Exploring the use of low-code software development in the automotive industry
- Thesis conducted at the Scania engine assembly facility

Mjukvaruutveckling med low-code inom fordonsindustrin

Matteus Henriksson

Examiner : Niklas Carlsson

External supervisors : Henrik Enochsson & Jonny Weidenblad
Upphovsrätt

Detta dokument hålls tillgängligt på Internet - eller dess framtida ersättare - under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnt som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/.

Copyright

The publishers will keep this document online on the Internet - or its possible replacement - for a period of 25 years starting from the date of publication barring exceptional circumstances.

The online availability of the document implies permanent permission for anyone to read, to download, or to print out single copies for his/hers own use and to use it unchanged for non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional upon the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: http://www.ep.liu.se/.

© Matteus Henriksson
Abstract

The growing demand for IT professionals to develop and maintain software is creating huge concerns for organizations looking to keep up and build resilience. The trends show that there is a large shortage compared to the demand. Manufacturing companies are no exception to this, and the adoption of Industry 4.0 technologies will most likely increase this need even further. However, developing software solutions often requires specialized knowledge, and organizations are looking far and wide for the competence. If it is present, it is often found in the organization’s centralized IT departments.

Low-code development might offer a solution, by lowering the threshold for employees and individuals to develop, deploy and maintain applications with limited technical expertise. Low-code development is a method that enables software applications to be built using graphical user interfaces in so-called low-code development platforms. The platforms provide templates, drag-and-drop modules and integration with other systems to reduce the complexity of the application development process. These platforms could offer advantages for manufacturing firms, including quicker development that could aid in responding to new requirements. Other benefits include: cost savings and shorter development cycles, allowing for faster releases which can help enterprises stay more competitive and reduce risk.

This thesis probes the factors to consider with low-code platforms, such as the role of citizen developers and the balance between accessibility and control in the context of manufacturing industries. It explores how low-code development could enhance efficiency and presents a case study from the engine assembly facility in Scania Södertälje. Best practices for building an organization around low-code developers are discussed. The insights from this study provide an understanding of the complexities and potential of low-code development, laying the groundwork for further exploration and implementation. Combined, the results and insights offer a clear perspective on how low-code development can assist organizations in the digital age.
Acknowledgments

I would like to thank my family and friends for all their support. Special thanks go to my mother for her endless encouragement and belief in me, my father for challenging me and pushing me beyond what I thought was possible, my brother for inspiring me to enroll and pursue this degree and my girlfriend for supporting and reminding me to write. I would also like to thank my close friend Sebastian for always motivating me to persevere and move forward, even when things seem tough. Finally, I would like to thank my examiner, Niklas, for his support and patience throughout this thesis. Thanks to all of you, I managed to finish this thesis despite my doubts.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>iii</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>iv</td>
</tr>
<tr>
<td>Contents</td>
<td>v</td>
</tr>
<tr>
<td>List of Figures</td>
<td>vii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>viii</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Motivation</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Aim</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Research questions</td>
<td>2</td>
</tr>
<tr>
<td>1.4 Contributions</td>
<td>2</td>
</tr>
<tr>
<td>1.5 Delimitations</td>
<td>3</td>
</tr>
<tr>
<td>1.6 Thesis outline</td>
<td>3</td>
</tr>
<tr>
<td>2 Background</td>
<td>4</td>
</tr>
<tr>
<td>2.1 Scania</td>
<td>4</td>
</tr>
<tr>
<td>2.2 Low-code development (LCD)</td>
<td>4</td>
</tr>
<tr>
<td>2.3 Robotic process automation (RPA)</td>
<td>5</td>
</tr>
<tr>
<td>2.4 Microsoft power platform</td>
<td>5</td>
</tr>
<tr>
<td>3 Related work</td>
<td>8</td>
</tr>
<tr>
<td>3.1 Adoption rate and limitations of low-code development</td>
<td>8</td>
</tr>
<tr>
<td>3.2 Low-code development in enterprise environments</td>
<td>10</td>
</tr>
<tr>
<td>3.3 Evaluation and assessments in low-code development</td>
<td>11</td>
</tr>
<tr>
<td>3.4 Other</td>
<td>12</td>
</tr>
<tr>
<td>4 Method overview</td>
<td>13</td>
</tr>
<tr>
<td>5 Pre-study</td>
<td>14</td>
</tr>
<tr>
<td>5.1 Literature review</td>
<td>14</td>
</tr>
<tr>
<td>5.2 Observations</td>
<td>14</td>
</tr>
<tr>
<td>5.3 Expert interviews</td>
<td>14</td>
</tr>
<tr>
<td>5.4 Results from interviews</td>
<td>15</td>
</tr>
<tr>
<td>5.5 Identified challenges</td>
<td>15</td>
</tr>
<tr>
<td>6 Case study: issue tracking system with low code development</td>
<td>19</td>
</tr>
<tr>
<td>6.1 Case study framework</td>
<td>19</td>
</tr>
<tr>
<td>6.2 Criteria</td>
<td>19</td>
</tr>
<tr>
<td>6.3 Data collection</td>
<td>20</td>
</tr>
</tbody>
</table>
List of Figures

2.1 Template application in power apps. .......................................................... 6
4.1 Overview of the method and the contents of each chapter. ......................... 13
6.1 Activity diagram of the tasks performed by the engineers. .......................... 21
7.1 System context diagram (level 1) using the C4 model. ............................... 23
7.2 System container diagram (level 2) using the C4 model. ........................... 24
List of Tables

5.1 Identified challenges with LCDPs in literature ........................................ 16
6.1 System requirements for the issue tracking system .................................... 20
8.1 User feedback on the issue tracking system ............................................ 26
This chapter introduces the motivation to study low-code development platforms (LCDP), scope and research questions for it. The thesis aims to explore the potential and challenges of low-code development (LCD) and was conducted in Scania's engine assembly facility. The focus is on understanding the implementation challenges with LCD and how it potentially can be used to reduce risk in projects that involve software development. With empirical research and a case study approach, the thesis provides insights for organizations considering to invest in LCDPs.

1.1 Motivation

With the recent advancements of artificial intelligence we are seeing more and more generated code being produced and deployed. Research and media companies are presenting the advancements as one of the most emerging technologies of 2023\(^1\) Githubs CEO claims he believes that 80% of the code that is written in the future is going to be generated by AI and with the obvious uptrend of OpenAIs ChatGPT\(^2\) it is easy to see where these bold claims come from. But, generating code is useful if the recipient knows how to read and use the code, something that is not common for the vast majority. To solve this, enterprises are investing on platforms allowing LCD, i.e. software development through the graphical user interface and the people that will use them, citizen developers.

Predictions vary regarding the adoption of LCDPs, with some expecting them to be used for the vast majority of applications in the next years \[^3\], while others are more skeptical \[^4, 17\]. The different opinions and views on LCDPs underscore the complexity, balancing the benefit of quicker development with concerns over potential restrictions \[^17\].

The growing interest in LCDPs is partly driven by a shortage of developers and IT professionals \[^22\], creating a demand for innovative solutions that can bridge this gap. LCDPs respond to this need by offering opportunities for enterprises of all sizes. By enabling a wider range of individuals to engage in software development, these platforms are not only meet-

ing the demand but could also drive innovation and help enterprises stay competitive [14, 14].

