Hochschule Neu-Ulm
University of Applied Sciences

Bachelorarbeit

im Bachelorstudiengang **Wirtschaftsinformatik**

an der Hochschule für angewandte Wissenschaften Neu-Ulm

How Data-Driven Capabilities enable for Low-Code Application Development – an Empirical and Theoretical Investigation

Erstkorrektor: Prof. Dr. Weeger

Zweitkorrektor: Prof. Dr. Buchwald

Verfasser: Mihael Hristov (Matrikel-Nr.: 3137146)

Thema erhalten: 01.06.2024

Arbeit abgeliefert: 19.08.2024

Digitaler Anhang

1x ZIP-Datei

Abstract

In recent years, Low-Code platforms have emerged as powerful tools for rapid software development, enabling non-traditional programmers, known as Citizen Developers, to contribute significantly to the development process. These platforms not only facilitate a faster development cycle but also empower Citizen Developers to configure and innovate through intuitive, user-friendly interfaces. Central to the success of these platforms are their data-driven capabilities, which play a crucial role enabling the integration, management, and manipulation of diverse data sources. By leveraging data-driven functionalities such as automated workflows and third-party APIs integrations, Low-Code platforms allow applications to be more responsive, adaptive and data-centric. These capabilities are significant as they enable Citizen Developers to create data-powered applications without requiring deep technical expertise. As a results, organisations can harness the potential of data to drive innovations and maintain a competitive edge in an increasingly data-centric business landscape. The integration of data-driven capabilities into Low-Code development not only enhances the functionality of and scalability of applications but also broadens the scope of what can be achieved through Citizen Development, making a transformative shift in how software is conceptualized and delivered.

Keywords: Low-Code Application Development, Data-Driven, Technical Capabilities, Citizen Developer, Organisational Capabilities

Table of Contents

List of Figures

List of Tables

.

List of Abbreviations

GUI	Graphical User Interface
LC	Low-Code
LCA	Low-Code Application
LCAP	Low-Code Applications Platform
LCP	Low-Code Platform
LCDP	Low-Code Development Platform
UI	User Interface

1 Introduction

Faster software development is crucial in the current time of digital transformation, given that the development time of applications highly exceeds the delivery time. The results of that is that the growing demand for applications is higher than what IT departments can supply (Chang and Ko 2017).

Speed and agility are highly required to maintain customer's requirements. This is why Low-Code Platforms (LCPs) offer a new way to shape organisations and their software delivery. These changing circumstances also affect developers and their delivery time by opening new opportunities. Characterised by its simple and easy to learn functionalities, LCPs offer a high abstraction environment, which can be configured faster than high complex programming solutions (KPMG 2022). Through the drag and drop functionality, Low-Code (LC) offers a new way to develop and configure software, also enabling users with minimal to no programming experience to take part in the process (Richardson et al. 2014).

In this current era of a new digital shaping, in a report by KPMG (2022), low-code platforms have been identified as a key technology, to enable faster software development. Also, back in 2019 a report by Gartner, Vincent et al. (2019) predicted that by 2024 Low-Code development will be accountable for more than 65% in the software development market, which shows its popularity nowadays. In its first introduction by the research company Forrester, LC is being assimilated to rapid application development by use of declarative and visual techniques, so that coding is minimized (Rymer and Appian 2017). LC is characterised by a graphical user interface (GUI), with drag and drop operations and pre-designed templates being the most common features (Luo et al. 2021). These kinds of operations have a higher learning curve.

However, since LC is a simpler approach to develop software, it is still requiring some technical expertise, where knowledge is required (Sahay et al. 2020). For the enablement of LC one of the most important technical capabilities are involving creation of workflows, Third-Party Integration via APIs and different aspects involving integration, management and manipulation of data.

Given that Low-Code Applications (LCAs) are often developed by Citizen Developers – individuals who may not have formal training in software development - several important questions arise. Since the literature suggested that, despite the simplified nature of Low-Code development, the process may still involve technical knowledge, which requires a specific skillset (Khorram et al. 2020). This raises concerns about the extent to which Citizen Developers can effectively leverage these platforms without encountering significant challenges. Furthermore, it prompts an exploration into whether current LCPs adequately support these users in acquiring and applying the necessary technical skills, particularly in the context of data-driven capabilities.

In the context of LCPs, data-driven capabilities encompass the processes and tools that enable the integration, management and analysis of data, within LCPs. These tools are not only foundational for creating functional and robust applications but also ensuring that the applications can scale, adapt, and meet evolving business needs. There has been some practical insight, which have developed data-centric applications within different LCPs, but have not yet shown how data is being enabled and driven (Gürcan and Taentzer 2021).

Therefore, this study seeks to address this research gap by investigating how data-driven capabilities enable low-code application development. By examining these capabilities, the study aims to contribute to both the theoretical understanding and practical application of Low-Code Application Platforms (LCAPs), providing valuable insights for organizations looking to optimize their use of these platforms in data-centric application development scenarios.

- How can Data-Driven Capabilities enable Low-Code Application Development?

The aim of this thesis is to present a theoretical background regarding LC and where data is being enabled. Afterwards qualitative research has been conducted by interviewing experts within LCPs. For this study 7 participants have been interviewed within one company to elaborate on how they enable the data-driven capabilities within their applications.

2 Theoretical Background

The goal of this chapter is to provide the theoretical foundation and definitions of LC. First of all, the term Low-Code and its characteristics are presented. In order to understand how a Low-Code Platform works, it is necessary to give an overview of its development roles. Furthermore, a demonstration of data capabilities in LC environment is explained. By showcasing this, the data-driven capabilities, as well as their challenges are discussed. Lastly, this chapter ends with the introduction of the research model and the resulting hypotheses.

2.1 Low-Code

2.1.1 Definition

Several indicators suggest that the term low-code was presumably coined by the market research company Forrester in 2014 (Bock and Frank 2021b; Richardson et al. 2014). Low-code platforms enable rapid delivery of business applications with minimal hand-coding. They provide comprehensive declarative tooling, including model-driven development and visual configuration, allowing users to define data, logic, flows and forms with little to no coding. The drag-and-drop composition feature simplifies the application development process, making it accessible to non-developers (Richardson and Rymer 2016). Furthermore, LCPs (Low-Code Platforms) are a cloud-based environment, allowing users a seamless integration of multiple development tools and libraries (Sufi 2023).

Furthermore, common features of LCPs include components for defining data structures, accessing external data sources through APIs, and developing graphical user interfaces. They also support various external services, offer mechanisms for basic functional specifications like business rules, and often provide libraries of standard operations. Moreover, these platforms typically include workflow modelling components, advanced coding options, and role-based access control systems. Additionally, LCPs differ from traditional software development by consolidating various development tools into a single system. While conventional development requires separate tools for tasks like coding, modelling, database management, and deployment, LCPs integrate these functionalities. This integration reduces the need to switch between different environments and simplifies the process of maintaining consistency and integrating different development artifacts, thereby streamlining the overall development workflow (Bock and Frank 2021b).

2.1.2 State of Art in Research

A summary of the various application characteristics that can be enabled within LCDPs to guarantee data capabilities with their corresponding reference is given in Table 1. The first column provides a high-level term for the corresponding characteristic. Secondly, the description gives a brief overview of the corresponding term. The column “Reference” provides information about the research papers or authors that have mentioned these characteristics. Multiple mentions are possible in this context.

Low-Code Characteristics	*Description*	*Reference*
Low-Code Execution Environments	Provides environments where users can deploy and manage applications without deep technical expertise, often including drag-and-drop interfaces and pre-built modules.	(Hyun 2019)
Application Modeler	Enables users to model applications using visual tools, which can then be deployed without extensive coding.	(Chang and Ko 2017; Lethbridge 2021; Martins et al. 2020)
Model-Driven Engineering (MDE) Support	Supports model-driven approaches for software development, enabling complex workflows and integration with other engineering models.	(Ihirwe et al. 2020; Tisi et al. 2019)
Automation Capabilities	Facilitates the automation of business processes, reducing manual intervention and improving efficiency.	(Waszkowski 2019)
Usability and Testing Tools	Provides tools for testing, debugging, and improving the usability of applications, which is crucial for adoption among non-technical users.	(Khorram et al. 2020; Silva et al. 2021)
Scalability and Performance Optimization	Allows for the development of scalable applications, including support for large-scale data processing and optimized performance through efficient resource management.	(Horváth et al. 2020)
Integration of External Services	Connects to external services through APIs and other connectors, allowing seamless integration and data exchange between different platforms.	(Bock and Frank 2021a; Sahay et al. 2020; Zolotas et al. 2018)
Platform Backend Management	Handles code generation, optimization, and management of database systems, microservices, APIs, and model repositories.	(Sanchis et al. 2019)
Advanced Technology Integration	Facilitates the integration of machine learning (ML), internet of things (IoT), and artificial intelligence (AI) into applications, enhancing the platform's capabilities.	(Arora et al. 2020; Chang and Ko 2017; Martins et al. 2023)

Table 1: Overview of the main capabilities in Low-Code based on Prinz et al. (2021)

2.2 Technical Capabilities

According to Sahay et. al (2020) the main components of the platform can be categorised in three tiers. The first one is the application modeler, which allows users to modify their abstracted environment. Every LCP provides developers with a GUI designer, where pre-defined components and templates can be used (Bock and Frank 2021b). Within this stage, the designing of the user interface (UI) is considered

simple. The options of available components and their scaling can differ from platform to platform. Furthermore, there is the possibility to design a custom template. Moreover, static and dynamic pages can be created, where a report dashboard can be added to an application. The configuration of the UI is done via drag and drop (Gürcan and Taentzer 2021).

Furthermore, the components need to be connected to a data source. The field of a component represents an attribute of the data table. In the paper of Gürcan and Taentzer (2021), the authors have implemented an application in order to test the capabilities of three of the main LC vendors from the Magic Quadrant (Vincent et al. 2019) – PowerApps, Mendix and OutSystems. A further capability that LCPs offer is regarding data modelling. Based on the examined platforms, data can be created manually and stored within the environment. There is also the possibility to integrate external data sources using connectors, that are offered by the vendors catalogue. Data manipulation is a feature that is not always supported by the platforms, but some vendors allow users to view and edit data within the environment (Gürcan and Taentzer 2021).

The second tier is the server-side functionality. It handles the backend operations of the platform, including managing and optimizing the application model. This is defined by components such as database systems, model repositories, micro-services, APIs and connectors. This tier manages the technical aspects like authentication, business logic consistency, data integrity and security (Sahay et al. 2020). In the paper from Gürcan and Taentzer et al. (2021), the authors also reported on their experience with the above mentioned platforms. It is noticed that the LCPs use workflows, such as microflows and server actions, to handle the complex backend operations, like data manipulation, integration with external services and custom logic development. This allows seamless interaction with external systems while abstracting the technical complexities from the user.

The third tier integrates the different external services such as micro-services, third-party connectors, collaboration platforms and databases. It handles the integration of these external services, so developers do not need to manually manage these connections (Sahay et al. 2020). This also includes the use of APIs, which are enhancing the functionality and interoperability of LCAs. In Gürcan and Taentzer et al. (2021) a connection to a RESTful API on a local server was created. Low-Code Development Platforms (LCDPs) provide varying levels of API management capabilities, as explained by Overeem et al. (2021). LCPs generally support modern API protocols like REST and SOAP and offer basic security features such as Oath 2.0. However, advanced API management tasks, such as lifecycle management (e.g. versioning), performance optimization (e.g. traffic prioritization), and observability (e.g. monitoring and analytics), often require custom implementation or external tools. While LCDPs aim to simplify API management for non-technical users, many complex tasks still necessitate technical expertise, highlighting the need for further advancements in these platforms for better support of the API management. Considering the data connection, Sahay et al. (2020) noted that there can be pre-implemented connectors offered by the LCP, but also custom connectors, which give the flexibility to connect from heterogeneous data sources. Moreover, there is also the possibility to use data gateways to bridge cloud platforms with on-premises servers, enabling local data to be utilized within cloud applications without data migration. Furthermore, APIs can be created from scratch. This usually involves specifying endpoints, requests and responses structures and authentication methods. Basic authentication is offered and supported by the LCP.

2.2.1 Architecture of Low-Code Applications

There are different propositions for the architecture of low-code platforms and there is not one clear architecture that addresses all LCPs, although most of the layers have the same purpose. However, the

layered architecture by Sahay et al. (2020) displayed in Figure 1, covers the topics featuring in this thesis.

In Figure 1, the features of Low-Code are covered by every layer in the architecture. Firstly, there is the deployment layer, which covers the infrastructure of the platform and features on-premises environments. The data integration layer is handling the integration of data sources, which can be from heterogeneous or homogeneous sources and allows the different data operations. The service integration layer aims to connect the platform with different services (e.g. APIs). The development is done on the application layer, which enables the use of the graphical interface and its features, allowing for users to modify the design (Sahay et al. 2020).

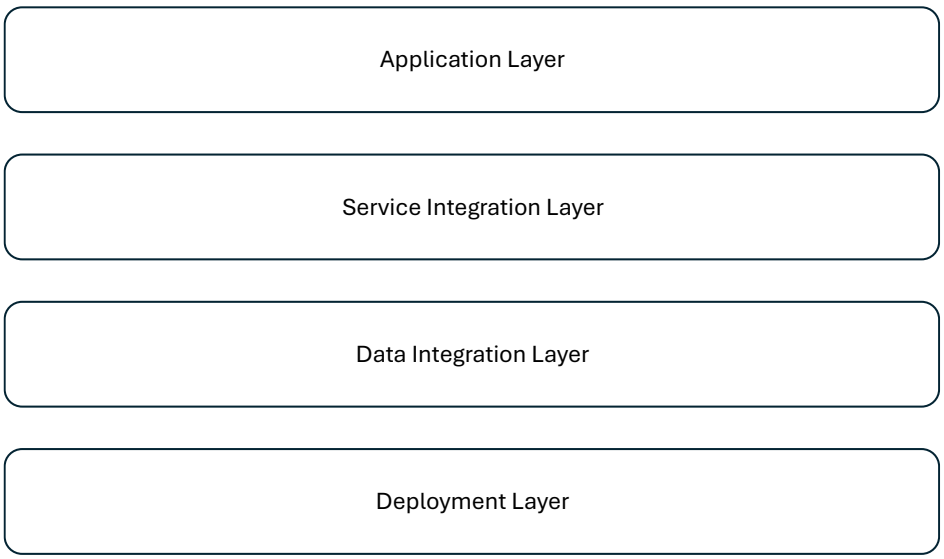

Figure 1: Architecture of LCDPs based on Sahay et al. (2020)

For a better understanding how the features of LCPs are utilized, in Bock and Frank (2021b) a model that structures the different functionalities to a perspective has been created. In the static perspective all the data operations are done. This perspective includes data modelling, data structure, use of internal databases and the possibility of accessing an external data source through API. In the functional perspective all the functionalities of the platform are handled. This perspective includes specific expression languages for decision rules and dialog-based ways of specifying program flow conditions. Moreover, there is the possibility to connect the platform with external services such as APIs and different web services. Furthermore, there is the dynamic perspective, which covers occasional features such as workflow management and its modelling. The interactive perspective is the intersection for the user. The main feature, which is provided by every platform is the GUI designer. It enables users to bind the different forms to the data instances, so that at the end the final solution is created, which can be accessed on different end-user devices. The author also adds an additional aspect such as roles and user rights and pre-implemented functionalities for business intelligence, artificial intelligence and robotic process automation (Bock and Frank 2021b).

