Model-Based Approaches in Precision Agriculture: A Survey and Insight into the Potential and Challenges of Low-Code Development Platforms

David Cristobal Munoz

August 21, 2023

WAGENINGEN
UNIVERSITY & RESEARCH

DISCLAIMER

This report is written by a student of Wageningen University as part of a master programme and is executed under supervision of the chair Information Technology. This report is not an official publication of Wageningen University and research. The content of this report is not the opinion of Wageningen University and Research.
Use of information from this report is for own risk and it is advised to check this independently before the information is used. Wageningen University is never liable for the consequences that result from use of information from this report. It is not allowed to publish or reproduce the information from this report without explicit written consent of:

Wageningen University and Research
Information Technology Group
PO Box 16
6700 AA WAGENINGEN
T: +31 (317) 482980
E: office.fte@wur.nl

Contents

1 Introduction **1**

1.1 Precision agriculture 1

1.2 Background delimitations 1

1.3 Current Status 2

1.4 Anticipated State 3

2 SLR **4**

2.1 Problem Articulation 4

2.2 Research Aims and Contributions 5

2.3 Research methodology 6

2.4 Limitations and Threats to Validity 9

3 Results of the SLR **10**

3.1 Scopus' evaluation and analysis 10

3.2 IEEE evaluation and analysis 13

3.3 ACM evaluation and analysis 15

3.4 Wiley's evaluation and analysis 16

3.5 Literature Selection and Justification 17

3.6 Research Question 1 (RQ1):What instigates the incorporation of model-based approaches that inspire low-code technology in precision agriculture? 19

3.6.1 Addressing the Complexity of Precision Agriculture 20

3.6.2 Accelerating the Software Development Process 21

3.6.3 Democratizing Precision Agriculture 22

3.6.4 Consolidated Insights on Low-Code Technology's Impact on Precision Agriculture 22

3.7 Research Question 2 (RQ2): What unique services can model-based approaches facilitate to advance precision agriculture through low-code platforms? 23

3.7.1 Stakeholder-Centric Services of Low-Code Platforms in Precision Agriculture . . . 23

3.7.2 Data Collection Facilitation 25

3.7.3 Data Analysis Enhancement 25

3.7.4 Decision Support Services 26

3.7.5 System Integration Services 27

3.7.6 Enhancing Accessibility and Democratization 28

3.7.7 Comprehensive Analysis of Low-Code Platform Services in Precision Agriculture . 30

3.8 Research Question 3 (RQ3): which low-code platform fits better solutions with the model-based approaches on their usage to optimize precision agriculture? 30

3.8.1 Introduction to RQ3: 30

3.8.2 Analysis of Low-Code Platforms for Precision Agriculture: 31

3.8.3 Overview of Specific Low-Code Platforms: 32

3.9 Research Question 4 (RQ4): What are the critical steps involved in developing a low-code system based on the model-based approach in precision agriculture? 33

3.9.1 Understanding the Developer Categories and Their Needs 33

3.9.2 Requirement Analysis and Feasibility 34

3.9.3 Data Modeling 36

3.9.4 User interface design 37

3.9.5 Implementation of Business Logic and Integration of External Services 37

3.9.6 Testing and Deployment 37

3.9.7 Maintenance and Continuous Improvement 38

3.10 Research Question 5 (RQ5): What challenges are associated with implementing model-based approaches that could encourage low-code technology in precision agriculture? . . 38

4 Comparative Analysis of Node-RED and Python Implementations for Irrigation Control in Precision Agriculture **39**

4.1 Methodology 39

4.2 Node-Red simulation design 40

4.3 Python simulation design 42

4.4 Code Elaboration 42

4.5 Implementation Overview 44

4.5.1 Comparison of Node-RED and Python 45

4.6 Metrics Evaluation Revision 46

4.7 Case study results and analysis 47

4.8 Case study discussion 47

4.9 Case study conclusion 48

5 General conclusion **48**

6 Discussion and Future Work 50

Abstract

This study presents an in-depth examination of the intersection between model-based approaches in precision agriculture and low-code development platforms (LCDPs), two evolving and transformative areas. LCDPs have the potential to expedite the software development process by reducing the need for extensive coding skills, but their scalability in complex domains like precision agriculture remains a challenge. Precision agriculture, on the other hand, is focused on improving crop health and productivity by using IoT, data analytics, and advanced technologies. The integration of these technologies could be simplified by LCDPs, provided farmer expertise and data scalability issues are properly addressed.

This research makes use of a systematic literature review (SLR) to outline the opportunities and challenges of utilizing LCDPs in precision agriculture. It sheds light on the influence of citizen developers and the impact of user experience on the adoption and success of LCDPs within this context. The study also explores the motivations for integrating LCDPs into precision agriculture, possible applications, and the criteria for selecting appropriate platforms. It outlines a detailed process for building low-code systems and highlights the related challenges, paving the way for future research in this area.

In addition to the SLR, a comparative analysis was performed using Node-RED, a well-known LCDP, and Python, a high-level programming language. An irrigation control system served as the focal point for this comparison, which covered critical metrics like development time, resource utilization, reliability, and maintainability. The results revealed that Node-RED is more accessible and efficient for simpler tasks, while Python provides greater flexibility and strength for more intricate operations.

This multidimensional research underlines the importance of LCDPs in the sphere of model-based approaches to precision agriculture, suggesting that the choice between low-code and high-level programming depends on specific requirements, user abilities, and task complexity. By interweaving insights from both the SLR and the comparative analysis, this study contributes significantly to the understanding of software development best practices in precision agriculture, and sets the stage for future research directions.

1 Introduction

1.1 Precision agriculture

Today, it's hard to ignore the impact of precision agriculture. As the world's population is quickly growing, we're facing the pressing need for escalated food production. But we must not forget our responsibility to the environment. Here, precision agriculture plays a starring role. It's a significant game-changer in sustainable agronomy, helping us find a delicate balance between skyrocketing agricultural productivity and the need to preserve ecological integrity Lal (1991).

What makes precision agriculture a matter of global concern? The answer lies in its unique capacity to maximize resource utilization. This is achieved by using cutting-edge technology and automation to manage resources down to the minutest details. The result? A remarkable reduction in waste and a spike in efficiency. In essence, it allows us to tailor our efforts based on specific needs of different segments within a field Dobermann et al. (2004).

Like most good things, the implementation of precision agriculture comes with its share of hurdles. It can be seen among these, the substantial cost of technology and equipment. It's not just about creating sensors, GPS systems, automated machinery, and data analysis software. The real challenge lies in their affordability, particularly for small-scale farmers, which can inhibit the widespread acceptance of precision agriculture Ayaz et al. (2019).

Another obstacle in this journey is the constant need for skilled personnel. Precision agriculture isn't simply about running machines; it's also about interpreting vast amounts of data to make informed decisions about resource management. Unfortunately, the scarcity of skilled personnel can represent significant barriers to implementing precision agriculture. This is even more pronounced in developing countries, where educational and training resources are often scarce Dobermann et al. (2004).

Another significant concern is data management. In precision agriculture, sensors and other technologies produce an overwhelming volume of data. This data needs to be securely stored, processed, and subsequently analyzed. All of these demand robust data management systems and advanced analytical tools. However, their implementation can be complicated and costly, being yet another challenge Bhakta et al. (2019).

Despite these obstacles, the potential benefits of precision agriculture are too significant to be overlooked. By optimizing resource usage, precision agriculture could be our best shot at boosting agricultural productivity while minimizing environmental repercussions. This assumes particular importance in light of climate change, which poses daunting challenges to agricultural production. By managing resources with precision, we can build resilience to climate change, thereby ensuring the sustainability of agricultural production amid changing environmental conditions Lal (1991).

Although precision agriculture is full of considerable challenges, its importance in the current era, and should not be overstated. It's a lighthouse guiding us toward sustainable agronomy. As we confront and overcome these challenges, we're paving the way to fully harness the potential of precision agriculture, striking a balance between escalating agricultural productivity and preserving our precious ecological integrity.

1.2 Background delimitations

The rise of Low-Code Development Platforms (LCDPs) is creating quite interest in the sphere of software development. it is useful to ask, what sets these platforms apart? The answer lies in their user-friendly, model-based visual development environments that simplify application component development into a mere activity of drag-and-drop. LCDPs are increasingly capturing the interest of the tech world for their impressive ability to streamline software development cycles, reduce the necessity for extensive coding knowledge, and supercharge productivity Collomb & Hascoët (2008); Richardson & Rymer (2016a). But like every other new tech solution, LCDPs have their fair share of challenges. One of the pressing issues is the scalability of applications and handling their complexity when built using these model-based platforms Kolovos, Paige, & Polack (n.d.); Kolovos, Rose, et al. (n.d.).

By taking a look into precision agriculture, an intriguing domain where model-based approaches are making a substantial impact. Envisioning this as a futuristic form of farming where data-driven insights guide the way we manage crops and soil Ahmed et al. (2018); Y. Zhang (2019). In this high-tech agricultural landscape, the Internet of Things (IoT), primarily based on model-based systems, takes center stage by enabling real-time tracking of a vast range of agricultural parameters and even automating a host of farming procedures Elijah et al. (2018).

A question that seems almost inevitable at this point is – what if we brought together LCDPs and precision agriculture? Just considering this possibility brings into focus a variety of potential benefits. We could leverage LCDPs to tailor-make IoT applications specifically for precision agriculture, augmenting the efficacy with which farmers monitor and control their crops Jayaraman et al. (n.d.). Imagine having applications available that pull together data from a variety of field-deployed sensors, process this data in real-time, and furnish farmers with actionable insights for immediate implementation Gayatri et al. (n.d.).

However, the journey to integrating LCDPs into the realm of precision agriculture isn't devoid of obstacles. One major roadblock could be the farmers limited familiarity with navigating these model-based platforms Sahay et al. (n.d.). Another concern to consider seriously is the scalability of applications developed via LCDPs, particularly in light of the voluminous data demands and real-time data processing requirements inherent in precision agriculture Kolovos, Rose, et al. (n.d.).

But don't let these challenges eclipse the potential benefits of integrating LCDPs with precision agriculture. By simplifying the developmental process through model-based approaches, LCDPs could make advanced IoT applications more farmer-friendly, thereby potentially driving up productivity and sustainability in the agricultural sector Ihirwe et al. (n.d.); Jayaraman et al. (n.d.).

Using LCDPs in precision agriculture also aligns with the broader trend towards digital transformation currently sweeping the agricultural sector. As these digital technologies continue to evolve, they're poised to play an increasingly critical role in overcoming the challenges that modern agriculture faces, from improving resource efficiency to bolstering crop yields Buttafuoco & Lucà (2016); Himesh et al. (2018).

In a nutshell, the fusion of model-based low-code development and precision agriculture holds immense potential for reshaping the future of farming. However, more research is needed to overcome the hurdles associated with the adoption and application of LCDPs in precision agriculture. there is a need to devise strategies to enable farmers to effectively use these platforms and explore solutions to enhance the scalability and performance of low-code applications Bucchiarone et al. (2020); Sahay et al. (n.d.).

This comprehensive exploration into the convergence of model-based low-code development and precision agriculture not only sets the stage for future research but also underscores the potential for innovative solutions in the agricultural sector. As we continue to deepen our understanding and application of these technologies, we're inching closer to a future where farming is not only more productive and efficient, but also more sustainable Tzounis et al. (2017); McBratney et al. (2005).

1.3 Current Status

The modern landscape of low-code development and precision agriculture is quite moving forward, showcasing remarkable advancements, but there's still a clear gap in integration between these two domains. Low-code development, has been getting more attention based on its inherent model-based approach, and has turned into a new perspective for more efficient application development. We see platforms like OutSystems, Mendix, and Microsoft PowerApps developing more interest from a diverse group of stakeholders U. Frank et al. (2021). The reason? Well, their model-based features make software development quicker and more efficient, essentially making coding skills less of a necessity. This opens the door wide open to non-technical users Lethbridge (2021), shaking things up across various sectors, agriculture being one of them Waszkowski (2019).

On the other side, precision agriculture, firmly grounded in model-based systems, is gradually picking up to be considered as a potent method to boost agricultural productivity and sustainability. The use of IoT devices, data analytics, and a suite of cutting-edge technologies in the realm of precision agriculture has shown promising results in upping crop yield and reducing environmental damage Y. Zhang (2019). For instance, the application of IoT in smart farming is a big topic of research, with studies spotlighting its potential in delivering smart agricultural solutions for superior yield Jarolimek et al. (2017).

However, despite these impressive strides in both fields, a noticeable void exists in combining model-based low-code development into the framework of precision agriculture. It's a bit surprising, given the potential advantages this merger could bring. Low-code platforms could be used to whip up applications for data aggregation, analysis, and decision-making in precision agriculture. This could streamline the process and make it more user-friendly for farmers and agricultural workers Talesra & Nagaraja (2021).

So, why this lack of integration? A plausible explanation could be the perceived complexity and technical nature of both fields. Although low-code development, especially with its model-based focus, aims to simplify software development, it's still often seen as a playground for IT pros. Likewise, precision agriculture, with its reliance on high-tech, could seem intimidating to those without a tech background ALSAADI et al. (2021). Such perceptions could stand in the way of integrating low-code development into precision agriculture.

Moreover, current low-code platforms might not be fully equipped to handle the specific demands of precision agriculture. Take, for instance, the need to manage colossal amounts of data in precision agriculture, requiring advanced data processing and analysis capabilities. While some low-code platforms, particularly those employing model-based approaches, do offer data management features, they may not cut it for the specialized needs of precision agriculture Cabot (2020).

To sum it up, while low-code development and precision agriculture have each made fantastic progress in their respective domains, their integration still seems like a distant achievment. The potential benefits of such integration are massive, but considerable hurdles need to be tackled. It's essential for future research and development initiatives to prioritize bridging this gap and exploring how model-based low-code development can be effectively roped in within the context of precision agriculture.

1.4 Anticipated State

In an ideal world, combining low-code platforms using model-based approaches with precision agriculture would be close to a revolution in both domains. Low-code platforms, designed to make the development process smoother, could make precision agriculture more accessible to a wider user base, including those without much coding prowess. This would essentially democratize precision agriculture, bringing more voices to the table in its evolution and deployment Lethbridge (2021).

Integrating model-based low-code platforms into precision agriculture could also supercharge the efficiency and effectiveness of agricultural practices. Consider this – precision agriculture relies heavily on collecting and analyzing data to guide farming decisions. Low-code platforms, with their model-based schemas, could simplify the process of creating applications for data collection and analysis, thereby enhancing the efficiency of these operations Nikkilä et al. (2010).

Additionally, the use of low-code platforms could pave the way for more advanced, customized solutions for precision agriculture. For instance, model-based low-code platforms could be used to develop applications specifically designed to tackle the unique needs and challenges of different types of farming, like organic farming, permaculture, or aquaponics Waszkowski (2019).

However, merging low-code platforms and precision agriculture as mentioned before has a lot of challenges. One of the main challenges is the need for a deep understanding of both fields. This calls for a multidisciplinary approach that blends expertise in software engineering, agriculture, and other relevant fields Kitchenham & Brereton (2013).

Another issue is the need for sturdy and reliable low-code platforms, especially those using model-based approaches, that can handle the complexity and scale of precision agriculture applications. This calls for continued research and development in the low-code platforms domain, along with thorough testing and validation of these platforms U. Frank et al. (2021).

In the end, what the merger is looking at is to have the possibility of impulsing a new era of precision agriculture that's more accessible, efficient, and effective. This wouldn't just be a win for the agriculture sector, but would also advance broader societal goals, like boosting food security and sustainability McBratney et al. (2005).

2 SLR

2.1 Problem Articulation

In efforts to untangle the connection between low-code platforms with model-based approaches and precision agriculture, one can encounter a big amount of scattered information. Found in various research papers and studies, the information is often shown indirectly or in broader contexts. This fragmentation underscores the need for a targeted, systematic exploration of this particular convergence. Currently, there is a noticeable absence of a systematic literature review (SLR) that concentrates solely on the application of low-code solutions, specifically model-based ones, in precision agriculture or smart farming. This study aspires to unify this fragmented knowledge and create a comprehensive, systematic review of the applications of low-code technology with model-based approaches in these particular sectors.

The main goal of this research is to extract a clear overview of present applications and potential trajectories of model-based low-code technology in agriculture. The focus is on directing the attention on how this transformative technology can offer innovative viewpoints and effective solutions to help with the complex challenges inherent to precision agriculture and smart farming. A specific area of intrigue is the exploration of user experience and the role of citizen developers. These individuals, despite potentially lacking comprehensive knowledge in software development, are in a prime position to gain from and contribute to the model-based low-code domain.

This research endeavors to cultivate a deeper understanding of the potentials and challenges of deploying model-based low-code technology within the agricultural context. The aim is not only to catalogue existing uses, but also to anticipate the problems that may pop up during its wider implementation. The insights from this study will serve as a solid base for future research across various disciplines, offering a structured and easily accessible pool of information for researchers to draw from. Consequently, the potential contribution of this SLR extends beyond the borders of precision agriculture, sparking progress in model-based low-code applications across different sectors.

2.2 Research Aims and Contributions

As mentioned, before, the objective of this Systematic Literature Review (SLR) is to generate a comprehensive framework outlining the current landscape and future trajectory of model-based approaches and their implication in the advent of low-code platforms in Precision Agriculture (PA). This framework is concocted by posing five distinct research questions, each showing a unique aspect of this intersecting domain. These queries are designed to capture the broad range and complexity of the subject, dissecting the rationales and mechanisms of embedding model-based methods to advance low-code technology in PA, while simultaneously evaluating both the potential benefits and challenges.

Research Question 1 (RQ1): What instigates the incorporation of model-based approaches that inspire low-code technology in precision agriculture? This question goes into the initial reasons and catalysts for analyzing the application of model-based methods in PA that stimulate the use of low-code platforms. As this is an emergent field, especially in the sphere of model-driven engineering, it's imperative to grasp the underlying reasons and potential benefits that comprehend this fusion of technologies as a promising area of exploration.

Research Question 2 (RQ2): What unique services can model-based approaches facilitate to advance precision agriculture through low-code platforms? This question seeks to demonstrate the potential applications and services offered by model-based methods that could make PA a better option, considering their implications for low-code technology. It's pivotal to test these unique strengths and features, which could position low-code platforms as a crucial instrument in modern agriculture.

Research Question 3 (RQ3): which low-code platform fits better solutions with the model-based approaches on their usage to optimize precision agriculture? This question addresses the necessity for a consistent appraisal and comparison of various model-based approaches, thereby identifying the most fitting use of low-code platforms in PA. The answer to this question is essential in finding future applications and deployment.

Research Question 4 (RQ4): What are the critical steps involved in developing a precision agriculture system using a model-based approach that could facilitate low-code application? This question underscores the importance of comprehending the process of building a system within the PA and low-code context, utilizing model-based methods that could be based on low-code application. The insights obtained from this research question can serve as a guide for future developers and researchers in the field.

Research Question 5 (RQ5): What challenges are associated with implementing model-based approaches that could encourage low-code technology in precision agriculture? The final question seeks to show potential obstacles, challenges, and factors that might obstruct the effective implementation of model-based approaches in PA that could lay the groundwork for low-code technology. This question aims to offer a balanced viewpoint, ensuring that the benefits, application processes, and potential difficulties of implementing such methods are depicted after being investigated.

These research questions have been created to offer a thorough, balanced, and pragmatic perspective on the application of model-based methods and their potential to implement the low-code technology use in PA. The questions strive to comprehend the technological feasibility, practical application, potential hurdles, and influential factors for the successful integration of model-based approaches in precision agriculture, thereby clearing the way for low-code platforms. The collective responses to these questions will contribute to achieving the research objective, providing a deeper understanding and more efficient use of model-based methods in precision agriculture, and setting the stage for future low-code platform implementation.

2.3 Research methodology

This study is inspired and based on the methodology developed by Bucaioni et al. (2022), a methodological approach that has its roots in the guidelines provided by Kitchenham & Brereton (2013) on undertaking systematic reviews within the domain of software engineering. This methodology is categorized into three integral stages: Planning, Conducting, and Reporting. as can be seen in figure:1

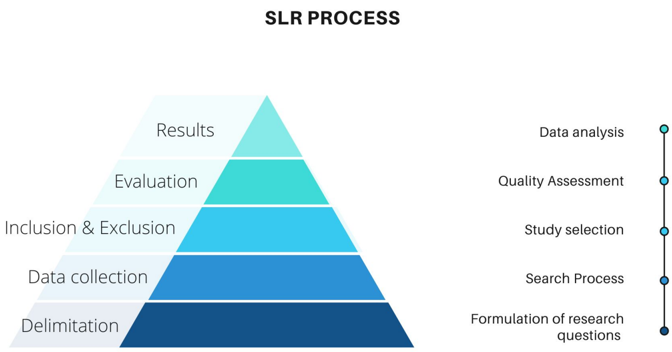

Figure 1: Steps on the SLR process

The planning stage plays a fundamental role in framing the scope of the review. The primary goals within this phase were to confirm the necessity for a systematic review focused on the use of low-code in precision agriculture, to corroborate the research objective and associated research questions, and to construct a methodical research protocol that ensures a systematic execution of the study. The culmination of the planning phase is manifested in a comprehensive research protocol Bucaioni et al. (2022).

Going into the conducting phase, we started on carrying out the activities meticulously detailed in the research protocol. These activities, fundamental to the research process, included collection of relevant data, defining the criteria for inclusion and exclusion of said data, extraction of key data, and the subsequent evaluation of the extracted data Bucaioni et al. (2022).

The procedure for search and selection, is an integral component of the data collection phase, four relevant scientific databases were selected. The research also incorporated an automated exploration of grey literature, which entries were selected based on a qualitative assessment of each paper's methodology and purpose. This exploration was facilitated using the Google search engine, followed by the implementation of selection criteria to filter and identify pertinent studies. The search process was further developed through snowballing, a strategy employed on Google Scholar to track and include related studies from the reference lists of initially selected articles. Kitchenham & Brereton (2013). figure:2

The next phase involved the delineation of the data extraction method, wherein a unique set of parameters was established to classify and draw comparisons among the primary studies. These parameters were derived in alignment with the research goal and employed a conventional key wording process. The data extraction method was used on the information extracted from each individual paper for analysis Bucaioni et al. (2022).

The Data Analysis stage comprehended the rigorous examination of the information extracted such as abstracts to derive insightful responses to the research queries, largely through the application of quantitative analysis techniques.

In the reporting phase, the researcher synthesized the derived data, creating a detailed record that not only documents the study's findings but also acknowledges potential threats to the study's validity.

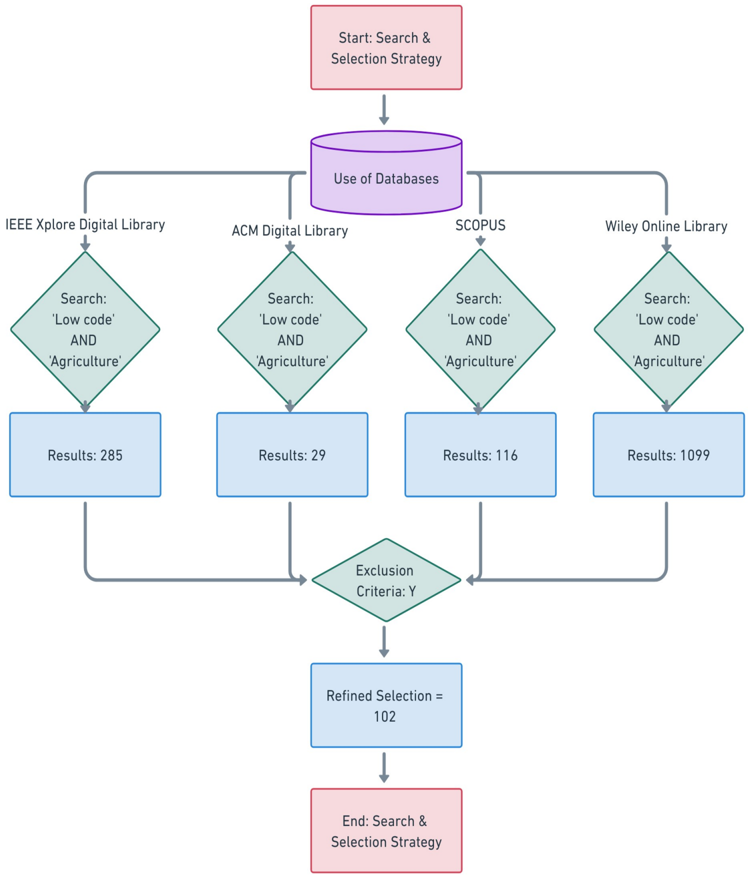

Figure 2: Steps on the SLR Selection process

The main goal of the systematic literature review is to discern, categorize, and critically assess the trends in publications, fundamental concepts, and to illustrate the contemporary and prospective application of low-code and model-based approaches in precision agriculture Kitchenham & Brereton (2013).

