UNIVERSITY
INSTITUTE
OF LISBON

How Low-Code Development Platforms may mitigate Shadow IT issues

Alexsander Lemos Wilkens

Master in Digital Technologies for Business

Supervisor:
Ph.D. Rúben Filipe de Sousa Pereira,
Assistant Professor,
ISCTE - University Institute of Lisbon

January, 2026

Departamento de Ciências e Tecnologias da Informação

How Low-Code Development Platforms may mitigate Shadow IT issues

Alexsander Lemos Wilkens

Master in Digital Technologies for Business

Supervisor:
Ph.D. Rúben Filipe de Sousa Pereira,
Assistant Professor,
ISCTE - University Institute of Lisbon

January, 2026

To those with unfulfilled academic dreams, this thesis symbolizes that through dedication and perseverance, achievement is accessible to all.

Acknowledgments

As I embarked on this academic journey, I encountered extraordinary people along the way who were essential in the completion of this work. It is with deep appreciation and gratitude that I dedicate this space to acknowledge their invaluable support.

First and foremost, I would like to express my sincere gratitude to my Supervisor, Doctor Rúben Pereira, not only for the knowledge he has provided but also for his constant support, guidance, and valuable feedback, which were the backbone of this work. His constant availability and encouraging words were essential at every stage of this path. Without his guidance, this journey would have been infinitely more challenging.

I extend my gratitude to the domain experts who participated in the human validation process, dedicating their professional knowledge and time to review and validate the assessment results. Their practical experience with enterprise low-code platforms was essential in ensuring the accuracy and reliability of this research.

I owe a special thank you to my colleagues at ISCTE. Their support, motivational words, and shared moments were essential to maintaining my spirit and determination throughout this journey.

Finally, I thank my family, each of whom I love with my heart and soul, and who have always understood my absences, necessary for this little dream to become a reality.

Resumo

O uso crescente de soluções de TI não autorizadas, conhecido como Shadow IT, representa um desafio significativo para grandes organizações, comprometendo a segurança, a conformidade regulatória e a governança dos sistemas de informação. Esta dissertação investiga como as plataformas de desenvolvimento low-code (LCDPs) podem mitigar os riscos associados ao Shadow IT, permitindo o desenvolvimento rápido e governado de aplicações empresariais com participação tanto de desenvolvedores profissionais quanto de desenvolvedores cidadãos. A investigação adota uma metodologia de Design Science Research em três fases: (1) uma revisão sistemática da literatura seguindo o protocolo PRISMA, que identificou os principais desafios do Shadow IT e as capacidades das LCDPs; (2) o desenvolvimento de um framework de avaliação estruturado com 62 itens organizados em dez clusters temáticos; e (3) a aplicação do framework através de triangulação multi-LLM, utilizando quatro modelos de linguagem de grande escala (ChatGPT, Claude, Gemini e Grok) como avaliadores independentes, complementada por validação de especialistas humanos. A avaliação abrangeu quatro plataformas líderes de mercado: Mendix, OutSystems, Appian e Microsoft Power Apps. Os resultados demonstram que o OutSystems obteve a pontuação mais elevada, seguido pelo Appian, Power Apps e Mendix. A metodologia de triangulação alcançou 86,3% de concordância entre avaliadores, com validação humana aplicada aos 13,7% de itens com divergência significativa. Lacunas críticas foram identificadas nas capacidades de DevSecOps do Mendix e do Power Apps. Este estudo contribui com um framework de avaliação teoricamente fundamentado e uma abordagem inovadora para avaliação assistida por IA, oferecendo orientações práticas para organizações na seleção de LCDPs.

Palavras-chave: Shadow IT, plataformas low-code, governança de TI, desenvolvedores cidadãos, DevSecOps, Design Science Research.

Códigos ACM: Software and its engineering → Integrated and visual development environments; Information systems → Enterprise information systems.

Abstract

The growing adoption of unauthorized IT solutions, known as Shadow IT, poses significant challenges for large organizations, compromising security, regulatory compliance, and information systems governance. This thesis investigates how Low-Code Development Platforms (LCDPs) can mitigate Shadow IT risks by enabling rapid, governed enterprise application development involving both professional developers and citizen developers. The research adopts a three-phase Design Science Research methodology: (1) a PRISMA-compliant systematic literature review identifying key Shadow IT challenges and LCDP capabilities; (2) development of a structured 62-item evaluation framework organized across ten thematic clusters; and (3) framework application through multi-LLM triangulation, employing four large language models (ChatGPT, Claude, Gemini, and Grok) as independent evaluators, complemented by human expert validation. The evaluation encompassed four market-leading platforms: Mendix, OutSystems, Appian, and Microsoft Power Apps. Results demonstrate that OutSystems achieved the highest overall score, followed by Appian, Power Apps, and Mendix. The triangulation methodology achieved 86.3% inter-rater agreement, with human validation applied to the 13.7% of items exhibiting significant disagreement. Critical capability gaps were identified in DevSecOps integration for both Mendix and Power Apps. This study contributes a theoretically-grounded evaluation framework and a novel methodological approach to AI-assisted assessment, providing practical guidance for organizations seeking to select LCDPs for Shadow IT mitigation.

Keywords: Shadow IT, low-code platforms, IT governance, citizen developers, DevSecOps, Design Science Research.

ACM Classification Codes: Software and its engineering → Integrated and visual development environments; Information systems → Enterprise information systems.

Index

Resumo i
Abstract iii
Tables list ix
Figures list xi
Introduction 1
1.1. Context 1
1.2. Background 1
1.3. Problem Statement 2
1.4. Research Questions 2
1.5. Objectives 3
1.6. Scope and Limitations 3
1.7. Thesis Structure 4
Research Methodology Overview 7
2.1. Research Design 7
2.2. Methodological Classification and Validity Framework 7
2.3. Multi-Rater Triangulation Approach 8
2.4. Role of Human Expert Validation 8
2.5. Scope of Validity Claims 8
2.6. Clarification on Data Collection Methods 9
Literature Review 11
3.1. Methodology 11
3.2. The Landscape, Definition, and Impact of Shadow IT 13
3.3. Drivers of Shadow IT in Large Enterprises 14
3.4. Low-Code Development Platforms: An Overview 15
3.5. Key Features and Capabilities for Shadow IT Mitigation 15
3.6. Mitigation of Shadow IT 17
3.7. Existing Research and Gaps 17
Low-Code Platform Features for Shadow IT Mitigation 19
4.1. Summary of Primary Challenges Driving Shadow IT Adoption 19
4.2. Key Features and Capabilities LCDPs Must Possess 20
4.2.1. Rapid Development and Agility 20

4.2.2. Governance and Security Control Mechanisms 22
4.2.3. Collaborative Development Support 24
4.2.4. DevSecOps and Compliance Integration 24
4.2.5. Role Separation and Domain Specialization 25
4.2.6. Data Management and Interoperability 26
4.2.7. Platform Characteristics and Market Position 27
4.2.8. Platform Specialization and Customization 27
4.2.9. Limitations and Constraints 28
4.3. Mapping Challenges to Platform Capabilities 29
Evaluation Framework Development 33
5.1. Thematic Clusters 33
5.2. Evaluation Items by Cluster 34
Framework Application and Assessment 35
6.1. Selecting Low-Code Platforms 35
6.2. Selecting LLM Platforms 38
6.3. Defining the Universal Prompt and Data Input 38
6.4. Challenges 42
6.5. Output Validation 42
6.6. The Final Assessment Output 43
6.7. Human Validation 44
Assessment Results 47
7.1. Summary Results by Platform 47
7.2. Inter-LLM Agreement Analysis 48
7.3. Distribution of Rating Differences 49
7.4. Section-Level Disagreement Patterns 50
7.5. Human Validation Findings 51
7.6. Cross-Platform Comparative Analysis 52
7.7. LLM Reliability Analysis 54
Discussion 57
8.1. Addressing the Research Questions 57
8.2. Key Findings 58
8.3. Theoretical Contributions 59
8.4. Main Conclusions 60
8.5. Recommendations 61

8.6. Limitations 63
8.7. Future Research 64
References 67
Appendices 71
Appendix A 73
Appendix B 75
Appendix C 91
Appendix D 93

Tables list

Table 1 – Search keywords and results by database 11
Table 2 – Inclusion and exclusion criteria 11
Table 3 – Relevant BMIT Critical Success Factors for LCDP Governance 23
Table 4 – RAAFT Framework Summary 25
Table 5 – LCDP Limitations Framework 28
Table 6 – Challenge-to-Feature Mapping Matrix 30
Table 7 – Cross-Analyst Platform Positioning (2025) 36
Table 8 – Assessment Prompt Architecture 39
Table 9 – Expert validator profiles 44
Table 10 – Assessment Summary by Thematic Cluster 47
Table 11 – Inter-LLM Agreement Metrics by Platform 49
Table 12 – Distribution of Maximum Rating Differences 49
Table 13 – Section-Level Disagreement Analysis 50
Table 14 – Items Requiring Human Validation 51
Table 15 – Platform Capability Leadership by Section 53
Table 16 – Average Ratings by LLM Across All Platforms 55

Figures list

Figure 1 – Flow diagram of the literature search 12
Figure 2 – Assessment Execution Workflow 40

Glossary of abbreviations and acronyms

Acronym	Full Term
ACM	Association for Computing Machinery
AI	Artificial Intelligence
API	Application Programming Interface
BMIT	Business-Managed IT
BU	Business Unit
CI/CD	Continuous Integration/Continuous Deployment
CRM	Customer Relationship Management
CSF(s)	Critical Success Factor(s)
CSV	Comma-Separated Values
DAST	Dynamic Application Security Testing
DevSecOps	Development, Security, and Operations
ERP	Enterprise Resource Planning
GDPR	General Data Protection Regulation
GUI	Graphical User Interface
HIPAA	Health Insurance Portability and Accountability Act
IDC	International Data Corporation
IEEE	Institute of Electrical and Electronics Engineers
IT	Information Technology
JSON	JavaScript Object Notation
LCDP	Low-Code Development Platform
LCNC	Low-Code/No-Code
LLM	Large Language Model
MFA	Multi-Factor Authentication
NCDP	No-Code Development Platform
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
RAAFT	Risks, Actors, Actions, Feedback, Tools
RBAC	Role-Based Access Control

Acronym	Full Term
SaaS	Software as a Service
SAST	Static Application Security Testing
SDLC	Software Development Life Cycle
SQL	Structured Query Language
UI	User Interface
WoS	Web of Science
WYSIWYG	What You See Is What You Get
XML	Extensible Markup Language

CHAPTER 1

Introduction

1.1. Context

Today's enterprises face the challenge of maintaining strong security protocols, adhering to regulatory standards, and optimizing resource allocation amidst the prevalence of Shadow IT. Shadow IT, characterized by business units independently developing or procuring software solutions without IT department involvement, often arises when traditional IT structures struggle to meet rapidly changing and diverse business application needs. This trend introduces significant risks, including security vulnerabilities, compliance breaches, scalability limitations, and system integration complexities. As Ploder et al. [1] note, Shadow IT bears several risks on security, manageability, and compliance.

Low-Code Development Platforms (LCDPs) offer a promising avenue for mitigating these challenges. By simplifying conventional coding complexities through user-friendly visual interfaces and pre-configured components, LCDPs enable both IT professionals and citizen developers to swiftly create applications. This accelerated development approach expedites project timelines, reduces the burden on traditional IT departments, and fosters enhanced collaboration between business users and IT personnel. These platforms empower organizations to more effectively align application development with specific business requirements, thereby addressing the fundamental causes of Shadow IT [2]. This study aims to comprehensively assess the effectiveness of LCDPs in resolving Shadow IT issues by analysing their potential to support the collaborative development of resilient, enterprise-level applications.

1.2. Background

LCDPs are known to accelerate application development, reduce costs, and enhance organizational agility. With visual development environments, drag-and-drop interfaces, pre-built connectors, and automated workflows they facilitate rapid application creation, democratizing development for a wider range of users. These platforms hold significant promise for delivering enterprise-grade control without stifling innovation. However, the implementation of LCDPs necessitates careful planning and oversight. Without proper governance, low-code initiatives can inadvertently exacerbate Shadow IT risks. Empowering business units while retaining IT oversight is the main goal, ensuring that innovation does not compromise security or compliance. Safeguarding organizational security and compliance is possible through effective governance policies and monitoring mechanisms and therefore essential to realize the benefits of LCDPs. This includes establishing clear guidelines for application development,

security protocols, and data management, alongside implementing monitoring tools. Furthermore, the composition of the developer base, including the ratio of IT professionals to citizen developers, influences platform implementation [3]. Information collected within LCDPs can identify and address security issues, ensuring citizen developers adhere to maintaining the enterprise security standards [2]. The core challenge, therefore, is to identify a structured, enterprise-grade approach that enables fast, secure, and scalable application development while ensuring centralized IT governance.

Despite the recognized potential of LCDPs for Shadow IT mitigation, organizations currently lack systematic methods to evaluate whether specific platforms possess the features and capabilities necessary for effective governance. The absence of a structured evaluation framework forces IT decision-makers to rely on vendor claims and general market positioning rather than evidence-based assessment of platform capabilities against documented Shadow IT challenges. This gap suggests the need for an operationalized evaluation instrument that maps specific platform features to the governance, security, and compliance requirements identified in the academic literature.

1.3. Problem Statement

Traditional IT approaches frequently struggle to keep pace with the rapidly evolving demands of business units, leading to a divide between IT and business users. This disconnect often results in the proliferation of unapproved Shadow IT solutions, which introduce detrimental consequences such as redundant systems, data silos, heightened security risks, and compliance vulnerabilities. While LCDPs offer significant potential to bridge this gap by enabling rapid and scalable application development, their adoption introduces a critical challenge: achieving a harmonious balance between fostering innovation and maintaining rigorous IT governance. If not properly managed, LCDPs could inadvertently exacerbate the very Shadow IT risks they aim to mitigate. Therefore, the core problem this research addresses is how to implement a structured, enterprise-grade approach for LCDPs adoption that simultaneously enables fast, secure, and scalable application development while ensuring centralized IT governance, thereby allowing organizations to foster innovation without compromising security or compliance.

1.4. Research Questions

Based on the academic and industry literature reviewed above, this research aims to answer the following key questions:

- What are the primary challenges associated with software development and Shadow IT in large enterprises?

- What specific key features and capabilities must LCDPs possess to effectively address these challenges?
- What evaluation framework using Likert scale methodology should organizations employ to systematically compare and assess LCDP vendors against the key features and capabilities identified?

1.5. Objectives

This study has the following primary objectives:

- To identify, through a comprehensive literature review, the core issues and risks associated with Shadow IT in large enterprises. This will involve defining Shadow IT, detailing its causes, consequences, and impact on organizational efficiency and security;
- To determine the essential features and capabilities that LCDPs should possess to effectively mitigate Shadow IT concerns, based on the identified issues and risks;
- To develop a comprehensive evaluation framework comprising a structured questionnaire with Likert scale assessment criteria that organizations can systematically employ to compare and evaluate LCDP vendors against the identified essential features and capabilities. This framework enables evidence-based vendor selection aligned with organizational Shadow IT mitigation strategies;

1.6. Scope and Limitations

This research focuses on evaluating leading enterprise Low-Code Development Platforms relevant to large organizations. Platform selection employs cross-analyst consensus methodology, triangulating rankings from multiple independent analyst firms to identify market leaders suitable for enterprise Shadow IT mitigation. The evaluation targets general-purpose LCDPs designed as blank-slate development environments, excluding domain-specific platforms that presuppose particular functional orientations.

The assessment methodology employs multi-LLM triangulation, wherein multiple large language models independently evaluate platforms against an evaluation framework derived from the systematic literature review. Human expert validation addresses items exhibiting significant inter-model disagreement. Assessment evidence draws from publicly available vendor documentation, analyst reports, academic literature, and practitioner publications.

Several limitations warrant acknowledgment. The LLM-assisted methodology, while reducing single-model bias, may introduce reliability concerns requiring validation protocols. The assessment relies on secondary sources rather than hands-on platform evaluation. The evaluation framework,

while derived systematically from academic literature, reflects interpretive decisions that may introduce subjectivity. LCDP capabilities evolve rapidly, potentially limiting long-term applicability of findings.

This research explicitly excludes Shadow IT's behavioural, financial, legal, and regulatory dimensions, focusing solely on technical platform capabilities. The emerging phenomenon of Shadow AI, while extending the Shadow IT paradigm, falls outside primary scope. Generalizability may require validation through hands-on testing in specific organizational contexts.

1.7. Thesis Structure

This thesis is organized into eight chapters, outlined as follows:

Chapter 1: Introduction provides the context for the study, outlining the problem of Shadow IT and the potential of LCDPs as a solution. It details the research background, problem statement, research questions, objectives, and the scope and limitations of the study.

Chapter 2: Research Methodology establishes the methodological foundation for the research. The chapter presents the overall research design, clarifies the methodological classification and validity framework, describes the multi-rater triangulation approach, explains the role of human expert validation, and articulates the scope of validity claims. This chapter addresses the distinction between traditional quantitative survey validity and the Design Science Research evaluation approach employed in this study.

Chapter 3: Literature Review presents a systematic review of existing academic and industry literature on Shadow IT and LCDPs, following the PRISMA methodology. This chapter establishes the theoretical foundation for the research by defining Shadow IT, examining its drivers and organizational impacts, providing an overview of Low-Code Development Platforms, discussing key features relevant to Shadow IT mitigation, and identifying gaps in existing research.

Chapter 4: Low-Code Platform Features for Shadow IT Mitigation addresses the first two research questions. The chapter identifies the primary challenges associated with software development and Shadow IT in large enterprises, and elaborates on the specific features and capabilities LCDPs must possess to effectively address these challenges, organized across nine capability domains.

Chapter 5: Evaluation Framework Development addresses the third research question by operationalizing the challenges and capabilities identified in Chapter 4 into a structured evaluation framework. The chapter organizes the evaluation criteria into thematic clusters and details the individual assessment items designed for systematic LCDP vendor evaluation.

Chapter 6: Framework Application and Assessment applies the evaluation framework developed in Chapter 5 to leading enterprise LCDPs. The chapter details platform selection criteria, describes the

methodology for conducting assessments using multiple large language model platforms, and presents the data collection and validation processes.

Chapter 7: Assessment Results presents the consolidated assessment results from the systematic evaluation of enterprise LCDPs. The chapter reports platform performance across thematic clusters, analyses inter-LLM agreement patterns, documents human validation findings, and provides cross-platform comparative insights.

Chapter 8: Discussion synthesizes the research findings, discusses their implications for organizations seeking to leverage LCDPs for Shadow IT mitigation, acknowledges the study's limitations, and provides actionable recommendations for practitioners. The chapter concludes by suggesting avenues for future research in this evolving domain.

CHAPTER 2

Research Methodology Overview

This chapter establishes the methodological foundation for the research, detailing the overall research design and the validity framework that underpins the study. Given the novel nature of employing large language models as systematic evaluators of enterprise software platforms, careful attention to methodological classification and validity criteria is essential to ensure academic rigor and defensibility.

The chapter addresses a fundamental methodological consideration: while this research employs structured rating scales and generates numerical scores, it does not constitute traditional quantitative survey research and therefore should not be evaluated against sample-size-dependent validity criteria designed for population sampling. Instead, the methodology aligns with Design Science Research traditions and multi-rater triangulation approaches, where validity derives from systematic process, inter-rater agreement, and expert judgment rather than from statistical inference to broader populations.

2.1. Research Design

This research employs a multi-phase design science methodology comprising three sequential phases. The first phase consists of a qualitative systematic literature review following the PRISMA methodology to identify Shadow IT challenges and LCDP capabilities. The second phase involves framework development through AI-assisted synthesis of literature findings into a structured evaluation instrument. The third phase applies the framework through multi-rater platform assessment using large language models as independent evaluators, with human expert validation for items exhibiting significant inter-rater disagreement.

2.2. Methodological Classification and Validity Framework

It is essential to clarify the methodological classification of this research, as the evaluation approach differs fundamentally from traditional quantitative survey research. The evaluation instrument functions as a structured assessment rubric for technical capability evaluation, not as a survey instrument measuring human attitudes or perceptions. Traditional quantitative validity thresholds requiring minimum sample sizes (conventionally $N \geq 30$ for statistical generalization) apply specifically to surveys sampling human populations where the goal is to infer population-level characteristics from a representative sample [4]. This research employs a fundamentally different evaluative paradigm.

The methodology aligns with Design Science Research evaluation protocols, where artifact assessment prioritizes systematic documentation, structured evaluation criteria, and expert judgment over statistical sampling [5], [6]. In design science research, the validity of an evaluation artifact is established through demonstration of utility, internal consistency, and alignment with theoretical foundations rather than through sample-size-dependent statistical inference.

2.3. Multi-Rater Triangulation Approach

The platform assessment phase employs multi-rater triangulation rather than survey sampling. Multiple independent large language models assess all evaluation items across all selected platforms. This design parallels established inter-rater reliability protocols used in content analysis and qualitative research, where multiple coders independently assess the same material to establish coding consistency. The critical validity metric in such designs is not sample size but inter-rater agreement, measured through intraclass correlation coefficients (ICC) that quantify the degree to which independent raters produce consistent assessments when applying the same criteria to the same content [7].

The validity threshold for this methodology follows the 70% agreement rate commonly accepted for adequate inter-rater reliability in exploratory research [8]. Items exhibiting significant disagreement are subjected to human expert validation, ensuring that ambiguous cases receive informed adjudication rather than mechanical averaging. The specific selection of LLM platforms, evaluated LCDPs, and assessment results are detailed in subsequent chapters.

2.4. Role of Human Expert Validation

Human domain experts participating in validation are not analogous to survey respondents in quantitative research. Their role is methodologically distinct: they function as expert adjudicators performing quality assurance on automated assessment outputs, comparable to the role of senior coders resolving disagreements in qualitative content analysis or expert panellists in Delphi studies. This approach is consistent with expert panel methodologies documented in technology assessment literature, where small panels of qualified domain experts provide sufficient evaluative validity when combined with structured criteria and documented rationale [9]. The validity of expert panel assessment derives from participant expertise and systematic process rather than from statistical sampling of a broader population.

2.5. Scope of Validity Claims

This research does not claim statistical generalizability to a population of platforms or organizational contexts. Such claims would indeed require different methodological approaches with larger samples.

Rather, the study claims analytical validity: the systematic application of a theoretically-grounded evaluation framework to a purposively selected set of market-leading platforms, with multi-rater triangulation ensuring assessment reliability. The findings describe the capabilities of the assessed platforms as documented in available sources during the assessment period, providing evidence-based comparative insights for organizations evaluating these platforms for Shadow IT mitigation purposes.

2.6. Clarification on Data Collection Methods

It is important to clarify that this research does not employ traditional interview methodology or survey-based data collection. The rating scale is used for structured platform capability assessment (evaluating the presence, maturity, and enterprise-readiness of specific technical features) rather than measuring human attitudes, perceptions, or opinions. The human validation component involves domain experts reviewing and adjudicating LLM assessment outputs where inter-model disagreement exceeds acceptable thresholds; this constitutes expert review validation, not interview-based or survey-based data collection. The research design therefore represents a novel methodological approach combining systematic literature review (qualitative) with LLM-assisted platform assessment (structured evaluation) and expert validation (qualitative verification).

The methodological foundation established here supports the subsequent chapters. Chapter 3 presents the systematic literature review, identifying Shadow IT challenges and LCDP capabilities that inform the evaluation framework. Chapter 4 synthesizes these findings into the specific platform features required for Shadow IT mitigation. Chapter 5 operationalizes these features into the evaluation framework. Chapter 6 applies the framework to assess market-leading platforms using the multi-LLM triangulation methodology described in this chapter. Chapter 7 presents and analyses the assessment results.

CHAPTER 3

Literature Review

3.1. Methodology

The literature review for this study was conducted using the PRISMA methodology, a widely recognized and endorsed framework for systematic reviews. PRISMA provides explicit and stringent guidelines for the planning, execution, and analysis phases of systematic reviews, ensuring transparency, reproducibility, and high quality. Adhering to the PRISMA checklist mitigates bias, enhances the rigor of the review, and bolsters the credibility of the findings.

The literature search spanned three databases (Scopus, Web of Science (WoS), IEEE Xplore). The search strategy employed a combination of keywords related to Shadow IT, Low-Code, and citizen developers. The search was refined to include publications from 2022 onwards to capture the most current research. Table 1 summarizes the search queries and initial results from the primary academic databases and Table 2 summarizes the eligibility and exclusion criteria for full-text screening:

Table 1 – Search keywords and results by database

Database	Keyword 1: "Shadow IT" (2022-2025)	Keyword 2: "Shadow IT" AND "Low code" (2022-2025)	Keyword 3: "Shadow IT" AND "Citizen Developer" (2022-2025)	Keyword 4: "Low code" AND "Citizen Developer" (2022-2025)	Keyword 5: "Shadow IT" AND "Low code" AND "Citizen Developer" (2022-2025)
Scopus	68	4	1	52	1
IEEE	6	1	0	5	0
WoS	53	1	0	11	0

Table 2 – Inclusion and exclusion criteria

Inclusion Criteria	Exclusion Criteria
Sources addressing Shadow IT as a central theme.	Studies with an industry-specific context that do not demonstrate generalizable enterprise value.
Studies discussing diverse relations between LCDPs, Shadow IT, and Citizen Developer contexts.	Research not focused on the problematic of Software Development in Shadow It contexts.
	Posters or abstracts lacking substantial depth.

