land

Article

Feasibility of Low-Code Development Platforms in Precision Agriculture: Opportunities, Challenges, and Future Directions

Emin Guresci [1], Bedir Tekinerdogan [1,*], Önder Babur [1,2] and Qingzhi Liu [1]

[1] Information Technology Group, Wageningen University & Research, 6708 PB Wageningen, The Netherlands; emin.guresci@wur.nl (E.G.); onder.babur@wur.nl (Ö.B.); qingzhi.liu@wur.nl (Q.L.)
[2] Department of Mathematics and Computer Science, Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands
* Correspondence: bedir.tekinerdogan@wur.nl

Abstract: Low-Code Development Platforms (LCDPs) empower users to create and deploy custom software with little to no programming. These platforms streamline development, offering benefits like faster time-to-market, reduced technical barriers, and broader participation in software creation, even for those without traditional coding skills. This study explores the application of LCDPs in Precision Agriculture (PA) through a systematic literature review (SLR). By analyzing the general characteristics and challenges of LCDPs, alongside insights from existing PA research, we assess their feasibility and potential impact in agricultural contexts. Our findings suggest that LCDPs can enable farmers and agricultural professionals to create tailored applications for real-time monitoring, data analysis, and automation, enhancing farming efficiency. However, challenges such as scalability, extensibility, data security, and integration with complex IoT systems must be addressed to fully realize the benefits of LCDPs in PA. This study contributes to the growing knowledge base in agricultural technology, offering valuable insights for researchers, practitioners, and policymakers looking to leverage LCDPs for sustainable and efficient farming practices.

Keywords: no code development platform; low code development platform; precision agriculture; software engineering; systematic literature review

check for updates

Citation: Guresci, E.; Tekinerdogan, B.; Babur, Ö.; Liu, Q. Feasibility of Low-Code Development Platforms in Precision Agriculture: Opportunities, Challenges, and Future Directions. *Land* **2024**, *13*, 1758. https://doi.org/10.3390/land13111758

Academic Editors: Safwan Mohammed and Endre Harsányi

Received: 20 September 2024
Revised: 16 October 2024
Accepted: 21 October 2024
Published: 25 October 2024

1. Introduction

Precision farming or Precision Agriculture (PA) is a farming management concept that relies on the use of advanced technology and data-driven approaches to optimize agricultural practices, reduce input, improve yield, and reduce environmental impact [1–3]. PA is crucial for addressing the growing global need for increased food production while maintaining environmental responsibility [4]. It maximizes resource utilization through advanced technology and automation, reducing waste and boosting efficiency [5]. As remote sensors and GPS are used significantly in PA, it could tailor IoT applications for agriculture, providing real-time insights to farmers [6]. However, challenges include the high cost of technology, the need for skilled personnel, and complex data management [7,8]. Despite these obstacles, PA offers significant benefits, particularly in the face of climate change, by optimizing resource usage and ensuring sustainable agriculture [1]. Overcoming these challenges is essential for harnessing its full potential and achieving a balance between productivity and ecological preservation [3,4,9].

Software plays an important role in PA in achieving these goals using software for real-time monitoring, data analysis, and automation [6,10]. As a result, PA is experiencing a growing reliance on software [11]. However, software development in PA is not always trivial. PA increasingly demands smart, complex, and interconnected systems. These typically involve heterogeneous software ecosystems ranging from embedded or on-premise software to cloud services [11]. The complexity of the agricultural settings with high variability in physical and environmental factors [2], the need for specialized knowledge from different fields of expertise (which are typically far from software development) [12], the complex network of stakeholders with interdependent yet distinct and sometimes competing interests [11], and the high cost associated with traditional development, present significant challenges in PA [12].

To cope with these challenges, there is a growing need for higher abstraction in programming that allows for a more declarative and accessible software development [13]. Low-Code Development Platforms (LCDPs) provide indeed a solution for this problem by providing visual interfaces and pre-built components that simplify the software development process. These platforms enable users, including those with limited programming experience, to create customized applications rapidly, making software development more accessible to a broader range of professionals, including those in PA [14]. The integration of LCDPs with PA thus holds significant promise, particularly in overcoming the technical barriers that have traditionally hindered the adoption of advanced software solutions in agriculture [15]. By reducing the need for extensive and dedicated coding knowledge, LCDPs can empower agricultural stakeholders to develop and deploy software applications tailored to their specific needs in PA [16,17]. However, the application of LCDPs in PA by itself is also not without its challenges [18,19]. Issues such as scalability, extensibility, data security, and integration with complex Internet of Things (IoT) systems remain critical hurdles that must be addressed to fully realize the potential of these platforms [6,20–22].

While several studies have examined the aspects and challenges of LCDPs, few have directly addressed their application. This study explores the feasibility of using LCDPs in the PA domain through a systematic literature review (SLR). Initially, the review takes a broad approach to understand the general characteristics and challenges of LCDPs, synthesizing these findings. Concurrently, it identifies key insights from existing studies on PA and those explicitly linking PA and LCDPs to derive lessons for their applicability in the PA domain. This study adds to the expanding knowledge base on agricultural technology, providing valuable insights for researchers, practitioners, and policymakers aiming to leverage LCDPs for sustainable and efficient farming practices.

The paper is further organized as follows: Section 2 describes the research methodology and the literature review process. Section 3 describes the results of the SLR giving answer to the research questions. Section 4 presents the discussion and explains the threats to validity of this research. Section 5 presents the related work dedicated to LCDPs. In Section 6, the conclusion for this research is given.

2. Methodology

In this section, the research methodology will be explained. For this research, we adapted the SLR methodology of [23], as outlined in Figure 1. We have used a similar approach in our previous work, in [24,25], to tailor it for the investigation of software engineering literature, as well as other domains, as in [26,27].

Figure 1 illustrates the steps that were taken during this research. The research questions were identified in step 1 (Table 1), and the search technique and consequent search strings that were determined in step 2. The sources to search (step 2.1) and the choice of search terms (step 2.2) were incorporated in the formulation of the search protocol (step 2) (Table 2). The search technique was described as an iterative procedure in which the results of one search string were utilized to modify the next. This will be covered in further detail in Sections 2.1 and 2.2.

In step 3, we outlined the standards of the selection criteria (Table 3) by which we chose the primary studies by performing study selection (step 3.1). The selection criteria will be further explained in Section 2.3.

In step 4, we used the quality assessment criteria, and evaluated the search results in step 4.1. The selection criteria for the quality assessment are visible in Table 4, and will be further explained in Section 2.4.

In step 5, we developed a standardized data extraction form to retrieve key data from the chosen papers. This was an iterative process in which we used the research questions as our baseline with adjustments made by performing data extraction, and by integrating this into the extraction form in step 5.1. The data extraction form will be further explained in Section 2.5.

In step 6, the procedures for data synthesis and analysis were developed. We collected the papers, which we used to conduct a thorough analysis. As a qualitative assessment methodology, this evaluation was conducted with a summative content analysis approach. Research employing a summative approach to qualitative content analysis begins by identifying and quantifying the occurrence of these specific content or concepts in text to understand the larger context in which these data are used [28]. We used this approach to identify concepts, themes, and patterns about LCDPs in academic literature. Thereafter, the collected data are used for the exploration of theoretical claims regarding LCDPs. This qualitative approach refers to the content analysis, interpretation, and discovery of the underlying meanings of these identified concepts in our study after synthesizing the data extraction results in step 6.1 [28]. This will be further explained in Section 2.6. The data were synthesized, and the discussion was outlined with the final results in step 7.

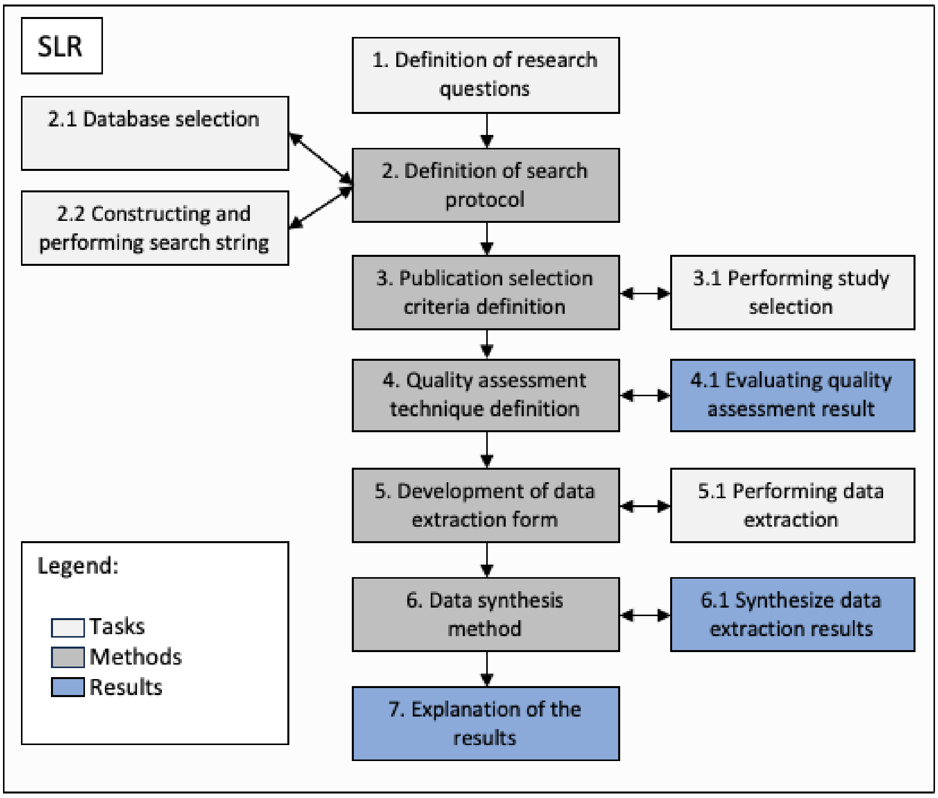

Figure 1. SLR process used in this study.

2.1. *Definition of Research Questions*

Table 1 shows the research questions that are identified with the aim. As can be seen from the research questions, we wish to identify the literature on LCDP and focus on its objectives and challenges for PA. We will discuss the exploration into the development and application of LCDPs specifically designed to meet the unique challenges and requirements of precision agriculture. To answer the research questions, an SLR process is used to integrate these LCDP results in the literature with PA. Furthermore, this research efforts to inform the continuous development and refinement of LCDPs by analyzing the methods provided to address challenges and exploring future perspectives. Finally, this thorough examination of LCDPs will allow researchers, practitioners, and decision-makers to acquire a better grasp of the existing environment, discover possible applications in their respective domains, and make informed choices about the adoption and use of LCDPs.

The first research question (RQ1) is answered by determining the LCDP features in the academic literature. The second research question (RQ2) is addressed by identifying the objectives indicated in the studies where the LCDP is used (RQ2). Concerning the third question, we determined the objectives for LCDPs specifically in PA (RQ3). We obtained a better knowledge of the challenges (RQ4) through posing the fourth question. Lastly, we aim to explore future directions for LCDP regarding the literature (RQ5).

Table 1. Research questions of this study.

RQ	Question	Aim
1	What are the identified features of LCDPs?	The aim is to identify and compare existing LCDPs capabilities and suitability for a particular development task
2	What are the objectives of LCDPs described in scientific literature?	The aim is to identify the specific goals and purposes for which LCDPs are employed in different software development projects and settings.
3	To what extent are the identified LCDP objectives and features applicable for PA?	The aim is to gain understanding in the agricultural industries and technical application of LCDPs implemented across the PA domain.
4	What are the challenges that have to be overcome for LCDPs in PA?	The aim is to identify and define the obstacles, difficulties, and deficiencies that arise during LCDP adoption in PA. Furthermore, this question aims to understand and emphasize the techniques, best practices, and suggestions provided in academic research to address the challenges identified.
5	What are the future directions for LCDPs in PA?	The aim is to explore potential developments and trends in LCDPs and provide insights on where the technology may be advancing for PA.

2.2. Search Protocol

We conducted a thorough search of the literature in the following high-quality digital libraries: IEEE Xplore, ACM Digital Library, Wiley Interscience, Science Direct, and ISI Web of Knowledge, to be able to respond to the research questions outlined in the previous section (Table 1). This choice is based on the sources employed in other SLRs, including [24,25], and is reported to achieve a good coverage of the existing literature. Research on LCDPs is rather recent; therefore, findings were limited to the last ten years only. Consequently, our analysis encompassed a review of the literature spanning the past 10 years (2013 to 2023). Journal, conference, and workshop papers were the papers' intended sources. Both an automated search and a human search were conducted to yield as many pertinent results as feasible. With the aid of search terms from the aforementioned sources, an automated search was carried out. By looking through the list of references of the publications that the automatic search turned up, a manual search was carried out. Although the search string's syntax varied depending on the source, it generally came down to its essence:

(no code OR low code AND platform)
AND (EXCLUDE(PUB YEAR before 2013))

We intentionally used the term "platform" in our study, after obtaining results from the iterative search, for various compelling reasons that correspond with our research aims. We found many articles in the literature that used 'low code' approaches where code was written to solve individual software use cases, whereas our goal was to identify and focus on the body that provided the framework for predefined low-code development. The term "platform" has been established in the industry to define entire environments or frameworks containing a variety of tools, components, and services for specific purposes. This choice was motivated by the fact that the term "platform" is widely recognized and used in the technology domain, particularly in the context of software development and related services. The research questions in this study delves into various components of the LCDP environment, as shown in Table 2. It was apparent that the term "platform" implies a more extensive and diverse environment for software development and maintenance. Since our study addresses the broader foundation and framework for the usage of low code, which includes integration with additional components, third-party services, and developer communities, the term "platform" provided the highest rate of articles that are clarifying and specifying our research scope.