The search and selection strategy joined with the utilization of four primary databases that hold a considerable relevancy in the realm of software engineering: IEEE Xplore Digital Library, ACM Digital Library, SCOPUS, and Wiley Online Library. A customized search string was generated, reflecting the research goals and questions. Given the early stage of low-code as a concept, the search string was minimized to "Low code" AND "Agriculture", which, when used on the mentioned databases, shown minimal results Kitchenham & Brereton (2013).

The exploration of grey literature, an activity largely reliant on manual efforts, led us to valuable insights offered by specific entities operating in the low-code platform industry, including companies like Mendix. However, it was determined that these insights would be left out of this study on the questions non directly related to them (RQ3), to avoid potential bias that may be created from their economic interests, thereby potentially compromising the objectivity of this research Garousi et al. (2019).

Acknowledging the minimum amount of specialized information related to the application of low-code in precision agriculture, a method to go around it was designed, the method's first step was to distinctly outline the software requirements intrinsic to precision agriculture. This allowed for a comparison of these needs with the solutions provided by low-code development platforms. To broaden the scope of our research, the relatively newer term 'low-code' was replaced with its predecessor, 'Model Driven Engineering (MDE)', and the temporal range for the study was expanded from 2010-2022 to encompass the years 2000-2022. This modification aimed to encapsulate a more comprehensive span of time and to account for the evolutionary journey of low-code, dating back to its earlier form as MDE. Table 1

Table 1: Search references

Problematic	Ref	LC	Ref
PA systems are not well adapted to the needs of the users and are not easy to integrate into effective crop production decision support systems	Jayaraman et al. (n.d.)	Complexity reduction	Richardson & Rymer (2016a)
Scientists, developers, and farmers deal differently with reduction of uncertainties	Jayaraman et al. (2016)	Involvement of business profiles.	Richardson & Rymer (2016b)
Farmers do not know about the benefits of the resources that precision agriculture can offer to them.	Kolovos, Paige, & Polack (n.d.)	Digital twins	Richardson et al. (2014)
The development decision support systems to improve automation	Jayaraman et al. (n.d.)	Rapidity	Rymer et al. (2015)
Current Agricultural DSSs are not used to their full potential, and are not adapted to the complexity of farmers decision making	Lundell & Lings (2004)	Cross-platform accessibility	Sahay et al. (n.d.)
Lack of observability, lack of incentive to adopt new practices	Punjabi et al. (2017)	Enable citizen developers	Sahay et al. (n.d.)
Tedious data input requirements, poor user interface design for farmers situation	R Shamshiri et al. (2018)	Minimization of inconsistent requirements	Rymer et al. (2015)

After analyzing the results of the new keywords used on the string for the search of relevant information, the following numbers were obtained on the raw search.Table2

Table 2: Search stage 1

Database	Number of papers
IEEE	285
ACM	29
Scopus	116
Wiley	1099

Having a total of 1529 items in between journals, books, magazines, and conferences, the next step was to manually refine the selection based first on the title of the item later a second filter using the abstract, and by the end basing the filter on the content.Bucaioni et al. (2022) Table 3

Table 3: Search stage 2

Library	keywords	title	Abstract	Content
IEEE	285	95	57	31
ACM	29	23	20	20
Scopus	116	64	55	46
Wiley	1099	8	5	5
Total	1529	190	120	102

In a Systematic Literature Review (SLR), the primary goal is to comprehensively identify, evaluate, and interpret all available research relevant to a particular research question, topic area, or phenomenon of interest. To achieve this, it is important to have a clear, structured, and systematic approach to assess and grade each paper's relevance and quality. The grading system mentioned below serves this purpose:

S. No: The serial number or index provides a unique identifier for each paper, which is essential for managing and referencing the papers during the review process. This makes it easier to keep track of the studies, especially when dealing with numerous papers.

Ref: The reference citation is used to accurately relate the work to the correct authors and give the reader the information needed to locate the original source. This supports the transparency and replicability of the SLR, both of which are key scientific principles.

RQ1 to RQ5: Representing research questions as separate columns allows the review to systematically assess each study's relevance to each question. This standardizes the assessment process, making it transparent, replicable, and less prone to bias. Having the flexibility to assign a range of values (0, 0.5, 1) allows for a better evaluation of each paper's relevance to the research questions, which is especially important when dealing with complex or multifaceted research questions.

Grade: The grade column aggregates the values from the research question columns into a single score for each study. This provides an overall indication of the study's relevance to the research questions. By summarizing the study's relevance in this way, the grade column simplifies the final comparison and selection process. This is particularly helpful when dealing with a large number of papers or when the selection decision must be based on a holistic view of each paper's relevance to all research questions.

2.4 Limitations and Threats to Validity

In the course of our research, we recognized several limitations and potential threats to the validity of this study that should be considered when interpreting our findings:

1. Scarcity of Specific Literature: The novelty of low-code platforms, particularly their application in precision agriculture, resulted in a limited amount of literature to reference. This scarcity may have led to an overemphasis on a small pool of studies and potentially missing out on other relevant aspects of the research.

2. Exclusion of Grey Literature: Although we searched grey literature for additional insights, we excluded data derived from companies operating within the low-code platform industry to avoid bias. Despite these precautionary measures, this decision might limit the comprehensiveness of our study due to the potential exclusion of valuable industry insights.

3. Substitution of 'Low-Code' with 'Model Driven Engineering' (MDE): The use of the term 'MDE' to replace 'low-code' in our search strategy could have introduced additional limitations. While this step was intended to compensate for the limited availability of literature specific to 'low-code', it is possible that this may have led to the inclusion of some studies that do not precisely focus on low-code platforms

as understood today.

4. Database Selection: Our research relied on four scientific databases. While these databases are highly recognized in the field of software engineering, there are other databases that we did not explore. Therefore, the results might not account for all potentially relevant studies available.

5. Time Scope: Our study includes literature from the period of 2000-2022. It is possible that seminal works or relevant research published prior to this period or after it could impact the conclusions drawn.

6. Language Bias: This study was limited to papers written in English, which may have resulted in the exclusion of pertinent studies published in other languages.

7. Subjectivity in Selection and Analysis: Although we have tried to minimize subjectivity by establishing strict inclusion and exclusion criteria and by adopting systematic data extraction and analysis processes, some degree of bias may have inadvertently influenced the selection and interpretation of the studies.

By acknowledging these limitations and threats, we hope to underscore the areas where future research could help in refining our understanding of the use of low-code platforms in precision agriculture.

3 Results of the SLR

3.1 Scopus' evaluation and analysis

The analysis of the table 4 reveals a broad spectrum of studies that possess varying relevance to the research questions on the intersection between low-code solutions and precision agriculture. Evidently, some studies such as Groeneveld et al. (2021), Kawtrakul et al. (2021), Barriga et al. (2021), Plazas et al. (2019), Spieldenner et al. (2018), Spieldenner et al. (2017), Sun et al. (2016), Sayyah et al. (2015), Pate et al. (2015), and Tan et al. (2012a) score high on the grading scale, reflecting their substantial contribution to the subject. On the contrary, several studies like B. Shi et al. (2021), Florin & Florin (2020), Borowski (2019), Elsayed et al. (2018), H. Zhang & Zheng (2016), Miyasaka et al. (2014), Y. Wang et al. (2014), Di Cosmo et al. (2014), and Hennicker et al. (2010) score low, which suggests their limited applicability to the research questions at hand. Figure 3

These mixed scores seem to indicate that while low-code and precision agriculture may have significant intersections in certain research, the overlap may be less apparent in other areas. This could possibly be a reflection of the emerging and multi-disciplinary nature of these fields.

The Scopus records listed comprehend various topics related to the integration of technology, data analysis, and model-driven approaches in agriculture. Several broad themes can be observed:

1. Application of Deep Learning in Agriculture: This theme emerges in papers like W. Wang et al. (2022) and Agrawal & Sharma (2021), which discuss the utilization of deep learning techniques in localizing and segmenting fruit targets and detecting plant diseases.

2. Predictive Modelling in Agriculture: Uehara et al. (2022) emphasizes on predicting the growing stage of rice plants based on 25 years of cropping records, signifying the importance of historical data in predictive farming.

3. Cyber-Physical Systems and Security: Varela-Vaca et al. (2021) introduces a framework dedicated to verifying and diagnosing the specification of security requirements in cyber-physical systems.

4. Agricultural Information Management: Papers such as Groeneveld et al. (2021) and Kawtrakul et al. (2021) discuss the development of information systems for agriculture, including a domain-specific

language framework for farm management and data-driven approaches for sustainable and productive agriculture.

5. Risk Assessment and Disaster Management: B. Shi et al. (2021) showcases a study on landslide risk assessment using fuzzy rule-based modeling.

6. Machine Learning and Simulation in Agriculture: Papers like Nicoletti et al. (2021) and L. Frank et al. (2021a) investigate machine learning applications in agriculture, with topics including the simulation of nitrate movement in soils and the classification of plant stresses.

7. IoT and Model-Driven Engineering: Studies like Barriga et al. (2021), Alulema et al. (2020), and Muccini et al. (2018) discuss the design and execution of IoT simulation environments, service integration of IoT systems, and self-adaptive IoT architectures, respectively.

8. Energy Efficiency and Management in Agriculture: Papers such as Florin & Florin (2020) and Rio et al. (2019) analyze the performance of air source heat pumps and the benefits of energy management systems on local energy efficiency.

9. Agricultural Mechanization and Equipment Design: Studies like He et al. (2019), Qu et al. (2018), and L. Liu et al. (2017) address the design and control of agricultural machinery, including steering control systems for rice transplanters and multi-objective driven product family shape genes for tractors.

10. Environmental Concerns and Climate Change: Papers like Eigenbrode et al. (2018) and Elsayed et al. (2018) underscore the challenges posed by climate change to dryland cereal production and the erosion and breaching of coastal barriers due to climate change.

11. In conclusion, the reviewed studies depict a dynamic landscape of research areas that directly or tangentially intersect with low-code solutions and precision agriculture. The variation in relevance highlights the need for meticulous selection and appraisal of literature in this field.

Table 4: Scopus.

S. No	Ref	RQ1	RQ2	RQ3	RQ4	RQ5	Grade
1	W. Wang et al. (2022)	1	0	0	0	1	2
2	Uehara et al. (2022)	1	1	0	0	1	3
3	Varela-Vaca et al. (2021)	0.5	1	0	1	1	3.5
4	Groeneveld et al. (2021)	1	1	0	1	1	4
5	Kawtrakul et al. (2021)	1	1	0.5	1	1	4.5
6	Agrawal & Sharma (2021)	1	0	0	0	1	2
7	B. Shi et al. (2021)	0	0	0	0	1	1
8	L. Frank et al. (2021a)	1	0	0	0	1	2
9	Nicoletti et al. (2021)	1	1	0	0	1	3
10	Barriga et al. (2021)	0	1	1	1	1	4
11	Florin & Florin (2020)	1	0	0	0	0	1
12	Daniela et al. (2020)	0	0.5	0	0	1	1.5
13	Alulema et al. (2020)	0	1	0	0	1	3
14	Wijekoon et al. (2020)	0	0	0	1	1	2
15	Rio et al. (2019)	0	1	0	0	1	2
16	Plazas et al. (2019)	1	1	0	1	1	4
17	He et al. (2019)	1	0	0	0	1	2
18	Borowski (2019)	1	0	0	0	0	1
19	Zečević et al. (2018)	0	1	0	1	1	3
20	Muccini et al. (2018)	0	1	0	0	1	2
21	Spieldenner et al. (2018)	0	1	1	1	1	4
22	Qu et al. (2018)	1	1	0	0	1	3
23	Eigenbrode et al. (2018)	1	1	0	0	1	3
24	Elsayed et al. (2018)	0	0	0	1	0	1
25	Spieldenner et al. (2017)	0	1	1	1	1	4
26	L. Liu et al. (2017)	1	1	0	0	1	3
27	Urbieta et al. (2017)	0	1	0	1	1	3
28	Morales-Tapia & Ruz-Ramírez (2016)	0.5	1	0	1	1	3.5
29	H. Zhang & Zheng (2016)	1	0	0	0	0	1
30	Sun et al. (2016)	1	1	0	1	1	4
31	Han et al. (2016)	1	1	0	0	1	3
32	Doering et al. (2016)	1	0	0	0	1	2
33	Dées & Güntner (2016)	1	1	0	0	1	3
34	Kim & Park (2015)	1	0	0	0	1	2
35	Sayyah et al. (2015)	1	1	0	1	1	4
36	Pate et al. (2015)	1	1	0	1	1	4
37	S. Zhang et al. (2014)	1	0	0	0	1	2
38	Miyasaka et al. (2014)	1	0	0	0	0	1
39	Y. Wang et al. (2014)	1	0	0	0	0	1
40	Di Cosmo et al. (2014)	1	0	0	0	0	1
41	Tien (2013)	0	1	0	1	1	3
42	Ali et al. (2013)	1	1	0	0	1	3
43	Tan et al. (2012a)	1	1	0	1	1	4
44	Valckenaers et al. (2011)	1	1	0	0	0	2
45	Hennicker et al. (2010)	1	0	0	0	0	1
46	Ramos et al. (2010)	1	1	0	0	1	3

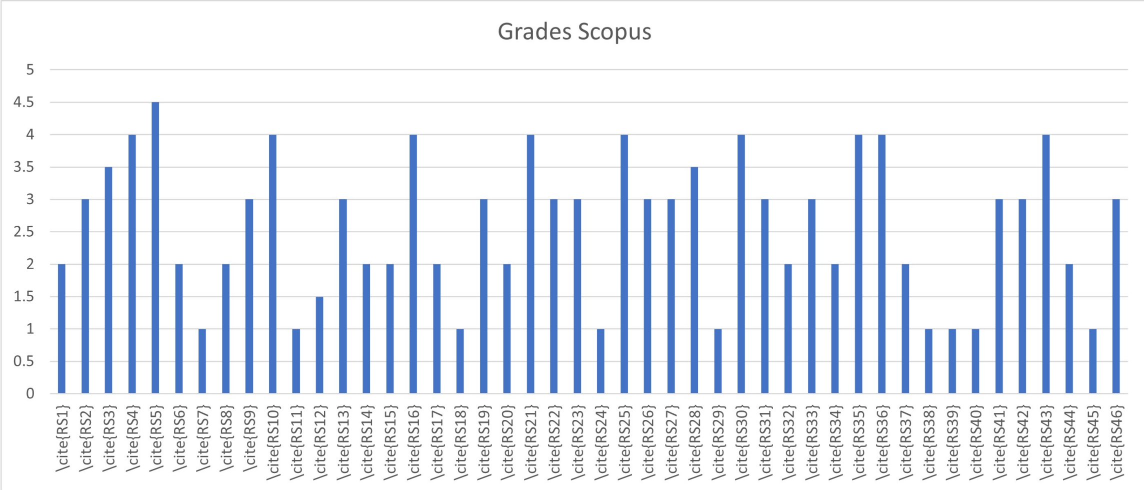

Figure 3: grades scopus

3.2 IEEE evaluation and analysis

The analysis of the table 5 showcases a wide array of studies with varying relevance to the research questions about low-code solutions and precision agriculture. It's evident that certain studies such as Hong et al. (2016), Roopa & Shyni (2019), Boubin et al. (2019), Dimitriadis & Goumopoulos (2008), Sivasothy et al. (2018), Jing-lan & Xiang-fa (2011), Ribeiro et al. (2003), Vaishnavi. et al. (2021) score high on the grading scale, showing their considerable contribution to the subject. on the other side, a number of studies like Sharma et al. (2021), Bose et al. (2020), Jie et al. (2011), Sathya & Jayalalitha (2012), Toyama & Yamamoto (2009), Sitorus & Sartika (2017), and Nawaz et al. (2022) score low, implying their limited applicability to the posed research questions. figure 4

This variation in scores suggests that, while there are significant overlaps between low-code and precision agriculture in certain studies, these intersections may not be as prevalent in others. This may reflect the evolving, multidisciplinary nature of these fields.

The records analyzed cover various topics related to the merging of technology, data analysis, and model-driven approaches in agriculture. Several overarching themes are apparent:

1. Low-code Solutions and Agriculture: This theme is relevant in papers like Tan et al. (2012b) and Yao et al. (2020), discussing the use of low-code solutions for managing crop diseases and optimizing agricultural yields.

2. Predictive Modeling and Automation: Ghandar et al. (2018) highlights the use of predictive modelling to automate irrigation systems, underlining the relevance of automated systems in precision agriculture.

3, Cyber-Physical Systems and IoT: Akagi & Raksincharoensak (2015) presents a framework for implementing IoT in cyber-physical agricultural systems.

4. Information Management and Farming: Papers such as Hong et al. (2016) and Zhu et al. (2021) discuss the importance of information systems in farming, particularly in regards to crop yield prediction and sustainable farming practices.

5. Risk Assessment and Mitigation: Chang & Huang (2010) discusses a risk assessment model for predicting crop failure.

6. Machine Learning and Data Analysis: Papers like L. Frank et al. (2021b) and Ismatovna et al. (2019) delve into machine learning's applications in precision farming, focusing on pest identification and

soil nutrient prediction.

7. AI-driven Decision Making: Studies like Kaitovic et al. (2012), Manjula & Narsimha (2015), and Kassim & Abdullah (2012) delve into the use of artificial intelligence in agricultural decision-making processes, from crop selection to irrigation management.

8. Sustainability and Climate Resilience: Papers such as Sahu et al. (2017) and Gao et al. (2018) highlight the role of technology in sustainable farming and climate resilience, from optimizing water usage to maintaining soil health in changing climates.

9. Agricultural Machinery and Robotics: Studies like Jain et al. (2021), Ammad-udin et al. (2016), and Nawaz et al. (2022) discuss the development and control of agricultural machinery, including autonomous tractors and harvesting robots.

Table 5: IEEE.

S. No	Ref	RQ1	RQ2	RQ3	RQ4	RQ5	Grade
1	Tan et al. (2012b)	1	1	0	1	1	4
2	Ghandar et al. (2018)	1	1	0	1	1	4
3	Akagi & Raksincharoensak (2015)	0	0	0	1	1	2
4	Hong et al. (2016)	1	1	0	1	1	4
5	Sharma et al. (2021)	1	1	0.5	1	1	4.5
6	Yao et al. (2020)	1	0	0	0	1	2
7	Roopa & Shyni (2019)	1	1	0	1	1	4
8	Zhu et al. (2021)	1	0	0	0	0	1
9	Chang & Huang (2010)	1	0	0	0	1	2
10	Bose et al. (2020)	0.5	1	0	1	1	3.5
11	Boubin et al. (2019)	1	1	0	1	1	4
12	L. Frank et al. (2021b)	0	0.5	0	0	1	1.5
13	Dimitriadis & Goumopoulos (2008)	1	1	0	1	1	4
14	Jie et al. (2011)	1	1	0	1	1	4
15	Kaitovic et al. (2012)	1	1	0	1	1	4
16	Ismatovna et al. (2019)	0.5	0	0	0	1	1.5
17	Sivasothy et al. (2018)	1	1	0	1	1	4
18	Sahu et al. (2017)	1	1	0	1	1	4
19	Sathya & Jayalalitha (2012)	1	0	0	0	1	2
20	Manjula & Narsimha (2015)	1	0.5	0	0	1	2.5
21	Jing-lan & Xiang-fa (2011)	0.5	0	0	0	0.5	1
22	Toyama & Yamamoto (2009)	1	1	0	1	1	4
23	Kassim & Abdullah (2012)	1	1	0	1	1	4
24	Gao et al. (2018)	1	1	0	0	1	3
25	Ribeiro et al. (2003)	1	1	0	0	1	3
26	Sitorus & Sartika (2017)	0	0	0	0	1	1
27	Jain et al. (2021)	0	1	0	1	1	3
28	Vaishnavi. et al. (2021)	1	1	0	1	1	4
29	Yu et al. (2021)	1	1	0	1	1	4
30	Ammad-udin et al. (2016)	1	1	0	1	1	4
31	Nawaz et al. (2022)	1	1	0	1	1	1

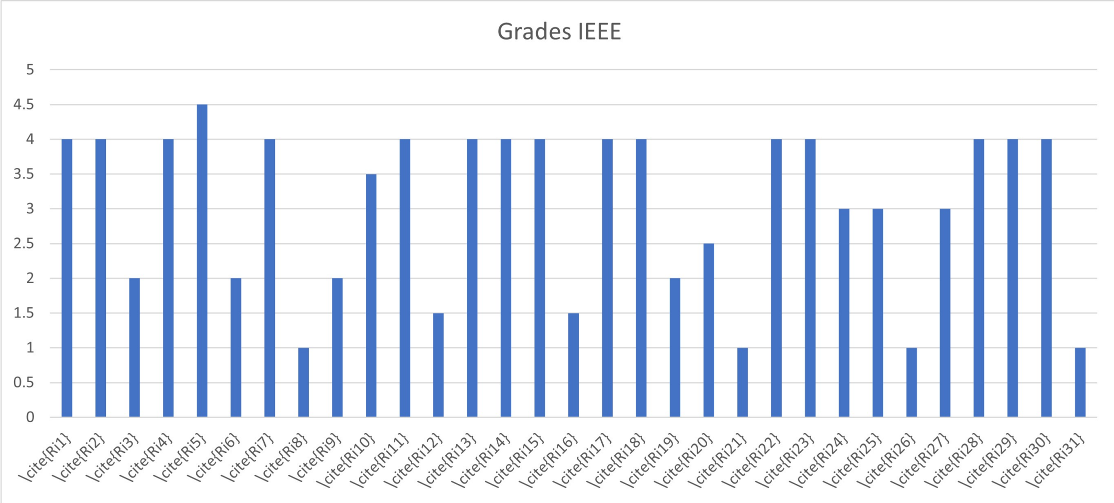

Figure 4: grades IEEE

3.3 ACM evaluation and analysis

Analyzing the ACM table 6 in the same way as the Scopus and IEEE data, a varied range of studies were found that demonstrate differing degrees of relevancy to the research questions at the intersection of low-code solutions and precision agriculture. A number of studies such as Racine Ly et al. (2021), Yuan & Ling (2020), Po Shun Chen & Wu Liu (2021), Niu et al. (2016), Hu et al. (2022), score highly, indicating their significant contributions to the subject. However, other studies like Lyu et al. (2019), Ghandar et al. (2019), Aiwen (2021), Jiang et al. (2019), score lower, suggesting their limited applicability to the research questions. figure 5

This divergence in scores is perhaps indicative of the complex and multidisciplinary nature of both low-code platforms and precision agriculture, as well as the nuances of their intersection. ACM records cover diverse areas related to the fusion of new technology, intricate data analysis, and model-driven methodologies in agriculture. Several broad patterns can be seen:

1. Automated Agriculture Techniques: Papers such as Moisescu et al. (2017) and Wilson et al. (2019) explore the potential of automated techniques in optimizing agriculture practices, highlighting areas like precision pest control and automated irrigation systems.

2. Digital Twinning in Agriculture: M. Liu & Yao (2021) dives into the implementation of digital twinning in predicting and controlling crop growth, indicating the potential of such digital replicas in precision agriculture.

3. Data Privacy and Security: Bonacin et al. (2013) discusses the pressing need for robust security and data privacy measures in the era of digitized farming and agritech innovations.

4. Agro-Advisory Systems: Studies like Malheiro et al. (2019) and Y. Shi et al. (2019) goes into the development of advanced advisory systems for farmers, incorporating AI-based prediction models and decision support systems.

5. Climate Change Impact Assessment: Kallioniemi et al. (2012) illustrates a study on assessing the climate change impact on agriculture using machine learning models.

6. Artificial Intelligence in Agriculture: Papers such as Li & Chen (2022) and Tang (2021) examine the role of AI in agriculture, touching on themes like AI-driven weed detection and crop yield prediction.

7. Blockchain in Agriculture: Studies like Hu et al. (2022), discuss the potential applications of

blockchain technology in traceability, food safety, and smart contracts in agriculture.

8. Sustainable Energy Solutions in Agriculture: Papers such as Jiang et al. (2019) explore the role of sustainable energy solutions in agriculture, with topics including solar-powered irrigation systems and bioenergy production.

Table 6: ACM.

