Ilke Söylemez

Accelerating Low Code Automation Development with Generative Artificial Intelligence

School of Technology and Innovations
Master's thesis in Industrial Management
Master of Science in Economics and Business Administration

Vaasa 2024

UNIVERSITY OF VAASA
School of Technology and Innovations
Author: Ilke Söylemez
Title of the Thesis: Accelerating Low Code Automation Development with Generative Artificial Intelligence
Degree: Master of Science in Economics and Business Administration
Programme: Industrial Management
Supervisor: Ville Tuomi
Supervisor (Case company): Antti Hannonen
Year: 2024 **Page:** 74

ABSTRACT:

This thesis investigates the integration of Generative Artificial Intelligence (Generative AI) into low code automation development platforms, with a particular emphasis on enhancing Robotic Process Automation (RPA). This study centers on three crucial points: the influence of Generative AI on the functionality of low code automation tools, the challenges and limitations of using Generative AI within these platforms, and the strategies to mitigate such challenges. Through a comprehensive literature review and a case study approach involving semi-structured interviews with RPA experts and citizen developers, this research addresses critical questions concerning the efficiency, challenges, mitigations, and future prospects of incorporating Generative AI into low code automation development efforts. The research findings highlight that while Generative AI offers considerable advantages in terms of functionality enhancement and accessibility, it also poses significant challenges such as issues with accuracy, reliability, and a steep learning curve for effective utilization. The study proposes strategic mitigation efforts, including improved training and algorithmic adjustments, user education and interaction design, continuous learning and adaption and enhanced contextual understanding, to address these challenges. This research contributes insights into the integration of Generative AI in low code automation development platforms, suggesting a promising yet challenging path forward in leveraging these technologies to foster more inclusive, efficient, and accessible development practices.

KEYWORDS: Generative AI, low code automation development, RPA

Contents

1 Introduction 6

1.1 Research problem, questions and objectives 9

1.2 Limitation 11

1.3 Structure of the study 11

2 Literature Review 13

2.1 Low code automation 13

2.1.1 Transformation of RPA with AI 21

2.2 Generative Artificial Intelligence and its implications 23

3 Methodology 30

3.1 Research design and strategy 30

3.2 Data collection 31

3.3 Data analysis 33

3.4 Limitations 34

4 Results 36

4.1 RPA development at the case company 36

4.2 Task Descriptions 38

4.3 Functionality enhancement, challenges and limitation and mitigation of them 40

5 Conclusion and discussion 57

5.1 Conclusion 57

5.2 Discussion 63

References 67

Appendices 73

Appendix 1. Structure of the interview 73

Appendix 2. Role of interviewees 74

Figures

Figure 1 Adoption advantages of RPA (Moreira, Mamede, & Santos, 2023) 17

Figure 2 RPA implementation challenges (Moreira, Mamede, & Santos, 2023) 20

Figure 3 Model, system, and application-level view on generative AI (Feuerriegel, Hartmann, Janiesch, & Zschech, 2023) 25

Figure 4 Taxonomy of the most popular generative AI models (Martínez, et al., 2023) 26

Figure 5 Identified themes from thematic analysis 41

Abbreviations

AI	Artificial Intelligence
DL	Deep Learning
ERP	Enterprise Resource Planning
IA	Intelligent Automation
LLM	Large Language Models
ML	Machine Learning
MVP	Minimum Viable Product
RPA	Robotic Process Automation
UVT	User Validation Testing

1 Introduction

Low code development is a technology that enables individuals without deep coding expertise to create custom applications and digital solutions for specific operational needs. This approach simplifies coding, allowing people directly involved in business operations to develop solutions. They use an intuitive graphical interface with pre-built components, enabling them to design applications through a drag-and-drop method. This process automatically generates the necessary code, allowing users with basic coding knowledge to independently develop fully functional digital tools without the need for professional software developers. (Martinez & Pfister, 2023)

In recent years, there has been an introduction of numerous low code tools designed to facilitate rapid application development. (Kirchhof, Jansen, Rumpe, & Wortmann, 2023) Recognizing the advantages of reduced development time and costs, increased accessibility, and simplified collaboration, an increasing number of companies are strategically incorporating these solutions into their digital transformation plans. In the face of the ever-evolving digital landscape, where modernizing software systems comes at a high cost, low-code development emerges as a practical and efficient solution. By leveraging these solutions, companies empower a broader range of employees, including roles have limited or no coding experience, to actively participate in the creation of digital solutions. This situation represents a fundamental reorientation towards user-centric, accessible development environments.

Companies leveraging these technologies are not only optimizing their internal processes but are also establishing a foundation for sustained competitiveness in an ever-changing business landscape. Low code solutions frequently serve as a bridge between various third-party services. They enable individuals without extensive programming knowledge to integrate and utilize third-party tools effectively. (Kirchhof, Jansen, Rumpe, & Wortmann, 2023) This democratization of development not only enhances collaboration but also accelerates the pace of innovation, enabling companies to respond promptly to market demands and maintain a competitive edge. In context of utilization

of the low code platforms with Enterprise Resource Planning (ERP) systems, and the synergy between low code platforms and ERP systems is highlighted. Low code platforms allow for the efficient and cost-effective creation of applications, while ERP systems offer crucial data and insights for managing business processes and operations. This synergy enhances organizational agility, essential for competitiveness in the digital economy (Picek, 2023). As the digital landscape continues to evolve, organizations are strategically positioning themselves to not only adapt to change but to thrive in an era where agility and innovation are important for sustained success.

Among the advantages offered by low code platforms, low code automation tools stand out for their ability to streamline business processes. These tools specialize in simplifying the automation of business processes, workflows, and routine tasks. By harnessing low-code automation, companies can automate complex processes with minimal coding effort, enhancing efficiency and reducing the time and resources spent on manual tasks (Bhattacharyya & Kumar, 2023). In an era where time is highly valued, automating repetitive tasks helps businesses to allocate their resources more judiciously, focusing on areas that generate value and drive competitive advantage. This strategic deployment of automation enhances productivity, fosters innovation, and enables companies to respond more agilely to changing market dynamics.

The low code no code approach has also transformed the landscape of process automation by democratizing the automation space and allowing individuals without diverse technical backgrounds to actively contribute to the enhancement of digital processes. The inclusion of Robotic Process Automation (RPA) in this research underscores the importance of task-specific automation in the broader context of digital transformation. RPA's ability to automate repetitive, rule-based tasks without altering underlying systems makes it a valuable tool for immediate efficiency gains. However, the process of creating these applications can still be time-consuming and error prone. Low code automation, coupled with the power of Generative Artificial Intelligence, has emerged as a transformative force in empowering businesses to achieve these goals more effectively.

This synergistic combination presents a paradigm shift, democratizing access to process automation for a broader range of employees while simultaneously accelerating the pace of automation. By 2024, organizations that have democratized access to digital collaboration and process automation tools for their frontline workers are anticipated to experience a revenue boost of 20% as a result of increased productivity (Loomis, 2022).

Generative AI has the potential to accelerate low code automation by automating many of the low-level tasks involved in the development process. By using Generative AI, developers can focus on the higher-level aspects of the automation. The adoption of Generative AI applications empowers business users by providing access to extensive internal and external information sources. This widespread use of Generative AI is set to democratize knowledge and skills within enterprises, with large language models enabling a conversational style of interaction that incorporates rich semantic understanding (Gartner, 2023). Large language models refer to a specific type of AI algorithm that can execute many kinds of tasks like text generation, text analysis, translation, sentiment analysis, question answering, and other related functions (Orru et al., 2023; Rajaan et al., 2024).

This thesis explores the dynamics between Generative AI and low code automation platforms, focusing on their collaborative potential to accelerate automation development processes by enhancing problem-solving capabilities, and democratizing technology development. Through a detailed investigation and practical application, this study aims to answer questions regarding the effectiveness, challenges, mitigations, and prospects of employing Generative AI within low code automation development. The low code automation development can be defined as a development approach that minimizes the need for manual coding, enabling faster automation development and deployment through visual programming environments and pre-built templates.

The aim of this study is to explore and understand the implications of integrating Generative AI into low code automation development tools with a specific focus on RPA. The

study seeks to assess how Generative AI can enhance or transform the functionality of these tools, thereby potentially accelerating the development process and making it more accessible to non-expert users. Additionally, the study aims to identify the potential challenges and limitations that might arise from the utilization of Generative AI in low code automation development environments. Finally, the study intends to propose strategies to mitigate the identified challenges and limitations, aiming to maximize the benefits of Generative AI in low code automation development while minimizing the risks. This research aims to provide valuable insights for developers, businesses, and policymakers on harnessing the power of Generative AI in low code automation tools, thereby laying the groundwork for more innovative, efficient, and inclusive development practices.

1.1 Research problem, questions and objectives

Despite the increasing adoption of Generative AI in the low code automation development tools, there is a significant knowledge gap regarding its practical implementation and the potential advantages it offers to businesses. This gap underscores the need for a deeper investigation into how Generative AI can be effectively utilized in low code automation development to enhance automation capabilities, streamline development processes, and unlock new opportunities for innovation. As companies increasingly look to leverage these technologies to drive efficiency and foster innovation, identifying the specific benefits, operational improvements, and competitive advantages Generative AI can bring to low code automation development becomes important. This exploration is crucial not only for maximizing technological investment but also for shaping the future landscape of digital transformation and process automation.

This research aims to explore the integration of Generative AI into low code automation development tools, focusing on its impact, associated challenges, and strategies to overcome these challenges. The study is structured around three core objectives, each corresponding to the presented research questions.

RQ1: How does the integration of Generative AI influence the functionality of low code automation development tools?

This question seeks to understand the transformative effects of Generative AI on low-code platforms, focusing on automation capabilities, development efficiency, and the broadening of automation scopes. It delves into the qualitative aspects of how Generative AI enhances low code automation development, based on the experiences and insights of users and experts.

RQ2: What are the potential challenges and limitations in using Generative AI in low code automation development?

This question aims to identify the hurdles faced when using Generative AI for low code automation development. It focuses on identifying the potential challenges and limitations associated with utilizing Generative AI in low code automation development. This encompasses technical hurdles, such as integration complexities and operational challenges. By pinpointing these challenges, the research will shed light on the practical difficulties of utilizing Generative AI within low code automation development.

RQ3: How can the previously mentioned challenges and limitations be mitigated?

Expanding upon the exploration of challenges and limitations associated with utilizing Generative AI in low code automation development, this research question shifts the focus towards formulating actionable strategies to address the challenges and limitations identified. The objective is to provide a set of best practices and recommendations for seamlessly utilizing Generative AI in low code automation development, thereby maximizing the potential benefits while minimizing the drawbacks.

1.2 Limitation

Security is an essential component within AI systems. It plays a pivotal role in protecting data and privacy. As AI technologies continue to advance, their integration into various facets of society and industry underscores the importance of implementing robust security measures. Given the vast amounts of data that AI systems handle, often containing personal and confidential details, the need for robust security measures is important. However, the complexities of AI introduce unique security challenges that extend beyond traditional approaches. The dynamic nature of AI, characterized by its adaptability and learning capabilities, necessitates a comprehensive understanding and implementation of security protocols to mitigate potential risks and vulnerabilities.

While recognizing the significance of security and ethical considerations in AI, this thesis study has intentionally narrowed its scope to concentrate on specific facets of AI without delving into the complexities of these areas. This decision arises from the need for a focused and comprehensive exploration of core AI concepts.

The exclusion of security and ethical considerations from this thesis does not diminish their crucial role in the responsible development and deployment of AI. Rather, it underscores the need for dedicated studies that comprehensively address these complex and multifaceted domains. As AI continues to evolve, ensuring its ethical implementation will remain an ongoing effort, requiring collaboration among various fields to navigate the ethical and societal implications of this rapidly changing technology.

1.3 Structure of the study

The proposed study aims to explore the integration of Generative AI within low code automation development tools, focusing on its impacts, the challenges of implementation, and strategies for overcoming these challenges. The structure of this study is

designed to guide the reader through a comprehensive journey from the introduction of the topic to the final recommendations and conclusions.

The study begins with an introduction, setting the stage for the investigation. This section provides an overview of Generative AI in enhancing low code automation tools, articulates the research problem, and outlines the study's objectives and research questions. Following the introduction, the literature review delves into the theoretical foundations and current state of research on the integration of Generative AI within low code automation tools. The methodology section outlines the qualitative approach taken, detailing the data collection methods, participant selection criteria, and the process for analyzing the data collected from semi-structured interviews. The case study section presents the key themes emerged from the interview data, providing in-depth analysis on how Generative AI influences low code automation development tools, the challenges faced in utilizing Generative AI features, and the identified strategies to mitigate these challenges. The discussion section interprets these findings within the broader context of the study's objectives and the existing literature. Concluding the study, the conclusion summarizes the key findings. It offers recommendations for practitioners in the field of low code development and proposes directions for future research. This final section emphasizes the study's contributions to understanding the integration of Generative AI into low code automation tools and its potential impact on technology development practices. Furthermore, references and appendices follow the conclusions section and provide supplementary materials.

2 Literature Review

This section examines the intersection of low code automation development with specific focus on RPA and Generative AI, illustrating how this blend is impacting the low code automation landscape.

2.1 Low code automation

This section delves into the streamlined approach of developing applications with minimal coding effort. It starts with the foundational concepts and importance of low code platforms in the tech landscape. Following that, it examines RPA as a tool for automating routine tasks. Finally, it discusses the enhancement of RPA capabilities through the integration of AI, which allows for more complex and intelligent process automation.

The evolution of low code automation represents a significant milestone in the development practices of software, emphasizing a shift toward more accessible, efficient, and flexible application design and deployment. This transition is marked by the move from traditional, code-intensive methods to more inclusive and simplified approaches, catering not only to experienced developers but also to non-technical domain experts. The journey of low code automation is fundamentally a narrative of technological advancement and democratization, offering a lens through which the continuous transformation of software development can be appreciated (Juhas, et al., 2022).

