

# Understanding Low-Code Evolution, Adoption and Ecosystem for Software Development\*

Syed Asad Ali Naqvi<sup>1,2,\*</sup>, Paul Drews<sup>1</sup>

<sup>1</sup>Leuphana University of Lüneburg, Lower Saxony, 21335, Lüneburg, Germany

<sup>2</sup>Accenture Ecosystem, Balanstrasse 73, bldg 17, 81541 Munich, Germany

## Abstract

Faced with the high demands of digital transformation, organizations have turned to low-code as a solution, utilizing visual programming to meet these challenges. Given that low-code is an emerging research phenomenon, there is little understanding of its origins, adoption motives and approaches, and its ecosystem. In my PhD thesis, I use qualitative and visualization methods to address these research gaps. My work has five contributions. First, I developed an intellectual map of the low-code literature and identified related research streams. Second, I created a model underscoring aspects that support or hinder low-code adoption. Third, I provide use cases demonstrating low-code adoption in established organizations and its use in the public sector during crises. Fourth, I identify the technology partnerships forming the low-code ecosystem. Fifth, I investigate the impact of generative AI on low-code adoption and whether it presents a threat to the low-code ecosystem. In conclusion, I outline potential directions for future research.

## Keywords

Low-code, Low-code platforms, Low-code ecosystem, Technology adoption, Software development

## 1. Problem Definition

Digital transformation (DT) is vital for organizations to create value, improve competitiveness, and achieve success [1]. DT of organizations uses various approaches to develop software applications tailored to specific contexts. Despite the development of software development strategies, such as outsourcing and customizing commercial software, software development projects continue to experience high failure rates [2]. Low-code is enabled by low-code platforms (LCPs) that streamline the development, testing, and implementation of software applications by significantly reducing the need for manual coding through the use of pre-built software elements and visual design tools [3]. Low-code refers to an emerging trend in software development that helps organizations address DT challenges and meet their growing digitalization needs [4]. LCPs are designed to be user-friendly and enable use without extensive training. Hence, LCPs are perceived as a means of democratizing software development, broadening access to non-IT professionals, often called citizen developers [5, 6].

### 1.1. Understanding low-code and its intellectual structure

Over the past decade, the body of literature on low-code is expanding rapidly. Despite the widespread popularity of LCPs, the term low-code is still ambiguous [7]. Primarily, this arises from the fact that low-code research originates from a range of interdisciplinary fields and several evolving trends within software engineering [8, 9]. A shared understanding of what constitutes low-code, can help researchers advance theoretical understanding, aid practitioners in communicating its value and realizing its potential. Given low-code's various origins, we argue that we need to take a broad perspective to understand low-code. While there is a significant increase in research on low-code, we still lack a

---

*The 15th International Conference on Software Business (ICSOB 2024), November 18–20, 2024, Utrecht, the Netherlands*

\*Corresponding author.

†These authors contributed equally.

✉ ali.naqvi@accenture.com (S. A. A. Naqvi); paul.drews@leuphana.de (P. Drews)

🌐 <https://www.linkedin.com/in/asadalinaqvi1/> (S. A. A. Naqvi); <https://www.linkedin.com/in/pdrews/> (P. Drews)

🆔 0009-0007-9178-5139 (S. A. A. Naqvi); 0000-0002-9845-5024 (P. Drews)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

intellectual structure for this body of work. An intellectual structure maps the existing literature on a given phenomenon and aggregates it to research streams. Such a structure guides research and influences future scholarship, making it vital for understanding a field [10]. I aim to address the ambiguity of low-code and produce useful insights into the emergence and evolution of low-code by evaluating links between existing literature. Using the intellectual structure, I aim to highlight different disciplines' perspectives on low-code, creating a low-code definition that is derived from its intellectual structure, to support the understanding of low-code.