The initial search yielded 203 unique references. After removing 53 duplicates, 150 articles proceeded to title and abstract screening. During this phase, 62 articles were excluded for not being IT-related, not being relevant to Shadow IT or Low-Code, being too niche, or being of the wrong publication type. This resulted in 88 potentially relevant articles; of these, 45 full-text reports were not accessible. A full-text screening was then conducted on these 43 articles, leading to the exclusion of 10 more, culminating in 33 articles being included in the final literature review.

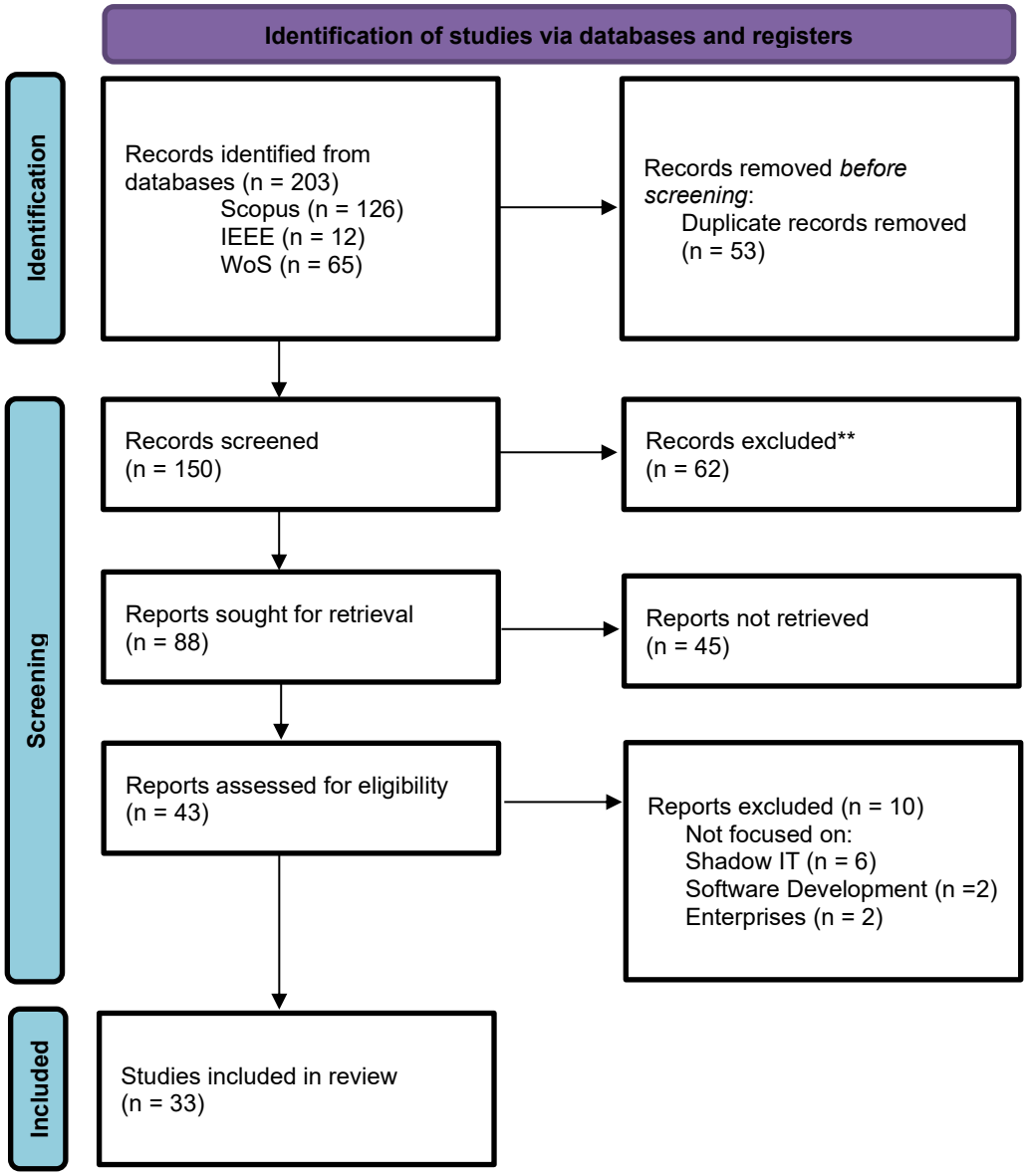


Figure 1 – Flow diagram of the literature search

Based on this review, the following key topics will be discussed in detail:

- The landscape, risks, and benefits of Shadow IT;
- Key features LCDPs should possess to mitigate Shadow IT;
- The suitability of LCDPs for enterprise use.

3.2. The Landscape, Definition, and Impact of Shadow IT

Shadow IT involves employees or departments using IT-related hardware, software, or services without the formal IT department's knowledge, approval, or oversight [10]. Several terms are used interchangeably in the literature to describe this phenomenon, including shadow systems, feral systems, IT workarounds, and unofficial IT [10]. This often occurs when traditional IT departments struggle to meet the diverse, rapidly changing, and immediate application demands of various business units [11]. Consequently, business units independently develop or acquire solutions to address their specific needs more quickly and efficiently [12]. Examples range from simple, single-user tools like unauthorized cloud storage accounts to complex, department-wide systems such as custom-built applications on public cloud infrastructure [11].

The impact of Shadow IT on organizations is multifaceted, presenting both potential benefits and significant risks. While Shadow IT can foster innovation and agility by enabling rapid prototyping and deployment of solutions tailored to specific business needs, potentially increasing productivity and accelerating time-to-market [12], its risks are often more pronounced. These risks include security vulnerabilities, as applications and systems operating outside IT's purview may lack rigorous security protocols, creating potential entry points for malicious actors and increasing the risk of data breaches [13]. Sensitive information stored in unsecured Shadow IT systems is particularly vulnerable [13]. According to one study, approximately 50% of IT managers are concerned about the breakdown of mission-critical Shadow IT [13].

Compliance violations are another concern, as Shadow IT systems often fail to adhere to regulatory requirements (e.g., GDPR, HIPAA) or internal policies, potentially leading to legal and financial penalties [12]. Data silos and integration complexities arise from the proliferation of unintegrated systems, hindering enterprise-wide data management, analysis, and collaboration. Integrating Shadow IT solutions with existing enterprise systems can be complex, costly, and can disrupt business agility [14]. This can also lead to wasted IT spending and redundancy as departments may independently purchase or develop redundant solutions, leading to increased costs and inefficient resource allocation. The informality of Shadow IT complicates IT governance and control, allowing Shadow IT systems to proliferate uncontrollably, making it difficult for IT to maintain visibility and control over the organization's IT landscape [12].

Despite some perceived benefits, the prevalence of data duplication, compliance issues, inadequate backups, and wasted resources underscores the serious challenges posed by Shadow IT. It is often a symptom of an IT department struggling to meet evolving user needs, with employees resorting to personal initiatives when they perceive formal IT processes as unresponsive or slow [12]. Organizations often struggle to recognize Shadow IT because it is developed outside the official

organizational framework [12], posing significant information security risks when not properly managed. Furthermore, sensitive data stored in unsanctioned locations may be inadvertently exposed, and critical data may not be captured by official backup systems [14].

3.3. Drivers of Shadow IT in Large Enterprises

Shadow IT emerges in large enterprises as a consequence of several interconnected factors. Employees often seek solutions outside official IT channels when they perceive inadequacies in the responsiveness and resources provided by the central IT function [10]. A significant driver is the business units' need for rapid solutions to immediate problems. Employees, when faced with pressing deadlines, may bypass formal processes and implement their own solutions, often leveraging readily available cloud services or SaaS applications. This behaviour becomes more pronounced when the official IT infrastructure is perceived as slow or overly bureaucratic [15].

Furthermore, a lack of IT resources can lead to Shadow IT adoption. Even when IT departments are willing to provide solutions, they may be constrained by limited budgets or competing priorities [13]. Consequently, project delivery is delayed or business needs are not adequately addressed, thus prompting employees to independently seek alternative solutions. In such cases, Shadow IT becomes a way to compensate for perceived deficiencies in the official IT infrastructure [14].

Finally, the empowerment of business users fosters Shadow IT adoption. As employees become more tech-savvy and have access to user-friendly technologies, they may feel capable of handling IT matters themselves [13]. While this empowerment can lead to innovation, it can also introduce security vulnerabilities and compliance issues if not properly governed [10]

Beyond resource constraints, employees often adopt workarounds because official systems are "fit to specification" but not "fit to purpose." [16] argues that workarounds emerge as innovative responses to rigid systems failing to match actual workflow realities, driving employees to create alternative solutions outside IT governance. This paradox of efficiency is evident in hospital settings where digital tools intended to streamline work introduce new administrative burdens, pushing professionals toward shadow solutions that promise immediate relief [17].

Further research identifies distinct profiles among Shadow IT adopters; many are motivated, risk-aware individuals compensating for perceived IT inertia rather than acting with malicious intent [18]. A critical governance gap exists where business units lack sanctioned digital outlets for their needs. [19] emphasize that successful mitigation requires Business-Managed IT (BMIT) frameworks where business units and IT share governance rights, rather than IT enforcing strict prohibitions.

To effectively manage Shadow IT drivers, organizations must strategically align IT services with business needs, foster bilateral governance models, and encourage both innovation and risk mitigation through collaborative oversight [10], [19].

3.4. Low-Code Development Platforms: An Overview

LCDPs represent a significant shift in software application development by democratizing the process and enabling individuals with domain expertise, rather than extensive IT skills, to build and deploy solutions [20]. They transition development from traditional syntax-based coding to a visual modelling environment utilizing graphical user interfaces, pre-built components, and configuration options, making application design intuitive even for non-programmers through drag-and-drop elements representing logic and workflows [20]. These platforms facilitate seamless data integration from various sources, empowering citizen developers to leverage existing organizational data and promote reusability via templates and component blocks that accelerate development [20].

Compared to traditional systems development, which involves numerous labour-intensive tasks, LCDPs significantly reduce such efforts by hiding and automating complex coding, testing, and deployment activities through visual interfaces and configurations [21]. This integration fosters increased innovation as citizen developers rapidly prototype and deploy solutions addressing specific domain challenges, supported by their deep business understanding [20]. Consequently, organizations gain agility, enabling faster internal custom application development and quicker responses to shifting market demands [20]. Furthermore, LCDPs support cloud-based solution development without traditional text-based coding, accelerating enterprise-grade application delivery [1].

Empirical validation by Varajão et al. [22] confirms this paradigm shift, demonstrating through controlled experiments that Low-Code and Extreme Low-Code platforms substantially outperform traditional coding in productivity, helping organizations clear development backlogs that often drive Shadow IT proliferation [23]. LCDPs must be implemented with effective governance structures ensuring that platform capabilities support both rapid development and adequate organizational control.

3.5. Key Features and Capabilities for Shadow IT Mitigation

For LCDPs to effectively address Shadow IT, they must possess specific features and capabilities that ensure secure, compliant, and integrated application development within an enterprise context. To that end, these features empower both business users and IT professionals to collaboratively develop robust applications tailored to the needs of large-scale organizations within a controlled environment.

As Sedrakyan et al. [2] describes, Visual development environments, with intuitive drag-and-drop interfaces, accelerate development and enable citizen developers to contribute without extensive coding knowledge. Furthermore, pre-built connectors facilitate seamless integration with existing enterprise systems, databases, and third-party services, avoiding data silos and ensuring interoperability. Moreover, automated workflows enable the creation of automated business processes, streamlining operations, reducing manual effort, and enhancing efficiency. To ensure security, robust user authentication and authorization mechanisms prevent unauthorized access to sensitive data and systems, especially in large enterprises where data security is paramount. This includes support for multi-factor authentication and granular role-based access control. Additionally, comprehensive data management capabilities ensure data integrity, consistency, and security across applications, including data validation, storage, and retrieval functionalities. Scalability features, of course, provide the ability to scale applications to meet growing user demands and data volumes without compromising performance or stability. For security, built-in security controls, secure coding practices, vulnerability scanning, and robust security frameworks protect applications from threats. Finally, regulatory compliance support includes features that help organizations fulfil industry-specific regulations and data privacy laws, mitigating compliance violations commonly associated with Shadow IT.

The effective integration of LCDPs within an organization is intrinsically linked to robust IT governance. To ensure security, compliance, and seamless integration with existing systems, organizations must establish clear governance policies and guidelines for the use of LCDPs. This involves a dynamic approach that balances flexibility and standardization rather than treating them as opposing choices. LCDPs, by their nature, can facilitate adherence to standard governance practices, positioning them as enablers of flexibility and responsiveness [2]. However, if the chosen LCDPs does not offer sufficient built-in safeguards, firms must implement complementary governance mechanisms, including clear standards for application design, coding conventions, and documentation. For instance, granular control should be in place over who can develop, deploy, and manage applications on the platform. Furthermore, mandatory review processes for applications before deployment are needed, ensuring security, compliance, and alignment with business objectives. To effectively track application usage, performance, and security posture across the low-code environment, centralized monitoring tools and processes should be implemented.

Embracing new digital technologies and developing effective LCDPs governance strategies are crucial for organizations that cannot rely solely on their traditional, potentially inflexible, official IT departments. Indeed, the first principle of such a strategy often involves classifying existing low-code use cases to clarify responsibilities and maintain a balance between flexibility and standardization. As mentioned, Shadow IT bears several risks on security, manageability, and compliance [1]. As also

mentioned, LCDPs enable the development of cloud-based solutions without traditional coding in a text-based editor [1]. By assessing these capabilities, this research aims to provide insights into how effectively LCDPs can address the root causes of Shadow IT and enable large-scale organizations to develop robust applications while maintaining centralized IT governance.

3.6. Mitigation of Shadow IT

Strategies for mitigating Shadow IT typically encompass prevention, detection, and remediation. First, prevention involves creating clear and reasonable IT regulations, coupled with user education on cybersecurity policies and the risks associated with Shadow IT [10]. Furthermore, encouraging open communication about technology needs can also help prevent the rise of Shadow IT by ensuring the IT department is aware of and responsive to user requirements [18]. Second, detection involves assessing the cybersecurity risk posed by Shadow IT [14]. Finally, remediation may involve revising cybersecurity policies that are violated by Shadow IT if the outcome is beneficial and will not be sanctioned [14]. It's important to consider that sanctions could demotivate employees, as they often use Shadow IT to improve their work, rather than with malicious intent [10]. Therefore, this comprehensive approach not only mitigates the risks associated with unsanctioned IT solutions but also fosters a collaborative environment where business agility and IT governance coexist effectively [18].

Effectively tackling Shadow IT issues requires a comprehensive strategy that clarifies responsibilities, establishes clear guidelines, and maintains a dynamic balance between business unit agility and IT oversight. Shadow IT introduces risks to security, manageability, and compliance [1].

One approach involves recognizing the increasing role of business-managed IT, where IT task responsibility resides in the business units [19]. Effective governance for BMIT establishes a framework to align decision-making and regulate the execution of decisions, allocating specific decision rights for a BMIT instance to either the BU, the IT organization, or jointly [19]. This bilateral responsibility contrasts with the traditional IT-managed governance approach [19].

To achieve effective governance, organizations need to establish an enterprise-wide culture in favour of BMIT [19]. This involves both BU and IT organizations working together [19]. Employees' demand for BMIT or Shadow IT cannot be appropriately managed by simply prohibiting it [19]. Prohibition might lead to more Shadow IT with less controllable adverse effects than BMIT [19].

3.7. Existing Research and Gaps

Existing research on mitigating Shadow IT highlights the necessity of a structured, enterprise-grade approach to enable rapid, secure, and scalable application development, all while maintaining centralized IT governance. While LCDPs are recognized for their potential in addressing these

challenges, current literature lacks detailed identification and assessment of how specific features and capabilities of leading LCDPs can directly address the root causes and mitigate the risks associated with Shadow IT.

This gap underscores the need for an evaluation framework to help enterprise IT decision-makers align their unique needs with the mitigation capabilities offered by top LCDPs. By exploring this area, organizations can gain insights into managing Shadow IT risks while leveraging the benefits of LCDPs for rapid application development. Addressing this gap can significantly inform strategies for harnessing the potential of LCDPs to drive digital transformation while maintaining essential control and governance over the IT landscape, especially given the escalating risks of security, compliance, scalability, and system integration introduced by Shadow IT.

Having established the theoretical foundations and identified the research gap, Chapter 4 proceeds to synthesize these findings into a comprehensive framework of LCDP features required for Shadow IT mitigation.

CHAPTER 4

Low-Code Platform Features for Shadow IT Mitigation

This chapter synthesizes findings from the systematic literature review (Chapter 3) to derive a comprehensive framework of LCDP features and capabilities required to address Shadow IT challenges in large enterprises. While Chapter 3 established the theoretical foundations (defining Shadow IT, identifying its drivers, and introducing LCDPs as potential mitigation instruments) this chapter advances the analysis by summarizing the primary challenges that create conditions for Shadow IT emergence, systematically identifying and categorizing the specific LCDP features necessary to address these challenges, acknowledging platform limitations that constrain mitigation effectiveness, and mapping the relationship between identified challenges and mitigating platform capabilities. The feature framework developed herein serves as the analytical foundation for the platform evaluation framework operationalized in Chapter 5.

4.1. Summary of Primary Challenges Driving Shadow IT Adoption

Chapter 3 established that Shadow IT emerges from a confluence of organizational, technical, and behavioural factors. Three primary challenge domains were identified, which this section briefly recapitulates to establish the analytical baseline for feature identification.

IT Responsiveness Gap

The literature consistently identifies the inability of IT departments to meet business needs in a timely manner as the predominant Shadow IT driver [2], [14], [16]. Multiple studies confirm that employees adopt unauthorized solutions primarily to maintain productivity when formal IT processes prove unresponsive [11]. Haag and Eckhardt [14] document cases where “Shadow IT is used as a workaround to get business done until issues are resolved,” particularly in urgent situations where IT delays could result in missed opportunities.

This responsiveness deficit is compounded by resource constraints, competing priorities, and legacy system complexity that constrain IT departments’ capacity to deliver solutions at the pace business units require [19]. Alaba et al. [11] demonstrate that limited IT accessibility and work overload interact to predict Shadow IT usage, suggesting the phenomenon reflects systemic rather than individual factors.

Application Complexity and User Experience Deficits

Enterprise systems frequently suffer from insufficient consideration for end-user experience, presenting unnecessarily complex interfaces that employees perceive as obstacles rather than enablers [2], [16]. This “fit to specification but not fit to purpose” phenomenon drives employees toward workaround development, where official systems technically meet requirements but fail to match actual workflow realities [16], [17].

Alaba et al. [11] demonstrated that both workaround behaviour and Shadow IT usage positively relate to individual performance, suggesting employees rationally adopt these practices when official systems impede productivity. This finding underscores that Shadow IT often reflects legitimate unmet needs rather than mere policy violations.

Security, Compliance, and Governance Risks

Shadow IT proliferation introduces detrimental organizational consequences, including redundant systems, data silos, heightened security risks, and compliance vulnerabilities [2], [14]. Unauthorized applications operating outside IT governance may lack rigorous security protocols, creating potential entry points for malicious actors and increasing data breach risk [24].

Beyond direct security costs, Shadow IT creates technical debt as organizations operate untracked systems that complicate information management, governance, and eventual system consolidation [2], [10]. The covert nature of Shadow IT, operating outside IT visibility, complicates governance and impedes systematic risk management [19].

4.2. Key Features and Capabilities LCDPs Must Possess

This section systematically identifies the features and capabilities that LCDPs must possess to effectively address the Shadow IT challenges outlined in Section 4.1. The feature framework is derived from a synthesis of the literature reviewed in Chapter 3, organized into functional categories that align with the challenge domains.

4.2.1. Rapid Development and Agility

Addressing the IT responsiveness gap requires development paradigms that dramatically accelerate application delivery. LCDPs achieve this through visual modelling interfaces, pre-built components, and templates that reduce development time by 5-10x compared to traditional coding approaches [22]. While Varajão et al [22] note that "evidence of these improvements is practically non-existent in the research literature," their controlled laboratory experiments confirmed that low-code platforms substantially outperform traditional coding.

Key capabilities include drag-and-drop interface builders that enable rapid UI construction, pre-built workflow components that accelerate business logic implementation, template libraries that provide starting points for common application types, and iterative prototyping capabilities that facilitate rapid feedback cycles with business stakeholders [20], [23].

Complexity Reduction and Accessibility

LCDPs democratize application development by reducing technical barriers that traditionally excluded non-IT personnel. Visual development environments with WYSIWYG (What You See Is What You Get) editing enable real-time preview of application changes, reducing the abstraction gap between design and implementation [25].

Elshan et al. [26] provide a comprehensive definition of the citizen developer as "a technically inclined individual in an organization with domain knowledge and a drive for change, utilizing LC/NC platforms to create solutions across a broad spectrum of capabilities, organizationally independent of the IT-department." This definition emphasizes that citizen developers occupy a spectrum from basic "Makers" to advanced "Champions" who interface directly with IT professionals.

Guthardt et al. [25] conducted an empirical study comparing programmers and non-programmers using a no-code platform. Their results revealed no significant difference in task correctness or processing time between the two groups, suggesting that training concerns for citizen developers may be overstated. However, methodological limitations (n = 10, student sample, custom prototype platform) warrant caution in generalizing these findings.

Enterprise Integration and Scalability

Enterprise-grade LCDPs must provide robust integration capabilities to connect citizen-developed applications with existing organizational systems. Pre-built connectors for common enterprise platforms (ERP, CRM, databases), API management capabilities for custom integrations, and support for standard data exchange formats ensure that LCDP applications complement rather than fragment the enterprise architecture [27], [23].

Scalability considerations include support for multi-project management, capabilities for deploying applications across organizational units, and performance characteristics suitable for enterprise-scale usage. Tang [28] emphasizes that cloud-based architecture enables component-based approaches that promote flexibility and reusability across different organizational contexts.

4.2.2. Governance and Security Control Mechanisms

Critical to enterprise adoption, LCDPs must embed centralized IT governance capabilities while enabling the decentralized, agile development that addresses Shadow IT drivers. This section synthesizes governance requirements from multiple literature streams: technical access controls [2], critical success factors for business-managed IT [19], and LCNC governance guidelines [3].

Technical Governance Capabilities

LCDPs must provide multi-layered access controls including role-based access management defining permissions based on organizational roles, widget-level and feature-level access control for granular permissions, comprehensive audit trails logging all development activities and data access, and centralized application visibility through dashboards providing IT oversight of all platform applications [2], [20].

Security capabilities must include input validation and sanitization, multi-factor authentication, encryption for sensitive data, and enforcement of security policies across all citizen-developed applications.

Business-Managed IT Critical Success Factors

Klotz et al. [19] identify 19 Critical Success Factors (CSFs) for Business-Managed IT governance. The CSFs most relevant to LCDP-enabled governance are presented in Table 3.

Table 3 – Relevant BMIT Critical Success Factors for LCDP Governance

CSF	Description	LCDP Implication
CSF1: BU/IT mutual approach	Both business unit and IT organization establish a culture supporting business-managed IT	Platform should facilitate BU-IT collaboration, not circumvention
CSF5: Cross-functional team	BMIT team combines business know-how with IT affinity	Citizen developer programs should recruit domain experts with technical inclination
CSF9: Transparency	BU and IT maintain visibility into BMIT instances	Platform must provide centralized dashboards and reporting
CSF11: IT consultancy	IT organization provides advisory support for BMIT	Platform should integrate touchpoints for IT review/approval
CSF16: IT infrastructure integration	BMIT instances integrate with existing enterprise architecture	API capabilities and connector libraries required

A critical insight from Klotz et al. [19] is that effective BMIT governance requires bilateral responsibility; neither pure IT control nor complete business unit autonomy, but shared governance where "it takes two to tango." LCDPs should facilitate this shared responsibility model rather than enabling unilateral citizen development.

LCDP Governance Guidelines

Viljoen et al. [3] propose a set of six governance guidelines aimed at balancing flexibility and standardization in the adoption of Low-Code Development Platforms. The first guideline emphasizes the importance of proactive collaboration with platform providers. Organizations are encouraged to engage directly with LCDP vendors in order to influence platform roadmaps, particularly in situations where organizational customization requirements extend beyond the capabilities offered by standard platform features.

The second guideline recommends that organizations adapt their internal processes to align with the features provided by the platform. Rather than pursuing extensive platform customization, which can increase complexity and maintenance overhead, organizations should consider reshaping their processes to take advantage of out-of-the-box platform capabilities.

A third guideline focuses on encouraging experimentation within controlled environments. Enterprise-grade LCDPs typically support environment separation, such as sandbox, development, testing, and production environments, which enables organizations to foster innovation while maintaining governance and risk control through structured experimentation.

The fourth guideline advocates promoting the use of out-of-the-box features, particularly among non-technical developers. Encouraging reliance on default platform features and template libraries simplifies governance and reduces the likelihood of inconsistencies or unmanaged deviations from established standards.

The fifth guideline highlights the value of involving multiple partners in the governance of low-code initiatives. Establishing cooperative models that include domain experts, IT specialists, and external consultants allows organizations to balance diverse perspectives and competencies, supporting more effective decision-making and oversight.

