Charles Christiaan Versfeld  
Student Number: ST10091991  
PATHWAY: INRS7321  
Lecturer: Dr. Lizelle van der Walt  
17th May 2024  
[Word count: 2 821]

Investigating the relationship between the cyber security risks when using traditional hand coding practices in software development and the cyber security risks when using Low Code Development Software (LCDS) in software development amongst young IT software developers in software development industries in Gauteng.

Declarations:  
A blue screen with black text

Description automatically generated

A close-up of a text

Description automatically generated

TABLE OF CONTENTS

1. INTRODUCTION PG 4
2. TOPIC   
   2.1 Topic Introduction   
   2.2 Topic Background   
   2.3 Research Objectives   
   2.4 Scope and Limitations PG 4-5
3. BACKGROUND PG 5
4. PROBLEM STATEMENT PG 5  
   4.1 Global Issue  
   4.2 Local Issue  
   4.3 Assignment Problem Statement PG 5-6
5. RELEVANCE PG 6
6. PRIMARY RESEARCH QUESTION PG 6

6.1 Primary Question  
6.2 Primary Objectives

1. PRIMARY AND SECONDARY RESEARCH HYPOTHESIS AND HYPOTHESES PG 6  
   7.1 Ha1  
   7.2 Ha2 PG 6-7  
   7.3 Hypotheses 1  
   7.4 Hypotheses 2

7.5 Hypotheses 3  
7.6 Hypotheses 4 PG 7-8

1. THEORETICAL FOUNDATION PG 8-9
2. LITERATURE REVIEW PG 9  
   9.1 Introduction of LCNC in the 4th Industrial Revolution  
   9.2 The Ongoing Rise of LCNC Development  
   9.3 Benefits of LCNC Platforms  
   9.4 Cyber Security Concerns of LCNC  
   9.5 LCNC vs. Traditional Coding PG 9-10  
   9.6 Mitigation Strategies for LCNC Vulnerabilities  
   9.7 Conclusion
3. CONCEPTUALISATION PG 10-11
4. CONCLUSION PG 11  
   REFERENCE LIST PG 11-12

**1. INTRODUCTION**  
The research investigates common security vulnerabilities in Low Code No Code (LCNC) development, comparing them to traditional coding practices, and proposing mitigation recommendations. As young IT developers in Gauteng increasingly adopt LCNC practices, understanding security implications becomes crucial.

Alamin et al. (2021) highlight the productivity benefits of LCNC tools, including flexibility, agility, reduced bug fixing, lower development effort, and easier maintenance. However, they also caution against cybersecurity vulnerabilities inherent in these tools, which could lead to wasted company resources if the wrong LCNC tool is chosen.

With the LCNC platform market projected to reach $21 billion by 2022 (Alamin et al., 2021), and around 65% of large enterprises expected to adopt LCNC platforms by 2024 (Gartner), addressing security vulnerabilities in LCNC development is increasingly urgent.

This research assignment provides a comprehensive background and understanding of the study, outlining research objectives such as identifying security vulnerabilities, comparing LCNC with traditional coding, and proposing mitigation strategies. Additionally, it offers a literature review on LCNC use in the software development industry, providing a theoretical framework and analyzing previous literature for key concepts.  
  
**2. TOPIC**

"Security Vulnerabilities in LCNC Development: Comparative Analysis & Mitigation"

**2.1 Introduction:**

This study investigates security vulnerabilities in applications developed using LCNC compared to traditional hand-coding practices. Through a comparative analysis, we explore mitigation strategies to enhance LCNC application security.

**2.2 Background:**

LCNC development offers benefits like reduced production time and easier maintenance. However, concerns about cybersecurity vulnerabilities have been raised. Hon (2023) emphasizes these risks, while Alamin et al. (2023) highlight widespread developer confusion regarding LCNC platforms.

**2.3 Objectives:**

This study aims to:

Investigate security vulnerabilities in LCNC compared to traditional coding.

Analyze key LCNC vulnerabilities, including Authorization Misuse, Authentication Failures, Vulnerable Components, Security Logging Issues, and Misconfigurations.

Propose mitigation strategies to enhance LCNC application security.

**2.4 Scope & Limitations:**

This study focuses on comparing security vulnerabilities in LCNC and traditional coding practices, specifically examining five selected LCNC cybersecurity vulnerabilities. The scope is limited to literature review and may not cover all aspects of LCNC security.  
  
