two interviews were conducted in private rooms at the Lund University premises. The choice of location was based on each participant’s preference. English was chosen as the language of communication with the participants, primarily because many of our participants as well as ourselves preferred English and felt they could provide better answers. Additionally, since not all participants were fluent in Swedish, having all data in English allowed us to have consistent questions and an easier data analysis process.

After each interview, transcriptions were done using Whisper, an open-source Automatic Speech Recognition (ASR) model by OpenAI. However, each transcription was manually checked and corrected by both researchers to ensure correct data collection.

The research investigates low-code technologies across diverse industries within various organisational settings. These industries include but are not limited to healthcare, manufacturing, and the public sector. All data was collected in April 2024.

3.2.2 Participants, sampling technique, criteria, and size

The sample size consists of a total of nine participants, referred to as P-01, P-02, etc., in Table 2, and Participant 1, Participant 2, etc., throughout the text. Most participants work with Microsoft Power Platform, started using it a few years ago, and are part of organisations with more than 8 000 employees. For an overview of the participants, see Table 2.

The sampling technique used in this bachelor’s thesis combined purposive and snowball sampling. The reason behind purposive sampling is the goal of selecting participants with specific characteristics and unique experiences related to our research questions. In our case, the individuals considered suitable for providing rich and diverse data were professionals who led low-code technology implementation in an organisation (such as project manager, consultant, CTO, or similar) and who are in the post-adoption stages. For this bachelor’s thesis, we specified the post-adoption stage as the period during which the low-code technology has been implemented for at least three months in the organisation. Additionally, snowball sampling has been chosen since the participants we were interested in interviewing were not easily found and accessed. Therefore, due to time and resource constraints, we first identified a few initial participants who met the criteria and asked them to refer others they knew that fit the selection criteria.

Table 2. Research participants' overview

ID	**Role in the discussed organisation**	**Company size (approximate)**	**Year of low-code technologies adoption in the company**	**Type of low-code technology used**
P-01	Consultant / Product owner	150	2020	Microsoft Power Platform
P-02	Head of Data Application / Center of Excellence (CoE)	25 000	2021	Microsoft Power Platform
P-03	Customer Success Manager	12 000	2020	Smaller low-code SaaS platform
P-04	Project Manager	40 000	2024	Microsoft Power Platform
P-05	SharePoint Specialist	12 000	2022	Microsoft Power Platform
P-06	Consultant / Technical Lead	40 000	2020	OutSystems
P-07	Consultant / Technical Solution Architect	1 000	2018	Microsoft Power Platform
P-08	Microsoft Power Platform Owner	8 000	2019	Microsoft Power Platform
P-09	Consultant / Power Platform Architect	700	2023	Microsoft Power Platform

3.3 Method of Data Analysis

The method chosen for data analysis in this bachelor's thesis is thematic analysis, as it is recognised as a foundational method for qualitative data analysis, providing a good basis for both development of core skills useful for future qualitative analysis, as well as a flexible and useful research tool that leads to rich, detailed, and complex data (Braun & Clarke, 2006). As defined by Braun and Clarke (2006, p. 79), "Thematic analysis is a method for identifying, analysing and reporting patterns (themes) within data. It minimally organises and describes your data set in (rich) detail." A 6-phase guide adopted from Braun and Clarke (2006) is used for conducting thematic analysis, and the phases are as follows: (1) Familiarising with the data, (2) Generating initial codes, (3) Searching for themes, (4) Reviewing themes, (5) Defining and naming themes, and (6) Producing the report.

In line with the guide by Braun and Clarke (2006), we first transcribed the interviews and read them multiple times to gain a comprehensive understanding of the text. While becoming familiar with the data, we noted interesting points and made initial observations. Initial codes were generated separately by both researchers by highlighting specific phrases, sentences, or paragraphs that seemed relevant to the research questions. These initial codes were descriptive and directly tied to the language used by the participants, such as: *governance*, *decision-power*, and *key people*. The initial codes were then compared between the researchers and discussed when there were discrepancies. After all the initial coding was completed, a second round of coding was conducted by grouping the initial codes into categories such as: *stakeholder involvement*, *familiarity pre-adoption*, and *ownership sharing*. A total of 34 categories, descriptive but not always tied to the language used by the participants, were identified. Using mind mapping, the categories were connected based on patterns to visualise potential themes and their relationships. The initial themes were reviewed and examined to ensure they accurately reflected the data and addressed the research questions. Finally, five overarching themes were identified, which were then named in a descriptive and concise manner, capturing their meaning while differentiating them from other themes. These five themes represent the research findings and they are presented and explained in the coming Chapter 4.

3.4 Reliability and Validity

Reliability refers to the consistency and stability of the analysis, ensuring that the same results would be obtained if the study were replicated under similar conditions (Oates, 2006). To enhance reliability, inter-coder reliability checks were conducted, where two researchers independently coded a subset of the data to ensure agreement in the identification and interpretation of themes. Any discrepancies were resolved through discussion and consensus.

Validity, on the other hand, pertains to the accuracy and credibility of the findings, ensuring that the analysis accurately represents the domain from which we are gathering data in a rightful way and that it captures the perspectives and experiences of the participants (Oates, 2006). To enhance validity, Klein and Myers (1999) provided guidelines for conducting interpretive research in IS, and four key aspects presented in their work were considered and included in our methodological approach to ensure validity: (1) the hermeneutic circle, (2) contextualisation, (3) dialogical reasoning, and (4) multiple interpretations. In line with the hermeneutic circle and contextualisation, we included the social and historical aspects and focused on obtaining a holistic view of the topic. Considering dialogical reasoning and multiple interpretations during the analysis of our qualitative data, we reflected on our preconceptions and recognised different interpretations of the results.

3.5 Ethical Considerations

To ensure that the bachelor’s thesis is conducted with integrity, respect for participants' rights, and according to ethical guidelines, several measures were taken.

Firstly, all participants were informed about the purpose of the research, their rights to withdraw at any time, and any potential benefits associated with participation. An informed consent form was used to ensure all participants understood the aim of the research and agreed to participate (for the informed consent form used, see Appendix B). Only after the participant had signed the consent form was the interview conducted.

While no harm or discomfort was expected during the interviews, to ensure participants’ protection, confidentiality was ensured by not revealing participants' names during the interviews to other participants or in the bachelor’s thesis itself. Furthermore, information about the participants was not presented in a way that could allow the individual to be identified. If any question during the interview seemed to cause distress or discomfort, the interviewer acted with sensitivity and refrained from pressuring participants for an answer.

Finally, Whisper was downloaded and run locally on our personal computers instead of in the cloud, in line with Lund University School of Economics and Management's guidelines for processing personal data in student projects. The transcriptions were stored on our password-protected personal computers, and no information was shared with external parties.

4 Findings

4.1 Theme 1: Distinct Uses

The theme "Distinct uses" reflects the varied possibilities and uses of low-code technologies as reported by participants. Based on all interviews, it was evident that organisations use low-code technologies in different ways, depending on the perception and understanding of these. However, while the use varies across different organisational contexts, it can generally be categorised into two primary functions: (1) as a technical facilitator for fast and cost-effective software development, or (2) as a digitalisation and innovation enabler.

In the first use, the organisation uses low-code technology to build applications or systems in a more effective way, with the main benefits being faster and cheaper development or replacement of old legacy systems with newer, more modern technologies. As Participant 3 describes it: *“The difference that has been with these projects is they have been successful and delivered fast in just a few months. That's unheard of.”* Similarly, Participant 1 provides an example of a successful low-code technology use with time-related benefits: *“An MVP took six weeks to build. Then the next step was doing different extensions to the platform essentially.”* Participant 3 further underscores the focus on the technology as a facilitator, rather than as an organisation-wide, strategy-driven choice: *“I think there was an IT strategist on the regional level who was basically looking for quicker ways to implement it and [...] this initiative would have been too expensive in other platforms. [...] Get more stuff done for less money, I don’t think that’s the way they call it, but I think that's the main driver.”* Participant 7 describes the organisation’s reasoning behind changing legacy systems: *“[...] they're trying to get rid of all the machines and keep everything essentially in the cloud [...] and at least [organisation] is aware [...] they need to get rid of old systems and what can they replace them with and it makes sense to use low-code solutions.”* Furthermore, the barriers to developing new solutions are lower since developing is quicker, cheaper, simpler, and in many ways, fits really well with the existing systems that are already in place. As Participant 9 describes, *“[...] most of the clients, I think, they like that quite much, that they already pay for something in Microsoft, and then they can use it much more than from the start. So we use something that they already pay for.”* This way of using low-code technologies is mainly driven by time, effort and cost benefits, and focused on solving the organisation’s software development needs more efficiently.