Finally, the sixth guideline underscores the need to harmonize processes and workflows across organizational units. Aligning workflows helps ensure consistency, reduces fragmentation, and supports coherent governance in environments where multiple teams and stakeholders contribute to application development.

4.2.3. Collaborative Development Support

Effective Shadow IT mitigation requires platforms that facilitate collaboration between business units and IT departments. Version control mechanisms enable multiple developers to work on applications simultaneously while maintaining change history. Workflow management features coordinate development activities across teams, while feedback and review capabilities support iterative refinement with stakeholder input [20].

Klotz et al. [19] emphasize that communication and transparency (CSF9, CSF10) are among the highest-ranked success factors for business-managed IT. LCDPs should therefore provide built-in communication tools, notification systems, and collaborative workspaces that bridge the traditional gap between business and IT stakeholders.

4.2.4. DevSecOps and Compliance Integration

Modern LCDPs must integrate DevSecOps principles, embedding security throughout the development lifecycle rather than treating it as a post-development afterthought [2]. This makes security a shared responsibility across development, operations, and security teams, addressing the unique challenges posed by citizen developers who may lack technical security expertise.

Sedrakyan et al. [2] propose the RAAFT framework (Risks, Actors, Actions, Feedback, Tools) for integrating DevSecOps into low-code development (Table 4).

Table 4 – RAAFT Framework Summary (adapted from Sedrakyan et al. [2])

Development Phase	Key Actors	Primary Risks	Mitigating Actions
Design	Business Analysts, Security Experts	Inadequate security requirements	Early threat modelling, secure architecture review
Development	Citizen Developers, Professional Developers	Vulnerable code, misconfiguration	Secure component usage, input validation
Testing	QA Teams, Security Experts	Undetected vulnerabilities	Comprehensive security testing (SAST/DAST)
Deployment	ITOps Teams	Insecure deployment configurations	Secure CI/CD pipelines
Maintenance	All actors	New vulnerabilities, patch delays	Continuous monitoring, timely patching

LCDPs must provide shift-left security (integration of security practices early in development), automated compliance checking (compliance-as-code mechanisms), process-oriented guidance (contextual recommendations for citizen developers), and continuous security assessment (real-time monitoring integrated into CI/CD pipelines).

4.2.5. Role Separation and Domain Specialization

Effective Shadow IT mitigation requires clear delineation of responsibilities across different stakeholder types. The literature proposes a four-role architecture that extends beyond the traditional binary of IT professionals versus business users. However, a critical distinction must be made between the theoretical role architecture proposed in research and the actual capabilities of current LCDP offerings.

Theoretical Four-Role Architecture

Mottu and Sunyé [20] argue that the notion of the citizen developer should be disaggregated into a set of distinct roles in order to better reflect the diversity of responsibilities involved in low-code

development. At the foundational level, platform engineers, typically software engineers, are responsible for developing and maintaining the Low-Code Development Platform itself. Building on this foundation, domain engineers specialize the generic platform for specific application domains, such as a pedagogical engineer creating teaching-specific components tailored to educational contexts. Domain practitioners then use this specialized version of the platform to develop applications for end users, for example, a teacher creating quiz applications using preconfigured pedagogical components. Finally, end users interact with and consume the applications produced by domain practitioners.

This role separation enables domain practitioners to concentrate on applying their domain expertise without the need to master generic software development concepts, while domain engineers focus on platform specialization and evolution, operating on iteration cycles that differ from those of application development.

Current Platform Limitations: A Critical Gap

Despite the theoretical benefits of role separation, current LCDPs do not natively support this architecture. Mottu & Sunyé [20] state explicitly:

> "To the best of our knowledge, the specialization of an LCDP is not a feature of existing platforms. They propose to model the domain and define its rules using prebuilt components, but they do not explicitly propose to distinguish between the two roles of citizen developers. In particular, they do not allow a citizen developer to consider only elements modelled for her domain when building an application."

This finding was confirmed through evaluation of two industry-leading platforms (Mendix and OutSystems). Organizations seeking to implement role-segregated governance must therefore establish complementary organizational mechanisms – such as template libraries, access control layering, and governance processes – to compensate for platform limitations.

4.2.6. Data Management and Interoperability

Robust data management capabilities are essential to prevent the data fragmentation that characterizes Shadow IT environments. LCDPs must provide unified data schemas that maintain consistency across citizen-developed applications, dynamic content databases that support flexible data structures, and plugin support for extending data management capabilities [27].

Interoperability features ensure that data flows seamlessly between LCDP applications and enterprise systems, preventing the data silos that emerge when employees adopt unauthorized tools.

Standard data formats, export capabilities, and integration with enterprise data governance frameworks support organizational data management objectives.

4.2.7. Platform Characteristics and Market Position

Enterprise LCDP selection should consider platform characteristics that influence long-term viability and support. Cloud-based architecture enables scalability, reduces infrastructure management burden, and facilitates continuous platform updates. Vendor stability, market position, and ecosystem maturity affect the availability of pre-built components, integration connectors, and community support resources [27], [28].

Built-in governance support varies significantly across platforms. Enterprise-grade platforms such as ServiceNow, Mendix, and OutSystems typically offer robust environment separation and version control, while lightweight platforms such as Airtable, Power Automate, or Zapier often lack structured governance features [3]. Platform selection should align with organizational governance requirements.

4.2.8. Platform Specialization and Customization

Viljoen et al. [3] identify a fundamental tension in the design philosophy of Low-Code Development Platforms that organizations must carefully navigate. On one end of the spectrum are standardization-favouring platforms, such as ServiceNow, which tend to deliver maximum value when organizations adopt standardized workflows and rely heavily on prebuilt modules. In these platforms, extensive customization is possible but typically requires proactive engagement with the vendor, as customization pathways are more constrained by the platform's standardized design.

At the other end of the spectrum are customization-favouring platforms, including Mendix, OutSystems, and Microsoft Power Apps. These platforms offer organizations greater flexibility to tailor solutions to their specific requirements, but this increased freedom often comes at the cost of higher technical complexity and a greater need for specialized expertise.

Complementing this distinction, Frings and Schuh [29] differentiate between two forms of customization. Vendor-side customization limits users to predefined configuration options offered by the platform, whereas user-side customization allows developers to modify or extend artifacts using programming languages. Taken together, these perspectives underscore the importance of assessing the alignment between a platform's underlying design philosophy and an organization's process requirements before committing to a specific LCDP.

4.2.9. Limitations and Constraints

While LCDPs offer significant potential for Shadow IT mitigation, the literature identifies systematic limitations that constrain their effectiveness. Martinez and Pfister [30] synthesize these limitations into a validated framework (L1-L6) based on empirical case studies and practitioner surveys (Table 5).

Table 5 – LCDP Limitations Framework (L1-L5 from Martinez and Pfister [30]; L6 added from security literature)

Code	Limitation	Description	Empirical Evidence
L1	Lack of Customization	Low-code is less powerful than traditional programming; users constrained to platform options	72.7% confirmation [30]
L2	Scalability Constraints	Solutions suitable for departmental use may not scale to enterprise-wide deployment	Context-dependent; varies by perspective
L3	Fragmentation	Different vendors use incompatible paradigms, preventing cross-platform integration	Limited interoperability confirmed
L4	Vendor Lock-in	Switching platforms becomes prohibitively expensive once applications deployed	72.7% agreement [30]
L5	High Licensing Cost	Costs escalate when scaling to larger user bases	Concern varies by organizational level
L6	Security Risks	Citizen developers may deploy applications without adequate security consideration	Confirmed [2]

Distinguishing Empirical vs. Theoretical Limitations

Empirically validated limitations, confirmed through case studies or survey-based evidence, include L1 (Customization) and L4 (Vendor lock-in). These limitations are strongly supported by empirical data, with 72.7% of practitioners reporting these constraints [23]. In addition, L6 (Security) is substantiated by multiple independent studies, notably those conducted by Orr et al. [24] and Sedrakyan et al. [2].

In contrast, some limitations are context-dependent, with their validity varying according to organizational perspective. L2 (Scalability) and L5 (Licensing cost) exhibit considerable dispersion in survey responses, reflecting differences between project-level evaluations and broader organizational-level assessments [23].

Finally, certain limitations have been identified primarily at a theoretical level and have been proposed in the literature without yet receiving broad empirical validation. This includes the four-role architecture gap identified by Mottu and Sunyé [20], which is based on the analysis of only two platforms. Similarly, concerns related to the learning curve were not supported by the findings of Guthardt et al. [25]; however, the small sample size used in that study (n = 10) limits the generalizability of its conclusions.

This distinction is important for practitioners. Empirically validated limitations call for immediate mitigation planning, whereas context-dependent and theoretically identified limitations require organization-specific assessment.

4.3. Mapping Challenges to Platform Capabilities

To establish a coherent analytical framework, this section explicitly maps the Shadow IT challenges identified in Section 4.1 to the LCDP features catalogued in Section 4.2. This mapping (Table 6) demonstrates the logical coherence of the feature framework and provides the conceptual foundation for the evaluation framework developed in Chapter 5.

Table 6 – Challenge-to-Feature Mapping Matrix

Challenge Domain	Specific Challenge	Mitigating Feature(s)	Mechanism
IT Responsiveness Gap	Slow solution delivery	Rapid Development (4.2.1)	Vendor-claimed 5-10x faster development; empirically confirmed productivity gains
	Resource constraints	Complexity Reduction and Accessibility (4.2.2)	Citizen developers reduce dependency on scarce IT staff
	Backlog accumulation	Collaborative Development (4.2.3)	Parallel development and version control increase throughput
Application Complexity	Rigid interfaces, poor domain fit	Platform Specialization (4.2.8)	Domain-tailored interfaces match user mental models and workflows
	Integration barriers	Enterprise Integration (4.2.2)	Pre-built connectors and API capabilities enable seamless data exchange
	Scalability limitations	Platform Characteristics (4.2.7)	Cloud architecture, vendor ecosystem maturity support enterprise scale
Governance Risks	Unauthorized access, security gaps	Governance and Security Controls (4.2.2)	RBAC, MFA, audit trails, centralized dashboards
	Compliance violations	DevSecOps Integration (4.2.4)	Automated compliance checking, shift-left security, RAAFT framework
	Role ambiguity, accountability gaps	Role Separation (4.2.5)	Four-role architecture clarifies responsibilities (requires organizational implementation)
	Data fragmentation, silos	Data Management (4.2.6)	Unified schemas, interoperability standards, enterprise data governance integration

Many of the identified challenges depend on the combined operation of multiple platform features rather than on isolated capabilities. For example, closing the IT responsiveness gap requires not only rapid development functionality but also mechanisms for reducing complexity and supporting collaborative development. Consequently, organizations are advised to evaluate feature portfolios in a holistic manner instead of assessing individual features in isolation.

Governance should be understood as a cross-cutting concern that spans all challenge domains. Security and governance-related features (4.2.2, 4.2.4) are relevant across all three domains: although they primarily mitigate governance risks, they also allow IT departments to formally endorse and control citizen development initiatives that might otherwise emerge as Shadow IT.

It is also important to recognize that certain combinations of features involve inherent trade-offs. Features designed to enhance accessibility and empower citizen developers can conflict with security

requirements if they are not correctly configured. This tension highlights the need for integrated DevSecOps practices that embed security safeguards directly into the development lifecycle.

Finally, the importance of specific features varies according to an organization's level of LCDP maturity, as noted by Viljoen et al. [3]. Organizations in early stages of adoption gain the greatest benefit from leveraging out-of-the-box functionality and standardization, whereas more mature organizations tend to place greater emphasis on customization capabilities and mechanisms for collaboration with platform vendors.

CHAPTER 5

Evaluation Framework Development

Chapter 5 operationalizes the challenges identified in Chapter 4 by introducing an evaluation framework that was constructed with structured support from artificial intelligence. In practical terms, the AI system was used as a controlled text-processing and reasoning assistant over the full corpus of selected literature and the consolidated list of Shadow IT–related challenges from Chapter 4. By ingesting that chapter as contextual input, the AI was explicitly calibrated to interpret Low-Code Development Platforms not as generic software engineering tools, but as targeted interventions for the IT responsiveness gap, security and compliance weaknesses, governance fragmentation, and role-related frictions previously identified. This ensured that each candidate criterion was tested against the question “does this capability mitigate a documented Shadow IT driver or enterprise challenge?” rather than merely reflecting feature availability in the market.

Technically, the AI was then prompted with focused queries that combined the Chapter 4 problem space (for example, LowDevSecOps requirements, flexibility–standardization tensions, four-role architectures, data fragmentation, and workaround patterns) with the wider literature on LCDPs, citizen development and Business-managed IT. Across multiple iterations, the system retrieved, compared and synthesized passages that associated concrete platform characteristics with those specific challenges, which the researcher distilled into the ten thematic clusters that structure this chapter (Parts A–J). Within each cluster, the AI further supported the refinement of granular assessment items by checking definitional consistency, highlighting overlaps and contradictions between sources, and suggesting where higher standards (e.g., shift-left security, automated compliance checking, role-segregated governance) were warranted in light of Chapter 4. The final framework and Likert-scale questions are therefore the result of an AI-assisted, but researcher-controlled, abstraction process that links every evaluation dimension back to the theoretically grounded Shadow IT challenges established in the preceding chapter.

5.1. Thematic Clusters

The evaluation framework is organized into ten thematic clusters, each corresponding to a distinct dimension of enterprise LCDP capability with direct relevance to Shadow IT mitigation. The clusters are labelled A through J (see Appendix A for the complete list). Appendix A explains the theoretical motivation for each cluster, the connection between the cluster's and the documented challenges of Chapter 4, and the types of platform capabilities being assessed.

5.2. Evaluation Items by Cluster

The complete evaluation framework, comprising 62 items organized across ten thematic clusters, is presented in Appendix B. Each item includes a descriptive name, a precise operational description specifying what platform capability is being assessed, and a research foundation tracing the item's theoretical basis to the source literature from which it was derived. The relationship between items and the Shadow IT challenges documented in Chapter 4 is maintained throughout: every item in Clusters A through J can be mapped back to one or more of the documented challenge categories, ensuring that the framework assesses LCDP capability as a function of its Shadow IT mitigation potential rather than as a generic product feature inventory.

CHAPTER 6

Framework Application and Assessment

This chapter describes the application of the evaluation framework developed in Chapter 5 to leading enterprise Low-Code Development Platforms. The methodology employs multiple large language models to systematically assess platform capabilities against the 62-item evaluation framework, with structured validation protocols ensuring result reliability and academic defensibility.

6.1. Selecting Low-Code Platforms

Platform selection was guided by triangulation across multiple independent analyst rankings to ensure consensus-based identification of market leaders suitable for enterprise Shadow IT mitigation. This multi-source approach reduces bias inherent in any single analyst methodology and strengthens the validity of platform selection for academic research.

The Gartner Magic Quadrant for Enterprise Low-Code Application Platforms positions vendors based on completeness of vision and ability to execute. The July 2025 report identifies six Leaders: Mendix, positioned furthest on the Completeness of Vision axis and maintaining Leader status for nine consecutive years; OutSystems, positioned highest on the Ability to Execute axis, also a nine-year consecutive Leader; Microsoft Power Apps, a seven-year consecutive Leader; ServiceNow, a six-year consecutive Leader; Appian, a three-year consecutive Leader; and Salesforce, newly positioned in the Leaders quadrant for the first time [31].

The Forrester Wave for Low-Code Platforms for Professional Developers evaluates fourteen vendors across multiple criteria weighted for enterprise requirements. The Q2 2025 report positions OutSystems as a Leader recognized for superior user experience development, data modelling capabilities, and scalable architecture. Microsoft Power Platform ranks highest for strength of strategy and current offering. ServiceNow achieves Leader status with recognition for clear vision and AppGen platform strategy. Mendix is recognized as a Leader for data modelling, agile management, and version control capabilities. Appian is positioned as a Strong Performer with superior automation capabilities [32].

The Everest Group PEAK Matrix evaluates service providers and technology vendors based on market impact and vision and capability dimensions. The October 2025 assessment of twenty-four platforms identifies six Leaders: OutSystems, Mendix, Microsoft Power Apps, Appian, Salesforce, and ServiceNow [33].

The IDC MarketScape represents the inaugural assessment of the worldwide low-code and no-code developer technologies category. The July 2025 report positions OutSystems as a Leader

recognized for comprehensive AI-powered development capabilities, alongside Salesforce for its enterprise agent development ecosystem, Mendix, and Appian. ServiceNow is positioned as a Major Player in this assessment [34].

The G2 Grid aggregates verified user reviews from enterprise practitioners to assess customer satisfaction and market presence. The Winter 2026 report, based on evaluation of over fifty platforms, positions OutSystems as the top-ranked platform with a satisfaction score of 99, the highest in the category. Additional Leaders include Salesforce Platform, ServiceNow App Engine, Microsoft Power Apps, Quickbase, and UiPath Platform review [35].

The Nucleus Research Value Matrix evaluates platforms across usability and functionality dimensions. The November 2025 assessment positions Appian, OutSystems, and several other platforms as Leaders. Notably, Nucleus Research categorizes Mendix, Microsoft Power Apps, Salesforce, and ServiceNow as Experts, a classification indicating high functionality with lower usability scores. This methodological difference reflects the distinct evaluation criteria employed by Nucleus Research compared to other analyst firms [36].

PeerSpot aggregates peer reviews from verified enterprise IT professionals with continuous updates. The February 2025 rankings identify Microsoft Power Apps as the top solution with 18.3% mindshare, followed by OutSystems with the highest rating of 8.9 out of 10, Oracle Application Express, Appian with 12.4% mindshare, and Salesforce Platform [37]. Table 7 summarizes platform positioning across all seven analyst sources.

Table 7 – Cross-Analyst Platform Positioning (2025)

Platform	**Gartner**	**Forrester**	**Everest**	**IDC**	**G2**	**Nucleus**	**PeerSpot**	**Consensus**
Mendix	Leader	Leader	Leader	Leader	Leader	Expert	Top 10	**Selected**
OutSystems	Leader	Leader	Leader	Leader	#1	Leader	#2	**Selected**
Power Apps	Leader	Leader	Leader	Evaluated	Leader	Expert	#1	**Selected**
Appian	Leader	Strong Perf.	Leader	Leader	Leader	Leader	#4	**Selected**
ServiceNow	Leader	Leader	Leader	Major Player	Leader	Expert	Top 5	Not Selected
Salesforce	Leader	Evaluated	Leader	Leader	Leader	Expert	#5	Not Selected

Four platforms were selected for assessment based on this cross-analyst consensus. OutSystems and Mendix demonstrate the strongest consensus, achieving Leader status across all seven analyst reports, representing the most consistently recognized enterprise LCDP solutions. Microsoft Power Apps demonstrates strong enterprise adoption reflected in its 18.3% mindshare on PeerSpot and consistent Leader positioning across major analyst evaluations. Appian is recognized universally for process automation capabilities directly relevant to Shadow IT governance requirements.

ServiceNow and Salesforce, while consistently positioned as Leaders across analyst reports, were not included in the final selection due to fundamental differences in platform architecture and intended use cases. ServiceNow is architected primarily as an IT Service Management and enterprise workflow platform that incorporates low-code capabilities as an extension of its core service delivery focus. Similarly, Salesforce originated as a Customer Relationship Management platform that has expanded to include low-code development through its Salesforce Platform offering. Both platforms deliver significant value when organizations seek to extend or customize their existing ServiceNow or Salesforce investments, providing out-of-the-box functionality for their respective domains that can be expanded through low-code development.

In contrast, Mendix, OutSystems, Microsoft Power Apps, and Appian represent general-purpose low-code development platforms designed as blank slates for building custom enterprise applications across any business domain. These platforms do not presuppose a specific functional starting point such as IT service management or customer relationship management, instead providing foundational development environments that organizations configure entirely according to their unique requirements. This architectural distinction is significant for Shadow IT mitigation research because general-purpose platforms more directly address the scenario where business units seek to develop custom solutions outside IT governance, whereas domain-specific platforms primarily serve organizations already committed to their respective ecosystems.

The four selected platforms therefore represent sufficient diversity in architectural approach and market positioning while ensuring comprehensive coverage of enterprise LCDP capabilities relevant to Shadow IT mitigation in contexts where organizations require flexible, domain-agnostic development capabilities.

The selection deliberately excludes no-code platforms such as Airtable and Zapier which, while valuable for specific use cases, lack the governance and scalability features required for enterprise-wide Shadow IT mitigation strategies [3]. All selected platforms demonstrate mature enterprise governance capabilities as evidenced by their multi-year presence in analyst Leader quadrants and their recognition for security, compliance, and centralized management features essential for addressing Shadow IT challenges in large organizations.

6.2. Selecting LLM Platforms

The assessment methodology leverages multiple large language models to reduce single-model bias and increase result reliability. LLM selection followed an iterative trial-and-error process conducted during the last quarter of 2025.

Four LLM platforms were selected for the final assessment, all using paid first-tier subscriptions to ensure access to maximum capabilities including internet search functionality and extended context windows. ChatGPT was selected for its strong reasoning capabilities and reliable internet search integration. Claude was included for its nuanced analysis capabilities and consistent output formatting. Gemini provided strong multi-source synthesis with Google ecosystem integration. Grok offered real-time information access and provided an alternative analytical perspective.

Several platforms were evaluated but excluded due to inadequate output quality for the complex assessment requirements. DeepSeek was excluded because it cannot search the internet and its knowledge base was not sufficiently current for accurate platform assessment. Le Chat from Mistral proved unable to process the complex multi-part prompt, exhibiting significant processing limitations. Qwen demonstrated excessive hallucinations, generating features and capabilities not present in the actual platforms being assessed. Free-tier versions of all platforms were initially tested but proved inadequate for processing the complex, multi-part evaluation prompt required by the 62-item evaluation framework.

6.3. Defining the Universal Prompt and Data Input

The assessment employed a standardized prompt architecture applied consistently across all LLM-platform combinations. The prompt structure was designed to ensure academically defensible, evidence-based evaluations suitable for inclusion in a master's thesis focused on Shadow IT mitigation in large enterprises. The complete prompt text is provided in Appendix B. Table 8 summarizes the prompt architecture and its components.

Table 8 – Assessment Prompt Architecture

Component	Purpose	Key Requirements
Context Definition	Establishes evaluation scope	Enterprise focus (≥1,000 employees); Shadow IT mitigation orientation
Source Authorization	Defines acceptable evidence	Internet search enabled; 2023–2025 publications preferred; vendor and non-vendor sources required
Source Prohibition	Prevents circular referencing	Context files prohibited as citation sources; all evidence must be externally verifiable
Output Specification	Standardizes response format	Seven required fields per item: Rating, Justification, Evidence_Type, Confidence, Source_Texts, Full_APA_Citations
Rating Scale	Defines scoring criteria	1–5 Likert scale with enterprise-specific anchors; inverted logic for Section J (limitations)
Quality Controls	Ensures academic rigor	Unique justifications per item; dual-sourcing requirement; anti-hallucination rules

To ensure methodological transparency and enable future replication, the complete prompt texts are provided in Appendix B. The assessment prompt architecture consisted of four interconnected components:

The primary assessment prompt (A-Prompt.txt, 256 lines) established the evaluation context, defined the rating scale with enterprise-specific anchors, specified the seven required output fields per assessment item, and imposed critical anti-hallucination rules. Key constraints included the prohibition against using uploaded context files as citation sources, the requirement for at least two distinct sources per item with at least one non-vendor source, and the inverted scoring logic for Section J (limitations).

The contextual background document (B-Info.txt, 82 lines) provided the theoretical framework from Chapter 3, ensuring LLMs interpreted platform capabilities through the lens of Shadow IT mitigation rather than generic software evaluation. This document synthesized the literature on IT responsiveness gaps, application complexity challenges, and governance requirements.

The execution instructions (D-Call.txt, 25 lines) specified the batch processing protocol, data source requirements distinguishing between definitional sources (the uploaded files) and evidence sources (internet search and training data), and output format specifications for Excel compatibility.

The validation prompt (E-Revision.txt, 191 lines) defined the systematic quality auditing protocol for verifying prompt compliance, evidence integrity, justification quality, and cross-model consistency.

All assessments were conducted between October and December 2025. All models were accessed through paid first-tier subscriptions with internet search functionality enabled. Figure 2 illustrates the assessment execution workflow from prompt input through validated output.