**3. BACKGROUND**

In today's dynamic world, the demand for innovative solutions has led to the emergence of Low Code No Code (LCNC) platforms as agile alternatives to traditional hand-coding practices (Manci, 2024). Initially introduced commercially in 2011, LCNC platforms have since evolved to meet evolving industry needs, tracing their origins to fourth-generation programming languages (Wikipedia, 2024).

However, amidst LCNC's growing popularity, concerns regarding security vulnerabilities have arisen. Gert Van de Ven stresses the importance of understanding these vulnerabilities for effective risk mitigation (Ven, 2023). The lack of security expertise among business users further complicates matters, as they may prioritize business value over security considerations, inadvertently introducing vulnerabilities into applications (Ven, 2023).

Recent incidents, such as the security breach handled by Microsoft's Detection and Response Team (DART), highlight the severe consequences of poor security in LCNC applications (Ven, 2023). This underscores the need for organizations to prioritize security in LCNC development and implement robust security controls.

This study aims to provide clarity on LCNC use, identifying appropriate scenarios for its application and offering guidance on mitigating security vulnerabilities associated with LCNC applications.  
**4. PROBLEM STATEMENT**  
**4.1 Global Issue:**

The widespread adoption of Low Code No Code (LCNC) platforms globally has sparked concerns about cybersecurity risks, with researchers highlighting common vulnerabilities such as insecure coding practices and lack of visibility and control (Hon & Ven, 2023).

**4.2 Local Issue:**

In South Africa, LCNC adoption is rapidly increasing, particularly in Gauteng, according to Business Tech findings. However, there remains a significant gap between adoption and implementation, compounded by the country's high cybercrime rates and challenges faced by small to medium businesses (Manci, 2024).

**4.3 Assignment problem statement:**

The aim of this research is to address the mentioned concerns by providing young IT developers in Gauteng with better knowledge of the cyber security risks associated with LCNC applications compared to traditional hand-coding practices. By assessing the security effectiveness of LCNC applications relative to traditional coding standards, the research seeks to identify key vulnerabilities and propose recommendations for mitigating security risks. Ultimately, the goal is to empower young IT developers to make informed decisions and enhance the security posture of organizations leveraging LCNC technologies.

**5. RELEVANCE**

This research holds significance for young IT professionals in the software industry, providing crucial insights into Low Code No Code (LCNC) applications, their benefits, and associated challenges. By enhancing understanding and awareness of LCNC, developers can utilize the platform more effectively, thereby mitigating cybersecurity concerns raised by researchers such as Van de Ven, Manci, and Quixy.

Understanding LCNC is essential for software development organizations, as it enables quicker and safer production of applications. By addressing security vulnerabilities in LCNC applications and comparing their efficiency with traditional hand-coded applications, this study aims to provide actionable solutions and guidance for young IT developers, fostering the formulation of effective and secure coding methodologies.

**6. PRIMARY RESEARCH QUESTION**  
**6.1 Primary Question:**   
 “*Is coding an application more safer and efficient being developed using traditional  
 practices then an application being developed by LCDS (Low Code Development  
 Software) ?”***6.2 Primary Objectives :**  
1. Assess the security effectiveness of LCNC applications relative to traditional coding  
 standards.  
2. Identify key vulnerabilities in LCNC applications.  
3. Propose recommendations for mitigating security risks associated with LCNC  
 development.  
4. Provide guidance for young IT developers in the software development industry to  
 formulate effective and safer coding methodologies.

**7. PRIMARY AND SECONDARY RESEARCH HYPOTHESIS AND HYPOTHESES**

**7.1 Ha1**  
*“when using LCDS (Low-Code-Development-Software) the productivity is much greater than using traditional hand coding practices as the time taken to finish an application takes less time taken.”*This is proven by multiple sources such as Alamin, Hon, Quixy, Manci and Ven all agree that the use of LCDS does in fact increase a software development process of an application.  
**7.2 Ha2** *“when using traditional hand coding practices the chances of security vulnerabilities are much less than using LCDS.”*This is debated by many industrial experts as there are too many variables that come into motion such as expertise in the specific code, knowledge of security practices and expertise in application development however according to eSystems Nordic, Forrester Research says “Applications built on low-code platforms can be more secure than those built with more traditional coding methods. Low-code vendors take on major responsibilities for securing their platforms on their ‘own’ clouds and ensuring the technical quality of applications built with their tooling.” And looking deeper into the study made by Carielli, Bratincevic and Rymer their findings thus this hypothesis is proven not accurate by the researcher.**7.3 Hypotheses 1**  
*“If LCDS (Low-Code-Development-Software) code security snippets used in the code development stage of the software development life cycle is better refined and improved upon.”*According to a short article by Quickflow arguing why it is important to implement good security coding standards for LCNC which indicates that the tool itself is quiet limited for developers, it is further argues that implementing better security features, standards, capabilities and better practices can “harness the full potential of low-code platforms” as stated in the article thus meaning if LCNC is better refined many experts will look at the platform differently thus making this hypotheses accurate.