The term "low code" was first mentioned by Forrester in 2014, introducing a new paradigm aimed at simplifying application development. This approach leverages principles like visual programming and model-driven development, utilizing various diagrams to expedite the development process. This foundational step has set the stage for an evolutionary trajectory where, as predicted by Gartner, 70% of new applications developed by organizations will employ low code platforms by 2025, showcasing the burgeoning significance of low code in the future of software development (Juhas et al., 2022; Ray et al., 2023).

The progression of low code automation has also been marked by the integration of AI-powered tools such as ChatGPT and Copilot. These advancements underscore the synergistic relationship between visual programming interfaces and natural language instruction tools, significantly enhancing programmers' productivity and task success. This blend of technologies not only simplifies the coding process but also enriches the development ecosystem with intuitive and efficient programming methodologies (Rao, Tsay, Kiran, Hellendoorn, & Hirzel, 2023).

Further reflecting on the historical evolution, low code development has been pivotal in democratizing software development. It empowers "citizen developers" to create software using visual interfaces, while still accommodating extensions in traditional programming languages for more sophisticated functionalities. Citizen developer is a non-technical individual within an organization who, driven by the need for rapid innovation and process efficiency in a volatile economic and rapidly evolving technological environment, utilizes low code or no code platforms to develop applications or automate processes. These individuals possess a profound understanding of their domain's specific needs and challenges, enabling them to tailor solutions directly addressing these issues without the direct involvement of IT departments or professional software developers. This inclusive approach has been instrumental in bridging the gap between non-programmers and IT professionals, fostering a collaborative environment conducive to meeting business needs swiftly and effectively (Brüggemann, 2023; Pezzini & Khiyara, 2023).

Gartner predicts a substantial shift towards low code development, projecting that by 2025, 70% of new applications developed by organizations will leverage low code platforms. This growth underscores low code's emergence as a pivotal element in software development, driven by its ability to address critical industry challenges, such as the shortage of skilled developers and the increasing demand for rapid application development (Juhas, et al., 2022). In conjunction with AI tools like ChatGPT, low code automation

is transforming how software is developed, enabling a broader range of people, including those without extensive programming expertise, to create and refine AI-driven applications. This democratization of AI application development, facilitated by tools like LowCoder that combine visual programming interfaces with AI-powered natural language interfaces, not only boosts productivity but also accelerates innovation by making AI more accessible to citizen developers. As such, low code approaches, especially when integrated with AI, are setting a new standard in software development by reducing barriers to entry, fostering inclusivity, and empowering users to implement complex AI pipelines with minimal coding effort. (Rao, Tsay, Kiran, Hellendoorn, & Hirzel, 2023)

Moreover, low code platforms are noted for their ability to democratize software development, enabling domain experts to create applications without needing a formal programming background. These platforms also allow for traditional programming extensions to meet more complex needs, although this does highlight the necessity for additional support in security and advanced coding techniques for citizen developers (Brüggemann, 2023).

Low code development offers substantial advantages over manual coding by fostering a more inclusive, flexible, and rapid development environment. It empowers individuals with limited coding knowledge to contribute effectively, aligns closely with business needs, and accelerates the development lifecycle. This approach is reshaping how software is conceptualized, developed, and deployed, underscoring a significant paradigm shift in the tech industry (Rokis, Karlis ; Kirikova, Marite, 2023).

Robotic Process Automation is a term coined in 2012 by Pat Geary who was the chief evangelist for Blue Prism at that time. According to Taulli, despite its name, "robotic" refers to software-based robots or bots, not physical robots, that automate human actions in clerical and administrative tasks. These bots emulate human actions within digital systems, capturing data, manipulating applications, and executing business processes efficiently (Taulli, 2020).

Gartner describes Robotic Process Automation (RPA) as a software technology that automates tasks within business and IT processes through software scripts that mimic human interactions with application interfaces (Ray, et al., 2023). It is applicable to all processes classified as support or back-office functions. Furthermore, it is suitable for implementation in enterprises of various types that feature a centralized hub of processes. (Uklańska, 2023)

According to a literature review conducted by Moreira et al. (Moreira, Mamede, & Santos, 2023), the following four phrases are most frequently used phrases for definition of RPA in selected papers:

- Emerging/advanced technology
- Software that performs human tasks
- Automation of routine and rule-based tasks
- Software that autonomously executes processes, tasks, activities, and transactions

It echoes the broader understanding of RPA as not just a technological innovation but as a transformative tool designed to streamline business operations. By emphasizing RPA's role in handling mundane, repetitive tasks, this analysis highlights its potential to free up human workers for more complex tasks. Additionally, the characterization of RPA as autonomous software underscores its ability to operate with minimal human intervention, further enhancing operational efficiency and accuracy. This evolution towards more sophisticated automation technologies signals a shift in how businesses approach task management and process optimization, marking a pivotal step towards the future of work.

Transitioning from this broad perspective, it's important to delve into the tangible impacts of RPA. A research paper by Grande et al. explores the multifaceted benefits of

RPA, particularly highlighting the significance of integrating RPA into the strategic framework of an organization. The study illustrates how RPA can lead to significant cost savings, enhanced accuracy, and improved productivity. Grande et al. stress the importance of aligning RPA with the overall strategic goals of the company, ensuring that the technology is not merely an operational add-on but an integral component of the organizational strategy. This perspective emphasizes a comprehensive approach to RPA deployment, advocating for its incorporation as a key element in driving organizational success and efficiency. (Grande, Campos, Facin, & Batistela, 2022) This insight underscores the need for a holistic approach to RPA adoption, where the technology is not just an operational tool but a part of the broader strategic vision of the organization.

The following figure presents data on the adoption advantages of RPA and the count of references derived from a literature review which is conducted by Moreira, Mamede and Santos.

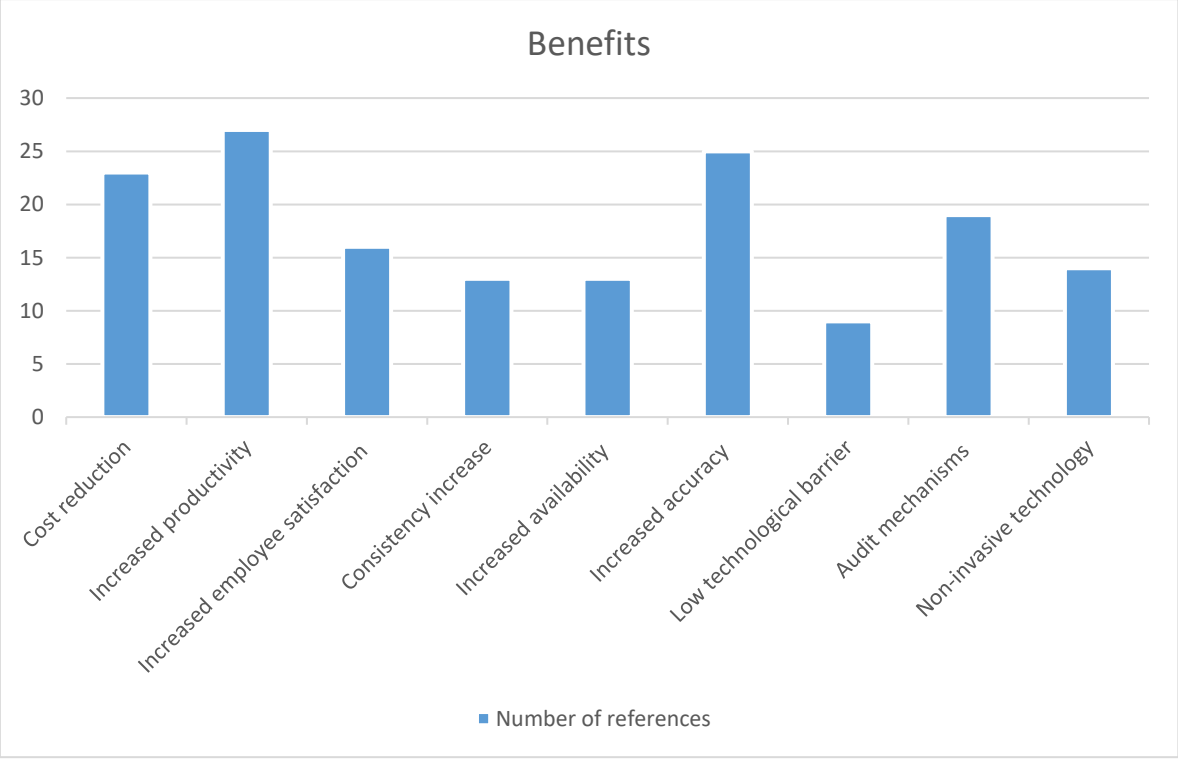

Figure 1 Adoption advantages of RPA (Moreira, Mamede, & Santos, 2023)

Enhanced productivity emerges as the most cited advantage of RPA, mentioned 27 times, highlighting its importance and broad recognition among researchers. Cost reduction and improved accuracy also receive considerable attention, underlining the financial and operational benefits that RPA brings to the table. Pramod's 2022 research resonates with the acknowledged positive effects of RPA, reaffirming its contributions increased productivity, cost savings, and improved accuracy. However, it also highlights the inherent challenges that call for careful planning and management (Pramod, 2022). This balance between benefits and challenges is a recurring theme in RPA literature. For instance, Prabodha and Liyanage identifies a lack of interdepartmental coordination as a major barrier in the logistics industry, suggesting that while RPA can streamline operations, it also necessitates effective communication and collaboration across different departments (Prabodha & Liyanage, 2023).

In the context of accounting and auditing, Perdana, Lee, and Kim discuss RPA's potential to enhance the efficiency and accuracy of audit tasks. They provide insights into both the benefits and real-world implementation challenges, indicating that while RPA can significantly improve process efficiency, it also requires a thorough understanding of the processes to be automated (Perdana, Lee, & Kim, 2023).

The research by Grande et al. and Pramod echoes these benefits, noting RPA's role in cost savings, accuracy, productivity, and consistency in sectors like banking, healthcare, and manufacturing. However, challenges remain, as Eulerich et al. pointed out, including issues related to control, security, governance, and the potential loss of process knowledge (Eulerich, Waddoups, Wagener, & Wood, 2023; Grande, Campos, Facin, & Batistela, 2022; Pramod, 2022).

Eulerich et al. provide a broader perspective on the challenges associated with RPA. They identify five critical challenges, including the tendency to use RPA as a temporary solution to deeper system issues, control and security risks, misunderstandings about the true costs involved, complex governance requirements, and the potential loss of process

knowledge. These challenges underscore the necessity for a strategic approach to RPA, moving beyond seeing it as a quick fix to operational issues. (Eulerich, Waddoups, Wagener, & Wood, 2023)

Moreira et al. reveals challenges related to humans, organizational and technological factors. The integration of automation technology often triggers concerns among employees, predominantly due to the fear of job displacement. This situation necessitates a strategic approach by the management to assure employees that the objective of RPA is to complement their skills, not replace their roles. Simultaneously, the challenge of identifying which processes to automate is integral to the successful deployment of RPA. The decision is not straightforward, as it demands a thorough analysis of the organization's operations to pinpoint tasks. Complementing these challenges is the widespread lack of in-depth RPA knowledge within organizations. This gap manifests as a barrier to effective implementation, with many organizations struggling to grasp the full spectrum of RPA capabilities and limitations. Overcoming this hurdle often requires significant investment in training and development or the acquisition of external expertise. (Moreira, Mamede, & Santos, 2023) The following figure presents data on the challenges in implementing RPA and the count of references derived from a literature review which is conducted by Moreira, Mamede and Santos.

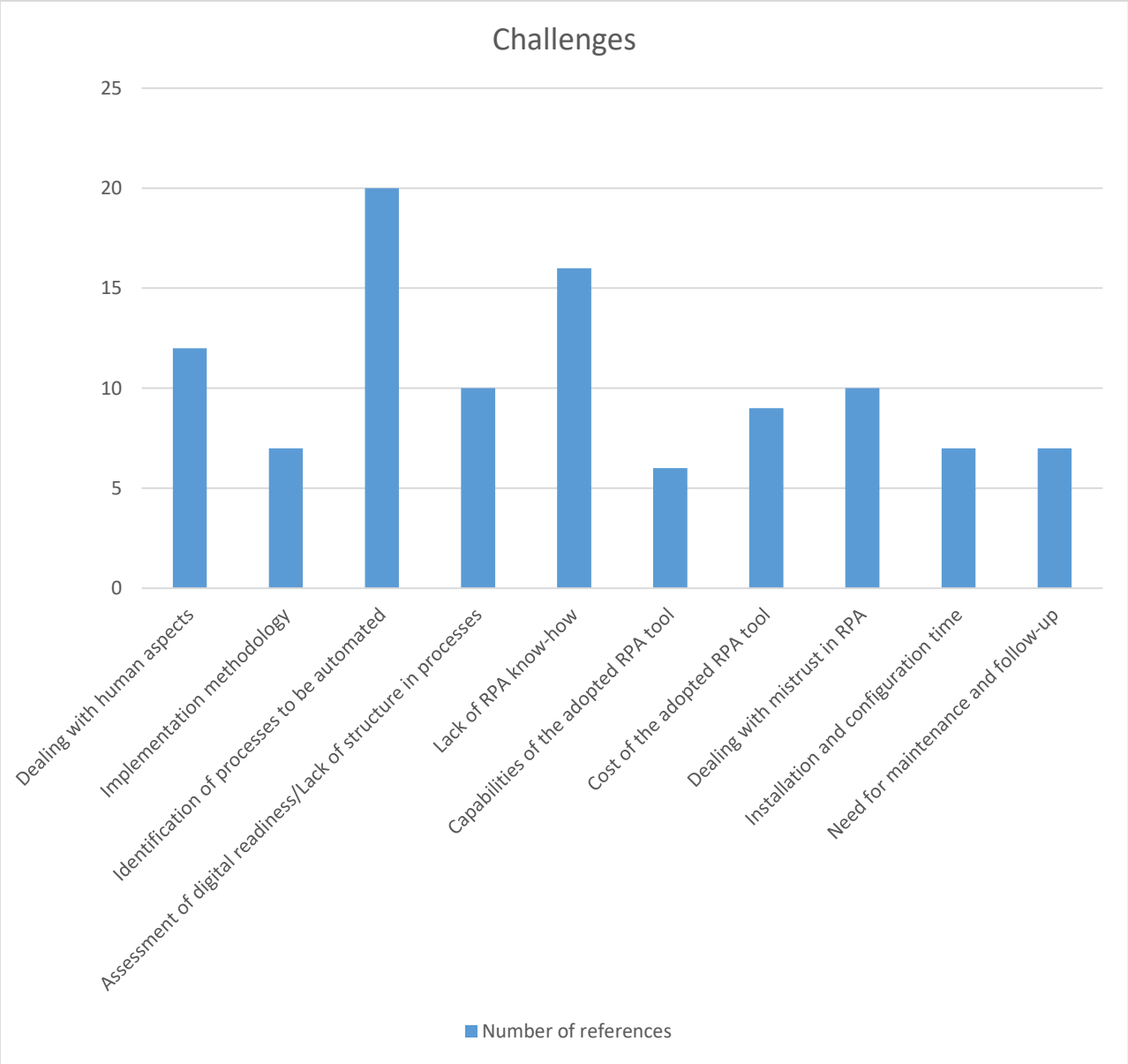

Figure 2 RPA implementation challenges (Moreira, Mamede, & Santos, 2023)

The challenge of automating processes stands out, suggesting that a significant concern in RPA implementation is identifying and adapting processes that lend themselves to automation. This is closely followed by the difficulties related to the lack of RPA know-how and the lack of structure in processes, emphasizing the need for a clear understanding of both the technological aspects of RPA and the business processes at hand. Challenges such as overcoming resistance from employees and fostering confidence in RPA underscore the human and organizational dimensions of adopting new technology.

