Journal of Computer Science & Computational Mathematics, *Volume 15, Issue 01, March 2025*
DOI: 10.20967/jcscm.2025.01.001

Managing Workforce Challenges in Software Development: Towards A Citizen Development Model for Collaborative Innovation

Imran Harith Azmy*, Azri Azmi, Nazri Kama, Hazlifah Mohd Rusli

Faculty of Artificial Intelligence, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia
*Corresponding author email: imranharithazmy@graduate.utm.my

***Abstract*:** The increasing complexity of software development, coupled with a growing demand for software products, has led to a shortage of professionals with the necessary programming skills, resulting in challenges in adhering to project schedules. Citizen Development has emerged as a promising solution to address these challenges by incorporating non-developer personnel into the software development process. This research investigates Software Engineers' perspectives on Citizen Development through a comprehensive survey, revealing significant optimism about the potential of such collaboration. Building on these findings, a software development model has been developed, offering a structured framework to integrate Citizen Developers and traditional developers effectively. This preliminary model addresses workforce challenges and enhances collaborative innovation in software development. Future works will focus on detailing implementation processes and validating the model's practicality in real-world applications, paving the way for advancements in software development methodologies.

***Keywords*:** Software development, project management, citizen development, citizen developers, non-programmers, rapid application development.

1. Introduction

Traditional software development has long relied on individuals with high levels of specialized expertise to navigate the complexities of ever-more-complex systems [1]. Significant demands are placed on professional software developers due to the competitive nature of the industry, which intensifies the pressure to deliver quick results [2–5]. In addition, a recent report by a leading global market research firm, Forrester Research, predicts that the United States will face a shortage of over 500,000 software developers in the coming years [6, 7].

The evolution of application development methodologies has enabled non-programmers, also known as 'Citizen Developers', to create software. Citizen Development is a method of software development that allows non-programmers to participate in the application development process [8–10]. Citizen Development reduces the need for intermediaries such as technical developers and encourages business personnel to design applications to meet their immediate needs without waiting for developers or having a budget for the eventual development of such applications [11]. Citizen Developers are frequently able to work with IT teams and professional developers to design and deliver software, integrations, automations, business intelligence or analytics, or artificial intelligence (AI) models [10, 12–14]. New digital literacy skills and their practical applications are required for personal and professional development across all disciplines [15]. These capabilities depend on the workers' knowledge, experience, and other non-professional skills [16]; the workers, in turn, depend on these qualities.

Citizen Development is the participation of non-programmers, typically existing team members, in application development tasks. In order to bridge the gap between the industry's high demands for software developers, their heavy workloads, and the dearth of in-house software developers, citizen development platforms and methods have become increasingly popular and essential [17]. It is believed that by leveraging the domain expertise and creativity of these non-programmers, organizations can expand their software development capabilities and reduce the burden on professional software developers. According to a recent report by Gartner (2019), the number of Citizen Developers will surpass that of professional software developers in the near future [18, 19]. In this study, a survey to ascertain software developers' perceptions of citizen development practices has been conducted.

2. Literature Review

Those with extensive programming and software development knowledge, or those who have earned formal education in a discipline relevant to software development (at the degree or professional level), are referred to as 'programmers' [20]. Those who have not studied software development or any related courses, who have no prior programming experience, or who are extremely new to programming are referred to as 'non-programmers' [20]. Due to the growing adoption of automated application creation, distribution, and maintenance methods, the level of technical expertise necessary for application development has significantly dropped. As a result, platform-specific apps could now be created by a wide range of inexperienced developers.

The term 'Citizen Developers'—rather than just 'non-programmers'—has recently gained significant traction in the IT industry to describe software developers with 'mini-

mal or no programming or software engineering background' [21, 22]. When working with IT teams and professional developers to design and deliver software, integrations, automations, business intelligence/analytics, or artificial intelligence (AI) models, a citizen developer is typically a non-programmer, business-oriented power user [13].

2.1 Software Development and Non-Programmers

"Programmers" are defined as individuals who possess substantial programming and software development experience or who have completed formal education in a software development-related field - at the degree or professional level [20]. Meanwhile, the term "non-programmers" are people who have never taken software development or related courses, have no prior programming experience, or are relatively new to the language [20]. Because automated methods for creating, delivering, and maintaining applications are so common, the technical skills needed for application development have significantly decreased.

The widespread use of automated methods for creating, delivering, and maintaining software has significantly reduced the level of technical skills needed for application development. This made it possible for a large group of inexperienced developers to create platform-specific software [21]. Instead of simply being "non-programmers," the trend of using developers with "minimal or no programming or software engineering background" to create software has gained significant traction in the IT industry recently and is now known as "Citizen Developers" [22]. To work with IT teams and professional developers to design and deliver software, integrations, automations, business intelligence/analytics, or artificial intelligence (AI) models, a citizen developer is typically a business-oriented power user who is not a programmer and is sufficiently skilled with common technical tools [13].