In the second use, low-code technologies are being used to drive innovation and digitalisation across the organisation. The focus is on democratising software development, reducing reliance on IT, and optimising day-to-day processes. The organisation sees low-code technologies as a tool that should primarily be for non-technical uses, often referred to as citizen developers, to optimise their own processes and build apps that can help them in day-to-day work, rather than "bigger" applications or systems. As mentioned by Participant 9: *“And if someone says, ‘oh, it would be nice with the app that can be doing this’ and then maybe five hours later it will be done. And then a change will be done in 10 minutes. [...] It should be done quickly by anyone.”* This way of using the technology also includes coming up with ideas and new applications that the whole organisation can use and benefit from. Participant 3 describes the importance of citizen developers, who are close to the business processes, for achieving this culture of innovation: *"Being able to really use it to innovate [...], if you really want to use this, you want to get up speed. You want to produce more applications, want to do more things and also you want to be getting even closer to the organisation."* This way of using low-code technologies has as its main goal to foster a culture of innovation and encouraging all employees to be a part of the digital transformation process.

The two uses imply different primary users of the low-code technology. Organisations using low-code technologies as a software development facilitator consider them being something mainly used by IT professionals. Participant 6 provides a reason being that low-code technologies are not as simple as many believe: *"Well, they market it like that but when I, with 15 plus years of experience, when I tried out PowerApps, I didn't find it that easy."* While low-code usage is usually portrayed as easy for anyone, practitioners using technology in this way don't agree and prefer to do the development by either an internal IT team, or by a close external partner with whom the organisation is closely collaborating with. Furthermore, as described by Participant 9, when low-code technologies are not used by professionals, it can lead to bigger problems in the long run: *"People can do quite much damage if it gets too big. So if someone builds something, a very good app, and the whole organisation get dependent on it. [...] it can't break, I think that's the biggest fear sort of. Don't let anyone build something that they don't know what it will be. Because it will cost a lot of money. This app must work."* In other words, organisations use low-code technologies to solve their software development needs, often by outsourcing the service with the goal of creating high-quality solutions without extensive financial or time investment. On the other hand, organisations who use low-code technology to empower everyone to create digital experiences see them as a tool that should be available across the organisation. As Participant 2 puts it: *"If you understand the reason why low-code is there, then you should never aim for centralising this technology. Quite the opposite, [...] the principle number one for you responsible for this technology should be to actually make it accessible and available to as many people as possible."* This use emphasises not only the simplicity of low-code development but also promotes its use by end-users.

Furthermore, the two uses have varied implications for resource allocation, support from management, strategic planning, as well as internal structures, and depending on the use of low-code technologies, different benefits are realised. When using low-code technologies as a technical facilitator, organisational top management is not particularly interested in the technology itself, but more in what will be delivered, in what time, at what cost and quality, as described by Participant 6: *"Customers don't care what technology/platforms you're using but they are more interested in how fast the problem is getting solved. Low-code project works the same way as how you execute normal high-code project. You build user stories, you talk [...]. When customer sees the demo they don't realise that it's built with low-code [...]."* The organisational structures, processes and resources are either not affected at all by the low-code technology in use, or just minorly adjusted so the newly developed applications or systems can be maintained. In other words, the IT strategy is not built around the platform used and it is not prioritised when it comes to resource allocation. On the other hand, when using low-code technologies as an enabler of innovation and digitalisation, organisations focus more on building internal skills and resources, IT strategy around the technology, and are prepared to allocate more resources to it – both financial- and people-related resources. Participant 6 describes, *"It was more on the organisational structure change. They needed to get rid of vendors and then they've tried to utilise internal resources, be it employees or the tools,* *what they already have. It was more on high level.”*, exemplifying how an organisation decided to switch from outsourcing towards building internal resources after realising low-code technology’s potential.

However, no matter how it's used, the focus remains on understanding the different options and using the low-code technology according to the organisation's goals. As described by Participant 8: *“But that's also from organisation to organisation, very different how many users have been building, right? [...] Do you want to set it free so we become an agile organisation? Or do you want to be in control? Because you cannot have both on the Power Platform. You have to accept that there's something you cannot control if you do the whole citizen development thing.”* This highlights the importance of the strategic choices organisations face in relation to different uses of low-code technologies.

4.2 Theme 2: Appropriate Teams

The second theme identified describes teams that are directly assigned to the use of low-code technology within the organisation. These teams have specific roles and responsibilities and work with low-code technologies based on those. As Participant 9 described why having the right competencies is important: *“I think that even if you create a PowerApp and you're going to create a magic PowerApp, you still need to have a group that makes it finished. So, it will be a good product because you're making a product that will be an internal product. So, it needs to be correctly made.”* Participant 5 emphasises the importance of having assigned roles and responsibilities by saying: *“He's responsible for Power Platform. And what's part of that role is I would say kind of unclear. It's basically you take care of this and you do what you think is best. That's how they have. That's the directive of his assignment [...]. I would say you should have more clear objectives.”* However, these teams should be appropriate in relation to how the low-code technology is being used and what the organisational goal is in relation to the technology in use. As Participant 2 provides an example: *“Well, there are different personas or profiles, right? And based on the persona and the needs, then you're going to find different use cases. My team is responsible for the Centre of Excellence around the Power Platform. So we are also in charge of fostering the use of low-code technologies, Power Platform particularly, among regular people.”* Based on the data, two different types of teams are identified. The first type of team is responsible for developing digital solutions with the help of low-code technologies, and the second type of team is responsible for building structure and processes and supporting organisation-wide use of the technology.

The first type of team identified, which is more focused on the building aspects rather than fostering innovation among employees, is similar to traditional IT departments. However, additional competencies related to development using low-code technologies are present and seen as important. While the members of this team can serve as support to others in the organisation using the technology as well, the main purpose is on implementing the needed solutions rather than sharing knowledge and building a culture of innovation within everyone. As Participant 4 describes: *“You can quite easily create some sort of automation. [...] I'm not counting into this. I think if you need to build an application with low-code, then you need people that know what to do. It's not that easy. [...] So the right people need to work with it.”* This statement exemplifies the common view that low-code is not as easy to use, is often oversimplified, but can be a great tool for creating "bigger” applications and systems, which then requires specific expertise and knowledge about these technologies. This type of team is important for ensuring that solutions, often solutions directly related to the core business areas and widely used across the organisation, are done with high quality.

On the other hand, when low-code technology is being used for the purpose of innovation and digitalisation, an internal Centre of Excellence (CoE) team is crucial for several reasons. Firstly, it ensures the creation of guidelines everyone should follow, frameworks for the technology that is being used, necessary trainings for more people to use it, as well as proper governance. As Participant 8 provides an example of a structured learning way the organisation has implemented for a more secure and efficient citizen-development process: *"We have created something we're really proud of, actually, 'driving licences'. [...] we have recorded a few videos on our own, like this is best practice when you create a flow [...], sort of targeting the learning material to our users. I think it's super important. Otherwise they'll just drown in the material. [...] It's like trying to prevent that they build business functionality on top of templates without really understanding what's going on."* Similarly, Participant 2 describes a gated approach to citizen development: *"We want to have a gated approach, so we select based on interest, users that are already building a lot of things or people that have manifested somehow the interest on the (Power Platform) Academy. We select certain people to participate in it and then we offer them some trainings and licences as well and environments so they can learn and then they also have the proper access."* As Participant 7 mentions, "*Not everybody needs to be a citizen developer, nor everybody wants to be one"*, indicating that the empowerment still requires an internal team who will be in control and structure the usage of low-code technologies.

The use of the term fostering is particularly noteworthy, as it highlights the team's role in guiding proper use, as well as creating a culture of innovation that empowers employees on all levels to learn and build using low-code technologies on their own, with the support of an internal team. Participant 8 provides an example of how supporting citizen developers' knowledge can be done, *"[...] we have put a lot of effort into having a community, ask us anything. [...] So we sit and listen on that channel and try to help. 'I need to do this' - You can look here. And then there's, we have a monthly coffee on the Power Platform, where we introduce subjects. Normally we pitch four subjects and people can vote what they want to hear about."*, reflecting on the importance of helping citizen developers to understand low-code technologies and use it in the proper way. Participant 3 describes the collaboration between citizen developers, who are close to the operational processes, and the internal team that provides guidance and support: *"Maybe they don't have to build a fully functioning delivered product in the start. They can just use a lot of proof of concepts and then they could take it to a partner and say like 'okay we built something here in in the platform. What do you think about this? Could we realise this into a fully functioning application?'."* Similarly, Participant 8 agrees and describes why this support is important: *"So the obstacle I'm very focused on is, because you could start building and then it gets difficult and then you stop. And that's where we need to be. So in a supportive way of course, right?"* However, while support roles are important, the ownership of the solutions should still be on those "ordering" and using the solutions itself. As Participant 8 mentions: *"So for instance, if we have somebody that needs help, we can sort of go out and support their work as a project. Help them build it. And then when it's done, we can leave again. So now it's your responsibility.*" When low-code technology is being used as an innovation and digitalisation enabler, the support teams are crucial, but the ownership of the process lies with direct users.