## **1.2. Investigating the socio-technical aspects that influence low-code adoption**

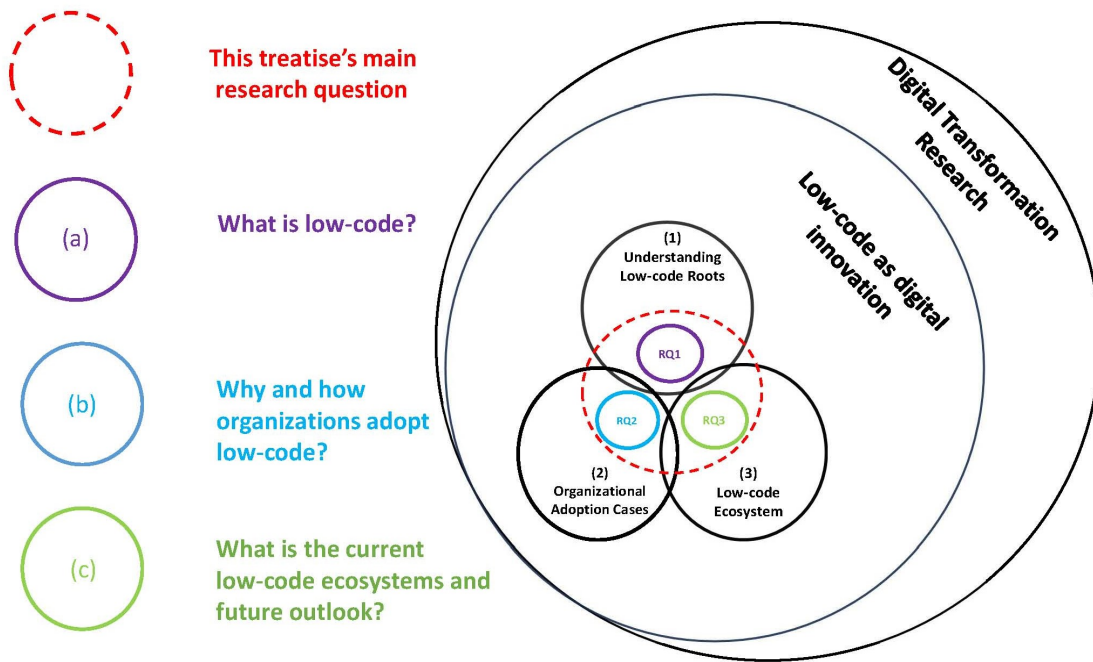
Adoption refers to the diffusion and use of technology to enhance productivity or achieve economic benefits [11]. The adoption of low-code differs significantly from the traditional software development methods found in the literature. In contrast to traditional packaged software solutions such as CRM, or methodologies such as SCRUM, low-code adoption involves embracing a technology along with a novel approach to developing software simultaneously. Despite the increasing popularity of the low-code approach among practitioners, we lack empirical explanations of why organizations opt for low-code as a software development strategy and which aspects hinder its adoption [7, 9]. In addition, most studies have investigated low-code only in technological contexts and the existing literature does not provide an understanding of how the interplay of technological, organizational and environmental factors influences organizational decisions to embrace low-code as a software development strategy [12, 13]. Since low-code is an emerging research phenomenon, low-code adoption has not yet been investigated in various contexts, such as incumbent organizations and the public sector. Since DT in such organizations is often slowed down due to the outdated existing IT infrastructure, the replacement or change of processes and digital applications within such organizations is complicated and involves organizational, legal, ethical, and social concerns. Whether low-code can support DT in such organizations with large existing IT landscape and how such organizations can systematically adopt low-code remain unexplored. A comprehensive understanding of context-specific low-code adoption could help explain the role of low-code in organizations' DT as well as aspects that hinder it.

## **1.3. Low-code Platforms ecosystem**

To simplify software development and reduce the need to switch between different systems, LCPs integrate tools such as the design of the graphical user interface (GUI), database management systems, and application programming interfaces (API) into one platform [7]. Research shows that due to the wide-ranging functionalities offered by LCPs necessary for simplifying software development, even leading tech companies cannot deliver the entire LCP on their own [14]. Consequently, to create value, LCP vendors integrate and leverage technology partnerships with other vendors. Despite the high relevance of LCPs for software development, the existing literature does not provide a comprehensive view of the LCP ecosystem. Evaluating and accessing the LCP ecosystem can help to understand similarities and differences between LCPs and inform strategic adoption.