Software development projects often face challenges such as cost overruns, delays, changing requirements, and security concerns [21, 29, 10, 5, 7, 32] despite these challenges it is clear that digitalizing has a positive impact on industries creating new opportunities for growth and efficiency [24, 35]. However, these challenges can impact the quality and success of the project significantly. The LCDPs offer a promising solution to some of these challenges making it interesting for enterprises like Scania, potentially reducing delays, lowering costs and reducing security concerns with reusable and tested components. While the adoption of LCDPs offers numerous advantages, it also introduces its own set of complexities that organizations must navigate. Interoperability, customization, maintenance, and lack of documentation [11] become critical factors. The involvement of citizen developers and the necessity for high-level testing have been recognized and while LCDPs could potentially improve efficiency they also introduce new challenges that must be studied.

Through empirical research, challenges have been identified and explored to some extent [1, 17, 15]. This master thesis further explores the potential and challenges of LCDPs, an emerging technology for democratizing software development. Conducted within an automotive factory setting, the study investigates how LCD can be integrated into organizational processes across various industries. Through literature review, observations, interviews and a case-study the thesis also probes certain factors to consider with LCD, such as the role of citizen developers, training and the balance between accessibility and control.

1.2 Aim

The aim of this thesis is to investigate the maturity of LCDPs, identify the challenges that arise during implementation and to provide empirical data such as observations and expert interviews. This study also serves as a guide for organizations considering investing in these technologies, while also highlighting areas where further research is needed.

1.3 Research questions

The following research questions will be answered.

1. What challenges arise when trying to incorporate low-code development platforms into organizations?
2. How can low-code development be used to reduce risk in projects that involve software development?
3. How can organizations integrate more employees into their digitalization strategies and what specific actions can be taken?

1.4 Contributions

The contributions of this thesis can be summarized as follows:

1. Understanding low-code development challenges: The thesis sheds light on common challenges brought up in the literature. Chapter 5 captures these challenges.

2. Risk management through user-centric development: The C4 model presented in Chapter 7 clearly illustrates dependencies and interactions, thereby spotlighting potential vulnerabilities. Furthermore, the Think-Aloud protocol presented in Chapter 8 proactively addresses user needs and potential issues. Together, these tools equip stakeholders with the ability to identify and address potential challenges.
3. **Enterprise digitalization case study**: The thesis showcases how low-code development platforms can be used for organizations looking to adopt the technologies. A description of the case study can be found in Chapter 6, which is further discussed in Chapter 9.

### 1.5 Delimitations

This thesis uses the Microsoft power platform as a low-code development platform for the case study. While there are other low-code platforms available, this study focused on the Microsoft power platform to stay aligned with strategic choices made by Scania.

Second, the case collection and interviews were conducted at Scania so the research tends to lean more on how these tools can help in industries that manufacture goods. However, due to privacy concerns and avoiding revealing potentially sensitive information some of the collected data and some of the analysis performed during this project will not be publicly published. This thesis can therefore only be seen as a shorter summary of the work conducted during this thesis project.

Finally, while cost benefits will be mentioned, the thesis does not explicitly analyze the cost benefits aspects of LCD.

### 1.6 Thesis outline

The thesis begins with a background section in Chapter 2 consisting of the foundational elements, such as LCD, robotic process automation (RPA) and the Microsoft power platform. After that the related work is presented in Chapter 3 where the adoption rate and limitations, LCD in enterprise environments and other related topics are discussed. Before moving into the study the method is presented in Chapter 4. After that the study starts with a pre-study in Chapter 5 gathering insights from the literature review, observations and expert interviews. This is followed by a case study section in Chapter 6 where the issue tracking using LCD is presented. Chapter 7 presents the system design and in Chapter 8 the system is evaluated using usability testing. The reflections and certain implications of the study are discussed in Chapter 9 before wrapping up with the conclusion in Chapter 10. Finally, Interview questions and references can be found in the appendix and bibliography section.
2 Background

This chapter aims to explain concepts necessary to understand the thesis. A brief explanation of Scania’s engine assembly facility will also be given since the thesis was conducted there, besides that the LCDP that was used in the thesis will also be explained a bit further.

2.1 Scania

Scania is one of the leading manufacturers of heavy transport vehicles. The company is based in Södertälje Sweden and is focusing on manufacturing trucks, buses and engines for industrial machines and marine vehicles. The company has production sites in more than 100 countries with a total of approximately 54,000 employees.

2.1.1 Engine assembly facility in Södertälje

The engine assembly facility in Södertälje is a key production site that specializes on straight 5 and 6 diesel engines as well as V8 diesel engines. These engines are among other used for trucks, buses and for marine vessels. The facility has highly automated assembly lines that are using operational technology in combination with IT-systems.

2.2 Low-code development (LCD)

Low-code development (LCD) is a technology that allows individuals with various skills to develop software applications. This is typically achieved through a platform equipped with a graphical user interface. Low-code platforms (LCDPs) offer pre-built components, templates and libraries. This approach accelerates the development of new applications in a more cost-effective way.

The rise in popularity of LCD can be attributed to several factors. One of the primary drivers is the need for speed in software development projects. In today’s fast-paced world, businesses are under increasing pressure to deliver quickly and LCD offers a means to meet

this demand. Another contributing factor is the ongoing shortage of IT professionals. LCDPs democratize software development, enabling a broader range of individuals to participate in the development process. This helps to alleviate the pressure on IT departments and can lead to more diverse and innovative solutions. With this, LCD could help to reshape the software development landscape, making it more efficient and adaptable to the needs of businesses and industries.

2.2.1 No-code development

No-code development is another term used in conjunction with LCD. The method is the same—using a graphical interface to develop applications—however with no-code development in contrast to LCD the aim is to completely remove the need for writing code. This thesis primarily focuses on LCD and since the distinctions between the two is unclear so there may be instances where the two are interchanged.

2.3 Robotic process automation (RPA)

Robotic Process Automation (RPA) is a widely adopted software technology across various industries that serves to automate repetitive and time-consuming tasks. RPAs are designed to mimic actions and can navigate through graphical user interfaces and perform tasks such as identifying and extracting data.

One of the key advantages of an RPA is its ability to be implemented with minimal disruption to existing processes. When combined with LCDPs, RPAs can be created and maintained even by individuals with less technical experience. Another benefit of RPAs is their compatibility with legacy systems that may lack APIs or other modern technologies, making them a versatile tool in workflow automation.

2.4 Microsoft power platform

The Power Platform is a set of software tools or products that have been developed by Microsoft. The platform consists of five products: Power Automate, Power Apps, Power Virtual Agents, Power BI and Power Pages.

2.4.1 Power automate

Power Automate, previously known as Microsoft Flow is an RPA tool. The tool has been released in two versions with the same name, one is cloud-based and the other is for desktop use. The difference is that the cloud-based version leverages the scalability of cloud computing, providing larger access to computing power which in turn provides better ability to handle larger workloads. The desktop version is an application that works independently on single hosts and is more suited for smaller teams or individual use where there is a limited need for scalability.

The tool provides integration possibilities with connectors to popular services such as SharePoint, Dynamics 365, and Microsoft 365, as well as the ability to create custom connectors.

2.4.2 Power apps

Power Apps is a platform for web and mobile application development. The platform like many other in the series allow for integration with other software systems and databases with the help of connectors. The platform is commonly used in combination with Microsofts other products such as Power Automate (Section 2.4.1) and can utilize AI technologies through Microsofts AI Builder module. Some examples of use-cases where the platform has been used are: inventory systems, digital injury prevention plans and online insurance portals. Figure 2.1 shows the graphical user interface (GUI).