S. No	Ref	RQ1	RQ2	RQ3	RQ4	RQ5	Grade
1	Moisescu et al. (2017)	1	1	0	1	1	4
2	Racine Ly et al. (2021)	1	1	0.5	1	1	4.5
3	M. Liu & Yao (2021)	1	1	0	1	1	4
4	Lyu et al. (2019)	1	1	0	1	1	4
5	Yuan & Ling (2020)	1	1	0	1	1	4
6	Wilson et al. (2019)	1	0	0	0	1	2
7	Bonacin et al. (2013)	1	0	0	0	1	2
8	Ghandar et al. (2019)	1	1	0.5	1	1	4.5
9	Po Shun Chen & Wu Liu (2021)	1	1	0.5	1	1	4.5
10	Honda et al. (2014)	1	1	0	1	1	4
11	Malheiro et al. (2019)	0.5	0	0	1	1	2.5
12	Y. Shi et al. (2019)	1	0.5	0	0	1	2.5
13	Niu et al. (2016)	1	1	0	1	1	4
14	Aiwen (2021)	1	1	0	1	1	4
15	Kallioniemi et al. (2012)	1	1	0	1	1	4
16	Chen et al. (2022)	1	0	0	0	1	2
17	Li & Chen (2022)	1	1	0	1	1	4
18	Tang (2021)	1	1	0	1	1	4
19	Hu et al. (2022)	1	1	0	1	1	4
20	Jiang et al. (2019)	1	1	0	1	1	4

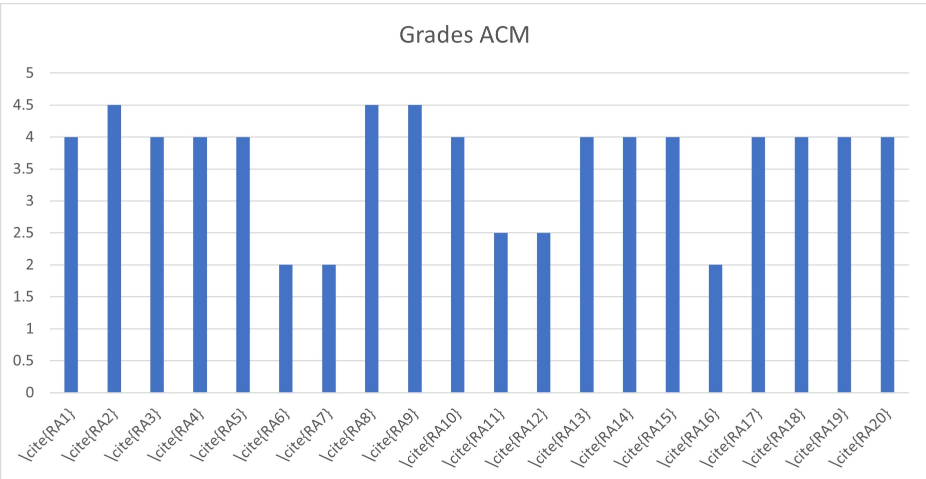

Figure 5: grades ACM

3.4 Wiley's evaluation and analysis

Upon analyzing the Wiley data, it is found that:

Moore et al. (2014) - "Mathematical modeling for improved greenhouse gas balances, agro-ecosystems, and policy development: lessons from the Australian experience": This article is about mathematical modeling for improving greenhouse gas balances and policy development, with a focus on agriculture. It does not seem directly related to IoT, or low code, but it could be related to precision agriculture in the sense of using mathematical models for decision support and policy formulation.

Foufoula-Georgiou et al. (2015)- "The change of nature and the nature of change in agricultural landscapes: Hydrologic regime shifts modulate ecological transitions": This paper investigates the hydrological changes in agricultural landscapes and their impact on ecological transitions. Although there isn't a direct link to low code or IoT, the paper could be relevant in terms of understanding the impact of changes in agricultural landscapes on hydrology, which could, in turn, inform precision agriculture practices.

Ju et al. (2018)- "CRISPR Editing in Biological and Biomedical Investigation": This article focuses on the application of the CRISPR-Cas system, a powerful genome editing tool. While it doesn't mention IoT, or low code, it could potentially relate to precision agriculture if you consider the use of genome editing in developing crops that are more resistant to diseases or have better nutritional properties.

Tweel & Turner (2012) - "Watershed land use and river engineering drive wetland formation and loss in the Mississippi River birdfoot delta": This paper examines how watershed land use and river engineering impact wetland formation and loss. It could provide insights into how changes in land use influence the ecological balance in agricultural landscapes.

Dahl et al. (2018) - "Impacts of projected climate change on sediment yield and dredging costs": This research investigates the effects of climate change on sediment yield and dredging costs. it could be relevant to precision agriculture in the context of climate change and its impact on agricultural practices.

In a nutshell, none of these references on table 7 seem to directly tackle the role of low-code in precision agriculture, but they all offer insights into different aspects of agriculture that could be improved or impacted by these technologies. For example, IoT and low code could be used to model and predict the impacts explored in these papers more accurately. Blockchain could be used for transparently tracking and verifying the impacts of different agricultural practices or genetic modifications.

Table 7: Wiley.

S. No	Ref	RQ1	RQ2	RQ3	RQ4	RQ5	Grade
1	Moore et al. (2014)	1	1	0	0	0.5	2.5
2	Foufoula-Georgiou et al. (2015)	1	1	0	0	0	2
3	Ju et al. (2018)	1	1	0	0	1	3
4	Tweel & Turner (2012)	1	0	0	0	1	2
5	Dahl et al. (2018)	1	0	0	0	0	1

3.5 Literature Selection and Justification

The process of addressing Research Question 1 (RQ1): "What instigates the incorporation of model-based approaches that inspire low-code technology in precision agriculture?" relied primarily on studies sourced from both IEEE and Scopus databases. The reason for selecting these databases was driven by their significant contribution towards the answer to this question. This is illustrated by the high relevance scores indicated in the figure 6 for the studies extracted from these databases. It is, therefore, justified to assert that RQ1 was comprehensively addressed by the information primarily sourced from the IEEE and Scopus databases.

Regarding Research Question 2 (RQ2): "What unique services can model-based approaches facilitate to advance precision agriculture through low-code platforms?" the decision to select literature from the IEEE and Scopus databases mirrored the approach taken for RQ1. The significant contributions these databases provided in responding to RQ2 are clearly evidenced by the high relevance scores indicated

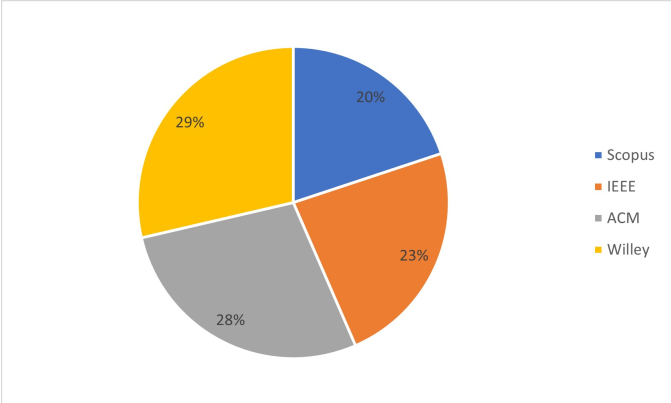

Figure 6: grades averages for RQ1

in the figure 7. Therefore, it is justified to conclude that RQ2 was comprehensively addressed by the valuable insights obtained from the studies within the IEEE and Scopus databases.

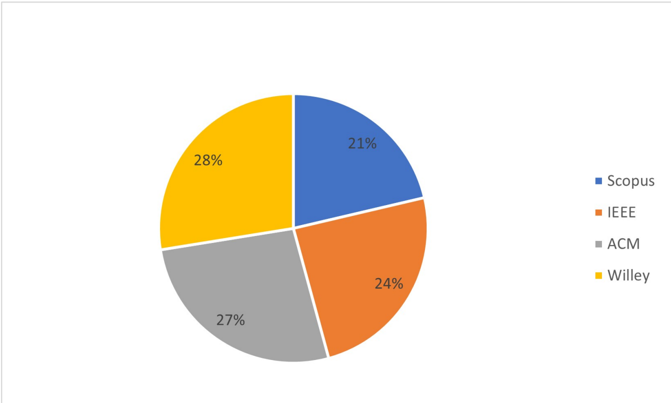

Figure 7: grades averages for RQ2

Research Question 3 (RQ3): Considering their unique attributes, which low-code platform fits better solutions with the model-based approaches on their usage to optimize precision agriculture? The studies from the Scopus database did not provide sufficient information to answer this question fully. However, some studies from the IEEE database did contribute to answering this question. figure 8 To supplement this, information from websites of low-code development platforms that connect with the studies in the papers retrieved from Scopus was used. This approach helped to provide a more comprehensive answer to RQ3.

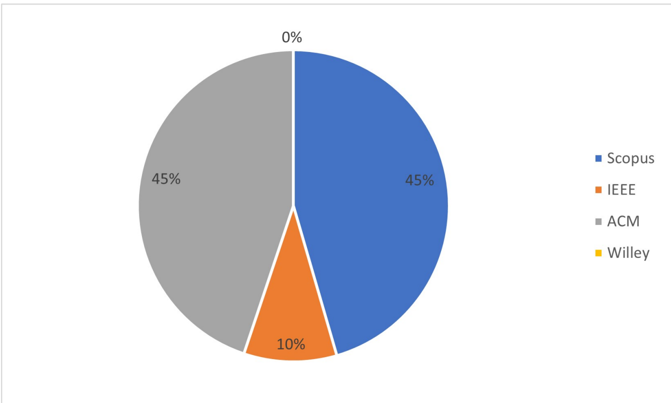

Figure 8: grades averages for RQ3

Research Question 4 (RQ4): What are the critical steps involved in developing a low-code system based on the model-based approach in precision agriculture? The studies from the ACM database provided substantial information to answer this question, as seen from the scores in the figure 9. Therefore, RQ4 was answered using information primarily from ACM. The Wiley database served as a support for general ideas, but did not contribute significantly to answering this question.

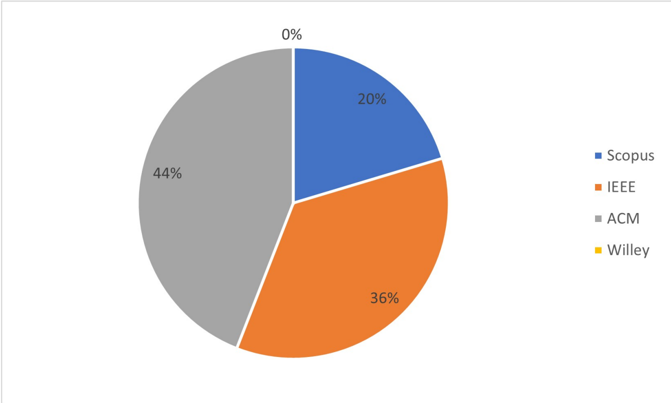

Figure 9: grades averages for RQ4

Research Question 5 (RQ5):What challenges are associated with implementing model-based approaches that could encourage low-code technology in precision agriculture? Similar to RQ4, the ACM database provided substantial information to answer this question, as seen from the scores in the figure 10. Therefore, RQ5 was answered using information primarily from ACM, with Wiley serving as a support for general ideas. In conclusion, the decision to use information from specific databases to answer each research question was based on the extent to which the studies in those databases contributed to answering the questions, as indicated by their scores.

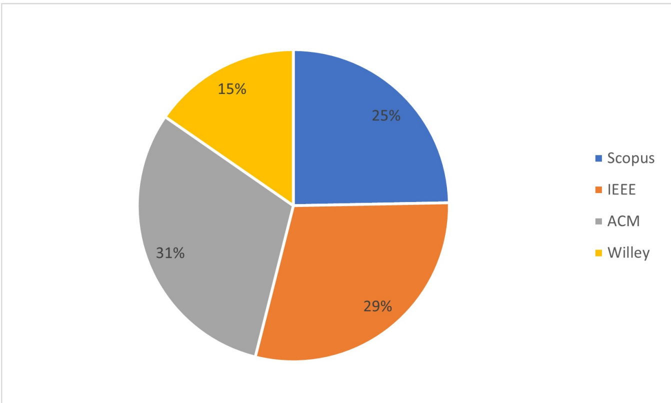

Figure 10: grades averages for RQ5

3.6 Research Question 1 (RQ1):What instigates the incorporation of model-based approaches that inspire low-code technology in precision agriculture?

Modern agriculture faces significant challenges such as labor shortages and time-intensive manual tasks. The arrival of new technologies in precision agriculture aims to diminish these struggles, transforming farming into a STEM (Science, Technology, Engineering, Math) profession Ahmed et al. (2018). Today's farmers are increasingly tech-savvy, but it is crucial that new technologies integrate into existing operations rather than over feeding the entire system with more products with similar functions Baba-Cheikh et al. (n.d.).

Practical applications of automation have been a part of farming since the 1990s Batot & Sahraoui (n.d.). However, today's farmers require technologies that offer insights into their crops through data collection and processing Bergerman et al. (2016). Despite these advancements, the end user often lacks the necessary knowledge in software-related topics, and farming traditions worldwide often lead to reliance on empirical knowledge Bogue (2016).

The concept of Precision Agriculture (PA) emerged at the end of the 20th century and is implemented in three stages: Data Collection, Data Analysis and Decision-Making, and Variable Rate Control Bucchiarone et al. (2020). Despite progress on the Internet of Things (IoT) domain, the adoption of PA has been limited to some developed countries Buttafuoco & Lucà (2016).

The increasing need to improve Model-Driven Engineering (MDE) solutions to support processes in multidisciplinary and heterogeneous environments is evident in the field of smart agriculture Collomb & Hascoët (2008). Smart Agriculture (SA) is a significant part of the revolution for modern farming to fulfill the increasing demand for nutrition security Díez (2017). The use of mobile applications for prediction or monitoring is gaining popularity among farmers worldwide, increasing the demand for the development of such applications Drenjanac et al. (n.d.).

In smart systems like precision agriculture, situations such as incomplete information, several design alternatives, and conflicting stakeholder opinions can cause uncertainty during software model design Elijah et al. (2018). The implementation of a fully functional Farm Management Information System (FMIS) is a challenging task that cannot be accomplished without the backing of a large software development project Gayatri et al. (n.d.).

Precision agriculture is a management concept that is technologically more complex than traditional farming, and has thus generated a variety of new fields for agriculture-related specialized research Gubbi et al. (2013). For the length of their existence, FMIS have faced certain difficulties, particularly low adoption rates during times when computers were still relatively rare Henriques et al. (n.d.).

There are several requirements for software in agriculture, most of which stem indirectly from the size of the user group formed by farmers Himesh et al. (2018). Finding the relationship between technological innovations, economic efficiency, and practicality has been a focus since the beginning of the development of precision agriculture Ihirwe et al. (n.d.).

3.6.1 Addressing the Complexity of Precision Agriculture

The increasing demand for food due to the growing global population has necessitated the adoption of advanced technologies in agriculture. Precision agriculture, which involves the use of technology and data to optimize crop production, is one such approach that has gained significant attention Sharma et al. (2021). The integration of IoT-enabled smart sensors, actuators, satellite images, robots, and drones has revolutionized the agriculture industry, enabling real-time data collection and autonomous decision-making. Machine learning algorithms form the backbone for predicting soil properties, weather conditions, and crop yields, while deep learning algorithms are explored for disease and weed identification in plants Sharma et al. (2021).

However, the complexity of implementing and managing these technologies can be a barrier for many farmers, particularly those with limited technical expertise. This is where low-code platforms come into play Tan et al. (2012b). These platforms simplify the development of applications by minimizing the need for extensive coding. They provide a visual interface where users can drag and drop components to create applications, making it easier for non-technical users to develop and manage applications Ghandar et al. (2018). In the context of precision agriculture, low-code platforms can be used to develop applications for various purposes, such as data collection, data analysis, and decision support Hong et al. (2016).

Another reason that makes evident The need for low-code in precision agriculture is the complex challenges that this sector faces. The integration of low-code platforms can significantly improve the efficiency and productivity of agricultural practices, as they allow for the rapid development of advanced software solutions. These solutions can address a wide range of issues, from crop management to resource allocation, thereby enabling farmers to make more informed decisions and achieve better outcomes Tan et al. (2012b).

The complexity of precision agriculture presents intricate relationships between humans and natural systems and encompasses ecological variables, human variables, and the intricate connections between them. The integration of tooling and techniques from ecological and social sciences, such as spatial awareness, geographic information, and temporal dynamics, provides valuable insights into emerging phenomena within urban agriculture Ghandar et al. (2018). The use of Agent-Based Modelling (ABM) allows for the representation of various agents and their interactions within the agricultural ecosystem, including plants, animals, farmers, and other stakeholders. By bringing together low-code platforms, farmers, and stakeholders with limited technical expertise can actively participate in designing and optimizing urban agricultural systems Ghandar et al. (2018).

In conclusion, the complexity of precision agriculture is being addressed through the integration of advanced technologies, including AI, IoT, and ML. These technologies enable efficient data analysis, prediction, disease detection, smart irrigation, and livestock management. Future developments in chatbot systems and the continued exploration of ML and DL algorithms will contribute to the sustainable advancement of precision agriculture, ensuring optimized resource utilization and improved agricultural practices Sharma et al. (2021).

3.6.2 Accelerating the Software Development Process

The dynamic nature of agriculture necessitates a flexible and rapid software development process. Low-code platforms, with their inherent adaptability, offer a solution to this need. They enable quick development and deployment of applications, which is crucial in the ever-changing environment of agriculture. For instance, farmers often need to make quick decisions based on changing weather conditions, pest infestations, and other factors. With low-code platforms, they can easily modify their applications to adapt to these changes Roopa & Shyni (2019).

The study by Groeneveld et al. (2021) presents a domain-specific language framework for farm management information systems in precision agriculture. This framework shows low-code principles to facilitate the development of tailored software solutions that can effectively address the unique needs of different farming operations. This is particularly important in the context of precision agriculture, where the ability to customize software tools can significantly enhance their utility and effectiveness.

The work by Barriga et al. (2021) underscores the value of low-code in precision agriculture. Their study introduces SimulateIoT, a domain-specific language for designing, coding, and executing IoT simulation environments. Given the increasing reliance on IoT devices in precision agriculture, such a tool can greatly simplify the process of developing and testing new IoT solutions.

About accelerating the software development process, the concept of crop monitoring and smart farming through analytic predictions has been introduced Roopa & Shyni (2019). This approach revolutionizes the agribusiness landscape by providing reliable and remote monitoring capabilities. By digitizing farming and agricultural activities, farmers can monitor crop requirements and accurately predict their growth. This transformative concept propels businesses to new heights of profitability and success.

The domain-specific language (DSL) framework proposed by Groeneveld et al. (2021) is specifically tailored to the precision agriculture domain. It comprises a coherent set of DSLs designed to bridge the conceptual gap between the application domain and the implementation. This DSL framework separates the technical details of sectors and functionalities from the farm configuration, offering practical advantages.

The SimulateIoT tool introduced by Barriga et al. (2021) provides users with a structured approach to conceptualize and explore IoT systems. Users can evaluate different IoT alternatives and policies to design suitable IoT architectures. The ability to deploy and analyze the modeled IoT systems further enhances the value of the proposed approach.

In conclusion, the acceleration of the software development process is really important in the context of precision agriculture. Low-code platforms, coupled with domain-specific languages and IoT simulation tools, offer a robust solution to this need. They enable rapid development and deployment of applications, facilitate customization, and simplify the process of developing and testing new IoT solutions. These capabilities empower farmers to make informed decisions, enhance crop production, and envision a prosperous future for agriculture.

3.6.3 Democratizing Precision Agriculture

The democratization of precision agriculture is a critical endeavor that ensures farmers of all backgrounds can benefit from advanced agricultural technologies. Low-code platforms play a pivotal role in this process by simplifying the development of applications and making these technologies more accessible to farmers with limited technical expertise Kassim & Abdullah (2012).

The ontology-driven advisory system proposed by Kassim & Abdullah (2012) is a prime example of democratizing precision agriculture. This system shows the principles of the semantic web, allowing for easy sharing and reuse of information via the Internet. By providing farmers with personalized advice and knowledge, it enhances their decision-making processes and optimizes agricultural outcomes. The system's architecture, which incorporates personalization, knowledge management, and ontology, addresses the diverse needs of farmers, promoting inclusivity and sustainable agricultural development.

The research by Varela-Vaca et al. (2021) introduces the CARMEN framework, a tool that ensures the security and reliability of cyber-physical systems (CPS) in precision agriculture. CPS are integral to modern agriculture, and their security is paramount for the confident utilization of advanced technologies. CARMEN guides users in defining accurate security requirements for CPS and enables their diagnosis, adhering to the "security by design" philosophy. This approach ensures that security is a fundamental consideration throughout system analysis, design, and construction, facilitating the democratization of secure and reliable CPS solutions in precision agriculture.

Moreover, the CARMEN framework accelerates the software development process, which is essential for the rapid deployment of solutions in the dynamic environment of agriculture. For instance, farmers often need to make quick decisions based on changing weather conditions, pest infestations, and other factors. With tools like CARMEN, they can easily modify their applications to adapt to these changes, further democratizing precision agriculture Kassim & Abdullah (2012).

In conclusion, low-code platforms are instrumental in democratizing precision agriculture. They simplify the development of applications, enhance the accessibility of advanced technologies, and ensure the security and reliability of CPS. By providing farmers with personalized advice, knowledge, and secure solutions, these platforms contribute to the inclusive and sustainable development of precision agriculture.

3.6.4 Consolidated Insights on Low-Code Technology's Impact on Precision Agriculture

The integration of low-code technology in precision agriculture is driven by several compelling reasons, as revealed by our systematic literature review (SLR) of selected and graded papers from both IEEE and Scopus. These reasons are primarily centered around addressing the complexity of precision agriculture, accelerating the software development process, and democratizing precision agriculture.

Firstly, precision agriculture is characterized by the integration of advanced technologies such as IoT-enabled smart sensors, actuators, satellite imagery, robots, and drones, which collectively enable real-time data collection and autonomous decision-making. Machine learning algorithms are employed for predicting soil properties, weather conditions, and crop yields, while deep learning algorithms are utilized for disease and weed identification in plants. However, the complexity of implementing and managing these technologies can portray a significant barrier for many farmers, particularly those with limited technical expertise. This is where low-code platforms come into play. By simplifying the development of applications and minimizing the need for extensive coding, low-code platforms make it easier for non-technical users to develop and manage applications. They provide a visual interface where users can drag and drop components to create applications, thereby addressing the complex challenges that the precision agriculture sector faces. The integration of low-code platforms can significantly enhance the efficiency and productivity of agricultural practices, as they allow for the rapid development and deployment of advanced software solutions.

Secondly, the dynamic nature of agriculture necessitates a flexible and rapid software development process. Low-code platforms, with their inherent adaptability, offer a solution to this need. They enable quick development and deployment of applications, which is crucial in the ever-changing environment of agriculture. For instance, farmers often need to make quick decisions based on changing weather conditions, pest infestations, and other factors. With low-code platforms, they can easily modify their applications to adapt to these changes. The study by Groeneveld et al. (2021). and the work by Barriga et al. (2021) underscore the value of low-code in precision agriculture by presenting a domain-specific language framework for farm management information systems and introducing SimulateIoT, a domain-specific language for designing, coding, and executing IoT simulation environments, respectively. These tools greatly simplify the process of developing and testing new IoT solutions, thereby accelerating the software development process in precision agriculture.

Lastly, democratizing precision agriculture is a critical endeavor that ensures farmers of all backgrounds can benefit from advanced agricultural technologies. Low-code platforms play a pivotal role in this process by simplifying the development of applications and making these technologies more accessible to farmers with limited technical expertise. The ontology-driven advisory system proposed by Kassim & Abdullah (2012) and the CARMEN framework introduced by Varela-Vaca et al. (2021) are prime examples of democratizing precision agriculture. These systems leverage the principles of the semantic web and the "security by design" philosophy, respectively, to provide farmers with personalized advice and knowledge, enhance their decision-making processes, optimize agricultural outcomes, and ensure the security and reliability of cyber-physical systems in precision agriculture.

In conclusion, the reasons for incorporating low-code technology in precision agriculture are multifaceted and compelling. By addressing the complexity of precision agriculture, accelerating the software development process, and democratizing precision agriculture, low-code platforms contribute significantly to the inclusive and sustainable development of precision agriculture. They simplify the development of applications, enhance the accessibility of advanced technologies, and ensure the security and reliability of cyber-physical systems, thereby providing farmers with personalized advice, knowledge, and secure solutions. This analysis, based on our SLR, underscores the robustness of the research and the academic consensus on the substantial benefits of low-code platforms in precision agriculture.