Finally, irrespective of the type of team in question, low-code initiatives are considered much more successful if the teams are led internally within an organisation. Participant 1 mentions the importance of sharing the ownership and building capacities internally so that time and resources are allocated appropriately to support low-code utilisation: *"So you would want to anchor it into the organisation. [...] So if you say, hey, let's have just a meeting with some chocolate every week, essentially, just to talk about what we want to do. [...] And a lot of people share the responsibility. And then if you do leave at some point, then you have a whole group of people who have seen everything you're doing at the same time."* Furthermore, as mentioned by Participant 3, having internal resources allows organisations to lower the barriers to development even more by quickly building minimum viable products and testing them, rather than depending on external partners: *"I'm guessing if you would have it internally, it would be an easier process where you just employ your own organisation into building [...], you just need a project organisation that kind of keeps track of the quality, [...] they can kind of lower the barriers even more."* Participant 5 agrees and argues there are financial reasons behind investing in such teams: *"If we have 10 000 hours (of work), that it would take to create those. But we can save 10 000 hours each year approximately. Then even in these times, when we have layoffs, we can still show that then we can probably have a business case to recruit more people or buy it from consulting companies for example. That would be great."* However, not all organisations need, want, or can invest in internal teams. As Participant 3 mentions: *"I don't think they have the organisation to build that up. They probably would need somebody more like a technical project leader or somebody who owns the initiative. I don't think they have the competencies, but I don't know if they really want to as well."* There is a common similarity among the participants' experiences, that although internal teams are crucial for driving internal projects and assisting citizen developers in software development, there has to be support and commitment from the organisation to allocate necessary resources for these teams. As Participant 5 shows dissatisfaction: *"I have told them a lot of times and they have said: 'We'll do it in the future'."* Similarly, Participant 1 states: *"But then when push comes to shove, then it's, as you say, just getting a meeting every two weeks with a little group was always, 'oh, no, I am on holiday' but I'm actually online. Or I'm doing this. Or we don't have time. Or we have so much to do. Or it's the summer. Or we're doing other things. Then nothing's going to happen. It doesn't happen by itself."* Without this kind of support, these teams lack motivation, structure, and clear leadership guidelines.

4.3 Theme 3: Solution Catalysts

The third theme that emerged from the data is related to the individuals within the organisation who understand the specific organisational needs and drive the use of low-code technology. These individuals have knowledge about the low-code technology adopted in the organisation and play a key role in identifying and pushing the new projects forward. However, solution catalysts differ based on how the low-code technology is being used within the organisation.

In organisations that use low-code technology to solve organisation-wide needs more effectively, this internal driver is a person close to the business who has a specific role and responsibility to actively seek potential good projects that align with larger, more strategic organisational goals. As Participant 1 describes it: *"So if you work in Agile teams and you have a business analyst who figures out, this is what we need to do. Or this is a wish from the organisation. Then it goes to the product owner and says, 'cool, we're going* *to do these things in this order'."* Participant 3 provides examples of strategic reasons behind having such a role: *"[...] that's probably more on a higher level, like the bosses and the administrators that work in the organisation. They probably look at how much things they can kind of automate and like remove tasks and make the tasks easier but then on a strategical level they probably see like how this can push the whole organisation to another thing, like start to get things done."* Participant 8 even argues that there should be a team responsible for this: *"It would be great having a team consisting of cross-functional resources. [...] So I go sit with you, with my IT architect, with my Power BI guy, with whatever .NET we need. And then we build something."* In other words, the IT department can use low-code technologies to build actual solutions (applications or systems) in a more efficient way, but the individuals who find potential projects and organisational needs, and bridge the communication gaps between technical and non-technical teams, play a pivotal role in leveraging the low-code technology in a way that supports business needs and strategic objectives.

Additionally, using citizen developers as solution catalysts is identified as successful for driving innovation and digitalisation. In this case, solution catalysts are multiple individuals who both understand the technology, see the need on a more practical, day-to-day level, and have the decision power to test the solution directly. As Participant 2 provides an example of solution catalysts across the organisation: *"Then you have people like in finance, HR, operations, you name it, that are using the platform because it's accessible to everyone."* The importance of solution catalysts in all important parts of the organisation is evident in an example provided by Participant 5: *"Then basically, here we have someone who knows something about this, that office will start using it quite a lot, but other places don't have that. I think they might not have tested it yet."* Participant 8 underscores the importance of just citizen developers as solution catalysts: *"Because that's end-users that sit and do the same thing again and again and again and again. And they know exactly what they would like to do, whereas me as the technical expert got no idea what they need to do. So instead of having that gap, then it's about giving the tools to them."* Overall, participants highlight the role of non-technical users in identifying business needs, conceptualising potential solutions, and actively participating in digitalisation efforts.

The most important aspect of these individuals, regardless of the technology used, is that they understand the organisations' needs and can identify where the potential for improvement exists. The importance of being close to the actual problem or need is further highlighted by Participant 1: *"As close to the actual operation as possible. So that customer service really gets to say, 'We think these data points are the most important. So can we focus on those?' And then I think everybody should say 'yes'."* Another crucial aspect is that they understand what possibilities the low-code technology adopted has and if the project in mind is an appropriate choice for that technology. As mentioned by Participant 5: *"The flow takes 14 to 15 hours every day so it's getting too complicated. If I knew that then I would recommend more, there's another solution that's more, not low-code. [...] It started like, this is simple and now it's huge. And it's really, really hard to do changes if you don't know exactly what you're doing."* This theme highlights the crucial role of the individuals with the right knowledge and information in driving low-code technology utilisation.

4.4 Theme 4: Low-code Technology Readiness

The fourth theme identified, Low-code Technology Readiness, relates to the extent to which an organisation is equipped to effectively utilise low-code technologies. As Participant 2 describes: *“It was adopted, it was there but there was not a centralised team doing an adoption or managing. It was kind of a wild west, no one was accountable for it so that's the reason actually why they created a position that I originally took here.”* This theme highlights the need for clear structures, roles, and processes being in place within the organisation for the low-code technology to be assimilated across the organisation. As Participant 5 describes, *“We have this solution, we can help you build the forms, we can automate stuff. But then we need to have organisation, we need to have processes, we need to have that before we can roll it out. [...] We need to really mature in IT before.”* Participants mention several aspects that an organisation that is fully ready for low-code utilisation has, such as governance, lifecycle management, understanding project limitations and scope, stakeholder involvement, as well as best practices put in place before any development starts, no matter if it is done by experienced developers or citizen developers.

One of the aspects mentioned and highlighted repeatedly is governance. While low-code technologies enable faster and simpler development, traditional IT practices, such as governance, remain crucial to ensure high quality and secure solutions. As Participant 8 says: *“The governance part and licence part and capacity usage part belongs in the backbone IT.”* Participant 4 agrees and describes: *“That's also an area I think many companies are missing out on. They are kind of ‘We got these tools, let's go!’, but you need to have a bigger strap around it, and you need to have the governance side of everything in place before you start. Because if you start, then it will be applications that will not be used or very unsecure or they will die away. Yeah, it will not be a good experience.”* Moreover, Participant 3 describes the similarities between low-code and traditional development, *“I think when it comes to low-code technology, it aids you in working, like with fast sprints and fast cycles, but you can still make the same mistakes as you do with regular development”*, emphasising that low-code technology is just a set of tools that enables fast development, but still faces standard project risks and requires careful governance, strategy, and testing. As Participant 6 describes: *“It's a normal IT development life cycle but when you're working with low-code you should be extra cautious and be ahead in getting clarity in requirement because it's not like a high code [...]. You should prepare the initial phase [...] in detail because the moment you say it, within weeks the things will be developed [...].”* This underscores, once again, that a low-code technology is just an efficient tool, which still requires standard project management and development practices for correct and safe usage.

Ready organisations also have an in-depth understanding of the technology and what it can and cannot be used for. In the case of low-code technologies, this is mostly related to the complexity and size of the projects. As Participant 3 mentions: *“I think, it's not really a technology problem. It's more of a project problem.”* Participant 5 adds: *“And I also build quite big solutions based on that, but that's not what it's meant for. I think that's one of the key knowledge points, what should you use it for and what should you not. When is it good, when do you need to just write your own code that's more simple than using a low-code solution.”* Participant 3 further explains the importance of using low-code properly for it to be successful: *"And I think that was done according to best practices when it comes to low-code. You need to keep it small and deliver fast in the first application."* The project type needs to fit the low-code technology capabilities to ensure effective utilisation of resources, and this is knowledge that an organisation needs to have beforehand. Participant 6 describes how they as a consultancy service make sure these aspects are taken care of: *"They (the client) need to verify a couple of criteria for the requirement, whether it is for only internal usage, and how big project is, how complex, what are the security aspects [...]."*

Additionally, to be able to use the technology in the most strategic way, there has to be governance from the beginning as well as an understanding of what these technologies can do and how the organisation wants to use them. Participant 9 states: *"At the start, it would be to gain control before you lose control."* Participant 8 agrees and emphasises the importance of ensuring clear alignment with the higher leadership goals: *"Of course, set up all the ground rules. [...] Install the CoEs, and then start immediately with cleaning up all the old stuff. [...] And then, if possible, get a leadership guideline on. Do you really want to support this?"* Even though low-code technologies promise simplicity and ease of use, these factors are seen as crucial for ensuring that the technology is used for the right projects, aligned with the strategic goals, and managed from the beginning.