2.4.3 Power virtual agents

Power Virtual Agents is a product designed to create chatbots also known as virtual agents. Virtual agents can be used for websites, mobile apps or messaging platforms. The product is based on natural language processing (NLP) and certain AI-technologies to give better responses to the user.

One of the main benefits of Power Virtual Agents is that it can automate tasks such as answering frequently asked questions with established answers. Additionally, Power Virtual Agents can be integrated with other software products, such as Power Apps and Power Automate to create more sophisticated solutions. However the product is restricted to only integrate with Microsoft platforms and services.

2.4.4 Power BI

Power BI is a business intelligence platform. The platform is used to analyze and visualize data and is one of the most popular BI tools on the market.

---

There has been a growing trend towards business intelligence technologies especially from enterprises that want to make use of their data. The large availability of data and the need to make better decisions has been a driving factor towards these tools. The product provides features for data analysis and visualization that is most often presented in dashboards that can be embedded into applications.

2.4.5 Power pages

Power Pages\footnote{https://powerpages.microsoft.com/ (Accessed August 3, 2023).} is a platform used to build web pages similar to power apps, even though it specialized on web pages. The platform is also used to develop and integrate with other Microsoft products like Power Apps and Power Automate. Some use-cases include: event registration pages, HR portals and school portals.
3 Related work

In this chapter the existing research on LCDP is presented. The first section (Section 3.1) presents studies on the adoption of LCDP and the current state of LCDP in the industry. In the second (Section 3.2) research on the use of LCDP in enterprises environments is presented. In the third (Section 3.3) research on the challenges developers face during the development process. We also touch on the difficulties associated with debugging and testing low-code solutions. Finally, in the last section (Section 3.4) other research areas related to LCDP are discussed.

3.1 Adoption rate and limitations of low-code development

The LCDPs have been perceived as a potential transformative technology that is driven by empowering citizen developers through reduced entry barriers and simplified design features, freeing IT departments for more complex tasks. To gain insight into the rise and adoption rate has been studied.

In a study conducted by Sahay et al. [26] the authors highlight the interest in LCD and the platforms delivering the services, just like many others [4, 26, 1, 14, 31, 3, 27]. To pick and choose can be challenging since the field is rather new, and it is especially hard for those lacking knowledge in the area of software development. Sahay et al. [26] presented an analysis of eight platforms that according Gartner and Forrester have been considered the leaders. The paper discusses the four layers in a low-code architecture and five phases when developing with LCDPs. The data collection was gathered through a survey in conjunction with a benchmark application developed with all of the eight platforms which where then used to analysis a set of features including: the visual development environment, data management, user interface design, integrations, security, collaboration, and customization.

Sahay et al. [26] found that the most common way of developing with LCDPs is to either use a UI to Data or Data to UI approach. In the UI to Data approach, development starts with the user interface by designing views and pages. When the layout is established, the views are connected to data sources and later on deployed. The Data to UI approach however, begins with data modeling where developers start with defining and structuring the data,
then building a user interface around it. In this case it is also more necessary to be aware of the business logic in current solution to adapt or alter workflows before deployment. UI to Data prioritizes user experience, while Data to UI emphasizes the structure of data.

Sahay et al. [26] highlight the importance of selecting a platform that fits the requirements as it can affect the efficiency of the development process significantly, the authors identified that the type of solution was interesting, for Business-to-business (B2B) solutions the focus tends more towards business processes where it is common to use a approach leaning towards Data to UI while for Business-to-consumer (B2C) solutions the interactivity is considered crucial, especially for small to medium enterprises.

In B2C solutions the size of the user’s organization, the cost, time investment needed to learn the platform and the price of the LCDP where important to consider. However before embarking with LCDPs it is very likely to encounter challenges. Interoperability is a large obstacle due to closed-source code of the platforms, making it difficult for LCDPs to exchange information. Extensibility is another one where adding new functionalities to platforms is cumbersome and in worse-case unattainable. It is also possible to argue that the learning curve for the users is to steep, partly because many LCDPs lack intuitive interfaces and educational resources. Additionally, scalability is a challenge as the platforms should ideally be cloud-based and capable of managing large volumes of data, but the absence of standards makes it hard to evaluate.

With the research made by Käss et al. [14] it is obvious that there has been a surge of research into LCD. The number of publications has risen from only 9 publications between 2017-2019, to 23 in 2020 and almost twice as many in 2021. But just like Sayah, Käss show that there are several challenges left before large scale adoption.

In their paper, Käss et al. explored the adoption of Low Code Development Platforms (LCDPs) among organizations, employing the Diffusion of Innovation (DOI) framework to analyze this trend. This framework, which views technology adoption as part of a social system influenced by collective beliefs about the innovation, identifies five key characteristics that affect the adoption rate: compatibility, complexity, relative advantage, trialability, and observability.

Both the above identified the concern of the interoperability among different LCDPs and has classified it as a issue among a large portion of the papers. Other issues that have been scooped up where, integration with legacy systems and the exploration of emerging technologies like with machine learning and IoT technologies and the challenge of supplying novice developers and more experienced developers with the need for more sophisticated functionalities.

However a driver of LCDPs is that they are favorable for organizations with constrained IT resources and budgets, given that the functionalities in the modules available in the platform are enough. Also important to notice is that users might have to adapt their initial requirements.

A concern related to the scalibilty is the lack of handling more complex applications and collaborating with multiple developers also supplying services to a larger user base with the data volumes can be complex and sometimes not even possible. Käss analysis show that LCDPs generally lack capabilities needed for complex development projects like source code management and collaborative development.

In general both the above discussed that organizations are open to adopting the platforms the current research but due to the large concerns dealing with interoperability and complexity among others it is not really possible to fully adopt the technologies, also current research is not enough, something that has been highlighted by Bock et al. [4]. To address this Käss among others authors propose to conduct more empirical research in organizational environments, something that also motivates this thesis.
3.2 Low-code development in enterprise environments

LCD holds the potential to help with digital transformation and improve business processes across different domains, something that is appealing for enterprises seeking to stay competitive. Bies et al. [3] explore the potential for LCDPs and how it can be used to drive digital innovation in small and medium-sized enterprises (SMEs). Bies argue that while SMEs have increased the digitalization efforts in response to the COVID-19 pandemic, the lack of IT related expertise can and is very likely to still be a barrier to pursue such a strategy. The paper focuses on showing the state of low-code applications in a corporate reality. As a method the authors conducted a case study approach where a part of it consisted of a survey sent out to 32 SMEs in Germany and semi-structured interviews with 11 experts.

The expert interviews of the study point out advantages like cost and efficiency i.e. making application development more affordable and accessible for SMEs. However, the experts also said that the advantages of LCD decrease as the complexity of the application increases, pointed in previous studies as well, Käss among other [14].

In the study the authors point out that 40% of the SMEs develop software internally, while the rest rely on external support indicating that there is a strong dependency on external IT personnel and service providers. This dependency is argued risky due to the increasing shortage of IT specialists.

Sanchis et al. [27] discuss the topic as well and refer to it as enterprise resilience, which they state as a critical concern for businesses in today’s complex and dynamic world. Organizations must be capable of responding rapidly to the changing environment or else they risk being outcompeted. How enterprises adapt to the changing environment is therefore crucial. To achieve this, they believe that companies are turning to software solutions that can improve and transform their processes and think that LCD could be a part of that.