Table 2. Overview of the search query results and the selection process.

Sources	After Automated and Manual Search	After Applying Selection Criteria
ACM Digital Library	42	9
IEEE Xplore	42	20
Science Direct	121	1
Scopus	141	22
Wiley Online Library	26	0
Manual search	33	8
Total	405	60
Total after removing duplicates		44

The search string results are shown in the second column of Table 2, and the total number of studies acquired by performing the search query was 405 studies. The search was performed in the month of January 2023. The database with the greatest number of studies provided was Scopus (141 studies), while the one with the fewest was Wiley Online Library (26 studies). This study used both an automated search and a manual search to locate as many potential studies as possible. The search was performed manually by looking through the reference lists of potential studies that were found by automated search (also known as the backward/forward snowballing technique). After using the snowballing method, and after eliminating duplicates, 60 articles were collected, which are represented in Table 2.

2.3. Selection Criteria

The search string was defined broadly. This was purposeful, to give the search phrase a wide reach to find any possibly intriguing study. Due to the wide scope, there were several papers, which we sorted using the criteria shown in Table 3 to choose the most pertinent ones. The number of publications was reduced to 44, along with the papers found via the manual search, by applying the selection criteria manually first by studying the study's title and abstract. Secondly, the whole article was located, read, and selected based on the selection criteria to reduce the number of publications to 37 with higher reliability and content validity.

Table 3. List of selection criteria.

	Selection Criteria
1	Papers that do not have full text available
2	Papers not written in English
3	A duplicate publication that was found in multiple sources
4	Papers that are brief (less than 4 pages)
5	Papers that do not relate to software development
6	Papers that do not relate to low code development platforms
7	Papers that do not validate the proposed study
8	Papers which are experience and survey papers

2.4. Quality Assessment

In this step, we evaluated the quality of 44 studies before proceeding with the data synthesis process. This evaluation was incorporated into the data extraction phase and involved applying the quality criteria outlined in Table 4 to the studies. These criteria were based on established quality instruments and were adapted from [23] and other SLRs. The quality assessment was intended to rank the studies based on the overall quality of their final products and used a three-point scale (yes, partial, no) for each criterion encountered during the process. Additionally, we considered excluding any studies that scored less than 4 out of 8 points. In essence, these criteria cover the fundamental components of a sufficient value of the primary studies, such as clear aims, well-defined methodology, thoroughness in addressing research questions, and effective communication of results and conclusions. By requiring a minimum of 4 points, we ensure that the articles evaluated satisfy at least half of these important requirements; therefore, enhancing the quality and reliability of the primary studies. By establishing these criteria, a study can obtain minor shortcomings for our purposes in some categories while still being considered a meaningful contribution. This balanced approach recognizes the complexities of research quality while upholding a norm that involves the dissemination of reliable findings.

However, as shown in Figure 2, 37 of the 44 primary studies we analyzed scored higher than the minimum required score on the assessment. The dotted orange line in Figure 2 shows the studies that scored lower than the score of 4 and thus were excluded. In total, 7 studies were excluded. Five of the papers we excluded were exceptionally brief, containing fewer than 6 pages. While concise papers can potentially be valuable in certain contexts, these articles were determined to be deficient in depth and sufficiency across multiple evaluation criteria, especially in terms of adequate research process documentation (criterion 4) and the presentation of negative findings (criterion 6). Two more articles were excluded due to insufficient detail in clarifying their scope and conclusions, making it difficult to comprehend the significance of their contributions to our study. Combined with the shortcomings in their overall evaluation criteria, this led to the exclusion of these studies. In total, 37 studies were included in the data extraction phase.

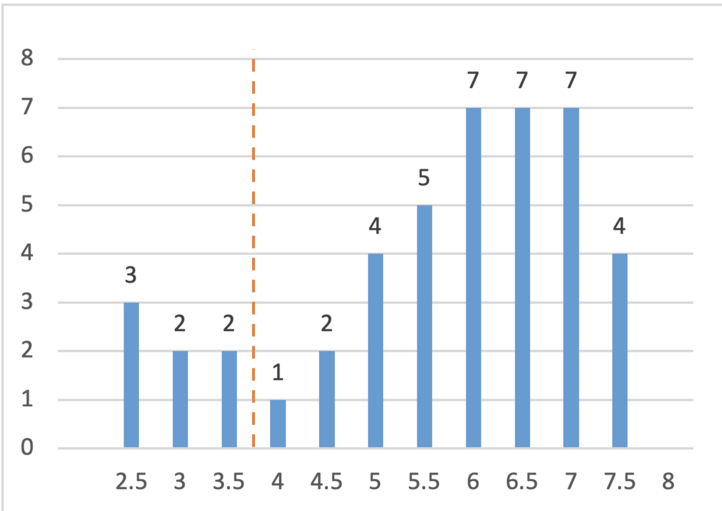

Figure 2. Quality score distribution of the selected studies.

Table 4. Quality assessment criteria.

	Question	Yes (1)	Partial (0.5)	No (0)
1	Aims clearly stated			
2	Scope and context clearly defined			
3	Variables valid and reliable			
4	Research process documented adequately			
5	All study questions answered			
6	Negative findings presented			
7	The main findings clearly stated			
8	Conclusions relate to the aim of the purpose of the study			

2.5. Data Extraction

The 37 primary studies ought to be carefully analyzed in order to provide answers to the research questions. To be able to gather and retrieve all necessary data from those studies, a data extraction form was created. Iterative design was used to create the data extraction form. In order to develop the data extraction form, we first randomly chose a number of articles and thoroughly read them. The initial data extraction form was then used to extract data from other selected articles. If it was discovered that the form did not yet include all the information, it was altered, and the updated form was applied to all the articles iteratively. The title of the study, the authors, the year of publication, and the publication venue are general components that are included in our final data extraction form. The extraction form also includes specific details that directly address the research questions, such as the type of article, the technical and industry domains that were targeted, the stakeholders, the motivation behind the study, the LCDPs that were identified with their objectives and features, the metrics used to evaluate those LCDPs, the challenges, suggested solutions, and the potential future research areas.

2.6. Data Synthesis

For the data synthesis, we used a summative content analysis. This is a data collection approach used for discovering the presence of certain words, themes, or concepts within qualitative data to explain a phenomenon, either inductively or deductively [28]. The process of identifying and arranging qualitative data to find themes is known as coding. We employed an inductive coding strategy without a predefined code frame. For classifying the various concepts, we utilized flat code frames, and for structuring the feature diagram, we used hierarchical code frames. We maintained our code frames responsively to ensure they could be adaptable to our iterative approach, and we made sure that while constructing codes, they covered balanced information. Balanced information indicated that the labels were covering not too much or too little information.

As described in the study of [29], coding can be performed for 3 different content types in the content analysis approach. Manifest, latent pattern, and projective content types require different coding. 'Manifest' content analysis is the process of searching for discrete elements and the occurrence of a specific content in textual material. In our study, we manifest the content of the LCDPs' characteristics by looking at what is mentioned in the primary studies. This was used in research question 1. For other research questions, the identified codes had to be placed in context by identifying patterns in these findings, what is called 'latent pattern' content coding. For categorizing the codes used in research questions 2 to 8, there was interference needed for recognizing a pattern in elements. Therefore, latent pattern type coding was used. If we look at the timing of defining the codes with this approach: keywords were identified before and during the data analysis to support the development of codes. These codes are derived from our experience and review of the literature. In summary, we implemented and used this 'latent pattern' content type coding on our unique set of data, including the targeted domains, objectives, identified stakeholders, metrics, features, challenges, solutions that have been presented, and future research directions proposed in studies.

Our objectives with this technique were to identify and comprehend themes, patterns, and links in the usage of LCDPs. Furthermore, we wanted to investigate how these data might be used to support theoretical claims made in research regarding LCDPs and to quantify these qualitatively gathered patterns and themes [28]. Our goal of content analysis is to manifest content by investigating in objective and discrete content characteristics of LCDPs and developing latent content coding categories with recognizing patterns in these identified characteristics.

To establish validity, the designer of the content analysis delivers a coding scheme that can guide readers in the analysis of the content [29]. Additional tables are provided in the results section with our understanding of the 'grouped' concepts and their concerns to make the findings of the data synthesis easy to understand and create the possibility for discussion. Furthermore, defining the 'grouped' concepts leads to narrowing down the degree of interpretation within these concepts. This increases the accuracy of the coded data what enhances the reliability. After the data synthesis, charts were used to illustrate and present the information that was generated.

Table A1 presents an overview of the articles that were selected for this SLR. The research query resulted in these 37 papers, in addition to these we have also considered the relation to PA and these papers are explicitly referred to in the discussion section.

3. Results

In this paper, we have reported the outcome of a thorough SLR, were we provided answers to the research questions 1 to 5. These research questions are presented in the methodology section in Table 1. The first observation of the SLR was that there is a scarcity of LCDPs used in PA, although it can provide benefits. In this following section, we focus on the key objectives and challenges related to PA. Since there are no precedents in research on the development and application of LCDPs specifically suited to the particular challenges and demands of agricultural software development, we will pave the way for future research of LCDP towards PA. We will focus on the potential of these objectives mentioned in the SLR. Furthermore, we will use the identified challenges in the primary studies to recognize the particular challenges of agricultural software development. Lastly, we will point out further research directions towards PA, to pave the way for future research to foresee these challenges.

The outcomes of the domain modeling phases of the primary studies will be discussed in this section. Every subsection will cover the findings of the primary studies focusing on the given research question particularly.

3.1. What Are the Identified Features of LCDPs?

LDCPs provide a wide range of features that can be classified into a number of different categories and each addresses important elements of the software development process. Assessing an LCDP's capabilities and suitability for a particular development task requires an understanding of these classifications. The classifications of these features are explored in depth in this section, giving information about the broad range of capabilities offered by LCDPs. In this section, we will further develop and deepen this framework using a bottom-up approach, evaluating the primary studies. We will build further to the work of [30], considering the features of the eight recognized LCDPs

Therefore, we will highlight and describe the features below. Among all the features we have analyzed in this paper, we identify a new top-level feature: AI and machine learning. Visible in the feature diagram (Figure 3) is how these features related to each other in the sublevel features.

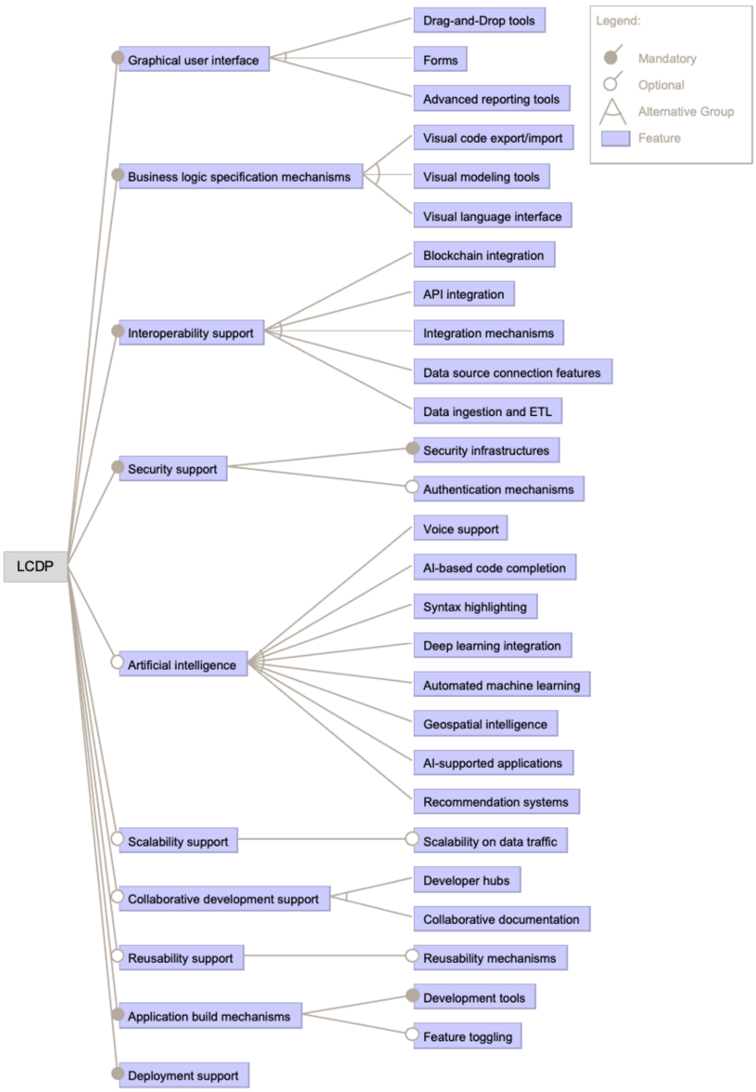

Figure 3. Feature diagram of LCDP and the identified sub-features in the primary studies.