If the organisation aims at fostering citizen development, unique factors such as training, policy creation, and support systems are all parts of a ready organisation. This is important for proactively tackling future organisational challenges, instead of allowing everyone in the organisation to use the platform in any way and solving issues afterward. As Participant 2 describes: *"[...] a more balanced system where you have what I call it 'democracy' and it's basically someone is in charge, naturally, because then someone needs to make the calls. [...] we need to ensure that there are proper and good rules of the game, people know how to behave right, people know what are the limits what are the responsibilities, [...] we call it Power Platform Framework."* This aspect involves defined and specific roles with areas of responsibility, which ensure that, while the organisation encourages citizen development, there are still clear guidelines that need to be set in place before it is accessible for anyone. As Participant 8 provides an example of encouraged citizen development with proper governance:

> *And we have configured our CoE with our own customisations. For instance, if we talk about business justification and to know what it's on, we actually just enable a functionality where we, through Teams approval cards, will reach out to you. It says, 'Okay, you have built this six months ago. You have not used it. Do you need it? Or can we delete it?' [...] But if you say, 'Oh, you cannot delete it', then you'll get another Teams card where you have to provide us with information on what is it. And then we have a weighted business score on the business justification. And if you score high, then I'll put in my diary. I need to talk to you. What you are building is, you have said it's super critical. So we need to go through a documentation process. You need to contact the data security office.*

Participant 5 further highlights the importance of citizen developers knowing the best practices: *"But if you send it (data) to other companies and customers, then you should have a testing environment. That's good practice."* Participant 8 adds: *"In my opinion, I would rather create too many environments because then I can delete instead of having* *too few environments because then you are sort of stuck if you have several users building in the same environment."* This aspect highlights the importance of an appropriate testing environment for ensuring the usability and functionality of applications before deployment, making it more difficult to remedy or adjust after an application is already in production. Fully ready organisations not only use low-code technologies across the organisation but seem to have achieved a balance of control and empowerment, as well as understanding that low-code technologies still require standard development practices.

4.5 Theme 5: Value Realisation

The fifth and final theme identified from the data is related to the capability of all levels of the organisation to recognise the value created by low-code technologies. While the IT department usually sees the value, top management and everyone else in the organisation have a harder time with it, making it less likely for the whole organisation to reap as many benefits as possible.

Participant 4 argues for the importance of everyone understanding the reason behind the technology being used in the first place: *"Get people to understand what we are talking about or what they are talking about. I think it's very important to kind of give the information, why are we doing this? Why?"* This view is highlighted by Participant 1 as well, who mentions the importance of all levels knowing the value behind the low-code technologies: *"So I think you have to sort of attack it from both angles. Think about who you need to talk to at which point to really make that work so that everybody works with it."* Value realisation as a theme focuses not only on spreading awareness but also on actual stakeholder engagement, which leads to cultural changes and higher commitment across the organisation. Participant 8 agrees and states: *"It's the whole mentality about it that we need to support it. But supporting it is not only telling users here are the tools. It's also teaching them how to use it and teaching users to respect that."* This theme is characterised by low-code technologies being recognised across the organisation as effective for supporting organisational needs as well as everyday operational needs.

Two ways identified that facilitate top management's awareness of values created are: (1) showcasing proofs of concepts, and (2) providing metrics on financial or strategic benefits. As described by Participant 4: *"So when I was talking to the program management and how to get it on board, [...] I didn't talk a lot about low-code or applications. [...] And I didn't get any pushbacks at all because I think everybody understood and realised that when I exemplified [...]."* Participant 2 adds: *"[...] build very early and showcase what was possible, [...] showcase it with something concrete that people can actually use to wrap the concepts."* Additionally, the initiatives that include top management or strategic goals directly are more prone to be seen in a positive light. As Participant 3 mentions, *"We worked together very closely with the case handlers or grant handlers, the people, the administration, basically, we had some more senior, the boss was in there too"*, underscoring the positive impact the inclusion of higher management in driving new projects can have.

When the technology is already in use, the value created is often recognised by those who work with the new app or the new flow. As Participant 1 describes: *"Get the feedback, get responses. And then you usually get questions: 'Can I also do this?' And then you say,* *‘Well, not yet, but we can make it do that. Why does that help you?’ And they say: ‘Well, that makes my day this much easier over the course of a week’. Okay, great, fantastic. Then we have a benefit from it. Then I'll make that.”* However, while understanding value by those directly affected is often easy once tested, understanding of the process behind it isn't. As described by Participant 1: *“I think they saw the benefit, but they didn't know the methods. I mean, people who work in a call centre are not trained in IT development. Why would they be? I mean, I work in IT, and I'm not trained to work in a call centre either. I would be terrible in their job.”* However, as described by Participant 1, realising value created in combination with understanding the processes behind it allows for better involvement, better use of the technology, and innovation: *“You share ownership, and then you get the possibility to grow in one aspect of it. Instead of me doing everything, I can then teach someone to do the actual development, and then I can teach the organisation at the same time how to think agile, in a sense, how to write a user story, how to write a request, how to change management, that sort of thing.”* In other words, by more people and organisational levels seeing value in low-code technologies, as well as understanding why and how it works, more uses of the technology are being generated.

5 Discussion

5.1 Overview

The previously described themes, which emerged from the data analysis and represent the research findings, answer our research questions. More specifically, Theme 1 answers the first research question, while the rest of the themes help answer the second research question. For an overview of the themes and their relation to the research questions, see Table 3.

Table 3. Overview of the found themes in relation to the research questions

Research Question	Theme
RQ1: How do organisations use low-code technologies post-adoption?	Theme 1: Distinct Uses
RQ2: What factors affect the assimilation of low-code technologies?	Theme 2: Appropriate Teams Theme 3: Solution Catalysts Theme 4: Low-Code Technology Readiness Theme 5: Value Realisation

The discussion is organised around the posed research questions. The findings that reply to each research question are then interpreted and discussed in light of both the literature review and resource-based theory.

5.2 How do Organisations Use Low-Code Technologies Post-Adoption?

Based on our findings, there are two distinct uses of low-code technologies within organisations. The first entails employing low-code technologies as a technical facilitator for fast and cost-effective software development and is primarily done by the IT department. This finding supports research by El Kamouchi et al. (2023) and Rokis and Kirikova (2023), who found cost, time, and effort savings as some of the most significant drivers of low-code technology adoption, indicating that some organisations aiming for cheaper, faster, and simpler development use low-code technologies as an appropriate solution. In line with studies by Aveiro et al. (2023) and Chuanjian et al. (2023), which provide evidence for faster development processes enabled by low-code technologies, we show that organisations using these technologies as a technical facilitator do indeed see the aforementioned benefits. These findings also confirm research by Ciucan-Rusu et al. (2023) and Brunninghaus and Rocker (2022), which showed that organisations use low-code technologies to quickly address their own development needs without the need for long and complicated development processes.

At the same time, the findings highlight that, while most organisations see these benefits, many have a perception of low-code technologies as too complex for everyone to use. This finding supports research by El Kamouchi et al. (2023) and Käss et al. (2023b), which found the complexity of low-code technologies to be one of the common challenges. This could potentially explain why some organisations decide to limit their low-code technology use to the IT department. Additionally, we observe that organisations that focus more on the technological facilitation use often see low-code technology as a tool that can help them solve main business issues in a more effective way, as well as leverage existing systems and licences. This supports findings by Pacheco et al. (2021), who showed that by using low-code technology, the development team was able to collaborate more effectively with the design team and ultimately create more value for their customers, which was their main business process. These findings further confirm research by Käss et al. (2023a), in which the authors explored what leads to continued use of low-code technologies after the initial adoption and found three archetypes as potential causes. In line with our findings, the Synergy Realiser archetype was someone within the organisation who promoted leveraging already existing systems or licences, and IT Resource Shortage Mitigators aimed at addressing limited IT resources by using a tool that can facilitate development (Käss et al. 2023a).

