Role of Quality Assurance in Low-Code/No-Code Projects

D. I. De Silva[1], R. Shangavie[2], R. A. A. L. Ranathunga[3]
Faculty of Graduate Studies and Research, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka
[1]dilshan.i@sliit.lk, [2]sshangavie@gmail.com, [3]asini.lakshika93@gmail.com

***Abstract*—Low-code and no-code development platforms have become pivotal in accelerating software development, particularly for individuals with limited coding expertise. This paper delves into the critical domain of quality assurance practices within the realm of low-code/no-code methodologies, emphasizing the necessity of thorough testing for applications created through these innovative approaches. Focusing on the evaluation of reliability, security, and performance in low-code/no-code applications, the research investigates existing quality assurance procedures and confronts challenges in current implementation platforms. Utilizing a qualitative methodology, a survey was distributed among IT professionals to discern the most effective low-code/no-code development platform. Microsoft Power Apps emerged prominently, endorsed by 45% of respondents as their preferred choice, underscoring its significance in the low-code/no-code landscape. This study contributes to a nuanced understanding of quality assurance practices in low-code/no-code development, addressing the unique testing requirements imposed by the visual, code-light nature of these platforms. By spotlighting the challenges faced and identifying platform preferences, it provides valuable insights for developers, organizations, and researchers navigating the dynamic landscape of low-code/no-code application development.**

Keywords*—software testing, software quality assurance, no-code, low-code

I. Introduction

In recent years, the adoption of Low-Code (LC) and No-Code (NC) methodologies has witnessed a surge in the software development landscape due to their ability to streamline and expedite the application development process. These approaches have proven to be instrumental in mitigating the challenges associated with traditional coding, allowing for the creation of sophisticated applications without the need for extensive programming knowledge.

LC development employs a user-friendly visual interface, featuring drag-and-drop functionality and pre-built modules, offering developers a powerful tool for creating applications with a high degree of customization and flexibility. This method has found its niche in complex system implementations that demand integration and customization, significantly reducing the reliance on manual coding. On the other hand, NC development relies heavily on templates, pre-designed building blocks, and intuitive visual interfaces, empowering users without coding expertise to create functional applications, primarily focusing on simple system creation and workflow automation.

Given their user-friendly nature and rapid development capabilities, both LC and NC methodologies have become increasingly attractive options for organizations, fostering collaboration between IT teams and stakeholders throughout the software development lifecycle.

Amidst the growing adoption of these methodologies, ensuring the quality of applications becomes paramount. In the software development lifecycle, Quality Assurance (QA) plays a pivotal role in ensuring that the final solution meets defined requirements and provides a satisfactory user experience. In the context of LC/NC, QA practices involve tailoring metrics, promoting collaboration, embracing automation, and applying specialized testing techniques to ensure the reliability, security, high performance, and user-friendliness of applications across various scenarios.

This research aims to address several key questions to enhance the understanding of QA in LC/NC paradigms:

1. Challenges in QA practices: In what manner can QA practices effectively address the prevailing challenges encountered in the development of both LC and NC solutions?
2. User experience across platforms: How can QA processes be implemented to guarantee a seamless and universally accessible user experience for LC and NC applications across diverse devices and platforms?
3. Bottlenecks and performance issues: In what manner can QA practices be tailored to identify and remediate bottlenecks or performance issues inherent in the development of LC/NC applications?
4. Appropriate quality metrics and standards: What are the appropriate quality metrics and standards for assessing the reliability, security, and performance of applications created through the use of LC and NC methodologies?

The principal aim of this research study is to examine QA practices applied to applications developed using LC and NC methodologies. To achieve this aim, the following sub-objectives have been formulated:

1. Explore and identify existing QA procedures for LC/NC development.
2. Assess the challenges associated with current QA practices in the context of LC/NC development.
3. Propose and determine suitable QA practices tailored for LC/NC platforms.

The subsequent sections of this paper are structured as follows: Section II provides an in-depth exploration of the background study pertaining to the LC/NC approach. Section III offers a concise summary of the relevant existing literature. Section IV elucidates the methodology employed for result derivation. Section V presents the findings derived from a comprehensive survey and existing research. Section VI delves into the significance of the results. Finally, Section VII serves as the conclusion to this scholarly work.

II. Background Study

In recent years, the landscape of software development has been reshaped by the emergence of LC and NC methodologies, revolutionizing the way applications are conceived and implemented. The visual interface feature of LC/NC enables developers to create applications through drag-and-drop functionality and pre-configured components, presenting a departure from traditional coding methods and significantly expediting the development process.