2.2.2 Low-Code Development

On top of that, Sahay et al. (2020) also presents the different steps of the development process in LCDPs. It begins with the initial phase of data modelling. This step is highly required to create a data schema for the application. For its initialization, users need to define entities, set up relationships, specify constraints and dependencies. This phase is used by the drag and drop functionality and from an application perspective it represents the backbone of an application. Next, users configure forms and pages that define and finalize the UI outcome of an application. In the third phase, users can manually program

a workflow. Different operations can be implemented within the configured workflow, which can trigger various actions. These workflows are using business process model and notations-like representation. Moreover, to extend an application and increase its scalability external services via Third-Party APIs can be connected. This step overlaps with the third-tier component of LCPs. To achieve an external APIs integration from third parties, users need to consult documentation to understand the data format and structure supported by the platform. Finally, Sahay et al. (2020) points out that application deployment is a simple development step to execute. Applications can be previewed pre-deployment and simply published.

Additionally, a study by Al Alamin et al. (2021) analyses the most common challenges in the development of LC. His study focuses on difficulties expressed in the online developer forum Stack Overflow. An empirical study of five thousand posts was completed, which contain questions and accepted answers of nine widely used LCPs. In the research, it was suggested that the development of LC can be compared to the agile development methodology, because the main goal of LCDPs are the continuous incremental improvement and customer satisfaction. Based on agile software development, the authors expand the previous presented development steps by (Sahay et al. 2020), with the inclusion of Testing and Maintenance. In Figure 2, a comparison with the similarities is displayed, having that LCDPs steps are represented in the inner circle. Furthermore, the authors clustered the initial thirteen topics into four categories ordered by most searched: Customization, Platform Adoption, Database Management and Third-Party Integration. The categories also include sub-categories, which are described in detail. Considering the issues experienced while developing, the authors have decided to ascertain the correlation between difficulty and popularity of the topics. The results computed by the previous explained metrics show that, the three most difficult topics that developers experience are concerning the Customization and Third-Party Integration. Dynamic Event Handling is a sub-category of Customization and relates to issues involving the creation of dynamic and interactive applications. The Third-Party Integration involves issues regarding external APIs and external web requests processing. In addition to the Third-Party Integration of a project by Dhoke and Lokulwar (2023), it is also highlighted that API implementations bring issues such as complex data structures, interaction with external resources, handling errors and edge cases and testing and debugging.

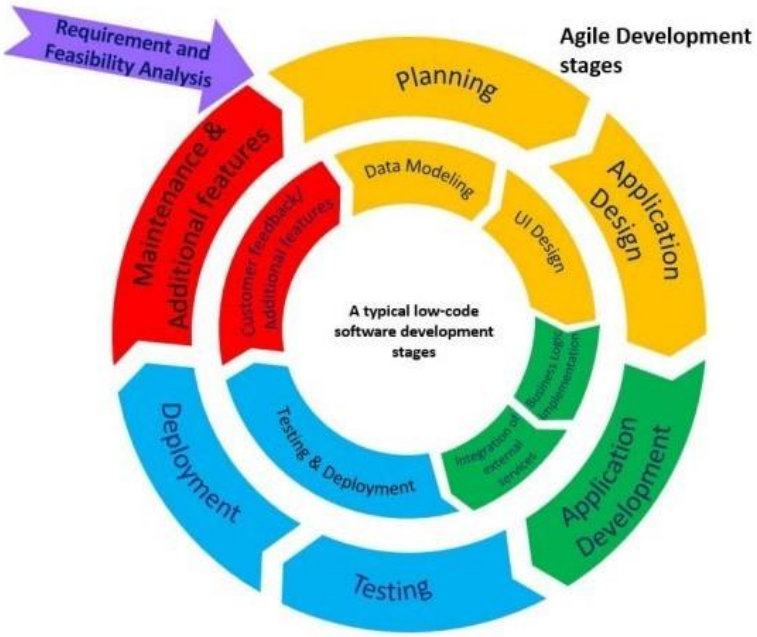

Figure 2: Agile methodology vs Development in LC

2.3 The Role of the Citizen Developer

2.3.1 Definition

LCDPs enable the inclusion of employees, apart from the IT-department, which are called Citizen Developers (Carroll and Maher 2023). In this current state of digital transformation there is a shortage of software developers (Breaux and Moritz 2021), so that IT-competencies should be wide-spread. For this matter, in urge to offload IT-departments, Citizen Developers are being adopted in organisations to rapidly develop business applications with low complexity (Carroll and Maher 2023). The Citizen Developer has an active role in the development of LCDPs (Khorram et al. 2020). These users are mostly characterised by having minimal to no coding experience and aim to develop software solutions by visual programming via drag and drop and using pre-designed templates, as mentioned above (Carroll and Maher 2023; Luo et al. 2021). The goal of this concept is to increase the operational efficiency of development that addresses specific business needs while reducing the IT-departments workload (Hoogsteen and Borgman 2022). Moreover, in a research by Lebens et al. (2021), it is stated that Citizen Developers are present in the majority of the 38 explored organisations regardless of size. Therefore, it can be confirmed that due to the rise of LC tools in organisations, the adoption of Citizen Developers reflects this trend accordingly.

An important feature of this concept is that Citizen Developers need to have domain expertise in their department so that they can properly use the toolset of LCPs. Given the low entry barrier of LCDPs, the users can focus mainly on the business logic of an application without needing extensive programming knowledge. Through empowerment of the independent application development by Citizen Developers, organisations reduce the backlog of IT and improve its business agility (Hintsch et al. 2021). Nevertheless, despite the low entry barrier for Citizen Developers, in Hoogsteen and Borgaman (2022) it is noted that some companies perceive the use of LC tools as a complex task requiring specialized skills.

2.3.2 Skillset

The current literature in the research has not yet standardised an exact skillset, that Citizen Developers should have. However, given the current research on LC and the technical capabilities, which were previously presented in this work, there can be some specific characteristics that Citizen Developers are expected to have or can develop during the development.

Furthermore, there are some disagreements regarding the skillset that a Citizen Developer should have. Some authors describe them having non-technical or not having formal IT education (Carroll and Maher 2023; Luo et al. 2021). In Binzer (2023), an analysis of the job market around LC and Citizen Developers and the required knowledge, skills and abilities has been presented. The authors concluded that the creativity and problem-solving skills of Citizen Developers are prized. Moreover, since collaboration is required for the development, companies value teamwork, clear communication and comprehension of business operations.

However, for the development of such platforms, there are some technical aspects, which are required to maintain the development and the business understanding of it. There are no specific attributes that a user must have, because the requirements from platform to platform differ and this is why a Citizen Developer is mostly described as an user with minimal to no coding experience. According to Smith et al. (2020), there are technical operations that users in these platforms must perform, such as SQL queries and understand the technical concepts of databases and APIs. Likewise, the examination by Binzer (2023) also suggest, that technical skills such as basic programming, web and mobile development, data analytics are found in the research. Citizen Developers should also be familiar with DevOps practices, cloud services, continuous integration and system integration using APIs.

2.3.3 Development & Testing

Taking it one step further, Citizen Developers also take actively part of the LCDPs. In a research paper by Krejci et al. (2021), it was found that Citizen Developers help developing applications. However, criticism is expressed about their involvement in the technical phase. The perceived problem is regarding the development, which should not be done on Citizen Developer's own, because the implementation is rather technical. Furthermore, considering the integration of data, it is stated that business users, referred to as Citizen Developers, struggle to handle data structures, even in a simple application (Krejci et al. 2021). There are also challenges concerning the data integrity and security that Citizen Developers have (Heuer et al. 2022). Given that the first step of the development begins with data modelling and with the possibility to integrate data sources the developed application can be negatively impacted as a result (Sahay et al. 2020).

Moreover, given the agile development process in Figure 2, testing is also a phase, where Citizen Developers can take part. In Khorram et al. (2020), it is pointed out that technical expertise is required when LC testing to correct bugs and potential quality issues. Furthermore, in order to test correctly a high level of automation to maintain the development speed is required. For this matter scripting languages must be written, which was also acknowledged in the job market analysis (Binzer 2023). Third-party integration is part of the development, as mentioned above, but the tools involving the third-party integration must also be tested. Integrating these tools within the LCDP environment and ensuring their usability can be difficult, adding to the challenges concerning the Citizen Developer (Khorram et al. 2020).

Given that, highly technical activities in LCDPs can be executed by Citizen Developers, there is also the possibility to enable APIs consumption. However, providers do not always support Citizen Developers API management, which forces Citizen Developer to seek help from the professional developers. Moreover, API Management is a task, which can often be required in some applications and with missing support from providers, the process can get highly complex or compromises developed applications (Overeem et al. 2021).

2.4 Organisational Aspect

Organisations play a critical role in LCPs towards the enablement of its potential, in order maintain and deliver digital solutions, in a market that has an increased demand for software developers (Alt et al. 2020). It is very important that an organisation applies the right strategy assigning Citizen Developers with application development in LC environment, but also selecting the right LCP, which can utilize the organisational needs (Carroll and Maher 2023). To further elaborate, organisations must analyse and deploy different aspects involving the structures around the LCDP lifecycle. These are categorized to social and technical systems (Bostrom and Heinen 1977). In Prinz et al. (2024) an overview is presented, using the perspectives proposed by (Bostrom and Heinen 1977) for Management Information Systems, which is displayed in Figure 3.

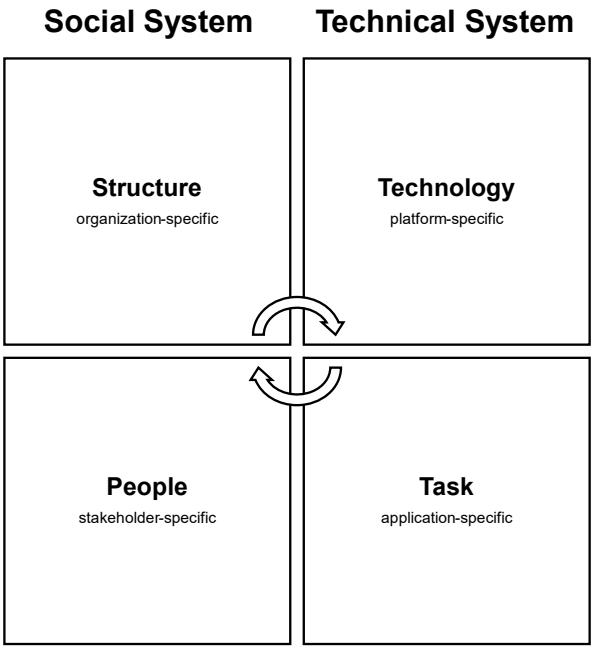

Figure 3: Social-Technical Model for LCDPs based on Prinz et al. (2024)

2.4.1 Social System

The social system includes the structure, and the people involved within the LCDP. In Prinz et al. (2024), the people are identified as stakeholders taking part in the development process and are operationally active within the development lifecycle being able to develop and maintain their applications. For this matter the organisations should provide Citizen Developers with training concepts and consultancy. This is essential, since Citizen Developers might perform technical activities in the development, such as using APIs (Overeem et al. 2021). Furthermore, given that Citizen Developers have an active role in the LCDP developing applications (Khorram et al. 2020), while not being a member of the IT-department (Lebens et al. 2021) and the perception of organisations viewing application development in LCDP as a high complex task (Hoogsteen and Borgman 2022), trainings and consultancy are necessary. The enablement concerns processes in LCDPs regarding data, APIs management, agile development and also application quality (Sahay et al. 2020).

Equally important is the structure of the organisation, which must establish and maintain different capabilities to effectively leverage LCDPs. Firstly, it is required to provide the infrastructure required for widespread LCPs. Additionally, a clear criteria for selecting LCDP needs to be set up, which is aligned with the organisations strategic goals and technical requirements (Prinz et al. 2024). Another key point is that a governance strategy must be implemented, so that the quality of an application during its lifecycle can be assessed and ensured. This is essential, since Citizen Developers can struggle in the development process, which can negatively impact the outcome of an application (Hintsch et al. 2021).

Along with that, security is an important feature in LCDPs, which must be addressed in detail. This concerns both social aspects. Hereby, Citizen Developers have a direct impact on security inconsistencies. These vulnerabilities can occur, when not having an organisational guidance. To elaborate on that, it is pointed out that most risks concerning Citizen Developers are data security, quality assurance, maintainability and poor architecture apps (Hoogsteen and Borgman 2022).

For this matter, on organisational level, there are security protocols, authentication and user access control infrastructures to be established (Sahay et al. 2020). A framework for setting up, running, maintaining, and constantly enhancing an information security management system (ISMS) is provided by

the ISO 27001 certification. This standard is essential because it makes sure that information security risks are properly addressed and mitigated while assisting organisations in managing and safeguarding their information assets in methodical and economical manner. Additionally, the certification serves as a ”signaling tool” that established the company’s credibility and trustworthiness by demonstrating its dedication to information security (Hsu et al. 2016).

An important aspect of the social-specific capabilities is to encourage employees to adopt and use LC (Prinz et al. 2024). In Carroll and Maher (2023), an enablement program called “Do it yourself” to enhance their digital transformation and Citizen Developers was started. It is highlighted that two years after the program was launched, critical success has been achieved, in which Citizen Developers were crucial. The tasks covered by Citizen Developers were involving data automation and visualization, improvement of inefficient workflows and customer’s or employee’s experience. However, as mentioned above, it is stated that a right governance should be established, which rather embraces than hinders the development.

2.4.2 Technical System

Along with that, the technical systems should also be enabled. From an organisational point of view, there are platform and application specific systems, which must be taken care of.

To ensure that effective development and management of applications within an organisation is granted, several key application capabilities must be established. It is essential that application classes are defined. This involves categorizing applications based on their complexity, functionality and business impact. Clear classification helps applying appropriate development and management strategies for each type of application (Heuer et al. 2022).

To support the classification process, organisations should develop a set of questions and response options. These questions guide the assessment and categorization of applications, ensuring a consistent and structured approach to classification (Prinz et al. 2024).

Furthermore, it is encouraged that software engineering best practices are applied to maintain software development, given that Citizen Developers can build applications on their own. This requires establishment and understanding of agile development, which must be passed on the developers in LCD. The practice highlights the fact that applications are an ongoing process, which is based on continuous improvement (Lethbridge 2021).

Proceeding with the platform specific capabilities, the setup of the LCDPs environment is crucial. This involves configuring the organisations specific needs, including setting up development, testing and production environments.

Along with this, providing user-specific access is essential. Since Citizen Developers mostly get to configure in a high-abstracted environment, this ensures that each user has the appropriate permissions and access levels based on their role (Hoogsteen and Borgman 2022). This was also acknowledged in (Carroll and Maher 2023), specifically highlighting that data is a critical resource for an application and its use should be tracked and controlled.