Based on the findings, the second use of low-code technologies entails driving digital transformation and fostering a culture of innovation within an organisation. This finding confirms results from Rokis and Kirikova's (2023) literature review, which show that low-code technologies not only allow time and cost efficiencies but also foster employee involvement and digital innovation through learning, experimentation, and creativity. This strategic approach to leveraging low-code technologies to nurture innovation and empowerment within organisations also supports research by Redchuk et al. (2023), which emphasises the pivotal role of citizen developers in driving organisational process optimisation. Furthermore, the findings further confirm the third archetype identified by Käss et al. (2023a) – The Application Development Democratiser archetype. This archetype was identified by the authors as crucial in promoting the idea that individuals without extensive programming knowledge can develop applications (Käss et al. 2023a).

Finally, the insights from our research on the two distinct uses – facilitated development or digital innovation and empowerment – confirm previous research on low-code technology and extend it by underscoring the importance of aligning the use with organisational strategic goals and operational needs. This supports scholars' views, such as Roberts et al. (2023) and Martínez-Caro et al. (2020), who argue that IT assimilation has the potential to improve organisations' operations and overall competitiveness but highlight the importance of an early-stage strategic deliberation, during which *if* and *how* specific technology should be used is discussed.

5.3 What Factors Affect the Assimilation of Low-Code Technologies?

The findings present several factors and their interaction as influential in the assimilation of low-code technologies within organisations. Central to this process is the presence of appropriate internal teams, equipped with specific skill sets tailored to the organisation's needs. Our findings reflect one of the important principles of low-code development identified by Rokis and Kirikova (2023, p.79), named by the authors "Empower citizen developers and establish a fusion team's approach”. Similar to our results, these scholars highlight the importance of teams that help citizen developers create more advanced solutions and serve as invaluable expertise in various software development aspects, such as security and risks. From our research, we confirm that no matter which way the low-code technologies are being used – as a technological facilitator or as digitalisation and innovation enabler – appropriate internal teams with specific skills are necessary and extend it by showing that two different types of teams are important, and which one is more important will depend on the use of low-code technology within the organisation.

In cases where low-code technology is used as a technological facilitator, the internal team consisting of individuals who have the relevant skill set and experience with the low-code technology is found to be very important. This idea, that using low-code technology is not easy and requires knowledge, supports research by El Kamouchi et al. (2023), who found that, despite low-code technologies often being presented as very simple and easy to use, building robust and scalable applications requires skills and proper planning and execution. Furthermore, Alamin et al. (2023) mention the importance of vendor support that should offer guidance, support, and training for smooth technology transition. We find support for these findings by showing that just because an IT department exists in the organisation, it does not mean the knowledge about low-code technologies is present per se, and there should be specific technical skills present in order for the technology to be used in the best way.

On the other hand, our findings show that when low-code technology is used as a digitalisation and innovation enabler, the internal team consisting of specific roles who are in charge of training, providing support, and governance might be more relevant. This finding supports previous research done by Rokis and Kirikova (2023), who found that establishing governance is of great importance when citizen developers are involved. According to the authors, without proper governance, there is a risk of neglecting security aspects (Rokis & Kirikova, 2023).

The findings further confirm research done by Käss et al. (2023a), who use the term champion to describe an individual within an organisation who is close to the business and its processes and has already successfully used low-code technologies. According to the authors, this person can then act as support to others who want to use it and, in such way, spread the word for further use of low-code technologies. The findings confirm the importance of such individuals and extend it by offering a more nuanced understanding of various roles with different contributions. Based on our research, organisations that want to assimilate low-code technologies, i.e. use it to improve development processes, digitalise work, and drive innovation, but in a more controlled way, can do this by implementing a proactive way of identifying the needs. This could be done by assigning an official role, such as a Business Analyst, who can actively look for needs across the organisation and then create projects for the IT department. On the other side, if an organisation seeks to assimilate low-code technologies in a more democratised way, citizen developers who have the interest and understand the need within their own area of work could be given the authority to build said solutions. In these scenarios, since these individuals would only work with needs specific to their area of work, it is important to have multiple such individuals across the organisation.

The findings on the organisation's readiness for the integration of such technologies support several previous findings. Firstly, the findings support research by Käss et al. (2022), which showed that, while low-code technologies do have plenty of benefits, compatibility, interoperability, and the creation of a wide range of functionalities are often seen as challenges. Secondly, Käss et al. (2023b) found a lack of low-code culture and reluctance to change as significant inhibitors based on the practitioners' perspective. While these scholars don't specifically mention low-code technology readiness, these challenges being often mentioned as critical for successful use imply that the organisation should be well aware before the implementation whether the low-code technology is the right choice for their organisational needs. This view aligns with the scholarly perspective of Denning (2023), highlighting the importance of strategic leadership involvement on all levels of the organisation and readiness for digital integration. Our findings support this perspective by showing that assimilation of low-code technology needs more than a mere tool implementation; it requires a change in strategy and organisation-wide commitment to encourage innovation and gain a competitive advantage.

The findings from our research on the importance of using the right project for low-code technologies in relation to its complexity and size closely align with the research by Käss et al. (2023a), which identified one archetype characteristic for organisations that do not pursue further low-code utilisation after initial adoption – The Intricacy Adversary Archetype. According to Käss et al. (2023a), when the application that should be developed is perceived as too complex and sophisticated, it leads to organisations deciding not to use low-code technologies for its development. Our findings extend this by showing that this can be seen as a positive aspect of an organisation, as it shows the realistic understanding of both capabilities and limitations of low-code technologies.

Interestingly, our research reveals that organisations see low-code technologies as secure, cost efficient and customisable solutions. However, this finding is not supported by previous research. El Kamouchi et al. (2023) identified those aspects as potential barriers to adoption. However, these contradictory findings could be understood from the organisation’s readiness perspective. Organisations who do not know enough about the specific technology might see challenges that do not exist, and therefore would not be ready to assimilate it in the best way or would adopt the technology but neglect the important aspects such as governance.

Stemming from the findings, low-code technology governance is seen as a crucial factor, but not one that is necessarily easily established. This is in line with the research by Alamin et al. (2023). The authors found that the topic of low-code technology maintenance constantly grows each year, as analysed through the number of Stack Overflow posts (Alamin et al., 2023). Within our research, their conclusion about the growing interest, awareness, as well as issues in relation to successful governance is supported and extended by showing that governance is something that should be done first, before any development even starts, rather than as a secondary action, especially in organisations where citizen development is encouraged.

In line with Martínez-Caro et al.'s (2020) perspective, which argues that for successful assimilation of IT, active involvement and support from managers are essential, are research reinforces the importance of recognising value on all levels within an organisation. Our findings also confirm that, while this value is usually understood by the IT team, it is of great importance that the whole organisation understands the capacity, potential, and the motivation behind the use of low-code technologies. This further supports research by Martínez-Caro et al. (2020), which found that active involvement and support from top management are important because they ensure employees’ perception of IT as useful in their work processes. The authors further mention the culture of innovation, which contributes positively to technology assimilation and promotes an organisation-wide commitment to its use (Martínez-Caro et al., 2020). The findings on the importance of everyone being involved in the process support research by Wang et al. (2019) as well. The authors emphasise how involving everyone, especially top managers, is crucial in organisations resistant to change and suggest practical solutions like help desks, training programs, and policies, which in combination with strong support from top management, help change efforts succeed (Wang et al., 2019). Similar to the results found by Gao et al. (2019), we highlight the importance of aligning top management and IT resources. Based on our findings and these previous studies, synergy between top management and other IT-related resources seems vital for assimilating technology across the organisation and allocating resources strategically.

However, when looking into factors that influence the adoption of low-code technology, Performance Expectation, Effort Expectation, Social Influence, Perceived Risk, and Perceived Cost are found to be essential and shape decision-making related to low-code technology adoption (Razak et al., 2024). Our research findings build upon this research, showing that while these factors are indeed important, we find that, for long-term value realisation, the motivation should be different and more related to the strategic benefits that can be achieved, rather than benefits related to cost and effort. Similarly, Käss et al. (2022) find that the most frequently discussed drivers for adopting low-code technologies are the technologies' ability to enhance software development efficiency, lower the entry barriers for non-skilled developers, and the overall simplification of the development processes, which indicates there is a high chance that many organisations focus more on the short-term benefits, rather than the long-term strategic understanding of what value the technology can bring. In our research, we find a reflection of Pinho et al.'s (2022) perspective, which reinforces the importance of understanding the usability of these technologies across the organisation, and an organisational mindset that doesn't just treat low-code technologies as an additional tool, but as a strategic choice.