3.1.1. Graphical User Interface (GUI) Features

The GUI capabilities of low-code development platforms are essential in making software development an approachable, productive, and collaborative process [31]. With the help of these features, developers can lessen their reliance on manual coding by designing user-friendly interfaces, data input forms, producing proficient reports, and managing data effectively. A deeper understanding of the potential of GUI features is crucial for businesses looking to streamline their software development processes and provide their users with high-quality applications as the adoption of LCDPs keeps expanding [32].

Drag-and-Drop Tools

The availability of drag-and-drop tools is one of the GUI category's notable characteristics [30]. Drag-and-drop tools belong to GUI, since users interact with these graphical components such as icons, buttons, folders, and menus to manipulate commands and run software code. By simply dragging and dropping elements onto a canvas, these tools enable developers to create application user interfaces [33]. Rapid prototyping and iterative design processes are made possible by this feature, which significantly reduces the need for manual coding [34]. In addition to speeding up development, it also promotes UI design experimentation and creativity [35].

Forms.

In a wide range of applications, forms are a fundamental part of user interaction. Developers can create and customize forms in LCDPs without diving into complex code due to the form-building features integrated into the GUI. Because of this, developers are better equipped to create data input interfaces, from straightforward data collection forms to intricate user registration and data entry forms [36].

Advanced Reporting Tools

In applications where data analysis and visualization are necessary, robust reporting capabilities are vital. Developers can create reports that are both visually compelling and insightful due to the advanced reporting tools identified within LCDPs' GUI. Making the data easy to understand, this not only improves the user experience, but also gives decision-makers valuable insights from the application's data [31].

3.1.2. Business Logic Specification Mechanisms

Visual Code Export/Import

Even in low-code environments, developers can benefit from being able to see and manipulate code within the GUI [37]. The platform's underlying code can be viewed and modified by developers due to features for exporting and importing visual code. This makes it easier for advanced users to customize the code while still maintaining the benefits of low-code for those who prefer a visual approach.

Visual Modeling Tools

Tools for visual modeling provide a high-level illustration of the logic and structure of an application. With the aid of these tools, developers can build visual representations of intricate workflows and connections within an application [34,38]. In addition to streamlining the design phase: this high-level illustration of the logic with the visual modeling approach can also serve as documentation for the application's architecture [37].

Visual Language Interface

A prominent trend in LCDPs is the use of a visual language GUI. It enables programmers to convey application logic through the use of visual components, sometimes in the form of flowcharts or diagrams [31]. This method makes the behavior of the application more understandable and accessible to stakeholders that are not technically inclined, such as business analysts or project managers [39,40]. A visual drag-and-drop interface advances the drag-and-drop idea by enabling developers to visually integrate and configure intricate functionality. Intricate coding activities like process automation or interaction with external services are abstracted by this feature into a condensed, graphical representation [41].

3.1.3. Interoperability Support

Support for interoperability and blockchain integration capabilities give developers the tools they need to make flexible and innovative applications. These applications can interact with a variety of external services and conveniently incorporate data from various sources due to these features. Access to external resources is made easier through API integration, and effective data management is ensured by connectivity to data sources. Integration mechanisms standardize communication, lowering the complexity of development, and data ingestion/ETL procedures guarantee the integrity of the data. The possibilities of LCDPs are expanded into the world of decentralized technologies via blockchain integration features. This creates opportunities for decentralized apps, immutable records, and trustful transactions. Developers can build apps in industries like finance, healthcare, and supply chain where transparency and trust are crucial by utilizing the decentralized characteristics of blockchain.

Blockchain Integration

One of the distinctive characteristics of blockchain is its decentralized nature. Using LCDPs, developers can use this feature by making applications that communicate with decentralized networks. Study [42] mentions that this creates possibilities for applications in various fields. For the purpose of establishing smart contracts and applications on blockchain networks, LCDPs provide tools and resources. According to [42], "smart contracts" are self-executing contracts with the terms of the agreement directly written into code. With LCDPs, developers can design and alter smart contracts to match the particular needs of their applications. The flexibility and adaptability of applications in blockchain ecosystems are improved by this capability. To ensure trust and immutability, developers can use these characteristics to install applications on blockchain systems securely and transparently.

API Integration

Through application programming interfaces, LCDPs provide the ability to interact with a wide range of external services [43]. With the use of this capability, developers can link their applications to a variety of services, such as cloud storage providers and third-party application software [43,44]. As a result, applications become more adaptable and able to use resources from external resources [31,45–47].

Integration Mechanisms

The techniques and protocols that allow seamless communication between various software services and components are referred to as integration mechanisms. In order to simplify customized integrations, LCDPs provide predefined integration techniques [48]. As a result, applications' effectiveness and connection are improved by enabling relatively easy interaction with external systems [49,50].

Data Source Connection Features

Effective data management often involves aggregating data from disparate sources. LCDPs facilitate developers to connect to different data sources, giving them access to manipulation and management of data [30]. These systems offer the essential connectors and protocols for effective data retrieval, regardless of whether the data are kept in on-premise databases or the cloud [48,51].

Data Ingestion and ETL

For data preprocessing and integration, Extract, Transform, Load (ETL) procedures are essential [52]. Developers can manipulate and transform data before incorporating it into applications by using the tools for data ingestion and ETL provided by LCDPs [53]. This function streamlines the management of data and ensures that applications have access to accurate and reliable data [54].

3.1.4. Security Support

The extensive security capabilities built into LCDPs demonstrate their dedication to offering secure environments for development and deployment.

Security Infrastructures and Authentication Mechanisms

To protect user identities and data when they interact with the application, authentication mechanisms, and security protocols act as strong protective layers [34,55,56]. User access control infrastructures give developers the ability to specify certain user privileges, improving security and upholding the principle of least privilege. Additionally, security tools like intrusion detection, vulnerability assessments, and encryption actively spot and reduce security risks, enhancing the application's resistance to threats [43,49]. Data integrity and regulatory compliance are strengthened by data table security and access control mechanisms, which make certain sensitive data remains private and unaltered [57].

3.1.5. Emerging AI and ML Features

Due to their capacity to automate processes, improve user experiences, and enhance decision-making, artificial intelligence (AI) and machine learning (ML) features in low-code development platforms are becoming increasingly significant. These features make use of AI and ML algorithms to accelerate up application development and give both expert and citizen developers the tools they need to produce intelligent, data-driven applications.

Voice Support

Voice support is a feature that allows developers to interact with the LCDPs using natural language and voice commands [58]. This feature can involve query interpretation, voice-activated code generation, and development environment navigation [34]. For developers who may favor spoken commands to traditional typing, voice support improves accessibility and expedites development tasks [53].

AI-Based Code Completion

Machine learning techniques are used in AI-based code completion to predict and offer code snippets, function names, and variable names as programmers type [58]. This feature has the potential to substantially boost developer productivity. It promotes consistency and accuracy in the code.

Syntax Highlighting

As developers type their code, AI-powered syntax highlighting may automatically find and indicate any syntax faults or problems. For developers who are less skilled in coding, it offers real-time feedback that makes it relatively simpler for them to see and fix errors [58].

Deep Learning Integration

Without having any prior knowledge of neural network architecture, deep learning integration enables developers to integrate deep learning models, such as neural networks, into their applications. Building AI-powered programs that use image recognition, natural language processing, and recommendation systems is rendered more straightforward using it [48,59].

Automated Machine Learning (AutoML)

AutoML is a feature that automates the process of building, training, and deploying machine learning models. The difficulties in creating ML models are abstracted. AutoML can be used to build ML models that fit to particular business demands by non-data scientists and developers with modest machine learning (ML) skills. Model deployment, model selection, and hyperparameter tuning are among the features that AutoML typically has [59].

Geospatial Intelligence

AI and ML are leverage geospatial intelligence features to process and evaluate geographic data. This makes it possible to create location-aware software like mapping, recommendations based on a user's current location, and route optimization [48].

AI-Supported Applications

Developers can integrate pre-built AI components or modules into applications using LCDPs that support AI. These elements might include, among others, chatbots, sentiment analysis, image and speech recognition. Without considerable AI expertise, developers can use AI capabilities [48,59].

Recommendation Systems

Recommendation systems in low-code platforms use AI algorithms to provide developers with intelligent suggestions and recommendations during the development process. These recommendations can range from suggesting the most appropriate data sources to offering code snippets or design patterns based on the developer's context [49,53,58].

3.1.6. Scalability Support

Scalability on Data Traffic

A fundamental part of software development is data management. LCDPs facilitate data management through features like event management, data preprocessing, and data monitoring [37]. These tools allow users to ensure the accuracy and security of data while allowing them to extract useful information from a variety of data sources. Data access is streamlined by integration with external systems and APIs, which makes it a crucial component of the LCDP ecosystem [44].

3.1.7. Collaborative Development Support

The collaborative development models used by LCDPs reflect the changing dynamics in modern software development teams. Different scenarios can be accommodated by online and offline collaboration options, encouraging efficient teamwork and project completion. Versioning improves the overall development process by managing code changes and streamlining conflict resolution, ensuring project integrity [37].

Developer Hubs

Developer hubs serve as platforms for knowledge exchange, encourage an environment of community, and offer developers useful resources. They support developers in mastering the LCDP environment and encourage continuous learning. Mechanisms that support innovation encourage developers to think creatively and try out novel concepts. Idea generation is stimulated by brainstorming platforms and innovation challenges, which in turn generates the creation of extensive applications.

Collaborative Documentation

Developers can work more effectively in teams and share knowledge when reusability features like collaboration and documentability are included [60]. Multiple developers can work harmoniously on projects through collaboration features, while documentability serves as a valuable resource for understanding and reusing components effectively.

3.1.8. Reusability Support

Reusability Mechanisms

The effectiveness and consistency of application development are substantially affected by reusability mechanisms in LCDPs. By offering reusable building blocks for different application features, pre-defined templates, pre-built dashboards, built-in forms, pre-built components, and workflows facilitate development [32,49,61,62]. This not only streamlines development, but also guarantees consistency and quality in the design and functionality of the applications [63].

3.1.9. Application Build Mechanisms

Development Tools and Feature Toggling

The functionalities that are included in code-related features are numerous. The quality and performance of the generated code are ensured via code transpiling, correctness checks, optimization, verification, and compilation [31,37,64,65]. Manual coding efforts are reduced via code generation and automated code generation. The robustness of the application can be verified by testing, which includes unit, integration, system, acceptance, regression, performance, and security testing. Version control and feature management are improved by feature toggling, killing, and rollback tools [66].

3.2. What Are the Objectives of LCDPs Described in the Scientific Literature?

After conducting data synthesis, 15 primary LCDP objectives in total have been identified (Table 5). Providing abstraction mechanisms was the most mentioned objective in the studies. In defining the objectives of the LCDPs, we investigated what the articles referred to as their objectives and research purpose rather than filling in additional objectives ourselves. We emphasized looking into what has been set up as the main objectives for the researched LCDPs in the primary studies. Simplifying software development, raising productivity, and speeding up deployment are other frequently mentioned objectives. Figure 4 gives an overview of the identified objectives, Table 5 provides a comprehensive summary of the definition of these classified objectives (domain scoping), and Table 6 gives an overview of the primary studies that mentioned these objectives.

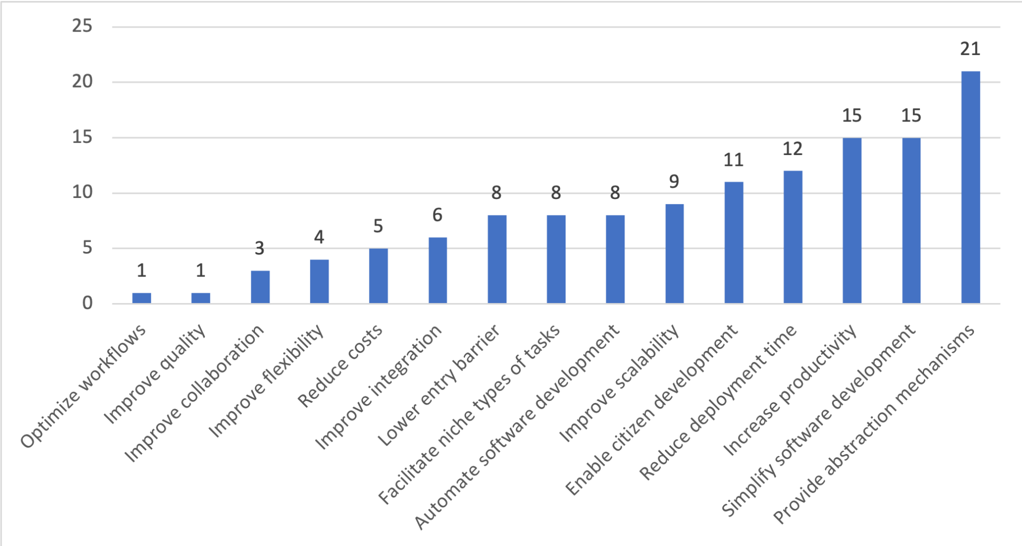

Figure 4. The identified objectives of the LCDP described in the literature.