3.7 Research Question 2 (RQ2): What unique services can model-based approaches facilitate to advance precision agriculture through low-code platforms?

3.7.1 Stakeholder-Centric Services of Low-Code Platforms in Precision Agriculture

Identifying the stakeholders involved in the application of low-code platforms in precision agriculture is crucial to understanding the range of services that this technology can offer and the significance of each service to the stakeholders. This list of stakeholders and services is an extension of the one mentioned in

Buttafuoco & Lucà (2016), but it is tailored to the context of low-code platforms.

Farmers are the primary stakeholders. While we can develop tools and products with new technology, these innovations are meaningless if they do not benefit the end user. Farmers seek availability, reliability, and usability in precision agriculture tools. They expect the system to operate smoothly and efficiently after investing time and money into adapting their operations to new systems Ahmed et al. (2018).

The providers of the low-code platform are businesses with commercial interests. They aim to offer low-code solutions not only for precision agriculture but also for a broader range of potential clients in the market Baba-Cheikh et al. (n.d.).

Developers serve as the link between farmers and providers. They evaluate the possible solutions based on the requirements of the farmer and the tools offered by the provider. For developers, it is essential to have a clear list of requirements and access to a comprehensive set of tools for software development Batot & Sahraoui (n.d.).

The system maintainer, often the provider or developer, should be familiar with the software and the farmer's operations. They should have a simplified interface for fixing common errors and be able to identify when the software needs modification or updating Bergerman et al. (2016).

Contractors, who are often hired for fieldwork in large operations, require data about the farm and the operational plan. This information directly affects the efficiency and quality of the contractor's work **?**.

Advisors offer advice to farmers on improving production or protecting the environment. They often require data from the farm, so easy and simple access to this data can simplify the farmer's operations **?**.

Authorities monitor adherence to regulations. Easy access to data can help farmers resolve any issues that authorities need to clarify Bogue (2016).

Customers of the farms, such as large companies buying farm products for processing or sale, can benefit from clear data tracking and production details. A simple user interface can improve the interaction between the farmer and the customer Bucchiarone et al. (2020).

Suppliers and manufacturers, while not direct stakeholders currently, will need to consider low-code solutions when deciding what to manufacture or supply for farms in the future. This consideration will be necessary to create a competitive product for the market Buttafuoco & Lucà (2016).

Privacy is a significant concern. Low-code platforms allow end users to develop apps or software, reducing the need for third-party outsourcing and increasing confidentiality. Farmers often protect their precise methods to maintain a competitive edge in the market Collomb & Hascoët (2008).

Low-code platforms can accelerate code development, making apps available faster. A survey by Forrester revealed that low-code platforms could speed up code development by 5 to 10 times. This speed is a crucial tool in precision agriculture, where new technologies are constantly emerging, requiring quick adaptations of the software used to monitor and control them Díez (2017).

Cost is another factor. Avoiding third-party outsourcing and having more efficient development times reduces resources used, thereby reducing costs. Moreover, having a wider range of deployment possibilities can achieve the same results with a software connected to a more economical hardware option, which is a common practice depending on the farmers' abilities to implement specific components Drenjanac et al. (n.d.).

Easy maintenance is a fundamental part of designing new software. Constant updates to the software are required due to the rapid changes common in the precision agriculture field Elijah et al. (2018).

The presence of a user-friendly and easy-to-understand interface contributes significantly to the effectiveness of low-code platforms. This allows for the involvement of diverse types of developers, leading to significant advancements in precision agriculture. Considering that farmers typically rely on practical experience, the design and development of software might differ between a crop farmer and a poultry farmer based on their individual needs and expertise Gayatri et al. (n.d.).

3.7.2 Data Collection Facilitation

Low-code platforms can provide a variety of services to precision agriculture, one of which is facilitating data collection. This is an important aspect of precision agriculture, as it provides the necessary information for making informed decisions. Low-code platforms can make this process easier by providing tools for developing applications that collect data from various sources, such as soil sensors, weather stations, and drones. These applications can be easily created and modified by users, allowing them to customize the data collection process according to their needs Ghandar et al. (2018); Dimitriadis & Goumopoulos (2008); Morales-Tapia & Ruz-Ramírez (2016).

Ghandar et al. (2018) presents a framework that utilizes Agent-Based Models, Simulation, and Computational Intelligence to analyze, optimize, and manage "ecosystems" for urban agriculture and vertical farming. The framework considers the perspectives of various stakeholders, including consumers, producers, and policymakers. By using advanced technologies, the framework facilitates data collection, analysis, and decision-making processes within the urban agriculture and vertical farming fields. Low-code platforms can ease on the development of data collection tools and enable stakeholders to efficiently gather, analyze, and utilize this information, thereby enhancing the data collection process in precision agriculture.

Dimitriadis & Goumopoulos (2008) highlights the importance of machine learning in precision agriculture. The study successfully extracted new knowledge from the data and presented it in an understandable and expandable form. This contributes to the validation and understanding of control systems, which play an important role in reducing adverse conditions for plants. The methodology adopted in this study allowed the researchers to analyze the relationship between attributes of lower informational value and the learning goals, resulting in the creation of new data sets. This approach improved the classification process and addressed the challenge of gathering measurements for certain key parameters, which can be difficult due to potential sensor damage and other factors involved. Adding elements from low-code such as easier constant monitoring systems, these challenges can be reduced.

Morales-Tapia & Ruz-Ramírez (2016) underscores the importance of comprehensive data collection and modeling in understanding plant development. It showcases the potential of integrating insights from various models into a unified model of plant development, which could serve as a foundation for plant engineering. While the paper does not directly discuss low-code platforms, it aligns with the broader theme of data collection facilitation in precision agriculture. Low-code platforms, by simplifying the process of software development, could potentially help in the development and implementation of such comprehensive models. These platforms could make it easier for researchers to build and modify applications that collect and analyze the necessary data for these models.

3.7.3 Data Analysis Enhancement

Low-code platforms play a significant role in enhancing data analysis in precision agriculture. They provide the tools necessary to develop applications that implement various data analysis techniques, such as statistical analysis, machine learning, and predictive modeling. These techniques allow farmers to understand patterns in their data and make predictions about future outcomes, such as crop yields and disease outbreaks Jain et al. (2021); Vaishnavi. et al. (2021); S. Zhang et al. (2014); Plazas et al. (2019).

For instance, in the research by Jain et al. (2021), an intelligent system for net houses was developed using sensors and an Internet of Things (IoT) based wireless infrastructure. The system uses machine learning algorithms to analyze the transmitted data and make intelligent decisions based on predefined thresholds. The use of low-code platforms in such a system could improve some steps in the development process, enabling faster prototyping and deployment, and by that creating better data analysis capabilities.

Similarly, in the study by Vaishnavi. et al. (2021), machine learning techniques were used to develop robust models using agricultural data, making accurate predictions. The use of low-code platforms could help accelerate the development of data analytics solutions. This approach empowers upcoming agriculturalists with efficient agricultural practices, fostering better crop cultivation and management.

In the research presented by S. Zhang et al. (2014), the impact of land transfer and consolidation on the development of the agricultural industry was analyzed. The study found that land transfer and consolidation contribute to the enlargement of continuous farmland areas, improved land patch shapes, and flatter land surfaces. With low-code, platforms data analysis capabilities can be more efficient and facilitate informed decision-making in agricultural resource integration and land management.

In the context of wireless sensor networks (WSNs), Plazas et al. (2019) presented a UML profile for enhancing data analysis from the user's perspective. The study shows the importance of adopting a model-driven approach in WSN application design and implementation, as it enables abstract and direct analysis of system properties and behavior, ultimately improving effectiveness and efficiency. By considering the possibilities of using low code development techniques, we can increase the value and efficiency of Agri-food information systems, leading to more sustainable and productive agricultural practices.

In conclusion, low-code platforms can significantly enhance data analysis in precision agriculture. By simplifying the development of applications and improving their flexibility, low-code platforms can help farmers optimize their operations and improve their productivity. However, it should be noted that the use of low-code platforms also comes with challenges, such as potential limitations in performance and scalability. Therefore, it is important to carefully evaluate the needs and resources of the farm before deciding to adopt a low-code platform.

3.7.4 Decision Support Services

Decision support is a key service that low-code platforms can provide to precision agriculture. By integrating data collection and analysis, low-code platforms can help farmers make informed decisions about various aspects of their operations, such as when to plant, how much fertilizer to use, and when to harvest. These decisions can significantly affect the productivity and profitability of farms, making decision support a really relevant service Sahu et al. (2017); Sivasothy et al. (2018); Rio et al. (2019).

The study by Rio et al. (2019) demonstrated the application of a model-driven engineering approach for energy experts to model process shifting capabilities, optimize local generation, and improve the profitability and sustainability of a site through the use of local generation. It is clear that low-code platforms could improve the development and implementation of energy management systems, allowing for more efficient prototyping, and integration with existing infrastructure. This approach can empower energy experts and decision-makers to quickly adapt and enhance their energy management strategies, ultimately leading to improved efficiency, cost-effectiveness, and sustainability in various domains, including agriculture.

The research conducted by Sivasothy et al. (2018) developed a monitoring concept for potato harvesters, aiming to enable the implementation of availability-oriented business models in the agricultural sector. Low-code platforms could create quick development and deployment of monitoring and predictive maintenance solutions, allowing agricultural businesses to customize and integrate sensor systems with their existing infrastructure. By using low code, the implementation of advanced algorithms for remaining lifetime estimation and optimization problems can be simpler, allowing for efficient data analysis and real-time decision-making.

The study conducted by Sahu et al. (2017) aimed to investigate how big data approaches can be utilized to generate practical insights for precision agriculture. The utilization of low-code platforms could enhance decision support services in crop analysis by expediting the development and implementation of data analysis models, thereby allowing for quick prototyping and iteration. By employing visual modeling interfaces and pre-built components, farmers and agricultural experts can simplify the creation of personalized data analysis workflows. Additionally, low-code platforms enable the seamless integration of diverse data sources, facilitating the incorporation of real-time information from sensors, weather predictions, and other pertinent sources. Furthermore, the use of low-code platforms empowers individuals without technical expertise to actively participate in the creation and customization of decision support systems, promoting collaboration and knowledge sharing within the agricultural community.

In conclusion, the integration of low-code platforms in decision support services can significantly create a better decision-making process in precision agriculture. By providing tools for the rapid development and deployment of applications, low-code media can help farmers and agricultural experts make data-driven decisions, optimize their operations, and improve productivity.

3.7.5 System Integration Services

System integration is a service that low-code platforms can offer. The field of precision agriculture frequently employs a variety of technologies, including GPS, remote sensing, and automated machinery. The integration of these technologies can be facilitated by low-code platforms, enabling them to operate in harmony. This has the potential to boost the efficiency and effectiveness of precision agriculture operations.

The research by Zhu et al. (2021) proposed a system integration approach that combines electromechanical hybrid systems and hydromechanical composite transmission to offer solutions to the operational needs of tractors. Low-code platforms can create a faster integration process of various components and subsystems, enabling quick prototyping and system iteration. Low-code platforms offer visual modeling interfaces and pre-built connectors, simplifying the integration of diverse systems and technologies, and reducing the time and complexity associated with integration tasks. Furthermore, low-code platforms enable flexible customization and configuration, empowering system integrators to adapt the solution to specific tractor requirements.

By harnessing the capabilities of low-code platforms, system integrators can efficiently incorporate the proposed MEH-PS into tractors, loaders, graders, and other vehicles. The modular nature of low-code platforms simplifies the integration process, permitting component reuse and reducing development efforts. Moreover, low-code platforms help with the collaboration among stakeholders, allowing domain experts, engineers, and manufacturers to actively participate in the integration process and contribute their expertise. This collaborative approach increases the reliability and performance of the integrated system, while also promoting knowledge sharing and continuous improvement.

In the study conducted by Chang & Huang (2010), a digital muskmelon dynamic growth system was developed using a process-based morphogenesis simulation model and component-based software. The system allows for the observation of muskmelon organs and the dynamic growth process of individual plants under varying environmental conditions, providing valuable insights into the plant's morphological status during its growth. The implementation of this system opens up possibilities for various applications in the field of system integration services. By integrating the digital muskmelon growth system with other technologies and tools, such as low-code platforms, its capabilities can be developed more and its applicability in research, decision support, education, and extension services can be improved.

The research by W. Wang et al. (2022) presented a fruit target detection and segmentation method using deep learning technology, with a focus on its application in system integration services. Low-code platforms could improve the fruit target detection and segmentation method. By using low-code development tools, the integration of various components and models becomes more simple and efficient. Low-code platforms enable visual modeling and pre-built components, reducing development time. This facilitates the integration of the fruit target detection and segmentation method into existing systems and applications, creating a better overall system performance.

The research by Uehara et al. (2022) proposed a feature engineering approach that quantifies hidden regional characteristics for prediction without relying on forecasts. In the context of system integration services, the use of low-code platforms can enhance the feature engineering process and improve predictive performance. Low-code development allows for the seamless integration of various data sources, including observational data and regional characteristics, enabling efficient feature extraction and modeling.

3.7.6 Enhancing Accessibility and Democratization

Low-code platforms can significantly contribute to precision agriculture by increasing and improving the accessibility of technologies and democratizing their use. These platforms can simplify the use of precision agriculture technologies, making them more accessible to farmers with varying levels of technical expertise. Furthermore, they can increase the speed for the development and deployment of applications, a feature that is particularly beneficial in the dynamic and rapidly evolving field of agriculture Yu et al. (2021); Nawaz et al. (2022); Ramos et al. (2010); Hennicker et al. (2010).

In the study by Yu et al. (2021), the integration of Deep Learning and Factorization Machine (FM) techniques was explored to improve models such as Factorization-supported Neural Network (FNN), Product-based Neural Networks (PNN), Inner PNN (IPNN), Wide and Deep, Deep and Cross, and DeepFM for addressing the Click-Through-Rate (CTR) problem. These models have applications beyond recommendation systems and can be utilized in various fields, including agriculture, meteorology, and disease prediction. The study demonstrated that by making minor improvements and parameter adjustments to DeepFM, excellent performance in terms of Area Under the Curve (AUC) was achieved compared to other models.

In the context of precision agriculture, these models can be used to predict crop diseases or pest infestations, enabling early intervention and reducing the workload of farmers. Furthermore, the incorporation of the Internet of Things (IoT) for monitoring changing data such as soil moisture, temperature, and nutrient levels enhances the timeliness and accuracy of these predictions. IoT sensors enable real-time transmission, ensuring timely predictions and early intervention. The integration of DeepFM and IoT creates a robust agricultural system that excels in forecasting and performance. This system can improve crop yields, optimize resource allocation, and enhance overall agricultural productivity. The use of low-code platforms in the development and deployment of such systems can later democratize access to advanced predictive models. Low-code platforms, with their drag-and-drop interfaces and pre-built templates, can enable farmers and agricultural specialists to build and customize their predictive models without needing extensive coding knowledge.

The research conducted by Nawaz et al. (2022) presents an intelligent human-machine interface, ANKSys, which integrates advanced numerical solutions and data-driven supervisory control to address the challenges faced by eco-efficient anaerobic ammonium oxidation (ANAMMOX) in wastewater treatment. ANKSys utilizes optimized functionality, including soft sensing, decision making, and simulating models, driven by advanced algorithms such as artificial neural networks, Kalman filters, principal component analysis, and least-square techniques.

Precision agriculture can make use of this, ANKSys could be adapted to manage and optimize irrigation systems, reducing water usage and improving crop yields. The real-time implementation of ANKSys in a pilot-scale wastewater treatment plant in Daegu, Republic of Korea, resulted in a significant improvement in energy efficiency, achieving a 16 percent reduction in energy consumption. This approach facilitated optimal and sustainable operation for the removal of biological nitrogen. The utilization of ANKSys and its integration with advanced algorithms and software tools exemplify the potential for improving accessibility and democratization of eco-efficient wastewater treatment technologies. Furthermore, the possibilities offered by low-code platforms can provide the development and deployment of intelligent systems like ANKSys, making them more accessible to a wider range of users. Low-code platforms, with their ability to integrate with various data sources and APIs, can facilitate the seamless integration of ANKSys with existing agricultural systems, thereby enhancing its accessibility and usability.

The research by Hennicker et al. (2010) presents a generic framework for computer-based environmental modelling aimed at enhancing the accessibility and democratization of integrative simulations. The framework facilitates the coupling of simulation models from diverse scientific disciplines, enabling the simulation of spatially distributed environmental processes with discrete time scales.

For precision agriculture this means that, this framework could be used to integrate models from various disciplines including meteorology, hydrology, plant physiology, and agronomy to create a comprehensive decision support system for farmers. Through the GLOWA-Danube project, the framework has been successfully applied to construct the distributed simulation system DANUBIA. DANUBIA integrates up to 15 simulation models from various disciplines including meteorology, hydrology, plant physiology, glaciology, economy, agriculture, tourism, and environmental psychology. This powerful simulation system serves as a decision support tool for sustainable planning and aids decision makers in shaping the future of water resources in the Upper Danube basin. The proposed framework and its application in DANUBIA exemplify the possibilities for enhancing the accessibility and democratization of computer-based environmental modeling. Furthermore, the utilization of low-code platforms can significantly contribute to improving accessibility and accelerating the development of integrative simulation systems like DANUBIA. Low-code platforms, with their ability to handle complex business logic and workflows, can simplify the process of building and deploying integrative simulation systems like DANUBIA, thereby making them more accessible to non-technical users.

The research by Ramos et al. (2010) focuses on the development and utilization of Unmanned Aerial Vehicles (UAVs) and their potential to perform various tasks with reduced risk compared to conventional manned aircraft. This technological advancement has brought economic benefits to sectors such as agriculture, energy, public safety, and telecommunication by offering cost-effective operations, improved maintenance, and enhanced safety measures.

In the context of precision agriculture, UAVs can be used for tasks such as crop monitoring, disease detection, and yield estimation. In Brazil, the Veículo Aéreo Não Tripulado (VANT) Project, led by the Air Force, Army, and Navy under the Department of Defense, aims to develop a UAV for multiple applications. The Brazilian Aeronautics Institute of Technology (Instituto Tecnológico de Aeronáutica - ITA) has played a significant role in this project through its Software Engineering Research Group, focusing on the development of a Preliminary Testing Monitoring Station (PTMS) known as Estação de Monitoramento de Ensaios Preliminares (EMEP). The PTMS plays an important role in the overall success of the VANT Project. This paper highlights the major contribution of adopting a comprehensive development life-cycle approach, leveraging state-of-the-art practices such as System Engineering, Rational Unified Process (RUP), Military Standards, Model-Driven Development (MDD), and Integrated Computer-Aided Software Engineering (I-CASE-E) tools. By embracing these modern approaches, the development of the Unmanned Aircraft System (UAS) has been achieved with timely integration and adherence to the specified requirements. Enhancing accessibility and democratization in UAV development is an important aspect that can be further improved. The utilization of low-code platforms offers the possibility of simplifying and accelerating the development process even more, allowing a broader range of stakeholders with varying technical backgrounds to contribute to the development of UAV systems. The application of low-code platforms in the development of the VANT Project can enhance accessibility and democratization, enabling a more inclusive and collaborative environment for stakeholders involved in

UAV development. Low-code platforms, with their visual development interfaces and automated testing capabilities, can simplify the process of developing and deploying UAV systems.

3.7.7 Comprehensive Analysis of Low-Code Platform Services in Precision Agriculture

After examining the various cases presented in the preceding subsections, we can deduce the significant role low-code platforms can play in improving precision agriculture. These cases underscore the importance of services such as data collection, data analysis, decision support, and system integration in the context of precision agriculture. Furthermore, they show how low-code platforms can democratize precision agriculture by making it more accessible to farmers with varying levels of technical expertise.

The ability of low-code platforms to facilitate data collection and analysis is particularly noteworthy. Precision agriculture heavily relies on data to make informed decisions about various aspects of farming operations. Low-code platforms can simplify the process of developing applications for data collection and analysis, enabling farmers to harness the power of data in their operations more effectively. This can lead to increased efficiency and productivity, which are key objectives of precision agriculture.

Another crucial point that emerges from these cases is the emphasis on system integration. Precision agriculture often involves the use of various technologies, such as sensors, drones, and satellite imagery. Low-code platforms can improve the integration of these technologies, enabling them to work together seamlessly. This can create better effectiveness of precision agriculture operations, leading to better outcomes.

The consistency in the findings from these diverse cases is significant, as it provides a strong basis for further research and development in this area. It suggests that the potential benefits of low-code platforms in precision agriculture are widely recognized, providing a strong incentive for further exploration of this technology. Moreover, the identification of common challenges provides a clear direction for future research aimed at overcoming these obstacles.

In conclusion, the analysis of the various cases reveals a strong consensus regarding the potential benefits and challenges of low-code platforms in precision agriculture. This consensus provides a solid foundation for further research and development in this area, with the ultimate goal of harnessing the full potential of low-code platforms to enhance the efficiency and productivity of precision agriculture.

3.8 Research Question 3 (RQ3): which low-code platform fits better solutions with the model-based approaches on their usage to optimize precision agriculture?

3.8.1 Introduction to RQ3:

The current surge in the market for low-code platforms (LCP) is a phenomenon that has been substantiated by big projections and concrete investment figures Tzounis et al. (2017). The question that arises is why this phenomenon is occurring at this particular moment in history.

Historically, the concept of elevating the level of abstraction was considered the next step in software development, dating back to the first looks of Computer-Aided Software Engineering (CASE) tools in 1991 Tisi et al. (n.d.). This indicates that over the past three decades, there have been numerous attempts to achieve this goal. So, what makes the current situation different?

It appears that vendors have realized that by using a term more familiar to developers, such as 'coding' instead of 'modeling', they could positively alter user perceptions. This approach appeals to both professional developers and those unfamiliar with coding, as the term 'low-code' seems more attractive and less confusing than 'model-driven development' Vincent et al. (2018).

Even though the adoption of the term 'low-code' has drawn attention to this trend, it remains unclear how to compare different vendors in a field that is still evolving. If we consider the basic aspects that low-code should encompass, such as the use of tools to dramatically reduce the amount of code that needs to be written to develop software, all vendors using the term are meeting this requirement in various ways Bergerman et al. (2016). However, when we delve deeper into the characteristics that low-code should possess to serve as an alternative to traditional coding practices, the path becomes more complex Whittle et al. (2013).

The lack of a clear conceptualization of LCPs is a significant motivation for conducting an exploratory study. However, without such a conceptualization, selecting platforms for comparison becomes a delicate task Wu et al. (2019). The absence of a clear concept for what low-code represents limits the methods available to analyze it. In this research, we are using the method proposed by Frank (2021) with slight modifications to facilitate a simpler comparison of specific platforms that may be of interest for precision agriculture Nikkilä et al. (2010).

3.8.2 Analysis of Low-Code Platforms for Precision Agriculture:

It's important to note that the selection of the best low-code platform for Process Automation (PA) in precision agriculture depends on several factors, including the specific needs of the farm, the complexity of the operations, the technical expertise of the users, and the resources available Tan et al. (2012b).

Low-code platforms vary in their features and capabilities. Some platforms are designed for simplicity and ease of use, making them suitable for users with limited technical expertise. These platforms provide a visual interface where users can drag and drop components to create applications, reducing the need for extensive coding Ghandar et al. (2018).

Other platforms are more advanced, offering a wide range of features for data collection, data analysis, decision support, and system integration. These platforms may require more technical expertise to use, but they can provide more flexibility and control over the applications Yao et al. (2020).

In the context of precision agriculture, a low-code platform that supports the integration of various technologies, such as sensors, drones, and satellite imagery, would be highly beneficial. This would allow for the collection of data from different sources, providing a comprehensive view of the farm Hong et al. (2016).

Additionally, a platform that supports advanced data analysis techniques, such as machine learning and predictive modeling, would be advantageous. This would enable the development of applications that can analyze the collected data and make predictions about future outcomes, such as crop yields and disease outbreaks Sharma et al. (2021).

A low-code platform that provides decision support would also be valuable. By integrating data collection and analysis, the platform could help farmers make informed decisions about various aspects of their operations, such as when to plant, how much fertilizer to use, and when to harvest Manjula & Narsimha (2015).

In terms of resources, a low-code platform that is cost effective and requires minimal hardware would be preferable. This would make the platform more accessible to small and medium-sized farms, which may have limited resources Toyama & Yamamoto (2009).