5.4 Low-Code Technology Assimilation from a Resource-Based Theory Perspective

Since IT assimilation is a multifaceted process that involves alignment and organisation of various aspects (Caldeira & Ward, 2003), it is important to understand the relationships among the identified factors. Our findings on low-code technology assimilation – as a process through which organisations leverage their internal resources, capabilities, and competencies to effectively integrate and exploit low-code technology investments – are in line with the previously mentioned view of IT based on resource-based theory. The emphasis is on the importance of aligning a variety of tangible and intangible resources, such as technology-related processes, human capital and teams, and organisational culture, with organisational goals to achieve assimilation. This is consistent with the resource-based view of IT/IS, as it mirrors Peppard & Ward's (2016) view, which states that organisations need to organise resources in a way that leverages new IT adoption and achieves sustainable competitive advantage. Drawing from the resource-based theory, which emphasises the strategic management of resources to achieve sustained competitive advantage (Barney & Clark, 2007), we can analyse how various factors contribute to organisational effectiveness and innovation.

An important factor for low-code technology assimilation is the presence of appropriate internal teams with specific skill sets. These teams serve as the backbone of the organisation's technical skills, who can leverage their expertise to guide others' use and navigate complexities related to low-code technologies. This is aligned with the resource-based view of IT/IS, as the expertise and knowledge possessed by these teams represent valuable resources, enabling effective utilisation of low-code technologies and enhancing the organisation's structures that support low-code technology assimilation (Caldeira & Ward, 2003). However, team effectiveness is not isolated; instead, it is linked to the presence of solution catalysts – individuals who promote innovation and drive change within the organisation. These catalysts reflect the importance of human capital in leveraging low-code technology. From a resource-based perspective, these individuals represent valuable human capital who understand the organisation's needs and processes and drive improvements, thereby contributing to the organisation's adaptability to change and constant improvement. Yet, from a resource-based theory perspective, the organisation’s collective capabilities are crucial (Barney, 1991). This is in line with the low-code technology readiness factor, which relates to organisational capabilities that determine the organisation's ability to effectively assess low-code technologies and plan the strategic use of the technologies that can aid competitive advantage. The presence of value realisation aligns with the resource-based view of IT/IS, since the recognition and realisation of value at all levels within the organisation require active involvement and support from top management, as well as a culture of innovation and commitment to technology adoption (Peppard & Ward, 2016). From a resource-based perspective, these organisational values and culture represent intangible resources that contribute to the organisation's competitive advantage by fostering innovation and enhancing organisational effectiveness (Caldeira & Ward, 2003).

Overall, our findings suggest that the assimilation of low-code technologies is influenced by a combination and interaction of organisational resources. The emphasis on these aspects as organisational resources aligns with the resource-based theory, which underscores the importance of valuable and rare resources within organisations (Barney, 1991). Research by Mata et al. (2003), which shows that creating competitive advantage from IT/IS long-term mainly relies on internal organisational factors – how IT is managed and organised - finds validation in our study. An organisation that leverages its resources to create teams, solution catalysts, organisational structures, and a culture of innovation affects assimilation, and in such way gains competitive advantage through the strategic use of low-code technologies, which is otherwise difficult to achieve.

At the same time, analysis of different uses of low-code technologies post-adoption from a resource-based theory perspective provides insight into what uses could be considered post-adoption assimilation, and what could be considered just mere adoption. According to the resource-based theory, using low-code technologies to facilitate development might create value, but not necessarily a competitive advantage, unless development is the core business area (Caldeira & Ward, 2003). On the other hand, using low-code technologies to drive innovation, optimise business processes, and adapt faster to market changes might be seen as a way for organisations to use the technology to create a competitive advantage (Mata et al., 2003).

5.5 Research Outcome

Our research explored low-code technology assimilation within organisations and identified multifaceted factors that are interconnected and interact with each other in various ways. Factors contributing to the overall process of low-code technology assimilation within organisations were identified, emphasising the importance of appropriate internal teams, solution catalysts, low-code technology readiness, and value realisation. Figure 2 presents a model synthesising the findings and illustrating low-code technology assimilation from a resource-based theory perspective.

Based on our findings, there are two primary uses of low-code technologies: (1) facilitating fast and cost-effective software development, and (2) driving digital transformation and innovation. It is essential for organisations to understand which, or if maybe both of these uses align best with their overall strategy and objectives. As illustrated in our model, organisational readiness in relation to low-code technologies is an overarching necessity and a process involving several aspects.

Firstly, before crafting the strategy, organisations must thoroughly understand the low-code technology and its possibilities and challenges. Secondly, once the organisation makes a strategic decision to use low-code technologies in a certain way, top management must be engaged and allocate resources accordingly to support the chosen use. Developing an IT strategy is then tailored and aligned with the low-code technology goals and the business needs.

The most important aspect developed parallelly with the strategy is the formation of appropriate internal teams and solution catalysts. Internal teams with specialised skill sets aligned with the chosen use may focus on development tasks or serve as a Centre of Excellence (CoE) responsible for governance, training, and expert support. Additionally, identifying solution catalysts, individuals who drive innovation and understand organisational needs, is crucial. The collaboration between the two is important to note, with internal teams providing technical expertise while catalysts bridging the gap between technology and business needs.

Value realisation is integral at every stage of the process. Once the low-code technology strategy is implemented, continuous value realisation becomes an active process. This process influences top management's perspective, resource allocation, and the organisational culture of innovation, which in turn helps build engagement on all levels, commitment to the low-code technology in use, and reaping additional benefits.

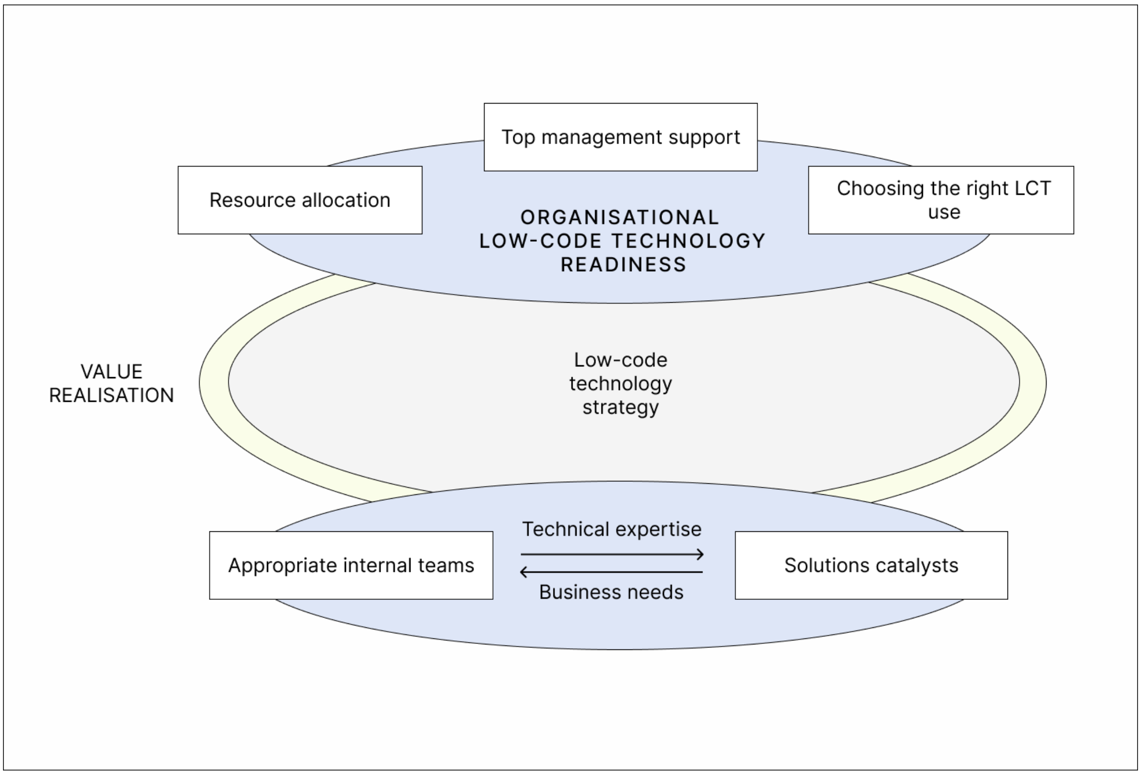
Top management support
Resource allocation
Choosing the right LCT use
ORGANISATIONAL LOW-CODE TECHNOLOGY READINESS
VALUE REALISATION
Low-code technology strategy
Technical expertise
Appropriate internal teams
Solutions catalysts
Business needs

Figure 2. Model of low-code technology assimilation from a resource-based theory perspective

From the perspective of resource-based theory and our research findings, aligning the use of low-code technology with organisational strategic goals and operational processes can enable organisations to reap more benefits and gain a competitive advantage. The outcome of this research provides practical insights and recommendations for organisations seeking to assimilate low-code technologies effectively:

1. In-depth understanding: Organisations should prioritise a thorough understanding of low-code technology, including its capabilities and challenges. This understanding should be aligned with the organisation's strategic goals to ensure alignment with how the technology is being used.
2. IT strategy adjustment: Organisations should tailor the IT strategy and operational processes to accommodate the chosen use of low-code technologies. Active involvement and support from management should be prioritised to ensure necessary resource allocation.
3. Communication of objectives: Organisations should foster a culture of innovation and commitment to the use of low-code technologies by effectively communicating the objectives and value of the initiatives across all levels of the organisation, as well as supporting organisation-wide learning.
4. Forming of specialised teams: Organisations should establish specialised teams equipped with skill sets and responsibilities aligned with the chosen use of low-code technologies. These teams should be structured to efficiently navigate the complexities of technology integration and utilisation.
5. Empowerment of solution catalysts: Organisations should identify and empower solution catalysts within the organisation – individuals who possess the expertise to drive organisational process optimisation and innovation. Empowering these catalysts can fuel creativity, problem-solving, and ensure strategic utilisation of low-code technology implemented.