**7.4 Hypotheses 2***“An application will be much more secure and quicker developed if Young IT software developers new to the IT Software development industry use a hybrid development technique which combines LCNC for more general algorithms and traditional hand coding practices for data handling security.”*According to Van de Ven there is a lack of expertise both in security expertise and the use of LCNC expertise which indicates developers are suing the LCNC platforms without understanding what they are working with, additionally Ven argues in his solutions that in order to relieve the situation the solution is to provide guidance and training on the LCNC Platforms. Additionally Ven states to use an external guidance source which is a suitable framework made by OWASP to help developers understand how to code using LCNC thus making this hypotheses accurate.  
**7.5 Hypotheses 3** *“if the above hypotheses are correct then much safer and efficient applications can be produced in a short period of time.”*According to Ven, Quicklfow and Quixy by providing the mentioned solutions stated by each of the authors sources as well as for more improvements as LCNC grows, sooner or later old coding methods will be put aside to give way to LCNC as each authors agrees LCNC is the future of coding however it is advised by many aswell such as Alamin, Manci and Nordic that further research and observations are needed before LCNC becomes the lead solution to software development thus making this hypotheses accurate.  
**7.6 Hypotheses 4**“If all the above hypotheses are correct then the IT software development industry will trust LCDS more.”  
This is accurate as all the mentioned authors and research done in each article has shown similar patterns that LCNC Platforms do have great potential however that potential is not at its full and provided with the solutions in each research source applied then the trust for LCNC Platforms in the IT software development from the industrial experts will increase much more thus making this hypotheses accurate.

**8. THEORETICAL FOUNDATION**  
Both LCNC and Traditional hand coding practices use the same set of theories (Agile Development) with LCNC only having one additional framework where in the agile development process the development aspects of the process is increase much greater and usually in coding practices the development aspect of the process is the one with the greatest time taken as proven by Alamin in the development phase an LCSD can be made multiple ways based of the approach that the developer takes.

A diagram of software development stages

Description automatically generated  
(Alamin, M.A.A. *et al.*,2021)  
  
According to OWASP’s framework for applying best security practices when using LCNC there is quiet a variety of aspects to look at and in each aspect given in the list is a detailed overlay and in-depth explanation of how to apply the specific security practice to an LCNC populated code aspect. This framework is most beneficial to young IT developers in all aspects of the globe as it highlights specific how to apply the practices correctly and how to ensure security in the application as a whole.  
 **9. LITERATURE REVIEW**

**9.1 Introduction of LCNC in the 4th Industrial Revolution**

In today's rapidly evolving software development landscape, Low Code No Code (LCNC) platforms have emerged as a significant innovation. While offering numerous benefits, they also raise critical cybersecurity concerns. This literature review explores the history, benefits, and security challenges of LCNC development, comparing it with traditional hand-coding practices and examining the specific context of LCNC adoption in South Africa.

**9.2 The Ongoing Rise of LCNC Development**

The concept of LCNC development dates to the 1990s and early 2000s with the advent of fourth-generation programming languages. However, it was not until 2011 that the first commercial LCNC platforms became available (Wikipedia, n.d.). According to a Forrester report, the LCNC market is projected to reach $21 billion by 2022, while Gartner predicts that by 2024, 65% of large enterprises will use LCNC platforms to some extent (Alamin et al., 2021).

**9.3 Benefits of LCNC Platforms**

LCNC platforms are celebrated for their ability to accelerate software development, providing flexibility and agility that traditional coding methods often lack. Alamin et al. (2021) argue that LCNC tools significantly reduce bug fixing, lower development effort, and simplify maintenance. Furthermore, the cost efficiency associated with shorter development cycles and reduced need for specialized coding skills makes LCNC an attractive option for many organizations (Manci, 2024; Quixy, n.d.).