Another key point is the seamless usage of APIs, which as mentioned above are proven to be a decisive factor for the interoperability in LCDP (Sahay et al. 2020). In Overeem et al. (2021), API-Management of four LCPs were analysed, concentrating on six different focus points. It stands out, that regarding the security of APIs, the researched LCPs provide modern security standards, but when it comes to Threat Detection and Protection developers are left on their own. A further aspect that has been identified is about the custom API solutions, which are beyond a Service Level Agreement. In such cases, third-party API management tools are required (Overeem et al. 2021). For these reasons, organisational

actions are necessary, to enable a smooth enablement of APIs, which are provided with the needed connectors (Prinz et al. 2024).

2.5 Research Model

This thesis aims to research the data capabilities of the LCPs, Citizen Developers and the organisation. To achieve this, the theoretical background of LC was presented for a better understanding of its structural properties. Since the data capabilities are a central part of this thesis, an overview of their current enablement was presented as well. Based on the research, a deeper understanding for data-driven processes is required. Given the ground foundations, it is now possible to formulate a research model, presented in Figure 4.

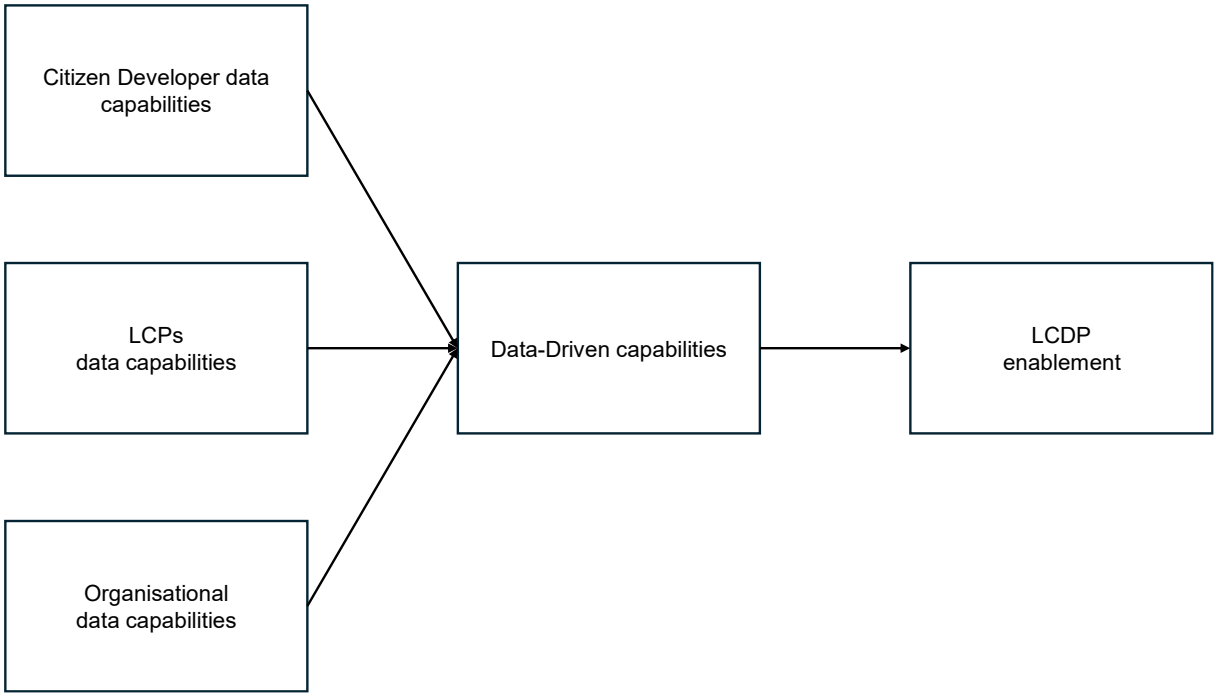

Figure 4: Research Model

The starting point of the research model are the data capabilities. The three starting points of this model have been identified in the theoretical background, where data capabilities are used. Having that, the are several factors that may affect the use within the different areas.

Firstly, Citizen Developers are seen as a part of the LCDPs for application development with minimal to no programming skills required (Luo et al. 2021). Although LCPs aim to simplify the application development, Citizen Developer with no solid understanding of the data capabilities could struggle to fully leverage the platforms potential. This lack of technical comprehension can hinder their ability to use databases and APIs, which are essential for building robust and functional applications (Smith et al. 2020). Therefore, to fully empower Citizen Developer and optimize the advantages of LCPs, technical knowledge enhancement is essential.

Furthermore, LCPs have several features considering the development process. Typical phases, which are performed begin with data modelling so that users can configure their data by creating entities, establish relationships and dependencies. Then, the user interface is configured by using the drag &

drop function. In this step the data, that was initiated in the first step is rendered. Afterwards, user might configure the workflow of the application. This step might require different operations of the workflow, which could be triggered in case a specific scenario is wished (Sahay et al. 2020). However, the previous explained steps are within the LCP environment. In order to bind an application with external services or data sources, an API connection is required. This can be utilized, only if the LCP provides data connectors (Bock and Frank 2021b).

Building on the last hypothesis organisations need to enable the technical abilities of the developer (Hoogsteen and Borgman 2022) and the technical abilities of the platform (Bock and Frank 2021b). Developers should be technically enhanced to develop applications unrestricted. In combination of that, organisations must grant the appropriate user access and to ensure that there is transparency of different LCPs and their licensing models. Furthermore, considering the development, it is critical to guarantee data availability by giving current systems access to APIs. This aspect includes the capability of the organisation to provide developers with available connectors that are available and verified in order to utilize full potential of the application (Prinz et al. 2024). This affects the availability of external services and external data connectors, which are accessed via APIs (Bock and Frank 2021b). For this reason, the following assumption can be made:

The previously discussed points highlight that data-driven capabilities can be enabled from three perspectives: organizational, LCP, and citizen developer. To trigger the enablement Citizen Developers must have a specific skillset, which is required to actively empower the rapid application development (Binzer 2023). Furthermore, the platforms must supply with functional and seamless data-driven capabilities so that Citizen Developers can utilize their performance (Sahay et al. 2020). Lastly, organisations must provide both platform and its employees with a seamless, abstracted and effective enablement of the data-driven capabilities (Prinz et al. 2024). .

3 Research Methodology

In the previous chapter, the theoretical foundations of LC were presented to (1) get an overview of the current state of LCDPs and their capabilities and (2) to be able to assign different topics to a clear structure in the overview. The purpose of this chapter is to outline the research methodology of this thesis. Firstly, the selected method of the research is presented and the suitability of this method to address the research question is explained. The sampling strategy and the companies, which were investigated are displayed next. Following that, the data collection process and the interview partners are presented.

The aim of this thesis is to research which data-driven capabilities are utilized in LCDPs. After identifying the various aspects of LCDPs, it is important to investigate the ways and circumstances in which the data-driven capabilities are used in the application development.

For this matter the qualitative research was chosen. The qualitative research involves understanding its distinct characteristics. There are numerous definitions, that highlight different facets of its uniqueness. According to Bryman et al. (2008), it favours language over quantification. Furthermore, Alasuutari et al. (1995) notices that the qualitative analysis is similar to solving riddles, where each clue helps identify a distinct style of reasoning and reduces the pool of potential answers (Hammersley 2012).

In Aspers and Corte et al. (2021) it is described as an iterative process aimed to achieving improved understanding within the scientific community by making significant new distinctions that come from closely studying the phenomenon. The methodological approach differs from the quantitative research, as it focuses on testing hypotheses using explicit research designs. Furthermore, data and analysis in the qualitative, are collected through words, texts and detailed descriptions, where the methods are interviews, observations and context analysis. This aims to understand a certain experience and its approach, rather generalizing (Aspers and Corte 2021).

Given the previous work by Aspers and Corte et al. (2019), there are four characteristics defined, which collectively contribute to the qualitative research. Beginning with distinction, it marks the creation of significant new distinctions, which are crucial providing new insights and contribute to a better understanding of a specific topic. In addition to that, articulating differences can occur, which can expand the scope of knowledge. The second characteristic involves the process, which is iterative and dynamic. Through the back and forth between theory and empirical data, researchers can refine their comprehension and gain an incremental improvement. It is remarked that being adaptive and flexible to fresh data. Thirdly, a closeness to the researched phenomenon is required, to fully comprehend the context and the point of view. This involves direct interaction through interviews, observations or other methods of data collection. Finally, through the combination of the previous characteristics mentioned, the main objective is to gain an improved understanding. This is enhanced by gaining new insights and a deeper comprehension. Moreover, this feature emphasizes the interpretive nature of qualitative research, where understanding is seen as an ongoing process of discovery and explanation.

For this matter, the thesis is built on qualitative research. The reasoning behind that, is that the main goal of this topic is to present the current data-driven capabilities of LCDPs and to gain an understanding of how these are used in the practical area. To ensure the acquiring of understanding, expert interviews were conducted for the detailed insights based on the topic. Experts are explained as individuals that have specific knowledge and hold a certain status or function in decision-making processes (Döringer 2021; Meuser and Nagel 1991). This applies to the methodology of this thesis, which aims to gain information from experts that are responsible for the development, implementation or control of a solution.

This helps to assess actions and knowledge about LCDPs. Furthermore, by examining individual perspectives and reconstructing actions, this approach can reveal the underlying motivations and thought processes that drive expert decision-making. This process helps to systematically break down, examine and interpret expert knowledge, providing a detailed understanding of how data-driven capabilities are perceived and implemented in LC environments (Döringer 2021).

3.1 Sample

Given the prediction by Vincent et al. (2019) that 65% of the application development will be done in LC, it would be suitable to examine whether there has been a growing tendency towards LCDPs since then. According to Waszkowski et al. (2019), the industrial sector can profit from the usage of LC, due to its rapid development, ease of use and involvement on non-professionals. In addition to that, LCPs can enable quicker responses to market changes and allow flexible application development. For this reason, this thesis will focus on one company from the industrial sector to evaluate the capabilities and efficiencies of the data-driven capabilities, which can enable the application development.

Moreover, it is also important to have a variety of LCPs instead of focusing on just one. This helps to have a better overview and direct comparison of the varying capabilities available of the different platforms. According to the Magic Quadrant, the LCPs are divided in four categories: niche players, visionaries, challengers and leaders. For the research of this thesis the capabilities of three LCPs were analysed. To ensure that there is variety of capabilities covered, two of the leaders from the Magic Quadrant have been examined, which are the platforms from Microsoft and OutSystems. Furthermore, ServiceNow from the challenger's category was also analysed. The reason for the selection of these LCPs is to have an insightful understanding of LCPs that can shape the direction of the market. Additionally, a perspective of a non-leader, which could be well-known or have a large following, but lacks features that prevent it to become a leader, is shown. Microsoft is a market leader in LCAP and their product, the Microsoft Power Platform, consists of PowerApps, Flow, PowerAutomate and Common Data Service. The platform works very well for use cases where it extends Dynamics 365 and Microsoft Office 365. Furthermore, there are many third-party data connectors included (Vincent et al. 2019). OutSystems, which also is a leader, offers web and mobile application for development. There are two integrated development environments. The first one is Service Studio for building user interfaces, data, business logic, and processes. The second one is Integration Studio for developing custom extensions and components (Gürcan and Taentzer 2021; Vincent et al. 2019). Lastly, ServiceNow, primarily know as an IT service management Software-as-a-Service provider, is a challenger and offers the NowPlatform as LCP. The primary business of offering Software-as-a-Service for IT users ensures a prepared audience for centralized application development on the NowPlatform. Developers can utilize the platforms capabilities, including portals and chatbots, for creating new applications (Vincent et al. 2019).

As a result of that, one company was selected to be examined. The reason behind that, is that a single company with multiple expert interviews can provide several advantages. This approach is particularly effective for the qualitative research, enabling practical insights, which can involve a collection of diverse perspectives and experiences within one instance. As an addition to that, the risk of misinterpretations and errors can be reduced, which allows the collected data to be accurately presented (Flyvbjerg 2006). For this purpose, the first step involved searching online for companies via LinkedIn and email that have deployed the previously mentioned LCPs. After establishing some contacts for interviews with one company, I built on those connections and used them for further networking. Initially one company was planned to be interviewed. It is important to notice that the series of interviews were conducted from a company on two distinct levels. The overarching entity, which is referred as company B, has a subsidiary referred to as company A. Although the different entities are part of the same organisation, they have

different focus areas. This structure allows for a diverse range of perspectives and specializations within the organisation. The experts within the organisation were contacted via email.

Company ID	*Company Type*	*Number of Employees*	*Description*	*Number of Interviews*
A	Automation Technology	34,000	Operates in the field of mechanical engineering and automation	6
B	Automotive	430,000	Automotive and technology company that provides hardware and software solutions	1

Table 2: Interviewed Companies

3.2 Data Collection

As mentioned above, the qualitative research has been used for a better and more insightful understanding of the LCDPs. In order gain a better comprehension a preparational phase had to be completed. A preparational phase involves the developing a general skill within the LC area, sampling strategy for selecting relevant participants, offering a wide range of perspective and an interview guide development (Assarroudi et al. 2018). As a result of that, an interview guideline had to be created. Since this thesis involves two target groups, Citizen Developers and professional LCP experts, two interview guidelines had to be developed. The approach in the interviews involved a method, where a structure for the interview guideline had to be designed. The purpose of this structure was to (1) gain a clear understanding of what position was occupied and in which projects LCPs were involved and the reasoning for its deployment (2) elaborate on different experiences when developing and the capabilities needed (3) give a personal review of the gained experience and how it could have gone better. To ensure that everything can be answered in detail, some specific questions were openly asked, to get become different answers. Furthermore, the interview process included the addition of new questions or the adjustment of existing ones. This was the case, when a question was aimed at a specific direction, but was not answered properly and for that a new one had to be added. Another point was that adaption was required during the interviews, since the perspective of every interviewee was leaning towards a specific direction.

The initial plan for the interview structure was to include two Citizen Developers and one IT expert per LCP. However, after networking within the company, it became clear that some individuals were not always available, which resulted in having one fewer Citizen Developer from ServiceNow and no IT expert from OutSystems. The interviews were conducted in 7 days, whereas the first one was on July 11th, 2024, and the last one was on July 18th, 2024. All interviews had been done remotely via Microsoft Teams, since availability of each interview partner was difficult to comply and for that reason video-call interviews were the most suitable.

The selected experts were divided in to two categories. The first one involves individuals, which are actively developing applications within the LCDPs, and do not have an IT-role. This approach gives an opportunity to provide insights from non-IT persons, which can elaborate on their experience how the developed applications assist them in their daily business. The second experts are responsible for the

deployment and widely usage of LCPs within the organisation. They occupy a role, which aims to provide the Citizen Developers about their LCPs and help them understand how the use-case of a Citizen Developer can be applied in the LCDPs. For this matter a structure had to be designed to gain an equal opinion for each platform. A representative of each above-mentioned LCP has been interviewed. These experts are actively engaged in daily operations within the organisation, which allows to get insights into their practises.