## **1.4. Motivation**

My Phd thesis has three core objectives that motivate my research. The first objective is to identify the origins of low-code and research streams that are discussed within the existing low-code literature. Using these research streams, my aim is to create a definition of low-code that is built on the existing literature and improve the understanding of low-code. The second objective is to understand the adoption of low-code in incumbent organizations for enterprise software development. Although the literature provides insight into how startups, small and medium enterprises can use low-code [15, 16], the adoption of low-code in incumbent organizations has not been studied. Examining low-code adoption in incumbent organizations is essential due to their existing IT infrastructure and unique adoption challenges compared to digital native start-ups. The aim is to gain insight into the aspects that support or hinder the adoption of low-code, and how incumbent organizations can leverage low-code



**Figure 1:** Illustration of the interconnection between the research questions, and the three research objectives within the study of low-code as digital innovation.

to support their existing IT infrastructure. To capture the differences in organizational nature, I conduct multiple case studies to evaluate how organization nature (early and late adopters), internal and external events, can influence adoption. The third aim is to identify key technology partnerships and partnering strategies within the LCP ecosystem and understand the future research discourse for low-code in context of generative AI.

I aim to use my research on low-code to support and help shape DT in organizations. Following our vision, I aim to capture the impact of low-code in supporting organizations' DT. The goal extends beyond publications; it focuses on producing research that aids organizations and stakeholders in making sense of low-code and adopt the low-code by developing capabilities that minimize risks and immediate positive impact on their everyday activities.

Therefore, I investigate the following research questions (RQs):

RQ1) What is low-code?

RQ2) Why and how organizations adopt low-code?

RQ3) What is the current low-code ecosystems and future research directions for low-code?

The scope of my thesis is illustrated in Figure 1. By investigating these research questions, my aim is to provide a state-of-the-art analysis of the evolution, adoption, and ecosystem of low-code. Using both qualitative and quantitative methods, I offer insight an understanding of low-code to various stakeholders such as chief information officers, chief technology officer, product managers, and developers on how organizations can understand and leverage low-code for software development. In addition, I conclude my thesis with future research directions for low-code.

## 2. Knowledge Gap

My PhD thesis seeks to bridge five significant knowledge gaps identified within the current body of literature. First, at the start of my thesis in 2022, I discovered a mere three scholarly works that synthesized low-code literature [17, 18, 19]. Concurrently, I noted a significant surge in the number of publications regarding low-code. Thus, I recognized the necessity of organizing existing research streams within the low-code literature. After synthesizing the available literature on low-code, it

became apparent that there is an absence of a formalized definition for low-code. Second, although certain studies highlight parallels between low-code as an emerging phenomenon and existing Bodies of Knowledge (BoKs), such as Model-Driven Engineering (MDE) as noted by [8], there is a gap in the literature in terms of exploring the relationship between low-code and existing BoKs. Additionally, the literature does not delineate which insights from existing BoKs are applicable to low-code, nor does it clarify which components are distinctive to low-code itself [7].

Third, while the potential of LCPs to streamline software development is noted in the literature, a comprehensive investigation of low-code adoption processes remains limited [16]. The adoption of low-code is distinct from traditional technology adoption as it integrates methodologies and technological solutions for software creation. Nevertheless, the majority of research concerning low-code primarily addresses the technological capabilities of specific platform use cases [7, 12]. There is a noticeable lack of detailed exploration of how these LCPs are adopted within incumbent organizations. Case studies provide valuable information on how new technologies and methodologies are adopted. Research has emphasized the need for a comprehensive examination of socio-technical aspects that affect the acceptance and challenges in the adoption of low-code [13, 16]. In response to this call, I investigated how incumbent organizations can cultivate the necessary capabilities to integrate low-code solutions into their existing IT infrastructure and the aspects that support or hinder the low-code adoption. Fourth, while low-code simplifies and accelerates application development, the literature lacks guidance on its role in supporting DT in crisis scenarios. To explore how LCPs could support DT in times of crisis, I investigated three distinct public sector organizations, each using a different LCP. From this study, I developed a comprehensive model that illustrates how low-code can support DT during crisis responses.

Fifth, the current literature lacks insight into the low-code ecosystem and LCPs' technology partnership strategies. Understanding technology partnership strategies in low-code ecosystem can support organizations in selecting LCP based on organizational DT goals. Therefore, I analyze the impact of technology partnerships within the existing low-code ecosystem. I develop a framework for platform-agnostic partnering strategies employed by LCP providers that illustrates how such partnerships shape the low-code ecosystem. Sixth, while the impact of LCPs in streamlining software development is widely acknowledged, research on how these platforms could further enhance software development processes through integration with emerging technologies such as generative AI is scarce [20]. In order to address this gap, I investigate if LCPs are complimented with generative AI or threatened by it. The concluding section of my thesis outlines potential avenues for future research in the field of low-code.