These issues highlight the importance of addressing the concerns and skepticism that employees may have towards automation and its impact on their jobs.

2.1.1 Transformation of RPA with AI

According to Taulli, RPA alone doesn't transform an organization entirely; for that, end-to-end intelligent automation is needed (Taulli, 2020). Kedziora and Hyrynsalmi also discuss the shift towards intelligent automation with machine learning in RPA technology, showcasing its application in invoice processing. This case study exemplifies the tangible benefits and operational efficiencies achievable when layering AI capabilities on top of RPA frameworks (Kedziora & Hyrynsalmi, 2023). Furthermore, the integration of RPA with natural language processing (NLP) technology in the domain of financial office automation, as explored by Kai et al. (2022), underscores the breadth of AI's applicability in enhancing RPA's utility beyond mere task automation to more complex cognitive functions (Kai, et al., 2022). This echoes the sentiment that intelligent automation is the cornerstone of a digitally transformed enterprise.

The projected growth in the AI and automation-related software market segments further underscores their importance. These segments are expected to grow at a cumulative annual growth rate of over 35% over the next five years, a rate significantly higher than that of the broader software market (Ray, et al., 2023). This growth highlights the role of intelligence and automation not just as tools, but as foundational enablers in running a digital business at a scale and preparing enterprises to be future-ready.

The significance of AI and automation in enterprise digital transformation is further highlighted by findings from IDC’s Worldwide Enterprise Intelligence Services Survey 2023. The survey reveals that the public launch of generative AI platforms like ChatGPT has more than doubled the number of enterprises actively exploring and implementing generative AI capabilities. These enterprises are experiencing significant benefits from generative AI in areas such as IT and cloud operations, data management and analysis,

cybersecurity, compliance, and in the automation of coding and development operations (Shanbhag & Sivalingam, 2023).

In 2022, the RPA software market grew to $2.8 billion, an increase from $2.3 billion in 2021. While this market was once the fastest-growing segment in software, it ranked as the eighth-fastest-growing in 2022. Despite its annual growth rate of 22% significantly outpacing the average worldwide software market growth of 11%, the expansion rate of the RPA market has decelerated over recent years, falling from 62% in 2019 to 39% in 2020, and further to 31% in 2021 (Ray, et al., 2023).

According to Ray et al., as of January 31, 2023, the RPA market encompassed over 60 vendors. They further forecast that by the year 2025, it is anticipated that 90% of vendors providing RPA will include generative AI-assisted automation in their offerings (Ray, et al., 2023).

ChatGPT and other large language models (LLMs) are significantly influencing automation in both business and IT sectors. The majority of RPA vendors are actively engaging with GPT models, with many either already incorporating them into their products or planning to do so. This includes integrating APIs from established GPT providers like OpenAI and introducing direct functionalities such as automated email creation, workflow design prompts, improved training for intelligent document processing (IDP) models, enhanced process-mining analytics, and the AI-powered development of RPA scripts. Over the coming one to two years, numerous RPA vendors are expected to roll out additional features, solidifying AI-generated automation as a key offering in the industry (Ray, et al., 2023).

2.2 Generative Artificial Intelligence and its implications

This section delves into the foundational elements of Generative AI and examines its outcomes.

In the realm of intelligent systems, the evolution from broad concepts of AI to the specialized functions of Generative AI has marked a significant shift towards increasingly sophisticated technological capabilities. This progression encapsulates the journey from machines performing basic cognitive tasks to creating novel content, mirroring the intricacies of human intelligence and creativity.

AI represents the widest spectrum of technologies that encompasses any technique which enables computers to mimic human behavior (Russell & Norvig, 2021). Janiesch et al. describes Machine Learning (ML) and Deep Learning (DL) as integral components of AI technology, where DL is an advanced manifestation of ML characterized by its deep neural network architectures. This relationship underscores the evolution and sophistication of AI systems capable of performing complex tasks with high accuracy and efficiency. The connection between ML and DL is foundational and hierarchical. Deep learning extends the principles of machine learning, focusing on a specific set of tools and models that are particularly effective for dealing with large-scale and high-dimensional data. While all deep learning models operate within the broader machine learning framework, not all machine learning models employ deep learning techniques. This distinction highlights the evolution of AI from simple, rule-based algorithms to complex systems capable of learning and making decisions with minimal human intervention (Janiesch, Zschech, & Heinrich, 2021).

The applications of these technologies are vast, stretching from healthcare, where they're used for diagnosis and disease treatment (Shamshirband, Fathi, Dehzangi, Chronopoulos, & Alinejad-Rokny, 2021) to cybersecurity, where machine learning technologies are used for creating cyber-attack detector (Mozo, et al., 2023). In every

application, the goal is to harness the strengths of AI, ML and DL to solve problems more efficiently and effectively than ever before.

The recent advancements within AI have expanded its scope, with newer algorithms and models that allow for a broader range of cognitive functions and more sophisticated decision-making processes, thus enabling AI systems to perform tasks that traditionally required human intelligence. It further accelerated by the release of platforms like ChatGPT in late 2022, which have expanded the public's engagement with and understanding of Generative AI's potential. (Kalota, 2024)

The survey conducted by KPMG in March 2023 among 255 business leaders highlights a robust momentum behind generative AI, with optimism for its value addition. Nearly all surveyed leaders (93%) believe that generative AI will provide value to their business. Moreover, a majority (80%) foresee a rise in generative AI investment by more than 50% in the next 6-12 months, mainly to enhance infrastructure and business scaling (KPMG, 2023).

Generative AI represents a frontier in computational innovation, characterized by its capacity to generate original and meaningful content across various mediums, such as text, images, and audio. This is achieved through sophisticated computational methodologies that leverage extensive training datasets, marking a significant leap in how technology can emulate creative processes. As highlighted by Feuerriegel, Hartmann, Janiesch, and Zschech, this rapidly evolving field has given rise to transformative applications, including Dall-E 2, GPT-4, and Copilot, reshaping our way of working and communicating (Feuerriegel, Hartmann, Janiesch, & Zschech, 2023). Building on this foundation, Kalota underscores the expansive impact of generative AI beyond technological aspects, suggesting its potential to broadly influence society, encompassing individuals, businesses, and organizations (Kalota, 2024). Together, these perspectives underscore the vast scope and profound **implications of Generative AI**, signaling a paradigm shift in both technological capability and societal interaction.

Infographic outlines how Generative AI models are developed, integrated into systems for interaction, and ultimately applied to serve various business and creative needs. The model level is the foundational layer where the underlying AI models for various data modalities are developed. At system level, the AI models are embedded into systems to provide an interface for interaction, turning them into user-friendly tools or services. The application level is where the generative models and systems are applied to solve real-world business problems and meet stakeholder needs.

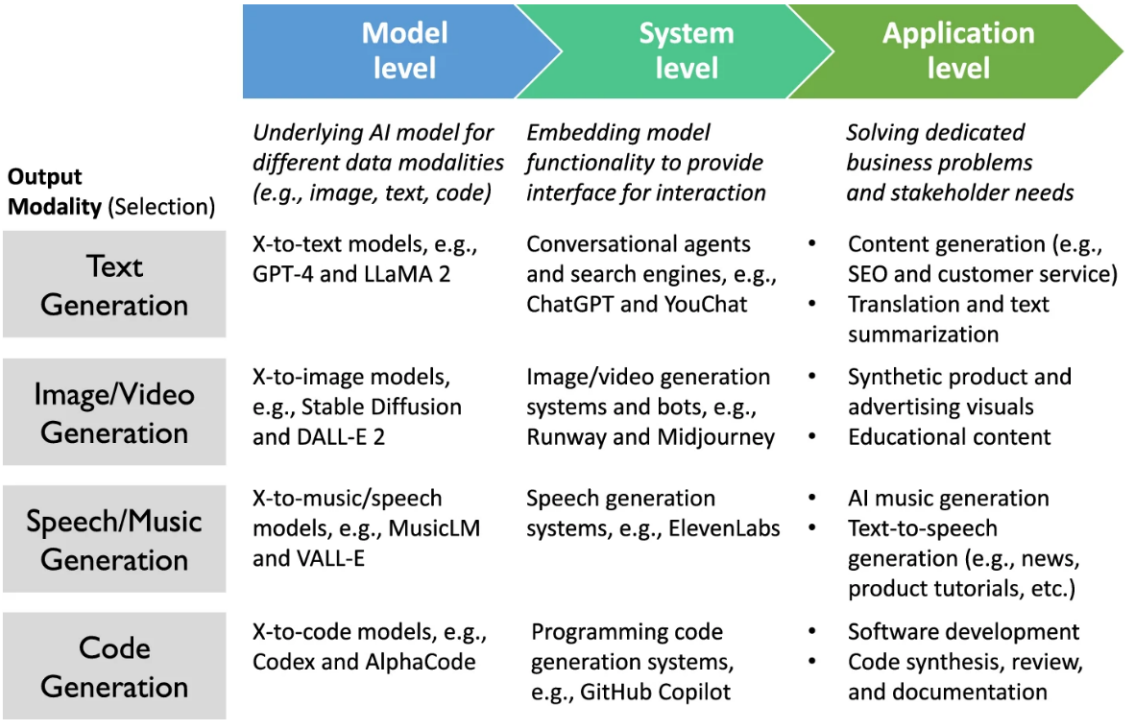

Figure 3 Model, system, and application-level view on generative AI (Feuerriegel, Hartmann, Janiesch, & Zschech, 2023)

The applications of Generative AI are described as nearly limitless, with the ability to train models to generate diverse multimedia formats from various input formats. This flexibility highlights the potential of Generative AI to innovate across different domains. (Gozalo-Brizuela & Garrido-Merchan, 2023)

The following figure which is constructed by Gozalo et. al illustrates a taxonomy of Generative AI models that captures the breadth of the field. This breadth of applications underscores the transformative potential of Generative AI across sectors, including its possible integration into low code automation platforms to facilitate the automatic generation of process automations and functionalities, further simplifying the development process for users without deep technical expertise.

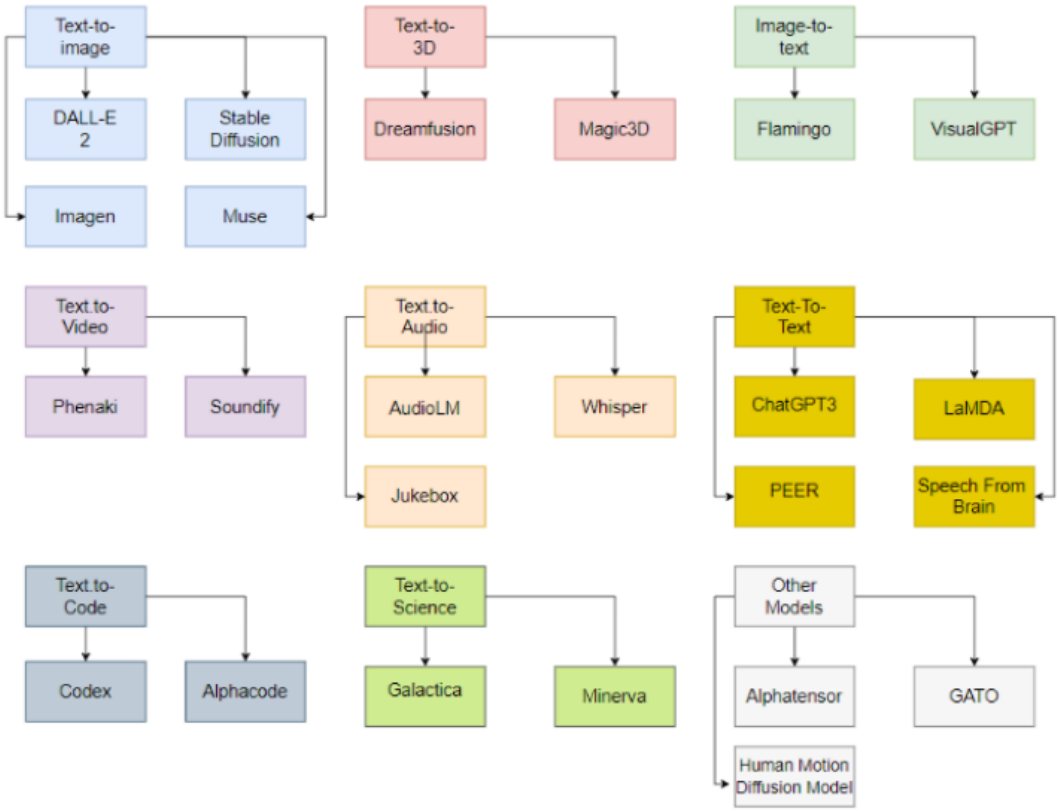

Figure 4 Taxonomy of the most popular generative AI models (Martínez, et al., 2023)