The course of an interview began with the introduction of an expert. Questions involved in the introduction aimed to gain a specific understanding of the role and the activities of an individual. Following that, questions to gather detailed personal capabilities were asked. After gaining a clear understanding, the approach involved more platform-oriented questions for the discovery of the technical capabilities. At the end, organisational knowledge was questioned with the idea to find out how the organisations support its users. This approach was applied in both interview categories. It is important to note, that only one interview was conducted formal.

ID	*Company*	*Role*	*Date of Conduction*	*Duration of Interview*
CD01	Steel	Trainee	July 11, 2024	39 min
CD02	Steel	Development Engineer	July 12, 2024	38min
CD03	Steel	Digital Transformation Engineer	July 12, 2024	1h 4 min
CD04	Steel	Sales & Product Management	July 15, 2024	39 min
IT02	Steel	Platform Owner ServiceNow	July 16, 2024	39 min
CD05	Steel	Process Owner	July 17, 2024	39 min
IT03	Automotive	Platform Owner PowerApps	July 18, 2024	37 min

Table 3: Interviewees

3.3 Data Analysis

The interview phase began with a brief introduction of the thesis's topic. Afterwards participants were asked, if the interview could be recorded and were informed that the data processing document would be sent by e-mail after the interview. The video-call was recorded on a phone. The file was later transferred to a laptop, where the recorded audio could be transcribed on a special software. After this, a text document was generated by the transcript software. For the next phase Microsoft Word was used to have a better overview of the transcript. This phase involved correcting the text, in case a speaker was not correctly identified or when the dialogue involved overlapping. For this matter, the recorded audio had to be compared to text output so that the inconsistencies could be corrected. Given the correct transcription, the important findings were transferred to Microsoft Excel for exact evaluation of the interview.

The work of this thesis is based on a qualitative input from individuals, which involves unstructured data collected through expert interviews within a single organisation. For the data analysis the technique of

Mayring's (2000) qualitative analysis was selected due to its systematic and transparent approach, which allows rigorous handling of qualitative data. Mayring's (2000) method combines the strengths of both qualitative and quantitative analysis by utilizing a structured process to categorize and interpret textual data. This approach ensures that the analysis is both methodologically sound and contextually rich, providing deep insights into complex topics. Furthermore, the methods acknowledged for the qualitative content analysis are either the deductive or the inductive (Assarroudi et al. 2018; Mayring 2000).

Regarding this, the thesis has involved the inductive approach. It helps compressing the large volume of raw data collected from the interviews into a summary. Afterwards, links can be established between the objectives and the summary findings derived from the data. This can help to focus on the relevant findings of the research. Lastly, based on the findings theories can be build considering the data-driven capabilities for application development (Thomas 2003).

Furthermore, to have a better overview of Mayring's (2000) qualitative content analysis Figure 5 demonstrates, which are involved.

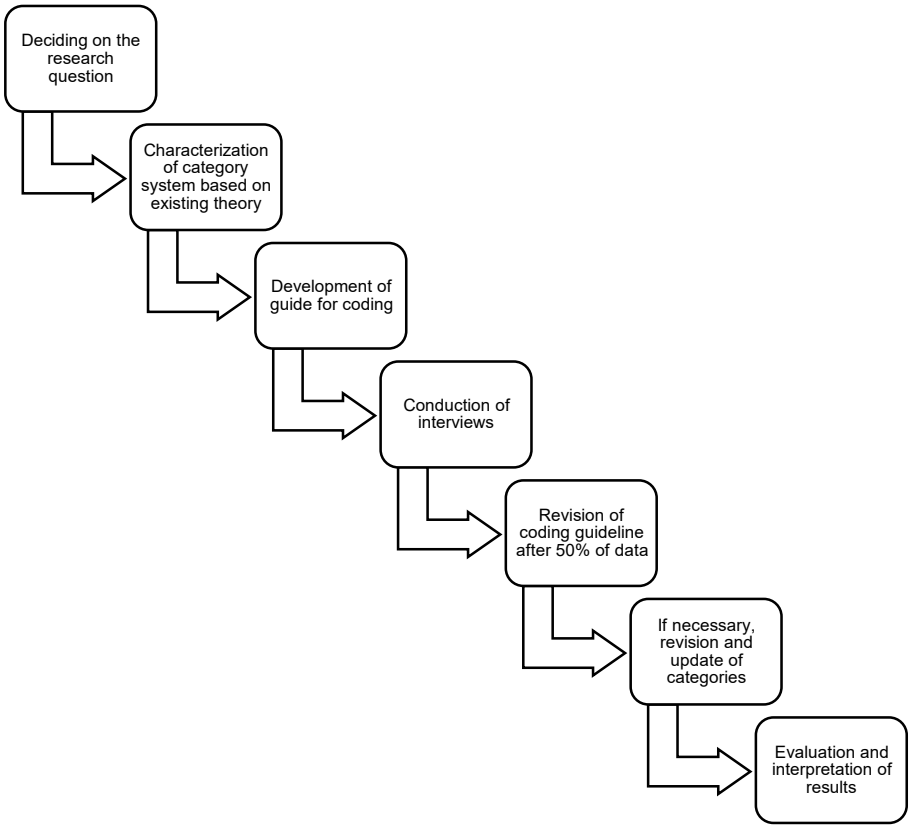

Figure 5: Qualitative content analysis based on Mayring (2000)

4 Results

In this chapter, the results conducted from the interviews are going to be presented. Given the structure of the theoretical background, a pattern was acknowledged involving Citizen Developers, LCPs and organisations within the data-driven capabilities for application development. For this reason, the structure of this chapter will be in the exact order as in Chapter 2.

4.1 Roles and Background

Due to the aim of this thesis to theoretically present and empirically research the data-driven capabilities in LCDPs, it is important to have a solid foundation of an individual's background. Hence, this chapter will present a detailed understanding of individuals role.

Beginning with the Citizen Developers, four of the five interviewed Citizen Developers were active in a non-IT department. In fact, to be more precise, a pattern was recognized between individuals. The daily business of one group is actively involving creating software solutions because their work is to drive the digital transformation of the company. The other group has a different focus area within the company and uses the software solutions to support their daily activities. Overall, it was evident that every Citizen Developer worked within a different area. One of the interviewees, working as a development engineer stated:

> *"At the moment, my focus is on the development of equipment applications, basically to assist in the area of development. I studied mechanical engineering and have been with the company about 16 years now."*
>
> - CD02

Furthermore, the previous experience regarding the software development was considered. It was apparent that, some of the interviewees had some experience with software, before transitioning to LC. This experience was based on educational background, self-taught skills and previous work.

Table 4 presents an overview over the individual's current roles and experience considering software development.

ID	***Role***	***IT-department?***	***Educational Background***	***Previous experience with Software Development?***	***Was Low-Code the first contact to Software Development?***
CD01	Trainee quality management	No	Non-IT Academic Degree	Basic experience through studies	Professionally, yes
CD02	Development Engineer	No	Non-IT Academic Degree	Has been programming professionally	No, has developed engineering simulations with software
CD03	Digital Transformation Engineer	No	Trained cutting machine operator	Self-taught experience within software programming	Professionally, yes

CD04	Sales and Project Manager	No	Non-IT Academic Degree	Has been consultant in an IT-company	No, has been involved in digitalization projects
CD05	Process Owner	Yes	Non-IT Academic Degree	Basic experience through studies	Professionally, yes

Table 4: Overview of Roles and Background

The other group of participants, consisting of IT experts for their corresponding LCPs, is responsible for the internal distribution and deployment for users. IT02 is a platform owner of ServiceNow. The individual is responsible for a broad range of activities but is mainly involved in the acquisition of new cases and their deployment. The reason behind that is that the ServiceNow instance was freshly established. IT03 is a group leader and platform owner responsible for a whole department company-wide embracing PowerApps operations and setup. The department has several sub-departments, responsible for support, governance and development, global services and customer enablement within PowerApps. As displayed above, it is important to notice that IT03 is part of the overarching company.

4.2 Types of Projects and Applications

Another important point, before heading to the data-driven capabilities of the different entities involved, was to understand what applications were developed and for what purpose. The applications developed by the Citizen Developers were used only internally within the organisation. With the used applications, there were two types of applications (1) applications that are used for processes company-wide (2) applications that are used to strengthen business departments aiming for the digital transformation. The first one involves business critical processes with the purpose that the company aims to high automated processes and the second one is about single departments, which utilize LCPs aiming for work facilitation that enables automation. All individuals agree that the LCAs are required to create automation within an area. One of the main reasons for this, are that previous daily work was mostly done on files like Excel, which are used to document data. Usually, there is not one file for all purposes, but many files serving different purposes, which makes the whole process even more complex to maintain. The Citizen Developers highlight this problematic, stating the following:

> *"At the moment, our projects are still working with, I think, up to 45 different Word, Excel and PowerPoint templates in addition to their normal systems, where they write their code, where they do their tests and so on. They work in all these documents and the documents are basically just dead data […]"*
>
> - CD05

> *"[…] these products have several component tests in our department during their production cycle and, before I started working on this topic, these tests were processed via an Excel list that was manually filled in by a colleague in another department who controls the process and so the colleagues here simply checked the parts that were on this list."*
>
> - CD01

Due to this working method, the operational efficiency can be slowed down, having to manually edit these files and to maintain an overview of them. This also effects the correctness of data, which must be maintained appropriately, since the use of these files are business critical. Furthermore, it is noticeable that collaboration problems also existed, when files had to be shared, which led to inefficient control problems. For this reason, Citizen Developers opted the switch to LCP, where processes can be data

centric. This eases to display the data, but also generate more value from it, given that it was only saved in the existing documents (CD05). Additionally, in some cases Citizen Developers publish applications, which are developed for specific use-cases. In such cases the Citizen Developer has the full responsibility for the development and maintenance of an application. To share responsibility and involve other users, Citizen Developers consciously decided in favour of LC, rather than having a hard-coded solution, so that the simplicity of LCDPs can actively involve different users (CD01).

Another reason for the switch to LC environment was, when one was not satisfied with the previous solution of an application, which was outsourced and hardcoded. This was the case, when CD04 relied on an external company for their application but the company has gone bankrupt. This has left the department with nothing, given that the external company did not provide them with the source code of the application, forcing them to search for an alternative. This has led to the change to a LC environment, giving the possibilities to rely on their own employees with developing.

The employees had mixed opinions about the transparency regarding the decision to make on which LCP to develop. Given the various LCPs, which are displayed in the Magic Quadrant, the employees had a decent pool of platforms to select from, with PowerApps, OutSystems and ServiceNow being the most widespread. Furthermore, one of the participants mentioned that at the beginning of their project, there was not even an instance within the organisation where they could present their use case, leading them to search for a solution within the overarching entity. After a consultation about different LCPs available, it was recommended that OutSystems is a suitable solution (CD05). Citizen Developers using PowerApps have based their decision due to the compatibility, which allows an easier usage and connection to tools like SharePoint. CD02 has not evaluated the available options of platforms and has based the decision to deploy OutSystems on shared experience with colleagues and the organisational initiative not comparing it to other platforms. CD04 has decided to firstly use ServiceNow for the project but plans to do some pilot projects on PowerApps and OutSystems.

Furthermore, it was noticeable that applications are used for workflow automation, but also for building applications intuitive for users, where different data sources are integrated and replaced. Given that many employees were manually filling lists before the LCP deployment, Citizen Developer aim with their project to minimize or even to replace these processes. As shown above, CD05 has started the project to allow users organisation-wide to have a central repository for their data. Since the individual is a process owner, the activities involving the project are regarding the quality of the processes. The application's development is done by an external department of the overarching organisation. The reason for that is that the Citizen Developer is not having the software development competence. However, this project aims to improve the operability of the employees from many departments by automating the coverage of important topics in terms of quality, compliance and legal within the development of the products.

CD01 also has developed an automatic test control with PowerAutomate within the department, so that data-driven logics can be applied for the prioritization of processes. For its creation a flow was created that integrates data from different sources and displays it on SharePoints. The created SharePoints are serving as the task planning tool for the employees, automatically entering parts that need inspection, calculating inspection dates and incorporating various status fields that trigger automated emails and activations. CD03 has also developed single use-cases for a better efficiency and digitization among the employees, enabling them to work smarter. It was evident that the use-cases within PowerApps were very data-centric, since most of the applications involved large amounts of data. For this reason, they have utilized the whole Microsoft infrastructure to work and deploy solutions easier.

Furthermore, given the engineering position of CD02, the Citizen Developer has deployed the OutSystems application to create a workflow for the different simulations involved in the product engineering process. This can not only centralize data, but also simplifies the initial process. Furthermore, CD05 was also relying on the workflow, but the primary use was to have a central repository for the product data, enabling users from different departments accessing it, to establish a central application, which covers the quality and compliance of the product data. For this reason, OutSystems has been used to maintain long-term applications, by providing them with a centralised repository.

Similarly, CD04, using ServiceNow, stated that the goal of the developed use case is to achieve workflow automation with a centralized data repository. Furthermore, IT02 elaborated on ServiceNow future use cases, stating that with the use of the LCP, the aim is to improve organisation-wide operational efficiency, particularly in departments not currently utilizing IT capabilities.

Table 5 presents an overview of the presented use-cases and applications.

ID	*Reason for LC switch*	*LCP used*	*Evaluation of LCPs*	*Use-Case*	*Goal of Application*
CD01	Reduce the dependency on manual filling of Excel by colleagues	PowerApps	Has been assigned with PowerPlaftorm since employment	Automate process of entering data	Enable data-driven logic to prioritize decisions
CD02	For simple development of applications	OutSystems	Has not evaluated, due to positive experience from colleagues and advertising by the company	Automate engineering simulations	Workflow automation and overall representation
CD03	Increase scalability of projects and reduce dependency of paper filling work	PowerApps	Has evaluated OutSystems, but prefers PowerApps due to its high scalability	Digitizing different manufacturing processes regarding logistic	Making processes data-centric
CD04	No dependency on external companies	ServiceNow	Has evaluated different LCPs and plans to create pivot projects on OutSystems and PowerApps	Document the lifecycle of an entire sales process and its information	Central data repository and workflow automation
CD05	Reduce dependency on Excel, Word, PowerPoint etc.	OutSystems	Had a consultation about the different LCPs	Document and maintain of product development processes	Clear documentation within the given regulations and add value to data

Table 5: Overview of Use-Cases within LCDPs

4.3 Data-Driven Capabilities

The main foundation of this thesis is to present the data-driven capabilities within LCDPs and how they are being used for application development. For this reason, the following results will provide an overview of how data-driven capabilities were utilized from a practical perspective. Firstly, the Microsoft users have made clear that their applications were cloud-native, meaning that every capability within the LCP can be accessed on-premises. Given the importance of the infrastructure, PowerApps users have noticed that Microsoft provides a whole ecosystem of different applications, which have different purposes. For this reason, data capabilities could be accessed more easily and provide a different storage for every entity involved in the application. CD03 provides an example stating:

> *"With the Dataverse, we have incredible possibilities that Microsoft provides us with, where we are simply travelling in the Microsoft universe. [...] The tables, etc., are all available in the Dataverse. This makes it more compatible with all applications. You can get data from anywhere."*
>
> - CD03

This statement underlines the importance of having the possibility to extend data. Furthermore, since every application from Microsoft has its own purpose, it is possible to experience different nuances of given data. To elaborate on that, the PowerApps participants, have noticed that Dataverse provides the data storage, which can be additionally integrated on PowerBI, where data can be visualized and analysed. This makes it possible to utilize data-driven capabilities since, as it has been deployed and used, it has gained value, allowing users to draw more conclusions from it. CD01 explained his experience by saying that this was essential for his application, which aim was to prioritize various decisions based on the data.