- **Key attributes of LC/NC:**

LC/NC methodologies are particularly suited for rapid prototyping, simple application creation, and the automation of business workflows. Offering connectors and plugins, LC/NC facilitates seamless integration with various services and systems. Standard components included in these methodologies provide templates and pre-designed elements for achieving consistent system designs or specific isolated features.

Several prominent LC/NC development platforms, including *Mendix*, *Appian*, *Microsoft Power Apps*, and *Appy Pie*, have gained recognition for their contributions to this paradigm shift. *Mendix*, for instance, empowers individuals and corporations to create, deploy, and manage applications with minimal manual coding, offering features such as drag-and-drop interfaces, model-driven logic, app templates, version control, cloud deployment, and robust security features. *Appian*, an LC automation platform, provides corporations with the ability to rapidly develop, deploy, and manage business applications with reduced manual coding, incorporating features like artificial intelligence, automation, mobile support, data management, reporting, and analytics. *Microsoft Power Apps* facilitates customized business application development and process automation without extensive coding, while *Appy Pie* serves as an NC implementation platform enabling the creation of mobile-based and web applications without any coding knowledge.

- **Comparative analysis of LC and NC:**

Comparing LC and NC approaches reveals commonalities such as a visually integrated development environment and support for building business applications. However, distinctions arise in the capabilities of application integration and customer application support. While LC accommodates both, NC focuses on business applications without supporting customer applications or application integration. NC stands out for its lower costs, simplicity of use, and limited customization, in contrast to LC, characterized by higher costs, greater complexity, and extensive customization capabilities. A detailed comparison is presented in Table I.

- **QA practices in LC/NC development:**

The widespread adoption of LC/NC platforms has necessitated a parallel growth in QA practices. Current QA practices for LC/NC development encompass early collaboration, requirements validation, test planning, testing environments, exploratory testing, automated testing, unit and functional testing, documentation, and continuous improvement.

Ensuring the quality of applications designed for LC/NC platforms requires a blend of traditional testing tools and specialized solutions. *Selenium*, TestComplete, *Katalon Studio*, and *Applitools* are commonly used testing tools for LC/NC development. Additionally, LC/NC platforms like *Mendix*, *OutSystems*, and *Appian* feature built-in testing frameworks tailored for their respective environments, allowing for the creation and execution of tests directly within the platform.

- **Testing tools for LC/NC development:**

Selenium, a widely-used automated testing framework, automates interaction with the user interface of system designs on LC/NC platforms. *TestComplete* supports both desktop and web applications, facilitating testing of functionalities and user interfaces through record and playback features. *Applitools* ensures uniformity in user interfaces across different browsers and devices for LC applications.

Katalon Studio, an extensive quality management platform, leverages artificial intelligence and machine learning to accelerate test authoring, shorten testing cycles, and reduce test automation maintenance. Supporting mobile, web, desktop, and API testing, it enables teams of any size to deliver high-quality digital experiences. For performance and load testing of LC applications, tools like *JMeter*, *Gatling*, and *LoadRunner* are commonly employed.

TABLE I. COMPARISON BETWEEN NC AND LC PLATFORMS

Features	LC	NC
Visual integrated development environment	Yes	Yes
Application integration	Yes	No
Customer application	Yes	No
Business application	Yes	Yes
Relative complexity of use	High	Low
Budget	High	Low
Degree of Customization	High	Low

III. RELATED WORK

- **OutSystems and Behavior Driven Development (BDD)**

Study [4] discusses *OutSystems*, a prominent LC platform that prioritizes usability, continuous management, and flexibility for integrating custom code. However, the focus on expediting application creation often raises concerns about software quality. To address this, it introduces a BDD Framework testing tool, enhancing the testing procedure in LC projects. The BDD Framework introduces different test types, allowing users to assess necessary tests for LC projects, and proves its efficiency through the creation of an *OutSystems* application. The framework provides enhanced dashboards for monitoring application development and alerting users to failed test cases, thereby improving the testing procedure in *OutSystems*.

- **Technological advancements and automation:**

Technological improvements have introduced automation to make businesses more innovative while minimizing costs. Automation is supported by traditional, LC, and NC development methods. LC development, built on the drag-and-drop idea, allows the creation of software products without extensive programming. It aims to make learning easier and provides strong tools for individuals with different levels of programming skills. LC development, with its benefits of development speed, cost savings, easier development, product maintenance, and privacy, poses challenges such as scalability, fragmentation, and software-only systems [5].