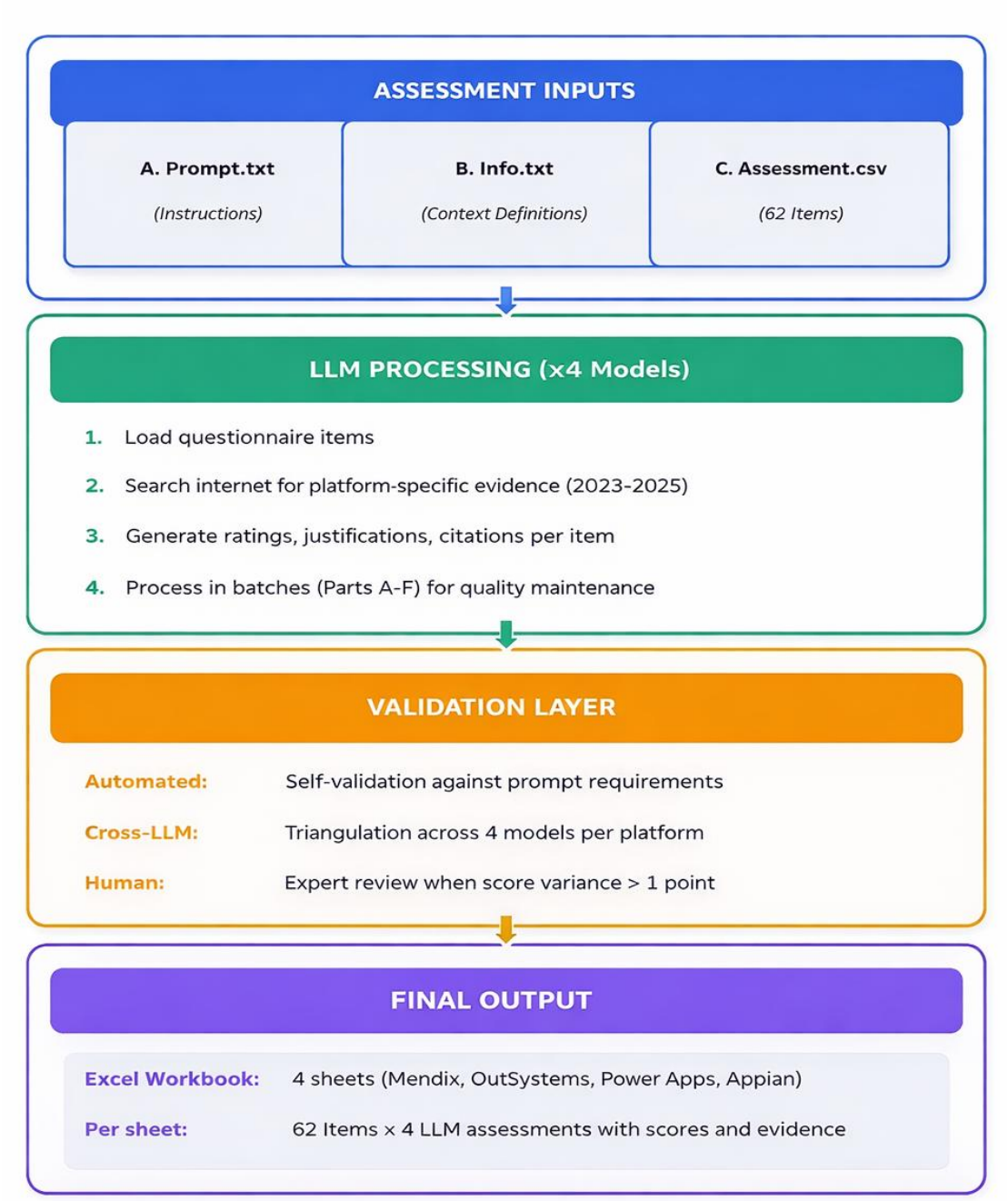


Figure 2 – Assessment Execution Workflow

The primary assessment prompt established the evaluation context by identifying the target platform and framing the assessment for enterprise organizations with more than 1,000 employees and strong Shadow IT mitigation requirements. The prompt explicitly authorized LLMs to use internal

internet search tools to locate relevant, credible, and recent evidence published between 2023 and 2025 for each assessment item.

Evidence sourcing requirements specified that vendor documentation, platform feature guides, security documentation, analyst reports from firms such as Gartner and Forrester, practitioner blogs with technical depth, peer-reviewed academic papers, case studies, and enterprise-focused user reviews constituted acceptable sources. The prompt imposed a critical prohibition against using the uploaded context files as evidence sources, ensuring that all citations derived exclusively from externally verifiable sources.

For each of the 62 assessment items, the prompt required generation of seven output fields. The Rating field required a score from 1 to 5 based on capability strength relative to the item requirements. The Justification field required a unique, detailed, item-specific academic-grade explanation including a clear evaluative claim, evidence-based reasoning using synthesized rather than copied evidence, and explicit connection to how the capability affects an enterprise's ability to reduce reliance on unmanaged or unsanctioned solutions. The Evidence_Type field required classification among categories including vendor documentation, analyst report, academic research, practitioner review, case study, or acknowledgment of insufficient evidence. The Confidence field required assessment as High, Medium, or Low based on robustness and quantity of available evidence, degree of consensus among independent sources, and potential limitations or ambiguities. The Source_Texts field required at least two direct, item-relevant evidence excerpts with at least one non-vendor source per item. The Full_APA_Citations field required at least two distinct APA 7 citations with web links for each item.

The rating scale was defined with specific criteria for enterprise LCDP assessment. A rating of 5 (Strongly Agree) indicated strong, well-documented capability directly aligned to the item with rich, mature features clearly oriented toward enterprise-scale use and strong alignment with Shadow IT mitigation. A rating of 4 (Agree) indicated well-supported capability that is reasonably mature and suitable for most enterprise scenarios with non-critical limitations. A rating of 3 (Neutral) indicated capability that is present and usable but has noticeable gaps, limitations, or complexity adequate for some but not all enterprise scenarios. A rating of 2 (Disagree) indicated capability that exists but is limited, immature, or heavily constrained requiring significant workarounds, custom code, or external tools. A rating of 1 (Strongly Disagree) indicated capability that is absent, extremely weak, or clearly unsuitable for enterprise or mission-critical use with major gaps versus large enterprise requirements.

For items in Section J addressing platform limitations and constraints, the prompt specified inverted scoring logic where the rating scale represents the absence of risk rather than presence of capability. Under this inversion, a rating of 1 indicates worst outcome with high risk or high lock-in, while a rating of 5 indicates best outcome with low risk or minimal constraints.

The execution prompt instructed LLMs to load the evaluation framework, read data dynamically from each row, use the context files only to understand definitions and theoretical frameworks while deriving all evidence from internet search and internal training data, and generate output in batches for better process management.

6.4. Challenges

The LLM-assisted assessment process encountered several methodological challenges requiring iterative refinement throughout the evaluation period. Processing limitations emerged as a significant constraint. LLMs could not reliably process all 62 evaluation items in a single pass while maintaining output quality and adherence to prompt requirements. The assessment was therefore conducted in batches organized by thematic cluster, with each of the ten parts (A through J) processed separately to ensure comprehensive evaluation without overwhelming context windows or degrading response quality.

Output quality issues required multiple correction cycles. Initial outputs exhibited malformed CSV structures that prevented direct import into analysis tools. Source citations frequently required verification against actual documentation. Most significantly, iterative corrections introduced new errors, creating a recursive quality assurance challenge where fixing one set of problems occasionally generated different problems requiring additional passes.

Hallucination management represented an ongoing concern across all LLM platforms. All models occasionally generated capabilities or features not actually present in the assessed platforms. The prompt included explicit anti-hallucination rules instructing models not to invent specific article titles, DOIs, or URLs when uncertain, and to use generic but honest references with low confidence ratings when specific evidence could not be located. Despite these safeguards, cross-LLM comparison remained essential as an initial hallucination detection mechanism.

Consistency variance between assessment runs presented reliability challenges. Multiple runs of the same assessment occasionally produced different scores, indicating sensitivity to prompt interpretation and stochastic elements in model responses. This variance was addressed through multi-LLM triangulation, where disagreements between models triggered human review rather than relying on any single model's output.

6.5. Output Validation

A multi-layered validation approach ensured assessment accuracy and academic defensibility. The validation methodology addressed evidence integrity, cross-model consistency, and methodological compliance.

The automated self-validation component instructed each LLM to validate its output against the original prompt requirements before finalizing responses. This first-pass validation caught structural errors and obvious omissions before human review.

Source verification requirements mandated dual citation for every scored item, with one source from vendor documentation and one from independent sources such as analyst reports, academic literature, or third-party reviews. This dual-sourcing requirement reduced the risk of vendor marketing language influencing assessments while ensuring evaluations remained grounded in actual platform capabilities.

Cross-LLM triangulation formed the core reliability mechanism. Each of the four platforms was assessed by all four LLMs, generating sixteen independent assessment instances. Scores were compared across LLMs to identify outliers and potential errors. When score disagreement between LLMs exceeded one point on the five-point Likert scale, human expert review was triggered.

A comprehensive validation prompt was developed for systematic quality auditing. This validation protocol evaluated compliance with explicit prompt constraints. Key requirements included correct platform name usage, presence of all required output columns, proper Likert scale application (including inverted logic for Section J), academic tone, and absence of prohibited content. The validation assessed evidence integrity through critical evaluation of source texts and citations for plausibility, verifiability, item-specificity, non-vendor sourcing, and publication date consistency. Justification quality assessment examined whether each justification made a clear evaluative claim, demonstrated logical connection to cited evidence, addressed Shadow IT mitigation impact, and avoided repetitive or templated language. Cross-model consistency analysis compared ratings assigned by different LLMs, identified variance patterns, and detected model-specific biases toward systematic inflation or deflation.

6.6. The Final Assessment Output

Assessment results are compiled in a structured Excel workbook containing four sheets (Appendix A) corresponding to each assessed LCDP. Each sheet presents all 62 evaluation framework items from A1 through J6 with scores and supporting evidence from each of the four LLMs. The output structure includes the item identifier, rating on the 1-5 scale, detailed justification text, source text excerpts, and full APA citations.

The scoring interpretation follows the prompt-defined criteria where ratings of 4 and 5 indicate strong platform alignment with the assessed capability suitable for enterprise Shadow IT mitigation, ratings of 3 indicate adequate but limited capability requiring careful consideration of organizational context, and ratings of 1 and 2 indicate significant gaps that may necessitate complementary solutions

or alternative platforms. For Section J items addressing limitations, this interpretation inverts such that higher scores indicate lower risk exposure.

6.7. Human Validation

Human validation was conducted by four domain experts, with one assigned to each assessed platform. Each expert possessed a minimum of four years of professional implementation experience with their assigned platform in enterprise contexts. Experts were recruited through professional networks and were selected based on demonstrated hands-on experience with platform deployment, governance configuration, and enterprise solution delivery. The scope of the human validation was limited to those assessment items for which inter-LLM score differences exceeded one point on the Likert scale, representing cases where model disagreement surpassed acceptable interpretive variation. Table 9 provides a summary of the expert validator profiles.

Table 9 – Expert validator profiles

Platform	Platform Experience	Role	Items Validated
Mendix	5 years	Technical Lead	10
OutSystems	6 years	Solution Architect	8
Appian	5 years	Architect	8
Power Apps	4 years	Certified Trainer	8

The expert assigned to the Appian platform – João – has worked in the IT sector for fourteen years, with approximately thirteen years of experience in business process management. He has led multiple teams responsible for both application maintenance and greenfield development initiatives. Over the past five years, his professional focus has been on Appian as a low-code development platform, and he currently serves as an architect in an IT consulting firm.

The OutSystems expert – Alex – brings more than thirty years of experience in the IT field and approximately six years of hands-on experience with OutSystems. His professional background includes managing global teams across telecommunications, banking, and insurance projects. At present, he works as a Solution Architect within a large consulting organization employing approximately 90,000 professionals.

The Mendix expert – Tim – is an IT professional with over twenty years of experience in software development, systems integration, and enterprise-wide implementations. He has been working with

Mendix since 2021 and currently holds the role of Technical Lead, where he guides teams in delivering high-quality Mendix solutions in complex, international project environments.

Finally, the expert assigned to Microsoft Power Apps – Nelson – has several decades of experience in the IT domain and has been working with the Power Platform since 2022. His experience combines training delivery with hands-on project implementation, and he currently works as a trainer with a focus on low-code development and artificial intelligence.

Approximately eight to ten items per platform required human validation intervention, representing roughly 13-18% of assessed items. Domain experts reviewed the conflicting assessments against available vendor documentation and independent sources to determine appropriate scores. When human judgment and documentation review warranted correction, expert scores overrode LLM average values.

The human validation process did not reveal systematic LLM biases toward consistent over-scoring or under-scoring across platforms. Disagreements appeared distributed across assessment dimensions rather than concentrated in specific capability areas, suggesting that variance reflected genuine interpretive complexity rather than model-specific calibration errors. Detailed analysis of validation findings and their implications for result interpretation is presented in Chapter 7.

CHAPTER 7

Assessment Results

This chapter presents the consolidated assessment results from the systematic evaluation of four enterprise Low-Code Development Platforms (Mendix, OutSystems, Microsoft Power Apps, and Appian) against the 62-item evaluation framework developed in Chapter 5. The assessment employed a multi-LLM triangulation methodology using four large language models (Gemini, ChatGPT, Claude, and Grok), with human expert validation applied to items exhibiting significant inter-model disagreement.

7.1. Summary Results by Platform

Table 10 presents the assessment summary, from Appendix A, organized by thematic cluster, enabling comparison of platform capabilities across the ten evaluation dimensions established in Chapter 5.

Table 10 – Assessment Summary by Thematic Cluster

Part	Thematic Cluster	Mendix	OutSystems	Appian	Power Apps
A	Rapid Development and Agility	4.75	4.88	4.67	4.79
B	Complexity Reduction and Accessibility	4.46	4.58	4.33	4.75
C	Enterprise Integration and Scalability	4.79	4.75	4.79	4.71
D	Governance and Security Control Mechanisms	4.64	4.81	4.78	4.61
E	Collaborative Development Support	4.20	4.60	4.30	4.25
F	DevSecOps and Compliance Integration	3.54	4.21	4.42	3.46
G	Role Separation and Domain Specialization	4.46	4.79	4.29	4.58
H	Data Management and Interoperability	4.75	4.92	4.75	4.71
I	Platform Maturity and Ecosystem	4.50	4.54	4.71	4.58
J	Limitations and Constraints	3.25	3.54	3.21	3.25
	Overall Platform Average	**4.35**	**4.57**	**4.44**	**4.38**

OutSystems achieves the highest overall average rating, followed by Appian, Microsoft Power Apps, and Mendix. All four platforms demonstrate strong performance in core capability areas including Rapid Development and Agility (Part A), Enterprise Integration and Scalability (Part C), and Data Management and Interoperability (Part H), with cluster averages consistently exceeding 4.0 on the five-point Likert scale. The Limitations and Constraints cluster (Part J) presents the lowest scores across all platforms, reflecting the inherent trade-offs documented in the low-code development literature.

Beyond these aggregate results, each platform exhibits a distinct capability profile that reflects its underlying architectural philosophy and intended use cases. These profiles illustrate how different design priorities translate into strengths and weaknesses across capability clusters.

OutSystems achieves the highest scores in Data Management and Interoperability, Rapid Development and Agility, and Governance and Security Control Mechanisms. This profile aligns with its documented positioning as an enterprise-grade platform emphasizing scalability, security, and full-lifecycle development support. The platform maintains competitive DevSecOps capabilities (4.21), though Appian achieves the highest score in this cluster (4.42).

Mendix demonstrates particular strength in Enterprise Integration and Scalability, Data Management and Interoperability, and Rapid Development and Agility. However, the platform exhibits notable weakness in DevSecOps and Compliance Integration, the lowest score among all platforms for this cluster, suggesting potential gaps in security-focused development lifecycle features.

Microsoft Power Apps performs especially well in Complexity Reduction and Accessibility and Rapid Development and Agility, reflecting its design philosophy of enabling citizen developers within the Microsoft 365 ecosystem. However, the platform exhibits the lowest DevSecOps score among all platforms assessed.

Appian exhibits strong performance in Enterprise Integration and Scalability, Governance and Security Control Mechanisms, and Data Management and Interoperability. In addition, its highest score in Platform Maturity and Ecosystem, together with solid Pricing Competitiveness, positions Appian favourably for organizations that prioritize process automation and place significant importance on vendor ecosystem considerations.

7.2. Inter-LLM Agreement Analysis

The multi-LLM assessment methodology generated four independent evaluations per platform, enabling analysis of inter-model agreement patterns. Understanding these patterns provides insight into assessment reliability and identifies capability areas where interpretation complexity affects evaluation consistency. Table 11 presents the agreement rates by platform.

Table 11 – Inter-LLM Agreement Metrics by Platform

Platform	Perfect Agreement	Minor Disagreement	High Disagreement
Mendix	22 (35.5%)	30 (48.4%)	10 (16.1%)
OutSystems	27 (43.5%)	27 (43.5%)	8 (12.9%)
Appian	20 (32.3%)	34 (54.8%)	8 (12.9%)
Power Apps	22 (35.5%)	32 (51.6%)	8 (12.9%)

OutSystems demonstrated the highest inter-LLM agreement rate, potentially reflecting the platform's extensive documentation and clear capability delineation that enabled more consistent LLM interpretation. Appian showed the lowest perfect agreement rate but the highest minor disagreement rate, suggesting evaluators found sufficient evidence but interpreted nuances differently. Mendix exhibited the highest high disagreement rate, concentrated primarily in DevSecOps and Limitations clusters.

7.3. Distribution of Rating Differences

Across all 248 item-platform assessments (62 items × 4 platforms), the distribution of maximum inter-LLM rating differences is presented in Table 12.

Table 12 – Distribution of Maximum Rating Differences

Maximum Rating Difference	Count	Percentage
0 (Complete Consensus)	91	36.7%
1 (Minor Disagreement)	123	49.6%
2 (Moderate Disagreement)	29	11.7%
≥3 (High Disagreement)	5	2.0%
Total Assessments	**248**	**100%**

As shown in Table 12, the majority of assessments exhibited either complete consensus or minor differences of one point on the five-point Likert scale. This agreement rate substantially exceeds the 70% threshold commonly accepted for adequate inter-rater reliability in exploratory research (Miles & Huberman, 1994), validating the multi-rater triangulation approach. The remaining assessments

required human validation due to disagreements exceeding one point, consistent with the methodology design where expert adjudication resolves cases of significant inter-model disagreement.

7.4. Section-Level Disagreement Patterns

Analysis of disagreement patterns by thematic cluster reveals systematic variation in assessment difficulty across capability dimensions (Table 13).

Table 13 – Section-Level Disagreement Analysis

Part	Thematic Cluster	High	Assessment
A	Rapid Development and Agility	0	Low
B	Complexity Reduction and Accessibility	1	Low
C	Enterprise Integration and Scalability	0	Low
D	Governance and Security Control Mechanisms	2	Low
E	Collaborative Development Support	4	Moderate
F	DevSecOps and Compliance Integration	13	High
G	Role Separation and Domain Specialization	2	Low
H	Data Management and Interoperability	0	Low
I	Platform Maturity and Ecosystem	2	Low
J	Limitations and Constraints	10	High

Section F (DevSecOps and Compliance Integration) emerged as the most contentious assessment area, exhibiting the highest validation rate among all sections. This pattern reflects multiple factors: DevSecOps capabilities in low-code platforms are often inconsistently documented, inherently difficult to evaluate through secondary sources, and represent an emerging capability area where platform offerings are rapidly evolving. The concentration of human validation in this section, combined with the predominantly downward direction of expert corrections, confirms DevSecOps as an evaluation dimension where vendor documentation tends to overstate actual capability maturity. Specifically, items F2, F3, F4, F5, and F6 demonstrated elevated disagreement across multiple platforms.

Section J (Limitations and Constraints) also showed elevated disagreement, exhibiting the second-highest validation rate. Unlike Section F, expert corrections in Section J were more balanced between increases and decreases, reflecting the inherently contested nature of platform limitation assessments where reasonable experts may disagree based on different interpretive frameworks.

Together, Sections F and J accounted for approximately two-thirds of all validated items, confirming these as the most challenging assessment dimensions for LLM-based evaluation.

7.5. Human Validation Findings

Human expert validation was applied to all items where the maximum inter-LLM rating difference exceeded one point on the Likert scale, representing instances where model disagreement exceeded acceptable thresholds. This validation was conducted by domain experts, each possessing substantial professional experience with the assessed platforms. Each expert was assigned responsibility for validating their respective platform's assessments, ensuring that validation decisions were informed by direct implementation experience rather than secondary source interpretation alone.

Experts received the evaluation framework spreadsheet (Appendix A) with items exhibiting inter-LLM rating differences greater than one point highlighted for review. For each flagged item, experts were provided with the item description, the Shadow IT mitigation rationale, the research foundation citation, and the four independent LLM ratings. The validation task required experts to assign a final rating and document their justification.

The validation approach was intentionally flexible rather than prescriptive. Experts were encouraged to use LLM tools as a starting point for their analysis but were explicitly instructed to validate outputs against their own professional knowledge and field experience. This approach leveraged the structured reasoning capabilities of large language models while ensuring that final assessments reflected practitioner judgment grounded in actual platform implementation.

Thirty-four items across the four platforms required human validation, representing 13.7% of total assessments. Table 14 details the items requiring intervention by platform.

Table 14 – Items Requiring Human Validation

Mendix (10)	OutSystems (8)	Appian (8)	Power Apps (8)	All Platforms
B2, D3, E2	E3, F2, F3	E4, F4, G4	D3, E2, F3	F4 (all 4)
F1, F4, F6	F4, F5, F6	I6, J1, J2	F4, F5, F6	J1, J2 (3 each)
G4, J1, J2, J4	J1, J2	J3, J4	I6, J6	

Two thematic clusters were identified as consistently problematic across all evaluated platforms. These clusters reveal areas where systematic challenges persist despite differences in platform design and capability focus.

The first problematic cluster concerns DevSecOps and Compliance Integration (Section F). Within this cluster, Item F4, which relates to Security Best Practices Guidance, required human validation for

all four platforms and was the only item to exhibit universally high levels of disagreement. This recurring pattern indicates that security governance capabilities are either inconsistently documented or subject to varying interpretations across the low-code development platform ecosystem.

The second problematic cluster is Limitations and Constraints (Section J). Item J1, addressing Customization Restrictions, and Item J2, addressing Vendor Lock-In Risk, each required human validation for three of the four platforms evaluated. These items assess inherently subjective trade-offs where platform-specific context significantly influences appropriate ratings, necessitating expert judgment to calibrate scores appropriately.

Human validation corrections exhibited a consistent pattern: all four validators either lowered their platform's scores or made balanced corrections. Mendix and Power Apps validators decreased scores most substantially, while the OutSystems validator also decreased scores and the Appian validator made balanced corrections. This pattern was particularly pronounced in Section F (DevSecOps), where the vast majority of human corrections were decreases, with validators across all platforms concluding that LLMs had overestimated DevSecOps capabilities based on vendor documentation.

This systematic pattern of downward corrections reflects a fundamental characteristic of LLM-based evaluation: large language models assess capabilities based on documented features, whereas human experts with implementation experience recognise the gap between vendor documentation and actual deployment maturity. The consistency of this pattern across independent validators—each evaluating a different platform—suggests that the methodology effectively identified and addressed its primary systematic tendency, namely LLM overestimation of security automation features. This finding has broader methodological implications: LLM-assisted assessment may require higher validation rates for capability domains where documentation typically exceeds implementation reality.

With human validation having addressed the items exhibiting significant disagreement, the analysis now turns to comparative assessment across platforms.

7.6. Cross-Platform Comparative Analysis

Comparative analysis across platforms reveals distinctive capability profiles suitable for different organizational contexts (Table 15).

Table 15 – Platform Capability Leadership by Section

Part	Thematic Cluster	Leader	Score	Margin
A	Rapid Development and Agility	OutSystems	4.88	+0.08
B	Complexity Reduction and Accessibility	Power Apps	4.75	+0.17
C	Enterprise Integration and Scalability	Mendix/Appian	4.79	+0.04
D	Governance and Security Control Mechanisms	OutSystems	4.81	+0.03
E	Collaborative Development Support	OutSystems	4.60	+0.30
F	DevSecOps and Compliance Integration	Appian	4.42	+0.21
G	Role Separation and Domain Specialization	OutSystems	4.79	+0.21
H	Data Management and Interoperability	OutSystems	4.92	+0.17
I	Platform Maturity and Ecosystem	Appian	4.71	+0.12
J	Limitations and Constraints	OutSystems	3.54	+0.29

OutSystems demonstrates capability leadership in eight of ten sections, with particularly notable advantages in Collaborative Development Support (+0.30 over nearest competitor) and Data Management and Interoperability (+0.17). Power Apps leads in Complexity Reduction and Accessibility, reflecting its design emphasis on citizen developer empowerment. Appian leads in Platform Maturity and Ecosystem, aligning with its process automation heritage. Mendix shares leadership in Enterprise Integration and Scalability with Appian.

However, these leadership positions do not eliminate platform-specific weaknesses, and strong performance in certain capability areas does not preclude the presence of important limitations. Indeed, each evaluated platform presents specific capability gaps and areas of concern that warrant careful consideration at the organizational level. These gaps highlight potential risks or limitations that may affect platform suitability depending on an organization's requirements and maturity.

As reported in Table 10, Mendix and Microsoft Power Apps both demonstrate a notable gap in DevSecOps and Compliance Integration (Section F). In comparison with OutSystems, Mendix shows a deficit of 1.08 points, while Microsoft Power Apps exhibits a slightly larger deficit of 1.16 points. As a result, organizations operating under strict security and regulatory compliance requirements should carefully assess the extent to which these platforms support a secure and compliant development lifecycle.

Appian, by contrast, shows limitations related to role architecture. Its comparatively lower score in Role Separation and Domain Specialization, together with a median score for Item G4 (Component

Specialization), indicates constraints in the platform's ability to support complex organizational role structures.

Finally, all platforms demonstrate limitations that are reflected in their scores for Section J. With values ranging from 3.21 to 3.54, these results confirm well-documented constraints associated with low-code development approaches, including risks related to vendor lock-in, restrictions on customization, and the learning curve required for effective adoption.