In conclusion, the best low-code platform for Process Automation (PA) in precision agriculture would be one that meets the specific needs of the farm, supports the integration of various technologies, provides advanced data analysis capabilities, offers decision support, and is cost-effective and resource-efficient.

However, it should be noted that the selection of the best platform also depends on the specific circumstances of the farm, including the complexity of the operations, the technical expertise of the users, and the resources available Kassim & Abdullah (2012).

Despite these considerations, the potential benefits of low-code platforms in precision agriculture are significant. As such, further research and development in this area are warranted to identify the best platforms and maximize their benefits Jain et al. (2021).

3.8.3 Overview of Specific Low-Code Platforms:

Aurea BPM: This tool provides extensive support for modeling, automation, managing, and optimizing business processes Barriga et al. (2021). It has been used in manufacturing, demonstrating its potential for complex, industrial applications Barriga et al. (2021); *Aurea BPM* (n.d.).

Kissflow: A cloud-based workflow automation software platform that can help users automate business processes Florin & Florin (2020); *Kissflow* (n.d.). Its cloud-based nature could be beneficial for precision agriculture, where data is often collected from various sources and needs to be accessible from different locations.

Now Platform by ServiceNow: This low-code platform is used for workflow automation Daniela et al. (2020); *Now Platform by ServiceNow* (n.d.). ServiceNow's interpretation of how best to deliver low-code workflow automation could be relevant for precision agriculture, where workflows can be complex and involve multiple steps and stakeholders.

OutSystems: OutSystems is a low-code platform that has been recognized for its strong all-around capabilities, including in data management, mobile application development, and customer experience Alulema et al. (2020); *OutSystems* (n.d.). These capabilities could be beneficial in precision agriculture, where data management is crucial, and applications often need to be accessible on mobile devices in the field.

Mendix: Mendix is a low-code platform known for its strong integration capabilities and comprehensive development environment Wijekoon et al. (2020); *Mendix* (n.d.). These features could be particularly useful in precision agriculture, where integration with various technologies (e.g., sensors, drones, satellite imagery) is often required.

Appian: Appian is a low-code platform that combines process management, data management, and collaboration in a single platform Rio et al. (2019); *Appian* (n.d.). This combination could be beneficial in precision agriculture, where these aspects are often intertwined.

Microsoft Power Apps: This platform allows users to build professional-grade apps with a low-code approach Plazas et al. (2019); *Microsoft Power Apps* (n.d.). Its integration with other Microsoft products could be beneficial for farms that already use these products.

Zoho Creator: This platform offers a balance between ease of use (with drag-and-drop interfaces and pre-built templates) and powerful functionality (with features like script builder for custom logic and integration with other Zoho apps) He et al. (2019); *Zoho Creator* (n.d.).

Quick Base: Quick Base is known for its user-friendly interface and powerful functionality, including the ability to build custom applications and automate workflows Borowski (2019); *Quick Base* (n.d.).

Betty Blocks: This platform stands out for its focus on citizen development, allowing non-technical users to build applications Zečević et al. (2018); *Betty Blocks* (n.d.). This could be beneficial in precision agriculture, where farmers and other stakeholders may not have extensive technical expertise.

In conclusion, the best low-code platform for PA in precision agriculture depends on the specific needs and resources of the farm. Factors to consider include the complexity of the workflows to be automated, the need for integration with other technologies, the technical expertise of the users, and the resources available for implementing and maintaining the platform. Therefore, a careful evaluation of these factors is necessary to select the most suitable platform Muccini et al. (2018).

3.9 Research Question 4 (RQ4): What are the critical steps involved in developing a low-code system based on the model-based approach in precision agriculture?

3.9.1 Understanding the Developer Categories and Their Needs

The construction of a software system needs to follow certain steps to guarantee the best outcomes. Traditional code development follows a sequential process that involves various stages, such as requirement gathering, design, implementation, testing, and deployment. However, with the emergence of low-code development platforms, the development process has undergone a paradigm shift, enabling developers to create software applications more efficiently and rapidly.

In the context of Moisescu et al. (2017), low-code development can play a significant role in enhancing the development of complex precision agriculture systems, specifically in the realm of Cyber-Physical Systems of Systems. By leveraging low-code platforms, developers can enhance the design and implementation phases, accelerating the development of integrated systems that combine physical and virtual environments.

In the domain of precision agriculture, low-code development can offer several advantages. It allows the integration of sensor networks, intelligent machines, actuators, and processes, fostering seamless communication and data exchange between components. Low-code platforms provide pre-built components, modules, and frameworks that facilitate the rapid development of complex architectures, such as the proposed Agricultural Enterprise System Architecture. Developers can use these platforms to quickly design and implement automated processes, by using the principles and methodologies offered by Cyber-Physical Systems. Moisescu et al. (2017).

In the realm of low-code development, there are two distinct groups: professional developers and citizen developers. Professional developers are those who are trained or possess knowledge in coding languages and traditional development steps due to their technical training and experience. On the other hand, citizen developers might not have extensive coding or software development expertise, but they bring to the table a deep understanding of specific topics or applications due to their firsthand experience.

A prime example of a citizen developer can be found in the field of precision agriculture. Consider a farmer who is aiming to automate farm processes. Despite not having formal coding training, the farmer's intimate knowledge of the farm's needs and operations makes them an effective developer in this context. This is where low-code development platforms come into play, providing an accessible tool for these citizen developers to create applications tailored to their specific needs.

To illustrate this, let's take the case of Senegal's agricultural sector, as described by Racine Ly et al. (2021). In Senegal, agriculture is a crucial economic sector, but it faces numerous challenges such as insect attacks, plant diseases, and varying soil conditions. Despite these challenges, there is a shortage of agricultural engineers to advise and help farmers due to the limited number of agricultural engineering schools. This is where citizen developers, such as the farmers themselves, can step in. With the aid of a low-code platform, they can develop a system that uses IoT devices to monitor field conditions in real-time, providing valuable data for decision-making.

In this scenario, the low-code platform serves as a bridge between the professional developers who built the platform and the citizen developers (the farmers) who use it to address their specific needs.

The platform allows the farmers to leverage their firsthand knowledge of their fields to create a system that provides real-time data, which can then be used by agricultural engineers for efficient and effective decision-making.

Therefore, it's important for low-code development platforms to accommodate the needs of both professional and citizen developers. By doing so, they empower individuals with varying levels of technical expertise to create solutions that address specific needs, ultimately driving innovation and efficiency in sectors like precision agriculture. Racine Ly et al. (2021).

Professional developers regard low-code platforms as an advanced toolbox that increases efficiency and optimizes resource use, building upon their existing knowledge. They anticipate that the low-code platform will provide them with the essential tools and features to improve their development process. In contrast, citizen developers, such as those involved in the "Research and Application of Teaching Model Reform and Application of Electrical Automation Technology in Higher Vocational Education Driven by Real Agricultural Application" project at Shandong Electric Engineering College, are in pursuit of a user-friendly and practical interface. This interface should enable them to attain outcomes similar to those achieved with traditional software development tools and coding languages. They necessitate a platform that is straightforward and comprehensible, allowing them to effectively achieve their objectives. The project at Shandong Electric Engineering College exemplifies this need, as it employs a new talent training model for automation technology, driven by real-world applications and centered on the student. This approach has proven effective in practice, demonstrating the potential of low-code platforms to meet the needs of both professional and citizen developers in diverse fields. M. Liu & Yao (2021).

In order to ensure optimal results in software system development, it is important to clearly define the stages that low-code developers must follow. This involves a detailed examination of how these stages differ from the steps involved in traditional code development. For instance, while the traditional approach often requires a deep understanding of complex programming languages, low-code development can be more accessible, allowing developers to create applications using intuitive, visual interfaces.

In modern agricultural equipment research, as exemplified by the electronic and information engineering specialty at South China Agricultural University, a characteristic course system was established. This system includes platforms and modules that cater to the needs of both professional and citizen developers. For professional developers, the system provides technical sub-platforms that serve as advanced toolboxes, enhancing efficiency and resource utilization. These sub-platforms leverage their existing knowledge and provide them with the necessary tools and features to optimize their development process.

On the other hand, for citizen developers, such as those involved in the agricultural machinery intelligent navigation technology, agricultural remote sensing, and the agricultural internet-of-things, the system offers specialized elective courses. These courses are designed to be user-friendly and practical, enabling citizen developers to achieve results comparable to those achieved using traditional software development tools and coding languages. They require a platform that is easy to understand and use, enabling them to effectively accomplish their goals.

In essence, low-code development platforms can cater to the specific requirements of both professional and citizen developers, providing them with the necessary tools and resources to effectively contribute to the development of modern agricultural equipment. This approach not only enhances the efficiency of the development process but also inspires the enthusiasm of developers to serve the "three rural areas", and rural people, thereby providing a new talent cultivation mode in the domestic agricultural engineering discipline development. Lyu et al. (2019).

3.9.2 Requirement Analysis and Feasibility

The first step in both low-code and traditional software development is the analysis of requirements and feasibility. Stakeholders, such as farmers or agricultural organizations, provide a set of requirements, which the developer then evaluates to determine if they are achievable. This is a critical step for any development project, be it a product, solution, or service, as it lays the foundation for the development request and guides the creation of the final solution.

In agricultural countries, where the industry has significant impacts on world hunger, poverty, and climate change, the development of software applications with features such as price forecasting can be really relevant. For instance, a proposed application designed to assist farmers in having better market knowledge and maximizing their profit would require a thorough analysis of requirements and feasibility.Yuan & Ling (2020).

To exemplify this process, consider a farmer's scenario. Their specific needs might encompass regular comprehensive system monitoring reports and pinpointing areas that necessitate more frequent surveillance. The analysis of these needs can be undertaken similarly in both low-code and traditional methodologies. However, a distinctive advantage of the low-code approach is the manner in which the client can communicate their requirements.

Drawing from Wilson et al. (2019), this process could be further improved by a user needs-driven knowledge management system, as seen in the mobile-based solution described in the study. This system could capture the farmer's specific needs and preferences through participatory sensing, creating an up-to-date knowledge base. The use of natural language processing and ontology theories could help in understanding and structuring these requirements, making them more accessible and actionable for developers. This approach could ensure that the final solution is not only technically feasible but also closely aligned with the user's needs, thereby improving its effectiveness and usability in the agricultural context.

These technologies simplify communication between the client and developer, improve the accuracy of translating requirements into code, and facilitate knowledge sharing and reuse in the development process.

In traditional methodologies, the client usually provides a list of requirements or discusses them in a meeting. However, with low-code, the client can directly input their requirements into a Low-Code Development Platform (LCDP). This approach allows the communication processes between the client and the developer, as they are both using the same platform. It eliminates the hurdles associated with translating requirements from verbal or written instructions into coding language and then interpreting the resulting code back to the client.

Incorporating insights from Bonacin et al. (2013), this process could be further enriched by employing a web ontology to represent key aspects of the problem. This could lay the basis for computational mechanisms that increase knowledge sharing and recovery, such as semantic search, text mining, and visualization techniques. These tools could assist in comprehending and structuring the requirements, making them more accessible and actionable for the developers. Moreover, they could facilitate the reuse and integration of information, addressing some limitations and challenges encountered in the ontology engineering process.

The low-code approach, specifically its capability to visually represent requirements on a functional system, can simplify the developer's tasks and enhance collaboration. This method allows developers to see the requirements in a functional format, which simplifies their job as they only need to adjust existing functions to improve efficiency, rather than starting from scratch. This visual representation of the requirements facilitates easier collaboration between the developer and the client, ensuring that the final solution aligns with the client's needs.

Moreover, the insights gained from Ghandar et al. (2019) regarding a modeling framework for urban agriculture and a decision support system for coordinating decentralized urban agricultural production units contribute to the overall discussion and provide added value. The proposed framework and decision support system highlights the potential of low-code platforms in managing complex, decentralized systems like urban agriculture. This aligns with the low-code approach's emphasis on simplifying complex tasks and creating better efficiency terms.

In urban agriculture, a low-code platform could potentially be used to visually represent and manage the various components of the system, such as different production units, supply chains, and resource allocation. This would allow for easier coordination and decision-making, ultimately leading to more efficient and sustainable urban agricultural practices. This example illustrates the potential of low-code platforms in addressing complex, real-world challenges, further underscoring the value of the low-code approach in software development.

Overall, while the analysis of requirements and feasibility is a crucial step in both low-code and traditional software development, the low-code approach offers the benefits of streamlined communication and visual representation of requirements. These advantages contribute to a more efficient and effective development process Po Shun Chen & Wu Liu (2021).

3.9.3 Data Modeling

When talking of low-code development, the next step after initial development is data modeling. This phase involves creating a simplified diagram that represents the software system and its data components. The diagram uses a combination of text and symbols to visually depict how data flows within the system. Data models serve as a blueprint for designing a new database or restructuring an already existing application.

To exemplify this, let's consider the FieldTouch agricultural information service platform, an innovative decision support service based on a multi-scale sensor platform. FieldTouch integrates multi-scale sensor data for field monitoring and provides functionality for recording agricultural practices. This data is then modeled and visualized in a user-friendly interface, allowing farmers to make informed decisions based on real-time information from their fields. The data model in this case encapsulates not just the software system, but also the real-world agricultural data it is designed to process. This example illustrates how data modeling in low-code development can be used to create powerful, user-centric applications that meet specific needs in various sectors, such as agricultureHonda et al. (2014).

With the use of Low-Code Development Platforms (LCDPs), the diagram can be constructed in a more functional manner. This means that while the diagram is being created, the functions that the final product should include can be added at the same time. This approach results in a more efficient model that doesn't rely on assumptions about data processing.

For instance, for smart irrigation systems, as mentioned by Malheiro et al. (2019), complex mathematical models are developed and programmed to achieve efficient water usage. These models, which can be represented in a simplified diagram on an LCDP, can be enhanced with additional functions related to evapotranspiration, soil moisture, and water infiltration. These functions can be added concurrently as the diagram is being created, allowing for a more comprehensive and efficient model. This approach not only provides the development process but also ensures that the final product is more accurately tailored to the specific needs of the irrigation system, thereby improving water efficiency. Data modeling in low-code development enhances efficiency and ensures that the final product aligns with the intended functionality by providing a visual representation of the software system and its data elements. This visual representation allows for the integration of necessary functions during the design phase, which contributes to the overall efficiency of the development process.

For instance, consider the development of an automatic driving system for agricultural machinery, as mentioned in Y. Shi et al. (2019). During the design phase, a coaxial angular displacement measuring device and a GNSS/INS/vehicle combination navigation method can be integrated into the data model. These functions, represented visually in the model, contribute to the system's machine matching and environmental adaptability. This means that the system can adapt to different machines and environments, enhancing its overall functionality and usability.

Furthermore, the inclusion of a human-computer interaction interface in the data model simplifies operation and reduces the workload of agricultural mechanics. This interface allows users to interact with the system more easily and intuitively, reducing the need for extensive training or technical knowledge.

3.9.4 User interface design

According to Malheiro et al. (2019), the third step of the Lifecycle Data Processing (LCDP) methodology involves designing the user interface. This step is comparable to planning educational experiences for students that raise awareness about climate change and water conservation. The goal of user interface design in LCDP is to create an interactive platform that meets the users' needs. This process entails using the functional model derived from data modeling to develop a preliminary functional code that requires user interaction. For example, in the development of smart irrigation systems, a user-friendly interface is essential for users to comprehend and interact with complex mathematical models. This design is similar to the application design in traditional software development, where the developer constructs an architecture to implement functions based on prior planning. Just as students must understand and engage with experiments, stakeholders must validate this architecture, assuming it will work flawlessly. The advantage of user interface design is the ability to observe these functions in action during the architecture-building process and their interaction with the user. This parallels how students can grasp concepts like soil differences, evapotranspiration, soil moisture, water infiltration in the soil, and water efficiency through simple experiments.

3.9.5 Implementation of Business Logic and Integration of External Services

The fourth and fifth stages in low-code development, namely the implementation of business logic and the integration of external services, can be executed concurrently. This process can be visualized as establishing connections or 'wiring' that links the software and the user interface (UI) to the specific conditions of the end user. The business logic is a critical component of the program that encodes real-world business rules, determining how data can be created, stored, and modified. This aspect of the program is distinct from other parts of the software that may deal with lower-level details such as managing a database, displaying the user interface, system infrastructure, or generally connecting various parts of the program. The integration of external services is a process that leverages all the tools already deployed within the company, and supplements any missing ones, to optimize the performance of the software Kallioniemi et al. (2012).

3.9.6 Testing and Deployment

In the sixth phase of low-code development, the traditional stages of software development, testing, and deployment are ingeniously integrated. The testing stage is crucial, as it is where any product defects are identified, tracked, fixed, and retested until the product meets the quality standards established during the requirements gathering phase. Deployment, on the other hand, is a process that unfolds in stages, as dictated by the organization's business strategy. Initially, the product may be released within a limited segment and subjected to testing within a real business environment.

What distinguishes the low-code approach is the ability to perform these two steps simultaneously. This is made possible by the Lifecycle Data Processing (LCDP) interface, which facilitates continuous testing and debugging during the software creation and deployment process. This simultaneous approach not only enhances efficiency but also ensures a higher level of product quality and reliability.

From Niu et al. (2016), this process could be likened to the event-driven prediction model used in stock market forecasting. Just as this model extracts relevant information from news articles and historical price data to predict stock prices, the LCDP interface in low-code development continuously extracts and utilizes feedback from the testing and deployment stages to refine the software product. This ensures that the final product is not only efficient and reliable, but also accurately meets the user's needs and expectations.

3.9.7 Maintenance and Continuous Improvement

Drawing from the concept of "Internet + agriculture" Aiwen (2021), the seventh stage of the process, known as Lifecycle Data Processing (LCDP), ensures the highest quality and adaptability of the software through a continuous cycle of customer feedback and the incorporation of new features over time. The LCDP platform, much like an Internet of Things (IoT) system in agriculture, facilitates consistent communication with the customer, enabling the tracking of functions and customization of potential tools. This is akin to how IoT in agriculture allows for real-time monitoring and decision-making based on collected data.

Just as IoT-based smart agriculture can transform fields into communicative entities, the LCDP approach allows for the software to be adaptable and responsive to the user's needs. This is achieved by providing the opportunity to make adjustments based on any modifications in requirements indicated by the farmer or client after the initial deployment. This continuous feedback loop ensures that the software remains relevant and useful in the ever-evolving context of the user's needs.

In conclusion, the low-code development approach, as exemplified by the LCDP methodology, offers a streamlined, efficient, and user-friendly alternative to traditional software development. By enabling concurrent stages, continuous testing, and direct customer feedback, it ensures a higher level of product quality, adaptability, and customer satisfaction.

3.10 Research Question 5 (RQ5): What challenges are associated with implementing model-based approaches that could encourage low-code technology in precision agriculture?

The emergence of low-code platforms has revolutionized the software development landscape, offering a simplified, user-friendly approach to building applications. However, when applied to precision agriculture, these platforms face a large amount of challenges that need to be addressed to fully harness their potential.

One of the primary concerns is the scalability of low-code platforms. As discussed by Moisescu et al. (2017), precision agriculture systems are complex, often requiring the integration of physical and virtual environments. The scalability of low-code platforms is often met with skepticism, as they are not designed to handle such complexity. This is further compounded by the need for real-time data processing and analysis, which is critical in precision agriculture Racine Ly et al. (2021).

The limitations of low-code platforms in supporting dynamic event handling are another significant challenge. M. Liu & Yao (2021) and Lyu et al. (2019) highlight the need for dynamic content display, dynamic form controller, and dynamic content binding in precision agriculture. These requirements are often not fully supported by low-code platforms, thereby limiting their applicability.

The adoption of low-code platforms in precision agriculture also raises issues related to platform adoption, including access control and security, client-server communication, and I/O, as well as cloud and on-premise confirmation Yuan & Ling (2020). These issues can pose significant barriers to the widespread adoption of low-code platforms in precision agriculture.

Data management is another challenge. Precision agriculture involves the collection, storage, and analysis of large volumes of data. Low-code platforms often struggle with SQL CRUD operations, data storage and migration, and entity relationship management Wilson et al. (2019). This can limit their effectiveness in managing the large amounts of data generated in precision agriculture.

Integration with third-party services is also a significant consideration. As shown by Bonacin et al. (2013), integrating different platforms for low-code development can present technical challenges. This is particularly relevant in precision agriculture, where data from various sources, such as weather stations, soil sensors, and satellite images, need to be integrated for effective decision-making Ghandar et al. (2019).

The need for scalable reactive model transformations in low-code platforms is another challenge. As discussed by Po Shun Chen & Wu Liu (2021), these transformations must be able to quickly respond to platform events, such as updating derived views of the model due to changes. However, if numerous events occur concurrently and the transformation actions are time-consuming tasks, processing congestion can quickly arise, hindering platform performance.

Finally, the exponential increase in devices in IoT-based systems presents several challenges. As discussed by Honda et al. (2014), every node in an IoT network transmits data to the remote cloud, resulting in cloud congestion. The main challenges in IoT-based systems include minimizing latency with low power requirements, optimizing bandwidth usage, and dealing with intermittent internet connectivity. These issues need to be addressed to effectively leverage IoT in precision agriculture using low-code platforms Malheiro et al. (2019); Y. Shi et al. (2019).

In conclusion, while low-code platforms offer a promising approach to software development in precision agriculture, several challenges need to be addressed. These include scalability, dynamic event handling, platform adoption, data management, integration with third-party services, and the need for scalable reactive model transformations. Addressing these challenges will be crucial in harnessing the full potential of low-code platforms in precision agriculture.

4 Comparative Analysis of Node-RED and Python Implementations for Irrigation Control in Precision Agriculture

This section presents a comparative analysis of two implementations of an irrigation control system in precision agriculture that provides a more visual and tangible way to answer the five previous research questions, one using Node-RED, a low-code platform, and the other using Python, a high-level programming language. The relevance of this comparison lies in the growing interest in low-code solutions in precision agriculture, particularly among farmers with limited coding experience. The irrigation control system serves as a practical example of a common task in precision agriculture that can benefit from automation.

4.1 Methodology

The comparison is conducted by the parallel development of the irrigation control system utilizing both Node-RED and Python. The central objective of the system is to supervise soil moisture levels and trigger the irrigation system once the moisture level dips beneath a specified threshold. Both these implementations make use of simulated sensor data to replicate the real-world functioning of such a system.

The logical underpinnings of the code have been designed with the intent of maintaining a significant degree of similarity across both implementations. This ensures that the comparison is devoid of any extraneous variability and remains as equitable as possible. to maintain a quantitative possibility for measuring the time of development, a 1-hour maximum time was set from starting to code to deployment.

In the following sections, readers will be guided through the steps employed in the deployment of both these simulations. After that, a detailed analysis comparing the advantages and disadvantages of each implementation will be furnished. The study endeavors to evaluate these two strategies through the lens of precision agriculture, considering their effectiveness, efficiency, and ease of use, particularly for those with limited coding expertise.

4.2 Node-Red simulation design

The successful operation of an agricultural monitoring and control system is often predicated on the processing of sensor data. However, in the early stages of system development, it is imperative to establish and test system functionalities using simulated data prior to the deployment in a real-world context. Node-RED provides a robust and intuitive infrastructure for simulating sensor data using its Inject and Function nodes.

The preliminary stage of the simulation process involves the integration of an Inject node into the system's flow. This is efficiently accomplished by selecting the Inject node from the palette and dragging it into the canvas. Upon its addition to the flow, customization of the Inject node is paramount to defining the rhythm of the system simulation. By double-clicking the node, the user can access the settings and set the node to auto-inject data at a specified interval, in this instance, every 20 seconds. This cyclic data injection simulates the periodic acquisition of data from physical sensors in a real-world agricultural monitoring scenario.

With the automated Inject node established, it must be connected to a Function node for the generation of simulated sensor data. This connection models the typical data flow from a sensor to a processing unit in a physical system. In the Function node, programmatic rules can be defined to generate simulated sensor data; in this particular context, temperature and humidity levels for both air and soil are being generated.

The simulation code provided within the Function node harnesses JavaScript's Math library to generate random temperature values ranging between 5 and 35 degrees Celsius, and humidity values ranging from 30 to 100 percent. This random data generation technique provides a diverse set of data points, adequately mimicking the variability of real-world sensor readings.

```

msg.payload = {
    air: {
        temperature: Math.floor(Math.random() * (35 - 5 + 1)) + 5,
        humidity: Math.floor(Math.random() * (100 - 30 + 1)) + 30
    },
    soil: {
        temperature: Math.floor(Math.random() * (35 - 5 + 1)) + 5,
        humidity: Math.floor(Math.random() * (100 - 30 + 1)) + 30
    }
};
return msg;
```

In the creation of an agri-tech solution, such as an automated irrigation and ventilation system, it is crucial to have an effective decision-making mechanism in place. This logic determines when to activate the system based on the input sensor data, thus ensuring the optimum conditions for plant growth are maintained. With Node-RED, such control logic can be encoded using a Function node.