Another key aspect is the preparation of data. This process involves the used data sources and its integration. In all the cases the individuals have been able to integrate from homogeneous and heterogeneous sources and furthermore have the possibility to create an internal model within the used LCPs. Practically speaking, this was important for the applications since this functionality allowed rare data to become data-driven since certain data source combinations could be applied. After the integration, participants have stated that data must be modelled for the establishment of relationships between the tables. Furthermore, it was recognised that databases could also be integrated into the LCPs. This has allowed the individuals to base their applications on existing data and excluded the data modelling step, which was applied automatically, since the databases already consisted of the data model. This option was commonly used among the participants, who noticed that most of the business-critical data was already stored in and retrieved from large databases. This approach allowed them to avoid the time-consuming duplication of the existing database.

Following successful integration and a well-defined structure, participants employed workflows to control the flow of the data. With the help of this LCP functionality, users were able to incorporate a particular logic into their application, making it data-driven by setting off predetermined actions. This has been an important component for the applications that have been created because, when the action criteria are met, activities that take place after deployment add value to the data. CD04 developing with ServiceNow has provided with the following example:

> *"And so, we need a database where this is documented, there is the product manager, the sales engineer, the engineering person who are coordinating and documenting all this information of the product and being able to have a workflow where they are passing on the information, where they are also being able to track and trace the information over a period."*
>
> - CD04

The last and foremost important part of the data-driven capabilities within LCDPs is the use of APIs. This functionality has been used or is planned to be used in every interviewed cases. The connection via APIs can be established using the pre-defined data connectors from the LCPs or a custom connection. The pre-defined data connectors are already established connection to a specific external system. Although the data connectors offered by the LCPs in this case may differ, many of the connectors are still the same, since organisational criteria must be met to easily establish connections to core systems. In the interviews, it was shown that the connection to external data is easily established by choosing a specific data connector from a drop-down menu. Citizen Developers had a positive experience considering the pre-defined data connectors describing them as easy and fast to use and deploy. Furthermore, a custom API connection can also be established given the experience from CD02. Custom connections are required for specific external systems, that are not displayed on the pre-defined list. A custom connection was established only in CD02's case. The individual has described the custom connection as following:

> *„ […] with OutSystems, you simply define these REST API calls. I mean, the API definition is given by the provider, i.e. the external application, and this is then simply integrated accordingly in the OutSystems. "*
>
> - CD02

In the case of data connectors via APIs the applications are even more scalable considering the data. The reason behind that is that with different data sources the application is given a foundation on which it can be built on. Given the existing foundation, it can be furtherly scale further, given that important data from external services is required. This provides the developed application, which has already been data-driven with a further data dimension. Moreover, CD02 stated that through the API connection, data could also be overwritten in the external software. In this case, data is not only managed within the dimensions of the application but also extends beyond the scope of the LCP.

Having presented all the data-driven capabilities within LCDPs, a further capability has been identified that can be enabled to scale the application and its data-driven capabilities. However, this capability involves programming knowledge. Programming languages are not playing a crucial role in the application development, but they provide to customise a certain solution. This capability is not affecting directly the data-driven capabilities but provides for a better configuration of the UI of the application.

A further additional capability has been presented by the PowerApps users, which is yet to be introduced in the future is called the Copilot. IT03 describes this function as an AI functionality, which is a chatbot. Considering this topic IT03 states the following:

> *"You have a chatbot that can probably answer basic questions. You can add more information to it and then say when a specific event happens, start an automation flow. You can then train it with data."*
>
> - IT03

Given the presented data-driven capabilities within LCDPs, the participants have provided with some success factors. The PowerApps users have said that their key success factor is that LCPs is being used within a whole ecosystem. This has enabled an easy and sustainable connectivity within the environment enabling data flow, which can be simply distributed. Moreover, it has been described that data can not only be spread and used, but furthermore it gives it more value due to the different applications that can be used within the environment. Another success factor identified is the data accuracy. This has a significant meaning since the data is the foundation of an application. This is described as important because it is necessary to have a successful preparational data phase in order to enable data-

driven capabilities. Since data cannot always be modified and corrected within LCPs, its accuracy and correctness are crucial.

Table 6 provides an overview for the full understanding of the presented data-driven and additional capabilities.

Preparation of data-driven capabilities	***Data-Driven Enablement***	***Success Factors for Data-Driven Enablement***	***Additional Capabilities***
Integration of data by homogeneous or heterogeneous sources	Workflow creation for distribution of data and its drive	Data accuracy and low error rate	Use of programming languages for custom configuration and higher scalability
Maintenance of data correctness	Use of external services (e.g. APIs or databases) to extend the scope of application via data connectors	Data safety	Chatbot functionality for action declaration within LCP environment
Data modelling	Provide a whole ecosystem for added data value	Wide-spread use of data	

Table 6: Overview of Data-Driven and Additional Capabilities

4.4 Technical Capabilities of LCP

The technical capabilities of the examined LCPs encompass features that enable data-driven processes for the application development in LCDPs. While the core technical capabilities of LCPs are generally consistent across platforms, the implementation and usage may vary by vendor. Additionally, it is evident that because Microsoft provides a comprehensive ecosystem that the organisation utilizes, PowerApps offers a broader range of environments and capabilities, which enhance the interoperability of the LCP. For this matter, the gained experience from the Citizen Developers and the knowledge of the IT-experts will be presented.

Firstly, there is agreement from all the participants that the used LCPs provide users with an user-friendly interface, which is easy to develop. Furthermore, users have stated that the workflow automation option is another reason for using LCPs. This enables them to define and maintain the business logic behind of an application.

Data and its usage play the key role for all the participants. All LCPs offer two possibilities for data usage during the pre-development phase of an application. First, there is the option to create data entities within the environment of an application. This process involves declaring different tables, add attributes and set primary keys. Afterward, the data entities must be modelled, which includes the establishment of relationships. This process is like creating a traditional database, except that it occurs within the LCP environment.

Furthermore, data can be integrated externally on all LCPs. This step requires users to decide, if the data sources should be added from heterogeneous or homogeneous sources. Participants have utilized both options, noting that the choice depends on the specific application and the volume of data involved. Logically, for applications that aim to handle large amounts of data or require complex data interactions, the selection data sources become even more critical. The results have shown a widespread use of heterogeneous data sources, with Excel sheets being among the most integrated. This is because many applications are designed for departments that use Excel sheets daily, such as sales, engineering, and production. There is also a reliance on existing databases, which must be integrated. Since these databases often contain vast amounts of data, manually recreating them in the LCP environment would

require a lot of time effort. Therefore, integration is used to replace the need for manual creation. However, some Citizen Developers noted that data must be correctly integrated, which is not always the case. Furthermore, LCPs do not provide fully automatic detection of data inaccuracies. CD01 and CD04 shared their opinions about this topic saying the following:

> *"You can customise your data in Power Automate. Let's say, simple things. I have a float, but I want it to be an integer. You can do things like that. You have spaces, you want to trim something. Power Automate offers you these functions. But if you get completely corrupted data, I don't think you'll get very far in Power Automate. If bizarre data types occur or errors or null values are passed, Power Automate is no longer suitable for dealing with them."*
>
> - C01

> *"If you don't make the changes before uploading or before connecting, it does not correct or identify the errors."*
>
> - CD04

This could lead to incorrect use of data, which can negatively affect the application, if data is integrated improperly. It was noticed that Citizen Developers must manually check and test data before it is integrated to sustain a consistent utilization. Moreover, after the integration of the data, sometimes there is the need to modify specifics entities, if additional information is required to be added. The OutSystems participant elaborated, stating:

> *"One difficulty I can think of with the OutSystems platform is that it is not easy, for example, to rename attributes, i.e. tables or column names. This is a bit more complicated with OutSystems. You can't simply rename a table without further ado […]"*
>
> - CD02

The reason for this is that different versions of an applications can be published. To ensure that applications are used correctly, OutSystems does not allow for users to modify their data. This is because when previous versions are published and new data has been added, data integrity will not be maintained.

Another key capability that has been applied in almost every application is the API usage. Different experiences were made using APIs within LCDP. It was made clear that not every Citizen Developer has the expertise to utilize APIs and has reached for experts to establish the connection. Furthermore, there connections via APIs can be predefined and custom. In the case of PowerApps, there is a high connector pool, since the LCP is part of an entire ecosystem. Furthermore, custom connectors can also be established.

IT03 explains the API use within PowerPlatform in the following way:

> *"Power Platform has a concept, which is an environment, and for each environment you can exercise governance over which connectors are permitted. There are two and a half thousand and by default almost all of them are blocked, you can set up a connector to Outlook, to SharePoint. Or you can also say, I allow web service calls to cloud services, or I don't allow Dropbox in the cloud, for example, but I do allow OneDrive for Business. You can also say, I don't want a ready-made connector, I want a connector of the HTTPS endpoint, then you can call any web service."*
>
> - IT03

This statement underlines the different capabilities of with connectors regarding APIs. From a practical perspective, CD03 confirms that there are several predefined data connectors, which can be used easily by dropdown. For this matter, the seamless integration has worked for Graph API, which has been used to retrieve and filter data. Furthermore, CD03 has stated that there has been a difficulty considering

external connectors or third-party connectors. In the case of the individual, a connection for the business processes, where all the production data is recorded. An important point to consider is that this specific external software covers and records every data considering the company, when it comes to production, material data etc. Regarding this, CD03 has established a connection to the external software and could retrieve data, but not overwrite it from PowerApps to the external software. This is being done for security purposes to protect the data.

In the case of OutSystems there is also a difference between predefined connectors and custom connectors. The difference to PowerApps is that OutSystems is not a complete ecosystem like Microsoft offering several products and easily integrating them. However, it was noticed that some of the software, which are widespread in the organisation and often used have well documented information to establish the connection via API. In the application of CD05, there is a connection to the user management system of the organisation, which is a different software. However, the user management system counts as a predefined connector and is easy to integrate. It is used to retrieve data from the application, to establish a creator of a specific document within the application. Additionally, CD05 noticed that in the future they plan to use the external third-party connector, which was mentioned above, but can see difficulties with the use. In the other OutSystems application developed by CD02, an external connector was used, which was also a third-party connector. The connection is to a further external software, which has not been mentioned yet. The external software is used for product data management and the aim to document results. Since the data connector was not predefined the REST-API Calls needed to be created and defined by the provider of the external software. The use of this API connection aims being to retrieve and overwrite data. Furthermore, CD02 mentioned that at the beginning there were some difficulties to overwrite due to security issues, which were handled afterwards, since the external software is not business critical. Moreover, CD02 stated that the organisation had to approve the API connection, but topics concerning the security are took over by the external vendor.

Since ServiceNow is freshly established in the organisation, a practical result cannot be presented, because the participants have not reached that stage yet. However, IT02 points out the importance of APIs, which will be needed in the future but not on single, small application, but instead organisation widespread projects, which are currently ongoing. Likewise, CD04 has not reached the stage of API connection, but has clear how it is going to be implemented. Given that he works in the sales division, microservices to get different prices will be important in this case. Additionally, the Citizen Developer sees possible restrictions regarding third-party connectors.

Another important point is the environment, which the vendors of the LCPs offer. PowerApps offers the biggest variety of single and different environments, which an individual can connect. IT03 provides with an example with the interaction of the environments, stating the following:

> *“But if you now want to read out the sales report, integrate it into the Dataverse, run an AI algorithm over it and create a Power BI report, everything is low code [...] What is perhaps still strong with Power platforms is the automation capabilities [...] the special thing is that the application itself runs in the cloud, but you can deploy gateways in different networks and then you can also access different systems in the data centre from the cloud.”*
>
> - IT03

Furthermore, IT03 notices that this interoperability with aim to automate different processes can be very complex and hard to maintain. For this reason, the individual underlines the importance of the training courses and states that the PowerApps applications should not be only developed on the side, to support business operations and can be very powerful for Citizen Developers, if they can fully utilize them.

OutSystems user CD02 also states, that there are different environments for developing, testing and production. However, it is noticed that these do not provide variety of capabilities like PowerApps, but rather different environments to configure and developer for the users.

Table 7 presents a comparison of the different LCPs.

	PowerApps	***OutSystems***	***ServiceNow***
User-Friendly UI	Yes	Yes	Yes
Workflow creation	Yes	Yes	Yes
Data Integration	Can combine different data sources	Can combine different data sources	Can combine different data sources
Data Connectors	Offers more than 200 predefined connects and custom connectors	Offers predefined connectors and custom connectors	Offers predefined connectors and custom connectors

Table 7: Comparison of LCPs

4.5 Capabilities of Citizen Developers

One of the main goals of this thesis is to gather more and specific information about the profile of a Citizen Developer. Since a Citizen Developer plays a crucial role to develop applications, it is even more important to present the skillset of an individual to enable data-driven capabilities for application development in LCDP. For a more understandable result, firstly the used skills from the Citizen Developers will be presented. The result of the interview has shown that a Citizen Developer and its potential can vary, based on the previous background. The fact that, not all the interviewed participants, which are developing within a LCDPs did have software development background led to a different development experience.

The interviewees have varying levels of experience in software development. Since there were several individuals with an academic degree, most of their software experience are based on the educational foundations, which have provided them with some knowledge to develop but was not a crucial factor. One of them even had experience in the area of artificial intelligence having developed machine learning algorithms for supply chain solutions. The minority of the Citizen Developers had developed programming skills, because of their personal willingness, but furthermore to be capable to scale their application, if programming skills are needed.

However, it was noticeable that every Citizen Developer using LCPs had a specific goal to achieve with the developed applications beforehand, rather than deploying their LCP and then finding a use case for it afterward. This played an important role in the required skillset, as most of them knew what to expect and how to approach it. This allowed them to acquire, or further develop, the necessary expertise if they had not already done so. Every LCP provides access to a training course, which some of the participants have completed, thereby enhancing their capabilities.

There were mixed opinions regarding the required capabilities. The difference in opinion lies in the level of skills needed. Some participants who maintain or plan to maintain complex applications believe that more advanced skills are necessary to actively develop applications. However, the other group of participants could maintain an application with simple functionalities like drag-and-drop and some understanding of data. Given that, an important difference in the skillset between beneficial and required capabilities was recognized.