**9.4 Cyber Security Concerns of LCNC**

Despite these benefits, LCNC platforms are not without their challenges. Hon (2023) identifies five major cybersecurity risks associated with LCNC development: insecure coding practices, reliance on third-party components, lack of visibility and control, shadow IT, and misconfigurations. Doerrfeld further elaborates on common vulnerabilities, such as authorization misuse and security misconfigurations, highlighting the potential for significant security breaches. Van de Ven emphasizes the lack of expertise and experience regarding LCNC and security knowledge.

**9.5 LCNC vs. Traditional Coding**

Comparative studies indicate mixed results regarding the security effectiveness of LCNC applications versus traditional hand-coded applications. LCNC platforms offer faster development and deployment but often sacrifice customization and control, which can lead to security vulnerabilities (Quickflow, n.d.). Traditional coding practices, though more time-consuming, allow for greater flexibility and more robust security measures. Carielli, Bratincevic, and Rymer argue that LCNC has the potential to exceed traditional coding practices if improvements and suggestions are implemented (Carielli et al., 2021).

**9.6 Mitigation Strategies for LCNC Vulnerabilities**

To address these vulnerabilities, several best practices have been recommended by frameworks such as OWASP. These include providing security training for business users, implementing robust security controls, and conducting regular security testing (Van de Ven, 2023; Quickflow, n.d.). Industry guidelines and frameworks from cybersecurity organizations offer further recommendations to enhance LCNC security and ensure efficient and safe practices are applied.

**9.7 Conclusion**

In South Africa, LCNC adoption is gaining momentum, with 81% of software development organizations considering it strategically significant, though only 31% have integrated it into their strategies (Manci, 2024). The country’s high cybercrime rates and the specific challenges faced by small to medium businesses, such as limited resources and lack of awareness, underscore the need for improved cybersecurity measures in LCNC development (TSBS Team, 2024).

The literature highlights both the potential and the risks of LCNC development, emphasizing the need for a balanced approach to leveraging its benefits while mitigating its security challenges. This research aims to fill gaps in the current understanding of LCNC security effectiveness compared to traditional coding, providing valuable insights and recommendations for young IT developers in Gauteng and beyond.  
  
**10. CONCEPTUALIZATION**

Low Code No Code (LCNC) Platforms – Development environments that allow users to create applications with minimal hand-coding, using graphical interfaces and configuration instead. These platforms are designed to accelerate development and make it accessible to users with limited programming skills (Alamin et al., 2021).

Traditional Coding Practices – Conventional software development methods that involve writing detailed code manually. This approach offers greater flexibility and control but requires significant time and expertise (Quickflow, n.d.).

Cybersecurity – The practice of protecting systems, networks, and programs from digital attacks. In the context of this study, cybersecurity focuses on identifying and mitigating vulnerabilities specific to LCNC platforms and comparing them to those in traditional hand-coded applications (Hon, 2023; Doerrfeld, 2023).

Security Vulnerabilities – Weaknesses in a system that can be exploited by threats to gain unauthorized access or cause harm. This study examines common vulnerabilities in LCNC platforms, such as insecure coding practices, third-party component risks, and misconfigurations (Hon, 2023; OWASP, n.d.).

Security Effectiveness – The ability of a system to prevent, detect, and respond to security threats. This research compares the security effectiveness of LCNC applications with traditional hand-coded applications, considering factors like the rate of vulnerabilities and the success of mitigation strategies (Carielli et al., 2021).

Mitigation Strategies – Techniques and practices used to reduce the severity and impact of security vulnerabilities. For LCNC platforms, this includes user training, robust security controls, and regular security testing (Van de Ven, 2023; OWASP, n.d.).

IT Software Development in South Africa – The context of this study, focusing on the adoption and challenges of LCNC platforms within South Africa's software development industry. Factors include the rapid increase in LCNC adoption, cybersecurity awareness, and the specific needs of local businesses (Manci, 2024; TSBS Team, 2024).

Security Awareness – The level of understanding and proactive behavior regarding cybersecurity practices among developers and end-users. This concept is crucial for mitigating risks associated with LCNC development (TSBS Team, 2024).

LCNC Adoption and Implementation – The process by which organizations incorporate LCNC platforms into their development workflows. This involves evaluating the strategic significance, integration challenges, and security implications of adopting LCNC technologies (Manci, 2024).