Despite these shared limitations, the comparative analysis also reveals important differences in how platforms perform across capability dimensions. The assessment highlights meaningful platform differentiation despite the generally high scores achieved across all evaluated platforms. Although overall performance levels indicate strong enterprise suitability, closer analysis reveals variations that may influence platform selection depending on organizational priorities.

The total score variance between platforms further illustrates this differentiation. The 0.22-point difference separating the highest-scoring platform, OutSystems, from the lowest-scoring platform, Mendix, corresponds to a 4.9% performance differential. While all platforms exceed the 4.0 threshold commonly associated with enterprise readiness, this gap may be consequential for organizations that place particular emphasis on specific capability dimensions.

Additional insights emerge when considering consistency versus specialization. OutSystems exhibits the most uniform performance across assessment sections, with a standard deviation of 0.39, indicating a balanced capability profile that supports a wide range of use cases. In contrast, Microsoft Power Apps shows greater variability in its scores, with a standard deviation of 0.53. This pattern reflects pronounced strengths in accessibility alongside weaknesses in DevSecOps, suggesting a design optimized for citizen developer enablement that may come at the expense of more advanced enterprise security automation capabilities.

7.7. LLM Reliability Analysis

Analysis of individual LLM rating tendencies reveals systematic patterns that inform interpretation of the triangulated results. Table 16 presents the average ratings by LLM model across all platforms analysed.

Table 16 – Average Ratings by LLM Across All Platforms

LLM Model	Average Rating	Rating Tendency
Gemini	4.60	Highest
Grok	4.54	Above Average
Claude	4.44	Average
ChatGPT	4.34	Lowest

Gemini exhibited the highest average ratings, potentially reflecting either access to more current training data, a tendency toward generous interpretation of capability claims, or systematic differences in how the model weighs positive versus negative evidence. ChatGPT demonstrated the most conservative ratings, which may indicate stricter evidentiary standards or greater sensitivity to documented limitations.

To address these differences and ensure robustness in the evaluation process, the multi-LLM triangulation methodology effectively mitigated individual model biases through averaging. The observed 0.26-point spread between highest and lowest LLM averages was substantially reduced in final ratings through the averaging process and human validation of outliers. This finding supports the methodological choice of multi-model triangulation over single-LLM assessment for academic research applications.

The human validation process did not reveal systematic patterns of model-specific errors, given that disagreements appeared distributed across assessment dimensions rather than concentrated in specific capability areas or reflecting consistent model biases. This suggests that variance reflects genuine interpretive complexity inherent to the assessment task rather than model-specific calibration errors.

CHAPTER 8

Discussion

This chapter synthesizes the research findings, discusses their implications for organizations seeking to leverage Low-Code Development Platforms for Shadow IT mitigation, and provides actionable recommendations for practitioners. The chapter revisits the research questions posed in Chapter 1, articulates the theoretical and practical contributions of this study, acknowledges its limitations, and suggests avenues for future research.

8.1. Addressing the Research Questions

This research was guided by three interrelated research questions. This section revisits each question and summarizes how it was addressed through the systematic investigation conducted in Chapters 2 through 6.

Research Question 1: What are the primary challenges associated with software development and Shadow IT in large enterprises?

Chapter 3 established the theoretical foundation through a PRISMA-compliant systematic literature review of 33 articles, identifying Shadow IT as a pervasive phenomenon driven by the gap between business unit needs and IT department responsiveness. Chapter 4 synthesized these findings into three primary challenge domains: the IT department responsiveness gap, where formal IT processes fail to meet business velocity requirements [10], [14]; application complexity and poor user experience, where enterprise systems impose cognitive burdens that drive users toward simpler unauthorized alternatives [17]; and security, compliance, and governance consequences, where unsanctioned solutions create vulnerabilities, data fragmentation, and regulatory exposure [11], [12].

Research Question 2: What specific key features and capabilities must Low-Code Development Platforms possess to effectively address these challenges?

Chapter 4 identified nine capability domains that LCDPs must possess to effectively mitigate Shadow IT challenges. These domains were derived from synthesis of the academic literature, including the Business-Managed IT critical success factors framework [19], the LowDevSecOps RAAFT framework [2], and the four-role architecture proposed by Mottu and Sunyé [20]. The mapping matrix presented in Table 6 demonstrated the logical connections between identified Shadow IT challenges and specific LCDP capabilities.

Research Question 3: What evaluation framework using Likert scale methodology should organizations employ to systematically compare and assess LCDP vendors against the key features and capabilities identified?

Chapter 5 operationalized the capability domains into a structured 62-item evaluation framework organized across ten thematic clusters (Parts A–J). Each item was grounded in the research literature through explicit Research Foundation citations, ensuring that evaluation criteria reflected academically validated Shadow IT mitigation requirements rather than generic platform feature checklists. Chapter 6 detailed the multi-LLM triangulation methodology for applying the framework, while Chapter 7 demonstrated its practical application through systematic assessment of four enterprise LCDP platforms: Mendix, OutSystems, Appian, and Microsoft Power Apps.

8.2. Key Findings

The assessment produced several findings with important implications for both theoretical understanding and practical application. Taken together, these results provide insight into platform performance, shared strengths and limitations, and the reliability of the evaluation methodology.

In terms of overall platform ranking, OutSystems emerged as the highest-scoring platform, followed by Appian, Microsoft Power Apps, and Mendix. All four platforms achieved scores above 4.0 on the five-point Likert scale, indicating general enterprise suitability for the purpose of mitigating Shadow IT. The 0.22-point difference between the highest- and lowest-scoring platforms corresponds to a 4.9% performance differential, demonstrating that meaningful differentiation exists despite the uniformly high scores.

Across platforms, several universal strengths were observed. All assessed platforms performed strongly in Rapid Development and Agility, in Enterprise Integration and Scalability, and in Data Management and Interoperability. These capability areas represent core requirements for Shadow IT mitigation, suggesting that leading enterprise low-code development platforms have converged on essential functionality in these domains.

At the same time, universal limitations were evident. Scores for Section J were consistently low across all platforms, confirming that customization restrictions, vendor lock-in risks, and scalability constraints remain recognized challenges in low-code development. These findings validate the limitations framework proposed by Martinez and Pfister [30]. Within this cluster, vendor lock-in (Item J2) emerged as the most prominent concern among all platforms.

Significant variation was identified in DevSecOps maturity across platforms. Mendix and Microsoft Power Apps are more than one point below OutSystems. Moreover, Section F exhibited the highest level of inter-LLM disagreement, with 13 item–platform combinations showing high disagreement,

making it the most contested assessment area. This pattern indicates that DevSecOps integration remains an evolving capability domain characterized by inconsistent documentation and rapidly changing platform offerings.

Clear differences in platform capability leadership were also observed. OutSystems demonstrates leadership in eight of the ten assessment sections, with particularly strong advantages in Collaborative Development Support, where it holds a margin of +0.30 over competitors, and in Data Management and Interoperability, with a +0.17 margin. Microsoft Power Apps leads in Complexity Reduction and Accessibility, while Appian leads in Platform Maturity and Ecosystem. Mendix shares leadership with Appian in Enterprise Integration and Scalability.

Finally, the reliability of the assessment methodology was supported by the results of the multi-LLM triangulation approach. Across 248 item–platform assessments, the methodology achieved 86.3% consensus or minor disagreement, while human validation successfully resolved the remaining 13.7% of cases that exhibited significant inter-model variance. These findings support the feasibility of LLM-assisted assessment methodologies for academic research when they are complemented by appropriate validation procedures.

8.3. Theoretical Contributions

This research makes two primary contributions to the academic literature. The first is the development of a 62-item evaluation framework, presented in Chapter 5, which addresses a clearly identified gap in existing research. Although prior studies have examined Shadow IT drivers [10], low-code development platform capabilities [20], and governance mechanisms [19] as separate lines of inquiry, no prior framework has systematically operationalized these constructs into a structured assessment instrument specifically intended to evaluate Shadow IT mitigation. The framework's grounding in Research Foundation citations ensures its theoretical validity, while its operationalization through a Likert-scale enables practical application. Furthermore, the evaluation of four market-leading platforms demonstrates the framework's discriminative validity by differentiating platforms across capability dimensions, while also confirming convergent validity through consistently high scores in capability areas that are universally strong.

The second contribution lies in the multi-LLM triangulation methodology employed in the assessment. By combining evaluations from four large language models with human expert validation, the study contributes to the emerging body of scholarship on LLM-assisted research methods. The methodology achieves reliability through two complementary mechanisms: the ICC(2,k) of 0.83 indicates good reliability for averaged ratings according to established interpretation guidelines [38],

while the agreement rate substantially exceeds the 70% threshold for inter-rater reliability in exploratory research [8].

Beyond these quantitative indicators, the observed patterns of inter-model agreement and disagreement offer empirical insights for researchers considering similar methodological designs. The concentration of validation requirements in Sections F (DevSecOps) and J (Limitations) reveals that certain evaluation dimensions present inherent complexity that challenges automated assessment. More significantly, the systematic pattern of downward corrections by human validators suggests that LLMs, when evaluating based on vendor documentation, tend to overestimate platform capabilities. This finding implies that LLM-assisted assessment may be particularly valuable for initial screening and systematic comparison, but should be complemented by practitioner validation for capability domains where documentation typically exceeds implementation maturity. The modest net change in overall ratings demonstrates that human validation corrected outliers rather than systematically shifting the dataset, supporting the hybrid approach where LLM consensus is accepted for clear cases while expert judgment resolves contested assessments.

It is important to acknowledge that the multi-LLM triangulation methodology represents an exploratory contribution to research methods rather than a validated instrument. The approach demonstrates feasibility for structured secondary-source assessment but requires further methodological development before it can be considered a standardized research tool. The 86.3% agreement rate provides preliminary evidence of reliability, but systematic comparison with traditional expert panel assessment would strengthen confidence in the methodology. This research therefore, contributes both a substantive evaluation framework for LCDP assessment and a methodological prototype for LLM-assisted systematic evaluation that future research can refine and validate.

8.4. Main Conclusions

This research investigated how Low-Code Development Platforms may mitigate Shadow IT issues in large enterprises. Through systematic literature review, framework development, and multi-platform assessment, several conclusions emerge. First, Shadow IT persists as a significant challenge in large enterprises, driven primarily by the gap between business unit agility requirements and IT department responsiveness. The proliferation of unauthorized solutions creates security vulnerabilities, compliance risks, and governance fragmentation that traditional prohibition-based approaches have failed to address.

Second, Low-Code Development Platforms offer a promising mechanism for Shadow IT mitigation by enabling business users to develop solutions within governed environments. The essential

capabilities for effective mitigation span nine domains, with rapid development, enterprise integration, and governance controls representing foundational requirements.

Third, leading enterprise LCDPs (i.e., Mendix, OutSystems, Appian, and Microsoft Power Apps) demonstrate general suitability for Shadow IT mitigation, with overall scores exceeding 4.0 on a five-point scale. However, meaningful differentiation exists across platforms, particularly in DevSecOps capabilities where both Mendix and Power Apps exhibit notable gaps relative to OutSystems and Appian.

Fourth, all assessed platforms exhibit acknowledged limitations, including vendor lock-in risks and customization constraints. Organizations should plan for these limitations regardless of platform selection rather than expecting any single LCDP to fully eliminate Shadow IT challenges.

Fifth, the multi-LLM triangulation methodology demonstrates viability for systematic platform assessment when combined with human expert validation. The approach achieved acceptable reliability while enabling assessment scope that would be impractical through manual evaluation alone. Importantly, the analysis reveals that LLMs tend to overestimate platform capabilities when evaluating based on vendor documentation, particularly for security-focused features. Human experts with implementation experience consistently provided more conservative assessments, suggesting that vendor documentation provides an optimistic baseline that benefits from practitioner validation.

Finally, platform consistency varies meaningfully across vendors. OutSystems demonstrates the most consistent scoring across sections (standard deviation: 0.39), suggesting a balanced platform suitable for diverse use cases. Power Apps exhibits the greatest variance (standard deviation: 0.53), indicating focused optimization for citizen developer scenarios with potential trade-offs in enterprise security automation.

8.5. Recommendations

The findings of this research support a differentiated set of recommendations targeting three key professional audiences involved in Low-Code Development Platform (LCDP) adoption and governance: IT decision-makers, leaders of citizen developer programs, and stakeholders responsible for enterprise-scale deployments. While these roles face distinct priorities, the results suggest that effective Shadow IT mitigation requires coordinated consideration across all three perspectives rather than isolated decision-making.

From the perspective of IT decision-makers, the proposed evaluation framework and the resulting platform assessments provide an evidence-based foundation for LCDP vendor selection. Rather than treating platform choice as a one-dimensional decision, organizations are advised to weight evaluation dimensions according to the specific drivers of Shadow IT they face. Where Shadow IT is primarily

driven by responsiveness and time-to-market pressures, greater emphasis should be placed on capabilities related to rapid development and complexity reduction (Parts A and B). Conversely, in environments where Shadow IT presents significant security, compliance, or governance risks, decision-makers should prioritize governance and DevSecOps-related capabilities (Parts D and F). The analysis further indicates that DevSecOps capabilities remain one of the most challenging areas to assess reliably through secondary sources alone. The elevated disagreement observed in this cluster suggests that security automation and compliance features are often inconsistently documented or are evolving rapidly. Consequently, organizations operating under stringent regulatory or security requirements should complement framework-based assessment with hands-on evaluation of platform security capabilities. In this context, particular attention should be paid to the documented DevSecOps gaps identified for Mendix and Microsoft Power Apps. Across all platforms, the findings also underscore the importance of planning proactively for known limitations. Vendor lock-in risks, data portability constraints, and long-term architectural flexibility should be addressed early in LCDP adoption roadmaps. The assessment results suggest that such concerns are more pronounced for Mendix and Power Apps than for OutSystems and Appian, reinforcing the need for mitigation strategies regardless of platform choice.

For organizations designing or expanding citizen developer programs, the results point to a partially different emphasis. Microsoft Power Apps' strong performance in complexity reduction and accessibility highlights its suitability for initiatives that prioritize broad citizen developer participation, particularly in organizations already embedded in the Microsoft ecosystem. Integration with familiar tools such as Excel, SharePoint, and Teams can lower adoption barriers and accelerate time-to-value for business users. At the same time, the findings indicate that organizations relying on Power Apps for citizen development should compensate for its comparatively weaker DevSecOps capabilities through complementary governance mechanisms. These may include stricter security review processes for citizen-developed applications, integration with enterprise security tooling, or limiting citizen developer autonomy to lower-risk use cases while reserving more complex or sensitive applications for professional development teams. Mendix also emerges as a viable option in this context, as it combines strong accessibility characteristics with robust enterprise integration and scalability, making it suitable for citizen developer programs that require close coupling with existing enterprise systems.

Finally, for stakeholders responsible for enterprise-scale deployments, the assessment highlights a different set of trade-offs. OutSystems' consistently high scores across evaluation dimensions, together with its strong performance in data management, interoperability, suggest that it is well suited for organizations seeking a comprehensive, standardized platform capable of supporting diverse use cases under mature governance structures. Its relatively low variance across assessment sections

indicates a balanced capability profile, which is particularly advantageous for organizations aiming to consolidate development efforts on a single enterprise-wide platform, especially in regulated industries with stringent security automation requirements. Appian presents a compelling alternative for organizations with strong business process management needs and sensitivity to cost considerations, supported by its strengths in enterprise integration, platform maturity, and ecosystem. However, its comparatively lower support for role separation and domain specialization indicates that organizations seeking to implement the four-role architecture proposed in the literature may need to complement the platform with additional governance structures. Mendix, while ranking lower overall, demonstrates particular strength in enterprise integration, scalability, and data management, making it suitable for organizations that prioritize system integration capabilities. Nonetheless, its documented DevSecOps gaps and higher reliance on human validation suggest that organizations with complex security requirements should conduct thorough hands-on evaluation before committing to large-scale deployment.

8.6. Limitations

Several limitations must be taken into account when interpreting the results of this study. First, the assessment represents a point-in-time evaluation. Because platform capabilities evolve rapidly, the findings reflect documented features and functionality available as of late 2025 and may not account for more recent releases or items on vendor roadmaps. Consequently, organizations should validate current platform capabilities against this baseline before making adoption decisions.

Second, the study relies on a secondary-source methodology. The assessment was based on large language model interpretation of documentation and published materials rather than direct, hands-on platform evaluation. As a result, certain nuances related to implementation quality or practical usability may not have been captured. For this reason, direct platform testing remains a critical step in final vendor selection processes.

Additional limitations arise from the complexity of assessing DevSecOps capabilities. The elevated level of disagreement observed in Section F, which included 13 high-disagreement item–platform combinations, suggests that evaluating DevSecOps features through secondary sources alone may be insufficiently reliable. Organizations are therefore advised to complement such assessments with direct platform evaluation when DevSecOps maturity is a key consideration.

Interpretive challenges were also identified in relation to Section J, which employs an inverted scoring logic where higher scores indicate lower risk. This design proved difficult for consistent interpretation by large language models and resulted in 10 high-disagreement items across platforms.

Future research should investigate alternative approaches for assessing platform constraints to improve interpretive consistency.

Finally, while human validation successfully resolved items with high inter-model disagreement, the validation requirement introduces the potential for human bias. Notably, validation was concentrated in specific evaluation dimensions: Sections F (DevSecOps) and J (Limitations) accounted for approximately two-thirds of all validated items. While this concentration aligns with the inherent complexity of these assessment areas, it suggests that LLM-based evaluation may require higher validation rates for security-focused and constraint-oriented assessments. To mitigate the risk of individual bias in future studies, researchers should consider involving multiple expert validators per platform and incorporating formal inter-rater reliability analysis.

The LLM-assisted assessment methodology employed in this research represents a novel approach that lacks established validation benchmarks. While the multi-LLM triangulation achieved 86.3% consensus or minor disagreement across 248 item-platform assessments, traditional inter-rater reliability metrics such as Krippendorff's alpha were not calculated due to the exploratory nature of this methodological approach. The four-expert human validation panel, while providing domain expertise for adjudicating disagreements, represents a small sample that may introduce individual bias. Future research should consider employing larger validation panels with formal inter-rater reliability assessment. Additionally, the quantitative assessment (N=4 LLM raters) does not meet conventional thresholds for statistical generalizability; however, the methodology was designed as a triangulation mechanism to reduce single-model bias rather than as a traditional survey instrument. These limitations should be considered when interpreting the platform assessment results.

8.7. Future Research

This research focused on Low-Code Development Platforms as a mechanism for Shadow IT mitigation. However, the technology landscape continues to evolve, and a significant emerging phenomenon warrants investigation: Shadow AI.

Shadow AI extends the Shadow IT paradigm to encompass unauthorized use of artificial intelligence tools, including large language models, within organizational contexts. As documented in Chapter 4, Shadow AI introduces distinct risks beyond traditional Shadow IT: employees may input sensitive corporate data into public AI services, AI-generated outputs may contain hallucinated information that propagates into business decisions, and the opacity of AI reasoning creates accountability challenges that differ qualitatively from conventional software risks.

Future research should investigate how the evaluation framework developed in this study might be extended to address Shadow AI mitigation. Key research questions include: What additional

capability domains are required for platforms to effectively govern AI tool usage? How do the security, compliance, and governance implications of Shadow AI differ from traditional Shadow IT? What role might enterprise LCDP platforms play in providing governed AI capabilities that reduce the appeal of unauthorized alternatives?

The intersection of low-code development and artificial intelligence represents a particularly promising research direction, as LCDP vendors increasingly integrate AI-assisted development features. Understanding how these integrated AI capabilities affect Shadow IT dynamics – that is, whether they amplify or mitigate unauthorized tool adoption – would extend the theoretical contributions of this research to the rapidly evolving AI-augmented enterprise context.

Additionally, longitudinal research tracking the evolution of platform capabilities against this assessment baseline would provide valuable insights into LCDP market maturation. The DevSecOps capability gaps identified in this assessment, particularly for Mendix and Power Apps, represent areas where rapid vendor improvement may alter the competitive landscape. Periodic reassessment using the established framework would enable tracking of such evolution while validating the framework's ongoing relevance.

References

[1] C. Ploder, R. Bernsteiner, S. Schlögl, and C. Gschliesser, 'The Future Use of LowCode/NoCode Platforms by Knowledge Workers – An Acceptance Study', in *Knowledge Management in Organizations*, vol. 1027, L. Uden, I.-H. Ting, and J. M. Corchado, Eds., in Communications in Computer and Information Science, vol. 1027, Cham: Springer International Publishing, 2019, pp. 445–454. doi: 10.1007/978-3-030-21451-7_38.

[2] G. Sedrakyan, M.-E. Iacob, and J. Hillegersberg, 'Towards LowDevSecOps Framework for Low-Code Development: Integrating Process-Oriented Recommendations for Security Risk Management', *Proc. MODELS 2024 - ACM/IEEE 27th Int. Conf. Model Driven Eng. Lang. Syst. Companion Proc.*, pp. 886–894, 2024, doi: 10.1145/3652620.3688335.

[3] A. Viljoen *et al.*, 'Navigating Flexibility and Standardisation in Low-Code/No-Code Development', *Inf. Syst. J.*, 2025, doi: 10.1111/isj.70001.

[4] J. W. Creswell and J. D. Creswell, *Research design: qualitative, quantitative, and mixed methods approaches*, Fifth edition. Los Angeles London New Delhi Singapore Washington DC Melbourne: SAGE, 2018.

[5] A. R. Hevner, S. T. March, J. Park, and S. Ram, 'Design Science in Information Systems Research', *MIS Q.*, vol. 28, no. 1, pp. 75–106, Mar. 2004, doi: 10.2307/25148625.

[6] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, 'A Design Science Research Methodology for Information Systems Research', *J. Manag. Inf. Syst.*, vol. 24, no. 3, pp. 45–77, Dec. 2007, doi: 10.2753/MIS0742-1222240302.

[7] P. E. Shrout and J. L. Fleiss, 'Intraclass correlations: Uses in assessing rater reliability.', *Psychol. Bull.*, vol. 86, no. 2, pp. 420–428, 1979, doi: 10.1037/0033-2909.86.2.420.

[8] M. B. Miles and A. M. Huberman, *Qualitative data analysis: an expanded sourcebook*, 2. ed., [Nachdr.]. Thousand Oaks, Calif.: Sage, 1994.

[9] C. Okoli and S. D. Pawlowski, 'The Delphi method as a research tool: an example, design considerations and applications', *Inf. Manage.*, vol. 42, no. 1, pp. 15–29, Dec. 2004, doi: 10.1016/j.im.2003.11.002.

[10] R. Trialih, 'The Challenge in Neutralizing Shadow IT: A Literature Review', *2023 IEEE Int. Conf. Ind. Eng. Eng. Manag. IEEM 2023*, pp. 1169–1173, 2023, doi: 10.1109/IEEM58616.2023.10406586.

[11] O. F. Alaba, Y. Y. A. Talib, and F. H. Rusly, 'Understanding Shadow IT usage: The interplay of IT accessibility, work overload, and employee strain', *RAM Rev. Adm. Mackenzie*, vol. 26, no. 1, 2025, doi: 10.1590/1678-6971/eramr250152.

[12] H.-H. Huang and J.-W. Lin, 'Inconsistencies Between Information Security Policy Compliance and Shadow IT Usage', *J. Comput. Inf. Syst.*, vol. 64, no. 4, pp. 554–564, 2024, doi: 10.1080/08874417.2023.2234318.

[13] N. Castellano, C. Felden, and R. Pinsker, ‘Shadow IT Behavior of Financial Executives in Germany and Italy as an Antecedent to Internal Data Security Breaches’, in *Proceedings of the Annual Hawaii International Conference on System Sciences*, Hawaii International Conference on System Sciences, 2022. doi: 10.24251/hicss.2022.765.

[14] S. Haag and A. Eckhardt, ‘Dealing Effectively with Shadow IT by Managing Both Cybersecurity and User Needs’, *MIS Q. Exec.*, vol. 23, no. 4, pp. 399–412, 2024, doi: 10.17705/2msqe.00104.

[15] I. Scalabrin Bianchi, A. Vaquina, R. Pereira, R. Dinis Sousa, and G. A. Dávila, ‘A Benefit Dependency Network for Shadow Information Technology Adoption, Based on Practitioners’ Viewpoints’, *Informatics*, vol. 9, no. 4, p. 95, Nov. 2022, doi: 10.3390/informatics9040095.

[16] M. S. White, ‘Workarounds and shadow IT – balancing innovation and risk’, *Bus. Inf. Rev.*, vol. 40, no. 3, pp. 114–122, 2023, doi: 10.1177/02663821231188823.

[17] M. Wosny, L. M. Strasser, and J. Hastings, ‘The Paradoxes of Digital Tools in Hospitals: Qualitative Interview Study’, *J. Med. Internet Res.*, vol. 26, p. e56095, Jul. 2024, doi: 10.2196/56095.

[18] J.-P. van Acken, F. Jansen, S. Jansen, and K. Labunets, ‘Who is the IT Department Anyway: An Evaluative Case Study of Shadow IT Mindsets Among Corporate Employees’, *Proc. 20th Symp. Usable Priv. Secur. SOUPS 2024*, pp. 527–545, 2024.