A variety of LCDP objectives aim toward enhancing software development procedures and making them cost-effective and user-friendly [67]. Although the precise focus of these objectives differs, there are similarities in how they relate. Many of these goals are connected in some way or another and reinforce each other. For instance, enhancing collaboration and supporting digital transformation can both help streamline processes and boost output [68], like how offering abstraction mechanisms and enhancing flexibility can facilitate the simplification and quality improvement of software development [48,62].

Furthermore, many of these objectives can be accomplished using the same tools and techniques. For instance, by utilizing graphical user interfaces (GUIs) and providing abstraction mechanisms, citizen development can be facilitated, which attempts to enable non-technical persons to construct and change apps [44]. At the same time, automation tools can be used to reduce deployment time and improve scalability [66].

Lastly, many of these objectives are driven by a desire to enhance user experience and increase accessibility of software development. Making software development more accessible and inclusive can be accomplished through lowering entry barriers, supporting specific types of tasks, and enhancing integration [36,41].

Overall, LCDPs encompass various objectives, with 15 primary objectives identified. The most frequently mentioned objective is providing abstraction mechanisms. Many of these objectives are interconnected and mutually reinforce each other. Most objectives complement user experience. The various objectives of LCDPs share a common goal of making software development more efficient, effective, and accessible. While there are differences in their specific focus, they are all aimed at improving the software development process, user experience, and delivering better solutions.

Table 5. The identified objectives and their concerns regarding LCDPs in the literature.

No	Objectives	Concern
O1	Provide abstraction mechanisms	With the use of LCDPs, developers can concentrate on higher-level functionality rather than detailed implementation details. As an outcome, this simplifies the development process and reduces the time required to build new applications, enabling users to focus on business value rather than technical implementation.
O2	Simplify software development	By offering pre-built components and integrations, visual modeling tools, and simplified coding interfaces, LCDPs seek to streamline the software development process. This enables developers to create applications with less complexity, improving the overall development process.
O3	Increase productivity	By providing a generative development approach that requires less coding and a shorter period to create new applications with higher value, LCDPs seek to boost productivity. As a result, developers may concentrate on tasks that generate higher economic returns like innovation, design, and user experience.
O4	Reduce deployment time	By offering pre-built components and integration, LCDPs can shorten the time it takes for developers to build and deploy applications. This reduces the time needed to develop and test new features, enabling companies to release innovative products to market more quickly.
O5	Improve scalability	By supplying an environment where developers may quickly scale up or down applications in response to shifting business needs, LCDPs seek to increase scalability. Because of this, businesses can handle rising traffic and data processing demands without having to completely redesign their applications.
O6	Enable citizen development	By giving non-technical users the tools and resources to build applications, LCDPs intend to promote citizen development. For instance, an LCDP can offer training, intuitive instructions, guides and other resources to help non-technical users become familiar with the platform and learn how to build applications.
O7	Automate software development	By offering automation mechanisms, automated testing, and deployment tools, LCDPs aim to automate the production of software. This objective aims to automate various software development processes, including code generation, testing, and deployment.
O8	Facilitate niche types of tasks	By offering pre-built components, combinations of features, tools, processes, and templates for common business processes, LCDPs can support specific tasks and projects in application development. This objective aims to support particular software development tasks, e.g., creating e-commerce applications.
O9	Lower entry barrier	By enabling non-technical techniques to construct apps and minimizing the requirement for coding skills, LCDPs seek to lower the entry barrier. Users can more easily construct their own applications by using an LCDP, which can offer a visual interface for designing and configuring apps.
O10	Improve integration	By offering pre-built connectors and APIs for common business applications and services, LCDPs seek to improve integration. With the use of various integration features provided by LCDPs, including API integration, third-party integration, and data connectors, this objective intends to enhance the integration of diverse software applications and systems to enable seamless data interchange and communication. A low code platform might offer connectors for widely recognized CRM or ERP systems, e.g., allowing users to quickly incorporate these systems into their applications.
O11	Reduce costs	By enabling organizations to create applications with greater speed and efficiency, LCDPs can help businesses cut costs by eliminating the need for costly customized programming or third-party developers. By giving organizations access to the various cost-saving tools provided by LCDPs, such as shorter development times, streamlined development procedures, and lower development costs, this objective aims to lower the cost of software development.
O12	Improve flexibility	Rapid prototyping and iteration are made possible by LCDPs to increase flexibility. A LCDP can make it straightforward for users to update and modify their applications, add new features or integration, and adjust to shifting business needs. By making it possible for developers to make changes quickly and easily to their applications, this objective seeks to increase the flexibility of software development.
O13	Improve collaboration	With the help of the various collaboration features provided by LCDPs, such as real-time collaboration, version control, and team collaboration tools, users can collaborate on the design and development of apps, share feedback, and monitor progress in real-time. LCDPs can help improve collaboration by offering a common platform for developers, business users, and other stakeholders to develop applications more interactively.
O14	Improve quality	By offering integrated testing and debugging tools, LCDPs can aid in enhancing the quality of software applications. With numerous mechanisms to enhance software quality, such as code reviews, testing, and quality assurance processes, this objective aims to increase the quality of software applications by lowering mistakes, defects, and other quality concerns.
O15	Optimize workflows	By streamlining and automating repetitive tasks, reducing manual effort requirements, increasing workforce efficiency, LCDPs aim to optimize workflows. By eliminating manual work and automating routine tasks, like data entry or approvals, and providing real-time visibility into the status of the process, this objective aims to streamline and optimize business processes.

Table 6. The LCDP objectives mentioned in the primary studies.

Study	Objectives Categories														
	O1	**O2**	**O3**	**O4**	**O5**	**O6**	**O7**	**O8**	**O9**	**O10**	**O11**	**O12**	**O13**	**O14**	**O15**
[54]					x		x	x							
[42]	x			x		x					x	x			
[45]	x	x					x			x			x		
[37]	x	x			x		x								
[58]	x	x								x					
[34]	x							x	x						
[53]	x	x											x		
[50]		x	x			x			x						
[59]	x		x			x									
[49]		x				x			x						
[48]	x		x		x				x			x			
[33]	x	x	x			x			x						
[69]	x	x	x			x									
[60]	x	x	x	x											
[66]	x		x				x								
[63]	x	x													
[61]			x	x	x										
[70]							x	x							
[41]	x				x					x					
[36]		x		x		x			x						
[46]			x	x											
[32]			x	x							x				
[47]	x	x								x					
[51]					x		x	x							
[57]		x		x				x							
[71]		x		x		x			x		x				
[43]	x				x					x		x			x
[62]	x	x						x	x					x	
[65]			x	x							x				
[39]			x			x			x						
[38]	x				x					x			x		
[40]	x			x		x					x				
[35]						x									
[55]			x	x			x	x							
[31]	x		x					x							
[44]	x			x	x							x			
[56]			x				x								

O1: Provide abstraction mechanisms; O2: Simplify software development; O3: Increase productivity; O4: Reduce deployment time; O5: Improve scalability; O6: Enable citizen development; O7: Automate software development; O8: Support specific tasks; O9: Lower entry barriers; O10: Improve integration; O11: Reduce costs; O12: Improve flexibility; O13: Improve collaboration; O14: Improve quality; O15: Optimize workflows.

3.3. To What Extent Are the Identified LCDP Objectives Valid in the Context of PA?

We identified 15 objectives, as we described in Table 5. Since these objectives are general, they can be valid in each domain. For PA, we will elaborate in this subsection on the applicability and relevance of the identified objectives.

The increase in LCDP research reflects the interest in user-friendly and agile software development. With the ability to build customized solutions that increase data-driven decision-making, automation, and overall efficiency in agriculture, this trend strongly correlates with PA. To optimize farming procedures, PA uses automation and data-driven decision-making [72]. PA software can be developed with the help of LCDPs to help farmers make educated decisions about crop management, resource allocation, and yield optimization, these platforms enable the rapid development of applications that gather, analyze, and visualize agricultural data [73]. Researchers and practitioners in this field should target the objectives of LCDPs.

Simplifying Software Development

The complexity and diversity of the agricultural domain must be taken into consideration. Supply chain management, farm management systems, GIS, smart farming applications, etc. are just a few examples of the diverse applications covered by agricultural software development [74]. Each subdomain has its own set of operational challenges, data structures, and requirements. Therefore, the absence of LCDPs specifically designed for agricultural software development may be a consequence of the need for more specialized and customized methodologies to meet the demands specific to the agricultural industry [75,76]. Applications that process large amounts of data from sensors, drones, and satellites are crucial to PA [77]. LCDPs simplify the development of these applications, making them accessible to farmers and researchers with limited coding knowledge. The development of data-driven tools for tasks like soil analysis, crop monitoring, and predictive analytics accelerates by decreasing complexity [21]. As described in [78], they built a scalable cloud architecture to control the growth of crops and irrigation management. As a result, farmers have a greater capability to make informed decisions, use resources efficiently, and ultimately increase crop yields [10]. Furthermore, due to the variation in field sizes, environmental conditions, and data volumes, the scalability of applications in PA is essential. As these variables change, LCDPs provide the adaptability required to scale applications up or down. As farms adopt more sensors and automated equipment, this improved scalability ensures that PA tools can effectively handle the rising demands of data processing. Ref. [79] indicates that the use and storage of data can be complex using Farm Management Information Systems (FMIS). Precision farming depends on data-driven insights, which must be accurate and timely [80]. These examples substantiate that simplifying software development is essential for PA.

Enabling Citizen Development

Additionally, the objective for enabling citizen development is also applicable to PA for education and training [81]. It makes use of e-learning platforms, knowledge-sharing portals, and decision support systems to communicate information, best practices, and specialist guidance to farmers, researchers, and extension workers [82]. Additionally, the identified environmental domain in LCDPs can contribute to agricultural software solutions with enabling citizen development. GIS applications for agriculture incorporate spatial data, maps, and satellite imagery to improve decision-making. Data on resource allocation, crop zoning, soil mapping, and land use can be analyzed and visualized utilizing GIS software [83].

When it relates to technical areas, the emphasis on “business process management” points out the value it can have on effective workflows in PA. Agribusiness processes like inventory management, sales, and customer relationship management can all be made more efficient through the use of LCDPs [84]. This will improve the productivity of agricultural practices as a whole with enabling citizen development.

The LCDP literature also shows a substantial amount of interest in “machine learning” and “software maintenance”. Enabling machine learning for citizen developers can be used to analyze sizable datasets from sensors and satellites in the context of PA, giving insights into crop performance and disease detection [85]. Particularly in remote and challenging farm environments, making software maintenance more accessible is crucial to ensuring the continuous functionality of agricultural applications.

Providing Abstraction Mechanisms

In LCDPs, abstraction mechanisms allow users to work with higher-level functionalities and components, which streamlines the development process. Through this simplification, people with different technical backgrounds, such as farmers and domain experts, can take an active role in software development. This collaborative approach fosters innovation and drives the adoption of best practices. Furthermore, studies show that it is challenging to develop custom software solutions for PA that are cost-effective and economically beneficial at the end [86,87]. Through the use of pre-built components, templates, and automation tools, LCDPs offer a solution by potentially lowering development costs. As a result, a wider range of farmers, including those operating on a small scale or with limited resources, may have easier access to advanced PA technologies [88]. The affordability of these solutions encourages wider adoption of data-driven farming practices, benefiting the agricultural industry.

Integration for Informed Decision-Making with AI and ML for Enhanced Decision Support

Integration of various data sources, including sensors, IoT devices, weather APIs, and satellite imagery, is essential for PA [89]. LCDPs can help the field by providing features for connecting to data sources, integrating APIs, and supporting interoperability, which is mentioned as a challenge by [77]. Real-time data from various sources can be seamlessly integrated into a centralized platform as a result. By gaining access to a holistic view of their farming operations, farmers can make data-driven decisions about crop rotation, fertilization, pest control, and irrigation [2,89]. The emerging AI and ML features in LCDPs hold immense potential for PA. Machine learning algorithms can examine historical data to provide insights such as optimal times for the planting process, early disease identification through picture recognition, and automation of farm machinery [90]. Crop rotation techniques can be recommended by AI-driven recommendation systems based on the soil conditions and past yield performance [91]. According to [90], these tools enable farmers to make data-driven decisions that maximize agricultural productivity and resource efficiency. The application of ML in PA has the potential to be supported and improved by LCDPs.