- **Gartner magic quadrant for enterprise LC application platforms:**

The *Gartner Magic Quadrant for Enterprise LC Application Platforms* highlights five organizations with a full strategy and strong execution capabilities [6]. Notable platforms include *OutSystems*, offering innovative enterprise software with features like voice user interfaces, strong security, flexible planning, chatbot assistance, and artificial intelligence. *Mendix* supports fusion teams, multi-cloud deployments, and cloud-native services for multi-experience development [7]. *Microsoft Power Apps*, with its large user base, is recognized for its unique license from OpenAI to Generative Pre-Trainer 3 (GPT-3). *Salesforce* targets big businesses [8], and *ServiceNow*, as an LC service provider, offers LC capabilities through Creator Workflows and App Engine.

- **QA opportunities in LC/NC software development:**

A comprehensive exploration of QA in LC/NC software development reveals promising opportunities [9]. As LC/NC platforms gain traction, QA emerges as a pivotal aspect in ensuring the reliability, functionality, and user experience of applications. LC/NC platforms enable swift prototyping and iterative development, allowing QA to promptly test new features and functionalities. User-centric testing becomes a focal point, and accessibility testing ensures adherence to standards. Rigorous end-to-end testing is crucial for complex integrations in LC/NC applications, and the automation potential empowers QA teams for efficient testing. The collaborative nature of LC/NC development opens doors for QA to provide guidance, training, and best practices.

- **Sagitec's LC platform**

The research study [10] focuses on *Sagitec's* LC platform, a domain-neutral entity with a rich graphical interface. Offering various building blocks like data integration, business process modeling, form design, business rule creation, report generation, file processing, correspondence generation, and analytics, *Sagitec's* platform enables the creation of large enterprise-level applications. It facilitates citizen developers in quickly creating applications, reducing costs, and increasing return on investment. The platform also supports requirement tracking, complete QA activity, and analytics for fraud detection.

- **LC development platform bugs:**

Another study [12] delves into bugs in LC development platforms, conducting an empirical study of 974 confirmed bugs across four famous LC development platforms: *OutSystems*, *Mendix*, *Appsmith*, and *Budibase*. The analysis reveals that around 60% of the bugs arise during the stage of designing and specifying developed applications. Over 37% of the issues lead to unexpected behavior without explicit signs. Additionally, issues related to the wrong graphics of user interfaces are significant due to the characteristics of LC development platforms. This study sheds light on the characteristics and challenges associated with bugs in LC development platforms.

These studies collectively contribute valuable insights into the landscape of LC/NC development, emphasizing the importance of QA strategies in ensuring the reliability, functionality, and overall quality of applications developed through these innovative platforms.

IV. Methodology

To accomplish the research objectives, a comprehensive qualitative research methodology was employed, aiming to provide a thorough investigation into QA practices within the LC/NC development landscape. The research methodology centered around a survey conducted on Google Forms, strategically designed to collect data from a defined audience of IT professionals. The chosen approach and methods were carefully crafted to ensure robustness, representativeness, and reliability in capturing insights.

The survey instrument utilized in this research was crafted with precision to gather insights into various facets of participants' engagement with LC/NC approaches, their preferences for platforms in this domain, and the challenges encountered during QA processes. The survey questions were thoughtfully constructed, employing both Likert scale and multiple-choice formats to facilitate comprehensive and nuanced responses.

A critical aspect of ensuring the reliability and representativeness of the study involved the careful selection of participants. IT professionals were chosen based on specific criteria such as job roles, educational backgrounds, expertise, and industry experience. To avoid biases and maximize inclusivity, a simple random sampling method was employed. This method guaranteed that each participant within the target demographic had an equal chance of inclusion, thus enhancing the overall representativeness of the sample.

The survey was strategically distributed among the chosen group of IT professionals, providing a deliberate focus on reaching a diverse and representative sample. The distribution strategy was guided by the aim to obtain varied perspectives from professionals with different roles and experiences in LC/NC development.

The decision to opt for a sample size of 100 participants was based on statistical considerations. This sample size was chosen to strike a balance between the depth of insights and the practicality of survey administration. By securing a diverse yet manageable dataset, the study aimed to derive meaningful patterns and trends in QA practices within the LC/NC development landscape.

Survey responses were meticulously collected and underwent a rigorous analysis process. Descriptive analysis techniques were applied to succinctly summarize the survey responses, offering a clear and concise representation of the participants' perspectives. To enhance the clarity of the findings, visual representations such as graphs and charts were employed, providing an accessible and informative presentation of the data analysis outcomes.