Beginning with the required skills, every of the Citizen Developers has used databases. It is evident that for the participants the foundation of an application is data. This leads to one of the most mentioned requirements, which is data understanding. One of the OutSystems user stated the following:

> *"[…] And you should have a certain understanding of how the data lies in the background, how it is interrelated and what data you need […]"*
>
> - CD05

Given that statement, it is apparent that knowing what data is used is important. All of the interviewees knew what type of data is going to be integrated into their application, since they are experts in their department. To elaborate on that, CD02 is a developer engineer, that knows what engineering data to expect, what type of data is used and how the data is related. Furthermore, in OutSystems the development of an application begins with data modelling, when data is not integrated, which leads to another recognized skill added to the profile. Furthermore, in other cases, such as PowerApps, data statements and understanding the logic behind functions are required. CD shares his experience, by revealing the following:

> *"The platform tries to explain the logic of a code to you visually, where otherwise you would say, I make an if clause, I make a when condition. […] If you want to capture the data, you can write an SQL query to get the data into the model already prepared. I personally use this a lot for my queries."*
>
> - CD01

This statement highlights the importance of having to understand certain operations to work efficiently, which leads to a better development experience. Moreover, since most of the participants have worked with databases, it is visible that classical database operations are very required in a LCDP. Some stated, that data understanding is not only required, because it is the foundation of an application, but to ensure the data security. To elaborate on that, wrong use of data can affect external systems connected to LCPs, which can be critical given that in a manufacturing company data is crucial, especially for product engineering and development.

Another important point about the required skills is the conceptual understanding. This point includes the conceptual understanding of the entities involved in a LCDP, which enables the development. To underline this skill, the following was said:

> *"I don't think you can get started without any understanding. I know that Low-Code means that you don't need any development experience at all, and I don't think that's quite right. You need to have a basic understanding of how an architecture, or a database is structured, how they are connected, how the keys are connected, where I need a parameter, where I need input variables, so a certain basic understanding of software development is required"*
>
> - CD05

It is evident that the conceptual understanding of the architecture leads to an enhanced comprehension of how the different components work. Since most of the participants knew beforehand what applications, they were going to use and how they would be used, their conceptual understanding was initially established, but it deepened further with continued development.

Furthermore, it was clear that the principles of agile development were applied throughout the projects, facilitating flexibility, collaboration, and iterative progress. This was not the case with every application developed, because some Citizen Developers like CD01 and CD02 using PowerApps were not involved in only one project but have been part of several smaller use-cases that improve the work of different departments. Projects that have developed one application, were aware that the applications must be continuously developed and improved. That was the situation with CD05, who is also part of the software

developer department, where the agile development is required. However, CD02 and CD04, which are not part of an IT-department, have acknowledged that agile practise is one of the key factors to have an iterative advancement.

Programming is a specific skill that has accompanied some of the Citizen Developers but not all of them. For this reason, programming skills are a beneficial, though not always required, but they can significantly expand the capabilities of an application. Some of them had even had previous experience in programming, given that CD04 has been developing machine learning algorithms. PowerApps users have programmed, given some specific cases, where a clear automation is needed. Furthermore, CD04 has mentioned that in their application programming skills will be needed, but they have not reached that stage of the development yet. CD05 sees having programming skills beneficial and said that professional software developers would have a much easier experience developing and scaling an application. However, CD05 itself has not developed programming skills, but aims to learn more about them in the future to eventually equip the applications with more features. Additionally, another OutSystems developer has developed its application with some programming, having to use JavaScript so that the functionalities can be expanded (CD02).

Based on the results from the Citizen Developers, there is one more requirement to consider: soft skills. It was evident that some of the participants have developed certain hard skills over time, but to achieve this a willingness to engage in a self-taught process was necessary. CD03 shares his experience of starting as a trained cutting mechanic with no academic background or prior knowledge in LC. Through self-learning, the person has acquired many skills and an understanding that enabled the development in the PowerPlatform. Furthermore, the opinions of the IT-experts are also valuable, as they can provide a clearer picture of the required skillset, given that these individuals are the ones who enable the Citizen Developers to succeed. One of the main points regarding the skillset of an individual is explained by a ServiceNow Platform Owner as follows:

> *"[…] personally, I am a fan of interpreting Citizen Developer in such a way that I have a baseline that has been built up and in the first step I use this Citizen Developer to make modifications, to make scaling, but not to develop that initially. [...] my current experience is that in order to perhaps further develop the Citizen Developers, you have to see how motivated they are, whether they take a liking to going deeper into it […]"*
>
> - IT02

This statement supports CD03's assertion that the process of self-learning enables a Citizen Developer. Furthermore, it is made clear that the enabling process should be initiated both by the user and by the higher authority providing the LCP. Moreover, this underlines that the motivation in the pre-development phase is one of the key success factors for being able to develop. Thereby, the initially showed motivation provides a foundation on which one can build and further develop certain technical skills to thrive and grow. The other IT-expert from PowerApps agrees on the previous statement and added:

> *Ideally, the skillset of the Citizen Developer will definitely be promoted during the development phase and, above all, if he also acquires programming skills in this case, then it is actually quite powerful.*
>
> - IT03

Additionally, it was mentioned that there is an initiative encouraging Citizen Developers to collaborate with IT departments, which can further enhance their development experience by dividing tasks and integrating them into an unified application. Based on the results regarding the skills required for development by Citizen Developers, the IT-experts provided more insights. First, the pattern between required and beneficial skills is evident. Moreover, all IT-experts agree that their LCP is a powerful tool, which

can be used to develop scalable applications, and these required skills are there to ease the development. Furthermore, the skillset of an individual depends heavily on the complexity degree of an application. An example was provided using PowerApps, comparing an application that collects data and overwrites it on a SharePoint, which can be done with simple clicks without a deeper understanding, to an application that has multiple data sources and must connect many entities in the development environment, which would require a significant maintenance effort (IT03). Similarly, as the Citizen Developers stated, many applications require a deep understanding of data, but there is also the factor of understanding the LCP environment and its capabilities. This includes knowing which data connector is needed for a specific use case and how to combine different entities to fully utilize an application.

Regarding this point IT02 from ServiceNow provided the following insight:

> *"I think the difficulty will be to understand the whole data structure at the beginning. ServiceNow is structured in tables and then the whole thing is nested. One table is extended from another table, and you must understand the relationships between the tables […] I think, of course, the complexity is gradually increasing, and it also depends a little on the requirement, what effort, internal, external access, etc., is required."*
>
> - IT02

Lastly, IT-experts expect the Citizen Developers to have a role that involves adapting to changing capabilities. This is necessary, because a Citizen Developer requires a specific skillset, which does not involve high standards but does demand adaptable mindset to effectively react to and fully utilize existing and newly introduced features in LCPs. IT03 compares this skill to programming languages, by saying the following:

> *"With programming languages, the basic understanding is often the same, whereas with low-code platforms there can be major differences. How it is deployed, how the code is managed, how much code is written, how much UI is involved. You just need special power platform training to be able to do anything with power platforms, I would say."*
>
> - IT03

To gain a better comprehension, what skills are required from Citizen Developers in LCDPs, Table 8 presents the results.

IT-skills	***Accompanying skills***
Data understanding	Motivation
Database use	Adaptability
SQL statements	Data security awareness
Creation of data models	
Conceptual understanding of LCPs	
Conceptual understanding of APIs	
Agile development	
Programming knowledge, if extensions needed	

Table 8: Profile of Citizen Developer

4.6 Organisational Capabilities

Organisational capabilities play an important role setting up a safe LCPs environment for application development. In this section the different organisational results and how these affect the development will be presented. It is important to make clear how the LCP vendors are structured within the organisation. Experts from PowerApps and OutSystems have departments in the overarching company. The participant IT02 from ServiceNow is the central instance within the company, however there is a central instance in the overarching company. There are several measures, which the highest instance has to take, which are mostly regarding the Citizen Developers. The platform owners of the LCPs have an active role in this process, maintaining and ensuring that development can be achieved easily and effectively.

One of the most important measures are the training courses, which have to be provided to the Citizen Developers for acquiring specific skills, understanding the concepts of LCDPs and being able to connect the features into a functional application. A pattern was identified between mandatory and optional trainings of the examined LCPs. IT03 has explained that for PowerApps development, mandatory training must be completed. The training courses are structured in a belt system, which must be passed through. For each course completed, a different belt colour is awarded. CD01 notices the following:

> *"You have to go through a belt system, so you first do a Yellow Belt and then a Green Belt and only after the Green Belt are you allowed to do Power Apps."*
>
> - CD01

IT03 added that before the yellow and green belt, there is the white belt, which provides a theoretical overview, including the capabilities that can be used within the LCP. After acquiring the white belt, the training becomes use-case-oriented. The yellow belt involves actions that only affect the individual. IT03 provided with an example stating:

> *"My e-mail that comes into Outlook is stored in SharePoint. I get a reminder sent to me. I have an automation or a small app in my context. No big business impact, your use case."*
>
> - IT03

Furthermore, IT03 explained that there are also "we-cases", which involve publishing application for company-wide use. In these cases, training is mandatory, and a face-to-face webinar must be attended to ensure users to understand what to be aware of when publishing applications. To complete this training, the knowledge is tested by a quiz at the end. Additionally, this has not always been the case with the trainings. CD01 and CD03 said that as they began with the use of PowerApps these trainings have not been mandatory and have done their training on their own initiative.

OutSystems does not require mandatory training courses to begin with. CD02 has stated that they have jumped in into the development. To achieve this the individual had gathered information from the LCP corresponding team about the training course and a documentation. It was evident that OutSystems provides with beginners and advanced courses, but there was not a specific guideline, which had to be followed for developing. CD05 agrees that no mandatory courses must be completed and adds that they have attended hackathons hosted by OutSystems to expand their knowledge about the LCP.

IT02 the corresponding expert to ServiceNow has said that trainings are not mandatory but discourages a direct development approach. Furthermore, the IT-expert stated the following:

> *"I also see it as my responsibility to perhaps provide people with the training material, so ServiceNow already offers a lot in this regard, they also have their own training portal. As a company, we have credentials for this, which means that we can also enable people to carry out the training and I think*

that really makes sense, because I don't think it makes much sense to let someone configure the platform directly […]"

- IT02

Regarding the trainings IT02 encourages it, but it is also evident that since the ServiceNow department of the organisation is newly established, there is no clear structure to guide the training process. CD04 added a practical perspective, stating that after the consultation with ServiceNow from the overarching company, one practical training course was completed, where participants developed their own small application. Moreover, CD04 stated that after the first course, they realized that for their expected use-cases, additional modules were required, which had to be completed through further training to fully understand them. Since CD04 and his team are working on the development, the courses were completed in parallel. However, IT02 added that there is the possibility for ServiceNow customers to participate in the so-called 'Family Release,' an annual event that encourages users to integrate the newest capabilities of the LCPs into their applications. Furthermore, IT02 mentioned that ServiceNow provides an additional environment where isolated applications can be developed for personal use and training, with the option to receive feedback from IT experts afterward.

Another important point is role management in LCDPs from an organisational perspective. This is handled by the platform owners of the corresponding LCPs. It was evident that several roles can be assigned within an environment. The roles specify whether a user has a role that is active or one that allows them to read. Furthermore, licensing models are based on the role concepts. CD04 elaborates on the several roles in ServiceNow having stated:

"And at the same time, there are three kinds of roles in Service Now. There is a fulfiller role like an admin, which is a person who is basically the master user or power user. Then there is a second kind of role, which is someone who has limited capability and can work, can make some changes. And then there is the viewer or the requester. So, the admin has the capability to check the data as well as to correct if there is something missing."

- CD04

Furthermore, one of the main tasks from organisational perspective is to maintain secure applications. Thereby, applications must be checked before published. All LCPs have specifications regarding the security. IT03 describes that PowerApps have a whole concept regarding the application security. Firstly, there is the organisational security infrastructure. IT03 explained that ISO27001 is the security standard in the organisation, where certain regulation criteria has to be completed. This concerns the different connections established in applications, which must be encrypted. Moreover, before starting the development of an application, an application owner must be assigned, ensuring that there is a responsible individual who will oversee the project, manage the resources, and ensure that the application aligns with the organisational goals and requirements. IT03 elaborated by stating the following:

"There are many requirements for application operation. And if every department can say, I'll develop an application and share it with my supplier, and someone develops something and is gone after three months. In the end, it's a business-critical application that has been developed and there are no support processes, there are problems."

- IT03

A further point regarding the direct insurance of application security is involving the use-case evaluation. In PowerApps the use-case must be evaluated, requiring Citizen Developers to answer 15 questions concerning application details and usage. After the evaluation, if the application data is accessed organisation-wide, then Citizen Developers must present a security concept. However, the results have shown that the two Citizen Developers using PowerApps have not done this, because at the beginning

of their application development process this has not been required. Furthermore, CD04 has stated that the application was not checked by OutSystems experts, but from the Cyber Security team in the company. CD03 using ServiceNow has not provided with a response regarding security proof.

Support is another crucial aspect when considering organisational capabilities. IT experts have provided insights into how they assist LCP users. One of the primary support sources has been Microsoft Teams channels, which allow users to post their issues and receive responses from other who have experienced similar situations. IT03 has stated that for PowerApps there are additional websites internally created for organisational use to provide further information. OutSystems has also initiated an additional source, holding weekly meetings where users can participate and ask questions. IT02 offers direct support, as the individual is responsible for acquisition and deployment of use-cases, but is not the central instance for primary support, which is handled by the overarching company. The results have shown that PowerApps and OutSystems provide personal and direct support for application development, however this service incurs additional charges. Given organisational support, Citizen Developers have mixed opinions about the quantity of the provided assistance. Citizen Developers using PowerApps have stated that the structure of the support levels is irritating and have struggled to reach the support they have wished for. This concern involves the direct support when Citizen Developers needed concrete technical assistance. Furthermore, OutSystems users have been content with the support they have received. Both Citizen Developers have received direct support, when specific technical assistance has been needed and additionally stated that the current state support has been helpful. CD04 had also positive experience about his support but is concerned about the lack of people involved in the process of assisting. Additionally, the individual expressed that he wishes a support in terms of hand-holding, meaning that ServiceNow experts should take operatively part of the application development for faster application release.

Table 9 provides an overview of the different organisational capabilities that directly affect the data-driven capabilities in LCAD.

LCPs	***Mandatory training?***	***Art of training***	***Security***	***Support***
PowerApps	Yes	Belt system	Detailed security involving evaluation of use-case criticality	Several channels and extended support
OutSystems	No	Basic and advanced training	Cyber security evaluation on roles and authentication	Team channel and weekly Low-Code talks
ServiceNow	No	Theoretical and practical training, with addition of isolated environment for training	Security is handled at overarching company	Support at overarching company

Table 9: Organisational Capabilities ensuring Data-Driven Capabilities

5 Discussion

Having presented the results on the data-driven capabilities in LCAP, this chapter aims to discuss how these capabilities were utilized in application development within LCDPs. To begin, a recap of the data-driven capabilities in the context of LCAPs will be provided. This will be followed up by the chapter "Implications for Theory," where the findings regarding data-driven capabilities are examined in relation to the existing theoretical literature. Subsequently, the implications for practice will be highlighted, focusing on how the leverage of data-driven capabilities can be applied in real-world scenarios. The limitations of the thesis will also be outlined. Finally, the sub-chapter on future work will be presented, pointing out various areas where future research should be conducted.