Improved Collaboration for PA

It is discussed in the papers by [17,92] that collaboration platforms benefit PA software development processes. PA must be advanced through collaboration between farmers, agronomists, and researchers [93]. Collaboration tools in LCDPs make it feasible for various stakeholders to work together. Version control and real-time editing are just two of the collaboration tools that LCDPs provide to encourage teamwork and knowledge sharing. Improved collaboration makes it possible to incorporate knowledge from various fields into applications for PA. To improve models, share data, and collectively address complex agricultural challenges, farmers can work with experts and researchers [93]. Different LCDP characteristics can enhance communication between the identified stakeholders. Agronomists and farmers can collaborate on data-collecting forms, analytics models, and decision-support systems with other recognized stakeholders [94]. Version control mechanisms also ensure that changes are tracked, and that decision-making is based on the most up-to-date data available [66]. Reusability mechanisms enable productive workflows to be standardized, enhancing the effectiveness and consistency of data collection and analysis [60]. Therefore, LCDP features like development tools for improved functionality present opportunities that can help PA [93]. PA applications can benefit from enhanced functionality by leveraging code-related aspects in LCDPs, such as automated code creation and feature toggling [77]. While feature toggling enables controlled deployment of new features and upgrades, automated code generation can produce scripts for data analysis.

To summarize, the incorporation of LCDPs may be vital in advancing farming techniques, improving crop yields, and contributing to sustainable agriculture as PA keeps developing. Focusing on the objectives, the goals of LCDPs are in line with the demands and difficulties of creating agricultural software solutions. LCDPs provide useful tools and strategies, from abstraction methods and streamlining development to increasing output, reducing deployment times, and supporting specific tasks. Furthermore, LCDPs benefit the responsiveness to the business, the reduction of IT backlog, maintenance reduction of the existing platform, and human resource availability [68]. By utilizing these objectives, agricultural software developers can improve productivity, accelerate development procedures, and produce software programs that optimize farming techniques, increase resource efficiency, and promote sustainable agricultural operations.

3.4. What Are the Challenges That Have to Be Overcome for LCDPs in PA?

We have discussed the objectives and the relations towards PA. Based on the literature, we also identified several challenges. Similar to the objectives, challenges can be different for different domains. In this subsection, we discuss on the challenges of LCDP that are applicable to PA.

LCDPs' limitations have an overall impact that extends far beyond their specific scope. These constraints interact in a complex way, affecting the course of development and having a cumulative effect. This discussion highlights the importance of adopting a holistic perspective when approaching these platforms, emphasizing the need to address limitations not as isolated entities but rather as essential components within a larger development ecosystem. By understanding the implications and dynamics of these particular limitations, academics and organizations can navigate the low code landscape with higher productivity and strategic foresight. When creating agricultural software, it is crucial to consider the unique requirements and challenges that the agriculture industry encounters.

User Interface and Experience and Delayed Learning Curve

'Literacy Rate' is mentioned in [77], meaning that literacy is a significant factor influencing the adoption ratio in PA. Farmers in regions with high illiteracy rates grow crops more likely depending on their experience. They do not use innovative agricultural technology, resulting in a loss of productivity. Farmers needs to be educated to grasp the technology, or rely on a third party for technical assistance. As a result of resource and education constraints, PA is not widely practiced in undeveloped communities with low literacy levels. As we identified objectives in our SLR to simplify software development, there are still challenges that need to be overcome. The LCDP needs to be user-friendly for developers to take full advantage of its features. In the agricultural context, where there may be a wide range of users with varying levels of technical skill, having an intuitive and user-friendly interface becomes necessary. Given that English may not be the primary language in all agricultural locations, and that agricultural terminologies can differ in cultural contexts, language barriers can also negatively impact user experience and production [38,95,96]. Developers of agricultural software will benefit from LCDPs that provide multilingual support because it would increase technology transfer and allow for the design of user interfaces that take agricultural strategies, requirements, and settings into account [2].

Limited Functionality or Extensibility

Furthermore, sophisticated "functionality and extensibility" may be needed in agricultural applications to meet specific requirements. Even though LCDPs offer pre-built components, certain platforms' restricted functionality and extensibility may make it difficult for developers to interface with particular hardware or implement advanced agricultural algorithms and methods, both mentioned in the studies of [2,77]. LCDPs with a wide range of features, permitting customization and extensibility, and being able to be tailored to the various needs of the agricultural industry will benefit the development of agricultural software.

Deep Understanding of Targeted Platform

As it is mentioned by [14], agricultural software developers may require a thorough knowledge of the underlying technologies and data models employed by the platform to effectively leverage an LCDPs' capabilities. For developers who may not have had prior platform-specific knowledge, this can be a barrier to entry. LCDPs that offer detailed documentation, tutorials, and support tailored to the particular needs of the agricultural domain may be advantageous for the development of agricultural software, since they enable developers to quickly gain essential knowledge.

Data Security, Scalability, and Confidentiality

Proprietary information, such as crop varieties, production estimates, and business strategies, are often used by farmers and agricultural enterprises [97]. Various security issues causing a variety of difficulties are addressed in [12]. A challenge mentioned by [77] is that data corrupted by an intruder reduces the effectiveness of PA. By implementing strong security frameworks and procedures like access control, authentication, and encryption, LCDPs can overcome this issue. These measures protect agricultural data from unauthorized access, ensuring that private data are kept secure. Scalability of data management becomes crucial as PA operations grow, and is seen as a challenge in PA [77,98]. Scalability capabilities provided by LCDPs allow the platform to manage significant amounts of data. These platforms can provide risk mitigations and trust for the rising demand for data-driven decision-making, whether it is a small family farm or a large agricultural organization.

Integration and Fragmentation

Finally, issues with integration and fragmentation may be a hurdle for agricultural LCDPs. According to [2], agricultural software development encompasses a wide range of subdomains. Heterogeneous LCDPs that are fragmented, lack integration options, and adhere to defined procedures could prevent the development of integrated agricultural software solutions. To overcome this limitation, LCDPs for agricultural software development should focus on providing developers with the tools they need for seamless integration, enabling them to build complex, integrated systems that cover many different agricultural industries. As mentioned by [14,77], a crucial challenge is the integration of agricultural software with different 'digital systems', such as meteorological services, IoT devices, or third-party APIs. LCDPs might provide compatibility and interoperability problems when integrated with various technology stacks or outdated systems. This challenge is enhanced due to the nature of agriculture where great weather variations can occur, leading to variations in the needed hardware. LCDPs with strong integration capabilities, support for industry-standard protocols, and interoperability with widely used agricultural technology and systems would be advantageous for the development of agricultural software.

Nevertheless, some domains have significant overlap in activities and are well suited for agricultural software development. IoT, which is primarily employed in PA, is the next-largest domain [6]. Applications and networks for agricultural sensors include the use of sensors and IoT devices in farming systems. Software solutions in this area enable the data collection, monitoring, and control of several parameters, such as soil moisture, temperature, humidity, and livestock health [20,99]. As mentioned by [100], integrating IoT via blockchain is still in the early phase, and has many complex technical challenges. The identified database management domain follows, which has the potential to bring significant relevance with appropriate adaptations for agriculture. This area can be investigated in software applications for processing substantial amounts of agricultural data [101]. To extract useful insights from many data sources and enable data-driven decision-making, it uses techniques like data mining, machine learning, and predictive modeling [102,103].

3.5. What Are the Future Research Directions for LCDP in PA?

The previous sections have shown that there are obvious objectives that can benefit PA. Some challenges must be overcome to pave the way for future research to meet these objectives.

Metrics

In the SLR, various metrics are identified to assess the LCDPs for specific needs. To build further on the applicability and comparison of LCDPs to PA, these metrics can be used to quantify collaboration, user-friendliness, development speed, scalability, security, and integration capabilities. It additionally give focus on cost-effectiveness, flexibility, performance, and support for various deployment models. Their emphasis on the value of user-friendly, adaptable, and data-secure software solutions that facilitate rapid decision-making and collaboration among farming stakeholders is in line with the requirements of PA [16,80]. These metrics offer a useful framework for selecting and developing technical tools that improve agricultural sustainability and efficiency while meeting the various requirements of PA technologies [2]. In the SLR, various metrics are identified to assess the LCDPs for specific needs. To built further on the applicability and comparison of LCDPs, these metrics can be used in future research.

Shaping an Inclusive Ecosystem for PA

The ecosystem diversifies as more stakeholders with domain-specific expertise engage in LCDPs. Due to this diversity, a wider range of tools and applications are available to meet the unique requirements of various farm types and agricultural practices. To ensure timely and effective resource delivery, reduce waste, and increase operational efficiency in the supply chain, logistics and manufacturing stakeholders can use LCDPs to optimize the distribution of agricultural inputs [104]. Using LCDPs, HR professionals can create specialized applications for managing farm labor, scheduling employees more efficiently, tracking employee performance, and enhancing productivity. PA innovators use LCDPs for quick prototyping and testing of novel solutions, like sensor-based applications, which makes it easier to integrate cutting-edge technologies into the industry [2]. To facilitate retail sales of agricultural goods and agricultural technology solutions, e-commerce specialists can build online platforms employing LCDPs, increasing market access and profitability [105]. For budget management, income forecasting, and return on investment (ROI) analysis, financial specialists create tailored financial apps using LCDPs [106,107]. These applications help ensure the sustainability of PA applications by enabling informed financial decision-making and creating an ecosystem that enhances PA.

Altogether, PA holds extensive promise for the research directions highlighted in LCDPs. To enhance farming techniques, PA significantly relies on technology and data-driven insights [2]. The creation of complex applications that enable real-time monitoring, data analysis, and informed agricultural decision-making can be facilitated by integrating LCDPs with IoT devices and data analytics [80,101]. Additionally, it is essential to emphasize increasing security measures through simulation and encryption to protect sensitive agricultural data [12]. By improving the usability of development platforms, even non-technical users in agriculture can efficiently interact with and benefit from technology. As it enables the development of specialized tools and libraries catering to the sector's specific requirements, the idea of domain-specific development platforms is particularly relevant in PA. Together, these developments have the potential to completely transform farming methods by fostering more effective, data-driven, and long-term PA solutions that tackle the problems faced by modern agriculture while boosting crop yield, resource management, and overall farm productivity.

4. Discussion

In this section, we provide an overall discussion of this study and the threats to validity.

4.1. Unveiling the Research Questions and Bridging This Towards PA

Systematic Exploration Through SLR

In the initial segment of this research, we answered the research questions focusing on the existing general literature available regarding LCDPs. With the adoption of the established SLR methodology from [108], we gathered extensive knowledge ensuring that our approach is built upon established frameworks.

The inclusion of several data sources from high-quality digital libraries including IEEE Xplore, ACM Digital Library, Wiley Interscience, Science Direct, and ISI Web of Knowledge widens our literature review. The use of thorough selection and quality assessment criteria indicates that the studies we choose are in line with the study objectives, enhancing the reliability and credibility of our findings.

The development and use of an adaptable data extraction form, along with a summative content analysis approach, ensured the extracting nuanced information and uncovering patterns, themes, and interactions within the academic research on LCDPs. The review we gave to validity and reliability in our coding technique and scheme improves the interpretation of our synthesized data.

Bridging Insights to PA Through Domain Analysis

After thoroughly investigating the objectives of LCDPs, the second phase of our research is analyzing the findings for their relevance to PA. This domain analysis enables us to connect the SLR findings to the context of precision agriculture. In this part, we broaden the scope of our study by examining how the insights from the LCDP literature could be strategically implemented in precision agriculture to foresee future research with perspective. We hope to contribute to the larger knowledge of LCDPs by exploring their practical application in the PA domain by investigating potential links in objectives, challenges, and future directions. This two-part structure provides a more accurate and nuanced approach, beginning with a systematic study of LCDPs and addressing an extensive domain analysis that connects our results with the specific difficulties and objectives of precision agriculture. As our goal is to focus on the agricultural domain, the interaction of these two components increases and expands the scope of our study, leading to insights for both academic and practical stakeholders in the PA domain.

4.2. Threats to Validity

Threats to validity are an essential component in making sure an SLR is thoroughly determined. The concept, internal variables, external variables, and conclusion all have the potential to undermine validity [109]. When conducting this study, we adhered to accepted practices for systematic research, including the creation of a comprehensive research protocol that has been acknowledged by external, independent researchers. Nevertheless, we recognize that there may have been validity threats that affected our research. We discuss potential validity threats and associated mitigation strategies in the sections that follow, according to [110].

External Validity

The purpose of our research is to pave the way for future research focusing on LCDP applications in the PA domain. As we were limited by the fact that the literature was scarce on the usage of LCDPs in the PA domain, we applied the mitigation according to [110]. In the first part of this paper, we used a systematic approach of doing an SLR on all the papers that were available regarding the LCDP research. We mitigated the validity threat of generalizability according to [110] by using a broad time and publication coverage by targeting all available LCDP applications. In the second part, the mitigation was enhanced by comparing the findings of the systematic research of LCDP to domain studies, by performing a domain analysis combined with agricultural software applications in the literature.

Internal and Conclusion Validity

We mitigated on the conclusion validity and internal validity of this SLR study by having an interchange among the authors while setting up the research questions for this study and establishing the relationship between the research questions and research goals. So, we assume that the probability of any connection between our findings has already lowered. Since only high-quality research publications are used in this study, the literature search concentrated on journal articles.