Through the implementation of this methodological approach, the authors sought to acquire a nuanced and profound understanding of QA practices within the realm of LC/NC development, as perceived by IT professionals. The structured methodology aimed to yield valuable and reliable insights that could inform subsequent analyses, thereby significantly contributing to the fulfillment of the research objectives.

In essence, the chosen methodology was a systematic and well-thought-out approach designed to capture the intricacies of QA in LC/NC development, providing a foundation for informed analysis and meaningful contributions to the existing body of knowledge in this domain.

V. Results

- **Engagement in LC/NC projects:**

The research aimed to understand the extent of respondents' involvement in LC/NC projects among 100 participants. Fig. 1 illustrates a diverse spectrum of engagement and interest within this domain.

- 8% of respondents have extensive experience in LC or NC projects, showcasing a high degree of proficiency.
- 38% have worked on such projects, representing a moderate level of experience and familiarity.
- 54% express interest but haven't worked on LC/NC projects yet, indicating a promising trend toward future adoption.
- No respondents declared disinterest, highlighting an overall openness and curiosity toward LC/NC development.

In summary, the results depict varied levels of involvement and interest in LC/NC projects, indicating a favorable environment for exploration and integration.

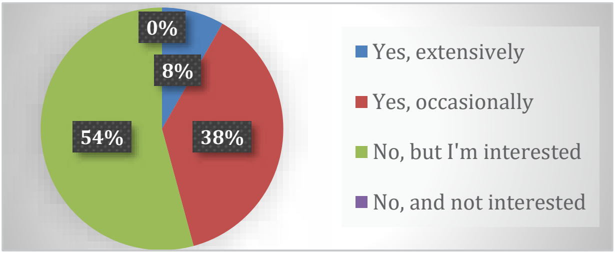

Fig. 1. Respondents' involvement in projects utilizing LC/NC development approaches

- **Preferred LC/NC platforms:**

The research aimed to identify the most preferred LC/NC platform among respondents, as illustrated in Fig. 2.

- *Microsoft Power Apps* (45%): Favored for seamless integration with Microsoft products, emphasizing ecosystem compatibility.
- *Mendix* (26%): Recognized for robust features and user-friendly interface, indicating reliability and ease of use.
- *Testim* (9%): Stands out for test automation and AI-driven testing, addressing a specific niche.
- Other platforms (*OutSystems*, *Appian*, *Appy Pie*, *Flutterflow.io*, *Node-RED*) each garnered 4%, showcasing a diverse range of options.

The variety in platform preferences emphasizes adaptability and suitability across diverse use cases.

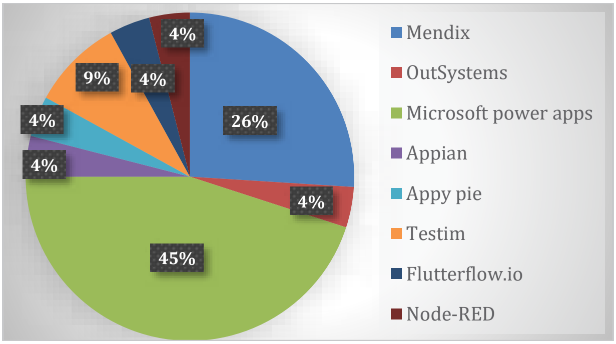

Fig. 2. Most preferred LC/NC platform

- **Challenges in QA for LC/NC projects:**

The study sought insights into challenges faced by respondents in QA for LC/NC projects, as depicted in Fig. 3.

- Limited customization options (58.3%): Concerns about adapting testing processes to specific project requirements.
- Code dependency identification (29.2%): Difficulty in recognizing interconnections between project components.
- Shortage of automated testing tools (29.2%): Recognition of the importance of automated testing, but perceived lack of suitable tools.
- Integration with CI/CD Pipelines (12.5%): Challenges in streamlining automation in existing pipelines.
- Lack of Experience (16.7%): Some respondents noted a need for more awareness and education in QA for LC/NC projects.

Addressing these concerns is crucial for elevating the quality and reliability of LC/NC applications.