### 3. Research Method

To address RQ1, I conducted a literature review of empirical papers on low-code. I choose nine databases including ACM Digital Library, AISel, IEEE Xplore Digital Library, JSTOR, Science Direct, Emerald, SpringerLink, and Wiley Online Library, that offer an extensive range of peer reviewed publications from various disciplines and selected seminal publications that explicitly deal with low-code development. I used *Connected Papers*<sup>1</sup>, a web tool for finding relevant academic papers, to identify key works related to a focal paper. Connected Papers integrates with the Semantic Scholar corpus<sup>2</sup>, accessing millions of articles across scientific fields. Consequently, I conducted a bibliographic analysis using 725 articles. Among these, I selected 105 articles that featured the term "low-code" in either the title or the abstract. This study streamlines the understanding of low-code by identifying six research streams. Next, based on the identified research streams, I am currently analyzing the existing software engineering disciplines as existing bodies of knowledge (BoKs) to identify the lessons that can be applied to low-code as emerging research phenomenon.

To address the gaps related to the low-code adoption (RQ2), I conducted three studies. Data collection for these studies involves conducting eighteen semi-structured expert interviews. Every interviewee

---

<sup>1</sup>[www.connectedpapers.com](http://www.connectedpapers.com)

<sup>2</sup>[www.semanticscholar.org](http://www.semanticscholar.org)

has experience in enterprise application development projects, having worked with both low-code and traditional software development methods. I anticipated that the interviewees might only emphasize the positive aspects of low-code adoption. To counteract this bias, I collected user reviews from Trustradius.<sup>3</sup> on LCPs. Trustradius is a widely used site for business technology reviews. I collected 654 user reviews for the three leading LCPs, Appian, Pegasystems, and Outsystems, recognized as key platforms for enterprise app development and rated as leaders and visionaries by Gartner and Forrester reports [21, 14]. For case studies examining the application of low code in supporting DT during crisis responses, I collect data through official public sector documents, news articles, LCP vendor reports, and market analyzes from Forrester, IDC, Gartner, and Accenture. For analysis, I employ qualitative methods and theoretical frameworks such as the DT Framework [1] and Dynamic Capability Theory [22].

To address RQ3 by building the LCP ecosystem map, I analyze reports from analysts such as Gartner and Forrester [21, 14], looking for LCPs that are frequently recognized as leaders, strong performers and challengers. This approach offered a diverse sample and resulted in a total of 16 LCPs. Then I used Partnerbase<sup>4</sup> and identified technology partnerships for these LCPs, resulting in a total of 1,086 technology partnerships. For analysis, I used a data-driven approach, as described by [23] to identify and curate a data set of LCP technology partnerships and build a visualization illustrating the LCP ecosystem.

## 4. Timeline

At the current stage of the dissertation project, five conference papers have been accepted. The dissertation includes seven publications, including a minimum of one journal paper and one conference paper currently in preparation. Table 1 outlines the scheduled activities and the timeline for the duration of the PhD thesis.

**Table 1**  
Timeline planned for the next activities

S.No.	Activity	Duration
1	Write outline of thesis wrapper and plan next steps for ICSOB Phd Retreat	October 2024
2	Conduct survey for generative AI impact on low-code	November 2024
3	Submit low-code relation to existing Boks paper to a journal	December 2024
4	Present two accepted papers at HICSS 2025 as first author	January 2025
5	Feedback and adjustment for the journal paper major revision 1	February-March 2025
6	Write generative AI impact on low-code ICIS/HICSS	April-May 2025
7	Write thesis wrapper and submit to primary supervisor	June-July 2025
8	Feedback and adjustment for the journal paper revision 2	August 2025
9	Adjust thesis wrapper as per feedback from primary supervisor	September 2025
10	Discussion and feedback from demo presentation of thesis defense	October 2025
11	Submit thesis to all three supervisors	November 2025
12	Present paper at ICIS/HICSS	December 2025
13	Feedback and adjustment for the journal paper revision 3	January 2026
14	Present and Defense PhD Thesis	February 2026