Intrinsic Motivation for Secure Development – The internal drive of developers to prioritize security in their work, regardless of external pressures or incentives. This concept is relevant in encouraging best practices in both LCNC and traditional coding environments (Ryan & Deci, 2000, cited in Singer, 2017).

**11. CONCLUSION:**  
The adoption of Low Code No Code (LCNC) platforms in the software development industry offers significant benefits, such as increased efficiency and accessibility. However, this research has highlighted that LCNC platforms also introduce critical cyber security challenges, including insecure coding practices, third-party component risks, lack of visibility and control, shadow IT, and misconfigurations. With the rising adoption of LCNC in regions like South Africa, where cyber security capabilities are still maturing, addressing these vulnerabilities is crucial.

This study aimed to equip young IT developers in Gauteng with the knowledge to navigate the cyber security risks associated with LCNC platforms. By comparing the security effectiveness of LCNC applications with traditional hand-coded applications, we provided insights to help developers make informed decisions. Ultimately, while LCNC platforms offer substantial productivity gains, a strong focus on security is essential to ensure their safe and effective use in the software development industry.

REFERENCE LIST   
  
Alamin, M.A.A. *et al.* (2021) *(PDF) an empirical study of developer discussions on low ...*, *An Empirical Study of Developer Discussions on Low-Code Software Development Challenges*. Available at: https://www.researchgate.net/publication/352805735\_An\_Empirical\_Study\_of\_Developer\_Discussions\_on\_Low-Code\_Software\_Development\_Challenges (Accessed: 13 May 2024).   
  
Carielli, S., Bratincevic, J. and Rymer, J. (2020) *Low-code development requires a security rethink*, *Forrester*. Available at: https://www.forrester.com/blogs/low-code-development-requires-a-security-re-think/ (Accessed: 17 May 2024).

Hon, F.P. (2023) *Top 5 cyber security risks for LCNC (low-code/no-code) platform.*, *LinkedIn*. Available at: https://www.linkedin.com/pulse/top-5-cyber-security-risks-lcnc-low-codeno-code-platform-fun-ping-hon/ (Accessed: 15 May 2024).

Manci, Y. (2024) *The rise of low-code and no-code platforms*, *Business Tech Africa*. Available at: https://www.businesstechafrica.co.za/technology/2024/05/06/the-rise-of-low-code-and-no-code-platforms/ (Accessed: 15 May 2024).   
  
Nordic, eSystems (2024) *The truth about low-code technology and Data Security*, *Medium*. Available at: https://esystemsnordic.medium.com/the-truth-about-low-code-technology-and-data-security-2201050fc6f5#:~:text=“Applications%20built%20on%20low-code,applications%20built%20with%20their%20tooling.” (Accessed: 17 May 2024).   
  
OWASP (2024) *Owasp low-code/no-code top 10*, *OWASP Low-Code/No-Code Top 10 | OWASP Foundation*. Available at: https://owasp.org/www-project-top-10-low-code-no-code-security-risks/ (Accessed: 17 May 2024).   
  
Quickflow (2023) *Unveiling the essence of good practices in low-code platforms*, *LinkedIn*. Available at: https://www.linkedin.com/pulse/unveiling-essence-good-practices-low-code-platforms-tpcdf/ (Accessed: 17 May 2024).   
  
Quixy (2023) *LCNC: Is it completely worth the hype?*, *Quixy*. Available at: https://quixy.com/blog/lcnc-is-it-worth-the-hype/ (Accessed: 16 May 2024).   
  
TSBS Team, N.E. (2024) *The looming threat: Cybersecurity and small to medium businesses in South Africa - The Small Business Site*, *The Small Business Site - Top stories of the week*. Available at: https://www.thesmallbusinesssite.co.za/the-looming-threat-cybersecurity-and-small-to-medium-businesses-in-south-africa/ (Accessed: 15 May 2024).   
  
Ven, G.V. de (2023) *Low-code/no-code security risks: Challenges and solutions for protecting LCNC Apps*, *Conquer your risk*. Available at: https://www.conquer-your-risk.com/2023/01/11/low-code-no-code-lcnc-security-risks-and-challenges/ (Accessed: 15 May 2024).

Wikipedia (2024) *Low-code development platform*, *Wikipedia*. Available at: https://en.wikipedia.org/wiki/Low-code\_development\_platform (Accessed: 15 May 2024).