Following the creation and connection of the previous Function node, which generates the simulated sensor data, the next step entails adding another Function node to the flow. This node will be responsible for processing the input data and making corresponding decisions.

The connection between these Function nodes symbolizes the flow of data from the data generation stage to the decision-making stage. This chain of nodes emulates the real-world scenario where the data acquired from sensors would be processed by a control system to make informed decisions.

Inside this new Function node, a set of conditional statements are defined. The conditions are based on the logic required for the greenhouse environment control. In the context of this particular application, the decision to activate the irrigation and ventilation system is driven by the temperature and humidity levels of the air and soil.

For instance, if the soil humidity drops below 50 percent and the air temperature rises above 30 degrees Celsius, the system will be activated, as per the stipulated conditions in the Function node. If these conditions are not met, the system remains inactive. This intelligent decision-making process helps maintain the environmental parameters within their optimum ranges, thereby promoting the healthy growth of crops.

```

if (msg.payload.soil.humidity < 50 && msg.payload.air.temperature > 30) {
    msg.payload.action = "Activate irrigation and ventilation system";
} else {
    msg.payload.action = "No action needed";
}
return msg;
```

The implementation of an effective and easily interpreted user interface is a crucial factor for ensuring optimal user engagement and seamless navigability, particularly for users possessing limited programmatic proficiency, such as agricultural practitioners. The visual programming tool Node-RED facilitates this process via the use of its integral Dashboard nodes.

The initial stage of this procedure requires the integration of Node-RED's Dashboard nodes, if these have not been pre-installed in the Node-RED environment. This installation procedure is efficiently executed by selecting the 'Manage palette' option found within the dropdown menu, commonly known as the hamburger menu, located at the upper right corner of the Node-RED interface, and subsequently searching for 'node-red-dashboard' within the Install tab. This action readies Node-RED for the development of a dynamic and interactive dashboard.

Upon successful installation of the Dashboard nodes, the creation of a real-time and visually rich data display becomes feasible using the Gauge nodes. Each sensor datum – encompassing air temperature, air humidity, soil temperature, and soil humidity – is represented by a Gauge node that is added to the flow via a simple drag and drop operation. The Gauge nodes are then connected to the output of the Function node, previously designated for simulating sensor data. By accessing the settings of each Gauge node, the values can be precisely mapped to correspond with their respective sensor data, ensuring accurate and visually compelling data representation.

To explicitly communicate the operational actions of the system to the user, a Text node is incorporated into the flow. This node, connected to the output of the decision-making logic Function node, is configured to exhibit the current action of the system as determined by the logical conditions. This contributes to an enhanced system transparency, offering the user immediate comprehension of the system's current state without the need to interpret lines of code or parse system logs.

To allow direct control over system operations by the user, a Button node is integrated into the dashboard, designed to manually trigger the irrigation system. The Button node adds an essential layer of interactivity to the system, accommodating user intervention when immediate adjustments are necessary. This node is inserted into the flow and connected to a Function node, which, when activated, generates a specific command indicating the manual activation of the irrigation system., add the following code:

```
msg.payload = {
    action: "Manually activated irrigation system"
```

```
};
return msg;
```

Continuing the Node-RED setup, the output port of the decision-making Function node is linked to the input port of a text display node. This connectivity ensures that the resultant actions determined by the decision-making logic are clearly communicated in the user interface.

After the development and linking of the necessary nodes in the flow, the next stage encompasses the deployment of the overall logic. This process is accomplished by clicking the 'Deploy' button, situated in the top right corner of the Node-RED interface. The act of deploying the flow corresponds to both preserving and executing the created flow. As a result, the simulation commences, and the data flow mimics the real-world data transmission from sensors to the control system.

Post-deployment, the user interface (UI), which visually represents the data and decisions, can be accessed. By navigating to http://localhost:1880/ui in a web browser, users can interact with the interface. The UI reflects the current temperature and humidity data, both of the air and soil, along with the corresponding system action. As such, users receive real-time feedback, enabling them to monitor the greenhouse environment effectively.

This simulation provides a demonstration of how Node-RED, as a low-code platform, can be utilized by farmers to control and oversee the conditions within their greenhouses. By representing complex data and control systems visually, Node-RED allows users, even those with limited coding experience, to leverage advanced technology for agriculture.

The interactive nature of the UI, including the provision of manual activation for the irrigation system, empowers farmers with immediate control over their systems. Thus, Node-RED serves as a powerful tool in the integration of technology with agriculture, offering intuitive control over the environment.

4.3 Python simulation design

In the Python-based implementation, we develop a minimalist web server utilizing the Flask framework, a lightweight web application framework written in Python. This server is responsible for visualizing the current sensor data and the associated system response. The script runs a loop to update the sensor data every 20 seconds, aided by the use of a background thread. The updated data can be accessed by navigating to http://localhost:5000 on a web browser.

4.4 Code Elaboration

The Flask application is initiated. Following that, an initial setup for sensor data is defined with values for air and soil temperature and humidity, along with an action indicating whether any remedial measures need to be taken.

In the function, update sensor data an infinite loop generates random sensor data mimicking the environmental conditions within the agricultural setting. The data includes air and soil temperature, along with humidity levels, generated using the random.randint() function. Subsequently, the script checks these generated values to determine whether conditions warrant the activation of the irrigation and ventilation system. If the soil's humidity is below 50 percent and the air temperature exceeds 30°C, the system action is updated to 'Activate irrigation and ventilation system'; otherwise, it maintains its default state of 'No action needed'. The loop then pauses for 20 seconds with time.sleep(20) before initiating the next data update cycle.

In the home function, we define a route at the home page of our web server. Using Flask's render template string function, HTML content is dynamically generated to display the current sensor data and system action. A button element provides the option to manually refresh the displayed data, accomplished via JavaScript's location reload function, which refreshes the entire page.

In the script's main section, a new thread is initiated using the threading. This ensures that the sensor data updates every 20 seconds without interfering with the main application thread. The Flask web server is then activated using app run (debug=True), with the debug mode enabled for tracking potential errors.

Overall, the Python implementation provides an example of using a high-level language to create a web-based interface for monitoring and controlling an agricultural environment. It demonstrates the flexibility and control that comes with traditional programming but requires a substantial degree of coding proficiency.

It is imperative to highlight that the Python script illustrated herein does not encompass a functionality to manually activate the irrigation system, a feature present in the Node-RED demonstration. Integrating such a manual control feature necessitates the development of a more sophisticated web interface, which entails employing JavaScript and AJAX for the establishment of an interactive, asynchronous user interface. This augmentation, however, extends beyond the remit of this elementary exemplification. The intent of this example is to primarily focus on the core functionality of the irrigation control system, thus, a more detailed user interaction interface is not considered within the purview of this Python-based implementation.

```

from flask import Flask, render_template_string
import random
import threading
import time

app = Flask(__name__)

# Initial sensor data
sensor_data = {
    'air': {'temperature': 0, 'humidity': 0},
    'soil': {'temperature': 0, 'humidity': 0},
    'action': 'No action needed'
}

def update_sensor_data():
    while True:
        # Generate random sensor data
        sensor_data['air']['temperature'] = random.randint(5, 35)
        sensor_data['air']['humidity'] = random.randint(30, 100)
        sensor_data['soil']['temperature'] = random.randint(5, 35)
        sensor_data['soil']['humidity'] = random.randint(30, 100)

        # Decide when to activate the irrigation and ventilation system
        if sensor_data['soil']['humidity'] < 50 and sensor_data['air']['temperature'] >
    30:
            sensor_data['action'] = 'Activate irrigation and ventilation system'
        else:
            sensor_data['action'] = 'No action needed'

        # Wait for 20 seconds before updating the sensor data
        time.sleep(20)

@app.route('/')
def home():
    return render_template_string('''
        <h1>Greenhouse Monitoring System</h1>
        <p>Air Temperature: {{ sensor_data.air.temperature }} C </p>
        <p>Air Humidity: {{ sensor_data.air.humidity }}%</p>
        <p>Soil Temperature: {{ sensor_data.soil.temperature }} C </p>
        <p>Soil Humidity: {{ sensor_data.soil.humidity }}%</p>
```

```
        <p>System Action: {{ sensor_data.action }}</p>
        <button onclick="location.reload()">Refresh Data</button>
    ''', sensor_data=sensor_data)

if __name__ == '__main__':
    # Start a background thread to update the sensor data
    threading.Thread(target=update_sensor_data).start()

    # Start the Flask web server
    app.run(debug=True)
```

4.5 Implementation Overview

In the Node-RED realization of the system, the infrastructure is embodied as a data flow diagram interconnecting individual nodes, where each node stands as a task executor performing distinct operations - these include, but are not limited to, sensor data interpretation or commanding the irrigation controls. The graphical user interface furnished by Node-RED facilitates an effortless visualization of the system's architecture and its inherent data flow, alongside enabling convenient modifications. This network of nodes thereby serves as a visual encapsulation of the system's logical structure and operational sequence, providing an intuitive platform for users to understand and interact with the system's components and their interrelationships. figure 11

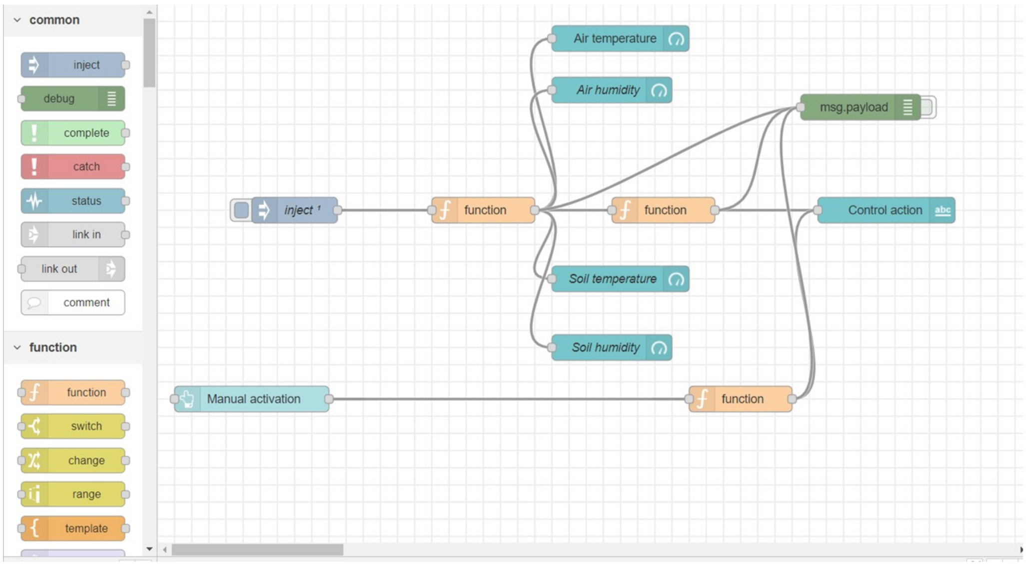

Figure 11: Node-red nodes

In the Python realization, the system is encapsulated within a script format where specific functions are allotted for sensor data interpretation, irrigation controls management, and orchestrating the principal control loop. The construction of the script incorporates standard Python libraries facilitating tasks such as managing chronological data and generating random numeric values. figure 12

The Python implementation necessitates a comprehensive understanding of programming constructs, syntax, and the idiosyncrasies of the Python language, as opposed to the visual approach employed by Node-RED. The Python script is executed within an integrated development environment (IDE), such as Spyder, which displays the written code but does not provide an intuitive, graphical representation of data flow or system structure.

This programming approach requires proficiency in reading and writing code, thus presenting a steeper learning curve, particularly for those with limited coding experience. Nevertheless, it allows for more complex and granular control over system operations, presenting its own set of advantages and disadvantages within the context of precision agriculture automation.

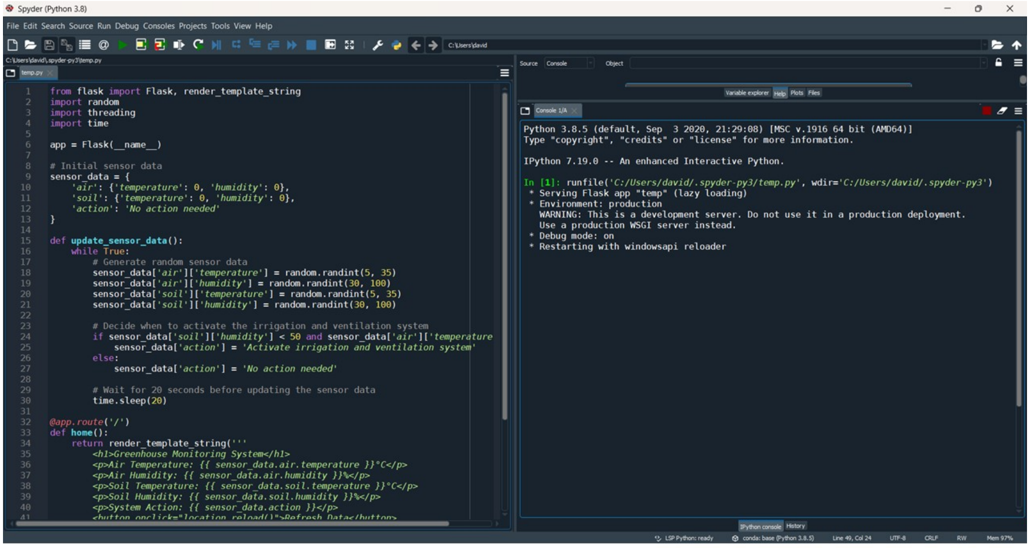

Figure 12: Python code

4.5.1 Comparison of Node-RED and Python

The Node-RED realization proffers the merit of a graphical user interface, enhancing comprehensibility and modification ease for the implemented system. This visual representation also enables real-time system monitoring and regulation. Despite its numerous benefits, Node-RED might exhibit limited flexibility and potency in handling more intricate tasks compared to traditional coding languages such as Python. figure 13

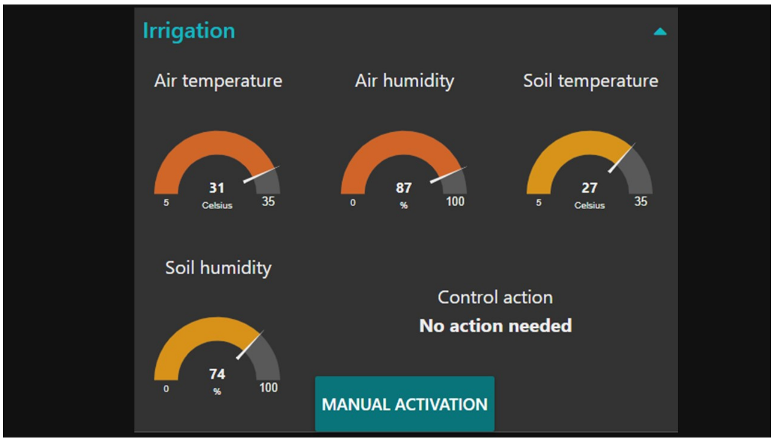

Figure 13: Node-red ui

On the contrary, Python's implementation, albeit demanding a more profound coding acumen, yields superior flexibility and computational power. The comprehensive array of Python's libraries and tools opens doors to more complex logical constructs and sophisticated data analysis capabilities. Nevertheless, the absence of a graphical user interface in Python might present a steeper learning curve, making the system somewhat enigmatic and challenging to modify for individuals possessing less coding experience. figure 14

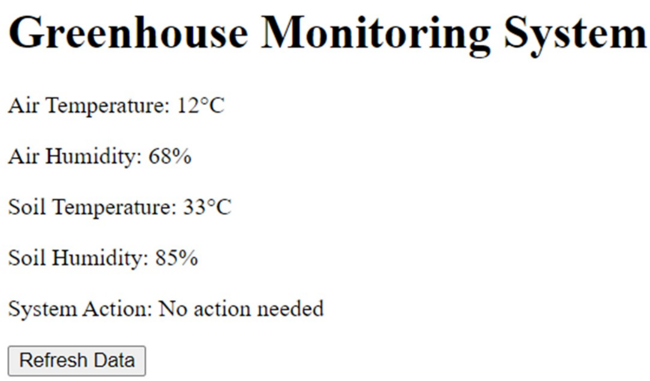

Figure 14: Python UI

4.6 Metrics Evaluation Revision

The evaluation metrics employed in this comparative study incorporated elements of development duration, resource utilization, reliability, and maintainability. To ensure a comprehensive understanding of these metrics by both technical and non-technical team members, we aligned these terms with the ISO/IEC 25000 series of standards, particularly the ISO/IEC 25010 taxonomy. This taxonomy explicitly defines eight critical product quality characteristics and 31 sub-characteristics, including functional suitability, reliability, performance efficiency, usability, security, compatibility, maintainability, and portability.

In this context, development duration, traditionally measured as the time taken to implement a system on each platform, may lean more towards the "usability" and "performance efficiency" characteristics, given the unique attributes of Python and Node-RED. Despite the experimenter's proficiency in Python, the visual, intuitive nature of Node-RED may expedite the development process, shifting this metric towards a more qualitative measure.

Resource utilization, assessed based on the computational resources required to operate the system, correlates with the "performance efficiency" characteristic. With both Python and Node-RED solutions being locally hosted, the computational demands remain comparable and relatively minimal.

Reliability, evaluated based on the system's ability to perform its tasks without errors, aligns with the "reliability" characteristic and remains a pertinent metric for both Python and Node-RED implementations.

Maintainability, gauged by the ease of modifying and updating the system, aligns closely with the "maintainability" characteristic from the ISO/IEC 25010 taxonomy. However, depending on the user's familiarity with coding, this measure may also extend into "usability" and "portability". Node-RED, with its visual interface, may be more accessible and portable for citizen developers, whereas Python, a text-based high-level language, might offer more intuitiveness and usability for experienced programmers.

While "functional suitability", "security", and "compatibility" were not directly evaluated in this study, they remain inherent aspects of both Python and Node-RED that could be addressed in a more comprehensive study.

By adopting the ISO/IEC 25010 taxonomy, these evaluation metrics offer an expansive view of the distinctive characteristics of Python and Node-RED, highlighting their unique strengths and weaknesses. This perspective underscores the complexity of applying standard software quality metrics in the emerging landscape of low-code platforms and traditional high-level languages.

4.7 Case study results and analysis

The observed results underscore the Node-RED implementation's efficiency in development time and maintainability, owing to its visual interface and intuitive design paradigm. Notably, a user proficient in coding managed to create a comprehensive Node-RED solution within a span of less than 40 minutes. The development process was facilitated by the readily available online references on Node-RED node functionalities, enabling the creation of an interactive user interface inclusive of a manual switch for the irrigation system.

Comparatively, the Python solution's development proved more time-consuming, with the same individual utilizing the full hour allotted for the creation process. The resulting Python solution offered a real-time monitoring capability for the simulated sensors, albeit lacking the manual switch for the irrigation system. The more extensive development time reflects the Python solution's inherent flexibility and power, necessitating a deeper understanding and proficiency in coding for a comprehensive implementation.

In essence, while the Python solution requires a higher coding literacy level and might offer superior flexibility and power for more complex tasks, the Node-RED solution provides an advantage in terms of development time and maintainability. Particularly for tasks where the complexity is not extreme and quick deployment is preferred, a low-code solution like Node-RED can provide significant benefits.

4.8 Case study discussion

The outcomes from this comparative analysis between a low-code platform, Node-RED, and a high-level programming language, Python, substantiate existing academic discourse pertaining to the application of low-code in precision agriculture. Consistent with the literature, Node-RED emerged as a more accessible and efficient alternative for less complex tasks, offering advantages in terms of rapid deployment, user-friendly visualization, and intuitive modification capabilities.

Meanwhile, Python, with its extensive libraries and powerful functionalities, demonstrated its superior flexibility and complexity, supporting complex tasks with a high degree of customizability. It's significant to note, however, that the exploitation of Python's flexibility and power necessitates a certain level of coding proficiency, potentially limiting its accessibility for individuals with limited coding experience.

Despite the enlightening outcomes from this analysis, it's crucial to recognize that the simplified irrigation control system used in this experiment may not comprehensively represent the intricacy associated with implementing low-code solutions in precision agriculture. Given that precision agriculture involves numerous variables and complex interdependencies, low-code platforms' real-world efficacy may vary depending on the specific application context and the complexity of the tasks at hand.

However, this study still provides a valuable initial exploration into the advantages and drawbacks associated with low-code and high-level coding approaches in precision agriculture. It's an indication that low-code platforms such as Node-RED have the potential to democratize precision agriculture by making it more accessible to farmers and stakeholders with limited coding experience. On the other hand, high-level programming languages like Python remain essential tools for more complex applications, where their flexibility, power, and comprehensive libraries can be fully utilized. Future research may delve into more complex and varied applications to further elucidate the optimal utilization of low-code platforms and high-level programming languages within the realm of precision agriculture.

4.9 Case study conclusion

In the context of the broader systematic literature review on low-code applications in precision agriculture, this case study has served to exemplify the practical use of a low-code platform, Node-RED, in accomplishing a fundamental precision agriculture task: an irrigation control system. It has simultaneously demonstrated the application of a high-level programming language, Python, for the same task, offering a direct comparison between these two coding approaches.

Both implementations were executed successfully, providing valuable insights into the potential benefits and challenges associated with each platform. The Node-RED implementation demonstrated the principles outlined in the research question 4 answer identified through the systematic literature review, which concerned the essential steps in the development of low-code solutions for precision agriculture. The steps included understanding the farming task, identifying the appropriate data inputs, designing the decision-making logic, and creating a user interface, all of which were successfully demonstrated in the Node-RED implementation.

The case study has affirmed the potential of low-code platforms like Node-RED in making precision agriculture more accessible to a broader range of individuals, especially those with limited coding experience. It has also reinforced the pivotal role of high-level programming languages like Python in managing more complex precision agriculture tasks, with their extensive libraries and advanced functionalities.

The case study has accomplished its objective of demonstrating the application of low-code platforms in precision agriculture, and it serves as a basis for further studies investigating the intersection between low-code platforms and precision agriculture. Future research could delve into more complex use cases and wider applications, seeking to further our understanding of how low-code platforms and high-level programming languages can best be utilized in the context of precision agriculture.

5 General conclusion

Guided by the objective to provide a relevant mapping of the current status and future projections of model-based approaches and their influence on the implementation of low-code platforms in Precision Agriculture (PA), our Systematic Literature Review (SLR) sought to construct a rigorous framework by answering five distinct research questions. Though a significant portion of the studies reviewed were not directly linked to low-code technology, their application of model-based systems was instrumental in providing valuable insights about the infusion of low-code solutions in the field of precision agriculture.

Tackling Research Question 1 (RQ1) exposed the main drivers for interweaving model-based approaches leading to low-code technology in PA, which include expedited software development, reducing dependence on extensive coding skills, and the potential to streamline technological integration in intricate PA systems. For instance, one study underscored the urgent requirement for software capable of managing the intricacies of modern agricultural systems while alleviating the learning curve for farmers.

In addressing Research Question 2 (RQ2), we discovered that the services offered by model-based approaches enhancing low-code platforms could include real-time data analysis, integrated operations management, and advanced data-driven decision support. These features, as elaborated in several reviewed articles, could considerably augment productivity and efficiency in PA.

Research Question 3 (RQ3) necessitated an assessment of various model-based approaches leading to low-code platforms. While no single approach emerged as the outright choice, a host of factors were recognized that could influence the selection. For example, some studies suggested the choice might depend on specific agricultural applications, user capabilities, and the degree of customization required.

Research Question 4 (RQ4) led to an exploration of the steps involved in developing a PA system using model-based approaches that could facilitate low-code application. This question directed us to studies depicting the processes of system design, software configuration, testing, and user training. These insights serve as a preliminary blueprint for future developers and researchers.

Lastly, Research Question 5 (RQ5) steered our inquiry into the challenges of implementing model-based approaches leading to low-code technology in PA. Despite the promising prospects, some studies spotlighted concerns surrounding scalability, data management, and integration with third-party services. Another recurrent challenge was the need for significant customization to cater to the unique demands of PA.