## 5. Preliminary Results

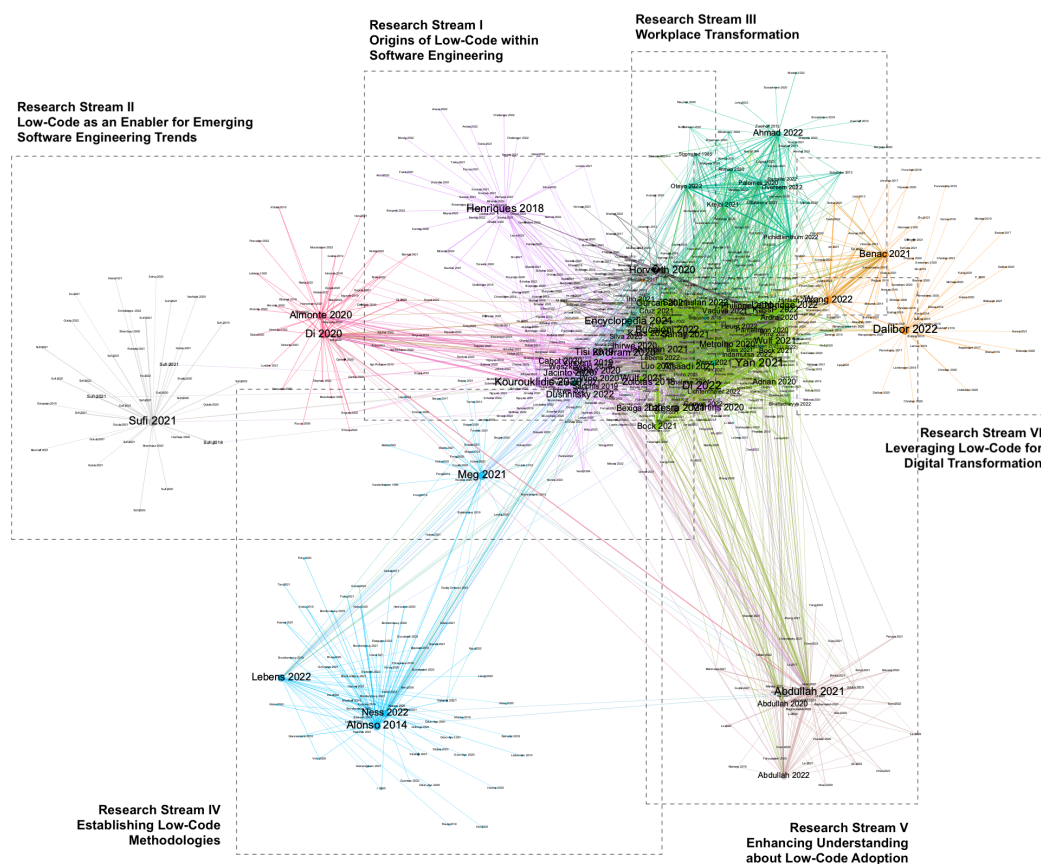
### 5.1. RQ1) What is low-code?

This question is addressed in two publications. First, I review the existing literature on low-code in the paper "The Low-Code Phenomenon: Mapping the Intellectual Structure of Research", which

<sup>3</sup>[www.trustradius.com](http://www.trustradius.com)

<sup>4</sup>[www.Partnerbase.com](http://www.Partnerbase.com)

is published at the conference HICSS 2024 [3] and is nominated for the Best Paper Award for the Software Technology and Software Development Track . I investigate: **RQ 1.1) What research streams characterize the low-code literature?** Through visualization, I identified 105 publications on low-code, revealing six distinct research streams in the existing literature. Furthermore, this analysis provided an intellectual map that detail the research landscape of the field, as illustrated in Figure 2. The research streams include the origins of low-code within software engineering, low-code as an enabler for emerging software engineering trends, workplace transformation, establishing low-code methodologies, enhancing understanding about low-code adoption, and leveraging low-code for DT. By thorough review of the low-code literature and pinpointing different research streams, I formulate a definition of the low-code phenomenon. *The low-code phenomenon originates from software engineering and integrates software engineering tools to offer unified capabilities for simplifying and advancing software development, and workplace transformation, thus, contributing to the digital transformation. At the same time, we need to establish and enhance low-code methodologies to promote low-code adoption.*