Data-driven capabilities are important for the whole application development process. Data-driven capabilities are enabled within the development using different combinations of provided technical capabilities. For their enablement firstly there is a preparational phase, which maintains the correctness of the data, to be target oriented in use-cases. After integrating data, the technical capabilities of the LCPs can be utilized combining data with certain operations. The key factors of enabling data-driven capabilities are the main capabilities of LCPs. Firstly, there is the UI, where data can be visually displayed and made accessible to users. To be able to enable data-driven capabilities, the processes adding more value to data must be involved. For this reason, workflows can be used for the distribution of data by certain actions, which enable the data flow. Moreover, the data amount can be extended with the use of data-connectors. This step allows for users to get out of scope data and combining it with existing data, which was integrated. It has been shown that this was a key factor for Citizen Developers, since the whole organisation relies on different documents and applications, where their data is stored. This has provided applications with a central data repository, where data is combined, but furthermore a new dimension where the combined data can be used for different purposes. Furthermore, the environment of LCPs plays a crucial role, giving users the possibility to extend data-driven capabilities, allowing them to use different applications and environments (e.g. PowerApps), where data value can be added.

Having presented the research model, which was based on the existing literature, the main roles for data-driven application development have been identified. Firstly, there are the technical data capabilities of LCPs. These capabilities directly affect the application development, since an LCAP is based on the provided functionalities involving data. The findings have shown that every main capability of the LCPs has been utilized for the data-driven enablement. Furthermore, there is the role of the Citizen Developer. Having examined the results, it has been evident that every of the interviewed participants developing applications fits the concept of the Citizen Developer, not being part of an IT department and have domain expertise. However, for the enablement of data-driven capabilities a Citizen Developer is required to have knowledge about the data used within LCDPs but also the ability to manage the different functionalities of LCPs. Furthermore, it is noticeable that every individual has a different application to develop, because there are small and single use-cases that desire short-term goals and there are high-complex use-cases that desire the long-term impact of certain applications. The third identified role within LCAD is the organisation, which manages the activities of the above-mentioned roles. For this reason, organisations must firstly provide and set up the LCP environment within a company. Having that, an overview and transparency of the available LCPs must be sustained for employees to have the option to evaluate, which LCP is the most suitable for their use-case. Based on the findings it is clear, that the company has established this, since the results are based on three LCPs. Another point regarding organisational data capabilities, are to provide the wide data use, so that every data needed from a development perspective can be implemented in the application. This is concerning Third-Party APIs, which have been widely used among the participants, but in some aspects, there were restrictions for

data overwriting on external systems. It has been evident that organisations must enable users shifting to LCDPs for the enablement of data-driven capabilities. This requires different organisational measures such as training and providing support to the individuals to secure safety development. The company has not been able to fully achieve this, which was leading Citizen Developers to take matters into their own hands.

To summarize, data capabilities must be enabled within the given roles to enable their data-driven purpose. It is important that the given roles are affecting each other directly and the neglect of a role can impact the process within LCDPs aiming to enable data-driven capabilities.

5.1 Implications for Theory

The goal of this thesis is to investigate the data-driven enablement within LCAs, which is requiring to examine the different factors and their impact on the application development process within LCDPs. For this reason, 7 interviews were conducted from one company for achieving the different development perspectives of three different LCPs. Therefore, insights have been gained about the development process from the three roles involved and the required skillset for users to be able to enable data-driven applications.

Given the first step based on Sahay et al. (2020), it was noticeable that data modelling is a crucial step within the development process in order to give business logic to an application. However, data modelling was not always necessary since some Citizen Developers have utilized the database integration, which automatically generates a data model within the LCPs. Moreover, data can be integrated from simple documents like Excel sheets, which may contain no standardised data structure and for that data corrections had to be completed. For this integration it has been identified that data should be tested before it gets integrated, because this may negatively impact the foundation of the application. Nevertheless, PowerApps provides the manual data correction in the LCPs environment, which extends the functionality that integrated sources can be modified afterwards. However, this was not the case with OutSystems. The reason behind this is that the LCP provides an additional functionality of application versioning, which enables users to switch back to an older version and restricts them to modify the integrated data model.

Given that, Citizen Developers do not always have IT-capabilities (Carroll and Maher 2023), LCPs do not provide users with an automatic data correction, which can negatively impact the preparational phase of the enablement of data-driven capabilities. Considering the enablement process of the data-driven capabilities, there are two key factors that can contribute for the enablement process and are mentioned in the development of LCPs by Sahay et al.(2020). Firstly, the creation of workflows is necessary, which is directly affecting the data flow, having that it can automate the process. From a practical perspective this functionality was provided from all the LCP vendors. This was a decisive factor for users in switching to LC environments, as it enables the automation of certain data by providing the ability to manage it and also offering clear direction on where the data should be headed. Secondly, Third Party APIs were hugely important for the application development. Given the results, that Citizen Developers did not only aim for the simple development, but rather wanted to centralize and combine different data sources. For this reason, Third-Party Integration via APIs have been utilized to extend the simple data-capabilities within an application, enabling the access to data sources out of scope. This was particularly important for the case of the Citizen Developer from the sales department, allowing to involve competition data. For the API management, Citizen Developers have expressed positive feedback due to the simplicity of usability. This somehow contradicts with the initial claim stated in Overeem et al. (2021) that the management might get technical. In this matter most of the Citizen Developers expressed that

the integration has been easy to deploy. However, regarding this matter, the interoperability of LCPs has not always been maintained due to organisational restriction of core systems data. This has affected the possibility to overwrite data from the LCP and transfer it to the core system. Moreover, the lack of dissolution of the functionality has not affected the enablement of the data-driven capabilities within the application but it has limited their further use outside of its scope.

Regarding the Citizen Developer the different definitions of individuals for not having an IT-background, being non-technical and not needing programming knowledge and skills has been confirmed (Carroll and Maher 2023; Krejci et al. 2021). Given the results, Citizen Developers had different background and previous experience with software development, which is confirmed by the literature. However, the most important aspect of the pre-development phase is to be enabled from the organisation. Since starting off with LCA might be easy at first due to the visual configuration of UIs, there are some further capabilities, which are important and need to be obtained while developing application for the full utilization of data-driven capabilities. Considering the starting point of an application, which is about the integration of the data (Sahay et al. 2020), data understanding has to be maintained throughout the whole lifecycle of an application. Moreover, Citizen Developers must be familiar with the capabilities regarding APIs, which are crucial for extending the data scope. As the results have shown, some of the skills have not been acquired before switching to LCDPs but have been acquired within the development process. This shows that Citizen Developers must have clear in mind how and for what they want to use a specific application for a better upfront understanding which data-capabilities skills to expect when developing and how to enable them.

Proceeding with the skillset of the users presented by Binzer (2023), which were examined from job market posting. The skills are being presented as business-related and knowledge-related, which is not accurate, since these are superficial terms and business and knowledge perceptions can vary, based on the focus points of organisations. For this reason, it would be more suitable to cluster the skillset into soft skills and IT-related skills. Given the examination from Binzer (2023) there are two additional soft skills that have to be added. This includes the adaptability when developing with LCPs, because the platforms and their capabilities might differ. For this reason, when a use-case is being implemented on one LCP, but afterwards it has been recognized that other LCPs provide more capabilities regarding a specific matter, users should be able to remain adaptive within different LCP environments. The other soft skill is considering the motivation. As the results have shown, most of the Citizen Developers are domain experts having a business understanding, which they want to digitally transform. However, it was evident that the desire must begin with the organisational enablement and requires a certain personal motivation from domain experts to digitize processes including manual filling of documents. By achieving this, as it was presented in Carroll and Maher (2023), data-driven capabilities will be made more accessible and furthermore democratized leading to a sustainable digital transformation. The required IT-skills are difficult to generalize, since they are based on which LCP a user is developing and even what role does the user have within the LCDP. Having said that, there is a clear differentiation between applications that are developed for the enabling of the personal use or the organisation wide use. Applications involving huge amounts of data and different connections must be maintained by experienced Citizen Developers, which know how to handle them. Furthermore, the data has shown that the IT-skills are highly dependent on the use-case and its goal within LCDPs. Nevertheless, the results have shown that the principles of professional software development have been followed in some cases using agile development for the continuous development. Citizen Developers have noticed that implementation of applications can take months, and it is required to follow an iterative development, in order to accordingly react in the development, when applications are published.

Given that data-driven capabilities must be firstly provided, the organisational perspective gains more importance. Based on Prinz et al. (2024), it can be confirmed that social and technical systems play a crucial role in LCDPs. For the social systems, the results have shown that support of any kind must be provided to fully enable data-driven capabilities. This involves having training courses, where theoretical understanding is gained so that it can be practically implemented. Trainings are essentially building the ground foundations of users, providing them with a structural plan on how to approach certain tasks. Not having training and jumping into LCP can lead to a difficult situation, not knowing how to address specific capabilities. This is also reflected in the presented results discouraging the development without knowledge. When developing it is crucial to have a central instance, where difficulties regarding the development can be expressed. In this matter the organisation has not always provided a clear transparency, which led to a struggling development experience forcing users to search information online. To elaborate on that, having to search online for a solution to a problem, might also lead to a chaos within the application, since the approach online might not align with the encouraged approach of the organisation.

Furthermore, the technical systems have also been considered. Data-driven applications have been observed by the company and categorised according to the degree of complexity (Hintsch et al. 2021). It is required that the used data in an application must be assessed based on who is accessing it and how it can be secured. This also involves the custom connection via APIs, when external services are used, which is even more important given that confidential data is made available through the connection. For this reason, the organisation requires users to come up with a safety concept of their application, when it is accessed by the whole company. Furthermore, the Third-Party pre-defined data connectors have been provided, allowing Citizen Developers to extend their connectivity to other external systems and providing them with a new data source.

5.2 Implications for Practice

Additionally, major practical consequences could be drawn from the presented results. The goal of this sub-chapter is to provide a recommendation for certain aspects of LCDPs to fully enable data-driven applications.

First, in some cases of the participants, there has somehow been a desire for a higher scalability of their applications, which required programming knowledge, but this was not implemented, since it would be too time-consuming. This has been the case, with Citizen Developers which had pre-LCDPs programming knowledge. It has been evident, that in the other cases individuals were aware of this, but have not taken it into consideration, since they had to acquire programming skills. Additionally, this indirectly concerns data capabilities, since in some applications it was important to represent data accurately and for this customisation was required. However, although the literature suggests that users with minimal to no programming knowledge can develop applications, the results indicate that programming skills are beneficial. Therefore, an individual's programming knowledge will likely affect the outcome of an application.

This can be combined with a new feature that was introduced in the results and may gain importance in the future, given that it additionally would simplify the development. This considers the inclusion of AI components in the LCDPs. Although there has been some research in this area, particularly with the implementation of different aspects of AI, such as Machine Learning, there has not yet been a commercialized solution for organizational use within LCPs. However, this will change, since PowerApps interviewees have stated that an AI functionality has been developed and implemented for the private consumption but is yet to be deployed on PowerApps. This is regarding a chatbot, which will be able to

process different prompts from users, enabling an easier automation of the LCPs. The assistance might change the enablement of the data-driven capabilities, allowing the simple execution just by input. Given the presented results, this could be especially important for the development experience. Having that some participants are not willing to extend their application with programming functionalities, the chatbot functionality would make this capability accessible leading to a more powerful application. Furthermore, this would present an opportunity at the integration of data, where some Citizen Developers had to manually adjust data from their data source, providing them with time-saving experience. Moreover, the chatbots could provide new insights to data, when findings must be made, which could furthermore extend the usability of data-driven applications.

An important discovery was made on the social structures of LCDPs, especially about concepts of training and support. The development process for providing data-driven capabilities has been highly variable because of the absence of organizational consistency in these areas. On the one hand, not all LCPs offer users consistent instruction since there are no company-wide, required training sessions. Individuals have differing levels of knowledge, which forces many Citizen Developers and even some technical users to individually research information and get familiar with the possibilities of the platform. Due to the differences in training, there is a knowledge gap, making it so that some people with more expertise or access to resources can make better use of the platform's potential while others can find it difficult to use its fundamental features. Similarly, the support structures provided within organizations are inconsistent, further complicating the development process. Developers can occasionally rely on specialized teams or well-defined policies, to get them through difficult situations. However, in other instances, support is minimal or even non-existent, leaving users to troubleshoot problems on their own or rely on informal networks of colleagues for assistance. The development process is slowed down by this inconsistency, which also raises the possibility of mistakes or inefficiencies while deploying data-driven capabilities. The lack of standardization has implications for the overall application development withing LCDPs, where data-driven capabilities might not be fully realized. As a result, it becomes critical for businesses to understand how important it is to set up consistent support networks and training courses so that all users, regardless of background or experience level, can take full advantage of LCDPs' tremendous features. In the end, this standardization would improve the effectiveness and enabling of data-driven capabilities within LCDPs, assist in closing the knowledge gap, and guarantee a more uniform application development process throughout the company.

5.3 Limitations

Given the main work of this thesis, the purpose was to present the data-driven capabilities within LCDPs, but there were also some limitations, which are outlined below.

One limitation was the lack of detailed literature specifically addressing development with the selected LCPs, particularly in the case of ServiceNow. Due to the absence of academic and empiric research on ServiceNow’s capabilities, it was necessary to assume that ServiceNow might have features that were comparable to those of other LCPs. Because of this, a large portion of the research into ServiceNow’s capabilities was based on insights from interviews rather than supported by robust, pre-existing literature. Given that the work of this thesis is only based on the data-driven capabilities of the selected LCPs, this has not impacted it directly, but might have been important for a broader overview.

Furthermore, although there is some literature available on development with LCPs, such as PowerApps and OutSystems, these studies often lack comprehensive analysis of data-driven capabilities. The existing literature focuses on more abstract aspects of the platforms functionality rather than providing a

holistic view of how data-driven capabilities are enabled and leveraged across various LCPs. This has led to a challenge in understanding these capabilities within the context of LCDPs.

5.4 Future Work

An interesting approach for future work has been identified in the interview process. Given that, PowerApps will offer the functionality of the Copilot, which is an AI-driven chatbot, it would be particularly interesting to investigate its use within organisations. This functionality has the potential to simplify even more the development process even more, transforming it to a configuration-oriented task, where input is defined declarative. With Copilot, users might be able to develop certain functionalities simply by using natural language prompts, which could lower the technical barriers even further and expand the pool of Citizen Developers.

Another important area where future work might provide more insights into the development process is Agile development and how it is being utilized, given the findings in the literature. Investigating how agile methodologies are adapted and implemented within LCDPs could reveal valuable information about the flexibility and efficiency of the platforms and developed applications. Moreover, researching the difficulties and best practices related to managing agile projects in a low-code context would be helpful, particularly regarding striking a balance between speed and quality.