Just like Käss and Sayah the authors also point out the limitations of LCDPs. As the complexity of the applications increase, the advantages tend to diminish. Moreover, issues with testing, performance and security still necessitate the need for IT experts.

In terms of adoption in SMEs, the authors state that 47% of the companies that have been surveyed have heard of LCD but are not familiar with it, 25% have not been involved with it at all and 28% are aware or already using LCDPs. It was also noted that many SMEs are not specifically seeking LCDPs, maybe since they are not familiar with it, but are more likely looking for specific systems to buy out of the box.

Another finding is that when SMEs do use LCDPs, they primarily focus on internal process optimization. Some examples include automation of postal processes, resource management, inventory applications, mapping processes of niche industries, and dashboards or minimum viable products for customer engagement.

Although many believe that the technology is not mature for major adoption, the authors highlight that 70% of SME respondents believe that LCD is highly relevant for SMEs, something that is agreed upon by 90% of the experts.

One of the things that was highlighted in the paper was the need for more transparency in the costs associated with LCDPs. The pricing models from LCDP providers are often non-transparent, making it easy for companies to encounter unexpected costs. This implies that providing SMEs with concrete use cases and more transparent pricing could motivate them to adopt LCD.

Another factor brought up is that of building IT expertise within the company. Around 41% of the surveyed SMEs believe they need to build more IT expertise to enable LCD. This involves training citizen developers. While these citizen developers need to understand business processes, they also require some level of IT knowledge to technically map these processes.

In the future working environment, experts feel that software developers will still be crucial, especially for maintenance, governance and for handling complex tasks. However, 82% foresee a shift in competencies and traditional roles due to citizen development. This might
reduce the need for traditional software developers but it is argued that it does not make them obsolete. Instead, it will allow them to focus on more demanding tasks and evolve the role of software developers.

However, the papers makes it clear that expecting every employee to develop applications using LCDPs is unrealistic and overly ambitious, as stated by the experts interviewed [14]. LCDPs are not a silver bullet to the IT skills shortage, since specialized IT personnel are still necessary for the complex application developments. However, citizen developers can mitigate the acute need for specialists and alleviate the workload. LCDPs can act as enablers in enterprises by broadening the base of employees capable of contributing to application development but are not a substitute for the specialized IT professionals.

LCDPs can offer significant benefits to enterprises within their domains. However, it is crucial to be aware of the potential issue of vendor lock-in. As applications grow and evolve over time, migrating applications or data can become increasingly challenging. One key point highlighted is the importance of fostering a collaborative culture between business units and IT departments.

3.3 Evaluation and assessments in low-code development

Given the potential surge of citizen developers it is crucial to ensure that the solutions are reliable for widespread deployment. Al Alamin et al. [1] present a study to research the types of challenges developers face when using low-code for software development and at which stage in the software development lifecycle they typically appear. Khorram et al. [15] present another study that focuses on testing of low-code solutions, where certain points are presented.

By analyzing Stack Overflow posts related to nine platforms Al Alamin et al. used topic modelling to extract topics that where grouped into different categories. The largest category was customization, the group held 40% of the questions. Where discussions where around implementing the business logic for the applications, linking the UI with backend, data storage and creating customizable UI components as examples. In another category "platform adoption" was discussed where a large amount of questions pointed towards Client-Server communication and debugging which seems to be a concern compared to traditional development.

Due to the closed-source nature and reliance on the GUI used in the low-code solutions they are perceived as very challenging not only to debug but also to test which was studied further by Khorram et al. [15] implicitly showing new demands for testing. Also brought up and discussed by Luo et al. [17] where they talk about the complexity and difficulty in debugging and maintaining of LCD-built applications, other subdomains as mocking when testing have also been studied [13].

In Al Alamin et al. study, Third-Party Integration was another category discussed. Despite receiving fewer questions, this category addresses important issues, showing that frequency of questions does not necessarily correlate with importance of the question. Specifically, it highlights the challenges of integrating with external services/APIs.

Among the topics however, Dynamic Event Handling stands out as the most challenging. It has the highest average view count and 76% of the questions do not have an accepted answer. A question like “How to add more data to an array of objects in a lightning component?”, which has been viewed 11,000 times over two years and still active, showcases the issue. Noteably is also that questions regarding access control & Security has the lowest average view count. That also aligns with the observation made by Luo et al. where developers perceived LCD platforms as capable of yielding more secure applications [17].

The studied findings indicate that there are trade-offs for using the platforms. Not only for the developers but also decision makers that need to be aware when setting strategies for learning, developing, deploying and testing.
3.4 Other

As mentioned by Käss et al. [14] there has been a surge of publications on LCD. The studies vary quite a bit, some address the technology itself or the development process [19, 33], some introduce frameworks [6, 8] and others try to pinpoint and extend functionality [16, 18].

Apart from the research on the LCDPs and technologies others also research problems and limitations related to using the platforms such as security [36] and other technologies like IoT [12] are also present and likely something that we will see more of.
This chapter presents the method used in this thesis and is divided into four parts: a Pre-study, a Case Study, the System Design, and Evaluation of the system.

The Pre-study, involves a literature review, observations, interviews and a summary of the identified challenges in the literature. The Case Study concentrates on the selection process while also providing a detailed description of the case study itself. The purpose of the case study was to understand and showcase how LCD can be used. The System Design includes details about the implemented issue tracking system and the choice of platform. The Evaluation chapter presents how the system was evaluated and the methods used to assess the usability of the system. Figure 4.1 provides a visual representation of these parts.
5 Pre-study

The pre-study consisted of three parts: a literature review to gain insights on the background and related research, observations of the engine assembly production line, and interviews with domain experts. Finally, the identified challenges with LCD was summarized in a table.

5.1 Literature review

To conduct the literature review relevant keywords were identified. The keywords were then used to search for peer-reviewed articles and literature in several search engines including Google Scholar, IEEE Xplore and UniSearch through Linkoping University. In order to narrow down the search, certain terms were excluded with boolean operators such as those related to low-level programming.

5.2 Observations

Patton highlights the value of observations in research, he states that the observations can provide detailed data about events and people in a more naturalistic setting. The observations can generate insights about behaviours and is a powerful tool in combination with other methods. For example when observing how the assembly workers interact with each other and with the IT-systems it is possible to see the surrounding factors that affect the workers, this would be much harder to explain through interviews or surveys.

To see the real world environment non-participant observations were conducted in one of the engine assembly lines. The observations were overt and unstructured and were carried out over the course of two full workdays. The purpose of the observations was also to understand the engine assembly environment, the facility and how technicians, operators and engineers interact with IT-systems and each other.

5.3 Expert interviews

Patton recommends using multiple methods, he refers to it as triangulation (increasing the credibility through multiple methods). As an example, interviews may generate deeper
discussions and thoughts but might not give a good understanding of the environment where observations can be used to provide a more complete picture of behaviours, like in the engine assembly lines. Besides this, Patton also notes that the observations can be used in methods such as case studies.

Following this, expert interviews where conducted, the experts consisted of: A data scientist with a hybrid engineering role and a project manager leading digitalization initiatives. The interviews were conducted in a semi-structured manner through Microsoft Teams and lasted about 30-60 minutes each. The interviews were meant to be conducted at a more general level, allowing the same questions to be asked to both interviewees without significant modifications. The purpose of the interviews was to map out the state of LCD and identify challenges when introducing new technologies, such as LCD. To protect the privacy of the interviewees, names or answers that may reveal their personal information will not be disclosed.