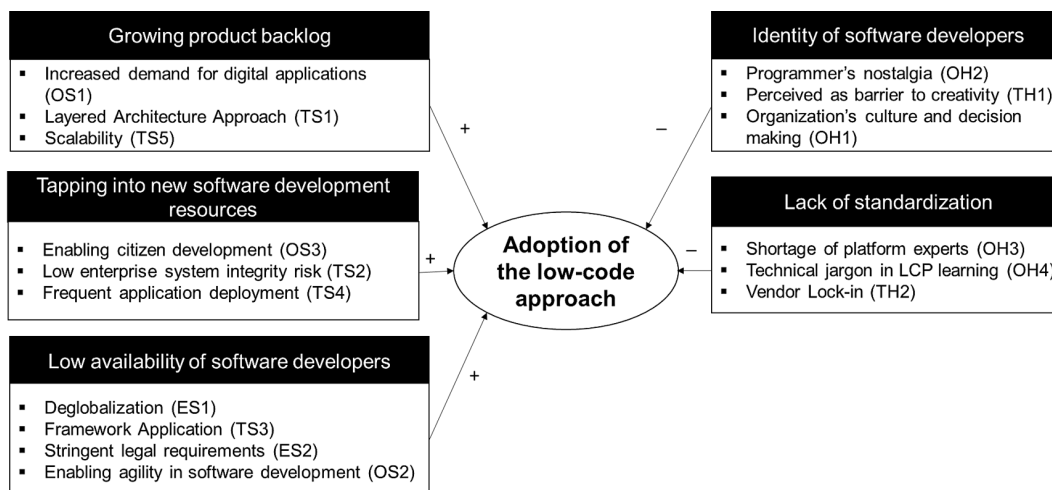
**Figure 2:** Intellectual Structure of Low-Code Research 2014-2022, originally published at [3]

Analyzing the research streams, I note that low-code is an emerging research phenomenon, rooted in existing body of knowledge (BoKs). In particular, in the first research stream, the existing literature identifies similarities between low-code and software engineering disciplines. By recognizing trends in emerging research phenomena and aligning them with existing BoKs, my aim is to propose a methodology that aids to formulate theories and delineate a comprehensive research agenda. In the second publication, I develop a methodology paper that provides a step-by-step approach to link emerging research phenomena with existing BoKs. This is an ongoing research initiative. I aim for publishing in a leading Information Systems journal. I investigate: **RQ 1.2) How is the emerging phenomenon of low-code related to the existing body of knowledge of end-user development and**

## rapid application development?

### 5.2. RQ2) Why and how organizations adopt low-code?

To answer this RQ, I conduct three studies. First, I analyze the low-code adoption from a holistic level in my publication "Understanding the socio-technical aspects of low-code adoption for software development" that I presented at ECIS 2023 [24]. I investigate: **RQ 2.1) Which aspects influence organizations to adopt low-code for software development?** The results of the study is a model that identifies 10 supporting and 6 hindering aspects of low-code adoption (see Figure 3). This study revealed interesting insights that some software developers perceive low-code as a barrier to their creativity and are concerned that citizen development would challenge their identity [24].



**Figure 3:** Aspects supporting and hindering the adoption of the low-code approach, originally published at [24]

Given that numerous organizations are only in the early phases of low-code implementation, research in this domain is still nascent [3]. Qualitative methods such as interviews and case studies are recommended for understudied and revealing cases [25]. The case study enables a deep exploration of a phenomenon, often requiring data from various sources [26]. The intersection of DT challenges in the public sector and the COVID-19 pandemic provides a compelling and unparalleled opportunity to explore the adoption of low-code within this sector. Hence, in my second publication, titled, "Empowering the Public Sector: Unveiling the Transformational Potential of Low-Code in Crisis Responses" [27], which is published at AMCIS 2024, I investigate the transformative role of LCPs in aiding public sectors to tackle crisis response challenges. I present three case studies of LCPs deployed in the public sector to provide digital applications to citizens during the COVID-19 pandemic to answer RQ 2.2) *How did public sector organizations leverage low-code capabilities during the pandemic to support digital transformation?* These applications, created with different LCPs, have diverse functionalities, illustrating low-code's significant role in advancing DT. Consequently, by doing cross-case analysis, I identify the capabilities of LCPs and six insights to enhance understanding of low-code adoption within the public sector. Moreover, using the DT building blocks proposed by [1], I demonstrate the substantial impact of low-code and its ability to aid DT processes in organizations (Figure 4).