Based on the study conducted by Walkowiak, examining the interdependencies between Generative AI and human employees in work environments, the workers and Generative AI are not merely parallel contributors to productivity but are deeply interdependent. This interdependency is characterized by a network externality in learning, where the collaboration and interaction between human employees and Generative AI significantly

drives productivity improvements. Particularly, the study finds that in environments where open learning is encouraged, there is a clear benefit to strategically matching Generative AI with workers based on their ability to detect errors. This approach not only optimizes the productivity gains from Generative AI integration but also addresses the potential risks and limitations associated with the indistinguishability of human-created and AI-generated outputs. The analysis provided by Walkowiak sheds light on the complex dynamics at play when integrating Generative AI into the workforce, offering a valuable framework for understanding how task interdependencies and learning externalities can be leveraged to enhance organizational productivity. (Walkowiak, 2023)

Generative AI can enhance the productivity and retention of individual workers. (Brynjolfsson, 2023) The role of Generative AI in enhancing developer productivity is also notable, with studies like that of Peng et al. demonstrating how tools like GitHub Copilot can drastically reduce task completion times, suggesting a promising future for AI in democratizing software development. The research conducted by Peng et al. presents findings from a controlled experiment where developers tasked with implementing an HTTP server in JavaScript were able to complete their tasks much faster with the aid of an AI pair programmer. Such evidence points to the promise of generative AI in not only enhancing productivity but also potentially aiding individuals in transitioning into software development careers (Peng, Kalliamvakou, Cihon, & Demirer, 2023). Another research which is conducted by Thomas Dohmke et al. provides a comprehensive analysis of how generative AI, particularly GitHub Copilot, influences productivity in software development. According to Thomas Dohmke et al. GitHub Copilot has significantly increased productivity among developers, with users accepting nearly 30% of code suggestions. This impact is even more pronounced over time, suggesting a learning effect where users become more adept at utilizing these suggestions. The benefits are particularly substantial for less experienced users, indicating that GitHub Copilot not only improves productivity but also accelerates the learning curve for newer developers (Dohmke, Iansiti, & Richards, 2023).

Despite these promising advancements, not all findings regarding the impact of Generative AI on software development are uniformly positive. Some research indicates that the use of generative AI platforms like ChatGPT may not significantly differ from traditional resources in terms of productivity and could even lead to increased frustration among users (Choudhuri, Liu, Steinmacher, Gerosa, & Sarma, 2023). This highlights the need for careful integration and usage of Generative AI tools.

The reliability and quality of code generators are critical, as highlighted by Boussaa et al., who emphasize the importance of verifying code generators' behavior to maintain software quality (Boussaa, Barais, Sunyé, & Baudry, 2020). Furthermore, the potential for human-AI collaboration in tasks like code translation is gaining traction.

The architecture of generative models and the vastness of their training data are critical aspects. Gozalo-Brizuela et al. note the feasibility of training models on extensive datasets, including Wikipedia, GitHub, social networks, and Google Images. The rise of computing power allows for the development of sophisticated neural networks, transformers, generative adversarial networks, and variational autoencoders (Gozalo-Brizuela & Garrido-Merchan, 2023). However, the exploration of Generative AI's interaction with the internet conducted by Martínez et al. raises intriguing questions about the future behavior of AI models trained on a mixture of real and AI-generated data. According to Martínez et al. the quality of generated content may degrade over time, indicating potential challenges in maintaining the effectiveness of Generative AI tools as they become increasingly integrated into digital ecosystems. (Martínez, et al., 2023)

Jiao et al. delves into the critical aspect of understanding the scopes of inputs and outputs within generative AI systems for code generation. Through a series of scenario-based design workshops with software engineers, the research highlights the complexities and challenges users face in grasping the capabilities and limitations of Generative AI models. Significant interest was shown in identifying the types of inputs that would lead the system to create high-quality code. This interest underscores the necessity for

AI systems not only to accept a wide range of input types but also to guide users in formatting these inputs to maximize the utility of the AI's output. Equally important is users' understanding of the system's outputs. The research reveals that users have diverse expectations regarding the outputs, including the code's style, efficiency, and adherence to best practices. These findings point to a need for Generative AI systems to offer customizable outputs or at least provide clear explanations about the nature and rationale behind the generated code. The discussion on output scopes also touches on the importance of transparency in AI-generated artifacts, suggesting that users benefit from knowing the limitations and strengths of the code produced by the AI (Jiao, et al., 2022).

Research indicates that Generative AI is poised to significantly impact various fields, including low code automation platforms. By enhancing developer productivity, simplifying content creation, and broadening the scope of what non-experts can achieve in software development, Generative AI holds promise for further democratizing technology creation and application. As generative AI continues to evolve, its integration into different sectors promises to bring both opportunities and challenges.

Generative AI will play a crucial role in democratizing software development, enabling more individuals from non-technical backgrounds to contribute to software innovation. This democratization is expected to address the growing demand for software development talent and help bridge the gap between supply and demand in the global job market (Dohmke, Iansiti, & Richards, 2023).

3 Methodology

This section outlines the research design and strategy, detailing the rationale for the chosen approach and how it aligns with the study's goals. It explores data collection methods, including techniques for data gathering and criteria for selecting sources. The discussion progresses to data analysis, highlighting the methodologies used to interpret data and contribute to the study's findings.

Additionally, this section discusses the limitations of the research, acknowledging potential constraints and their impact on the study's outcomes. By focusing on these limitations, the section underscores the study's transparency and the consideration of factors that might affect the results.

3.1 Research design and strategy

This study adopts a qualitative research design, employing an exploratory case study approach to investigate the evolving dynamics between Generative AI technologies and low code automation development. Qualitative research encompasses a diverse array of methodologies and philosophies. At its core, it is an approach designed to explore people's experiences in depth through various research methods like in-depth interviews, focus groups, observations, content analysis, and life stories (Hennink, Hutter, & Bailey, 2020). The focus on qualitative design is driven by the necessity to capture the nuanced impacts, challenges, and potential mitigation strategies associated with Generative AI usage in this specific technological context.

Qualitative research embarks on its journey with broad, explorative questions that dynamically evolve throughout the investigative process, allowing researchers to refine their aims or objectives based on emerging insights. This iterative nature fosters a deep engagement with the subject matter, enabling a nuanced understanding that adapts to the complexities of the research context. In contrast, quantitative research adheres to a

predetermined set of specific questions, maintaining their original form from inception to conclusion. This fundamental difference underscores the qualitative approach's flexibility and its capacity to uncover the depth of human experiences and perspectives (Denny & Weckesser, 2022).

The qualitative nature of this study enables a thorough examination of subjective experiences and operational realities associated with Generative AI's role in enhancing the functionality of low code automation development platforms. The exploratory case study method is chosen for its flexibility and depth, allowing for an in-depth examination of the practical applications, challenges, and strategic considerations associated with Generative AI technologies within specific low code automation development. According to Tracy, the qualitative approach emphasizes rich rigor, sincerity, and meaningful coherence, ensuring the study's findings are credible, resonant, and ethically conducted (Tracy, 2010), thereby contributing significantly to our understanding of Generative AI in low-code automation development.

3.2 Data collection

The methodology for gathering data in this research is conducted through semi-structured interviews, which enables a detailed comparison of the entities under study. Semi-structured interviews, while based on a standardized format, inherently offer flexibility in their execution, allowing for variations in the sequence of questions and the in-depth exploration of topics as responses dictate (Groenland & Dana, 2019). This flexibility is instrumental in adapting the interview process to the specific needs of the research while ensuring comprehensive coverage of the subject matter. Appendix 1 outlines the interview questions that ensure consistency across interviews while allowing flexibility for in-depth exploration.

Before each interview, participants were familiarized with the main themes and questions of the study. This preparatory step was crucial in providing the interviewees with a

comprehensive understanding of the various Generative AI capabilities and methodologies that could be applied in the development of low code automations. This groundwork not only ensured that the participants were well-informed about the study's focus but also enabled them to reflect deeply on their experiences and insights regarding the use of Generative AI in their low code automation development practices. Such thorough familiarization was instrumental in facilitating meaningful and informed discussions during the interviews.

The interviews were conducted over a period stretching from February 21st to March 22nd, 2024, each lasting approximately 20 to 45 minutes. Prior to these sessions, the researcher outlined a privacy policy, ensuring that all participants were made aware that their personally identifying information would remain confidential throughout the process. With the explicit consent of each interviewee, all sessions were recorded.

The interviews for the research were conducted remotely via the Teams application which eliminated the logistical challenges and costs associated with physical travel, while still capturing the essence and dynamic interaction of face-to-face meetings. The virtual format of the interviews brought the possibility of recording from start to finish. The recordings became an asset for the research process, providing a rich source of data that could be revisited and analyzed in depth. This method of data collection not only enhanced the efficiency of the research process but also enriched the quality of the research findings.

In total, ten people were interviewed, representing a cross-section of roles and experiences within the domain of low code automation development. In the selection of participants for the study, a purposive sampling strategy was employed to identify and engage individuals who possess specialized knowledge and expertise in the field of RPA as well as citizen developers who actively engage in developing automation solutions. This choice ensured that the study would benefit from a depth of insight and practical

experience, directly relevant to the research objectives concerning the integration and implications of Generative AI in low code automation tools.

By focusing on these two distinct groups, the research aimed to capture a comprehensive range of perspectives: from seasoned RPA experts, who bring a wealth of technical knowledge and industry experience, to citizen developers, who contribute a unique understanding of the accessibility, usability, and practical challenges faced by non-traditional developers in leveraging new Generative AI technologies within low code automation platforms.

This purposive sampling approach was instrumental in ensuring that the study could delve into the nuanced ways in which Generative AI influences the functionality and development process of RPAs. It also facilitated an examination of the potential challenges and limitations of implementing Generative AI from both a technical and user-centric viewpoint. By selecting participants who are directly involved in the development of low code automation solutions, the research positioned itself to uncover valuable insights into how these technologies can be optimized to meet the evolving needs of developers and businesses alike, thereby enriching the quality and applicability of its findings.

3.3 Data analysis

The thematic analysis approach is adopted to ensure a comprehensive analysis of the data collected from semi-structured interviews. The methodology behind thematic analysis is designed to ensure coherence and rigor in qualitative research. It follows a six-step framework that involves the careful definition of themes, which are then scrupulously examined against the dataset to ensure they accurately represent the underlying data. This approach not only enhances the validity of the research findings but also enriches the research narrative by providing a deep understanding of the data (Braun & Clarke, 2006; Lochmiller, 2021; Vaismoradi, Turunen, & Bondas, 2013).

The stages of thematic analysis, as described by Braun and Clarke, emphasize a structured yet flexible approach to qualitative data analysis. This methodological framework allows researchers to dissect and interpret complex data sets systematically, enabling them to extract meaningful insights and themes from their data. The process involves familiarizing oneself with the data, generating initial codes, searching for themes, reviewing themes, defining and naming themes, and finally, producing the report (Braun & Clarke, 2006). According to Nowell et. al the method's strength lies in its ability to handle data systematically and exhaustively, employing detailed recording and systematizing methods that enhance the credibility and trustworthiness of the analysis (Nowell, Norris, White, & Moules, 2017). This iterative process ensures a thorough examination of the data, enabling researchers to uncover patterns and themes that accurately represent the collected information, making thematic analysis an ideal choice for this thesis.

Braun and Clarke highlight thematic analysis as a method that is both accessible and theoretically flexible, making it a valuable tool for researchers across various disciplines (Braun & Clarke, 2006). The flexibility of thematic analysis gives possibility of exploring complex topics like generative AI by enabling capturing of varied perspectives on its influence, challenges, and potential solutions within low code automation development. The method excels in delving into the detailed narratives provided by interviewees, enabling a deep understanding of their experiences and opinions.

3.4 Limitations

The limited number of interviewees may not fully represent the diversity of experiences and opinions in the broader population of users engaging with low code automation tasks. While the insights collected from these participants are undoubtedly valuable, the narrow participant base may not encapsulate the full spectrum of experiences and opinions prevalent among the broader population engaged in low code automation tasks.

Moreover, the focus on tasks based on preview features, which are not widely available, introduces another layer of specificity to the study. While examining the cutting-edge aspects of technology offers an exciting glimpse into potential future developments, the findings' applicability to the broader, evolving landscape of low code automation is inherently limited. As these features are in their preview stages, they are subject to change based on user feedback and further development iterations.

4 Results

The section covers the strategic implementation of RPA within a case study organization, focusing on the integration of RPA into business processes for enhanced efficiency and innovation. It outlines the organizational structure designed for RPA deployment, emphasizing a strategic balance between central coordination and decentralized innovation to foster agility and governance in automation efforts. Additionally, it delves into specific task descriptions related to the integration of Generative AI features within the company's low code automation development tools. It covers the exploration of features such as workflow generation, expression generation, and code generation, designed to enhance the development process. Participants in this study were encouraged to engage with these features based on their relevance to their daily tasks, offering feedback through semi-structured interviews. Furthermore, the section presents a detailed analysis of the results from interviews.

4.1 RPA development at the case company

In the company’s business division, the development of RPA is carried out through a dynamic and agile methodology which incorporates employee involvement from the initial stage. This inclusive approach ensures that those who are familiar with the day-to-day operations contribute directly to the development process. The organizational structure for RPA development within the business unit consists of four key roles: the business users, who are empowered to take on the role of citizen RPA developers, thus fostering a sense of ownership and enabling grassroots innovation; a dedicated central RPA team who also known as platform integrator, which specializes in the technical construction and maintenance of RPA solutions; RPA managers, who oversee the strategic implementation and integration of RPA within business processes; and an RPA coordinator, who ensures cohesion and alignment across the different roles and business units, maintaining a clear vision and direction for RPA initiatives.