The questions can be found in Appendix A.1

5.4 Results from interviews

While we cannot share all details from the interviews, we share some insights here. First, the interviews revealed that employees lack the skills necessary to develop solutions using LCD tools. Introducing relatable and easy-to-understand cases with a focus on improving efforts could help to bridge this gap according to one of the interviewees.

One interviewee told that it is important to note that there is no one-size-fits-all solution, as knowledge about LCD tools and processes varies and will always do.

One approach that has been implemented by Scania is to establish a center of excellence to help employees update and develop new skills. The interviewee highlighted the importance of adapting to new technologies and to "create and maintain a learning environment" that encourages growth.

Another thing that was pointed out was the need for a different mindset, stating that "breaking free from traditional mindsets is essential for successful digital transformation." Empowering "citizen developers" is crucial rather than centralizing resources in the IT-departments.

In addition to these suggestions both of the interviewees highlighted the importance of making training and education more appealing and accessible. Combining different sources of content, such as podcasts, blogs, videos and workshops can inspire new ways of working and learning.

The idea of a "digital acceleration team" was also proposed. This team would promote digitalization within the organization and support the employees as needed.

5.5 Identified challenges

The literature on LCD reveals several challenges that must be considered when choosing and implementing software solutions with LCDPs. These challenges, ranging from interoperability to issues with maintenance, usability, and lack of documentation are essential to ensure that investments yield robust software applications. To give an overview of the most frequently mentioned ones in the literature, Table 5.1 presents these along with references to the studies.

5.5.1 Interoperability

One of the most discussed challenges among the literature was interoperability. Interoperability referring to the ability of systems to work well together despite differences such as

Table 5.1: Identified challenges with LCDPs in literature

<table>
<thead>
<tr>
<th>Challenge / Issue</th>
<th>Description</th>
<th>Literature References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interoperability</td>
<td>Difficulty in third-party integration; Lack of standards hampering interoperability</td>
<td>[14], [17], [12], [1], [26], [27]</td>
</tr>
<tr>
<td>Maintenance, Debugging and Testing</td>
<td>Difficulty in maintaining and debugging; challenges with automated testing; many unanswered questions on testing and debugging</td>
<td>[17], [12], [13], [1], [3], [15]</td>
</tr>
<tr>
<td>Customizability</td>
<td>Limitations in customizing applications, especially user interfaces and services</td>
<td>[14], [4], [17], [1], [26]</td>
</tr>
<tr>
<td>Economic Factors</td>
<td>Lack of transparency in price/performance measures; risk of wrong choice of LCSD application/platforms</td>
<td>[4], [17], [1], [3]</td>
</tr>
<tr>
<td>Documentation</td>
<td>Lack of platform and component documentation; severe lack of good tutorial-based documentation</td>
<td>[14], [1], [26]</td>
</tr>
<tr>
<td>Usability</td>
<td>Unfriendly user experience and limited usability leading to reliance on IT support</td>
<td>[14], [17]</td>
</tr>
<tr>
<td>Performance</td>
<td>Slow loading and publishing speeds</td>
<td>[17], [3]</td>
</tr>
<tr>
<td>Database Management</td>
<td>Challenges in SQL CRUD operations, data storage &amp; migration and file conversion among others.</td>
<td>[1]</td>
</tr>
<tr>
<td>Prior knowledge</td>
<td>Some platforms still require basic programming knowledge</td>
<td>[17]</td>
</tr>
</tbody>
</table>

protocols, operating systems, vendors and other factors. Sanchis et al. [27] identified interoperability as issue for LCDPs to reach adoption, especially legacy systems being motivating a part of their work. The lack of integration to legacy systems is also brought upp Käss et al. [14], other issues include absence of standard frameworks, leading to barriers when trying to exchanging architectural designs and components between vendors. Additionally LCDPs often lack the capability to integrate more advances technologies. Luo et al. [17] also pinpoint the potential for LCDPs to be lead to vendor lock-in potentially forcing users to specific tools. Related Interoperability issues, vendor lock-in and lack of standards where also brought by others [12], [1], [26].

5.5.2 Maintenance, Debugging, and Testing

The second most discussed challenge is the maintenance category which includes debugging and testing that is essential to high quality software. Luo et al. [17] discuss the issue with working on a high level of abstraction which is needed when using LCD. The problem is that to maintain and debug applications it is not always possible, since the developers do not have full visibility or control over the automatically generated code or the internal workings of the platform. Similar to this, Al alamin et al. [1] also bring up that testing and debugging are challenging due to the graphical nature of LCDPs which indicates a gap for testing and debugging. Bies et al. [3] mention that there is a certain complexity factor that forces tradi-
tionall programming knowledge in combination with LCD knowledge that hinders for those who do not have knowledge in both domains. Besides this unit testing, automated testing and mocking during development has been identified in multiple papers also hindering the maintenance of the applications [12, 13, 15].

5.5.3 Customizability

When it comes to customizability issues for LCD applications there are limitations mentioned in several papers, Käss et al. [14] discussed customizability and found that limited functionality hinders developers, which may lead to frustration due to the inability to develop complex applications. Bock et al. [4] also mentioned this while also pointing out that GUI design is limited to the pre-defined modules, which limits the developers. Luo et al. [17] also support this, saying that using LCD platforms can be easier and faster than traditional development but they often come with a restrictive customizability that potentially limits design and layout choices making it harder to create unique designs. Additionally added by Luo is that to fully customize all parts traditional coding is required, if it is even possible, since many LCDPs limit the access to lower level code. The emperical study provided by Alamin et al. [1] also provides some support to that since a large portion of the posts where related to this including UI customizability. Also brought up in the research by Sahay et al. [26] where the possibilities depend on the functionalities users might need to adjust their initial requirements depending on the options offered by the LCDP.

5.5.4 Economic Factors

Even though the economic implications are not directly analyzed in the literature, the implications have been discussed, Bock et al. [4] highlight that most LCD vendors do not disclose their pricing models making it harder to choose LCDPs. Some vendors integrate their pricing models with other services like consulting and educational offerings which can influence the price. Another factor pointed out by [17] is that some LCDPs charge a fee per user making it harder for smaller enterprises to keep the applications affordable and sustainable. Alamin et al. [1] also highlight these concerns with time and resource loss due to wrong choice of LCDPs, implying that there might be some issues with lack of transparency, they also bring up scalability concerns. Bies et al. [3] elaborate on this saying that the pricing models of LCDPs can be non-transparent, leading to unexpected costs for IT expertise.

5.5.5 Documentation

Good documentation leads to better quality and more maintainable code and is therefore relevant for LCDPs. Käss et al. [14] suggests that the citizen developers struggle to find relevant explanations to their problems which is supported by the study made by Alamin et al. [1] through a questionnaire sent out to developers. Sahay et al. [26] discusses this topic and believe that one reason is the insufficiancy of teaching materials, such as sample applications and online tutorials, which are needed to learn the best practices and the capabilites of the LCDPs.

5.5.6 Usability

For LCD to replace traditional software development it is important to address the usability, in this context meaning how easy it is to use the GUIs in the LCDPs. If the platforms are unintuitive or hard to use LCDPs lose their purpose. The paper by Käss et al. [14] discuss the complexity and point out that there are multiple occurrences that showcase the challenges for users. This issue in combination with the lack of documentation and the amount of training needed for the users suggests that that there still occurs some work to be done with LCDPs.
Luo et al. [17] point out that practitioners have conflicting views with the advantages of LCDPs, especially among the more experienced developers.