In my third publication on low-code adoption, "Overcoming Multi-Legacy Application Challenges through Building Dynamic Capabilities for Low-Code Adoption", which is accepted at HICCS 2025 [28], I tackle the challenge how incumbent organizations can develop capabilities to leverage low-code with their existing IT Infrastructure by investigating RQ 2.3) **How do organizations develop dynamic capabilities to leverage low-code to overcome siloed legacy application challenges?**

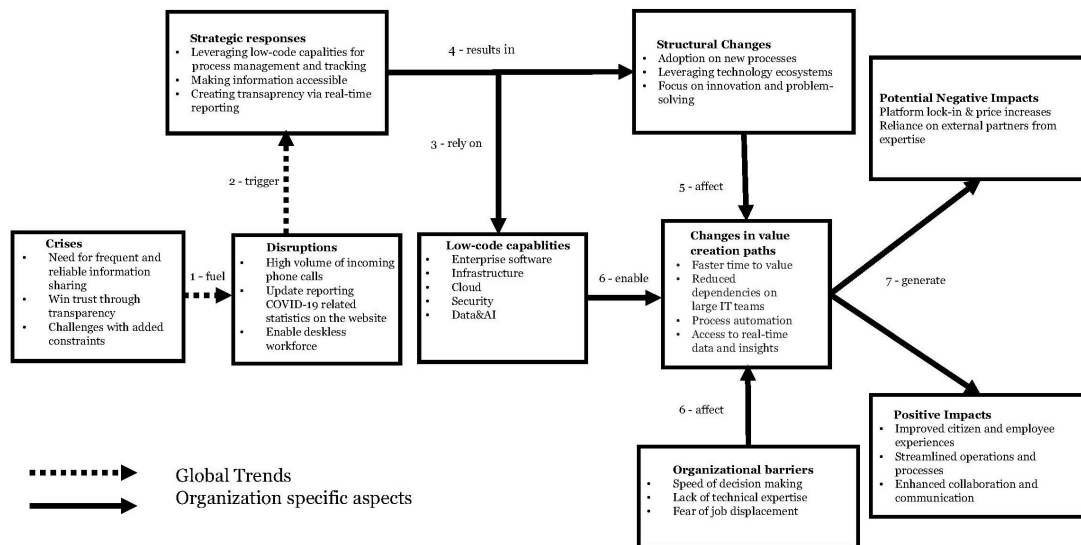


Figure 4: Low-code as a transformative process, originally published at [27]

### 5.3. RQ3) Low-code Ecosystem

I conduct two studies to investigate the low-code ecosystem. The first study is titled "Understanding Partnering Strategies in the Low-Code Platform Ecosystem", which is accepted at HICCS 2025 [29]. I investigated the **RQ 3.1) How do technology partnerships shape the low-code platform (LCPs) ecosystem?** I curated the technology partnerships for the 16 LCPs and categorize the partnerships into nine categories including cloud, collaboration, compliance and personnel, data and AI, enterprise software, infrastructure, marketing and automation, security and testing. I demonstrated that LCPs utilize diverse technology partnerships, characterized by varying partner network sizes and blends of core and complementary capabilities. In my second study, I engage with existing research that calls for investigating the impact of generative AI on low-code [20]. I explore the challenges in low-code adoption and whether technologies such as generative AI pose a threat to low-code and its ecosystem.

## 6. Conclusion

Software development (SW) is the cornerstone of digital transformation (DT). The ways to optimize SW has remained an interesting research avenue such as packaged software. My thesis offers practical insights on developing the understanding of low-code by synthesizing the existing literature, identifying the key aspects that support or hinder its adoption, and outlining the low-code ecosystem and future developments for low-code. By doing so, I unpack how organizations can leverage low-code to transform software development practices and solve organizational and societal challenges.

## References

- [1] G. Vial, Understanding digital transformation: A review and a research agenda, *The journal of strategic information systems* 28 (2019) 118–144.
- [2] J. K. A. . H. V. Engelbrecht, J., The influence of business managers' it competence on it project success, *International Journal of Project Management* 35 (2017) 994–1005.
- [3] S. A. A. Naqvi, M. Zimmer, P. Drews, K. Lemmer, R. Basole, The low-code phenomenon: Mapping the intellectual structure of research, *Hawaii International Conference on System Sciences* (2024).
- [4] M. M. Li, C. Peters, M. Poser, K. Eilers, E. Elshan, Ict-enabled job crafting: How business unit developers use low-code development platforms to craft jobs, *International Conference on Information Systems (ICIS)* (2022).