6. Value realisation: Organisations should continuously assess the realisation of value derived from low-code technology assimilation using financial, practical, and strategic metrics. Allocating additional resources as necessary can be important to support the assimilation process and ensure that strategic objectives are met effectively.

By implementing these recommendations, we believe organisations can streamline the assimilation of low-code technologies and leverage them as strategic assets to drive innovation, enhance efficiency, adaptability, and build competitive advantage.

6 Conclusion

6.1 Conclusions

This bachelor's thesis aimed to deepen our understanding of how organisations use low-code technologies post-adoption and to offer practical insights for practitioners seeking to maximise the benefits of low-code adoptions. To address our research questions – "How do organisations use low-code technologies post-adoption?" and "What factors affect the assimilation of low-code technologies?" – we utilised a qualitative approach. Based on the thematic analysis of nine semi-structured interviews, we identified five key themes: Distinct uses, Appropriate teams, Solution catalysts, Low-code technology readiness, and Value realisation.

Regarding the first research question, the findings identified two main uses of low-code technologies: enablement of fast and cost-effective software development, and driving digital transformation and innovation. The former enables organisations to streamline development processes, achieving cheaper, faster, and simpler development. The latter empowers organisations to foster a culture of innovation, experimentation, and creativity, and drive organisational process optimisation. These identified uses underscore the strategic importance of understanding low-code technology capabilities, organising resources effectively, and aligning resource allocation with broader organisational goals. Regarding the second research question, the remaining findings and their relationship were identified: Appropriate internal teams, Solution catalysts, Low-Code Technology readiness, and Value realisation. Overall, our findings emphasise that the assimilation of low-code technologies is a multifaceted process shaped by the interaction of various organisational resources.

6.2 Contributions

6.2.1 Theoretical contribution

Our findings were analysed through the resource-based theory lens, providing additional insights into the current theoretical understanding of IT/IS assimilation. Our research outcome confirms existing research indicating that sustainable competitive advantage in the IT/IS domain mainly stems from internal organisational factors, more specifically, how IT/IS are structured and managed. Drawing from resource-based theory, our research outcome suggests that, while adopting low-code technology may generate value, sustainable competitive advantage depends on its strategic use within the organisation. By leveraging internal resources to establish specialised teams, guide technology utilisation, identify needs, and promote a culture of innovation, organisations can exploit the potential of low-code technologies to reap more benefits. This highlights the significance of leveraging the possibilities that come with low-code technologies and integrating them within broader organisational strategies to optimise their impact.

Our theoretical contributions extend beyond previous literature by offering a detailed analysis of low-code technology assimilation through the framework of resource-based theory. This approach, to our knowledge, has not been previously explored in relation to low-code technologies. By synthesising empirical findings with theoretical perspectives, we contribute to a deeper understanding of how organisations can strategically leverage internal resources to drive successful low-code technology assimilation and achieve sustainable competitive advantage.

6.2.2 Practical contribution

This bachelor’s thesis offers insights into the factors influencing the low-code assimilation process within organisations, which can be of great value for organisations aiming to gain more benefits from its adoption. We identify two distinct uses, facilitation of software development and promotion of digital transformation, and highlight that organisations should align their use with their specific strategic goals and allocate resources accordingly. Creating an IT strategy tailored to low-code technology is essential, alongside forming specialised teams and empowering solution catalysts. Low-code technology readiness and value realisation remain crucial throughout, requiring deep understanding and proactive resource allocation based on the value created.

Our recommendations include prioritising understanding of the technology, adjusting IT strategy, communicating objectives, forming specialised teams, empowering solution catalysts, and ensuring value realisation across the organisation. By implementing these, we believe organisations can have a higher chance of assimilating low-code technologies, driving innovation, efficiency, and gaining competitive advantage.

6.3 Future Research

While this bachelor’s thesis explored post-adoption use and assimilation of low-code technologies within organisations, there are several opportunities for future research that can provide further understanding of low-code technology assimilation.

Firstly, a study specifically focusing on organisations using low-code technologies in both ways identified in this research could further provide insights into the synergistic effects or potential trade-offs between these dual uses. Secondly, our data hinted at differing perceptions of what constitutes "better" use of low-code technologies among participants. Further research exploring the distinct uses and their effects on organisational performance, such as innovation and competitive advantage, could provide insights into the most effective approaches to low-code technology assimilation. Lastly, governance emerged as a recurring topic among all participants. Exploring various governance frameworks in relation to low-code technology assimilation could provide valuable guidelines and best practices.

Addressing these research areas could be beneficial for further understanding low-code technology assimilation and for supporting organisations in leveraging low-code technologies for driving innovation and gaining competitive advantage.

References

LUSEMs Harvard referencing guide (3rd edition) was used as the referencing style.

Alamin, M.A.A., Uddin, G., Malakar, S., Afroz, S., Haider, T. and Iqbal, A. (2023). Developer discussion topics on the adoption and barriers of low code software development platforms, *Empirical Software Engineering*, 28(1), doi:10.1007/s10664-022-10244-0

Aveiro, D., Freitas, V., Cunha, E., Quintal, F. and Almeida, Y. (2023). Traditional vs. low-code development: comparing needed effort and system complexity in the NexusBRaNT experiment, 2023 IEEE 25th Conference on Business Informatics (CBI), pp. 1–10. doi:10.1109/CBI58679.2023.10187470 [Accessed 1 April 2024]

Barney, J. (1991). Firm Resources and Sustained Competitive Advantage. *Journal of Management*, 17(1), p. 99, doi:10.1177/014920639101700108

Barney, J.B. and Clark, D.N. (2007). *Resource-based theory: creating and sustaining competitive advantage*. Oxford University Press.

Bhattacharyya, S.S. and Kumar, S. (2023). Study of deployment of “low code no code” applications toward improving digitization of supply chain management, *Journal of Science and Technology Policy Management*, 14(2), pp. 271–287, doi:10.1108/JSTPM-06-2021-0084

Brunninghaus, M. and Rocker, C. (2022). Low-Code Development in Worker Assistance Systems: Improving Flexibility and Adaptability, 2022 IEEE 20th International Conference on Industrial Informatics (INDIN), pp. 366–373, doi:10.1109/INDIN51773.2022.9976178 [Accessed 1 April 2024]

Bucaioni, A., Cicchetti, A. and Ciccozzi, F. (2022). Modelling in low-code development: a multi-vocal systematic review, *Software & Systems Modeling*, pp. 1959–1981, doi:10.1007/s10270-021-00964-0

Braun, V. and Clarke, V. (2006). Using thematic analysis in psychology, *Qualitative Research in Psychology*, 3(2), pp. 77–101, doi:10.1191/1478088706qp063oa

Caldeira, M.M. and Ward, J.M. (2003). Using resource-based theory to interpret the successful adoption and use of information systems and technology in manufacturing small and medium-sized enterprises, *European Journal of Information Systems*, 12(2), pp. 127–141, doi:10.1057/palgrave.ejis.3000454

Chuanjian, C., Shuze, G. and Hua, W. (2023). Research on Software Development Based on Low-Code Technology, 2023 2nd International Conference on Artificial Intelligence and Autonomous Robot Systems (AIARS), pp. 210–213. doi:10.1109/AIARS59518.2023.00049 [Accessed 1 April 2024]

Ciucan-Rusu, L., Timus, M., Stefan, D., and Calin-Adrian, C. (2023). Low-code solutions integration for digital process automation - application in management, 2023 22nd RoEduNet Conference: Networking in Education and Research (RoEduNet), pp. 1–5. doi:10.1109/RoEduNet60162.2023.10274938 [Accessed 1 April 2024]

de Mattos, C.A. and Laurindo, F.J.B. (2017). Information technology adoption and assimilation: Focus on the suppliers portal, *Computers in Industry*, 85, pp. 48–57, doi:10.1016/j.compind.2016.12.009

Denning, S. (2023). Recognizing and outmaneuvering the resistance to digital transformation, *Strategy & Leadership*, 51(2), pp. 10–16. doi:10.1108/SL-01-2023-0002.