Integral to this setup are the RPA communities situated within the business unit. These communities serve as a collaborative ecosystem for innovation, composed of RPA managers, platform integrators and citizen developers. This approach decentralizes RPA development, providing a platform for grassroots innovation and ensuring that the solutions are finely tuned to the actual workflows they are intended to optimize. This democratized model of development not only encourages widespread adoption and enthusiasm for RPA across the company but also aligns with the goal of enhancing operational efficiency through direct employee engagement.

The RPA creation process begins with the identification of a suitable process for automation. This involves a business user recognizing a process that could significantly benefit from being automated, such as tasks that are repetitive, time-consuming, and prone to human error. Once a process is identified, a Minimum Viable Product (MVP) is implemented by citizen developer. It is the basic and functional version of the RPA that is designed to perform the necessary tasks with the least amount of development effort. The development of MVP is crucial as it allows the business user to test the effectiveness and efficiency of the RPA for the identified process. Their active participation in the initial creation of the RPA also ensures that the automation closely matches the real-world workflow, thus reducing the usual rounds of revisions and fine-tuning typically needed during the integration stage. In essence, this process democratizes RPA development and contributes to a more agile and responsive automation strategy. It aligns with lean principles by reducing waste and focusing on value-added activities, thereby accelerating the timeline from conception to deployment while ensuring that the automation is robust, relevant, and ready for scaling within the organization.

Once MVP is created, it is subjected to a thorough integration and quality assurance process. In this phase, the MVP is evaluated to confirm that it fulfils the necessary performance standards and adheres to the specified process. Any glitches or issues are identified and corrected during this stage. During this phase, the RPA is not only tested for its standalone functionality but also for its ability to integrate seamlessly with other

business systems and workflows. This integration testing is crucial for identifying any compatibility issues and ensuring that the bot interacts correctly with various interfaces and data sources.

After integration and QA phase, the RPA enters the UVT phase. In this stage, end-users interact with the bot within a controlled environment to validate its functionality from a user perspective. Feedback from this phase is crucial to refine the RPA and align it closely with user needs and business objectives. The UVT process plays a crucial role in ensuring that the RPA functions dependably and can be scaled up for full production.

Following UVT, the RPA is introduced to a live environment during the piloting phase. This final step involves deploying the RPA into the production environment, where it begins to automate the identified process end-to-end. Continuous monitoring during this phase ensures that the RPA operates as intended and delivers the expected benefits to the business. Any challenges encountered can be addressed with minimal impact on overall business operations.

This procedure, which spans from the initial recognition of a suitable task to the final implementation in production, guarantees a comprehensive evaluation and verification of the RPA. This enhances its dependability and effectiveness, while also minimizing the potential risks that come with the introduction of the solution.

4.2 Task Descriptions

In the design of the framework for the interviews, participants were presented with the currently available features of Generative AI to integrate into their daily activities as per their professional requirements. While a comprehensive range of features was made available—spanning workflow generation, expression creation, modification of existing workflows, and code generation—participants were not mandated to test all functionalities. Instead, they were granted the freedom to selectively engage with features they

deemed relevant to their needs. This approach allowed for a more organic assessment of each feature's applicability and usefulness, with participants providing feedback based on their experiences through semi-structured interviews, including interactions with support tools like Copilot or the company’s GPT platform.

Workflow generation is designed to enhance the ease and efficiency of creating automation workflows. It uses advanced Natural Language Processing (NLP) technologies, allowing the software to understand and interpret human language inputs. The core functionality of this feature is its ability to translate a textual process description into a structured automation workflow. The workflow generation feature is not just for users with limited technical knowledge; it also serves as a time-saving tool for experienced developers. It simplifies the initial stages of workflow creation, allowing developers to focus more on refinement and optimization. In essence, workflow generation represents a significant leap in making RPA more accessible and efficient. By bridging the gap between natural language and automated workflows, it democratizes access to automation technology and enhances productivity for a wide range of users.

The objective of the tasks under this category is assessing the feature’s ability to interpret and convert text descriptions into workflows. Users are asked to use this feature in their daily work for creating their MVP, modifying existing RPAs or familiarizing themselves with new features. This category is designed to cover various aspects of the workflow generation feature, from basic functionality to complex integrations, user experience, and practical applicability. This comprehensive approach will provide valuable insights into the feature's capabilities, areas for improvement and guidance for the usage in the future.

Expression generation is a feature, which is designed to transform natural language descriptions into expressions, enabling developers to concentrate more on solving the core problem instead of focusing on the syntax construction. It simplifies the development process where the emphasis is shifted from the technical details of code syntax to the

conceptual challenges of the task being solved. This not only streamlines workflow but also opens possibilities for greater innovation and efficiency in problem-solving, as developers can direct their efforts and creativity towards the bigger picture, rather than being overwhelmed by the complexities of programming languages.

Users are asked to use expression generation feature by describing expressions that are commonly encountered in their everyday work tasks. This could include specific financial calculations, data processing logic or data type conversion that they regularly deal with.

By focusing on expressions that users encounter in their professional environments, this task aims to provide insights into how well the feature integrates into and streamlines daily business operations.

The core objective of **code generation** task is to evaluate the effectiveness and reliability of the code generation feature for process automation by users. This involves assessing the feature's capability to accurately interpret and execute basic commands input by users for automating specific processes. The goal is to ensure that the generated code accurately performs the intended automation tasks without errors, thereby confirming the feature's suitability for simplifying and streamlining routine coding operations in process automation scenarios.

4.3 Functionality enhancement, challenges and limitation and mitigation of them

The exploration into the integration of Generative AI within low code automation development tools unveils a complex landscape of innovation, challenge, and adaptation. Through detailed thematic analysis of user experiences, three distinct yet interconnected themes have emerged:

- Functionality enhancement through Generative AI integration
- Challenges and limitations of Generative AI in low code automation
- Strategies for mitigating challenges and limitations

These themes provide insights into how Generative AI influences low code automation tools, the hurdles encountered by users, and potential approaches to overcome these obstacles. The following figure provides a comprehensive visualization of the main themes and their associated subthemes, as identified through the thematic analysis of interviews.

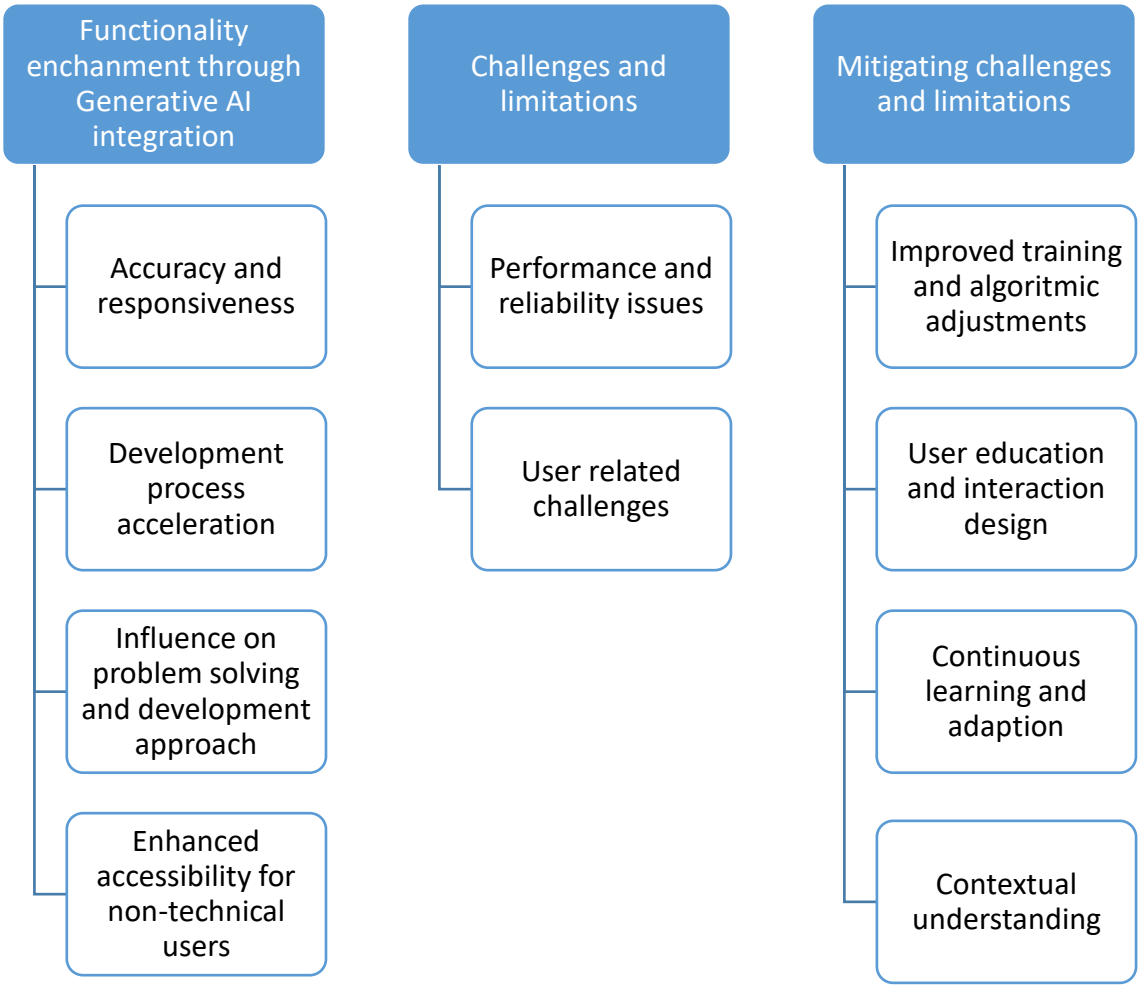

Figure 5 Identified themes from thematic analysis

Participants reported a range of experiences with the **functionality** of Generative AI features, highlighting both the potential for significant productivity gains and areas where expectations were not met. The feedback spanned a broad spectrum, from highly positive comments on efficiency improvements and transformative work practices enabled by Generative AI to critical observations pointing out limitations, inaccuracies, and usability challenges that detracted from the overall experience. These mixed reviews underscore the complicated process of incorporating Generative AI into everyday tasks related to low code automation development, pointing out the necessity for continuous improvement and adjustments to fully tap into its advantages while overcoming the flaws noted by users.

This section examines the results from interviews, focusing on the specific sub-themes identified regarding the functionality enhancement through Generative AI integration and usage. These themes, which have been identified through a thematic analysis of the interview data, encapsulate the direct experiences and observations of the participants.

The collective feedback from interviews regarding the **accuracy and responsiveness** of currently available features of Generative AI in understanding and executing requests offers a glimpse into the current state and potential of Generative AI integration and use cases. The responses vary significantly, reflecting a broad spectrum of experiences, from high satisfaction to notable challenges, underscoring the complexities of Generative AI's nature in practice.

During the interviews, participants shared a diverse range of experiences with the accuracy of currently available Generative AI features in recognizing and responding to their commands. A segment of the interviewees commended the features, highlighting their remarkable proficiency in interpreting and carrying out tasks. This was especially the case in basic and well-defined tasks where the Generative AI could apply its capabilities effectively. However, there have been exceptions even in executing well defined basic

tasks. A few participants also highlighted instances where the Generative AI features struggled with tasks that, on the surface, appeared straightforward.

> "When attempting to automate tasks in PowerPoint, the AI suggested using a browser activity, which was not relevant to the task. I tried to make my prompt clearer and wrote like open PowerPoint application explicitly and thought that it would use open application activity, but it still suggested using browser activity. The scenario which I wanted to automate was straight forward and simple, but it was not generating the correct output."

Such discrepancies often arose from nuances in the language used in prompts or the Generative AI's current limitations in understanding context deeply. Additionally, tasks involving the integration with specific software or services sometimes resulted in less than satisfactory outcomes, pointing out that while the Generative AI is proficient within certain bounds, its adaptability to a wide array of tools and environments is still a work in progress.

> "I mostly use UiPath for automating SAP related tasks. Generative AI features could not execute my requests related to SAP accurately. It worked okay with things like editing some string variables and opening web sites and clicking something there but not always perfectly. Sometimes I had to change some things manually."

A subset of users also reported encountering challenges when using the Generative AI for more complex automation tasks. These instances highlighted a perceived limitation in the Generative AI's ability to navigate tasks that lacked clear boundaries or involved higher levels of ambiguity.

These experiences underline the importance of ongoing refinement in Generative AI 's language processing and contextual understanding capabilities in low code automation development. Participants also highlight the need for users to maintain a level of skepticism and verification, particularly with tasks that, while seemingly simple, involve complex underlying processes or require precise outcomes. The feedback points toward a balanced approach in leveraging Generative AI features, embracing their strengths in

handling routine or well-defined tasks, while exercising caution and oversight for tasks that involve intricate details or critical outcomes.

Moreover, this variability in performance underscores the evolving nature of Generative AI technologies. As these tools learn and improve from user interactions and feedback, it's expected that their ability to navigate the nuances of human language and task complexity will enhance. This iterative development cycle, fueled by real world usage and feedback, is crucial for advancing Generative AI 's capabilities, making it a more reliable and versatile assistant in a broader spectrum of tasks. In conclusion, while Generative AI shows considerable promise, it's clear that its journey toward seamless integration and universal applicability is ongoing, marked by both impressive achievements and areas ripe for improvement.

The feedback suggests a dynamic interplay between user input and Generative AI interpretation, calling attention to the need for both sides to evolve, users to become more adept at issuing commands to Generative AI, and Generative AI systems to become more nuanced in understanding and executing a broader range of requests. This underscores an ongoing cycle of adaptation and learning that is essential for maximizing the potential of Generative AI tools in practical settings.

The collective insights gathered from interviews on the **acceleration of the low code automation development process** through Generative AI integration paint a comprehensive picture of its current impact and future potential. The feedback, encompassing a wide array of user experiences, underscores the varied impact of Generative AI on development process acceleration, highlighting both its capabilities and the challenges that come with its adoption.