### 5.5.7 Performance

When developing software with the intent of growing the performance is important to consider, all though this is not widely studied in the literature Luo et al. discusses the issues [17], they have found that developers have expressed concerns about slow loading and publishing speeds on some LCDPs. Bies et al. [3] also touch on this but add that the performance is highly linked to the complexity of the application, for more complex applications the performance may not scale as effectively.

### 5.5.8 Database Management

Since most of the applications need to communicate with data sources it could also be relevant to study how the LCDPs interact with the databases. Alamin et al. [1] is one of the few that discusses database management challenges in their paper revealing that complex SQL tasks and data management activities require a deeper control beyond what is typically offered by the providers. Moreover, managing data relationships often forces developers to traditional coding practices which in turn affects the platforms’ ease of use.

### 5.5.9 Prior Knowledge

The idea with LCD is that prior knowledge of software development should not be necessary to develop, deploy and maintain. However, in the study presented from Luo et al. [17] they found that prior knowledge is required for using LCDPs and while the platforms may simplify certain aspects of software development citizen developers still need to have a foundational understanding of programming. Also they think that this need often arises due to the limitations of of the LCDPs when handling more complex or customized application requirements. This requirement for basic programming knowledge can act as a barrier to entry, and in way remove the central promise of LCDPs, i.e. to make it accessible to a broader range of individuals with varying technical backgrounds.
6 Case study: issue tracking system with low code development

The aim of this chapter is to present the framework that was chosen, how the data collection was made and how the case study was chosen based on the criterias.

6.1 Case study framework

Yin proposed a framework for case studies [34], the method provides some key steps to conduct better case studies. Although the framework was initially developed for social sciences it has gained popularity across a broad range of disciplines and is applicable in different domains [20, 2, 28].

Yin suggests that a case study should be defined and designed clearly, the first step towards that is to define research questions (Section 1.3), after that boundaries and units should be analyzed. The unit of analysis in this study is a recurring event in Scania’s engine assembly production line, specifically the event where an Engine Control Unit (ECU) fails to update or load software.

As the second step Yin proposes to select a type of case study, either exploratory, descriptive, explanatory or evaluative based on research questions. This study primarily takes a descriptive approach, aiming to provide a detailed description of a recurring event in a real-world context. However, it also has elements of an explanatory case study, as it seeks to understand the causes and impacts of this event. Yin also suggests collecting multiple case studies for comparison. In this study, multiple cases were collected. However, due to the sensitive information involved, only one case will be shared and discussed in detail.

After that a case study was chosen based on the criterias (Section 6.2). The data collection and selection process will be described bellow (Section 6.3).

6.2 Criterias

Something that has been highlighted in the literature (Yin among others) is that it is important to consider whether the case study is relevant to the objectives (and in turn research questions) of the study. If it does not provide sufficient data the risk is that it will not offer the insights that can contribute to a better understanding of the subject. To ensure that this was not the case, it was important to have a dialog with the external supervisors. Furthermore,
Table 6.1: System requirements for the issue tracking system.

<table>
<thead>
<tr>
<th>No.</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Development time</strong>: The development of the software system should not take more than 3 months.</td>
</tr>
<tr>
<td>2</td>
<td><strong>Collaboration possibilities</strong>: The system should provide possibilities to share information and collaborate between members.</td>
</tr>
<tr>
<td>3</td>
<td><strong>Customizability and User friendliness</strong>: The user interface should be user-friendly, engaging and be easily customized by members of the team.</td>
</tr>
</tbody>
</table>

when selecting the case study, it was important to set system requirements to prioritize and avoid scope-creep. Table 6.1 shows the key requirements for the system. The development time was the most important aspect to take into account due to the limited timeframe of the thesis and the available resources from Scania. In addition it was important to select a case that allows for implementation with a LCDP.

6.3 Data collection

To gather a diverse pool of cases, different working groups, mainly consisting of engineers, were visited. The group members where asked about their working routines and procedures where the intent was to identify and specify procedures so that they could be used to find areas of improvement.

Before each meetings, the participants were asked to reflect on their routines and procedures that they experienced as unnecessarily time consuming or ineffective, in combination with that they where also given examples. The meetings were conducted in groups and one-on-one through Microsoft Teams and in person. This approach allowed for remote participation and made it easier to join for everyone.

Since the participants had somewhat similar tasks it was encouraged to discuss the procedures. The goal of the data collection was to describe potential problems, to specify the workflows, processes or the lack thereof.

The pool ended up with 17 potential case studies.

6.3.1 Screening process

Once the criterias and potential cases have been produced Yin [34] suggests to begin the assessment and to rank them against the criterias to determine which cases that are most suitable for the study.

By evaluating the potential cases it was possible to identify a case that was aligned with the research questions and criterias. The chosen case is described in Section 6.4.

6.4 Case study: issue tracking system for programming issues related to engine control units in the assembly line

The engine assembly line is a process that uses manufactured components to construct engines. The process starts of by mounting an engine block on a carrier that transports the engine through the assembly line. Throughout this process the engine passes various workstations where assembly workers in combination with machines perform tasks, such as mounting, painting and programming. When the engine is assembled and quality assured it is either delivered to the final assembly, where it is mounted on vehicles or sold in separate.

The programming refers to a process where the workers plug in a cable to the Electric Control Unit (ECU) that in turn uploads software into the ECU. In the event of a failed pro-
gramming a red screen appears above in the working station indicating that something went wrong.

![Activity diagram of the tasks performed by the engineers.](image)

6.4.1 Activity diagram

In the event of a failed programming the assembly line workers will call the engineers responsible for the software that is used to program the ECUs (as it is crucial to ensure that the assembly lines are up and running).

Figure 6.1 shows an activity diagram, providing a visual representation of the sequence of tasks carried out by the engineers.

Typically, an issue arises when the programming of an engine fails (The trigger). In these cases the technician contacts the process engineers (Await call from technician), that are considered as the experts. The engineers need to gather information about the engine from the technicians so that they can query data from the IT-systems (Talk to technician & Query data). Based on the responses the engineers make decisions on how to proceed with the engines (Act on Issue).

6.4.2 Problem

The problem arises when the engineers receive a call, before the call they typically are unaware of the issue (since it is hard to predict). To act on the problem they need to collect the information about the issue manually, which means that they have to ask technicians for each detail over the phone. This procedure is time-consuming, creates unnecessary delays and increases the probability for losing information about the issue.
7 System design using the C4 model

In Chapter 6 a suitable case study was chosen. This chapter presents a solution for the challenge outlined in Section 6.4.2 utilizing LCD. The interaction between the different actors and IT-systems is also included.

7.1 Selecting development platform

The LCDPs that was chosen for the case study was the Microsoft Power Platform (Section 2.4). The platform is a popular choice that used by numerous organizations and holds a top position in the domain of Enterprise LCD. Scania has chosen to invest in the services as a part of their digitalization strategy, to stay in line with that strategic choice it was decided to pick the same platform for this study.

7.2 C4 model

The C4 model is a notation technique used to model the architecture of software systems. The model is structured into different levels from 1 to 4, 1 being the highest abstraction level.

The first level shows the system on a context level and its relationship with users and other software systems. The second level decomposes a software system into containers, where containers can represent different elements. The third level decomposes the containers into components and the fourth level decomposes components into code, where it can be used in combination with other notation techniques such as Unified Modelling language (UML) and Entity Relation Diagrams (ERD).