We also mitigated the study inclusion/exclusion bias. Due to previous work, we were aware of the fact that conference papers can be a potential threat to validity [27]. When we started evaluating conference papers, we also found that most of them were compact and written concisely, which meant that they did not provide enough details to satisfy the study's stated research objectives. The extensive review procedures for journals are not comparable with conferences. Journal papers are thoroughly reviewed by scholars in the field before being approved for publication in notable journals. The majority of authors proceed further in their conference papers to have them accepted for publication in highly regarded journals, or the conference organizers promote and allow authors to go more thoroughly in their papers to be accepted for publication in journals. Therefore, only journal articles were included as primary studies in our SLR. Only including journal articles is also applied in previous SLR studies, and substantiates our approach [13,111].

To mitigate the threat of robustness and researcher bias, the majority of our time during the SLR research was dedicated to internal discussions aimed at ensuring the validity of the results. To ensure adherence to research methodology from developing research questions to drawing conclusions based on data obtained. We consider that the measures indicated have addressed the potential risks to this study.

Construct Validity

We also carefully integrated automatic and human search methods to locate the most recent publications. We also closely assessed search performance across writers, because we are aware that each targeted database has its own query style and employs various logical operators. This necessitates switching the query structure between databases. We evaluated the search results by reading the abstracts of the papers that returned. If the papers were not appropriate, the query was revised. The query had to be adjusted, and the search procedure had to be performed again. These actions are continued until the relevant documents are returned by the search query. We also employed snowballing to acquire as many relevant articles as we could as an additional strategy to reduce the likelihood that significant research was missed.

As mentioned, the LCDPs are emerging, and have gained much interest in the academic field. To keep this research representative of the current state-of-the-art developments, we aimed to keep the time between the performed SLR and the performed discussion on the results as short as possible. The primary studies were selected in June 2024, and after that, we focused on only these papers to maximize the relevance of the discussion on the results with significant contributions in the academic field of this subject.

5. Related Work

In this section, we discuss the literature (i.e., other reviews and surveys, not necessarily on PA) regarding LCDPs that was available. Although LCDP continues to play a significant role in many industrial software development processes, the research associated is still in its early stages. Researchers are working to better understand how LCDPs are evolving and what challenges need to be resolved to make them more effective for a wider audience. This is the first study to analyze the fundamental components of LCDPs, with a focus on user-centric paradigms, challenges, and scope for integration and utilization towards PA. We could not obtain a systematic literature review that covered the recent developments in LCDPs for agricultural implementations and purposes. This emphasizes the relevance of the current work, and is a pioneering initiative that lays the groundwork for the systematic review of the state-of-the-art understanding of LCDPs in PA.

Nonetheless, we identified one study that researched the implementation of PA and aimed at gathering knowledge on technologies used for PA [6]. They identify criteria for the comparison and selection of information technologies in PA. They discuss that the required knowledge and type of IT used is a criterion for the implementation of PA. The observations show that there is a scarcity on technology frameworks for the guidance of implementations for PA. They conclude that future research can focus on technologies that can improve the adoption of PA. In our study, we investigate the adoption of LCDPs for PA.

Furthermore, we identified seven studies that provided insights into certain elements of LCDP without focusing on the agriculture domain. These studies will be discussed in the following part of this section.

Using low code development, ref. [112] demonstrates a multi-vocal systematic review. They begin by conducting a review of academic and gray literature. Their research differs from ours in that it focuses on low code development rather than platforms, or "tools", as they refer to platforms. The methodology, foundational technologies, application domains, and advantages of LCD are the primary subjects of their research, which is not necessarily implemented in a concrete tool or platform. Secondly, rather than focusing on the platforms, they evaluate MDE concerning LCD. Ref. [112] revealed the following conclusions about MDE: it is by far the most common of the 27 key technologies that were determined by the study to be essential for LCD. As is further mentioned in the study, to analyze the state of the art in low-code development and make sense of metrics like scalability and performance, future research may include a more thorough analysis of the characteristics and complexity of applications created using this method. Our research questions proceed here, adding to the body of literature.

Ref. [30] provides an overview of eight significant LCDPs and their technical examination. They establish a foundation by identifying the fundamental characteristics that these platforms offer. Their work primarily focuses on specific development facilities offered by the platforms. They examined a select group of LCDPs that have been considered to serve as leaders in the market from recent Gartner and Forrester reports. As opposed to study [30], our research examines a wider variety of LCDPs, which are available in the academic literature. We focus on providing insights that apply to a wider context, particularly in PA. We contribute to their comprehensive review by analyzing the aspects of LCDP validation, such as academic metrics to evaluate LCDP, challenges, and future research directions. They suggest it as future research to refine their taxonomy by considering additional LCDPs. Instead of comparing these platforms or tools, our focus is on the LCDP landscape as it supports the development process.

In [67], the importance of LCDP in enabling agile responses to shift market needs and driving digital transformation in the manufacturing industry is emphasized. They focus on computer-aided software engineering principles as they discuss the developing low-code ecosystem. To identify deficient features, the study assesses the industrial sector's present platforms and does a context analysis of recent low-code research. They employ practitioner interviews as an additional form of empirical research methodology in their study. It leverages the vf-OS platform as an example of a multi-sided, open low-code framework designed to enhance collaboration in manufacturing and logistics. Their research focuses on identifying LCDP features and benefits specifically for the manufacturing area, which is different from our work. We contribute to the literature by investigating a broader classification of the general LCDP ecosystem and use these findings to integrate its applicability to the PA domain.

Ref. [68] focuses primarily on identifying limitations in the LCDPs and techniques that are currently in use in the context of low-code development. The inability of off-the-shelf solutions to align with business transformation goals, the need for improved scalability and adaptability, and the potential for improvement through the use of machine learning algorithms are the three identified limitations. Their study demonstrates the broad range of LCDPs, which has the potential to boost output in a variety of application domains. The goal of future research is to use machine learning techniques to improve the intelligence and robustness of these platforms. Our research proceeds in identifying 'challenges' and contributes by proposing solutions.

In [113], the Socio-Technical System (STS) model is used to conduct a review of the literature and categorize existing publications into technical and social system components. It demonstrates that the majority of research focuses on the technical system, even though a few articles have focused on the social system. This analysis clarifies for practitioners the technical and social aspects of LCDP development, implementation, and use while also making recommendations for future study. Even though our study's objective was different, its conclusions nevertheless benefit the LCDP ecosystem.

In [114], the use of LCDPs among developers is examined. In Saudi Arabia, an online survey of 49 professional developers was conducted, with 19 of the participants using LCDP while the remaining 30 did not. It was discovered that LCDP attracts developers because it enables the creation of applications more quickly and efficiently with little to no coding skills. However, some developers remained hesitant due to scalability problems and a lack of knowledge about these platforms. Overall, the study pinpoints factors that affect LCDP adoption, as well as obstacles that some developers encounter. As was mentioned in their study, to identify and offer appropriate solutions to the issues and obstacles of LCDP, researchers must conduct more studies and assessments. Our research contributes to this by offering a structured classification of challenges and proposed potential solutions.

Ref. [115] emphasizes that LCDPs are not necessarily revolutionary in terms of individual components, but excel at integrating pre-existing design features to speed up routine business application development tasks. What determines whether LCDPs are successful are the technical-economic conditions within specified boundaries and project requirements. It suggests that procedures are needed to assess these circumstances in upcoming studies. Furthermore, the trend toward low code may lead to an increase in conceptual modeling. However, they refer to the ambiguous connotation in the LCDP environment, which may impede communication clarity. The terms in the LCDP environment should be carefully examined before being incorporated into a technical vocabulary. Our research keeps moving toward clarifying the vocabulary by identifying classifications in different aspects of LCDPs.

6. Conclusions

LCDPs have demonstrated significant benefits across various application domains. As PA increasingly relies on software and data-driven applications, LCDPs hold substantial potential to drive efficiency and innovation in this field. Our SLR is the first of its kind to analyze the fundamental components of LCDPs with a specific focus on their applicability to PA, filling a critical gap in the existing research.

The SLR identified the scarcity of systematic research on LCDPs tailored for agricultural purposes, emphasizing the relevance and pioneering nature of this study. While previous studies have explored the role of technologies in PA and the general benefits of LCDPs in various domains, our work uniquely integrates these insights to evaluate the potential of LCDPs in agricultural software development. We found that LCDPs can significantly accelerate the adoption of digital tools in PA, enhance decision-making processes, and promote sustainability through more efficient farming practices.

Future work will focus on the selection of appropriate LCDPs and the development of applications specifically designed for PA. This will involve addressing identified challenges such as integration with IoT, improving scalability and adaptability, and refining the user experience. By advancing research in this area, LCDPs can play an important role in the digital transformation of precision agriculture, optimizing operations and contributing to the sustainability and technological advancement of the agricultural sector.

Author Contributions: Conceptualization, E.G., B.T., Ö.B. and Q.L.; Methodology, E.G., B.T., Ö.B. and Q.L.; Investigation, E.G., B.T., Ö.B. and Q.L.; Writing—original draft, E.G., B.T., Ö.B. and Q.L.; Writing—review & editing, E.G., B.T. and Q.L.; Supervision, B.T. and Ö.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A

Table A1. The 37 selected primary studies.

Study	Year	Study	Year	Study	Year
Almonte et al. [54]	2020	Lethbridge [60]	2021	Sahay et al. [71]	2020
Asawa et al. [42]	2021	Lopes et al. [66]	2021	Salgueiro et al. [43]	2021
Braganca et al. [45]	2021	Lourenco et al. [63]	2021	Schötteler et al. [62]	2021
Da Cruz et al. [37]	2021	Marek et al. [61]	2021	Sharma and Gupta [65]	2021
Daniel et al. [58]	2020	Martins et al. [64]	2020	Silva et al. [39]	2021
Deshpande et al. [34]	2022	Metrôlho et al. [41]	2019	Tisi et al. [38]	2019
Di Sipio et al. [53]	2020	Metrôlho et al. [36]	2020	J. Wang et al. [40]	2021
Fernandes et al. [50]	2020	Moin et al. [46]	2022	Y. Wang et al. [35]	2021
Ferreira and Costa [59]	2021	Pacheco et al. [32]	2021	Waszkowski [55]	2019
Indamutsa et al. [49]	2021	Pantelimon et al. [47]	2019	Weber [31]	2021
Iyer et al. [48]	2021	Philippe et al. [51]	2020	Zhuang et al. [44]	2022
Junior et al. [33]	2020	Pichidtienthum et al. [57]	2021	Zolotas et al. [56]	2018
Jyothi and Rajeswari [69]	2022				

References

1. Lindblom, J.; Lundström, C.; Ljung, M.; Jonsson, A. Promoting sustainable intensification in precision agriculture: Review of decision support systems development and strategies. *Precis. Agric.* **2017**, *18*, 309–331. [CrossRef]
2. Zhang, N.; Wang, M.; Wang, N. Precision agriculture—A worldwide overview. *Comput. Electron. Agric.* **2002**, *36*, 113–132. [CrossRef]
3. Bongiovanni, R.; Lowenberg-DeBoer, J. Precision agriculture and sustainability. *Precis. Agric.* **2004**, *5*, 359–387. [CrossRef]
4. Lal, R. Soil structure and sustainability. *J. Sustain. Agric.* **1991**, *1*, 67–92. [CrossRef]
5. Dobermann, A.; Blackmore, S.; Cook, S.E.; Adamchuk, V.I. Precision farming: Challenges and future directions. In Proceedings of the 4th International Crop Science Congress, Brisbane, Australia, 26 September–1 October 2004; Volume 26.
6. Cisternas, I.; Velásquez, I.; Caro, A.; Rodríguez, A. Systematic literature review of implementations of precision agriculture. *Comput. Electron. Agric.* **2020**, *176*, 105626. [CrossRef]
7. Bhakta, I.; Phadikar, S.; Majumder, K. State-of-the-art technologies in precision agriculture: A systematic review. *J. Sci. Food Agric.* **2019**, *99*, 4878–4888. [CrossRef]
8. Quy, V.K.; Hau, N.V.; Anh, D.V.; Quy, N.M.; Ban, N.T.; Lanza, S.; Randazzo, G.; Muzirafuti, A. IoT-enabled smart agriculture: Architecture, applications, and challenges. *Appl. Sci.* **2022**, *12*, 3396. [CrossRef]
9. Kassim, M.R.M. Iot applications in smart agriculture: Issues and challenges. In Proceedings of the 2020 IEEE Conference on Open Systems (ICOS), Kota Kinabalu, Malaysia, 17–19 November 2020; pp. 19–24.
10. Khanna, A.; Kaur, S. Evolution of Internet of Things (IoT) and its significant impact in the field of Precision Agriculture. *Comput. Electron. Agric.* **2019**, *157*, 218–231. [CrossRef]
11. Tekinerdogan, B. Software Ecosystems for Precision Agriculture. In *Encyclopedia of Smart Agriculture Technologies*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 1–10.
12. Yazdinejad, A.; Zolfaghari, B.; Azmoodeh, A.; Dehghantanha, A.; Karimipour, H.; Fraser, E.; Green, A.G.; Russell, C.; Duncan, E. A review on security of smart farming and precision agriculture: Security aspects, attacks, threats and countermeasures. *Appl. Sci.* **2021**, *11*, 7518. [CrossRef]
13. Pathak, H.S.; Brown, P.; Best, T. A systematic literature review of the factors affecting the precision agriculture adoption process. *Precis. Agric.* **2019**, *20*, 1292–1316. [CrossRef]
14. Sishodia, R.P.; Ray, R.L.; Singh, S.K. Applications of remote sensing in precision agriculture: A review. *Remote Sens.* **2020**, *12*, 3136. [CrossRef]