6 Conclusion

The research aimed to explore how data-driven capabilities enable application development within LCAPs. The guiding research questions was:

- How do Data-Driven capabilities enable Low-Code applications development?

To address this question, a comprehensive theoretical background was conducted, alongside qualitative research through expert interviews. In the theoretical framework the data-driven capabilities have been highlighted within LCAP and their impact on application development. It was identified that these capabilities are fundamental in facilitating the creation, integration, and management of data within applications, thereby empowering Citizen Developers to efficiently build and deploy applications. Furthermore, the literature has addressed the organisational aspect within LCDPs, involving social and technical systems, by categorizing them within the different roles involved. After reviewing the literature, it was evident that there are several data-capabilities enabled by the LCPs. Citizen Developers are required to have a specific skill, and the organisational aspect needs to enable the ease of use in the application development process in terms of accessible data and the utilization of its capabilities.

Since the qualitative research was chosen, it was important to gain the practical perspective of users, involved within LCDPs and for that reason only one company was analysed for the purpose of having one instance with different perspectives. Furthermore, to enhance the research with more knowledge and detail, participants from three LCPs – PowerApps, ServiceNow and OutSystems - were interviewed. Before conducting the interviews, two interview guidelines have been created to complete the interview with a clear approach. Moreover, this has supported the process for the revision, being able to easily categorise different topics. A different interview guideline was created for both Citizen Developers and IT experts, since the approach of the target groups was different.

Following the acquisition of the outcomes, it was important to gain a brief overview of the roles and backgrounds involved in LCAD to better understand the users involved. It became evident that all participants fit the concept of Citizen Developers, being domain experts who seek to digitize specific solutions using LCDPs in both the short and long term.

After gaining the results, the findings have revealed that data-driven capabilities, such as data modelling, integration with various data sources, and the ability to handle unstructured and structured data, are critical in supporting the application development process. These capabilities have been important, since data is the ground foundation in LCAD, which requires its handling. In this preparatory phase, PowerApps even allows data correction after integration, simplifying the development experience by enabling adjustments within the environment rather than correcting the data and having to reintegrate it afterward. On the opposite site, OutSystems had not allowed the data correction after it was integrated, due to application versioning reason. Moreover, the true power of data-driven capabilities has been enabled with the functionalities of workflow and the allowance for an application to use Third-Party APIs. These functionalities allow developers to automate processes and centralize disparate data sources, thus enhancing the overall functionality and scalability of applications. There was positive feedback about the API use, but many technical tasks accompany Citizen Developers throughout their development experience. The development process of an application is simple, but the rapid development has been impacted due to the several tasks, which had to be completed apart from the development. However, the research also uncovered challenges related to the interoperability of LCAPs with core organisational systems, which can limit the extent to which data-driven capabilities can be fully realized across the enterprise.

Since LCPs play a crucial role in providing the technical functionalities of the data-driven capabilities, it was clear that all capabilities had the same characteristics. All the examined LCPs had the main capabilities, which the literature has presented. However, one point which is still an open question is regarding the pre-defined data connectors. Participants have stated that they did not have problems with them, but the availability of certain data connectors may vary across LCPs.

The organisational capabilities providing the enablement of the individual skillset to develop data-driven capabilities has been partially part of the problem. Since the results have shown that some of the Citizen Developers have a different background, there is no standardised process for achieving a certain baseline pre-development. This is made clear by the difference between mandatory training and provided support. A further point, which is even more important to sustain is security regarding data and the applications. To achieve this, the company urges the help from ISO27001 to maintain a whole organisation security. The inclusion of individuals perspective is to examine the criticality of their use-cases by categorizing the application degree, if it is used organisation-wide or personal.

In conclusion, it can be confirmed that the popularity of LCDPs is rising, even in the industrial sector, enabling Citizen Developers to actively participate in application development, which they are using to climb the ladder toward digital transformation. Overall speaking, there is positive feedback from the whole use of LCDPs and its simplicity to enable data-driven capabilities. It is important to state, that LCPs offer the technicality, which can be enabled by Citizen Developers. However, with no clear organisational standardisation, application development process would struggle to enable the data-driven capabilities by Citizen Developers.

References

Al Alamin, M. A., Malakar, S., Uddin, G., Afroz, S., Haider, T. B., and Iqbal, A. 2021. "An Empirical Study of Developer Discussions on Low-Code Software Development Challenges," *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)*: IEEE, pp. 46-57.

Alasuutari, P. 1995. *Researching Culture: Qualitative Method and Cultural Studies*. Sage.

Alt, R., Leimeister, J. M., Priemuth, T., Sachse, S., Urbach, N., and Wunderlich, N. 2020. "Software-Defined Business: Implications for It Management," *Business & Information Systems Engineering* (62), pp. 609-621.

Arora, R., Ghosh, N., and Mondal, T. 2020. "Sagitec Software Studio (S3)-a Low Code Application Development Platform," *2020 International Conference on Industry 4.0 Technology (I4Tech)*: IEEE, pp. 13-17.

Aspers, P., and Corte, U. 2019. "What Is Qualitative in Qualitative Research," *Qualitative sociology* (42), pp. 139-160.

Aspers, P., and Corte, U. 2021. "What Is Qualitative in Research," *Qualitative Sociology*), pp. 1-10.

Assarroudi, A., Heshmati Nabavi, F., Armat, M. R., Ebadi, A., and Vaismoradi, M. 2018. "Directed Qualitative Content Analysis: The Description and Elaboration of Its Underpinning Methods and Data Analysis Process," *Journal of research in nursing* (23:1), pp. 42-55.

Binzer, B. 2023. "Low-Coders, No-Coders, and Citizen Developers in Demand: Examining Knowledge, Skills, and Abilities through a Job Market Analysis,").

Bock, A. C., and Frank, U. 2021a. "In Search of the Essence of Low-Code: An Exploratory Study of Seven Development Platforms," *2021 ACM/IEEE international conference on model driven engineering languages and systems companion (MODELS-C)*: IEEE, pp. 57-66.

Bock, A. C., and Frank, U. 2021b. "Low-Code Platform," *Business & Information Systems Engineering* (63), pp. 733-740.

Bostrom, R. P., and Heinen, J. S. 1977. "Mis Problems and Failures: A Socio-Technical Perspective. Part I: The Causes," *MIS quarterly*), pp. 17-32.

Breaux, T., and Moritz, J. 2021. "The 2021 Software Developer Shortage Is Coming," *Communications of the ACM* (64:7), pp. 39-41.

Bryman, A. 2008. "The End of the Paradigm Wars," *The SAGE handbook of social research methods*), pp. 13-25.

Carroll, N., and Maher, M. 2023. "How Shell Fueled Digital Transformation by Establishing Diy Software Development," *MIS Quarterly Executive* (22:2), pp. 99-127.

Chang, Y.-H., and Ko, C.-B. 2017. "A Study on the Design of Low-Code and No Code Platform for Mobile Application Development," *International journal of advanced smart convergence* (6:4), pp. 50-55.

Dhoke, P., and Lokulwar, P. 2023. "Evaluating the Impact of No-Code/Low-Code Backend Services on Api Development and Implementation: A Case Study Approach," *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*: IEEE, pp. 1-5.

Döringer, S. 2021. "'The Problem-Centred Expert Interview'. Combining Qualitative Interviewing Approaches for Investigating Implicit Expert Knowledge," *International journal of social research methodology* (24:3), pp. 265-278.

Flyvbjerg, B. 2006. "Five Misunderstandings About Case-Study Research," *Qualitative inquiry* (12:2), pp. 219-245.

Gürcan, F., and Taentzer, G. 2021. "Using Microsoft Powerapps, Mendix and Outsystems in Two Development Scenarios: An Experience Report," *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*: IEEE, pp. 67-72.

Hammersley, M. 2012. *What Is Qualitative Research?* Bloomsbury Academic.

Heuer, M., Kurtz, C., and Böhmann, T. 2022. "Towards a Governance of Low-Code Development Platforms Using the Example of Microsoft Powerplatform in a Multinational Company,").

Hintsch, J., Staegemann, D., Volk, M., and Turowski, K. 2021. "Low-Code Development Platform Usage: Towards Bringing Citizen Development and Enterprise It into Harmony,").

Hoogsteen, D., and Borgman, H. 2022. "Empower the Workforce, Empower the Company? Citizen Development Adoption,").

Horváth, B., Horváth, Á., and Wimmer, M. 2020. "Towards the Next Generation of Reactive Model Transformations on Low-Code Platforms: Three Research Lines," *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, pp. 1-10.

Hsu, C., Wang, T., and Lu, A. 2016. "The Impact of Iso 27001 Certification on Firm Performance," *2016 49th Hawaii International Conference on System Sciences (HICSS)*: IEEE, pp. 4842-4848.

Hyun, C. Y. 2019. "Design and Implementation of a Low-Code/No-Code System," *International journal of advanced smart convergence* (8:4), pp. 188-193.

Ihirwe, F., Di Ruscio, D., Mazzini, S., Pierini, P., and Pierantonio, A. 2020. "Low-Code Engineering for Internet of Things: A State of Research," *Proceedings of the 23rd ACM/IEEE international conference on model driven engineering languages and systems: companion proceedings*, pp. 1-8.

Khorram, F., Mottu, J.-M., and Sunyé, G. 2020. "Challenges & Opportunities in Low-Code Testing," *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, pp. 1-10.

KPMG. 2022. "Shaping Digital Transformation with Low-Code Platforms." from https://kpmg.com/xx/en/home/insights/2023/02/shaping-digital-transformation-with-low-code-platforms.html

Krejci, D., Iho, S., and Missonier, S. 2021. "Innovating with Employees: An Exploratory Study of Idea Development on Low-Code Development Platforms," *ECIS*.

Lebens, M., Finnegan, R. J., Sorsen, S. C., and Shah, J. 2021. "Rise of the Citizen Developer," *Muma Business Review* (5:12), pp. 101-111.

Lethbridge, T. C. 2021. "Low-Code Is Often High-Code, So We Must Design Low-Code Platforms to Enable Proper Software Engineering," *Leveraging Applications of Formal Methods, Verification and Validation: 10th International Symposium on Leveraging Applications of Formal Methods, ISoLA 2021, Rhodes, Greece, October 17–29, 2021, Proceedings 10*: Springer, pp. 202-212.

Luo, Y., Liang, P., Wang, C., Shahin, M., and Zhan, J. 2021. "Characteristics and Challenges of Low-Code Development: The Practitioners' Perspective," *Proceedings of the 15th ACM/IEEE international symposium on empirical software engineering and measurement (ESEM)*, pp. 1-11.

Martins, J., Branco, F., and Mamede, H. 2023. "Combining Low-Code Development with Chatgpt to Novel No-Code Approaches: A Focus-Group Study," *Intelligent Systems with Applications* (20), p. 200289.

Martins, R., Caldeira, F., Sa, F., Abbasi, M., and Martins, P. 2020. "An Overview on How to Develop a Low-Code Application Using Outsystems," *2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*: IEEE, pp. 395-401.

Mayring, P. 2000. "Qualitative Content Analysis, Forum," *Qualitative social research* (1:2), pp. 1-10.

Meuser, M., and Nagel, U. 1991. *ExpertInneninterviews—Vielfach Erprobt, Wenig Bedacht: Ein Beitrag Zur Qualitativen Methodendiskussion*. Springer.

Overeem, M., Jansen, S., and Mathijssen, M. 2021. "Api Management Maturity of Low-Code Development Platforms," *International Conference on Business Process Modeling, Development and Support*: Springer, pp. 380-394.

Prinz, N., Huber, M., Leonhardt, J., and Riedinger, C. 2024. "Unleash the Power of Citizen Development: Leveraging Organizational Capabilities for Successful Low-Code Development Platform Adoption,").

Prinz, N., Rentrop, C., and Huber, M. 2021. "Low-Code Development Platforms-a Literature Review," *AMCIS*.

Richardson, C., and Rymer, J. R. 2016. "The Forrester Wave™: Low-Code Development Platforms, Q2 2016," *Forrester, Washington DC*).

Richardson, C., Rymer, J. R., Mines, C., Cullen, A., and Whittaker, D. 2014. "New Development Platforms Emerge for Customer-Facing Applications," *Forrester: Cambridge, MA, USA* (15).

Rymer, J., and Appian, K. 2017. "The Forrester Wave™: Low-Code Development Platforms for Ad&D Pros, Q4 2017," *Cambridge, MA: Forrester Research*).

Sahay, A., Indamutsa, A., Di Ruscio, D., and Pierantonio, A. 2020. "Supporting the Understanding and Comparison of Low-Code Development Platforms," *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*: IEEE, pp. 171-178.

Sanchis, R., García-Perales, Ó., Fraile, F., and Poler, R. 2019. "Low-Code as Enabler of Digital Transformation in Manufacturing Industry," *Applied Sciences* (10:1), p. 12.

Silva, C., Vieira, J., Campos, J. C., Couto, R., and Ribeiro, A. N. 2021. "Development and Validation of a Descriptive Cognitive Model for Predicting Usability Issues in a Low-Code Development Platform," *Human factors* (63:6), pp. 1012-1032.

Smith, G., Papadopoulos, M., Sanz, J., Grech, M., and Norris, H. 2020. "Unleashing Innovation Using Low Code/No Code–the Age of the Citizen Developer," *ed: Arthur D. Little Prism*).

Sufi, F. 2023. "Algorithms in Low-Code-No-Code for Research Applications: A Practical Review," *Algorithms* (16:2), p. 108.

Thomas, D. R. 2003. "A General Inductive Approach for Qualitative Data Analysis,").

Tisi, M., Mottu, J.-M., Kolovos, D. S., De Lara, J., Guerra, E. M., Di Ruscio, D., Pierantonio, A., and Wimmer, M. 2019. "Lowcomote: Training the Next Generation of Experts in Scalable Low-Code Engineering Platforms," *STAF 2019 Co-Located Events Joint Proceedings: 1st Junior Researcher Community Event, 2nd International Workshop on Model-Driven Engineering for Design-Runtime Interaction in Complex Systems, and 1st Research Project Showcase Workshop co-located with Software Technologies: Applications and Foundations (STAF 2019)*.

Vincent, P., Iijima, K., Driver, M., Wong, J., and Natis, Y. 2019. "Magic Quadrant for Enterprise Low-Code Application Platforms," *Gartner report*).

Waszkowski, R. 2019. "Low-Code Platform for Automating Business Processes in Manufacturing," *IFAC-PapersOnLine* (52:10), pp. 376-381.

Zolotas, C., Chatzidimitriou, K. C., and Symeonidis, A. L. 2018. "Restsec: A Low-Code Platform for Generating Secure by Design Enterprise Services," *Enterprise Information Systems* (12:8-9), pp. 1007-1033.

Eidesstattliche Erklärung

I hereby confirm that the attached bachelor thesis is my own work and that it has not been used for other examination purposes: I have named all the sources and auxiliary material used, and I have marked appropriately quotations used verbatim or which I have given the gist of. I tolerate the check using anti-plagiarism software.

Ertingen, den 19.08.24

Place, Date

Signature