For levels one to three, the model uses five basic elements: Persons, Software Systems, Containers, Components and relationships. In this chapter a two leveled approach was utilized to illustrate the elements in the systems [30].

7.2.1 Level 1: System context diagram

Figure 7.1 shows four elements including the issue tracking system proposed in this thesis to store issue-related data. The Engineer entity is a person responsible for the predefined software package used to program/flash the ECUs, the Technician is a person that has the operational responsibility to program the ECU in the production line, the Production Information System is a set of Software Systems that stores information about the engine and production line.

The entities are connected to show the flow of information and the colors of the entities show which entities that belong to the scope. For example the light gray entity (Production Information System) is gray because it is not in the scope of this study, also it is not possible to modify it since the ownership lies at someone else.

The light blue entity (Issue Tracking System) is the system that was implemented. With this context diagram it is clear to see the dependencies for the systems, worth noting here is that the engineer depends on all of the elements and the technician depends on the engineer.

7.2.2 Level 2: System container diagram

From the requirements (Table 6.1) it was established that it is important to be able to store data concerning the issue (preferably before the technician calls), this was important so that the data from the Production Information System could be fetched earlier in the process. To achieve this the technician should post the data, preferably through a graphical user interface with minimal overhead to avoid any confusion and time wastage.

To meet the requirements it was decided to use a chatbot as it provides an intuitive and user-friendly way of interacting. The bot was integrated into a team in Microsoft Teams where all of the engineers, technicans and related staff are included.
Figure 7.2: System container diagram (level 2) using the C4 model.

Figure 7.2 shows the decomposed issue tracking system. The flow starts when the technician opens the integrated application in the team, the chatbot then prompts the user with questions and provides clickable options so that the user does not need to write anything manually. After the questions have been answered the data is forwarded and stored in a Microsoft List. The list is connected with Power Automate that triggers when a new item has been added or modified, Power Automate will in turn send an E-mail to the engineers to notify them of the issue.

In this chapter the issue tracking system (Chapter 7) is evaluated. Since it was important to assess how the end-users interact with the system a usability testing approach was taken in combination with a post test questionnaire, the testing approach allows to evaluate the system with close to realistic conditions so that areas of improvement can be found.

8.1 Evaluation methodology

To conduct the evaluation a Think-Aloud protocol proposed by Ericsson et al. [9] was used in combination with a survey that was sent out after the testing. The Think-Aloud method is a way of conducting user testing where the participants are being told to verbalize their thoughts, or “think loudly”, while testing. The purpose of that is to understand the thought process and to use that to identify issues and challenges with the system.

The first step was to select relevant participants to test the system. The target group consisted of two engineers and two technicians that work with programming of the ECUs.

The second step was to book meetings with all of the individuals and provide them with instructions of how the system will be be tested, here it was important to clarify and encourage them to “think loudly”.

After that it was time to conduct the actual testing, here the participants where asked to simulate a situation where the ECU programming has failed in the assembly line and that they were supposed to use the system.

8.2 Results from the think-aloud protocol

When presenting the issue tracking system to both of the technicians it was clear that the system filled a purpose and could potentially help both them and the engineers when dealing with these type of issues. As an example both of the technicians had experienced problems when calling the engineers, sometimes the engineers had a hard time to understand or have misheard what has been told over the phone due to loud environment or simply due to incorrect details being provided by mistake i.e. “human error”.

Another aspect that was pointed out by one of the technicians was that the system could be good to reduce lead time. The other technician was a bit more sceptic to this saying that
the system could potentially be good for reducing lead time but this assumes that the engineers are available to receive the information. The technicians main concern laid within the feedback from the system, it was not completely clear of how long it would take for the engineers to receive and act upon the issue. Here the technician also pointed out the importance of time as it could in worse case pause the production lines until the issue has been resolved.

Despite these concerns, both of the technicians stated that the system could be very useful for storing the issue data and to use it for the lessons learned process. Another thought and something was desired from the technicians was to add the possibilities to upload images. Some other thoughts that where brought up where suggestion on features that could help the technicians to solve the issue based on historical data.

The two engineers share the thoughts of the technicians and highlighted the part in the process where they received the call from the technician, they perceived it as a bit uncertain, sometimes a call could concern an issue related to the programming of the ECU and sometimes just questions concerning something else. The idea of receiving a notification of some kind was positive. The engineers also thought that the storing of the issues could be helpful in their work.

8.3 Results from survey

After completing the usability tests the study participants were provided with a post-test survey to gather feedback on their experience with the system.

The survey consisted of four questions, with responses measured on a scale from 1 (where the user strongly disagrees with the statement) to 10 (where the user strongly agrees).

After this, the average score was calculated, the average score was used to provide some indication of the satisfaction with the system (and was helpful to identify areas for improvement).

<table>
<thead>
<tr>
<th>Question</th>
<th>Observation</th>
<th>Average Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>How likely are you to continue using the system?</td>
<td>Users were positive about the system’s future use, indicating a high likelihood of continued usage.</td>
<td>9</td>
</tr>
<tr>
<td>Did you find the process time-consuming?</td>
<td>Users found the system to be easy to use and efficient, indicating low concern about the process being time-consuming.</td>
<td>2.5</td>
</tr>
<tr>
<td>Were the instructions for the evaluation clearly explained and would it be possible to navigate without them?</td>
<td>The users found the system easy to use and rather self explanatory, suggesting a user-friendly design.</td>
<td>9</td>
</tr>
<tr>
<td>Would you recommend the system to others in similar roles?</td>
<td>Users showed a high likelihood of recommending the system to their peers.</td>
<td>9.25</td>
</tr>
</tbody>
</table>

Table 8.1: User feedback on the issue tracking system.

Table 8.1 shows the result including the questions translated into English from Swedish, the feedback reveals that the system has met users expectations. The users are likely to continue using it and appreciate its easy-to-navigate design.
9 Discussion

This chapter discusses the results with prior research, looking into the methodological con-
straints and assesses the study’s replicability, reliability, and validity. The study also explores
the broader implications of the work, including benefits and potential risks, such as the emer-
gence of shadow IT.

9.1 Results

This section serves to discuss the results from the interviews, usability testing and how this
relates to the prior research.

9.1.1 Expert interviews

The interviews revealed a skill gap among employees who used LCDPs, some could with
minor help develop solution while others needed support through the whole process. This
aligns with the findings of Sahay et al. [26], who highlighted the challenge of finding a plat-
form that fits the requirements and different skill levels also how this affects the efficiency
of the development process significantly. The interviewees’ emphasised the importance of
creating a learning environment and adapting to new technologies, something that is found
in the Bies et al. [3] study aswell, who discussed the need for building IT expertise within
companies to enable LCD.

For larger enterprises the idea of a "digital acceleration team" could be benifitial, it is
also similar to the thoughts of Sanchis et al. [27] where they discuss the enterprise resilience,
emphasizing the need for organizations to be adaptable and responsive to rapid changes. The
idea of using a Center of Excellence (CoE) and a Digital Accelerator team could potentially
enhance the efficiency. The CoE can with its expertise establish best practices and provide
training while the Digital Accelerator team, on the other hand would drive innovation and
digital transformation, taking on the "high-risk", "high-reward" projects.

9.1.2 Think Aloud Protocol

The technicians feedback on the issue tracking system shows the potential benefits of a system
in reducing human errors, as well as its usefulness in storing issue data for future reference.
These findings are in line with the discussion by Sahay et al. [26] on the advantages of LCDPs, particularly in terms of efficiency and data management. However, the technicians’ concerns about the system’s feedback and the time it would take for engineers to receive and act upon the issues underscore the challenges. This is also reflected in the “Challenges” section of Table 5.1.