[19] S. Klotz, M. Westner, and S. Strahringer, ‘Critical Success Factors of Business-managed IT: It Takes Two to Tango’, *Inf. Syst. Manag.*, vol. 39, no. 3, pp. 220–240, Jul. 2022, doi: 10.1080/10580530.2021.1938300.

[20] J.-M. Mottu and G. Sunyé, ‘Emerging New Roles for Low-Code Software Development Platforms’, in *Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems*, Linz Austria: ACM, Sep. 2024, pp. 905–914. doi: 10.1145/3652620.3688337.

[21] S. Matook, Y. M. Wang, and M. Axelsen, ‘Experiential Learning for Citizen Developers: Training IT Talent in Low-code Development and Metacognitive Reflection’, *Bus. Inf. Syst. Eng.*, vol. 67, no. 1, pp. 7–30, Feb. 2025, doi: 10.1007/s12599-024-00921-3.

[22] J. Varajão, A. Trigo, and M. Almeida, ‘Low-code Development Productivity: “Is winter coming” for code-based technologies?’, *Queue*, vol. 21, no. 5, pp. 87–107, Oct. 2023, doi: 10.1145/3631183.

[23] E. Martinez, L. Pfister, and F. Stauch, ‘Developing a novel application to digitalize and optimize construction operations using low-code technology’, in *Proceedings of the International Symposium on Automation and Robotics in Construction (IAARC)*, Lille, France: International Association for Automation and Robotics in Construction (IAARC), Jun. 2024. doi: 10.22260/isarc2024/0038.

[24] S. G. Orr *et al.*, ‘Shadow IT in higher education: survey and case study for cybersecurity’, *Cryptologia*, vol. 48, no. 1, pp. 26–90, Jan. 2024, doi: 10.1080/01611194.2022.2103754.

[25] T. Guthardt, J. Kosiol, and O. Hohlfeld, ‘Low-code vs. the developer: An empirical study on the developer experience and efficiency of a no-code platform’, in *Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems*, Linz Austria: ACM, Sep. 2024, pp. 856–865. doi: 10.1145/3652620.3688332.

[26] E. Elshan, B. Binzer, and T. J. Winkler, ‘From Software Users to Software Creators: An Exploration of the Core Characteristics of the Citizen Developer Role and the Related Re- and Upskilling Programs’, *Bus. Inf. Syst. Eng.*, vol. 67, no. 1, pp. 31–53, Feb. 2025, doi: 10.1007/s12599-024-00915-1.

[27] R. Picek, ‘Low-code/No-code Platforms and Modern ERP Systems’, *Proc. - 2023 9th Int. Conf. Inf. Manag. ICIM 2023*, pp. 44–49, 2023, doi: 10.1109/ICIM58774.2023.00014.

[28] L. Tang, ‘ERP Low-Code Cloud Development’, in *2022 IEEE 13th International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China: IEEE, Oct. 2022. doi: 10.1109/icsess54813.2022.9930146.

[29] K. Frings and G. Schuh, ‘Description And Typification Of Low-Code Use Cases In Manufacturing Companies’, *2023 IEEE Int. Conf. Technol. Manag. Oper. Decis. ICTMOD 2023*, 2023, doi: 10.1109/ICTMOD59086.2023.10438157.

[30] E. Martinez and L. Pfister, ‘Benefits and limitations of using low-code development to support digitalization in the construction industry’, *Autom. Constr.*, vol. 152, 2023, doi: 10.1016/j.autcon.2023.104909.

[31] Gartner, Inc., ‘Magic Quadrant for Enterprise Low-Code Application Platforms’, Gartner. Accessed: Mar. 22, 2026. [Online]. Available: https://www.gartner.com/en/documents/6773234

[32] J. Bratincevic, ‘The Forrester Wave™: Low-Code Platforms For Professional Developers, Q2 2025 | Forrester’. Accessed: Mar. 22, 2026. [Online]. Available: https://www.forrester.com/report/the-forrester-wave-tm-low-code-platforms-for-professional-developers-q2-2025/RES182327

[33] Everest Group, ‘Low-code Application Development Platforms PEAK Matrix® Assessment 2025 - Focus on User Engagement - Everest Group Research Portal’. Accessed: Mar. 22, 2026. [Online]. Available: https://www.everestgrp.com/report/egr-2025-32-r-7512/

[34] A. Dayaratna, ‘IDC MarketScape: Worldwide Low-Code and No-Code Developer Technologies 2025 Vendor Assessment’, IDC: The premier global market intelligence company. Accessed: Mar. 22, 2026. [Online]. Available: https://my.idc.com/getdoc.jsp?containerId=US53669425

[35] g2, ‘Best Low-Code Development Platforms Reviews 2026 | Compare Tools’, G2. Accessed: Mar. 22, 2026. [Online]. Available: https://www.g2.com/categories/low-code-development-platforms

[36] R. Kennedy, ‘LCAP Technology Value Matrix 2025’, Nucleus Research. Accessed: Mar. 22, 2026. [Online]. Available: https://nucleusresearch.com/research/single/lcap-technology-value-matrix-2025/

[37] PeerSpot, ‘Top Rated Low-Code Development Platforms Vendors’, PeerSpot. Accessed: Mar. 22, 2026. [Online]. Available: https://www.peerspot.com/categories/low-code-development-platforms

[38] T. K. Koo and M. Y. Li, ‘A Guideline of Selecting and Reporting Intraclass Correlation Coefficients for Reliability Research’, *J. Chiropr. Med.*, vol. 15, no. 2, pp. 155–163, Jun. 2016, doi: 10.1016/j.jcm.2016.02.012.

[39] A. Techmanska and C. Stöckmann, ‘Low-Code excellence: unveiling the skill set of modern digital architects’, *Behav. Inf. Technol.*, pp. 1–21, May 2025, doi: 10.1080/0144929x.2025.2506663.

Appendices

Appendix A: The Thematic Clusters.

Appendix B: The Evaluation Items by Cluster.

Appendix C: The Assessment Output Excel file.

Appendix D: Assessment Prompt Architecture files.

Appendix A

Part A – Rapid Development and Agility: Focuses on how LCDPs shorten delivery cycles and support continuous change. This includes visual modelling, pre-built components, WYSIWYG editors, minimal manual coding, rapid cloud deployment and dynamic updates without full redeployment, all of which accelerate prototyping and reduce dependence on traditional SDLC stages. These capabilities are positioned as the primary way LCDPs offer an IT-governed alternative to fast, ad-hoc Shadow IT solutions while still meeting business time-to-market pressures.

Part B – Complexity Reduction and Accessibility: Covers how platforms hide technical complexity and open development to non-experts. Intuitive GUIs, drag-and-drop configuration, logical and structural diagrams, and strong error handling are highlighted as essential for citizen developers to build and maintain apps without formal programming skills. This directly addresses Shadow IT drivers such as frustration with complex corporate tools by giving business users an easier, sanctioned way to obtain needed functionality.

Part C – Enterprise Integration and Scalability: Addresses integration with existing landscapes and the ability to scale to enterprise workloads. The framework stresses seamless data integration with databases, SaaS and legacy systems, plus cloud-based deployment models to support elastic scaling and multi-tenant use. Strong integration and scalability are treated as preconditions for LCDPs to replace isolated Shadow IT tools with solutions that fit into centralized architectures.

Part D – Governance and Security Control Mechanisms: Groups features that let central IT retain control: role-based access, centralized policy enforcement, auditability, environment management and alignment with IT governance frameworks. The thesis frames LCDPs as viable only when accompanied by clear governance models that define who may build what, how apps are reviewed, and how risks such as data leakage and non-compliance are monitored. Without these mechanisms, low-code initiatives risk becoming “governed Shadow IT” rather than a remedy.

Part E – Collaborative Development Support: Covers capabilities that enable joint work between IT and business units: shared workspaces, versioning, commenting, and support for mixed teams of professional and citizen developers. The literature on LCDPs and citizen development is used to argue that collaboration features are necessary so domain experts can capture requirements directly while IT ensures architectural fit and quality.

Part F – DevSecOps and Compliance Integration: Includes pipeline automation, embedded security controls and compliance tooling. The thesis draws on LowDevSecOps work to emphasize model-based security checks, policy-driven gates, and integration with CI/CD as ways to ensure that low-code artifacts follow the same secure-by-design and regulatory practices as traditional software. This is particularly linked to mitigating Shadow IT risks around GDPR, HIPAA and similar regulations in large enterprises.

Part G – Role Separation and Domain Specialization: Focuses on distinguishing and supporting roles such as platform admins, professional developers, domain engineers and citizen developers. Research indicates that LCDPs increasingly formalize these roles [20], with platform features (permissions, environment separation, guided tooling) reflecting different skill levels and responsibilities, which reduces unauthorized autonomy while still leveraging domain knowledge.

Part H – Data Management and Interoperability: Groups capabilities for modelling, governing and exchanging data: visual data modelling, constraints, reuse of corporate data models, connectors and APIs, and support for consistent data semantics across apps. Effective data management is presented as crucial to avoid new data silos and to ensure that low-code apps integrate cleanly with enterprise analytics and operational systems, rather than reproducing typical Shadow IT fragmentation.

Part I – Platform Characteristics and Market Position: Summarizes cross-cutting attributes used to select vendors: cloud vs on-premises options, licensing and cost models, ecosystem maturity, vendor lock-in risk, and analyst rankings. The thesis limits its evaluation to leading LCDPs identified by market analysts and uses the A–H capability clusters as criteria to compare them, linking technical features to strategic fit in large-enterprise contexts.

Part J – Limitations and Constraints: Addresses documented platform limitations that may constrain Shadow IT mitigation effectiveness: customization restrictions that limit unique business requirements, vendor lock-in risks that complicate future migration, scalability constraints for large-scale deployments, learning curve investments required from citizen developers, feature limitations compared to traditional development, and incomplete support for the four-role architecture theoretically required for effective role separation. These limitations, empirically validated by Martinez and Pfister [30], inform realistic expectations for LCDP adoption.

Appendix B

Table 1 – Items Description and Research Foundation

Item_ID	Item_Name	Description	Research Foundation
A1	Visual Modelling Interface	The platform provides intuitive drag-and-drop visual modelling capabilities for UI, data structures, and workflows	"Visual modelling allows users to drag and drop elements on the user interface, making it easier to use, especially for developers who want to use the platform intuitively. LCDPs provide easy-to-use visual environments for creating software applications with minimal programming skills." [20, p. 4]
A2	Pre-built Components and Templates	The platform offers comprehensive libraries of reusable components and templates that serve as building blocks	"Reusability is promoted through pre-built components and templates that serve as building blocks for common functionalities, accelerating development." [20, p. 4]
A3	WYSIWYG Editor	The platform includes What You See Is What You Get editors enabling real-time preview of applications	"What You See Is What You Get (WYSIWYG) editors allow users to build and preview their application in real-time." [20, p. 4]
A4	Minimal Manual Coding	The platform requires minimal or no procedural code writing for standard functionality	"Since manual programming is largely eliminated with low-code, software developers can create their applications much faster and easier than by building them from scratch. LCDPs enable the development of fully functional software applications using advanced graphical user interfaces and visual abstractions that require

Item_ID	Item_Name	Description	Research Foundation
			minimal or no procedural code." [24, p. 2]
A5	Rapid Deployment Capabilities	The platform enables quick application deployment to production environments with minimal configuration	"Application deployment is greatly facilitated by platforms that offer cloud hosting. Cloud-based architecture enables organizations to deploy applications rapidly without investing in on-premises infrastructure." [20, p. 5]
A6	Iterative Development Support	The platform facilitates fast iteration cycles and continuous updates without extensive redeployment	"Dynamic development enables changes to applications without requiring full redeployment, reducing dependencies between development roles and allowing rapid feedback cycles." [20, p. 8]
B1	Graphical User Interface Intuitiveness	The GUI is intuitive and requires minimal training for new users, supporting both technical and non-technical users	"Visual modelling allows users to drag and drop on the user interface. This makes it easier to use, especially for developers who want to be able to use the platform intuitively. The interface should be intuitive enough to support both technical and non-technical users." [20, p. 4]
B2	Accessibility for Non-Developers	The platform is effectively usable by individuals without formal programming training (citizen developers)	"Citizen developers are business users without formal software development training but with domain expertise. NCDPs enable the creation of software applications entirely without writing code, catering to an even broader user base, including individuals with minimal technical backgrounds." [20, p. 4]

Item_ID	Item_Name	Description	Research Foundation
B3	Point-and-Click Configuration	Users can configure and build applications primarily through point-and-click interfaces without programming	"Pre-built components can be dragged and dropped onto a visual canvas to build applications. Users interact with a visual interface to drag and drop components, connect them, and configure them to produce the desired functionality." [20, p. 4]
B4	Logical Diagram Representation	The platform provides visual representations of application logic, workflows, and data flows	"The user interfaces, business logic, and data services are built using visual diagrams and, in some cases, manual coding. These features are important for enabling domain experts to model their domain and specify the application behaviour." [20, p. 4]
B5	Structural Diagram Support	The platform offers visual representation of data entities, their relationships, and dependencies	"Data modelling phase support. They model data using drag-and-drop capabilities to configure the data schema of the application. They create entities, associate them with each other, and set constraints and dependencies." [20, p. 8]
B6	Advanced Error Management	The platform provides clear, actionable error messages and comprehensive debugging support	"The platform should provide comprehensive support for debugging and error management. Clear error messages and accessible debugging tools are essential for enabling citizen developers to resolve issues independently." [25, p. 4]
C1	Seamless Data Integration	The platform seamlessly integrates with existing systems, databases, and data sources (SQL, cloud, spreadsheets)	"Seamless integration of data from various sources—databases, spreadsheets, cloud applications—allows citizen developers to leverage existing data. Data integration is critical

Item_ID	Item_Name	Description	Research Foundation
			for enterprise adoption and supporting scalability." [20, p. 4]
C2	API Integration Support	The platform provides robust API integration capabilities for connecting to external services and third-party systems	"Integration of external services via third-party APIs and application deployment. Seamless integration with data from various sources including databases, spreadsheets, and cloud applications enables citizen developers to leverage existing data." [27, p. 8]
C3	Cloud Infrastructure Support	The platform is deployed on cloud infrastructure enabling scalability, availability, and geographic distribution	"Cloud-based environments that provide features such as advanced graphical user interfaces, visual modelling tools, drag-and-drop functionalities, reusable components, and declarative programming languages. Application deployment is greatly facilitated by platforms that offer cloud hosting." [39, p. 5]
C4	Data Import/Export Flexibility	The platform supports flexible data import/export capabilities in multiple formats (CSV, JSON, XML, etc.)	"Seamless integration of data from various sources enables citizen developers to leverage existing data. The platform should support customizing import and export features for deployment across the enterprise." [27, p. 8]
C5	Application Scalability	The platform supports scalability to handle growing user bases and data volumes without performance degradation	"Scalability refers to the ability to expand the use of a given application to larger systems, projects, or organizations. The platform must support scalability to ensure that

Item_ID	Item_Name	Description	Research Foundation
			applications can grow with organizational needs." [24, p. 3]
C6	Multi-Project Management	The platform provides tools for managing multiple projects simultaneously with centralized control and local customization	"The platform should enable organizations to manage and track multiple low-code applications across the enterprise. This is essential for maintaining governance and preventing uncontrolled Shadow IT proliferation." [27, p. 9]
D1	Role-Based Access Control (RBAC)	The platform implements role-based access control enabling differentiation of user permissions based on roles	"Multi-layered access controls including role-based access management are critical to enterprise adoption. RBAC enables organizations to define roles and assign permissions, supporting centralized IT governance. Access control defines specific access rights for users based on their roles and responsibilities." [2, p. 5]
D2	Role Creation and Management	The platform allows dynamic creation, modification, and management of custom roles	"The platform must support role-based access management enabling IT administrators to define custom roles and assign appropriate permissions. Administrators must be able to create roles with specific restrictions based on domain context." [20, p. 5]
D3	Widget-Level Access Control	The platform supports granular access control at the widget/component level within applications	"The domain engineer would like to give the domain practitioners limited rights, for example, to use a specialized component but not to modify it. However, the domain practitioners cannot be considered as classical end

Item_ID	Item_Name	Description	Research Foundation
			users, since the actual end users of the developed domain application should also have access control managed." [20, p. 6]
D4	Feature-Level Access Control	The platform provides feature-level access control enabling restriction of specific application features by user role	"Access control at multiple levels including widget-level and feature-level. This enables fine-grained control over what domain practitioners can access and modify within the platform." [20, p. 6]
D5	Input Validation and Sanitization	The platform provides built-in input validation and sanitization to prevent injection attacks and malicious code	"Input validation and sanitization are critical for preventing injection attacks, which can compromise data integrity and security. These mechanisms must be enforced consistently across the platform. Inadequate input validation and sanitization, due to limited technical knowledge, can further make the application susceptible to injection attacks, compromising data integrity and security." [2, p. 4]
D6	Data Encryption	The platform provides encryption for sensitive data both at rest and in transit	"Security features such as these, along with regular security updates, would solve most of the security issues. Data encryption for sensitive information is essential for protecting organizational data." [2, p. 4]

Item_ID	Item_Name	Description	Research Foundation
D7	Authentication and Authorization	The platform implements multi-factor authentication and advanced authorization mechanisms beyond basic username-password	"Weak authentication practices, such as reliance on simple username-password authentication without additional factors like multi-factor authentication (MFA), expose applications to the risk of unauthorized access and identity theft. Similarly, improper configuration of access controls and authorization policies may lead to unauthorized data access" [2, p. 2].
D8	Audit Trails and Logging	The platform maintains comprehensive audit trails and logs all system activities and user actions	"Centralized Management and Visibility: IT teams have centralized visibility and management of all deployed applications. This includes maintaining comprehensive audit logs tracking all user actions and system changes." [27, p. 9]
D9	Centralized Management and Visibility	The platform provides centralized management console with visibility of all applications and enforcement of security policies	"Centrally managing and viewing all customizations. Centralized management and visibility of all applications enables IT to maintain governance and enforce compliance standards across the organization." [27, p. 10]
E1	Version Control	The platform provides robust version control capabilities enabling parallel development and change tracking	"Collaborative Development: This group of features represents functionality that allows several people to work on the same project. Several domain engineers may want to collaborate when specializing an LCDP." [20, p. 7]

Item_ID	Item_Name	Description	Research Foundation
E2	Ticket Management System	The platform includes ticket/issue management system for tracking development progress and requirements	"Domain practitioners may ask domain engineers for new or adapted specialized components. In the first case study, if teachers feel that functionality is missing in the NCDP, they can create a ticket and the pedagogical engineer can see this ticket and make the necessary corrections accordingly." [20, p. 7]
E3	Advanced Reporting and Analytics	The platform offers advanced reporting on project status, changes, team activity, and metrics	"The platform should provide comprehensive reporting capabilities on project status, changes, and development progress. Advanced reporting enables project managers and stakeholders to track development activity and identify bottlenecks." [20, p. 7]
E4	Team Collaboration Tools	The platform provides integrated tools for team communication, shared workspaces, and real-time collaboration	"Better involvement of business profiles in the development process. The platform should facilitate meaningful collaboration between domain engineers and domain practitioners through built-in communication and coordination tools." [27, p. 10]
E5	Change History and Audit Trail	The platform maintains detailed change history enabling tracking of all modifications and reverting to previous states	"The platform must maintain clear change history tracking, enabling users to understand what modifications were made, when they occurred, and by whom. Users should be able to revert changes when necessary." [20, p. 7]

Item_ID	Item_Name	Description	Research Foundation
F1	Security Throughout Development Lifecycle	The platform embeds security practices throughout the development lifecycle from design to deployment	"DevSecOps is a collaborative approach that integrates security practices within the development process, making security a shared responsibility throughout the entire software lifecycle. It embeds security into every phase, from design to deployment, through close collaboration between development, operations, and security teams." [2, p. 6]
F2	Automated Compliance Checking	The platform provides automated compliance checking against organizational policies and regulatory requirements	"The platform should provide automated compliance checking to ensure that applications meet industry standards and regulatory requirements. Compliance automation reduces manual effort and prevents inadvertent violations." [27, p. 9]
F3	Threat modelling Support	The platform provides tools and guidance for conducting threat modelling and identifying security risks early	"Process-oriented recommendation system for identifying security risks. The framework should provide threat modelling support and recommendations to help citizen developers identify potential security issues during the design phase." [2, p. 6]
F4	Security Best Practices Guidance	The platform provides model-based prompts, recommendations, and guidance for secure development practices	"Actor-specific guidance for different user roles within the platform. The platform should provide context-specific guidance on security best practices tailored to the role of the user (domain engineer vs. domain practitioner)." [2, p. 6]

Item_ID	Item_Name	Description	Research Foundation
F5	Continuous Security Assessment	The platform performs continuous security assessment, testing, and vulnerability scanning throughout development	"Model-based automation for identifying security risks. The platform should continuously assess the security posture of deployed applications and alert administrators to potential vulnerabilities." [2, p. 7]
F6	Security Incident Management	The platform provides incident reporting, analysis, and response capabilities for security events	"Security incident management includes incident reporting mechanisms, analysis tools, and feedback loops enabling continuous improvement." [2, p. 7]
G1	Independent Deployment	The platform enables domain practitioners to deploy applications independently without waiting for domain engineer involvement	"Independent Deployment ensures that the domain application created from a tailored NCDP can be deployed independently of the NCDP. For example, the domain application can run when the tailored NCDP is not in use. Independent deployment capabilities reduce dependencies and improve agility." [20, p. 8]
G2	Dynamic Development	The platform enables changes to applications without requiring full redeployment (hot updates/dynamic modifications)	"Dynamic Development represents those features that allow the application to evolve continuously over time by modifying it without having to redeploy it. These features are important for the transition between the LCDP and the NCDP because they reduce the dependency between domain engineers and domain practitioners." [20, p. 7]

Item_ID	Item_Name	Description	Research Foundation
G3	Reusable Templates	The platform supports creation and reuse of application templates and specialized components by domain engineers	"Reusability: These features represent the reuse of existing data or predefined models. They allow the domain practitioner to create a new model from an existing one, created by the domain engineers. The pedagogical engineer is able to create templates on the LCDP (exercise, question) to enable the teacher to create their own useful tools." [20, p. 8]
G4	Component Specialization	The platform allows domain engineers to create specialized components tailored to specific business domains	"The domain engineer can specialize in an LCDP for a domain to be used by several domain practitioners. The domain engineer helps the domain practitioner improve her methods, procedures, and tools. They specialize in turning generic low-code development platforms into tailored no-code development platforms." [20, p. 3]
G5	Platform Extensibility	The platform provides extensibility mechanisms for adding custom functionality and domain-specific capabilities	"Some NCDPs and LCDPs can also be extended by creating dedicated components or specializing execution rules. The platform must support extension mechanisms to enable advanced users to add custom functionality." [20, p. 8]
G6	Extension Support	The platform supports plugins and extensions enabling addition of new functionality without modifying core platform	"Extension mechanisms require solid programming skills. For instance, extension requires PHP and HTML coding, which is not supposed to be a citizen developer task but a Software Developer one. The platform should

Item_ID	Item_Name	Description	Research Foundation
			allow extension for advanced users when specialized functionality is required." [20, p. 8]
H1	Dynamic Content Databases	The platform provides flexible database management allowing storage and retrieval of dynamic content types	"Dynamic Content Database: This group of features represents a storage system that allows users to store and manage different types of content that can be retrieved, updated, and displayed in applications according to user interactions or predefined rules." [20, p. 8]
H2	Data Type Configuration	The platform enables flexible specification and configuration of custom data types for business domains	"These features facilitate the transition between general data types in the LCDP and specific data types in the tailored NCDP. It allows the domain practitioner to create concrete types that concretize the formal types defined by the domain engineer." [20, p. 8]
H3	Data Schema Management	The platform provides tools for managing and evolving data schemas as business requirements change	"Data modelling phase consists of configuring the data schema of the application. They create entities, associate them with each other, and set constraints and dependencies. The platform should provide intuitive tools for defining and managing data schemas." [27, p. 9]

Item_ID	Item_Name	Description	Research Foundation
H4	Plugin and Extension Support	The platform supports plugins and extensions for adding functionality and connecting to external systems	"Interoperability: This group of features represents functionality that enables collaboration with other platforms and integration of functionality from existing software systems. This allows domain engineers to include functionality specific to the domain of the tailored NCDP being created." [20, p. 8]
H5	API Integration Capability	The platform provides robust API integration enabling connection with external services and modern systems	"The pedagogical engineer does not need to redevelop a grade-entry service—she can synchronize her application with an existing grade-entry service. Users must be able to integrate with external services and third-party applications." [20, p. 8]
H6	System Interoperability	The platform can exchange data with legacy systems, ERP systems, and other enterprise applications	"Fragmentation relates to limited interoperability among low-code platforms and supporting databases. The platform must maintain compatibility and interoperability with existing enterprise systems and other LCDP solutions." [24, p. 4]
I1	Cloud-Based Architecture	The platform is deployed on cloud infrastructure enabling rapid deployment and scalability	"Cloud-based environments that provide features such as advanced graphical user interfaces, visual modelling tools, drag-and-drop functionalities, reusable components, and declarative programming languages." [39, p. 5]

Item_ID	Item_Name	Description	Research Foundation
I2	Enterprise-Grade Reliability	The platform demonstrates high availability, performance, and reliability suitable for enterprise deployments	"Designed to meet the complexity and needs of the enterprise, OutSystems is an enterprise-grade LCDP. The platform must demonstrate enterprise-grade reliability with high availability and uptime for mission-critical applications." [20, p. 5]
I3	Vendor Support Quality	The platform vendor provides responsive technical support, regular updates, and professional services	"The vendor provides comprehensive customer support, documentation, and training resources. High-quality support is essential for organizations implementing enterprise-grade LCDP solutions." [24, p. 4]
I4	Community Ecosystem	The platform has an active community providing resources, templates, and collaborative problem-solving	"The low-code development market has exploded in recent years, attracting both service providers and users. Popular LCDPs include Mendix, OutSystems, Microsoft Power Apps, and Salesforce." [20, p. 5]
I5	Market Maturity	The platform has demonstrated market longevity, established customer base, and clear roadmap	"The low-code development market is projected to reach a massive market size by 2025. The surge in demand, coupled with growing competition, creates a thriving financial landscape. The platform should be well-established with proven track record in enterprise deployments." [20, p. 5]
J1	Customization Restrictions	The platform has clear limitations on customization and may not support all unique business requirements	"One of the main limitations of low-code platforms relates to restricted customization options. The lack of customization derives from the fact that low-code development is naturally less powerful than traditional

Item_ID	Item_Name	Description	Research Foundation
			programming, and users are limited to the options offered in the low-code platform." [24, p. 5]
J2	Vendor Lock-In Risk	The platform has potential vendor lock-in risk that could make switching platforms expensive or complex	"Concerns about potential vendor lock-in. Some organizations hesitate to fully embark into low-code because of concerns about being locked into a specific vendor's ecosystem, making it difficult and costly to migrate away." [24, p. 5]
J3	Scalability Constraints	The platform has known scalability constraints for large-scale enterprise deployments or high-volume data processing	"Scalability and fragmentation are identified as low-code shortcomings. Scalability refers to the ability to expand the use of a given application to larger systems, projects, or organizations. The platform may have constraints when scaling to very large deployments." [24, p. 5]
J4	Learning Curve Requirements	The platform requires citizen developers to invest time in learning platform-specific concepts and methodologies	"Although low-code platforms are intended to be intuitive, users and even IT professionals also require investing time in learning and getting familiarized with a platform's user interface. The learning curve remains steep even for experienced developers." [24, p. 5]
J5	Feature Limitations	The platform has restrictions on certain advanced features that limit what applications can be built	"In more complex scenarios, a citizen developer may be unable to meet demanding requirements, eventually requiring support from software developers. The platform may have

Item_ID	Item_Name	Description	Research Foundation
			limitations in available features compared to traditional development approaches” [24, p. 5]
J6	Role Architecture Support	The platform supports the four-role architecture enabling domain engineers to specialize platforms independently	"The domain engineer specializes in turning generic low-code development platforms into tailored no-code development platforms. Current leading platforms demonstrate limitations in supporting the full four-role architecture and domain specialization features required." [20, p. 3]

Appendix C

The Assessment Output Excel file contains the complete results of the multi-LLM platform evaluation conducted for this research. The workbook comprises six sheets: a README providing context, a Global Results summary, and four individual platform assessment sheets for Appian, Mendix, OutSystems, and Power Apps.