Interviewees shared their experiences on how Generative AI has influenced the speed of their development projects. Many noted a marked improvement in the development process, attributing this acceleration to the AI's ability to rapidly generate code, and

assist with prototyping. This was particularly evident in tasks with well-defined and simple goals, where the Generative AI could leverage its strengths to effectively streamline processes.

> "Now, I can handle everything directly within the app's platform, including requesting details. I no longer have to use other websites; I can just input my queries there, and it automatically populates the code into the desired field. Additionally, I'm looking forward to integrating these autopilot features into the studio as well, so they can further assist with tasks seamlessly."

However, not all experiences were uniformly positive. Some participants pointed out instances where Generative AI's contributions did not reduce development time. Challenges were often attributed to Generative AI's limitations in understanding complex processes or its inability to integrate seamlessly with certain software and services.

> "There were times when the code generated by the AI or used prompt for generating task needed refinement. It is a helpful starting point but requires a discerning eye to ensure the final output aligns with the intended outcome."

These mixed experiences highlight the nuanced nature of Generative AI's impact on development process acceleration. While it offers considerable advantages in terms of speed and efficiency for certain tasks, its effectiveness is heavily dependent on the specificity of the task and the clarity of the instructions provided. Moreover, the feedback emphasizes the importance of ongoing development and refinement of Generative AI tools. As these technologies evolve, learning from user feedback and interactions, there is an anticipated improvement in their ability to understand complex requirements and function within diverse development ecosystems. This evolution is crucial for expanding the scope of tasks Generative AI can effectively accelerate.

> “When it achieves full functionality in the future, it would be a game changer. Now it just speeds things up a little bit. It will be interesting to see if and how quickly it will progress.”

The acceleration of the development process through Generative AI is a dynamic field marked by rapid advancements and learning curves on both the technological and user fronts. The interviews illustrate a landscape where Generative AI enhances development speed in many scenarios yet faces hurdles that require further innovation and adaptation. This ongoing interplay between user needs and capabilities of Generative AI underscores a development ecosystem that is increasingly adaptive, efficient, and poised for future advancements.

The impact of Generative AI on problem solving and the development approach is characterized by a blend of efficiency gains and new learning dynamics. It represents a shift towards more intelligent, AI-assisted development processes that promise productivity. Participants have shared a broad spectrum of experiences that highlight a transformative shift in problem solving and development approaches.

> "I think the problem solving is still there but, in another format. It is more problem-solving on, how to do the prompt correctly and to understand the code that AI generates."

Generative AI tools have been at the forefront of this shift, enhancing problem-solving efficiency. Participants have noted the ability of these tools to reduce the time spent on searching for code snippets or debugging solutions online, allowing them to focus more on solving higher-level problems. This increase in efficiency underscores the effectiveness of Generative AI in making the development process more streamlined. However, the possibility of delays in development and frustration related to incorrect outputs were also pointed out.

> "Generative AI can really speed things up by cutting down on the endless online search for solutions. For example, expression generation and verifying the expression that I have in mind is correct because it is very hard to remember those syntaxes by heart, especially if you use them rarely."

> "If you know how to structure your prompts you get quite good results."

> “It can also lead to delays if, for example generated code is incorrect, requiring the user to intervene and correct it.”

> “Sometimes, or even often, I think I could have done a better job or completed it faster if I had done it manually myself.”

The introduction of these Generative AI tools has also led to a notable shift in the development approach, moving towards a more prompt-driven methodology. This change underscores the importance of understanding how to communicate effectively with Generative AI, blending technical knowledge with a nuanced approach to prompt engineering. It is a skill that requires not only a deep understanding of the problem at hand but also an insight into the AI's processing and output generation mechanisms.

Generative AI features within low code automation tools **have enhanced accessibility for non-technical users**, democratizing the ability to contribute to low code automation development projects. This democratization of development represents a transformative shift, empowering a more diverse range of individuals to contribute their insights and expertise to the low code automation landscape. Participants also highlighted the transformative potential of generative AI features within low code automation tools, particularly for individuals with limited or non-technical backgrounds.

> “It can help people for with less technical background.”

> “I think it's more useful for those who do not have enough experience in low code automation development. So at least they get the scaffolding. When they try to describe the case that they would like to have automated, at least they get the scaffolding and then they can use that as a base.”

Overall, the interviews pointed out an optimistic picture of the future where Generative AI in low code automation platforms serves as a catalyst for broadening participation in low code automation development.

The journey of integrating Generative AI into the development process, while promising, is not without its hurdles. Participants' experiences shed light on a spectrum of **challenges and limitations** that underscore the complexities and nuances of employing Generative AI technologies effectively.

This section delves into the challenges and limitations encountered by users in utilization of Generative AI features in low code automation development. These challenges not only underscore the complexities but also highlight areas requiring attention for their improved adoption and effectiveness. The following sub-themes emerged from the analysis, painting a comprehensive picture of the challenges and limitations faced by participants.

This section presents a detailed examination of the **performance and reliability issues** associated with Generative AI, as derived from the collective feedback of interviewees. These insights offer a nuanced understanding of the challenges faced when integrating Generative AI into development processes, revealing a complex array of user experiences. The variability in performance and the reliability of Generative AI's features have emerged as concerns, demonstrating the technology's current limitations and the impact on user trust and workflow efficiency.

Participants highlighted varied experiences with the reliability of Generative AI, pointing out moments where it excelled in executing tasks with precision and efficiency. This was particularly evident in scenarios involving straightforward, well-defined tasks, where the technology could leverage its strengths to deliver quick and accurate results. However, alongside these successes, there were notable instances where Generative AI fell short of expectations. Challenges arose not only in complex or ambiguous tasks but surprisingly, also in tasks that appeared simple and well-defined at first glance.

Contextual understanding remains a significant hurdle, as evidenced by the experiences of several interviewees. Increasing the context awareness of Generative AI systems is

important to ensure that they can adapt to the nuances of various user requirements and operational environments. This improvement requires the integration of more sophisticated understanding and processing capabilities into Generative AI models, allowing them to perceive and react more skillfully to the nuances of various tasks and domains.

The necessity for clear and effective communication with Generative AI consistently emerges as a critical theme, emphasizing the importance of detailed and precise prompts for efficient AI assistance. This need for clarity and specificity in communication reflects the user's requirement to understand the underlying processes, highlighting the interplay between technical knowledge and the effective use of Generative AI features.

> "We are still in the early stages, and these technologies are quite new. Each of us faces challenges with providing the correct prompts. If the command given is even slightly incorrect or misleading, obtaining the right answer becomes more difficult. This is one of the aspects that comes to mind. In a way, it is about learning the best method to communicate effectively."

Challenges extend to the realm of data currency, with observations that Generative AI does not always operate on the most current data. This limitation is notably problematic when addressing new web activities or modern problems that have yet to be widely addressed, highlighting a significant challenge in ensuring AI systems are equipped with the latest information to provide relevant solutions.

> "It might not always be up to date with the newest updates, and the second is that you still need to know the code that you're like handling."

These reflections underscore the critical need for ongoing advancements in Generative AI's understanding and execution capabilities. The experiences shared by users emphasize the importance of a cautious and skeptical approach when leveraging AI for tasks that involve complex processes or require exacting outcomes. While Generative AI holds considerable promise for enhancing development processes, its journey towards achieving seamless integration and widespread applicability is evidently a work in progress.

> "The challenges and limitations for end users will involve understanding how the tool behaves. This includes knowing how to write questions correctly and how to interpret the output more effectively. You will receive an output that matches your needs, but you cannot always copy and paste that output, nor can you rely on it 100%. One of the biggest challenges will be understanding this aspect."

This underscores the evolving relationship between user commands and Generative AI interpretation, highlighting the necessity for both the technology and its users to adapt and learn. This iterative process of feedback and improvement is essential for advancing Generative AI's capabilities.

This section delves into the **user related challenges** encountered in the integration and utilization of Generative AI within development processes, as illuminated by insights from interviews. The challenges identified span a broad spectrum, reflecting a range of issues from the initial learning curve to the complexities of effectively communicating with Generative AI.

Users expressed a variety of difficulties in interfacing with Generative AI, indicating that while the technology promises to streamline and enhance development workflows, the reality of its application sometimes falls short of expectations. A significant portion of the feedback pointed to the complexities of formulating prompts or commands that Generative AI could interpret accurately and execute efficiently. The effectiveness of Generative AI features is deeply influenced by the clarity and detail of user prompts. The challenge lies not only in the Generative AI's technological capabilities but also in the users' ability to articulate their needs precisely.

> "It is not just about what the Generative AI can do; it is also about how well users can communicate with it. It takes time to learn how to ask the right questions in the right way; there is a learning curve involved."

Furthermore, the interviews highlighted an expectation-reality gap, where users anticipated a level of intuitiveness and adaptability from Generative AI that it could not consistently deliver. Many users approached Generative AI with high expectations for its potential to revolutionize development processes. This discrepancy often led to misaligned expectations, where Generative AI's actual capabilities fell short of what users believed it could do, affecting their overall satisfaction and trust in the technology. These insights highlight the critical need for improvement in Generative AI's capabilities in general and support for users to navigate the complexities of working with Generative AI.

The user related challenges in Generative AI integration and utilization reflect a dynamic interplay between human and machine, marked by a continuous process of negotiation, learning, and adjustment. Addressing these challenges is essential for realizing the full promise of Generative AI in development processes, necessitating a concerted effort to bridge the gap between technological capability and user experience.

Mitigating the challenges and limitations of Generative AI in low code automation development requires a multifaceted approach that includes technological advancements, educational initiatives, policy development, and community support. By addressing these areas, organizations can unlock the transformative potential of Generative AI, leading to increased efficiency, accessibility, and innovation in development processes. As the field of AI evolves, the strategies for overcoming these challenges will also advance, paving the way for a future where AI and human creativity collaborate to solve complex problems and create innovative solutions.

This section explores the strategic approaches and best practices designed to overcome the hurdles faced by organizations and developers when incorporating Generative AI technologies into low code automation platforms. This section will provide insights into enhancing the symbiosis between Generative AI driven features and developers, ensuring the successful adoption and maximization of Generative AI's potential in low code automation projects.

A dynamic feedback loop between developers and AI systems can significantly enhance collaboration. By providing AI with feedback on its performance, developers can help refine AI models, making them more aligned with user needs and expectations. Similarly, developers can learn from interactions with AI, gaining insights into more effective problem-solving strategies and automation techniques.

A central theme emerging from the feedback is the critical role of advanced training techniques in refining the performance of Generative AI systems. Participants highlighted the significance of comprehensive and diverse datasets in training AI models, ensuring a broader understanding and responsiveness to a wide array of user commands and scenarios.

> "AI needs to learn more from user interactions and that over time, with more recorded history, it might better understand issues."

> “The limitations due to input can potentially be mitigated. For instance, once a user has begun a project, the AI could analyze the work done so far to grasp the general concept of the user's goal. This understanding could then guide the AI in making more accurate predictions with the prompts provided. If I've already started a project and am midway through, asking the AI to generate code, a workflow, or an additional expression, it could review my progress or previous work to infer what I'm aiming to accomplish. This approach becomes even more nuanced as the AI recognizes individual behavioral patterns over time. For example, each person has their unique coding style. The more the AI is used, the better it becomes at identifying these patterns, allowing it to make suggestions that are increasingly tailored to the user's method of coding, enhancing the overall experience.”

These strategies for mitigating challenges and limitations through **improved training and algorithmic adjustments** highlight a dynamic and evolving approach to enhancing Generative AI integration. The ongoing efforts to refine Generative AI through improved training and algorithmic adjustments are critical in addressing the nuanced challenges faced by users. This approach underscores the importance of a continuous, iterative

process of development and feedback, aiming to bridge the gap between the potential of Generative AI and its practical application in development processes. The proactive pursuit of these strategies promises to enhance the reliability, accuracy, and user-friendliness of Generative AI technologies, paving the way for more seamless and effective integration into development workflows.

This section addresses the strategies for mitigating challenges and limitations associated with the use of Generative AI, with a particular emphasis on **user education and improvements in interaction design**. Insights derived from interviews underscore the importance of these areas in enhancing the overall effectiveness and user satisfaction with Generative AI technologies. By focusing on educating users and refining how they interact with AI systems, developers and organizations can take significant steps toward bridging the gap between the current state of AI capabilities and the expectations of end users.

Participants highlighted the role of user education in optimizing outcomes when utilizing Generative AI in low code automation development. A consistent theme was the need for enhancing users' proficiency in interacting with Generative AI tools and features. Better outcomes are closely tied to the users' ability to effectively communicate their requirements to the Generative AI, which hinges on a comprehensive understanding of how these tools interpret and process human input.

Educational initiatives that focus on teaching users about the structure and design of prompts can lead to more accurate and efficient AI generated results. Moreover, a deeper awareness of the AI's capabilities and limitations empowers users to set realistic expectations of the AI solutions in real world scenarios.

> "After some time, I think you will learn how to write prompts Generative AI understands. Maybe there could also be some instructions on how the prompt could look like."

> “I guess we need to understand its intended use first, and then, of course, figure out how to utilize it most efficiently.”

In parallel, advancements in interaction design play a critical role in simplifying and optimizing the user's engagement with Generative AI. Feedback from users has driven the development of more intuitive interfaces and interaction models, reducing the complexity and ambiguity in communicating with AI systems. Efforts to design interfaces that more accurately capture user intent and provide clear, contextual guidance are key to improving the overall user experience.

> "Refining the AI's user interface to include more guided input methods and contextual help options can help when writing correct prompts. It can reduce the errors and misunderstandings."

These strategies for enhancing user education and refining interaction design are important in addressing the challenges encountered by participants. By prioritizing the user's experience and providing the tools and knowledge necessary for effective interaction, organizations can significantly improve the usability and satisfaction associated with Generative AI technologies.

This section explores the capability of Generative AI tools to evolve and enhance their effectiveness through **learning from interactions and adapting continuously**. This adaptive learning process is important for mitigating the challenges and limitations currently faced by users of Generative AI technologies. By drawing insights from user interactions and continuously updating algorithms, Generative AI systems can improve accuracy, reliability, and overall utility in development environments. The discussion here is anchored in the feedback from interviews, emphasizing the dynamic nature of Generative AI and its potential to transform development practices through ongoing evolution.