In summary, the SLR offers a balanced outlook on the potential benefits and challenges of utilizing model-based approaches that lead to low-code platforms in PA. It emphasizes the necessity for continued research and development in this intersecting field to circumnavigate the impediments and optimize potential benefits. It is through such comprehensive and systematic investigation that we can enable the successful integration of low-code solutions in precision agriculture, grounded in robust model-based approaches.

This research underscores the promising advantages and potential hurdles of adopting model-based approaches that could lead to low-code platforms in the domain of precision agriculture. Such approaches can facilitate the integration of diverse technologies used in precision agriculture, offering farmers a consolidated view of their operations and facilitating the application of sophisticated data analysis techniques to optimize crop health and yield.

However, the adoption of model-based approaches leading to low-code platforms in precision agriculture comes with challenges. These encompass potential constraints in performance and scalability, as well as the necessity for considerable customization to meet the unique needs of agricultural applications. Addressing these challenges is really important to show the potential benefits of coupling model-based approaches with low-code platforms in precision agriculture, possibly initiating a new era of more accessible, efficient, and effective precision agriculture.

Scalability, particularly in light of the complexity of precision agriculture systems, is a notable concern. The need for real-time data processing and analysis, dynamic event handling, and the ability to manage large volumes of data are all areas where model-based approaches leading to low-code platforms may encounter obstacles. Moreover, issues related to platform adoption, such as access control, security, client-server communication, and cloud and on-premise configuration, can pose significant barriers to the widespread adoption of these platforms in precision agriculture.

Integration with third-party services, a crucial aspect of precision agriculture, can also present technical challenges for model-based approaches that could lead to low-code platforms. The need for scalable reactive model transformations and the challenges posed by the exponential increase in devices in IoT-based systems, such as minimizing latency with low power requirements, optimizing bandwidth usage, and dealing with intermittent internet connectivity, are additional hurdles that need to be surmounted.

This integration could have extensive implications, not only enhancing agricultural practices but also contributing to broader societal objectives such as boosting food security and promoting sustainability. The development of suitable platforms and methodologies for crafting low-code systems in precision agriculture based on model-based approaches should consider the need for a deep understanding of both software engineering and agricultural practices.

Moreover, it should address the challenges associated with data management, integration with third-party services, scalability, and platform adoption. A balanced approach that weighs both the potential benefits and challenges of model-based approaches leading to low-code platforms in precision agriculture is vital to ensure the successful development and implementation of such systems in this field. This approach would eventually lead to a more technologically advanced and data-driven agricultural sector, capable of meeting the increasing demands of a growing global population.

6 Discussion and Future Work

The combination of model-based approaches into low-code development platforms (LCDPs) within the fieldwork of precision agriculture (PA) has unveiled a diverse group of research prospects, as thoroughly evidenced through this Systematic Literature Review (SLR). Important aspects such as the involvement of citizen developers, motives and potential applications of model-based systems leading to LCDPs, comparative evaluation of specific platforms, and the choice between low-code and high-level programming were dissected. These elements form the base for forthcoming discussions and future explorations.

The engagement of citizen developers in the utilization of model-based approaches leading to low-code platforms in PA has proven to be significant. While the research has offered insights into the part they could play, future endeavors could comprehend a deeper exploration of their potential inputs. Can they truly expedite the transition toward a more technologically-centric agriculture sector, provided they are equipped with the right tools and guidance? Additionally, what mechanisms could be established to stimulate their engagement and ascertain their effective contribution? Comprehending these dynamics could influence the design and adoption of model-based systems leading to LCDPs in PA and potentially, other domains.

The benefits such as fast-tracked software development, integrated data management, and the potential for enhanced decision-making, among others, underline the bright future of these approaches leading to LCDPs in PA. However, the exploration of these applications is still in its infancy. The vast spectrum and depth of PA operations suggest an abundance of areas where these systems leading to LCDPs could be further leveraged. We posit that future research could focus on recognizing these potential applications and the unique advantages that model-based approaches leading to LCDPs could confer to each.

The comparative analysis of Node-RED and Python shed light on crucial findings about development time, resource utilization, reliability, and maintainability. However, it also gave rise to questions that merit exploration in future research. For instance, what other model-based approaches could lead to beneficial LCDPs for PA? Are there particular PA tasks that would be better addressed by high-level programming languages? Comparative studies involving more platforms and a broader range of agricultural tasks could help answer these questions and provide more nuanced guidance to developers and users.

Lastly, our research underscored that the choice between low-code and high-level programming relies on various factors such as specific needs, user capabilities, and task complexity. This suggests that a one-size-fits-all solution may not be applicable to all PA scenarios, further reinforcing the need for continued exploration of different programming approaches for diverse PA needs. Moreover, the potential of model-based systems as a precursor to the implementation of low-code solutions provides a pathway that future research in this domain could explore, driving forward the integration of advanced computational tools in precision agriculture.

Acknowledgements

I am deeply grateful for the immense support and unwavering dedication of my family throughout my academic journey. Their unconditional love, encouragement, and belief in my abilities have been the pillars that have sustained me during the ups and downs of this thesis. I am indebted to their sacrifices, patience, and understanding, which allowed me to pursue my dreams with confidence.

To my girlfriend Aarzoo Kohra, whose unwavering support and understanding have been an invaluable source of strength. Your presence, encouragement, and belief in my abilities have been a constant motivation for me. Thank you for being there every step of the way, for providing insightful feedback, and for being a constant source of inspiration. Your love and unwavering support have truly been the driving force behind my accomplishments.

I would like to express my heartfelt appreciation to my supervisors, Onder and Qingzhi. Their guidance, expertise, and continuous encouragement have been instrumental in shaping this thesis. Their deep knowledge, constructive feedback, and dedication to my development as a researcher have significantly enriched the quality of my work. I am grateful for the trust they placed in me, as well as their patience and willingness to share their wisdom and expertise. Their mentorship has been invaluable, and I am truly grateful for their support throughout this entire process.

Additionally, I extend my gratitude to the faculty members and staff who have contributed to my academic journey. Their commitment to excellence and their passion for education has fostered an environment of growth and intellectual curiosity, which has been instrumental in shaping my research.

Finally, I would like to express my sincere appreciation to all my friends and colleagues who have provided support, advice, and encouragement along the way. Your discussions, insights, and camaraderie have been instrumental in shaping my ideas and enhancing the quality of my work.

References

Agrawal, N. K., & Sharma, A. K. (2021). Detection of diseases in plants using convolutional neural networks. In *Proceedings of the 2021 10th international conference on system modeling and advancement in research trends, smart 2021* (p. 215-217). Retrieved from www.scopus.com

Ahmed, N., De, D., & Hussain, I. (2018). Internet of things (iot) for smart precision agriculture and farming in rural areas [Journal Article]. *IEEE Internet of Things Journal*, *5*(6), 4890-4899.

Aiwen, W. (2021). Research on the integrated model of "internet + agricultural products" under the background of rural revitalization: – analysis based on porter's five forces model. In *2021 2nd international conference on internet and e-business* (p. 52–55). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3471988.3471997 doi: 10.1145/3471988.3471997

Akagi, Y., & Raksincharoensak, P. (2015). An analysis of an elderly driver behaviour in urban intersections based on a risk potential model. In *Iecon 2015 - 41st annual conference of the ieee industrial electronics society* (p. 001627-001632). doi: 10.1109/IECON.2015.7392334

Ali, O., Valckenaers, P., Van Belle, J., Saint Germain, B., Verstraete, P., & Van Oudheusden, D. (2013). Towards online planning for open-air engineering processes. *Computers in Industry*, *64*(3), 242-251. Retrieved from www.scopus.com (Cited By :4)

ALSAADI, H. A., RADAIN, D. T., ALZAHRANI, M. M., ALSHAMMARI, W. F., ALAHMADI, D., & FAKIEH, B. (2021). Factors that affect the utilization of low-code development platforms: survey study. *Romanian Journal of Information Technology and Automatic Control*, *31*(3), 123–140.

Alulema, D., Criado, J., Iribarne, L., Fernández-García, A. J., & Ayala, R. (2020). A model-driven engineering approach for the service integration of iot systems. *Cluster Computing*, *23*(3), 1937-1954. Retrieved from www.scopus.com (Cited By :6)

Ammad-udin, M., Mansour, A., Le Jeune, D., Aggoune, E. H. M., & Ayaz, M. (2016). Uav routing protocol for crop health management. In *2016 24th european signal processing conference (eusipco)* (p. 1818-1822). doi: 10.1109/EUSIPCO.2016.7760562

Appian. (n.d.). Retrieved from https://www.appian.com/ (Accessed: 2023-01-14)

Aurea bpm. (n.d.). Retrieved from https://www.aurea.com/ (Accessed: 2023-01-14)

Ayaz, M., Ammad-Uddin, M., Sharif, Z., Mansour, A., & Aggoune, E.-H. M. (2019). Internet-of-things (iot)-based smart agriculture: Toward making the fields talk. *IEEE access*, *7*, 129551–129583.

Baba-Cheikh, Z., El-Boussaidi, G., Gascon-Samson, J., Mili, H., & Guéhéneuc, Y.-G. (n.d.). A preliminary study of open-source iot development frameworks [Conference Proceedings]. In *Proceedings of the ieee/acm 42nd international conference on software engineering workshops* (p. 679-686).

Barriga, J. A., Clemente, P. J., Sosa-Sanchez, E., & Prieto, A. E. (2021). Simulateiot: Domain specific language to design, code generation and execute iot simulation environments. *IEEE Access*, *9*, 92531-92552. Retrieved from www.scopus.com (Cited By :5)

Batot, E., & Sahraoui, H. (n.d.). A generic framework for model-set selection for the unification of testing and learning mde tasks [Conference Proceedings]. In *Proceedings of the acm/ieee 19th international conference on model driven engineering languages and systems* (p. 374-384).

Bergerman, M., Billingsley, J., Reid, J., & van Henten, E. (2016). Robotics in agriculture and forestry [Book Section]. In *Springer handbook of robotics* (p. 1463-1492). Springer.

Betty blocks. (n.d.). Retrieved from https://www.bettyblocks.com/ (Accessed: 2023-01-14)

Bhakta, I., Phadikar, S., & Majumder, K. (2019). State-of-the-art technologies in precision agriculture: a systematic review. *Journal of the Science of Food and Agriculture*, *99*(11), 4878–4888.

Bogue, R. (2016). Robots poised to revolutionise agriculture [Journal Article]. *Industrial Robot: An International Journal*.

Bonacin, R., Nabuco, O. F., & Junior, I. P. (2013). Conceptualizing the impacts of agriculture on water resources: Experiences and ontology engineering challenges. In *Proceedings of the fifth international conference on management of emergent digital ecosystems* (p. 262–269). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/2536146.2536185 doi: 10.1145/2536146.2536185

Borowski, S. (2019). The effect of the change in the composition of the substrate in the agricultural biogas plant on the logistics of maize chaff. In *Tae 2019 - proceeding of 7th international conference on trends in agricultural engineering 2019* (p. 71-74). Retrieved from www.scopus.com

Bose, B., Priya, J., Welekar, S., & Gao, Z. (2020). Hemp disease detection and classification using machine learning and deep learning. In *2020 ieee intl conf on parallel distributed processing with applications, big data cloud computing, sustainable computing communications, social computing networking (ispa/bd-cloud/socialcom/sustaincom)* (p. 762-769). doi: 10.1109/ISPA-BDCloud-SocialCom-SustainCom51426.2020.00121

Boubin, J., Chumley, J., Stewart, C., & Khanal, S. (2019). Autonomic computing challenges in fully autonomous precision agriculture. In *2019 ieee international conference on autonomic computing (icac)* (p. 11-17). doi: 10.1109/ICAC.2019.00012

Bucaioni, A., Cicchetti, A., & Ciccozzi, F. (2022). Modelling in low-code development: a multi-vocal systematic review. *Software and Systems Modeling*, 1–23.

Bucchiarone, A., Cabot, J., Paige, R. F., & Pierantonio, A. (2020). Grand challenges in model-driven engineering: an analysis of the state of the research [Journal Article]. *Software and Systems Modeling*, *19*(1), 5-13.

Buttafuoco, G., & Lucà, F. (2016). The contribution of geostatistics to precision agriculture [Journal Article]. *Ann. Agric. Crop Sci*, *1*, 1008-1009.

Cabot, J. (2020). Positioning of the low-code movement within the field of model-driven engineering. In *Proceedings of the 23rd acm/ieee international conference on model driven engineering languages and systems: Companion proceedings* (pp. 1–3).

Chang, L., & Huang, D. (2010). Development of morphogenesis model-based dynamic growth system in greenhouse muskmelon. In *2010 2nd international workshop on intelligent systems and applications* (p. 1-5). doi: 10.1109/IWISA.2010.5473649

Chen, X., Ou, X.-m., Liang, J.-y., Comite, U., Duarte, N., & Yue, X.-G. (2022). An empirical study on the effect of agricultural industrialization organization on the quality of agricultural products based on factor analysis and ols model. In *2021 4th international conference on e-business, information management and computer science* (p. 526–533). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3511716.3511796 doi: 10.1145/3511716.3511796

Collomb, M., & Hascoët, M. (2008). Extending drag-and-drop to new interactive environments: A multi-display, multi-instrument and multi-user approach [Journal Article]. *Interacting with Computers*, *20*(6), 562-573.

Dahl, T. A., Kendall, A. D., & Hyndman, D. W. (2018). Impacts of projected climate change on sediment yield and dredging costs. *Hydrological Processes*, *32*(9), 1223-1234. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1002/hyp.11486 doi: https://doi.org/10.1002/hyp.11486

Daniela, C., Radu, S., & Codruta, J. (2020). Kinematic modelling of the tracking system for parabolic trough collector. In *2020 7th international conference on energy efficiency and agricultural engineering, ee and ae 2020 - proceedings.* Retrieved from www.scopus.com

Di Cosmo, V., Hyland, M., & Llop, M. (2014). Disentangling water usage in the european union: A decomposition analysis. *Water Resources Management*, *28*(5), 1463-1479. Retrieved from `www.scopus.com` (Cited By :8)

Dimitriadis, S., & Goumopoulos, C. (2008). Applying machine learning to extract new knowledge in precision agriculture applications. In *2008 panhellenic conference on informatics* (p. 100-104). doi: 10.1109/PCI.2008.30

Dobermann, A., Blackmore, S., Cook, S. E., Adamchuk, V. I., et al. (2004). Precision farming: challenges and future directions. In *Proceedings of the 4th international crop science congress* (Vol. 26).

Doering, D., Vizzotto, M. R., Bredemeier, C., da Costa, C. M., Henriques, R. V. B., Pignaton, E., & Pereira, C. E. (2016). Mde-based development of a multispectral camera for precision agriculture. In *Ifac-papersonline* (Vol. 49, p. 24-29). Retrieved from `www.scopus.com` (Cited By :7)

Drenjanac, D., Tomic, S., & Hinterhofer, T. (n.d.). User interactions and network monitoring ease decision-making in a robotic fleet for precision agriculture [Conference Proceedings]. In *Proceedings of the second international conference on robotics and associated high-technologies and equipment for agriculture and forestry (rhea 2014), madrid, spain* (p. 21-23).

Dées, S., & Güntner, J. (2016). Forecasting inflation across euro area countries and sectors: A panel var approach. *Journal of Forecasting*, *36*(4), 431-453. Retrieved from `www.scopus.com` (Cited By :8)

Díez, C. (2017). Hacia una agricultura inteligente [Journal Article]. *Cuaderno de campo*(60), 4-11.

Eigenbrode, S. D., Patrick Binns, W., & Huggins, D. R. (2018). Confronting climate change challenges to dryland cereal production: A call for collaborative, transdisciplinary research, and producer engagement. *Frontiers in Ecology and Evolution*, *5*(JAN). Retrieved from `www.scopus.com` (Cited By :8)

Elijah, O., Rahman, T. A., Orikumhi, I., Leow, C. Y., & Hindia, M. N. (2018). An overview of internet of things (iot) and data analytics in agriculture: Benefits and challenges [Journal Article]. *IEEE Internet of Things Journal*, *5*(5), 3758-3773.

Elsayed, S. M., Oumeraci, H., & Goseberg, N. (2018). Erosion and breaching of coastal barriers in a changing climate: Associated processes and implication for contamination of coastal aquifers. In *Proceedings of the coastal engineering conference* (Vol. 36, chap. 2018). Retrieved from `www.scopus.com` (Cited By :1)

Florin, T. M., & Florin, I. (2020). Air source heat pump performance evaluation based on experimental measurements and neural network. In *2020 7th international conference on energy efficiency and agricultural engineering, ee and ae 2020 - proceedings.* Retrieved from `www.scopus.com` (Cited By :1)

Foufoula-Georgiou, E., Takbiri, Z., Czuba, J. A., & Schwenk, J. (2015). The change of nature and the nature of change in agricultural landscapes: Hydrologic regime shifts modulate ecological transitions. *Water Resources Research*, *51*(8), 6649-6671. Retrieved from `https://onlinelibrary.wiley.com/doi/abs/10.1002/2015WR017637` doi: https://doi.org/10.1002/2015WR017637

Frank, L., Wiegman, C., Davis, J., & Shearer, S. (2021a). Confidence-driven hierarchical classification of cultivated plant stresses. In *Proceedings - 2021 ieee winter conference on applications of computer vision, wacv 2021* (p. 2502-2511). Retrieved from `www.scopus.com`

Frank, L., Wiegman, C., Davis, J., & Shearer, S. (2021b). Confidence-driven hierarchical classification of cultivated plant stresses. In *2021 ieee winter conference on applications of computer vision (wacv)* (p. 2502-2511). doi: 10.1109/WACV48630.2021.00255

Frank, U., Maier, P., & Bock, A. (2021). *Low code platforms: promises, concepts and prospects. a comparative study of ten systems* (Tech. Rep.). ICB-Research Report.

Gao, X., Huang, J., Ma, H., Zhuo, W., & Zhu, D. (2018). Regional winter wheat maturity date prediction using remote sensing-crop model data assimilation and numerical weather prediction. In *2018 7th international conference on agro-geoinformatics (agro-geoinformatics)* (p. 1-5). doi: 10.1109/Agro-Geoinformatics.2018.8476094

Garousi, V., Felderer, M., & Mäntylä, M. V. (2019). Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information and Software Technology*, *106*, 101–121.

Gayatri, M., Jayasakthi, J., & Mala, G. A. (n.d.). Providing smart agricultural solutions to farmers for better yielding using iot [Conference Proceedings]. In *2015 ieee technological innovation in ict for agriculture and rural development (tiar)* (p. 40-43). IEEE.

Ghandar, A., Theodoropoulos, G., Zheng, B., Chen, S., Gong, Y., & Zhong, M. (2018). A dynamic data driven application system to manage urban agricultural ecosystems in smart cities. In *2018 4th international conference on universal village (uv)* (p. 1-7). doi: 10.1109/UV.2018.8642114

Ghandar, A., Theodoropoulos, G., Zhong, M., Zhen, B., Chen, S., Gong, Y., & Ahmed, A. (2019). An agent-based modelling framework for urban agriculture. In *Proceedings of the winter simulation conference* (p. 1767–1778). IEEE Press.

Groeneveld, D., Tekinerdogan, B., Garousi, V., & Catal, C. (2021). A domain-specific language framework for farm management information systems in precision agriculture. *Precision Agriculture*, *22*(4), 1067-1106. Retrieved from `www.scopus.com` (Cited By :5)

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions [Journal Article]. *Future generation computer systems*, *29*(7), 1645-1660.

Han, B., Xu, Y., Yao, J., Su, H., Ping, L., & Zhao, Y. (2016). Kinematic analysis of 2prc-prs parallel platform and development of control system. *Nongye Gongcheng Xuebao/Transactions of the Chinese Society of Agricultural Engineering*, *32*(14), 30-38. Retrieved from `www.scopus.com` (Cited By :1)

He, J., Zhu, J., Luo, X., Zhang, Z., Hu, L., & Gao, Y. (2019). Design of steering control system for rice transplanter equipped with steering wheel-like motor. *Nongye Gongcheng Xuebao/Transactions of the Chinese Society of Agricultural Engineering*, *35*(6), 10-17. Retrieved from `www.scopus.com` (Cited By :10)

Hennicker, R., Bauer, S. S., Janisch, S., & Ludwig, M. (2010). A generic framework for multi-disciplinary environmental modelling. In *Modelling for environment's sake: Proceedings of the 5th biennial conference of the international environmental modelling and software society, iemss 2010* (Vol. 2, p. 980-994). Retrieved from `www.scopus.com` (Cited By :13)

Henriques, H., Lourenço, H., Amaral, V., & Goulão, M. (n.d.). Improving the developer experience with a low-code process modelling language [Conference Proceedings]. In *Proceedings of the 21th acm/ieee international conference on model driven engineering languages and systems* (p. 200-210).

Himesh, S., Prakasa Rao, E., Gouda, K., Ramesh, K., Rakesh, V., & Mohapatra, G. (2018). Digital revolution and big data: A new revolution in agriculture [Journal Article]. *CAB Rev*, *13*(21), 1-7.

Honda, K., Ines, A. V. M., Yui, A., Witayangkurn, A., Chinnachodteeranun, R., & Teeravech, K. (2014). Agriculture information service built on geospatial data infrastructure and crop modeling. In *Proceedings of the 2014 international workshop on web intelligence and smart sensing* (p. 1–9). New York, NY, USA: Association for Computing Machinery. Retrieved from `https://doi.org/10.1145/2637064.2637094` doi: 10.1145/2637064.2637094

Hong, Z., Kalbarczyk, Z., & Iyer, R. K. (2016). A data-driven approach to soil moisture collection and prediction. In *2016 ieee international conference on smart computing (smartcomp)* (p. 1-6). doi: 10.1109/SMARTCOMP.2016.7501673

Hu, X., Ye, H., & He, X. (2022). Research on the two-way interactive development model of industrial chain in agricultural logistics park. In *2021 4th international conference on e-business, information management and computer science* (p. 191–196). New York, NY, USA: Association for Computing Machinery. Retrieved from `https://doi.org/10.1145/3511716.3511747` doi: 10.1145/3511716.3511747

Ihirwe, F., Di Ruscio, D., Mazzini, S., Pierini, P., & Pierantonio, A. (n.d.). Low-code engineering for internet of things: A state of research [Conference Proceedings]. In *Proceedings of the 23rd acm/ieee international conference on model driven engineering languages and systems: Companion proceedings* (p. 1-8).

Ismatovna, T. D., Ikramovich, K. A., Djaxanovich, I. S., & Farhodovich, M. F. (2019). Dynamic modeling of vibrating system n-s component parts which of that the mobile machines for fast acting pneumatic actuator with self-damping. In *2019 international conference on information science and communications technologies (icisct)* (p. 1-5). doi: 10.1109/ICISCT47635.2019.9012010

Jain, N., Awasthi, Y., & Jain, R. (2021). Ubiquitous sensor based intelligent system for net houses. In *2021 international conference on computing, communication, and intelligent systems (icccis)* (p. 219-225). doi: 10.1109/ICCCIS51004.2021.9397080

Jarolimek, J., Stočes, M., Masner, J., Vaněk, J., Šimek, P., Pavlik, J., & Rajtr, J. (2017). User-technological index of precision agriculture. *AGRIS on-line Papers in Economics and Informatics*, *9*(665-2017-1569), 69–75.

Jayaraman, P. P., Palmer, D., Zaslavsky, A., & Georgakopoulos, D. (n.d.). Do-it-yourself digital agriculture applications with semantically enhanced iot platform [Conference Proceedings]. In *2015 ieee tenth international conference on intelligent sensors, sensor networks and information processing (issnip)* (p. 1-6). IEEE.

Jayaraman, P. P., Yavari, A., Georgakopoulos, D., Morshed, A., & Zaslavsky, A. (2016). Internet of things platform for smart farming: Experiences and lessons learnt [Journal Article]. *Sensors*, *16*(11), 1884.