15. Far, S.T.; Rezaei-Moghaddam, K. Impacts of the precision agricultural technologies in Iran: An analysis experts' perception & their determinants. *Inf. Process. Agric.* **2018**, *5*, 173–184.
16. Jawad, H.M.; Nordin, R.; Gharghan, S.K.; Jawad, A.M.; Ismail, M. Energy-efficient wireless sensor networks for precision agriculture: A review. *Sensors* **2017**, *17*, 1781. [CrossRef] [PubMed]
17. Jensen, K.; Larsen, M.; Nielsen, S.H.; Larsen, L.B.; Olsen, K.S.; Jørgensen, R.N. Towards an open software platform for field robots in precision agriculture. *Robotics* **2014**, *3*, 207–234. [CrossRef]
18. Bhat, S.A.; Huang, N.F. Big data and ai revolution in precision agriculture: Survey and challenges. *IEEE Access* **2021**, *9*, 110209–110222. [CrossRef]
19. Kendall, H.; Naughton, P.; Clark, B.; Taylor, J.; Li, Z.; Zhao, C.; Yang, G.; Chen, J.; Frewer, L. Precision agriculture in China: Exploring awareness, understanding, attitudes and perceptions of agricultural experts and end-users in China. *Adv. Anim. Biosci.* **2017**, *8*, 703–707. [CrossRef]
20. Tan, L.; Haley, R.; Wortman, R.; Zhang, Q. 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 & Integration (IRI), Las Vegas, NV, USA, 8–10 August 2012; pp. 271–278.
21. Robert, P.C. Precision agriculture: A challenge for crop nutrition management. In *Progress in Plant Nutrition: Plenary Lectures of the XIV International Plant Nutrition Colloquium: Food Security and Sustainability of Agro-Ecosystems Through Basic and Applied Research*; Springer: Dordrecht, The Netherlands, 2002; pp. 143–149.
22. Schimmelpfennig, D.; Ebel, R. *On the Doorstep of the Information Age: Recent Adoption of Precision Agriculture*; Paper No. EIB-80; Economic Research Service: Washington, DC, USA, 2011.
23. Kitchenham, B.; Brereton, O.P.; Budgen, D.; Turner, M.; Bailey, J.; Linkman, S. Systematic literature reviews in software engineering–a systematic literature review. *Inf. Softw. Technol.* **2009**, *51*, 7–15. [CrossRef]
24. Gurbuz, H.G.; Tekinerdogan, B. Model-based testing for software safety: A systematic mapping study. *Softw. Qual. J.* **2018**, *26*, 1327–1372. [CrossRef]
25. Köksal, Ö.; Tekinerdogan, B. Obstacles in data distribution service middleware: A systematic review. *Future Gener. Comput. Syst.* **2017**, *68*, 191–210. [CrossRef]
26. Tummers, J.; Kassahun, A.; Tekinerdogan, B. Obstacles and features of Farm Management Information Systems: A systematic literature review. *Comput. Electron. Agric.* **2019**, *157*, 189–204. [CrossRef]
27. Krisnawijaya, N.N.K.; Tekinerdogan, B.; Catal, C.; van der Tol, R. Data analytics platforms for agricultural systems: A systematic literature review. *Comput. Electron. Agric.* **2022**, *195*, 106813. [CrossRef]
28. Hsieh, H.F.; Shannon, S.E. Three approaches to qualitative content analysis. *Qual. Health Res.* **2005**, *15*, 1277–1288. [CrossRef] [PubMed]
29. Potter, W.J.; Levine-Donnerstein, D. Rethinking validity and reliability in content analysis. *J. Appl. Commun. Res.* **1999**, *27*, 258–284. [CrossRef]
30. Sahay, A.; Indamutsa, A.; Di Ruscio, D.; Pierantonio, A. Supporting the understanding and comparison of low-code development platforms. In Proceedings of the 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Portoroz, Slovenia, 26–28 August 2020; pp. 171–178.
31. Weber, I. Low-code from frontend to backend: Connecting conversational user interfaces to backend services via a low-code IoT platform. In Proceedings of the 3rd Conference on Conversational User Interfaces, Bilbao, Spain, 27–29 July 2021; pp. 1–5.
32. Pacheco, J.; Garbatov, S.; Goulão, M. Improving collaboration efficiency between UX/UI designers and developers in a low-code platform. In Proceedings of the 2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), Fukuoka, Japan, 10–15 October 2021; pp. 138–147.
33. Junior, A.G.D.S.; Gonçalves, L.M.G.; Caurin, G.A.D.P.; Tamanaka, G.T.B.; Hernandes, A.C.; Aroca, R.V. BIPES: Block based integrated platform for embedded systems. *IEEE Access* **2020**, *8*, 197955–197968. [CrossRef]
34. Deshpande, A.; Kothari, D.; Salvi, A.; Mane, P.; Kolhe, V. Querylizer: An Interactive Platform for Database Design and Text to SQL Conversion. In Proceedings of the 2022 International Conference for Advancement in Technology (ICONAT), Goa, India, 21–22 January 2022; pp. 1–6.
35. Wang, Y.; Feng, Y.; Zhang, M.; Sun, P. The necessity of low-code engineering for industrial software development: A case study and reflections. In Proceedings of the 2021 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), Wuhan, China, 25–28 October 2021; pp. 415–420.
36. Metrolho, J.; Ribeiro, F.; Araujo, R. A strategy for facing new employability trends using a low-code development platform. In Proceedings of the INTED2020 Proceedings, IATED, Valencia, Spain, 2–4 March 2020; pp. 8601–8606.
37. Da Cruz, M.A.; de Paula, H.T.; Caputo, B.P.; Mafra, S.B.; Lorenz, P.; Rodrigues, J.J. Olp—A restful open low-code platform. *Future Internet* **2021**, *13*, 249. [CrossRef]
38. Tisi, M.; Mottu, J.M.; Kolovos, D.S.; De Lara, J.; Guerra, E.M.; Di Ruscio, D.; Pierantonio, A.; Wimmer, M. Lowcomote: Training the next generation of experts in scalable low-code engineering platforms. In Proceedings of the 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), Eindhoven, The Netherlands, 15–19 July 2019.

39. Silva, C.; Vieira, J.; Campos, J.C.; Couto, R.; Ribeiro, A.N. Development and validation of a descriptive cognitive model for predicting usability issues in a low-code development platform. *Hum. Factors* **2021**, *63*, 1012–1032. [CrossRef]
40. Wang, J.; Qi, B.; Zhang, W.; Sun, H. A Low-Code Development Framework for Constructing Industrial Apps. In Proceedings of the Computer Supported Cooperative Work and Social Computing: 15th CCF Conference, ChineseCSCW 2020, Shenzhen, China, 7–9 November 2020; Revised Selected Papers 15; pp. 237–250.
41. Metrôlho, J.; Araújo, R.; Ribeiro, F.; Castela, N. An approach using a low-code platform for retraining professionals to ICT. In Proceedings of the EDULEARN19 Proceedings, IATED, Palma, Spain, 1–3 July 2019; pp. 7200–7207.
42. Asawa, K.; Kukreja, S.; Gondkar, R. An NCDP for developing a blockchain based dynamic supply chain management with auto-generation of smart contract. In Proceedings of the 2021 26th International Conference on Automation and Computing (ICAC), Portsmouth, UK, 2–4 September 2021; pp. 1–6.
43. Salgueiro, J.; Ribeiro, F.; Metrôlho, J. Best Practices for OutSystems Development and Its Influence on Test Automation. In*Trends and Applications in Information Systems and Technologies*; Springer, Cham, Switzerland, 2021; Volume 1368, pp. 85–95.
44. Zhuang, W.; Gan, X.; Wen, Y.; Zhang, S. Easyfl: A low-code federated learning platform for dummies. *IEEE Internet Things J.* **2022**, *9*, 13740–13754. [CrossRef]
45. Bragança, A.; Azevedo, I.; Bettencourt, N.; Morais, C.; Teixeira, D.; Caetano, D. Towards supporting SPL engineering in low-code platforms using a DSL approach. In Proceedings of the 20th ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences, Chicago, IL, USA, 17–18 October 2021; pp. 16–28.
46. Moin, A.; Mituca, A.; Challenger, M.; Badii, A.; Günnemann, S. ML-Quadrat & DriotData: A model-driven engineering tool and a low-code platform for smart IoT services. In Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings, Pittsburgh, PA, USA, 21–29 May 2022; pp. 144–148.
47. Pantelimon, S.G.; Rogojanu, T.; Braileanu, A.; Stanciu, V.D.; Dobre, C. Towards a seamless integration of iot devices with iot platforms using a low-code approach. In Proceedings of the 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, 15–18 April 2019; pp. 566–571.
48. Iyer, C.K.; Hou, F.; Wang, H.; Wang, Y.; Oh, K.; Ganguli, S.; Pandey, V. Trinity: A No-Code AI platform for complex spatial datasets. In Proceedings of the 4th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery, Beijing, China, 2–5 November 2021; pp. 33–42.
49. Indamutsa, A.; Di Ruscio, D.; Pierantonio, A. A low-code development environment to orchestrate model management services. In Proceedings of the IFIP International Conference on Advances in Production Management Systems, Nantes, France, 5–9 September 2021; Springer: Cham, Switzerland, 2021; pp. 342–350.
50. Fernandes, J.P.; Araújo, R.; Zenha-Rela, M. Achieving scalability in project based learning through a low-code platform. In Proceedings of the XXXIV Brazilian Symposium on Software Engineering, Natal, Brazil, 21–23 October 2020; pp. 710–719.
51. Philippe, J.; Coullon, H.; Tisi, M.; Sunyé, G. Towards transparent combination of model management execution strategies for low-code development platforms. In Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, Virtual, 16–23 October 2020; pp. 1–10.
52. El-Sappagh, S.H.A.; Hendawi, A.M.A.; El Bastawissy, A.H. A proposed model for data warehouse ETL processes. *J. King Saud Univ.-Comput. Inf. Sci.* **2011**, *23*, 91–104. [CrossRef]
53. Di Sipio, C.; Di Ruscio, D.; Nguyen, P.T. Democratizing the development of recommender systems by means of low-code platforms. In Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, Virtual, 16–23 October 2020; pp. 1–9.
54. Almonte, L.; Cantador, I.; Guerra, E.; de Lara, J. Towards automating the construction of recommender systems for low-code development platforms. In Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, Virtual, 16–23 October 2020; pp. 1–10.
55. Waszkowski, R. Low-code platform for automating business processes in manufacturing. *IFAC-PapersOnLine* **2019**, *52*, 376–381. [CrossRef]
56. Zolotas, C.; Chatzidimitriou, K.C.; Symeonidis, A.L. RESTsec: A low-code platform for generating secure by design enterprise services. *Enterp. Inf. Syst.* **2018**, *12*, 1007–1033. [CrossRef]
57. Pichidtienthum, S.; Pugsee, P.; Cooharojananone, N. Developing Module Generation for Odoo Using Concept of Low-Code Development Platform and Automation Systems. In Proceedings of the 2021 IEEE 8th International Conference on Industrial Engineering and Applications (ICIEA), Chengdu, China, 23–26 April 2021; pp. 529–533.
58. Daniel, G.; Cabot, J.; Deruelle, L.; Derras, M. Xatkit: A Multimodal Low-Code Chatbot Development Framework. *IEEE Access* **2020**, *8*, 15332–15346. [CrossRef]
59. Ferreira, J.; Costa, C. Web Platform for Medical Deep Learning Services. In Proceedings of the 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Houston, TX, USA, 9–12 December 2021; pp. 1727–1732.
60. Lethbridge, T.C. Low-code is often high-code, so we must design low-code platforms to enable proper software engineering. In *Proceedings 10, Proceedings of the Leveraging Applications of Formal Methods, Verification and Validation: 10th International Symposium on Leveraging Applications of Formal Methods, ISoLA 2021, Rhodes, Greece, 17–29 October 2021*; Springer: Cham, Switzerland, 2021; pp. 202–212.