All participants consistently emphasized the necessity for Generative AI to deepen its learning from user interactions, highlighting the expectation that, over time and with an expanding record of interactions, AI will progressively enhance its understanding of user

needs and challenges. This consensus underlines a shared belief in the potential of Generative AI to evolve into a more intuitive and responsive tool, one that can adapt its functionalities more effectively to address specific issues encountered by users in their development workflows.

This expectation sets a direction for the ongoing development of Generative AI systems, emphasizing the need for advanced learning mechanisms that can capture and analyze user interactions in detail. The anticipation is that with a richer dataset of user engagements, AI tools will not only revise current limitations but also anticipate and navigate complexities more adeptly.

Moreover, this perspective reinforces the call for Generative AI platforms to prioritize adaptive learning algorithms that are capable of evolving based on a continuous feedback loop. Such an approach is envisioned to bridge the gap between the initial capabilities of AI tools and the dynamic needs of their human counterparts.

In addressing the challenges and limitations faced by users of Generative AI, a significant emphasis has been placed on the importance of enhancing the **contextual understanding** of these systems. This focus on contextual awareness is important for improving the accuracy, relevance, and overall effectiveness of Generative AI in diverse development environments. Drawing from insights gathered through interviews, this section highlights the collective acknowledgment among participants that deepening Generative AI's grasp of context is crucial for overcoming current obstacles and unlocking the full potential of Generative AI technologies in development processes.

Participants uniformly expressed the view that while Generative AI has shown promise in interpreting and executing tasks, its performance is often hampered by a lack of deep contextual understanding. The nuances of specific tasks, especially those involving complex or less common processes, are often lost in Generative AI systems. This limitation

not only affects the quality of outcomes but also impacts user trust and reliance on AI for complex tasks.

Improving the contextual understanding of Generative AI is identified as a critical step toward mitigating the challenges and limitations experienced by users. Through enhanced contextual awareness, Generative AI can achieve a more nuanced understanding of user needs, leading to more accurate, relevant, and trustworthy AI-assisted development processes.

5 Conclusion and discussion

The conclusion and discussion section aims to contextualize the findings from the thematic analysis within the broader literature on Generative AI and low code automation. This section will interpret the results, compare them with existing studies, and explore the implications of the findings. Furthermore, it will reflect on the study's limitations, reliability and validity and practical implications of the results.

5.1 Conclusion

Investigating the incorporation of Generative AI into low code automation development tools has revealed a transformative shift in the landscape of low code automation development. This thesis aims to understand the multifaceted impact of Generative AI on low code automation development platforms, the challenges and limitations of its use, and the strategies to mitigate these hurdles. Through a detailed analysis of literature and empirical data through semi structured interviews, thesis shed lights on the interplay between technological advancement and user engagement in the realm of low code automation.

The integration of Generative AI into low code automation tools stands at the forefront of a new wave of technological innovation in low code automation development. This study's findings highlight both the potential of this integration and the challenges that must be overcome. Through detailed thematic analysis of user experiences, three closely themes have been identified; enhancing functionality via the integration of Generative AI, the challenges and limitations associated with Generative AI in low code automation and approaches to address and mitigate these challenges and limitations. This analysis illuminates the ways in which Generative AI influences low code automation tools, the obstacles encountered by users, and the potential strategies to overcome these challenges.

As an answer to the first research question of how the integration of Generative AI influences the functionality of low code automation development tools, it was found that the integration of Generative AI with low code platforms demonstrates a transformative shift, simplifying the development process and expanding the range of individuals who can participate in low code automation development. This aligns with the predictions by Ray et al. and Juhas et al., who foresee low code as the driving force behind most of the application development by 2025 (Juhas et al., 2022; Ray et al., 2023). This trend underscores the growing significance of low code and Generative AI in shaping the future of low code automation development, emphasizing the shift towards more accessible, efficient, and inclusive development practices.

The investigation has underscored the significant potential of Generative AI to enhance the functionality of low code automation tools. The integration of Generative AI technologies, as highlighted in our findings, promises to revolutionize the development process by accelerating task completion, enhancing problem-solving capabilities, and broadening the accessibility of development tools to a wider audience, including those with limited or non-technical expertise. These advancements are crucial in addressing the growing low code automation development needs and the democratization of development, resonating with the predictions and observations of Juhas et al. (Juhas, et al., 2022), Peng et al. (Peng, Kalliamvakou, Cihon, & Demirer, 2023), and Dohmke, Iansiti, & Richards (Dohmke, Iansiti, & Richards, 2023).

Despite the promising growth of the RPA market and the strategic adoption of Generative AI, there are noted impediments to achieving automation objectives, particularly in front-office operations (Shanbhag & Sivalingam, 2023). This observation is in line with the challenges mentioned by interviewees regarding the limitations and integration issues of Generative AI features in complex development tasks. However, the expected integration of Generative AI-assisted automation by a vast majority of RPA vendors by

2025 (Ray, et al., 2023) signifies a shift towards more intelligent and adaptable automation solutions that could address these challenges.

The active engagement of RPA vendors with LLMs, including the integration of Generative AI functionalities into RPA products, marks a significant evolution in the automation landscape (Ray, et al., 2023). This development is expected to enhance the capabilities of RPA tools, offering more sophisticated features such as AI-powered script development and intelligent document processing. This aligns with the experiences and expectations shared by the interviewees, who envision a future where Generative AI features play a central role in their development projects, pointing towards a convergence of human intelligence and AI capabilities to streamline development workflows and enhance operational efficiency.

The majority of interviewees highlighted the positive impact of Generative AI features on their productivity. This aligns with the findings of Peng et al. and Dohmke et al., who demonstrated significant reductions in task completion times and increased code suggestion acceptances with GitHub Copilot (Dohmke, Iansiti, & Richards, 2023; Peng, Kalliamvakou, Cihon, & Demirer, 2023). Interviewees shared experiences that mirror these studies. However, it was also highlighted that the effectiveness of these tools varied depending on the clarity of user requests and the complexity of the tasks, pointing out that while Generative AI tools and features enhance productivity, their efficiency is highly contingent on user input and task specificity.

Despite the noted benefits, challenges such as learning curves, the accuracy of Generative AI outputs, reliability and integration issues were also mentioned in the interviews. These findings echo concerns in the literature about the potential for increased frustration among users (Choudhuri, Liu, Steinmacher, Gerosa, & Sarma, 2023) and highlight the importance of careful integration and usage of Generative AI tools and features. The feedback underscores the need for developers to possess a robust understanding of the code generated by Generative AI to effectively utilize these tools, reinforcing the

criticality of human oversight in the human-AI collaboration in low code automation development.

As an answer to the second research question related to the potential challenges and limitations in using Generative AI in low code automation development, we can conclude that the journey towards fully harnessing the benefits of Generative AI in low code automation is not without its challenges. The research has revealed significant obstacles, including issues related to the accuracy and reliability of Generative AI outputs, the steep learning curve for effectively leveraging these technologies, and the need for a deeper contextual understanding by Generative AI systems. These challenges highlight the gap between the potential and current capabilities of Generative AI, necessitating a concerted effort to refine and enhance these technologies for practical application.

The necessity for Generative AI tools to adapt and learn from user interactions was a recurring theme in the interviews. Participants noted that while Generative AI shows promise, its current limitations necessitate an ongoing cycle of adaptation and learning. This reflects the insights from the literature on the importance of dynamic feedback loops between users and Generative AI systems (Jiao, et al., 2022) By incorporating user feedback into the development and refinement of Generative AI models, it's possible to enhance their understanding and responsiveness, leading to more accurate and effective tool performance. The findings also underscore the critical role of context in effectively utilizing Generative AI, pointing to the need for Generative AI systems to progress from keyword recognition to a deeper understanding of user intent and task complexities. This evolution requires not only technological advancements but also a concerted effort in user education and interaction design to bridge the gap between Generative AI potential and practical application.

Generative AI’s role in democratizing low code automation development extends beyond simplifying automation processes. It encompasses the broader vision of empowering non-technical users to contribute meaningfully to low code automation development

projects. This broader vision aligns with insights from interviews indicating that Generative AI's capabilities are particularly beneficial for newcomers and those without technical backgrounds. This vision also resonates with the insights from Rao et al and Brüggemann, who highlights the inclusive approach of low code platforms in bridging the gap between non-programmers and IT professionals. By integrating Generative AI into these platforms, the barrier to entry for low code automation development is further lowered, enabling individuals with diverse backgrounds and skill sets to participate in the creation and refinement of AI-driven low code automation projects (Brüggemann, 2023; Rao, Tsay, Kiran, Hellendoorn, & Hirzel, 2023).

The thematic analysis of the interview transcripts reveals an optimistic view of Generative AI's role in low code automation development. While the potential for increased efficiency and accessibility is widely recognized, challenges related to technical limitations, the need for user education, and the importance of clear communication with AI are also evident. All participants expressed intentions to adopt Generative AI tools and functionalities in their future projects, particularly in anticipation of forthcoming enhancements. This unified inclination towards the future utilization of Generative AI, after its improvements, underlines the importance of advancing Generative AI’s capabilities. These steps are crucial for fully leveraging the advantages of Generative AI features in low code automation and expanding its reach.

Addressing the challenges and limitations identified necessitates a comprehensive approach that spans technological advancements, user education, and the design of intuitive user-Generative AI interaction frameworks. Enhancing the collaborative synergy between users and Generative AI systems through improved training, algorithmic adjustments, and continuous learning and adaptation is crucial. These efforts can refine AI models more accurately align with user expectations and facilitate a deeper understanding of complex problem-solving strategies.

As an answer to the third research question related to mitigating the mentioned challenges and limitations in utilizing Generative AI in low code automation development, the thesis proposes several strategic approaches. These include enhancing the contextual understanding and accuracy of Generative AI through continuous learning from diverse data sources and user interactions, which can improve the system's ability to interpret and execute user commands more effectively. Additionally, the thesis emphasizes the importance of user education to navigate the learning curve associated with Generative AI tools and features. Improving interaction design to facilitate clearer communication between users and Generative AI is also highlighted, aiming to reduce misunderstandings and increase the efficiency of Generative AI-assisted development. Furthermore, the study suggests fostering a culture of collaboration and feedback between users and Generative AI developers to continually refine and adapt Generative AI functionalities to user needs. Addressing these strategies is critical for overcoming the obstacles presented by Generative AI integration.

As we look to the future, the integration of Generative AI in low code platforms presents a landscape ripe with opportunities for innovation, inclusivity, and efficiency in low code automation development. The path forward requires a collaborative effort among researchers, developers, and end-users to navigate the challenges and harness the full potential of Generative AI technologies. Continuous engagement with the evolving capabilities of Generative AI, coupled with a commitment to user empowerment and education, will be key to unlocking the transformative impact of Generative AI in democratizing low code automation development and fostering a more inclusive digital future.

This research underscores the importance of continuous improvement in Generative AI technologies, user education, and the development of more intuitive interaction designs. As Generative AI continues to evolve, it holds the promise of revolutionizing low code automation by making it more accessible, efficient, and aligned with the dynamic needs of users. The future of low code development, powered by Generative AI, is not without its challenges, but with concerted efforts in technological advancements and user

support, it offers a landscape with possibilities for innovation and inclusivity in low code automation development.

5.2 Discussion

The evolving landscape of Generative AI and low code automation presents an ongoing opportunity for innovation and improvement. As these technologies continue to advance, the collective understanding and methodologies for leveraging them effectively will also progress. Through this iterative process of exploration, challenge, and adaptation that the full potential of Generative AI in transforming low code automation development can be unlocked. This thesis contributes to the current discussion on Generative AI and low code automation, delivering insights and strategies that will assist in clearing the path toward harnessing the complete capabilities of this intersection.

Through thematic analysis, the study systematically identifies, analyzes, and reports themes within the data, adding depth to the understanding of how Generative AI can be effectively utilized in low code automation development. This method not only strengthens the validity and reliability of the research outcomes but also deepens the study's narrative by offering comprehensive insights. This qualitative approach enhances the research's insights, making them contribute to the ongoing dialogue on leveraging Generative AI for enhancing low code automation development.

The findings of the study underscore the synergy between low code platforms and Generative AI, which enhances organizational agility and competitiveness in the digital economy. This synergy is particularly evident in the automation of business processes, where low code tools, augmented by Generative AI, enable rapid and cost-effective development of automations. Furthermore, the integration of Generative AI has facilitated a more inclusive approach to low code automation development, empowering a broader range of employees to contribute to digital solutions.

The integration of Generative AI within low code platforms signifies a major leap forward in democratizing low code automation development. By simplifying complex automation tasks and making development accessible to a broader audience, Generative AI has the potential to revolutionize the low code automation landscape. This shift aligns with the growing demand for rapid application development and the need to empower individuals with varying levels of technical expertise to contribute to innovation. The study's findings highlight the efficiency gains, enhanced problem-solving capabilities, and the expanded accessibility provided by Generative AI, corroborating the vision of a future where development is more democratized and inclusive. These findings support the predicted future where development practices become more democratized and inclusive, suggesting a paradigm shift in how technologies are leveraged. By demonstrating the benefits of Generative AI in streamlining processes, enhancing creative problem-solving, and making advanced tools more accessible, the research reinforces the potential for a more equitable and diverse technological landscape.

Furthermore, the study identified critical hurdles, including issues with the accuracy and reliability of Generative AI's outputs, the steep learning curve for effectively leveraging tools and features, and the need for enhanced contextual understanding by Generative AI. Addressing these challenges is important to unlocking the full potential of Generative AI in low code automation development. The strategies proposed, focusing on continuous technological advancements, user education, and the development of intuitive user-Generative AI interaction models, lay the groundwork for overcoming these obstacles.

With the evolving capabilities of Generative AI, there is a clear need for ongoing training programs for both technical and non-technical employees. Organizations are highly suggested to invest in upskilling their workforce to navigate the new landscape effectively, focusing on understanding how to best interact with Generative AI for optimal outcomes. This includes developing skills in prompt engineering, understanding the limitations of Generative AI, and integrating human oversight into the development process.