- [5] M. Lebens, R. J. Finnegan, S. C. Sorsen, J. Shah, Rise of the citizen developer, *Muma Business Review* 5 (2021) 101–111.
- [6] B. Binzer, E. Elshan, D. Fürstenau, T. J. Winkler, Establishing a low-code/no-code-enabled citizen development strategy, *MISQE* (2024).
- [7] A. C. Bock, U. Frank, Low-code platform, *Business & Information Sys Eng* 63 (2021) 733–740.
- [8] D. Di Ruscio, D. Kolovos, J. de Lara, A. Pierantonio, M. Tisi, M. Wimmer, Low-code development and model-driven engineering: Two sides of the same coin?, *Software and Systems Modeling* 21 (2022) 437–446. doi:10.1007/s10270-021-00970-2.
- [9] E. Elshan, D. Siemon, O. Bruhin, D. Kedziora, N. Schmidt, Unveiling challenges and opportunities in low code development platforms: A stackoverflow analysis, *Hawaii International Conference on System Sciences (HICSS)* (2024).
- [10] H. D. White, B. C. Griffith, Author cocitation: A literature measure of intellectual structure, *Journal of the American Society for Information Science* 32 (1981) 163–171.
- [11] E. Karahanna, D. W. Straub, N. L. Chervany, Information technology adoption across time: A cross-sectional comparison of pre-adoption and post-adoption beliefs, *MIS quarterly* (1999).
- [12] Iho, S., Krejci, D., & Missonier, S., Supporting knowledge integration with low-code development platforms, In *ECIS*. (2021).
- [13] D. Hoogsteen, H. Borgman, Empower the workforce, empower the company? citizen development adoption, *Hawaii International Conference on System Sciences (HICSS)*, 2022.
- [14] C. Richardson, J. R. Rymer, Vendor landscape: The fractured, fertile terrain of low-code application platforms, *Forrester, Janeiro* (2016) 12.
- [15] G. Dushnitsky, B. K. Stroube, Low-code entrepreneurship: Shopify and the alternative path to growth, *Journal of Business Venturing Insights* 16 (2021).
- [16] E. Elshan, E. Dickhaut, P. A. Ebel, An investigation of why low code platforms provide answers and new challenges, *Hawaii International Conference on System Sciences* (2023).
- [17] N. Prinz, C. Rentrop, M. & Huber, Low-code development platforms—a literature review, *Americas Conference on Information Systems (AMCIS)* (2021).
- [18] D. Pinho, A. Aguiar, V. Amaral, What about the usability in low-code platforms? A systematic literature review, *Journal of Computer Languages* (2022).
- [19] S. Käss, S. Strahringer, M. Westner, Drivers and inhibitors of low code development platform adoption, *IEEE 24th Conference on Business Informatics (CBI)* (2022).
- [20] O. Bruhin, E. Dickhaut, E. Elshan, M. Li, The rise of generative ai in low code development platforms—an analysis and future directions, *Hawaii International Conference on System Sciences (HICSS)* (2024).
- [21] P. Vincent, K. Iijima, M. Driver, J. Wong, Y. V. Natis, Magic quadrant for enterprise low-code application platforms, *Gartner Report* (2019).
- [22] D. J. Teece, Explicating dynamic capabilities: the nature and microfoundations of (sustainable) enterprise performance, *Strategic management journal* 28 (2007) 1319–1350.
- [23] B. Iyer, R. C. Basole, Visualization to understand ecosystems, *Communications of the ACM* (2016).
- [24] S. A. A. Naqvi, M. P. Zimmer, R. Syed, P. Drews, Understanding the socio-technical aspects of low-code adoption for software development, *European Conference on Information Systems (ECIS)* (2023).
- [25] S. Gregor, The nature of theory in information systems, *JSTOR*, 2006.
- [26] R. K. Yin, Case study research: Design and methods, volume 5, Sage, 2009.
- [27] S. A. A. Naqvi, M. Zimmer, P. Drews, Empowering the public sector: Unveiling the transformational potential of low-code in crisis responses, *Americas Conference on Information Systems (AMCIS)* (2024).
- [28] S. A. A. Naqvi, M. Zimmer, S. Rehan, P. Drews, R. Basole, Overcoming multi legacy application challenges through building dynamic capabilities for low-code adoption, *Hawaii International Conference on System Sciences (HICSS)* (2025).
- [29] S. A. A. Naqvi, M. Zimmer, P. Drews, R. Basole, Understanding partnering strategies in the low-code platform ecosystem, *Hawaii International Conference on System Sciences* (2025).