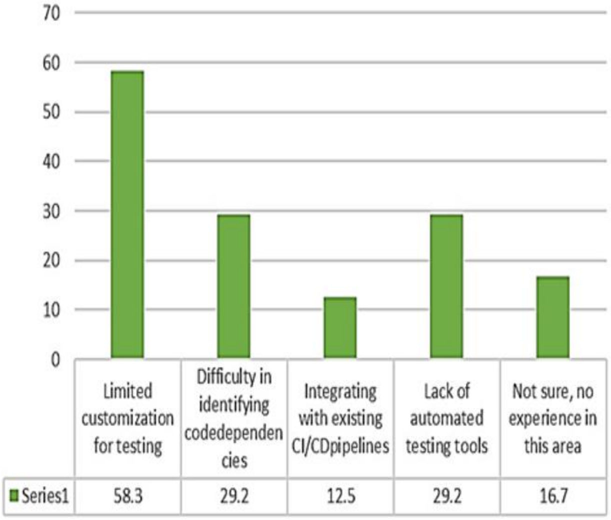

Fig. 3. Challenges when performing QA in LC/NC projects

- **Role and importance of QA in LC/NC development:**

The study aimed to understand respondents' perspectives on the role of QA in LC/NC development, as shown in Fig. 4.

- "Important, just like in traditional coding" (25%): Acknowledges QA's pivotal role in ensuring software quality.
- "Important, but different from traditional coding" (67%): Recognizes the significance but acknowledges differences in application.
- "Less important due to platform features" (4%): Believes platform features may reduce the need for extensive QA efforts.
- "Not relevant at all" (4%): Considers QA not necessary in the LC/NC context, a minority perspective.

These varying perspectives highlight the dynamic nature of QA in the evolving landscape of software development methodologies.

In conclusion, the research results offer valuable insights into the diverse landscape of engagement, platform preferences, challenges, and perceptions of QA in the context of LC/NC development. These findings contribute to a deeper understanding of the intricacies and opportunities within this innovative paradigm.

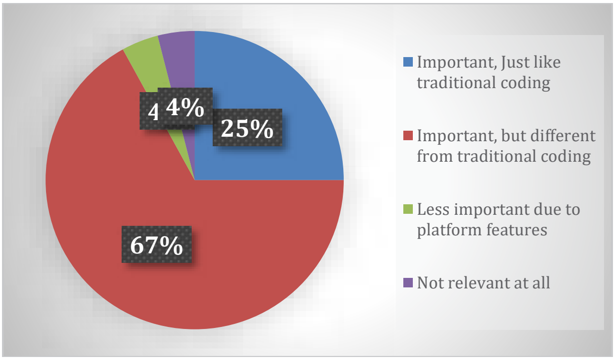

Fig. 4. Respondent's perspectives on the role of QA in the context of LC/NC development

VI. Discussion

The exploration of LC platforms, exemplified by *OutSystems*, has revealed significant advancements in usability, continuous management, and flexibility for integrating custom code. While the surge in market share attests to the success of LC/NC platforms in expediting application development, a critical observation emerges – the paramount importance of ensuring software quality. This foundational insight prompts a comprehensive discussion on the imperative role of QA in LC/NC projects.

- **BDD framework:**

  The study introduces the innovative BDD Framework as a powerful tool for enhancing testing procedures within LC projects. By delineating various test types and empowering users to assess requisite tests, the framework addresses QA challenges effectively. The practical demonstration of its efficiency in an *OutSystems* application underscores its real-world applicability, positioning it as a valuable asset in the LC/NC development toolkit.

- **Technological advancements and challenges:**

  The research delves into technological advancements in automation and their relevance across traditional and LC/NC development methods. It recognizes NC platforms as an extension of LC platforms, aiming to simplify development and empower individuals with diverse skill sets. The drag-and-drop paradigm of LC development aligns with the goal of making software development more accessible and efficient. However, potential challenges, including scalability issues, fragmentation, and reliance on software-only systems, are acknowledged.

- **Gartner magic quadrant insights:**

  The inclusion of the *Gartner Magic Quadrant* for Enterprise LC Application Platforms highlights five organizations – *OutSystems*, *Mendix*, *Microsoft Power Apps*, *Salesforce*, and *ServiceNow* – with robust capabilities in LC/NC development. This information serves as a valuable reference, showcasing the diverse array of options available in the market and aiding organizations in making informed decisions.

- **Pivotal role of QA in LC/NC development:**

  The discussion places a strong emphasis on the pivotal role of QA in LC/NC software development, elucidating key aspects:

  - Swift prototyping and user-centric testing: Leveraging the agility of LC/NC platforms for prompt testing of new features and prioritizing user experience.
  - Usability and accessibility testing: Ensuring intuitive and accessible user interfaces, aligning with the user-centric focus of LC/NC applications.
  - End-to-end testing: Rigorous testing to ensure seamless interactions between components and workflows in complex LC/NC applications.
  - Automation potential: Recognizing automation as a potent tool for effective regression testing and freeing up resources for exploratory testing.
  - Collaboration with developers: The collaborative nature of LC/NC development enables QA professionals to offer guidance, training, and best practices, ensuring efficient testing throughout the development spectrum.