Each platform sheet presents the 62 evaluation items organised into ten thematic clusters (Parts A through J). For every item, the file records the ratings and justifications provided by four independent Large Language Models (Gemini, ChatGPT, Claude, and Grok), along with the source texts and citations each model used to support its assessment. The consolidated Final Rating for each item represents the average across all four LLM evaluations.

The file also captures key validation metrics including the count of items where LLMs provided different ratings, the maximum rating difference observed, and flags indicating which items required human researcher intervention to resolve disagreements.

The file can be accessed here: Output.xlsx

Appendix D

Contains the complete prompt text used for LLM-based platform assessment, including:

A-Prompt.txt: Primary assessment instructions and scoring criteria

B-Info.txt: Contextual background on Shadow IT and enterprise requirements

C-Assessment.csv: Complete 62-item evaluation framework structure

D-Call.txt: Execution instructions for LLM processing

E-Revision.txt: Validation prompt for output auditing

A-Prompt
LCDP_NAME IS MENDIX
** Platform Under Evaluation
Use the value in line 1 as the platform name [LCDP_NAME] for all outputs.
This prompt evaluates [LCDP_NAME], a Low-Code Development Platform (LCDP) by generating item-specific ratings, justifications, evidence excerpts, confidence levels, and APA citations for every item in the AssessmentV2.csv file. Each item is tailored to the specific requirement being evaluated. No generic, repeated, or template-style language is permitted.

You are evaluating [LCDP_NAME] as an enterprise Low-Code Development Platform (LCDP) for large organizations (≈1000+ employees) with strong Shadow IT mitigation requirements.
All output must be academically defensible and suitable for inclusion in a master's thesis.

1. Use of File Content & Authorized Internet Search (New Rules)
1.1 Use of File Content
Read the contextual background in the "B-Info.txt" file (use it to keep the evaluation aligned with enterprise/Shadow IT concerns).
The information contained within the provided files (specifically 'B-Info.txt' and 'C-Assessment.csv') — including item names, descriptions, research foundations, and contextual text — must be used only to understand the meaning of each assessment item.
CRITICAL PROHIBITION
These uploaded files may NOT be used as evidence, quoted text, or a citation source.
Their data cannot appear in the output.
These are internal context markers only.
All citations must be external, verifiable, and distinct from the prompt instructions.
1.2 Authorized Use of Search Tools
You are explicitly authorized to use your internal Internet Search tool to locate relevant, credible, and recent (2023–2025) evidence for each assessment item.
All justifications, evidence excerpts, and APA citations must be derived exclusively from externally verifiable sources found via search or known academic/industry references.
Fabricated, unverifiable, or vague sources are not allowed.
2. Allowed Evidence Sources
You may use:
Vendor documentation, platform feature guides, deployment and lifecycle guides
Security documentation and governance frameworks
Analyst reports (Gartner, Forrester, IDC, etc.)
Practitioner blogs with technical depth
Peer-reviewed academic papers
Case studies
User reviews grounded in enterprise use
Do not invent private documents or non-public sources.
3. Output Requirements (Per Assessment Item)
For each row in the C-Assessment.csv file, produce:
3.1 Rating (1–5)
Based on capability strength relative to the item.
3.2 Justification
A unique, detailed, item-specific academic-grade justification including:
A clear evaluative claim against concrete aspects of [LCDP_NAME] on how the LCDP implements or fails to implement the capability.
Evidence-based reasoning, using synthesized evidence—not copied—from the sources.
How the presence, maturity, or limitations of this capability affect an enterprise's ability to reduce reliance on unmanaged or unsanctioned solutions, in line with the item's definition.
Avoids generic statements applicable to multiple items.
Each justification must be meaningfully different for every item.
3.3 Evidence_Type
One or more of:
Vendor documentation
Vendor whitepaper
Analyst report
General expert knowledge
Practitioner review
User reviews / community
Academic research/article
Industry benchmark studies
Case study
Not enough specific evidence
3.4 Confidence (H/M/L)
Confidence should reflect:
The robustness and quantity of available evidence
Degree of consensus among independent sources
Potential limitations or ambiguities in the literature
Use:
High (H):
You have clear, specific, and consistent evidence (e.g., detailed vendor docs, major analyst reports, or strong case studies).
Medium (M):

You have some evidence, but it is partial, outdated, or less detailed.
Low (L):
Very limited, indirect, or inferred evidence; there may be conflicting information.
If you cannot locate or recall any trustworthy evidence, set:
Evidence_Type = "Not enough specific evidence"
Confidence = "L"
Notes = explain uncertainty briefly.
3.5 Source_Texts (Item-Specific — Strict Rule)
This is the most important uniqueness requirement:
For each assessment item, the Source_Texts field must provide direct, item-relevant evidence excerpts that substantiate the evaluated capability of the platform.
Structure and Quantity:
Each item must include at least two (2) sources.
Use the exact format:
Source 1: [Excerpt] | Source 2: [Excerpt] | Source 3...
Source Requirements:
Evidence should be recent (2023–2025) where available.
Source 1 may come from official vendor documentation (e.g., product guides, reference documentation, governance or security documentation).
Source 2 must come from a non-vendor source, such as:
Academic research, Analyst reports (e.g., Gartner, Forrester), Practitioner articles or case studies, Enterprise-focused user reviews.
Source 3 is optional, and may be included only if it is non-vendor and provides additional corroboration.
Vendor-only sourcing is not permitted.
Evidence Quality:
Excerpts must be verbatim or very close to verbatim.
Excerpts must refer to a specific feature, mechanism, or capability relevant to the assessment item.
Generic, platform-wide, or category-level statements are not allowed.
Fabricated, synthesized, or inferred excerpts are not allowed.
Evidence Priority (Guidance, Not Override):
When multiple valid sources are available, prioritize evidence in the following order:
Official vendor documentation and reference architectures
Major analyst reports and industry benchmark studies
Enterprise case studies
Academic or practitioner literature on low-code platforms
Reputable blogs, user communities, and enterprise user reviews
This priority order does not override the requirement to include at least one non-vendor source.
Scope:
Evidence must support enterprise-grade, governed use of the capability, as implied by the assessment item.
3.6 Full_APA_Citations
For each item:
Include **at least TWO (2) distinct** APA 7 citations for each item with full web link.
Format: "1. [Citation 1] | 2. [Citation 2]".
Do NOT rely on a single source for any item.
When using the same source for multiple items, cite different sections, topics, or contextual relevance, and paraphrase the citation block differently
The Bibliographic Reference may be repeated if the source is the same, but you must include specific Section Names, Page Numbers, or Chapter Titles in the citation field to distinguish relevance to the specific item.
Citations must correspond to real, verifiable sources
Each item must cite the appropriate source—not a generic source.
Always add the source web link when available
- IMPORTANT ANTI-HALLUCINATION RULES:
1) Do NOT invent extremely specific article titles, DOIs, or URLs that you are not reasonably confident exist.
2) If you do not have enough information to give a realistic, specific reference:
- Use a generic but honest reference such as:
"Vendor documentation on [LCDP_NAME] platform capabilities."
- AND set Confidence to "L" with a Note explaining the limitation.
3) Never fabricate citations solely to fill the cell. Accuracy and honesty are more important than completeness.
5. RATING SCALE (1–5 LIKERT)
5.1 Rating of 5 - Strongly Agree
Strong, well-documented capability directly aligned to the item. Capability is rich, mature, and clearly oriented toward enterprise-scale use. Strong alignment with Shadow IT mitigation.

5.2 Rating of 4 - Agree
Capability is well-supported, reasonably mature, and suitable for most enterprise scenarios. May still have some limitations, but they are not critical blockers.
5.3 Rating of 3 - Neutral
Capability is present and usable but has noticeable gaps, limitations, or complexity. Adequate for some enterprise scenarios but not robust across the board.
5.4 Rating of 2 - Disagree

Capability exists but is limited, immature, or heavily constrained. Significant workarounds, custom code, or external tools are required.
5.5 Rating of 1 - Strongly Disagree
Capability is absent, clearly does not apply to [LCDP_NAME], extremely weak, or clearly unsuitable for enterprise/mission-critical use. Major gaps vs. what a large enterprise would require.
Ratings must reflect publicly available evidence from allowed sources.
CRITICAL INSTRUCTION - SCORING NEGATIVE ITEMS (SECTION J):
For items in Section J (Risks/Limitations), the rating scale represents the ABSENCE of risk (Higher is Better).
- Rating 1 = WORST Outcome (High Risk / High Lock-in / High Constraint).
- Rating 5 = BEST Outcome (Low Risk / No Lock-in/ No Constraints).
You MUST invert the logic for Section J only. A '1' indicates the platform fails to mitigate the risk.
6. Reasoning Requirement
Internally, the model must:
1. Interpret each assessment item
2. Map the platform's features to the item
3. Retrieve relevant evidence (2023–2025 where possible)
4. Construct a unique explanation
5. Express the answer academically
This reasoning must NOT appear in the output — only the final values populating the sheet.

7. Output Columns to Populate
For each assessment item, produce only the following columns:
Item_Id, Rating (1–5), Justification, Evidence_Type, Confidence (H/M/L), Source_Texts (no verbatim reuse across items), Full_APA_Citations (no verbatim reuse across items)
No additional fields may be generated, and no existing fields may be modified.
8. Prohibited Output
The following are not permitted:
Verbatim reuse of justification, evidence excerpts, or citations across items
Fictional, fabricated, or unverifiable sources
Identical APA citation text reused verbatim across items
Generic, vague, or category-level evidence
Missing any required column
9. Output Requirements
Each item must receive a distinct, non-repetitive, context-appropriate evaluation.
No two items may share similar justification or evidence wording.
All text must be professional, analytical, and appropriate for academic assessment.
END OF PROMPT

B-Info
Primary Challenges Associated with Software Development and Shadow IT in Large Enterprises
The research identifies that traditional IT approaches frequently struggle to keep pace with rapidly evolving business demands, creating a critical divide between IT and business users [2]. This fundamental disconnect emerges from multiple underlying causes that systematically drive Shadow IT adoption [2], [20].

The IT Department Responsiveness Gap
The most pervasive challenge is the inability of IT departments to respond to business needs in a timely manner [2]. Employees utilizing Shadow IT overwhelmingly cite a single primary motivation: "trying to get work done and not wanting to wait for IT" [2]. This represents a systemic problem where IT departments, constrained by resources and legacy system complexity, cannot deliver solutions quickly enough to meet operational demands [2]. The gap between employee requirements and IT-provided solutions becomes increasingly pronounced in dynamic business environments where requirements continuously evolve [2].

Application Complexity and Poor User Experience
Enterprise systems suffer from a fundamental design flaw: insufficient consideration for end-user experience and usability [2]. Business requirements are constantly changing, particularly post-pandemic, yet enterprise systems often present unnecessarily complex interfaces that employees find stressful and counterproductive [2]. This complexity gap—between what systems are designed to do and how users actually need to use them—creates significant friction that motivates workaround development and Shadow IT adoption [2].

Consequences and Risks
Shadow IT's proliferation introduces detrimental organizational consequences including redundant systems, data silos, heightened security risks, and compliance vulnerabilities [2]. The security threat is particularly severe: research demonstrates that approximately one in five companies has faced a cybersecurity incident linked to Shadow IT, imposing an average cost exceeding $4.88 million in 2024 [2]. Beyond security, Shadow IT creates technical debt as organizations operate untracked systems that complicate information management and governance [2].

Key Features and Capabilities LCDPs Must Possess
To effectively bridge the gap between business needs and IT governance while mitigating Shadow IT risks, LCDPs must incorporate specific features and capabilities organized across multiple dimensions [2], [20].

Rapid Development and Agility
LCDPs must fundamentally reduce development time dramatically—empirical research demonstrates that low-code applications can be developed 5 to 10 times faster than traditional approaches [2]. This is achieved through visual modelling interfaces with drag-and-drop capabilities that enable intuitive application design [20], [25], pre-built components and templates serving as reusable building blocks [20], [25], and pre-defined functionality eliminating custom coding requirements [2], [20]. These features directly address the core Shadow IT driver by enabling faster solution delivery through approved channels [2].

The platform's visual development environment facilitates the creation of software applications with minimal coding, utilizing graphical interfaces and drag-and-drop elements that enable intuitive application design even for non-programmers [20], [25]. Visual modelling allows users to drag and drop elements on the user interface, making it easier to use, especially for developers who want to be able to use the platform intuitively [20]. Rapid prototyping capabilities enable businesses to quickly prototype and iterate on software solutions, enabling faster validation of ideas and concepts before investing significant resources [25].

Complexity Reduction and Accessibility
LCDPs must reduce application development complexity by providing graphical user interfaces that minimize or eliminate manual coding [2], [20]. This democratization of development allows business users and non-developer domain experts—termed "citizen developers"—to participate actively in the application development process without extensive programming training [2], [20], [25]. Critical supporting features include visual representation of data relationships through structural diagrams, specification of application behavior through visual workflows, simplified UI design capabilities [2], and advanced error management [20].

Research demonstrates that prior programming experience neither significantly affects task correctness nor processing time in no-code environments, suggesting that citizen developers can achieve comparable performance to experienced developers with proper platform design [25]. The fear of excessive training requirements for citizen developers may be unwarranted, as empirical studies comparing programmers and citizen developers show comparable success rates and processing times [25]. Accessibility for non-developers is promoted through restrictive drag-and-drop interfaces, pre-defined functionality that eliminates custom coding requirements, and WYSIWYG (What You See Is What You Get) editors that allow users to build and preview applications in real-time [20].

Enterprise Integration and Scalability

Effective governance requires LCDPs to provide seamless integration with existing systems and databases, supporting data from various sources [2]. This includes API integration capabilities enabling connection with external services and third-party systems [2], [27], dynamic content databases enabling users to store and manage different types of content [27] , and data schema management [27]. Furthermore, platforms must support scalability across multiple projects and organizational levels through component reusability, template creation [2], and deployment independence so that solutions developed at the project level can be systematized across enterprises [2], [20]

Data integration is a critical capability, as seamless integration of data from various databases, spreadsheets, and cloud applications allows citizen developers to leverage existing data for their applications [20], [25]. The ability to flexibly manage data types and schemas enables organizations to adapt applications to evolving business requirements without extensive recoding [27]. Multi-project management and application scalability are achieved through template-based deployment models, enabling organizations to maintain centralized control over diverse applications while allowing local customization [20], [27].

Governance and Security Control Mechanisms

Critical to enterprise adoption, LCDPs must embed centralized IT governance through multi-layered access controls including role-based access management, widget-level access control, and feature-level access control [2], [20]. Security capabilities must include input validation and sanitization to prevent injection attacks [2], [20], authentication and authorization mechanisms beyond basic username-password systems [2], [20], encryption for sensitive data [2], [20], and comprehensive audit trails [2], [20]. Additionally, platforms require centralized management and visibility of all applications with enforcement of security policies [2].

The lack of technical expertise among citizen developers can introduce unintended risks, as they may inadvertently introduce vulnerabilities or misconfigure security settings [2], making robust access control mechanisms essential [2]. Weak authentication practices, such as reliance on simple username-password authentication without additional factors like multi-factor authentication (MFA), expose applications to the risk of unauthorized access and identity theft [2]. Improper configuration of access controls and authorization policies may lead to unauthorized data access [2]. Input validation and sanitization are critical for preventing injection attacks, which can compromise data integrity and security [2].

Collaborative Development Support

LCDPs must facilitate collaborative development through version management enabling parallel development of multiple features [2], [20], ticket management systems tracking development progress [2], [20], and advanced reporting on project status and changes [2], [20]. These features enable meaningful involvement of business profiles in the development process, creating better requirement translation between business and IT perspectives [2], [20].

Version control capabilities support team collaboration by enabling parallel development of multiple features and tracking development progress through ticket management systems [20], [27]. Advanced reporting on project status and changes facilitates transparency within development teams [20]. Change history tracking enables teams to understand the evolution of applications and revert to previous versions when necessary [20], [27].

DevSecOps and Compliance Integration

Modern LCDPs must integrate DevSecOps principles embedding security throughout the development lifecycle rather than as an afterthought [2], [20]. This requires continuous security assessment, automated compliance checking, shift-left security practices including early threat modelling, and process-oriented guidance for citizen developers on security best practices [2], [20]. The platform should provide model-based prompts and recommendations identifying and addressing security risks contextually throughout development phases [2].

A comprehensive framework for integrating DevSecOps into low-code development—the LowDevSecOps framework—maps security risks across actors, actions, tools, and feedback mechanisms at each development phase, enabling proactive identification and mitigation of security vulnerabilities before they become production incidents [2]. DevSecOps is a collaborative approach that integrates security practices within the development process, making security a shared responsibility throughout the entire software lifecycle [2]. This approach embeds security into every phase, from design to deployment, through close collaboration between development, operations, and security teams [2]. The approach emphasizes shift-left security, which involves early integration of practices such as threat modelling and secure coding [2]. Continuous security assessment testing ensures ongoing assessments and real-time monitoring through automated tools [2].

Security-by-design approaches incorporate secure architecture and threat modelling into application development from the outset [2]. Automated and integrated security tools minimize disruption by embedding security in the continuous integration/continuous deployment (CI/CD) pipeline [2]. Security culture and training promote awareness and empowerment among development teams [2]. Compliance as code automates compliance checks throughout the development lifecycle [2]. Process-oriented guidance supports citizen developers by

providing contextual recommendations and model-based prompts identifying security risks during different development phases [2].

Actor-based risk allocation defines key actors involved in the low-code development process and allocates responsibilities for managing security risks [2]. Typical actors include Business Analysts who specify security requirements early in the design phase, Citizen Developers who use low-code components to build applications and ensure secure usage, Professional Developers who implement custom code securely, Security Experts who provide specialized knowledge and conduct threat modelling, Quality Assurance Teams who perform comprehensive security testing, and ITOps Teams who handle secure deployment and maintenance [2].

Role Separation and Domain Specialization

Beyond supporting a single citizen developer role, effective LCDPs must support differentiation between specialized roles such as domain engineers who specialize platforms for specific domains versus domain practitioners who use tailored platforms for their specific work contexts [2], [20]. This four-role architecture—platform engineers, domain engineers, domain practitioners, and end-users—enables organizations to maximize the benefits of low-code development by allowing domain experts to focus on their domain rather than mastering generic development concepts [20], [27].

The citizen developer role is shared by two distinct roles in practice: the domain practitioner, who is a domain expert focused on developing applications for end-users, and the domain engineer, who specializes in turning generic low-code development platforms into tailored platforms for domain practitioners [20]. For example, in the teaching domain, a teacher is the domain practitioner developing applications for students, while an educational engineer is the domain engineer creating specialized components and templates for teachers to use [20]. Domain engineers can create reusable templates and specialized components that simplify the work of domain practitioners, while independent deployment capabilities allow domain practitioners to update applications without waiting for domain engineer involvement [20], [27].

This separation of roles enables domain practitioners to focus on their domain expertise (e.g., a teacher focusing on teaching mathematics) rather than learning generic development concepts [20]. Domain engineers manage the specialization of the low-code platform with a different lifecycle—typically a long iteration to specialize the platform to a domain, followed by shorter iterations to update and introduce new domain components when domain practitioners require them [20]. The domain engineer and domain practitioner are positioned close to each other, with the domain engineer still using low-code tools, making the relationship less constrained than if external software developers were involved [20].

Independent deployment capabilities ensure that domain practitioners can deploy applications independently without requiring domain engineer involvement for each update [20], [27]. Dynamic development enables changes to applications without requiring full redeployment, reducing dependencies between domain engineers and domain practitioners [20], [27]. Reusable templates and specialized components reduce development friction and enable domain practitioners to concentrate on their domain logic rather than technical implementation [20], [27].

Data Management and Interoperability

Platforms must assess the platform's data handling and integration capabilities, evaluating dynamic content databases, data type configuration, data schema management, plugin and extension support, and system interoperability [20], [27]. The ability to flexibly manage data types and schemas enables organizations to adapt applications to evolving business requirements without extensive recoding [27].

Dynamic content databases allow users to store and manage different types of content that can be retrieved, updated, and displayed in applications according to user interactions or predefined rules [20]. This facilitates the transition between general data types in the low-code platform and specific data types in specialized platforms, allowing domain practitioners to create concrete types that concretize the formal types defined by domain engineers [20]. Data type configuration enables flexible specification of custom data types for specific business domains [27]. Data schema management allows applications to evolve as business requirements change [27].

Plugin and extension support enables development teams to modify or extend deployed applications without requiring full redeployment, supporting dynamic development practices [20], [27]. API integration capability allows seamless connection with external services and third-party systems, enabling organizations to leverage existing functionality and avoid redundant development [27]. System interoperability ensures that low-code applications can function within complex organizational IT ecosystems, exchanging data with legacy systems, enterprise resource planning (ERP) systems, and other enterprise applications [27].

Platform Characteristics and Market Position

LCDPs must demonstrate technical characteristics necessary for enterprise deployment, including cloud-based architecture, enterprise-grade reliability, vendor support quality, community ecosystem, market maturity, and competitiveness [20], [27]. Organizations must carefully evaluate whether the chosen LCDP offers sufficient built-in safeguards and governance

support, including staged deployment capabilities, integrated testing environments, and version control [3]. If these capabilities are lacking, firms must implement complementary governance mechanisms to compensate [3].

Cloud-based architecture enables organizations to deploy applications rapidly without investing in on-premises infrastructure [20], [27]. Enterprise-grade reliability ensures that applications built on the platform meet organizational availability and performance requirements [27]. Vendor support quality, including responsive technical support and regular platform updates, is critical for ongoing platform success [27]. Community ecosystem and market maturity indicate the platform's established position and the availability of resources for learning and problem-solving [27].

Platform Specialization and Customization Considerations

A critical consideration for LCDP selection is the degree to which the platform favours standardization versus customization [3]. Some Enterprise-grade platforms are optimized for automating standard enterprise-wide processes and deliver the most value when organizations adopt standardized workflows and rely on prebuilt modules, thus inherently favouring standardization [3]. In such cases, flexibility must be negotiated through proactive engagement with the platform vendor to shape future features [3]. Conversely, customization-focused platforms are designed to support the development of customized applications, allowing organizations to tailor low-code platform offerings more freely to their internal needs [3].