El Kamouchi, H., Kissi, M. and El Beggar, O. (2023). Low-code/No-code Development: A systematic literature review, 2023 14th International Conference on Intelligent Systems: Theories and Applications (SITA), pp. 1–8. doi:10.1109/SITA60746.2023.10373712 [Accessed 1 April 2024]

Gao, P., Gong, Y., Zhang, J., Mao, H. and Liu, S. (2019). The joint effects of IT resources and CEO support in IT assimilation: Evidence from large-sized enterprises, *Industrial Management & Data Systems*, 119(6), pp. 1321–1338. doi:10.1108/IMDS-08-2018-0345.

Klein, H.K., Meyers, M.D. (1999). A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems, *MIS Quarterly*, pp 67–93, doi: 10.2307/249410

Käss, S., Strahringer, S. and Westner, M. (2022). Drivers and Inhibitors of Low Code Development Platform Adoption, IEEE 24th Conference on Business Informatics (CBI), pp. 196–205, doi:10.1109/CBI54897.2022.00028 [Accessed 1 April 2024]

Käss, S., Westner, M. and Strahringer, S. (2023a). A Multiple Mini Case Study on the Adoption of Low Code Development Platforms in Work Systems, *IEEE Access, Access, IEEE*, vol. 11, pp. 118762–118786, doi:10.1109/ACCESS.2023.3325092

Käss, S., Westner, M. and Strahringer, S. (2023b). Practitioners' Perceptions on the Adoption of Low Code Development Platforms, *IEEE Access, Access, IEEE*, vol. 11, pp, 29009–29034 doi:10.1109/ACCESS.2023.3258539

Martínez-Caro, E., Cepeda-Carrión, G., Cegarra-Navarro, J.G. and Garcia-Perez, A. (2020). The effect of information technology assimilation on firm performance in B2B scenarios, *Industrial Management & Data Systems*, 120(12), pp. 2269–2296. doi:10.1108/IMDS-10-2019-0554.

Mata, F.J., Fuerst, W.L. and Barney, J.B. (1995). Information Technology and Sustained Competitive Advantage: A Resource-Based Analysis, *MIS Quarterly*, 19(4), pp. 487–505, doi:10.2307/249630

Oates, B.J. (2006). *Researching Information Systems and Computing*, London: SAGE

Pacheco, J., Garbatov, S. and Goulao, M. (2021). Improving collaboration efficiency between ux/ui designers and developers in a low-code platform, ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), pp. 138–147, https://doi.org/10.1109/MODELS-C53483.2021.00025 [Accessed 1 April 2024]

Patton, M.Q. (2015). *Qualitative research & evaluation methods: integrating theory and practice*. 4. ed. SAGE Publications, Inc. [Accessed 1 April 2024]

Peppard, J. and Ward, J. (2016). *The strategic management of information systems. Building a digital strategy*. 4. ed. Wiley.

Pinho, D., Aguiar, A. and Amaral, V. (2023). What about the usability in low-code platforms? A systematic literature review, *Journal of Computer Languages*, 74. doi:10.1016/j.cola.2022.101185.

Razak, S.F.A., Ernn, Y.P., Yussoff, F.I., Bukar, U.A. and Yogarayan, S. (2024). Enhancing Business Efficiency through Low-Code/No-Code Technology Adoption: Insights from an Extended UTAUT Model, *Journal of Human, Earth, and Future*, 5(1), pp. 85–99, doi:10.28991/HEF-2024-05-01-07

Redchuk, A., Mateo, F.W., Pascal, G., and Tornillo, J.E. (2023). Adoption Case of IIoT and Machine Learning to Improve Energy Consumption at a Process Manufacturing Firm, under Industry 5.0 Model, *Big Data and Cognitive Computing*, 7(1), doi:10.3390/bdcc7010042

Richardson, C. and Rymer J. R. (2014). New Development Platforms Emerge for Customer-Facing Applications, available at: https://www.forrester.com

Roberts, N., Jeyaraj, A. and Pullin, J.E. (2023). Assessing the Connections among Top Management Support, IT Assimilation, and the Business Value of IT: A Meta-Analysis, *Journal of the Association for Information Systems*, 24(1), pp. 107–135. doi:10.17705/1jais.00772.

Rokis, K. and Kirikova, M. (2023). Exploring Low-Code Development: A Comprehensive Literature Review, *Complex Systems Informatics and Modeling Quarterly*, 2023(36), pp. 68–86. doi:10.7250/csimq.2023-36.04.

Sahay, A., Indamutsa, A., Di Ruscio, D. and Pierantonio, A. (2020). Supporting the understanding and comparison of low-code development platforms, 2020 46th Euromicro Conference on SEAA, pp. 171–178, 10.1109/SEAA51224.2020.00036 [Accessed 1 April 2024]

Schmidt, J. (2023). Mitigating risk of failure in information technology projects: Causes and mechanisms, *Project Leadership and Society*, 4, doi:10.1016/j.plas.2023.100097

Wang, N., Liang, H., Ge, S., Xue, Y. and Ma, J. (2019). Enablers and inhibitors of cloud computing assimilation: an empirical study, *Internet Research*, 29(6), pp. 1344–1369. doi:10.1108/INTR-03-2018-0126.

Appendices

Appendix A: Interview Guide

LUND UNIVERSITY

SCHOOL OF ECONOMICS AND MANAGEMENT

Interview Guide

Background/control questions

1. Can you tell us a bit about the organisation you work with in relation to size and the main organisational business?
2. What is your role in the organisation?
3. How would you describe your familiarity or knowledge regarding low-code technology is? If you could define it in your own words, what would you say?
4. Which low-code technology (LCT) is adopted in your organisation and since when approximately?
    a. Do you know what or who triggered this decision? (A project, a single worker, top management, etc.)
    b. If a consultant: for the purpose of this research, could we focus on a specific organisation that you have been working with for more than four months.

Main questions

1. Could you describe a bit more your role in relation to LCT in the organisation?
    a. Could you share some examples with us so that we get a better understanding?
2. Can you describe how the organisation is using LCT at the moment?
    b. How is the IT department using these tools?
    c. Are you aware of other departments using LCT in the organisation?
3. Are you aware of any strategies that are influencing how much or in which way the organisation currently uses LCT?
    d. Could you provide some examples?
    e. What would you say affects how LCT are being used and to what extent?
    f. How do you feel about that?
4. If there is a recent project or initiative where the organisation utilised low-code technologies, can you describe it?
    g. What motivated the organisation to explore or adopt low-code solutions for this project?
5. How did the process after adopting low-code technologies look like within the organisation?
    h. How was this decided, based on what?
6. Have you noticed any specific strategies or practices that aimed at further leveraging LCT?
    i. Are there any such strategies at the moment?
7. From the initial adoption until now, is there anything you would have changed?
8. How did the adoption of this LCT in the organisation affect the organisation (in terms of business process, new tools, etc.)
9. Did you notice any challenges after the adoption of LCT in the organisation?
10. From your experience, what has been important in supporting LCT use within the organisation?
11. What would you say is necessary for a successful implementation of LCT?
    j. Have you noticed if there are any specific factors that contribute to this?
12. What are your key learnings you would recommend to other organisations trying to leverage LCT?

- Any final comments or insights you would like to share with us regarding what we have discussed today?

Appendix B: Informed Consent form

SCHOOL OF ECONOMICS AND MANAGEMENT

I, the undersigned, confirm that (please tick the appropriate box):

1.	I understand the information about the project.	□
2.	I have been given the opportunity to ask questions about the project and my participation.	□
3.	I voluntarily agree to participate in the project titled *How Organizations Assimilate Low-Code Technologies: A Qualitative Study on the Influential Factors during the Post-Adoption Phase* (the possibility of making slight changes to the title are declared).	□
4.	I understand I can withdraw at any time without giving reasons and that I will not be penalised for withdrawing nor will I be questioned on why I have withdrawn.	□
5.	I understand that my identity and contribution will not be revealed nor shared with others outside this interview and those involved in this research, such as our supervisor.	□
6.	The use of the data in research, publications, sharing and archiving has been explained to me.	□
7.	I agree that this interview is recorded for the purpose of data collection.	□
8.	I understand that if I have questions about the research or about my role, I can contact Maria Dalgren (ma1141da-s@student.lu.se) or Natali Plazonic (na1012pl-s@student.lu.se) by e-mail.	□
9.	I, along with the researchers, agree to sign and date this informed consent form.	□

Participant:

________________________ ___________________________ ________________

Name of Participant Signature Date

Researcher:

March-April 2024

________________________ ___________________________ ________________

Name of Researcher Signature Date

Researcher:

March-April 2024

________________________ ___________________________ ________________

Name of Researcher Signature Date

AI Acknowledgment

Throughout this bachelor’s thesis, ChatGPT (AI-powered language model developed by OpenAI) has been used as a tool for spell-checking, finding synonyms, as well as reformulating our own text. We recognise and acknowledge AI’s contribution to maintaining a formal language throughout the text and improving the overall text quality.