Jiang, W., Wang, Y., & Qi, J. (2019). Study on the integrated model of modern agricultural variety breeding in the internet of things environment. In *Proceedings of the 2019 annual meeting on management engineering* (p. 198–201). New York, NY, USA: Association for Computing Machinery. Retrieved from `https://doi.org/10.1145/3377672.3378057` doi: 10.1145/3377672.3378057

Jie, L., Pengfei, Z., Min, L., Xiaoling, D., & Zhen, W. (2011). Study on the application of fuzzy-pid control to precision seeding technology of wheat. In *2011 second international conference on mechanic automation and control engineering* (p. 1690-1693). doi: 10.1109/MACE.2011.5987281

Jing-lan, R., & Xiang-fa, C. (2011). Impact of roller sheller parameters on the characteristics of the load. In *2011 international conference on new technology of agricultural* (p. 139-143). doi: 10.1109/ICAE.2011.5943768

Ju, X.-D., Xu, J., & Sun, Z. S. (2018). Crispr editing in biological and biomedical investigation. *Journal of Cellular Biochemistry*, *119*(1), 52-61. Retrieved from `https://onlinelibrary.wiley.com/doi/abs/10.1002/jcb.26154` doi: https://doi.org/10.1002/jcb.26154

Kaitovic, I., Rezende, R., Murillo, C., & Fantuzzi, C. (2012). Model-driven approach to design ict infrastructure for precision farming. In *Proceedings of 2012 ieee 17th international conference on emerging technologies factory automation (etfa 2012)* (p. 1-4). doi: 10.1109/ETFA.2012.6489709

Kallioniemi, P., Rajput, N., Mahajan, G., & Turunen, M. (2012). Data-model for voice search of agricultural information system. In *Proceedings of the first workshop on information and knowledge management for developing region* (p. 11–16). New York, NY, USA: Association for Computing Machinery. Retrieved from `https://doi.org/10.1145/2389776.2389780` doi: 10.1145/2389776.2389780

Kassim, J. M., & Abdullah, R. (2012). Advisory system architecture in agricultural environment to support decision making process. In *2012 second international conference on digital information and communication technology and it's applications (dictap)* (p. 453-456). doi: 10.1109/DICTAP.2012.6215427

Kawtrakul, A., Chanlekha, H., Waiyamai, K., Kangkachit, T., D'Orazio, L., Kotzinos, D., ... Spyratos, N. (2021). Towards data-and-innovation driven sustainable and productive agriculture: Bio-agri-watch as a use case study. In *Proceedings - 2021 ieee international conference on big data, big data 2021* (p. 3407-3415). Retrieved from www.scopus.com

Kim, J., & Park, J. (2015). Mathematical modeling of coastal marine environments using observational data for coastal management. *Ocean and Coastal Management*, *116*, 396-403. Retrieved from www.scopus.com (Cited By :4)

Kissflow. (n.d.). Retrieved from https://kissflow.com/ (Accessed: 2023-01-14)

Kitchenham, B., & Brereton, P. (2013). A systematic review of systematic review process research in software engineering. *Information and software technology*, *55*(12), 2049–2075.

Kolovos, D. S., Paige, R. F., & Polack, F. A. (n.d.). The grand challenge of scalability for model driven engineering [Conference Proceedings]. In *International conference on model driven engineering languages and systems* (p. 48-53). Springer.

Kolovos, D. S., Rose, L. M., Matragkas, N., Paige, R. F., Guerra, E., Cuadrado, J. S., ... Tisi, M. (n.d.). A research roadmap towards achieving scalability in model driven engineering [Conference Proceedings]. In *Proceedings of the workshop on scalability in model driven engineering* (p. 1-10).

Lal, R. (1991). Soil structure and sustainability. *Journal of sustainable agriculture*, *1*(4), 67–92.

Lethbridge, T. C. (2021). Low-code is often high-code, so we must design low-code platforms to enable proper software engineering. In *International symposium on leveraging applications of formal methods* (pp. 202–212).

Li, J., & Chen, X. (2022). Research on on-line marketing model of agricultural products based on data mining algorithm. In *Proceedings of the 3rd asia-pacific conference on image processing, electronics and computers* (p. 745–749). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3544109.3544346 doi: 10.1145/3544109.3544346

Liu, L., Zhang, Y., Ding, W., & Kang, M. (2017). Design of multi-objective driven product family shape gene for tractor. *Nongye Gongcheng Xuebao/Transactions of the Chinese Society of Agricultural Engineering*, *33*(17), 82-90. Retrieved from www.scopus.com (Cited By :1)

Liu, M., & Yao, Y. (2021). Exploration and practice of talent cultivation model of electrical automation technology major driven by real agricultural application. In *The 4th international conference on information technologies and electrical engineering.* New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3513142.3513233 doi: 10.1145/3513142.3513233

Lundell, B., & Lings, B. (2004). Changing perceptions of case technology [Journal Article]. *Journal of Systems and Software*, *72*(2), 271-280.

Lyu, S., Li, Z., Lu, S., Wang, W., & Sun, D. (2019). Characteristic course system establishment for electronic and information engineering specialty in agricultural university. In *Proceedings of the 2019 international conference on artificial intelligence and computer science* (p. 578–582). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3349341.3349471 doi: 10.1145/3349341.3349471

Malheiro, M. T., Pereira, R. M. S., Gonçalves, A. M., Pereira, P. A. S., Caldeira, A. C., & Lopes, S. O. (2019). Hands on experiments about water needs in agriculture and their mathematical modelling under climate change. In *Proceedings of the 2019 3rd international conference on e-society, e-education and e-technology* (p. 131–135). New York, NY, USA: Association for Computing Machinery. Retrieved from `https://doi.org/10.1145/3355966.3358411` doi: 10.1145/3355966.3358411

Manjula, A., & Narsimha, G. (2015). Xcypf: A flexible and extensible framework for agricultural crop yield prediction. In *2015 ieee 9th international conference on intelligent systems and control (isco)* (p. 1-5). doi: 10.1109/ISCO.2015.7282311

McBratney, A., Whelan, B., Ancev, T., & Bouma, J. (2005). Future directions of precision agriculture. *Precision agriculture*, *6*(1), 7–23.

Mendix. (n.d.). Retrieved from `https://www.mendix.com/` (Accessed: 2023-01-14)

Microsoft power apps. (n.d.). Retrieved from `https://powerapps.microsoft.com/` (Accessed: 2023-01-14)

Miyasaka, J., Yamanaka, Y., Nakagawa, S., Ohdoi, K., Nakashima, H., Shimizu, H., ... Mitani, T. (2014). Development of an electric vehicle by microwave power transmission: Development of small model vehicle and control of rectenna panel. *Engineering in Agriculture, Environment and Food*, *7*(2), 103-108. Retrieved from `www.scopus.com` (Cited By :1)

Moisescu, M. A., Sacala, I. S., Dumitrache, I., & Repta, D. (2017). Cyber physical systems based model-driven development for precision agriculture. In *Proceedings of the symposium on model-driven approaches for simulation engineering.* San Diego, CA, USA: Society for Computer Simulation International.

Moore, A. D., Eckard, R. J., Thorburn, P. J., Grace, P. R., Wang, E., & Chen, D. (2014). Mathematical modeling for improved greenhouse gas balances, agro-ecosystems, and policy development: lessons from the australian experience. *WIREs Climate Change*, *5*(6), 735-752. Retrieved from `https://onlinelibrary.wiley.com/doi/abs/10.1002/wcc.304` doi: https://doi.org/10.1002/wcc.304

Morales-Tapia, A., & Ruz-Ramírez, A. (2016). Computational modeling of auxin: A foundation for plant engineering. *Frontiers in Plant Science*, *7*(DECEMBER2016). Retrieved from `www.scopus.com` (Cited By :7)

Muccini, H., Spalazzese, R., Moghaddam, M. T., & Sharaf, M. (2018). Self-adaptive iot architectures: An emergency handling case study. In *Acm international conference proceeding series.* Retrieved from `www.scopus.com` (Cited By :23)

Nawaz, A., Arora, A. S., Ali, W., Saxena, N., Khan, M. S., Yun, C. M., & Lee, M. (2022). Intelligent human–machine interface: An agile operation and decision support for an anammox sbr system at a pilot-scale wastewater treatment plant. *IEEE Transactions on Industrial Informatics*, *18*(9), 6224-6232. doi: 10.1109/TII.2022.3153468

Nicoletti, J., De Miranda, J. H., Cooke, R., Christianson, L., & De Oliveira, L. A. (2021). Efficiency analysis of machine learning models for simulating nitrate movement in soils from illinois state. In *American society of agricultural and biological engineers annual international meeting, asabe 2021* (Vol. 4, p. 2113-2122). Retrieved from `www.scopus.com`

Nikkilä, R., Seilonen, I., & Koskinen, K. (2010). Software architecture for farm management information systems in precision agriculture. *Computers and electronics in agriculture*, *70*(2), 328–336.

Niu, D., Wang, M., Yuan, H., & Xu, W. (2016). Event-driven data mining methods for large-scale market prediction: A case study of an agricultural products company. In *Proceedings of the second acm sigspatialinternational workshop on the use of gis in emergency management.* New York, NY, USA: Association for Computing Machinery. Retrieved from `https://doi.org/10.1145/3017611.3017618` doi: 10.1145/3017611.3017618

Now platform by servicenow. (n.d.). Retrieved from https://www.servicenow.com/ (Accessed: 2023-01-14)

Outsystems. (n.d.). Retrieved from https://www.outsystems.com/ (Accessed: 2023-01-14)

Pate, M. L., Beard, F. R., Miller, B., Miller, R., Hatch, R., & Hall, K. (2015). Project based learning: Integration across agricultural systems technology courses. In *American society of agricultural and biological engineers annual international meeting 2015* (Vol. 1, p. 231-237). Retrieved from www.scopus.com (Cited By :1)

Plazas, J. E., Bimonte, S., De Sousa, G., & Corrales, J. C. (2019). Data-centric uml profile for wireless sensors: Application to smart farming. *International Journal of Agricultural and Environmental Information Systems*, *10*(2), 21-48. Retrieved from www.scopus.com (Cited By :2)

Po Shun Chen, A., & Wu Liu, C. (2021). Crafting asr and conversational models for an agriculture chatbot. In *2021 the 4th international conference on computational intelligence and intelligent systems* (p. 61–66). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3507623.3507634 doi: 10.1145/3507623.3507634

Punjabi, H. C., Agarwal, S., Khithani, V., Muddaliar, V., & Vasmatkar, M. (2017). Smart farming using iot [Journal Article]. *International Journal of Electronics and Communication Engineering and Technology*, *8*(1), 58-66.

Qu, J., Guo, K., Gao, H., Song, S., Li, Y., & Zhou, W. (2018). Experiments on collaborative control characteristics of driving and steering for agricultural flexible chassis based on pwm signal. *Nongye Gongcheng Xuebao/Transactions of the Chinese Society of Agricultural Engineering*, *34*(7), 75-81. Retrieved from www.scopus.com (Cited By :4)

Quick base. (n.d.). Retrieved from https://www.quickbase.com/ (Accessed: 2023-01-14)

Racine Ly, O., Gueye, K., Ouya, S., & Mendy, G. (2021). Collaborative platform for supervision and advice of agricultural engineer to farmers thanks to iot and webrtc. In *The 2021 9th international conference on computer and communications management* (p. 143–148). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3479162.3479184 doi: 10.1145/3479162.3479184

Ramos, D. B., Loubach, D. S., & Da Cunha, A. M. (2010). Developing a distributed real-time monitoring system to track uavs. *IEEE Aerospace and Electronic Systems Magazine*, *25*(9), 18-25. Retrieved from www.scopus.com (Cited By :9)

Ribeiro, A., Diaz, B., & Garcia-Alegre, M. (2003). Extracting fuzzy rules to describe weed infestations in terms of soil factors. In *The 12th ieee international conference on fuzzy systems, 2003. fuzz '03.* (Vol. 2, p. 1032-1037 vol.2). doi: 10.1109/FUZZ.2003.1206573

Richardson, C., & Rymer, J. R. (2016a). The forrester wave™: Low-code development platforms [Journal Article]. *Forrester, Cambridge, MA, USA.*

Richardson, C., & Rymer, J. R. (2016b). Vendor landscape: The fractured, fertile terrain of low-code application platforms [Journal Article]. *FORRESTER, Janeiro.*

Richardson, C., Rymer, J. R., Mines, C., Cullen, A., & Whittaker, D. (2014). New development platforms emerge for customer-facing applications [Journal Article]. *Forrester: Cambridge, MA, USA.*

Rio, A., Maurel, Y., Barais, O., & Bugni, Y. (2019). Benefits of energy management systems on local energy efficiency, an agricultural case study. In *2019 ieee international conference on communications, control, and computing technologies for smart grids, smartgridcomm 2019.* Retrieved from www.scopus.com (Cited By :2)

Roopa, V., & Shyni, C. E. (2019). Data driven approach for farm re-modeling using prediction analytics. In *2019 international conference on computational intelligence and knowledge economy (iccike)* (p. 209-214). doi: 10.1109/ICCIKE47802.2019.9004309

R Shamshiri, R., Weltzien, C., Hameed, I. A., J Yule, I., E Grift, T., Balasundram, S. K., … Chowdhary, G. (2018). Research and development in agricultural robotics: A perspective of digital farming [Journal Article].

Rymer, J. R., Richardson, C., Mines, C., Jones, D., Whittaker, D., Miller, J., & McPherson, I. (2015). Low-code platforms deliver customerfacing apps fast, but will they scale up? [Journal Article]. *Forrester Research*.

Sahay, A., Indamutsa, A., Di Ruscio, D., & Pierantonio, A. (n.d.). Supporting the understanding and comparison of low-code development platforms [Conference Proceedings]. In *2020 46th euromicro conference on software engineering and advanced applications (seaa)* (p. 171-178). IEEE.

Sahu, S., Chawla, M., & Khare, N. (2017). An efficient analysis of crop yield prediction using hadoop framework based on random forest approach. In *2017 international conference on computing, communication and automation (iccca)* (p. 53-57). doi: 10.1109/CCAA.2017.8229770

Sathya, A., & Jayalalitha, S. (2012). Engineering analogs for conservational decisions in restoring abandoned traditional rice varieties in tamil nadu. In *2012 - international conference on management issues in emerging economies (icmiee)* (p. 79-84).

Sayyah, P., Lazarescu, M. T., BOcchio, S., Ebeid, E., Palermo, G., Quaglia, D., … Lavagno, L. (2015). Virtual platform-based design space exploration of power-efficient distributed embedded applications. *ACM Transactions on Embedded Computing Systems*, *14*(3). Retrieved from `www.scopus.com` (Cited By :13)

Sharma, A., Jain, A., Gupta, P., & Chowdary, V. (2021). Machine learning applications for precision agriculture: A comprehensive review. *IEEE Access*, *9*, 4843-4873. doi: 10.1109/ACCESS.2020.3048415

Shi, B., Zeng, T., Tang, C., Zhang, L., Xie, Z., Lv, G., & Wu, Q. (2021). Landslide risk assessment using granular fuzzy rule-based modeling: A case study on earthquake-triggered landslides. *IEEE Access*, *9*, 135790-135802. Retrieved from `www.scopus.com`

Shi, Y., Xi, X., Wu, F., Zhang, R., Jin, Y., & Gong, J. (2019). Design and experimental study on motor type automatic driving system of agricultural machinery based on beidou navigation. In *Proceedings of the 2019 5th international conference on robotics and artificial intelligence* (p. 98–102). New York, NY, USA: Association for Computing Machinery. Retrieved from `https://doi.org/10.1145/3373724.3373743` doi: 10.1145/3373724.3373743

Sitorus, A., & Sartika, T. D. (2017). Analysis of design combine transplanter power by hand tractor: Transmission system and furrow opener. In *2017 international conference on computing, engineering, and design (icced)* (p. 1-6). doi: 10.1109/CED.2017.8308115

Sivasothy, P., Keksel, A., & Seewig, J. (2018). Technical realisation of availability-oriented business models for potato harvesters by using amr sensors. In *2018 ieee sensors* (p. 1-4). doi: 10.1109/ICSENS.2018.8630306

Spieldenner, T., Byelozyorov, S., Guldner, M., & Slusallek, P. (2017). Fives: An aspect-oriented virtual environment server. In *Proceedings - 2017 international conference on cyberworlds, cw 2017 - in cooperation with: Eurographics association international federation for information processing acm siggraph* (Vol. 2017-January, p. 103-110). Retrieved from `www.scopus.com` (Cited By :5)

Spieldenner, T., Byelozyorov, S., Guldner, M., & Slusallek, P. (2018). Fives: an aspect-oriented approach for shared virtual environments in the web. *Visual Computer*, *34*(9), 1269-1282. Retrieved from `www.scopus.com` (Cited By :6)

Sun, J., Wang, M., Pei, H., Chen, Z., & Yang, Y. (2016). Construction and evaluation of modern agricultural processing industrial upgrading model. *Nongye Gongcheng Xuebao/Transactions of the Chinese Society of Agricultural Engineering*, *32*(21), 13-20. Retrieved from `www.scopus.com` (Cited By :3)

Talesra, K., & Nagaraja, G. (2021). Low-code platform for application development. *International Journal of Applied Engineering Research*, *16*(5), 346–351.

Tan, L., Haley, R., Wortman, R., & Zhang, Q. (2012a). An extensible and integrated software architecture for data analysis and visualization in precision agriculture. In *Proceedings of the 2012 ieee 13th international conference on information reuse and integration, iri 2012* (p. 271-278). Retrieved from `www.scopus.com` (Cited By :9)

Tan, L., Haley, R., Wortman, R., & Zhang, Q. (2012b). An extensible and integrated software architecture for data analysis and visualization in precision agriculture. In *2012 ieee 13th international conference on information reuse integration (iri)* (p. 271-278). doi: 10.1109/IRI.2012.6303020

Tang, J. (2021). Coupling dynamic model of natural resources environment and agricultural environmental pollution based on big data. In *2021 3rd international conference on artificial intelligence and advanced manufacture* (p. 1815–1819). New York, NY, USA: Association for Computing Machinery. Retrieved from `https://doi.org/10.1145/3495018.3495491` doi: 10.1145/3495018.3495491

Tien, J. M. (2013). Big data: Unleashing information. *Journal of Systems Science and Systems Engineering*, *22*(2), 127-151. Retrieved from `www.scopus.com` (Cited By :141)

Tisi, M., Mottu, J.-M., Kolovos, D., De Lara, J., Guerra, E., Di Ruscio, D., ... Wimmer, M. (n.d.). Lowcomote: Training the next generation of experts in scalable low-code engineering platforms [Conference Proceedings]. In *Staf 2019 co-located events joint proceedings: 1st junior researcher community event, 2nd international workshop on model-driven engineering for design-runtime interaction in complex systems, and 1st research project showcase workshop co-located with software technologies: Applications and foundations (staf 2019).*

Toyama, S., & Yamamoto, G. (2009). Development of wearable-agri-robot mechanism for agricultural work. In *2009 ieee/rsj international conference on intelligent robots and systems* (p. 5801-5806). doi: 10.1109/IROS.2009.5354265

Tweel, A. W., & Turner, R. E. (2012). Watershed land use and river engineering drive wetland formation and loss in the mississippi river birdfoot delta. *Limnology and Oceanography*, *57*(1), 18-28. Retrieved from `https://onlinelibrary.wiley.com/doi/abs/10.4319/lo.2012.57.1.0018` doi: https://doi.org/10.4319/lo.2012.57.1.0018

Tzounis, A., Katsoulas, N., Bartzanas, T., & Kittas, C. (2017). Internet of things in agriculture, recent advances and future challenges [Journal Article]. *Biosystems engineering*, *164*, 31-48.

Uehara, H., Iuchi, Y., Fukazawa, Y., & Kaneta, Y. (2022). Predicting a growing stage of rice plants based on the cropping records over 25 years — a trial of feature engineering incorporating hidden regional characteristics. *IEICE Transactions on Information and Systems*, *E105D*(5), 955-963. Retrieved from `www.scopus.com`

Urbieta, M., Firmenich, S., Maglione, P., Rossi, G., & Olivero, M. A. (2017). A model-driven approach for empowering advanceweb augmentation: From client-side to server-side support. In *Webist 2017 - proceedings of the 13th international conference on web information systems and technologies* (p. 444-454). Retrieved from `www.scopus.com` (Cited By :1)

Vaishnavi., S., Shobana., M., Sabitha., R., & Karthik., S. (2021). Agricultural crop recommendations based on productivity and season. In *2021 7th international conference on advanced computing and communication systems (icaccs)* (Vol. 1, p. 883-886). doi: 10.1109/ICACCS51430.2021.9441736

Valckenaers, P., Van Belle, J., & Ali, O. (2011). Prosa and delegate mas for open-air engineering processes. In *Ieee international conference on emerging technologies and factory automation, etfa.* Retrieved from www.scopus.com (Cited By :3)

Varela-Vaca, J., Rosado, D. G., Sánchez, L. E., Gómez-López, M. T., Gasca, R. M., & Fernández-Medina, E. (2021). Carmen: A framework for the verification and diagnosis of the specification of security requirements in cyber-physical systems. *Computers in Industry*, *132*. Retrieved from www.scopus.com (Cited By :1)

Vincent, P., Baker, V., Natis, Y., Iijima, K., Driver, M., Dunie, R., ... Gupta, A. (2018). *Magic quadrant for enterprise high-productivity application platform as a service* (Report). Technical report, Gartner.

Wang, W., Qin, Y., & Han, Z. (2022). Research and design of a fruit target localization and segmentation method based on deep learning techniques. In *Proceedings of spie - the international society for optical engineering* (Vol. 12254). Retrieved from www.scopus.com

Wang, Y., Hills, A., & Blatt, M. R. (2014). Systems analysis of guard cell membrane transport for enhanced stomatal dynamics and water use efficiency. *Plant Physiology*, *164*(4), 1593-1599. Retrieved from www.scopus.com (Cited By :48)

Waszkowski, R. (2019). Low-code platform for automating business processes in manufacturing. *IFAC-PapersOnLine*, *52*(10), 376–381.

Whittle, J., Hutchinson, J., & Rouncefield, M. (2013). The state of practice in model-driven engineering [Journal Article]. *IEEE software*, *31*(3), 79-85.

Wijekoon, H., Schegolev, B., & Merunka, V. (2020). Knowledge representation of software design patterns: A model transformations perspective. In *Ceur workshop proceedings* (Vol. 2761, p. 179-187). Retrieved from www.scopus.com

Wilson, R. S. I., Ginige, A., Goonetillake, J. S., & Indika, W. A. (2019). User needs-driven enrichment of ontology: A case study in sri lankan agriculture. In *Proceedings of the 2019 7th international conference on information technology: Iot and smart city* (p. 581–586). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3377170.3377279 doi: 10.1145/3377170.3377279

Wu, Y., Wang, S., Bezemer, C.-P., & Inoue, K. (2019). How do developers utilize source code from stack overflow? [Journal Article]. *Empirical Software Engineering*, *24*(2), 637-673.

Yao, Z., Liu, G., Zhang, D., Shen, Y., & Wang, Z. (2020). Path tracking control for four-wheel-independent-driven agricultural high clearance sprayer with new front-rear-dual-steering-axle. In *2020 chinese automation congress (cac)* (p. 2966-2972). doi: 10.1109/CAC51589.2020.9326787

Yu, Z., Amin, S. U., Alhussein, M., & Lv, Z. (2021). Research on disease prediction based on improved deepfm and iomt. *IEEE Access*, *9*, 39043-39054. doi: 10.1109/ACCESS.2021.3062687

Yuan, C. Z., & Ling, S. K. (2020). Long short-term memory model based agriculture commodity price prediction application. In *Proceedings of the 2020 2nd international conference on information technology and computer communications* (p. 43–49). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3417473.3417481 doi: 10.1145/3417473.3417481

Zečević, I., Bjeljac, P., Perišić, B., Stankovski, S., Venus, D., & Ostojić, G. (2018). Model driven development of hybrid databases using lightweight metamodel extensions. *Enterprise Information Systems*, *12*(8-9), 1221-1238. Retrieved from www.scopus.com (Cited By :7)

Zhang, H., & Zheng, J. (2016). Comparison of straight line driving synchronous control methods and validation of 4wd sprayer chassis with hydraulic power. *Nongye Gongcheng Xuebao/Transactions of the Chinese Society of Agricultural Engineering*, *32*(23), 43-50. Retrieved from www.scopus.com (Cited By :7)

Zhang, S., Wei, C., Shao, J., & Li, Z. (2014). Resources integration and change in land value affected by land transfer and consolidation of rural villages in hilly areas. *Nongye Gongcheng Xuebao/Transactions of the Chinese Society of Agricultural Engineering*, *30*(12), 1-17. Retrieved from `www.scopus.com` (Cited By :6)

Zhang, Y. (2019). The role of precision agriculture [Journal Article]. *Resource Magazine*, *26*(6), 9-9.

Zhu, Z., Lai, L., Sun, X., Chen, L., & Cai, Y. (2021). Design and analysis of a novel mechanic-electronic-hydraulic powertrain system for agriculture tractors. *IEEE Access*, *9*, 153811-153823. doi: 10.1109/ACCESS.2021.3126667

Zoho creator. (n.d.). Retrieved from `https://www.zoho.com/creator/` (Accessed: 2023-01-14)