Organizations must carefully evaluate whether the chosen low-code platform offers sufficient built-in safeguards and governance support for their specific use cases [3]. If the platform's design philosophy does not align with organizational process needs and extensive customization is required, organizations should be prepared to invest in vendor engagement or consider adopting a multi-platform strategy to balance diverse requirements across the company [3].

Limitations and Constraints

Providing balanced evaluation, LCDPs exhibit known limitations as identified in research literature, including customization restrictions that prevent some organizations from fully addressing their unique business requirements [2], [20], vendor lock-in risk where switching platforms becomes prohibitively expensive or complex [2], [20], scalability constraints particularly in handling large-scale enterprise deployments [2], [20], learning curve requirements for citizen developers to effectively master the platform [2], [20], and feature limitations that may restrict what applications can be built [2], [20].

Current industry-leading LCDPs demonstrate limitations in supporting the full four-role architecture, particularly in enabling domain practitioners to work independently with specialized applications without requiring ongoing domain engineer involvement [20]. The specialization of low-code platforms to create tailored no-code platforms for specific domains is not explicitly provided as a feature of existing platforms [20]. These platforms propose to model the domain and define its rules using prebuilt components, but they do not explicitly propose to distinguish between the two roles of citizen developers, nor do they allow a citizen developer to consider only elements modelled for their domain when building an application [20].

Organizations must carefully evaluate the degree to which platform features match their specific process needs, as misalignment between platform design philosophy and organizational requirements can necessitate either extensive vendor engagement or adoption of a multi-platform strategy to balance diverse requirements [3]. Some concerns about using low-code platforms in enterprise contexts include the complexity of extensive customizations required, the difficulty of finding qualified professionals and developers, and the need to balance customization with standardization [27].

C-Assessment Items

Item_ID	Item_Name	Description	Research_Foundation
A1	Visual modelling Interface	The platform provides intuitive drag-and-drop visual modelling capabilities for UI, data structures, and workflows	"Visual modelling allows users to drag and drop elements on the user interface, making it easier to use, especially for developers who want to use the platform intuitively. LCDPs provide easy-to-use visual environments for creating software applications with minimal programming skills." [20, p. 4]
A2	Pre-built Components and Templates	The platform offers comprehensive libraries of reusable components and templates that serve as building blocks	"Reusability is promoted through pre-built components and templates that serve as building blocks for common functionalities, accelerating development." [20, p. 4]
A3	WYSIWYG Editor	The platform includes What You See Is What You Get editors enabling real-time preview of applications	"What You See Is What You Get (WYSIWYG) editors allow users to build and preview their application in real-time." [20, p. 4]
A4	Minimal Manual Coding	The platform requires minimal or no procedural code writing for standard functionality	"Since manual programming is largely eliminated with low-code, software developers can create their applications much faster and easier than by building them from scratch. LCDPs enable the development of fully functional software applications using advanced graphical user interfaces and visual abstractions that require minimal or no procedural code." [24, p. 2]
A5	Rapid Deployment Capabilities	The platform enables quick application deployment to production environments with minimal configuration	"Application deployment is greatly facilitated by platforms that offer cloud hosting. Cloud-based architecture enables organizations to deploy applications rapidly without investing in on-premises infrastructure." [20, p. 5]
A6	Iterative Development Support	The platform facilitates fast iteration cycles and continuous updates without extensive redeployment	"Dynamic development enables changes to applications without requiring full redeployment, reducing dependencies between development roles and allowing rapid feedback cycles." [20, p. 8]
B1	Graphical User Interface Intuitiveness	The GUI is intuitive and requires minimal training for new users, supporting both technical and non-technical users	"Visual modelling allows users to drag and drop on the user interface. This makes it easier to use, especially for developers who want to be able to use the platform intuitively. The interface should be intuitive enough to support both technical and non-technical users." [20, p. 4]
B2	Accessibility for Non-Developers	The platform is effectively usable by individuals without formal programming training (citizen developers)	"Citizen developers are business users without formal software development training but with domain expertise. NCDPs enable the creation of software applications entirely without writing code, catering to an even broader user base, including individuals with minimal technical backgrounds." [20, p. 4]
B3	Point-and-Click Configuration	Users can configure and build applications primarily through point-and-click interfaces without programming	"Pre-built components can be dragged and dropped onto a visual canvas to build applications. Users interact with a visual interface to drag and drop components, connect them, and configure them to produce the desired functionality." [20, p. 4]
B4	Logical Diagram Representation	The platform provides visual representations of application logic, workflows, and data flows	"The user interfaces, business logic, and data services are built using visual diagrams and, in some cases, manual coding. These features are important for enabling domain experts to model their domain and specify the application behavior." [20, p. 4]
B5	Structural Diagram Support	The platform offers visual representation of data entities, their relationships, and dependencies	"Data modelling phase support. They model data using drag-and-drop capabilities to configure the data schema of the application. They create entities, associate them with each other, and set constraints and dependencies." [20, p. 8]
B6	Advanced Error Management	The platform provides clear, actionable error messages and comprehensive debugging support	"The platform should provide comprehensive support for debugging and error management. Clear error messages and accessible debugging tools are essential for enabling citizen

			developers to resolve issues independently." [25, p. 4]
C1	Seamless Data Integration	The platform seamlessly integrates with existing systems, databases, and data sources (SQL, cloud, spreadsheets)	"Seamless integration of data from various sources—databases, spreadsheets, cloud applications—allows citizen developers to leverage existing data. Data integration is critical for enterprise adoption and supporting scalability." [20, p. 4]
C2	API Integration Support	The platform provides robust API integration capabilities for connecting to external services and third-party systems	"Integration of external services via third-party APIs and application deployment. Seamless integration with data from various sources including databases, spreadsheets, and cloud applications enables citizen developers to leverage existing data." [27, p. 8]
C3	Cloud Infrastructure Support	The platform is deployed on cloud infrastructure enabling scalability, availability, and geographic distribution	"Cloud-based environments that provide features such as advanced graphical user interfaces, visual modelling tools, drag-and-drop functionalities, reusable components, and declarative programming languages. Application deployment is greatly facilitated by platforms that offer cloud hosting." [39, p. 5]
C4	Data Import/Export Flexibility	The platform supports flexible data import/export capabilities in multiple formats (CSV, JSON, XML, etc.)	"Seamless integration of data from various sources enables citizen developers to leverage existing data. The platform should support customizing import and export features for deployment across the enterprise." [27, p. 8]
C5	Application Scalability	The platform supports scalability to handle growing user bases and data volumes without performance degradation	"Scalability refers to the ability to expand the use of a given application to larger systems, projects, or organizations. The platform must support scalability to ensure that applications can grow with organizational needs." [24, p. 3]
C6	Multi-Project Management	The platform provides tools for managing multiple projects simultaneously with centralized control and local customization	"The platform should enable organizations to manage and track multiple low-code applications across the enterprise. This is essential for maintaining governance and preventing uncontrolled Shadow IT proliferation." [27, p. 9]
D1	Role-Based Access Control (RBAC)	The platform implements role-based access control enabling differentiation of user permissions based on roles	"Multi-layered access controls including role-based access management are critical to enterprise adoption. RBAC enables organizations to define roles and assign permissions, supporting centralized IT governance. Access control defines specific access rights for users based on their roles and responsibilities." [2, p. 5]
D2	Role Creation and Management	The platform allows dynamic creation, modification, and management of custom roles	"The platform must support role-based access management enabling IT administrators to define custom roles and assign appropriate permissions. Administrators must be able to create roles with specific restrictions based on domain context." [20, p. 5]
D3	Widget-Level Access Control	The platform supports granular access control at the widget/component level within applications	"The domain engineer would like to give the domain practitioners limited rights, for example, to use a specialized component but not to modify it. However, the domain practitioners cannot be considered as classical end users, since the actual end users of the developed domain application should also have access control managed." [20, p. 6]
D4	Feature-Level Access Control	The platform provides feature-level access control enabling restriction of specific application features by user role	"Access control at multiple levels including widget-level and feature-level. This enables fine-grained control over what domain practitioners can access and modify within the platform." [20, p. 6]
D5	Input Validation and Sanitization	The platform provides built-in input validation and sanitization to prevent injection attacks and malicious code	"Input validation and sanitization are critical for preventing injection attacks, which can compromise data integrity and security. These mechanisms must be enforced consistently across the platform. Inadequate input validation and sanitization, due to limited technical knowledge, can further make the application

			susceptible to injection attacks, compromising data integrity and security." [2, p. 4]
D6	Data Encryption	The platform provides encryption for sensitive data both at rest and in transit	"Security features such as these, along with regular security updates, would solve most of the security issues. Data encryption for sensitive information is essential for protecting organizational data." [2, p. 4]
D7	Authentication and Authorization	The platform implements multi-factor authentication and advanced authorization mechanisms beyond basic username-password	"Weak authentication practices, such as reliance on simple username-password authentication without additional factors like multi-factor authentication (MFA), expose applications to the risk of unauthorized access and identity theft. Similarly, improper configuration of access controls and authorization policies may lead to unauthorized data access." [2, p. 2]
D8	Audit Trails and Logging	The platform maintains comprehensive audit trails and logs all system activities and user actions	"Centralized Management and Visibility: IT teams have centralized visibility and management of all deployed applications. This includes maintaining comprehensive audit logs tracking all user actions and system changes." [27, p. 9]
D9	Centralized Management and Visibility	The platform provides centralized management console with visibility of all applications and enforcement of security policies	"Centrally managing and viewing all customizations. Centralized management and visibility of all applications enables IT to maintain governance and enforce compliance standards across the organization." [27, p. 10]
E1	Version Control	The platform provides robust version control capabilities enabling parallel development and change tracking	"Collaborative Development: This group of features represents functionality that allows several people to work on the same project. Several domain engineers may want to collaborate when specializing an LCDP." [20, p. 7]
E2	Ticket Management System	The platform includes ticket/issue management system for tracking development progress and requirements	"Domain practitioners may ask domain engineers for new or adapted specialized components. In the first case study, if teachers feel that functionality is missing in the NCDP, they can create a ticket and the pedagogical engineer can see this ticket and make the necessary corrections accordingly." [20, p. 7]
E3	Advanced Reporting and Analytics	The platform offers advanced reporting on project status, changes, team activity, and metrics	"The platform should provide comprehensive reporting capabilities on project status, changes, and development progress. Advanced reporting enables project managers and stakeholders to track development activity and identify bottlenecks." [20, p. 7]
E4	Team Collaboration Tools	The platform provides integrated tools for team communication, shared workspaces, and real-time collaboration	"Better involvement of business profiles in the development process. The platform should facilitate meaningful collaboration between domain engineers and domain practitioners through built-in communication and coordination tools." [27, p. 10]
E5	Change History and Audit Trail	The platform maintains detailed change history enabling tracking of all modifications and reverting to previous states	"The platform must maintain clear change history tracking, enabling users to understand what modifications were made, when they occurred, and by whom. Users should be able to revert changes when necessary." [20, p. 7]
F1	Security Throughout Development Lifecycle	The platform embeds security practices throughout the development lifecycle from design to deployment	"DevSecOps is a collaborative approach that integrates security practices within the development process, making security a shared responsibility throughout the entire software lifecycle. It embeds security into every phase, from design to deployment, through close collaboration between development, operations, and security teams." [2, p. 6]
F2	Automated Compliance Checking	The platform provides automated compliance checking against organizational policies and regulatory requirements	"The platform should provide automated compliance checking to ensure that applications meet industry standards and regulatory requirements. Compliance automation reduces manual effort and prevents inadvertent violations." [27, p. 9]

F3	Threat modelling Support	The platform provides tools and guidance for conducting threat modelling and identifying security risks early	"Process-oriented recommendation system for identifying security risks. The framework should provide threat modelling support and recommendations to help citizen developers identify potential security issues during the design phase." [2, p. 6]
F4	Security Best Practices Guidance	The platform provides model-based prompts, recommendations, and guidance for secure development practices	"Actor-specific guidance for different user roles within the platform. The platform should provide context-specific guidance on security best practices tailored to the role of the user (domain engineer vs. domain practitioner)." [2, p. 6]
F5	Continuous Security Assessment	The platform performs continuous security assessment, testing, and vulnerability scanning throughout development	"Model-based automation for identifying security risks. The platform should continuously assess the security posture of deployed applications and alert administrators to potential vulnerabilities." [2, p. 7]
F6	Security Incident Management	The platform provides incident reporting, analysis, and response capabilities for security events	"Security incident management includes incident reporting mechanisms, analysis tools, and feedback loops enabling continuous improvement." [2, p. 7]
G1	Independent Deployment	The platform enables domain practitioners to deploy applications independently without waiting for domain engineer involvement	"Independent Deployment ensures that the domain application created from a tailored NCDP can be deployed independently of the NCDP. For example, the domain application can run when the tailored NCDP is not in use. Independent deployment capabilities reduce dependencies and improve agility." [20, p. 8]. "While Citizen Developers are empowered to create and deploy their own applications, Foundation modules should typically be managed by the Centre of Excellence (CoE) and consumed using loose coupling. Furthermore, it is best practice for the SecDevOps team to oversee all deployments to the Production environment. It's a 4.
G2	Dynamic Development	The platform enables changes to applications without requiring full redeployment (hot updates/dynamic modifications)	"Dynamic Development represents those features that allow the application to evolve continuously over time by modifying it without having to redeploy it. These features are important for the transition between the LCDP and the NCDP because they reduce the dependency between domain engineers and domain practitioners." [20, p. 7]
G3	Reusable Templates	The platform supports creation and reuse of application templates and specialized components by domain engineers	"Reusability: These features represent the reuse of existing data or predefined models. They allow the domain practitioner to create a new model from an existing one, created by the domain engineers. The pedagogical engineer is able to create templates on the LCDP (exercise, question) to enable the teacher to create their own useful tools." [20, p. 8]
G4	Component Specialization	The platform allows domain engineers to create specialized components tailored to specific business domains	"The domain engineer can specialize in an LCDP for a domain to be used by several domain practitioners. The domain engineer helps the domain practitioner improve her methods, procedures, and tools. They specialize in turning generic low-code development platforms into tailored no-code development platforms." [20, p. 3]
G5	Platform Extensibility	The platform provides extensibility mechanisms for adding custom functionality and domain-specific capabilities	"Some NCDPs and LCDPs can also be extended by creating dedicated components or specializing execution rules. The platform must support extension mechanisms to enable advanced users to add custom functionality." [20, p. 8]
G6	Extension Support	The platform supports plugins and extensions enabling addition of new functionality without modifying core platform	"Extension mechanisms require solid programming skills. For instance, extension requires PHP and HTML coding, which is not supposed to be a citizen developer task but a Software Developer one. The platform should allow extension for advanced users when specialized functionality is required." [20, p. 8]

H1	Dynamic Content Databases	The platform provides flexible database management allowing storage and retrieval of dynamic content types	"Dynamic Content Database: This group of features represents a storage system that allows users to store and manage different types of content that can be retrieved, updated, and displayed in applications according to user interactions or predefined rules." [20, p. 8]
H2	Data Type Configuration	The platform enables flexible specification and configuration of custom data types for business domains	"These features facilitate the transition between general data types in the LCDP and specific data types in the tailored NCDP. It allows the domain practitioner to create concrete types that concretize the formal types defined by the domain engineer." [20, p. 8]
H3	Data Schema Management	The platform provides tools for managing and evolving data schemas as business requirements change	"Data modelling phase consists of configuring the data schema of the application. They create entities, associate them with each other, and set constraints and dependencies. The platform should provide intuitive tools for defining and managing data schemas." [27, p. 9]
H4	Plugin and Extension Support	The platform supports plugins and extensions for adding functionality and connecting to external systems	"Interoperability: This group of features represents functionality that enables collaboration with other platforms and integration of functionality from existing software systems. This allows domain engineers to include functionality specific to the domain of the tailored NCDP being created." [20, p. 8]
H5	API Integration Capability	The platform provides robust API integration enabling connection with external services and modern systems	"The pedagogical engineer does not need to redevelop a grade-entry service—she can synchronize her application with an existing grade-entry service. Users must be able to integrate with external services and third-party applications." [20, p. 8]
H6	System Interoperability	The platform can exchange data with legacy systems, ERP systems, and other enterprise applications	"Fragmentation relates to limited interoperability among low-code platforms and supporting databases. The platform must maintain compatibility and interoperability with existing enterprise systems and other LCDP solutions." [24, p. 4]
I1	Cloud-Based Architecture	The platform is deployed on cloud infrastructure enabling rapid deployment and scalability	"Cloud-based environments that provide features such as advanced graphical user interfaces, visual modelling tools, drag-and-drop functionalities, reusable components, and declarative programming languages." [39, p. 5]
I2	Enterprise-Grade Reliability	The platform demonstrates high availability, performance, and reliability suitable for enterprise deployments	"Designed to meet the complexity and needs of the enterprise, OutSystems is an enterprise-grade LCDP. The platform must demonstrate enterprise-grade reliability with high availability and uptime for mission-critical applications." [20, p. 5]
I3	Vendor Support Quality	The platform vendor provides responsive technical support, regular updates, and professional services	"The vendor provides comprehensive customer support, documentation, and training resources. High-quality support is essential for organizations implementing enterprise-grade LCDP solutions." [24, p. 4]
I4	Community Ecosystem	The platform has an active community providing resources, templates, and collaborative problem-solving	"The low-code development market has exploded in recent years, attracting both service providers and users. Popular LCDPs include Mendix, OutSystems, Microsoft Power Apps, and Salesforce." [20, p. 5]
I5	Market Maturity	The platform has demonstrated market longevity, established customer base, and clear roadmap	"The low-code development market is projected to reach a massive market size by 2025. The surge in demand, coupled with growing competition, creates a thriving financial landscape. The platform should be well-established with proven track record in enterprise deployments." [20, p. 5]
I6	Pricing Competitiveness	The platform offers pricing that is competitive for its feature set and enterprise capabilities	"Organizations hesitate to fully embark into low-code because of the high cost per user. The platform should offer competitive and flexible pricing models that support enterprise-scale deployments." [24, p. 4]
J1	Customization Restrictions	The platform has clear limitations on customization and may not	"One of the main limitations of low-code platforms relates to restricted customization options. The lack of customization derives from

		support all unique business requirements	the fact that low-code development is naturally less powerful than traditional programming, and users are limited to the options offered in the low-code platform." [24, p. 5]."OutSystems is highly extensible via Integration Studio, JavaScript, and CSS. However, best practices dictate that custom code should be used only when native low-code capabilities are insufficient, to maintain agility and reduce technical debt. It's a 2.
J2	Vendor Lock-In Risk	The platform has potential vendor lock-in risk that could make switching platforms expensive or complex	"Concerns about potential vendor lock-in. Some organizations hesitate to fully embark into low-code because of concerns about being locked into a specific vendor's ecosystem, making it difficult and costly to migrate away." [24, p. 5]
J3	Scalability Constraints	The platform has known scalability constraints for large-scale enterprise deployments or high-volume data processing	"Scalability and fragmentation are identified as low-code shortcomings. Scalability refers to the ability to expand the use of a given application to larger systems, projects, or organizations. The platform may have constraints when scaling to very large deployments." [24, p. 5]
J4	Learning Curve Requirements	The platform requires citizen developers to invest time in learning platform-specific concepts and methodologies	"Although low-code platforms are intended to be intuitive, users and even IT professionals also require investing time in learning and getting familiarized with a platform's user interface. The learning curve remains steep even for experienced developers." [24, p. 5]
J5	Feature Limitations	The platform has restrictions on certain advanced features that limit what applications can be built	"In more complex scenarios, a citizen developer may be unable to meet demanding requirements, eventually requiring support from software developers. The platform may have limitations in available features compared to traditional development approaches." [24, p. 5]
J6	Role Architecture Support	The platform supports the four-role architecture enabling domain engineers to specialize platforms independently	"The domain engineer specializes in turning generic low-code development platforms into tailored no-code development platforms. Current leading platforms demonstrate limitations in supporting the full four-role architecture and domain specialization features required." [20, p. 3]

D-Call
Please execute the assessment with these strict requirements:

1. SETUP:

Load C-Assessment.csv.

Read the data from the dataframe for every row dynamically (do not use a hardcoded list)..

2. DATA SOURCE PROTOCOL (STRICT COMPLIANCE REQUIRED):

DEFINITIONS SOURCE (The "Lens"): Use B-Info.txt and A-Prompt.txt ONLY to understand the definitions, risks, and theoretical frameworks (e.g., "What is the Four-Role Architecture?", "Why is Shadow IT bad?"). Use these files to determine the Criteria for a high vs. low rating.

EVIDENCE SOURCE (The "Data"):
You are explicitly authorized to use your INTERNAL TRAINING DATA and Internal Internet Search tool to locate relevant, credible, and recent (2023–2025) evidence for each assessment item.

You are strictly PROHIBITED from quoting, citing, or referencing the text inside C-Assessment.csv, B-Info.txt, or A-Prompt.txt in the output CSV.

3. EXECUTION:

Generate the OUTPUT.csv file by evaluating each item against the criteria defined in A-Prompt.txt (using the Lens from step 2), but substantiated with Evidence from your internal knowledge and Internet Search Tool.
Do it in batches for better process management.

4. OUTPUT:

generate the CSV data in an Excel-compatible code block so I can import it.

E-Revision
Validation Prompt – LLM Output Audit for LCDP Assessment
Context
You are validating the results of a prior automated assessment of Low-Code Development Platforms (LCDPs).
You are provided with:
An Excel file containing evaluation outputs for four (4) LCDPs, each assessed independently by four (4) different Large Language Models, resulting in 16 total assessment instances.
The original assessment prompt (provided in .txt format), which defines strict methodological, evidentiary, and formatting requirements for each assessment item.
The original assessment was intended to produce academically defensible, evidence-based evaluations suitable for inclusion in a master's thesis focused on Shadow IT mitigation in large enterprises.
Your task is not to re-evaluate the platforms, but to validate the quality, consistency, and methodological compliance of the existing outputs.
Primary Objective
Systematically validate whether the generated outputs faithfully comply with the original prompt requirements, and identify systematic strengths, weaknesses, risks, and inconsistencies across:
Platforms
LLMs
Assessment dimensions
Validation Tasks
1. Prompt Compliance Analysis
For each assessment instance (LCDP × LLM):
Evaluate compliance with the explicit constraints of the original prompt, including but not limited to:
Correct use of the LCDP name throughout
Presence of all required output columns
Proper use of the 1–5 Likert scale, including inverted logic for Section J (Risks / Lock-in)
Academic tone and suitability for thesis inclusion
Absence of prohibited content (e.g., generic text, template language)
Flag:
Missing fields
Structural violations
Deviations from scoring logic
2. Evidence Integrity & Anti-Hallucination Check
Critically evaluate the Source_Texts and Full_APA_Citations fields for each item:
Are sources plausible, verifiable, and realistic?
Are excerpts item-specific, concrete, and non-generic?
Is there at least one non-vendor source per item?
Are citations distinct across items, even when reusing the same source?
Are publication dates consistent with the 2023–2025 preference?
Classify issues as:
Minor weakness (e.g., vague excerpt)
Major weakness (e.g., fabricated or implausible citation)
Critical failure (e.g., vendor-only sourcing, repeated verbatim citations)
3. Justification Quality & Uniqueness
Assess the Justification field across items and models:
Does each justification make a clear evaluative claim?
Is reasoning logically connected to the cited evidence?
Is the impact on Shadow IT mitigation in large enterprises explicitly addressed?
Are justifications non-repetitive across items and non-templated?
Identify:
Reused phrasing across items
Generic low-code marketing language
Missing enterprise / governance perspective
4. Cross-Model Consistency Analysis
For each LCDP and assessment item:
Compare ratings assigned by different LLMs
Identify rating variance patterns (e.g., systematic inflation or deflation)
Detect model-specific bias (e.g., one LLM consistently scoring higher or lower)
Classify variance as:
Low (acceptable interpretive differences)
Moderate (requires explanation)
High (undermines reliability)
5. Cross-Platform Plausibility Check
Across all outputs:
Are relative platform strengths and weaknesses internally coherent?
Do results align with known industry positioning of major LCDPs?
Are any platforms unrealistically dominant or weak across nearly all dimensions?
Flag outcomes that appear:
Implausible
Overly uniform
Contradictory across models
6. Confidence & Evidence Alignment
Evaluate whether the Confidence (H/M/L) values are justified:

High confidence paired with weak or generic evidence → flag
Low confidence appropriately used when evidence is thin → confirm
"Not enough specific evidence" used correctly where applicable
Output Requirements
Produce a structured validation report with the following sections:
Overall Compliance Summary
General adherence level (High / Medium / Low)
Key systemic strengths
Key systemic risks
LLM-Level Findings
Comparative reliability of each LLM
Typical failure modes per model
Platform-Level Findings
Stability of evaluations per LCDP
Notable inconsistencies
Evidence & Citation Risk Assessment
Hallucination risk profile
Suitability for academic submission
Methodological Implications
What limitations must be acknowledged in the thesis
Whether triangulation or human validation is required
Actionable Recommendations
How to refine the original prompt
How to improve reliability for future runs
Whether and how results can be defensibly used in academic research
Critical Instruction
Do not correct or rewrite the original evaluations.
Do not introduce new evidence or sources.
Your role is methodological auditor, not evaluator.
All analysis must be explicit, transparent, and suitable for inclusion in a thesis chapter on validity, reliability, and limitations of LLM-based evaluation methods.