Bridging these training initiatives with the practical application of Generative AI in low-code automation, the approach significantly lowers the barrier to entry for non-technical users to engage in development. By extending training to a wider segment of the workforce, organizations not only democratize low code automation development but also foster a culture of innovation across all levels. Such inclusivity paves the way for the rise of citizen developers, enabling them to transform their ideas into functional prototypes. Furthermore, creating a collaborative ecosystem that includes both citizen developers and experts, focused on harnessing Generative AI within low code platforms, catalyzes the pace of innovation and facilitates the exchange of best practices. Engaging with this vibrant community through forums, workshops, and collaborative projects can drive the forward momentum of Generative AI technologies.

As a result of this thesis work, some areas of further research were found. As we look to the future, the integration of Generative AI within low code automation development tools holds the promise of a more democratized approach to low code automation development. The potential to empower a broader range of individuals to contribute to the development process is an exciting prospect. However, realizing this vision requires continued research, development, and collaboration between technologists, users, and educators.

While this study provides valuable insights into the integration of Generative AI in low-code automation, it is not without limitations. The thematic analysis was based on a select number of interviews, which may not fully capture the diversity of experiences and perspectives in the wider developer community. Additionally, the rapid pace of technological advancements in Generative AI means that the findings may quickly evolve as new Generative AI capabilities emerge. Future research should aim to address these limitations by incorporating a broader and more diverse sample of users, including those from different industries and with varying levels of expertise in low code automation development. The studies could provide deeper insights into how the integration of Generative AI evolves over time and its long-term impacts on development practices. Further

comparative studies between different Generative AI platforms could shed light on specific features or approaches that enhance or hinder effective integration into low code environments.

References

Shanbhag, P.;& Sivalingam, J. (2023). *Closing the AI and Automation Skill Gap in Modern Enterprises with Outcome-Focused Training and Certification.* IDC. Noudettu osoitteesta https://start.uipath.com/rs/995-XLT-886/images/IDC_Closing_the_AI_and_Automation_Skill_Gap_with_Training_and_Certification.pdf

Bhattacharyya, S. S.;& 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*. doi:https://doi.org/10.1108/JSTPM-06-2021-0084

Boussaa, M.;Barais, O.;Sunyé, G.;& Baudry, B. (2020). Leveraging metamorphic testing to automatically detect inconsistencies in code generator families. doi:https://doi.org/10.1002/stvr.1721

Braun, V.;& Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative research in psychology*. doi:https://doi.org/10.1191/1478088706qp063oa

Brüggemann, L. (2023). Securely extending and running low-code applications with C#. doi:https://doi.org/10.48550/arxiv.2307.06340

Brynjolfsson, E. L. (2023). Generative AI at Work. doi:10.3386/w31161

Choudhuri, R.;Liu, D.;Steinmacher, I.;Gerosa, M.;& Sarma, A. (2023). How Far Are We? The Triumphs and Trials of Generative AI in Learning Software Engineering. doi:https://doi.org/10.48550/arxiv.2312.11719

Denny, E.;& Weckesser, A. (2022). How to do qualitative research?: Qualitative research methods. doi:https://doi.org/10.1111/1471-0528.17150

Dohmke, T.;Iansiti, M.;& Richards, G. (2023). Sea Change in Software Development: Economic and Productivity Analysis of the AI-Powered Developer Lifecycle. doi:https://doi.org/10.48550/arxiv.2306.15033

Eulerich, M.;Waddoups, N.;Wagener, M.;& Wood, D. A. (2023). The Dark Side of Robotic Process Automation (RPA): Understanding Risks and Challenges with RPA. doi:https://doi.org/10.2308/HORIZONS-2022-019

Feuerriegel, S.;Hartmann, J.;Janiesch, C.;& Zschech, P. (12. September 2023). Generative AI. Business & Information Systems Engineering. doi:https://doi.org/10.1007/s12599-023-00834-7

Gartner. (16. October 2023). Gartner Identifies the Top 10 Strategic Technology Trends for 2024. Noudettu osoitteesta https://www.gartner.com/en/newsroom/press-releases/2023-10-16-gartner-identifies-the-top-10-strategic-technology-trends-for-2024

Gozalo-Brizuela, R.;& Garrido-Merchan, E. C. (2023). ChatGPT is not all you need. A State of the Art Review of large Generative AI models. doi:10.48550/arxiv.2301.04655

Grande, V. A., Campos, R. d., Facin, A. L., & Batistela, G. C. (2022). An analysis of the benefits, challenges and methods of process selection to adopt robotic process automation. *GEPROS : Gestão da Produção, Operações e Sistemas*. doi:https://doi.org/10.15675/gepros.v17i3.2934

Groenland, E.;& Dana, L.-P. (2019). *Qualitative Methodologies And Data Collection Methods: Toward Increased Rigour In Management Research.*

Hemmer, Patrick ; Westphal, Monika ; Schemmer, Max ; Vetter, Sebastian ; Vössing, Michael ; Satzger, Gerhard. (2023). Human-AI Collaboration: The Effect of AI Delegation on Human Task Performance and Task Satisfaction. doi:https://doi.org/10.48550/arxiv.2303.09224

Hennink, M.;Hutter, I.;& Bailey, A. (2020). *Qualitative research methods.* Sage Publications.

IBM. (n.d.). *RPA and Intelligent Automation*. Noudettu osoitteesta https://www.ibm.com/topics/rpa

Janiesch, C.;Zschech, P.;& Heinrich, K. (2021). Machine learning and deep learning. doi:https://doi.org/10.1007/s12525-021-00475-2

Jiao, S.;Liao, Q. V.;Muller, M.;Agarwal, M.;Houde, S.;Talamadupula, K.;& Weisz, J. D. (2022). Investigating Explainability of Generative AI for Code through Scenario-based Design. doi:https://doi.org/10.48550/arxiv.2202.04903

Juhas, G.;Molnar, L.;Juhasova, A.;Ondrisova, M.;Mladoniczky, M.;& Kovacik, T. (2022). Low-code platforms and languages: the future of software development. doi:https://doi.org/10.1109/ICETA57911.2022.9974697

Kai, L.;Wenxin, L.;Ran, Z.;Suihua, K.;Dazhu, Y.;Min, Z.;& Wenhong, T. (2022). Research on "AI+RPA" Interactive Technology of Intelligent Financial Management Platform. doi:https://doi.org/10.1109/ICCWAMTIP56608.2022.10016525

Kalota, F. (2024). A Primer on Generative Artificial Intelligence. *Education Sciences, 14*(2). doi:https://doi.org/10.3390/educsci14020172

Kedziora, D.;& Hyrynsalmi, S. (2023). Turning Robotic Process Automation onto Intelligent Automation with Machine Learning. doi:https://doi.org/10.1145/3593743.3593746

Kirchhof, J. C.;Jansen, N.;Rumpe, B.;& Wortmann, A. (2023). Navigating the Low-Code Landscape: A Comparison of Development Platforms. doi: https://doi.org/10.1109/MODELS-C59198.2023.00135

KPMG. (2023). *KPMG Generative AI Survey.*

Lochmiller, C. (2021). Conducting Thematic Analysis with Qualitative Data. *Qualitative report*. doi:https://doi.org/10.46743/2160-3715/2021.5008

Loomis, A. (7. December 2022). *IDC FutureScape: Worldwide Future of Work 2023 Predictions*. Noudettu osoitteesta https://blogs.idc.com/2022/12/07/idc-futurescape-worldwide-future-of-work-2023-predictions/

Martinez, E.;& Pfister, L. (2023). Benefits and limitations of using low-code development to support digitalization in the construction industry. *Automation in Construction, 152*, 104909. doi:https://doi.org/10.1016/j.autcon.2023.104909

Martínez, G.;Watson, L.;Reviriego, P.;Hernández, J. A.;Juarez, M.;& Sarkar, R. (2023). Combining Generative Artificial Intelligence (AI) and the Internet: Heading towards Evolution or Degradation? doi:https://doi.org/10.48550/arxiv.2303.01255

Moreira, S.;Mamede, H. S.;& Santos, A. (2023). Process automation using RPA – a literature review. *Procedia Computer Science, 219*, ss. 244-254. doi:https://doi.org/10.1016/j.procs.2023.01.287

Mozo, A.;Karamchandani, A.;de la Cal, L.;Gómez-Canaval, S.;Pastor, A.;& Gifre, L. (2023). A Machine-Learning-Based Cyberattack Detector for a Cloud-Based SDN Controller. doi:https://doi.org/10.3390/app13084914

Nowell, L. S.;Norris, J. M.;White, D. E.;& Moules, N. J. (2017). Thematic Analysis: Striving to Meet the Trustworthiness Criteria. *International journal of qualitative methods*. doi:https://doi.org/10.1177/1609406917733847

Orrù, G.;Piarulli, A.;Conversano, C.;& Gemignani, A. (05 2023). Human-like problem-solving abilities in large language models using ChatGPT. *6*, ss. 1199350-1199350. doi:1199350-1199350

Peng, S.;Kalliamvakou, E.;Cihon, P.;& Demirer, M. (2023). The Impact of AI on Developer Productivity: Evidence from GitHub Copilot. doi:https://doi.org/10.48550/arxiv.2302.06590

Perdana, A.;Lee, W. E.;& Kim, C. M. (2023). Prototyping and Implementing Robotic Process Automation in Accounting Firms: Benefits, Challenges, and Opportunities to Audit Automation. *International journal of accounting information systems*. doi:https://doi.org/10.1016/j.accinf.2023.100641

Pezzini, M.;& Khiyara, S. (2023). Democratizing automation with citizen developers: navigating the pitfalls and opportunities. Noudettu osoitteesta https://www.proquest.com/trade-journals/democratizing-automation-with-citizen-developers/docview/2808489787/se-2?accountid=14797

Picek, R. (2023). Low-code/No-code Platforms and Modern ERP Systems. doi:https://doi.org/10.1109/ICIM58774.2023.00014

Prabodha, S.;& Liyanage, H. (2023). Enhancing the Existing Benefits of Robotic Process Automation in the Logistics Industry. doi:https://doi.org/10.1109/MERCon60487.2023.10355398

Pramod, D. (2022). Robotic process automation for industry: adoption status, benefits, challenges and research agenda. *Emerald Journals; Scopus*. doi:https://doi.org/10.1108/BIJ-01-2021-0033

Raiaan, M. A.;Mukta, M. S.;Fatema, K.;Fahad, N. M.;Sakib, S.;Mim, M. M.;. . . Azam, S. (2024). A Review on Large Language Models: Architectures, Applications,

Taxonomies, Open Issues and Challenges. *12*, ss. 26839-26874. doi:https://doi.org/10.1109/ACCESS.2024.3365742

Rao, N.;Tsay, J.;Kiran, K.;Hellendoorn, V. J.;& Hirzel, M. (2023). AI for Low-Code for AI. doi:https://doi.org/10.48550/arxiv.2305.20015

Ray, S.;Villa, A.;Alexander, M.;Wang, A.;Saha, M.;& Joshi, S. (2023). *Magic Quadrant for Robotic Process Automation.* Gartner. Noudettu osoitteesta https://www.gartner.com/doc/reprints?id=1-2EDXTGOY&ct=230705&st=sb

Rokis, Karlis ; Kirikova, Marite. (10 2023). Exploring Low-Code Development: A Comprehensive Literature Review. *Complex systems informatics and modeling quarterly, 2023 (36)*, 68-86. doi:https://doi.org/10.7250/csimq.2023-36.04

Russell, S. J.;& Norvig, P. (2021). *Artificial intelligence : A Modern Approach, Global Edition.*

Shamshirband, S.;Fathi, M.;Dehzangi, A.;Chronopoulos, A. T.;& Alinejad-Rokny, H. (2021). A review on deep learning approaches in healthcare systems: Taxonomies, challenges, and open issues. doi:https://doi.org/10.1016/j.jbi.2020.103627

Sharma, Deepak Kumar ; Mishra, Jahanavi ; Singh, Aeshit ; Govil, Raghav ; Srivastava, Gautam ; Lin, Jerry Chun-Wei. (2022). Explainable Artificial Intelligence for Cybersecurity. doi:https://doi.org/10.1016/j.compeleceng.2022.108356

Taulli, T. (2020). *The Robotic Process Automation Handbook: A Guide to Implementing RPA Systems.* California : Apress.

Tracy, S. J. (2010). Qualitative Quality: Eight “Big-Tent” Criteria for Excellent Qualitative Research. doi:https://doi.org/10.1177/1077800410383121

Uklańska, A. (2023). Robotic Process Automation (RPA) – Bibliometric Analysis and Literature Review. *Foundations of management, 15*(1), ss. 129-240. doi:https://doi.org/10.2478/fman-2023-0010

Vaismoradi, M.;Turunen, H.;& Bondas, T. (2013). Content analysis and thematic analysis: Implications for conducting a qualitative descriptive study. *Nursing & health sciences*. doi:ttps://doi.org/10.1111/nhs.12048

Walkowiak, E. (October 2023). Task-interdependencies between Generative AI and Workers. *Economics Letters, 231*, 111315-111315. doi:https://doi.org/10.1016/j.econlet.2023.111315

Appendices

Appendix 1. Structure of the interview

Background information

- Briefly describe your job and responsibilities at work.
- What is your experience level with low code platforms?
- Do you have any technical background?

Generative AI usage related questions

1. Were the Generative AI features able to understand and execute your requests accurately?
2. Did you encounter any limitations in the capabilities of the Generative AI features?
3. Did the Generative AI features influence your approach to problem-solving in the development process?
4. How does the integration of Generative AI influence the functionality of low code automation development tools?
5. What are the potential challenges and limitations in using Generative AI features in low code automation development?
6. How can the previously mentioned challenges and limitations be limited?
7. Based on your experience, how likely are you to continue using these AI features in future projects?
8. Any other comments or feedback about your experience?

Appendix 2. Role of interviewees

Role of interviewee
RPA expert
Citizen developer
Citizen developer
Citizen developer
RPA expert
RPA expert
Citizen developer
RPA expert
RPA expert
RPA expert