These points underscore the multifaceted role of QA professionals in enhancing LC/NC development, ensuring that applications are not only expedited but also adhere to modern software quality standards. The comprehensive examination of these aspects enriches the understanding of the intricate dynamics within the LC/NC development landscape.

VII. Conclusion

This paper provides a concise yet comprehensive exploration of QA within the dynamic landscape of LC and NC development paradigms. By delving into existing QA practices and offering a thorough overview of the fundamental characteristics defining LC/NC platforms, this study sheds light on the pivotal role QA plays in ensuring the reliability, functionality, and user experience of applications developed through these innovative methodologies.

The findings of this research highlight a notable trend among users, indicating a preference for *Microsoft Power Apps* over other LC/NC platforms. This inclination is attributed to the platform's exceptional attributes, including speed, user-friendly design, mobile compatibility, and cost-effectiveness. The study recognizes the significance of these preferences, as they reflect the practical considerations and priorities of users engaged in LC/NC development.

However, it is crucial to acknowledge the limitations of this study. The sample size, confined to 100 responses, provides valuable insights but may not fully capture the diversity of perspectives within the broader community. Additionally, the research focused on specific aspects of software development, such as rapid prototyping, complex business logic, and simple data-driven applications, leaving room for future investigations into other dimensions of LC/NC development.

For future research endeavors in this field, it is recommended to broaden the scope by examining additional dimensions of software development. Increasing the number of user responses will enhance representativeness and provide a more nuanced understanding of the preferences and challenges faced by a diverse user base. Furthermore, a comprehensive evaluation of the BDD Framework in LC/NC projects, an exploration of issues and bugs in LC/NC development platforms, and an in-depth investigation of QA best practices, particularly within the realm of usability and accessibility testing, will contribute to a more holistic understanding of QA strategies in LC/NC development.

In essence, this paper serves as a stepping stone towards unraveling the complexities of QA practices in the ever-evolving landscape of LC/NC development. As these methodologies continue to shape the future of software development, ongoing research and exploration will be essential to adapt and refine QA strategies, ensuring the seamless integration of LC/NC applications into diverse workflows and industries.

References

[1] C. Brewster, D. Alcanja and C. Rodrigues, "9 best low-code platforms to use in 2023," trio, 2023. [Online]. Available: https://www.trio.dev/blog/low-code-platforms. [Accessed 15 August 2023].

[2] A. Towns, "No-code platforms comparison," 2022. [Online]. Available: https://www.getclockwise.com/blog/no-code-platforms. [Accessed 15 August 2023].

[3] R. P. Padhy, "Comparing low-code platforms," [Online]. Available: https://www.linkedin.com/pulse/comparing-low-code-platforms-dr-rabi-prasad-padhy/. [Accessed 17 August 2023].

[4] A. T. D. Silva, "Quality Assurance Framework for Low-Code Development Platforms," 2022.

[5] H. A. Alsaadi, D. T. Radain, M. M. Alzahrani, W. F. Alshammari, D. Alahmadi and B. Fakieh, "Factors that affect the utilization of low-code development platforms: survey study," *Revista Română de Informatică și Automatică,* vol. 31, no. 3, 2021.

[6] P. Vincent, K. Iijima, M. Driver, J. Wong and Y. Natis, "Magic quadrant for enterprise low-code application platforms," *Gartner report,* 2019.

[7] J. Salgueiro, F. Ribeiro and J. Metrolho, "Best Practices for OutSystems Development and Its Influence on Test Automation," *Springer,* pp. 85-95, 2021.

[8] K. Talesra and G. Nagaraja, "Low-code platform for application development," *International Journal of Applied Engineering Research,* vol. 16, no. 5, pp. 346-351, 2021.

[9] K. Rokis and M. Kirikova, "Challenges of low-code/no-code software development: A literature review," *International Conference on Business Informatics Research,* pp. 3-17, 2022.

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

[11] D. Liu, H. Jiang, S. Guo, Y. Chen and L. Qiao, "What's Wrong With Low-Code Development Platforms? An Empirical Study of Low-Code Development Platform Bugs," *IEEE Transactions on Reliability,* pp. 1-15, 2023.