9.1.3 Relationship to prior research

The results from the expert interviews, think aloud protocol, and post-test survey provide some empirical data that supports the discussions in the literature. The results also highlight the benefits (and challenges) when using LCDPs in enterprise environments as well as the need for further research.

The findings also underscore the importance of considering the unique context and requirements of each organization when adopting LCD tools. As noted by Käss et al. [14], the adoption of these tools is influenced by various factors, including complexity and relative advantage.

9.1.4 Shadow IT

The low-code platforms could also lead to the spread of “shadow IT” within organizations. Shadow IT referring to IT systems and solutions built and used within organizations without organizational approval, often driven from centralized IT departments. While shadow IT can lead to innovation and efficiency, it also introduces risks with security and compliance. This is relevant in the context of LCD, since this allows a large amount of people to develop software applications and solutions.

9.2 Method

This section evaluates the methodology in this study, the replicability, reliability, and validity is discussed and it includes discussion of the sources used taking a critical standpoint.

9.2.1 Replicability, Reliability and Validity

The method employed in this study, which involved a pre-study phase, a case study, system design and evaluation phase is possible to replicate however due to the privacy concerns it is not possible to fully take part all details. The literature review process, the observations, and the expert interviews carried out as well as the use of established frameworks like Yin’s case study framework and the C4 model, should provide enough instructions so that the study can be repeated by other.

The reliability of the study, or the likelihood of obtaining the same results with the same method, will most likely be hard to attain and affected by the dynamic nature of the field. The rapidly evolving technology and the potential change in the engine assembly production line at Scania may lead to different results if the study is repeated in the future.

The validity of the study could be affected by the limited number of participants in the usability testing and the potential bias introduced by the selection of participants. To get a higher degree of validity it is needed to acquire more participants, especially for the survey.

9.2.2 Source Criticism

The sources used in this study were selected to ensure higher relevance. Peer-reviewed articles and literature were primarily used, and the search was conducted using reliable search engines. However, it is important to note that the most relevant information for the study was not always found in scientific literature. In many cases, valuable insights were obtained
through communication with software developers and through discussion forums. The lack of formal peer-review processes in these sources introduces a potential source of bias.

9.2.3 Participant selection for usability testing

The selection of participants for the usability testing was limited to two engineers and two technicians that worked with the programming of the ECUs. While this group was directly related to the problem at hand, the selection could have been more diverse to capture a wider range of user experiences. For instance, including participants from different roles within the organization, with varying levels of familiarity with the IT systems, could have provided another understanding of the system’s usability.

Moreover, the selection of participants could have also considered the diversity in terms of age, gender, and technical proficiency. These factors can significantly influence the way users interact with a system, and thus, their inclusion could have induced new findings.

9.2.4 C4 framework

The C4 model, was useful for visualizing the system architecture at two levels. However, it can not capture the complexity between different components of the system. The model’s focus on elements might have overshadowed certain aspects.

9.2.5 Evaluation approach

The evaluation method employed in this study was usability testing, along with a post-test questionnaire. Although this method yielded valuable insights into the user experience with the system, it may not have fully encompassed the system’s effectiveness in the real-world setting of the engine assembly line.

An alternative approach that could have been considered is a field study, where the system is observed being used in its natural environment for an extended period. This approach could provide better insights on how the system fits into work routines and how users adapt to it over time.

A field study could capture more of the user-system interaction that are often missed in controlled usability testing. For instance, it could reveal how users handle unexpected behavior during stressful events in the assembly line or how they incorporate the system into their existing routines.

Also a field study could uncover unforeseen uses or issues that may not arise in a controlled testing environment. Issues related to system performance under real-world conditions, integration with other systems in the work environment and impact on organizational culture. Additionally, a field study could provide understanding of the system’s impact. This approach could reveal not only the system’s effectiveness in terms of task completion or less error rates, but also its impact on other aspects of work, such as work efficiency and collaboration among the assembly workers.

9.3 The work in a wider context

The rise of LCDPs can democratize software development, enabling a broader range of individuals to create applications. While this has its benefits, it also introduces ethical and societal considerations.

First, there are concerns about job displacement. Platforms automate some aspects of software development, potentially reducing the need for some developers. However it can also be argued that these platforms are more likely to redefine developer roles than replace them.
Accountability is another issue. In traditional development, it is easier to trace the source of a bug or security issue. With LCD, this becomes more challenging due to the abstraction of underlying code. If a low-code application fails or causes harm, determining responsibility can be difficult.
Conclusion

This thesis explored LCD and serves to provide details for enterprises considering to implement the technology. The central aim was to demonstrate how LCD could improve operational efficiency and was conducted within an automotive factory setting but is relevant for enterprises in different industries as well. The results show that there could be operational improvements by integrating the technology.

10.1 Research questions

The thesis answered the following research questions:

1. **What challenges arise when trying to incorporate LCDPs into organizations?**

   Through the studied literature, observations, expert and group interviews the thesis identified several challenges presented in Chapter 5. These challenges varied widely, from issues such as lack of awareness and training to technical concerns in the platforms. Challenges related to interoperability, maintenance, customizability, documentation, debugging and testing were some of the most discussed in the literature.

2. **How can LCD be used to reduce risk in projects that involve software development?**

   For the early stages in a project the thesis suggests a approach based on higher levels, where a system design such as the one presented in Chapter 7 and user-centered evaluation methodology like the one presented in Chapter 8. By utilizing this it is possible to get early feedback in the projects creating possibilities for decisionmakers to react.

3. **How can organizations integrate more employees into their digitalization strategies and what specific actions can be taken?**

   Enterprises that are looking for strategic investments and want to digitalize can benefit from LCDPs. A starting point could consist of a data collection such as the one presented in Chapter 6. The thesis proposes a approach where citizen developers can take part of training in from experts to share best practices and knowledge as discussed in Chapter 9.
10.2 Future work

The thesis has pinpointed some challenges based on both literature and empirical data collected from the Scania facilities. However, there remains some work to be done. The lack of standards, training and the closed source nature of many LCDPs makes development restrictive, to propose solutions that could mitigate the impact more research is needed.

Further research could focus on:

1. Development of cross-platform low-code development tools
   Interoperability was identified as the biggest challenge, research could focus on creating tools that allow for cross-platform development to develop applications that can be deployed on multiple platforms.

2. Education and training
   As LCD grows, there will be a need for more education and training. This could involve courses, best practices on how to navigate issues like the once discussed in this thesis. To engage citizen developers, the learning could incorporate gamification and utilize interactive and competitive learning experiences.

3. Security
   Closed source low-code platforms can present security concerns due to their lack of transparency. this should be studied further to ensure more reliable software systems.

In conclusion, LCD is a dynamic field that is speculated to play an influential role in software development. However, it is important to anticipate the challenges that LCD presents.
A.1 Expert interview questions

1. What is your current job position or role at Scania, and what are your main responsibilities and tasks?

2. What are some of the biggest challenges that companies are facing when trying to encourage to use these tools?

3. What do you think are the reasons behind the limited awareness and understanding of the tools included in the Power Platform and how does this affect organizations?

4. What strategies or approaches do you think could be effective for raising interests in these tools?

5. What do you think are some of the obstacles that companies are facing and how can they overcome these?
Bibliography