61. Marek, K.; Śmiałek, M.; Rybiński, K.; Roszczyk, R.; Wdowiak, M. BalticLSC: Low-code software development platform for large scale computations. *Comput. Inform.* **2021**, *40*, 734–753. [CrossRef]
62. Schötteler, S.; Laumer, S.; Schuhbauer, H.; Scheidthauer, N.; Seeberger, P.; Miethsam, B. A no-code platform for tie prediction analysis in social media networks. In *Proceedings of the Innovation Through Information Systems: Volume II: A Collection of Latest Research on Technology Issues*; Springer: Cham, Switzerland, 2021; pp. 475–491.
63. Lourenço, H.; Ferreira, C.; Seco, J.C. OSTRICH-a type-safe template language for low-code development. In Proceedings of the 2021 ACM/IEEE 24th International Conference on Model Driven Engineering Languages and Systems (MODELS), Fukuoka, Japan, 10–15 October 2021; pp. 216–226.
64. Martins, R.; Caldeira, F.; Sa, F.; Abbasi, M.; Martins, P. An overview on how to develop a low-code application using OutSystems. In Proceedings of the 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), Bengaluru, India, 9–10 October 2020; pp. 395–401.
65. Sharma, U.; Gupta, D. Email Ingestion Using Robotic Process Automation for Online Travel Agency. In Proceedings of the 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 3–4 September 2021; pp. 1–5.
66. Lopes, B.; Amorim, S.; Ferreira, C. Solution discovery over feature toggling with built-in abstraction in outsystems. In Proceedings of the 2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), Fukuoka, Japan, 10–15 October 2021; pp. 47–56.
67. Sanchis, R.; García-Perales, Ó.; Fraile, F.; Poler, R. Low-code as enabler of digital transformation in manufacturing industry. *Appl. Sci.* **2019**, *10*, 12. [CrossRef]
68. Phalake, V.S.; Joshi, S.D. Low code development platform for digital transformation. In Proceedings of the Information and Communication Technology for Competitive Strategies (ICTCS 2020) Intelligent Strategies for ICT, Jaipur, India, 11–12 December 2021; Springer: Singapore, 2021; pp. 689–697.
69. Jyothi, S.; Rajeswari, T. Accelerating SQL with Complex Visual Querying. In Proceedings of the 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS), Coimbatore, India, 23–25 February 2022; pp. 1126–1131.
70. Martin, J. *Rapid Application Development*; Macmillan Publishing Co., Inc.: New York, NY, USA, 1991.
71. Sahay, A.; Di Ruscio, D.; Pierantonio, A. Understanding the role of model transformation compositions in low-code development platforms. In Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, Virtual, 16–23 October 2020; pp. 1–5.
72. Abbasi, R.; Martinez, P.; Ahmad, R. The digitization of agricultural industry—A systematic literature review on agriculture 4.0. *Smart Agric. Technol.* **2022**, *2*, 100042. [CrossRef]
73. Groeneveld, D.; Tekinerdogan, B.; Garousi, V.; Catal, C. A domain-specific language framework for farm management information systems in precision agriculture. *Precis. Agric.* **2021**, *22*, 1067–1106. [CrossRef]
74. Verdouw, C.; Tekinerdogan, B.; Beulens, A.; Wolfert, S. Digital twins in smart farming. *Agric. Syst.* **2021**, *189*, 103046. [CrossRef]
75. Barriga, J.A.; Clemente, P.J.; Sosa-Sánchez, E.; Prieto, Á.E. SimulateIoT: Domain Specific Language to design, code generation and execute IoT simulation environments. *IEEE Access* **2021**, *9*, 92531–92552. [CrossRef]
76. Holzworth, D.P.; Snow, V.; Janssen, S.; Athanasiadis, I.N.; Donatelli, M.; Hoogenboom, G.; White, J.W.; Thorburn, P. Agricultural production systems modelling and software: Current status and future prospects. *Environ. Model. Softw.* **2015**, *72*, 276–286. [CrossRef]
77. Shafi, U.; Mumtaz, R.; García-Nieto, J.; Hassan, S.A.; Zaidi, S.A.R.; Iqbal, N. Precision agriculture techniques and practices: From considerations to applications. *Sensors* **2019**, *19*, 3796. [CrossRef]
78. López-Riquelme, J.; Pavón-Pulido, N.; Navarro-Hellín, H.; Soto-Valles, F.; Torres-Sánchez, R. A software architecture based on FIWARE cloud for Precision Agriculture. *Agric. Water Manag.* **2017**, *183*, 123–135. [CrossRef]
79. Nikkilä, R.; Seilonen, I.; Koskinen, K. Software architecture for farm management information systems in precision agriculture. *Comput. Electron. Agric.* **2010**, *70*, 328–336. [CrossRef]
80. Tantalaki, N.; Souravlas, S.; Roumeliotis, M. Data-driven decision making in precision agriculture: The rise of big data in agricultural systems. *J. Agric. Food Inf.* **2019**, *20*, 344–380. [CrossRef]
81. Pogorelskaia, I.; Várallyai, L. Agriculture 4.0 and the role of education. *J. Agric. Inform./Agrárinform. Folyóirat* **2020**, *11*, 45–51. [CrossRef]
82. Gunasekera, K.; Borrero, A.N.; Vasuian, F.; Bryceson, K.P. Experiences in building an IoT infrastructure for agriculture education. *Procedia Comput. Sci.* **2018**, *135*, 155–162. [CrossRef]
83. Yousefi, M.R.; Razdari, A.M. Application of GIS and GPS in precision agriculture (a review). *Int. J. Adv. Biol. Biomed. Res.* **2015**, *3*, 7–9.
84. Kamble, S.S.; Gunasekaran, A.; Gawankar, S.A. Achieving sustainable performance in a data-driven agriculture supply chain: A review for research and applications. *Int. J. Prod. Econ.* **2020**, *219*, 179–194. [CrossRef]
85. Liakos, K.G.; Busato, P.; Moshou, D.; Pearson, S.; Bochtis, D. Machine learning in agriculture: A review. *Sensors* **2018**, *18*, 2674. [CrossRef]
86. Baio, F.H.; Silva, S.P.d.; Camolese, H.d.S.; Neves, D.C. Financial analysis of the investment in precision agriculture techniques on cotton crop. *Eng. Agrícola* **2017**, *37*, 838–847. [CrossRef]

87. Silva, C.B.; do Vale, S.M.L.R.; Pinto, F.A.; Müller, C.A.; Moura, A.D. The economic feasibility of precision agriculture in Mato Grosso do Sul State, Brazil: A case study. *Precis. Agric.* **2007**, *8*, 255–265. [CrossRef]
88. Lowenberg-DeBoer, J.; Erickson, B. Setting the record straight on precision agriculture adoption. *Agron. J.* **2019**, *111*, 1552–1569. [CrossRef]
89. Kamilaris, A.; Kartakoullis, A.; Prenafeta-Boldú, F.X. A review on the practice of big data analysis in agriculture. *Comput. Electron. Agric.* **2017**, *143*, 23–37. [CrossRef]
90. Sharma, A.; Jain, A.; Gupta, P.; Chowdary, V. Machine learning applications for precision agriculture: A comprehensive review. *IEEE Access* **2020**, *9*, 4843–4873. [CrossRef]
91. Emrullah, A.; Ozerdem, M.S.; Ustundag, B.B. Machine learning based regression model for prediction of soil surface humidity over moderately vegetated fields. In Proceedings of the 2019 8th International Conference on Agro-Geoinformatics (Agro-Geoinformatics), Istanbul, Turkey, 16–19 July 2019; pp. 1–4.
92. Wolfert, S.; Goense, D.; Sørensen, C.A.G. A future internet collaboration platform for safe and healthy food from farm to fork. In Proceedings of the 2014 Annual SRII Global Conference, San Jose, CA, USA, 23–25 April 2014; pp. 266–273.
93. Kutter, T.; Tiemann, S.; Siebert, R.; Fountas, S. The role of communication and co-operation in the adoption of precision farming. *Precis. Agric.* **2011**, *12*, 2–17. [CrossRef]
94. Aceto, G.; Persico, V.; Pescapé, A. A survey on information and communication technologies for industry 4.0: State-of-the-art, taxonomies, perspectives, and challenges. *IEEE Commun. Surv. Tutorials* **2019**, *21*, 3467–3501. [CrossRef]
95. Arcury, T.A.; Estrada, J.M.; Quandt, S.A. Overcoming language and literacy barriers in safety and health training of agricultural workers. *J. Agromedicine* **2010**, *15*, 236–248. [CrossRef]
96. Fomenko, T.; Bilotserkovets, M.; Klochkova, T.; Statsenko, O.; Sbruieva, A.; Kozlova, O.; Kozlov, D. Overcoming barriers in intercultural communication: A case study on agricultural idioms in English, Ukrainian and Chinese. *Acad. J. Interdiscip. Stud.* **2020**, *9*, 157–166. [CrossRef]
97. Stafford, J.V. Implementing precision agriculture in the 21st century. *J. Agric. Eng. Res.* **2000**, *76*, 267–275. [CrossRef]
98. Liu, W.; Shao, X.F.; Wu, C.H.; Qiao, P. A systematic literature review on applications of information and communication technologies and blockchain technologies for precision agriculture development. *J. Clean. Prod.* **2021**, *298*, 126763. [CrossRef]
99. Yao, Z.; Liu, G.; Zhang, D.; Shen, Y.; Wang, Z. Path Tracking Control for Four-Wheel-Independent-Driven Agricultural High Clearance Sprayer with New Front-Rear-Dual-Steering-Axle. In Proceedings of the 2020 Chinese Automation Congress (CAC), Shanghai, China, 6–8 November 2020; pp. 2966–2972.
100. Torky, M.; Hassanein, A.E. Integrating blockchain and the internet of things in precision agriculture: Analysis, opportunities, and challenges. *Comput. Electron. Agric.* **2020**, *178*, 105476. [CrossRef]
101. Wolfert, S.; Ge, L.; Verdouw, C.; Bogaardt, M.J. Big data in smart farming—A review. *Agric. Syst.* **2017**, *153*, 69–80. [CrossRef]
102. Frank, L.; Wiegman, C.; Davis, J.; Shearer, S. Confidence-driven hierarchical classification of cultivated plant stresses. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Virtual, 5–9 January 2021; pp. 2503–2512.
103. Nicoletti, J.V.M.; De Miranda, J.H.; Cooke, R.; Christianson, L.E.; De Oliveira, L.A. Efficiency analysis of machine learning models for simulating nitrate movement in soils from Illinois State. In Proceedings of the 2021 ASABE Annual International Virtual Meeting, Virtual, 12–16 July 2021; American Society of Agricultural and Biological Engineers: St. Joseph, MI, USA, 2021; p. 1.
104. Bosona, T.; Gebresenbet, G. Food traceability as an integral part of logistics management in food and agricultural supply chain. *Food Control* **2013**, *33*, 32–48. [CrossRef]
105. Leroux, N.; Wortman, M.S., Jr.; Mathias, E.D. Dominant factors impacting the development of business-to-business (B2B) e-commerce in agriculture. *Int. Food Agribus. Manag. Rev.* **2001**, *4*, 205–218. [CrossRef]
106. Belhadi, A.; Kamble, S.S.; Mani, V.; Benkhati, I.; Touriki, F.E. An ensemble machine learning approach for forecasting credit risk of agricultural SMEs' investments in agriculture 4.0 through supply chain finance. *Ann. Oper. Res.* **2021**, *306*, 1–29. [CrossRef] [PubMed]
107. Kyriazi, F.; Thomakos, D.D.; Guerard, J.B. Adaptive learning forecasting, with applications in forecasting agricultural prices. *Int. J. Forecast.* **2019**, *35*, 1356–1369. [CrossRef]
108. Van Geest, M.; Tekinerdogan, B.; Catal, C. Design of a reference architecture for developing smart warehouses in industry 4.0. *Comput. Ind.* **2021**, *124*, 103343. [CrossRef]
109. Zhou, X.; Jin, Y.; Zhang, H.; Li, S.; Huang, X. A map of threats to validity of systematic literature reviews in software engineering. In Proceedings of the 2016 23rd Asia-Pacific Software Engineering Conference (APSEC), Hamilton, New Zealand, 6–9 December 2016; pp. 153–160.
110. Ampatzoglou, A.; Bibi, S.; Avgeriou, P.; Verbeek, M.; Chatzigeorgiou, A. Identifying, categorizing and mitigating threats to validity in software engineering secondary studies. *Inf. Softw. Technol.* **2019**, *106*, 201–230. [CrossRef]
111. Catal, C.; Giray, G.; Tekinerdogan, B. Applications of deep learning for mobile malware detection: A systematic literature review. *Neural Comput. Appl.* **2022**, *34*, 1007–1032 . [CrossRef]
112. Bucaioni, A.; Cicchetti, A.; Ciccozzi, F. Modelling in low-code development: A multi-vocal systematic review. *Softw. Syst. Model.* **2022**, *21*, 1959–1981. [CrossRef]
113. Prinz, N.; Rentrop, C.; Huber, M. Low-Code Development Platforms-A Literature Review. In Proceedings of the AMCIS, Virtual, 9–13 August 2021.

114. Alsaadi, H.A.; Radain, D.T.; Alzahrani, M.M.; Alshammari, W.F.; Alahmadi, D.; Fakieh, B. Factors that affect the utilization of low-code development platforms: Survey study. *Rom. J. Inf. Technol. Autom. Control. Română Inform. Autom.* **2021**, *31*, 123–140. [CrossRef]
115. Bock, A.C.; Frank, U. Low-code platform. *Bus. Inf. Syst. Eng.* **2021**, *63*, 733–740. [CrossRef]