2.2 Emergence of Citizen Developers

In a survey by Thacker et al. (2021) on business students have revealed that despite the initial reluctance of most of the respondents, they would cherish the chance to work as technology developers. The hesitation, though, is brought on by a lack of knowledge and expertise in technological development. The findings highlight a significant opportunity that suggests non-programmers, including business students, would have a strong desire to build digital technologies but would face obstacles like a lack of practical technical knowledge and expertise [10]. Any non-technical people who work in business-related roles or who are at least business-educated may be a promising candidate to be an effective 'Citizen Developer' in order to effectively and efficiently solve business problems by creating digital solutions if they have a good understanding of business processes.

In a study, Hoogsteen and Borgman (2022) addressed factors that may influence organizational citizen development adoption decisions, such as decisions on end-user participation in software development processes [9]. The causes are centered on the need for organizations to digitally convert their internal information systems in order to keep up with the market's rapid shift in demand [23]. This digital transformation tsunami has considerably raised the need for software and software developers, and IT departments are struggling to keep up with the growing backlog of such business innovation-related work. Types of end users, types of organizations, and types of technologies have all been classified as dependencies. Due to worries about integration challenges, it was advised that businesses must select the suitable use cases for citizen development from the discussed dependencies, but they still require less technical skill [9].

In addition, citizen developers could remove the need for an intermediary in IT, allowing for the rapid development of prototypes that increase efficiency and productivity while preventing miscommunication between the business and IT domains. According to a study published in [21], allowing citizen developers to create software would facilitate the automation and simplification of software development activities, as well as the integration of business knowledge, since ideas could be readily transformed into solutions. Nevertheless, it is more important than ever to equip the workforce with the proper technological tools, as today's and tomorrow's workers expect technology to be robust and easy to use, as they have grown up with the internet, are highly collaborative, and use technology more instinctively than previous generations [22]. In addition, many IT organizations and businesses struggle with how to equip their non-IT employees with technologies in order to maximize their intellectual potential and cover the growing software developer shortage. When regarded in this manner, citizen developers can be an opportunity for organizations and businesses to capitalize on these population and workplace trends.

2.3 Potentials for Citizen Development

Even though the majority of respondents initially expressed hesitancy, a study on business students by Thacker et al. (2021) revealed that they would value the chance to work as technology developers. But the reluctance stems from a lack of knowledge and expertise in technology development. According to the findings, there is a significant chance that non-programmers, like business students, would be highly motivated and interested in creating digital technologies, but they would face obstacles like a lack of practical technical knowledge and experience [10]. Hence, any non-technical individuals working in business-related roles or at least having a business education may be a good candidate to be a "Citizen Developer" in order to effectively and efficiently solve business problems by creating digital solutions. This is because they need to have a good understanding of business processes.

Main factor that may influence organisational adoption decisions on citizen development is the decisions related to including non-technical roles participation in software development processes [9]. The motivation that drives the factor is to keeping up with the rapid changes in market demands, which require businesses to digitally transform their internal processes and capabilities [23]. This wave of digital transformation has led to a significant increase in the demand for

software and software developers, while IT departments are struggling to keep up with the growing backlog of tasks related to business innovation. Three dependencies have been identified: end-user type, organisation type, and technology type. Though citizen developers still require less technical competence, it was advised that organisations select the right use cases for citizen development because of concerns about integration problems [9].

2.4 Significance of Citizen Developments

Numerous studies and projections suggest that a substantial portion of future applications will be created by Citizen Developers. Over 70 percent of respondents to a survey by Lebens and Finnegan (2021) on the adaptation of Citizen Development in business organizations concurred that employees outside their IT department develop technology solutions using Citizen Development tools [24]. Vincent et al. (2019) predicts that 65 percent of applications will be developed by Citizen Developers, supporting diverse use cases including reporting, analysis, event processing, user interfaces, data services, and business logic [25]. In addition, over ninety percent of significant organizations plan to use or are considering the use of Citizen Development platforms in the coming years [18]. Moreover, over 75 percent of organizations have employed no-code/low-code platforms, according to a global survey conducted in 2021 [6].

Consistently, these studies indicate that Citizen Developers will play a significant role in application development now and in the future, highlighting the growing significance and potential of Citizen Development in influencing the landscape of software development. It is evident, based on these studies and projections, that organizations view Citizen Development as a valuable resource for addressing software development challenges and leveraging the diversified skills and expertise of non-programmers. Citizen Development is the participation of non-programmers, typically existing team members, in application development duties. Citizen Development platforms and methods have grown in prominence and significance as the demand for professional software developers increases and the number of in-house software developers remains limited [17].

By leveraging the domain expertise and inventiveness of non-programmers, organizations can expand their software development capabilities and reduce the workload of professional software developers. In the near future, Citizen Developers could surpass professional software developers, according to Gartner [18, 19]. In addition, a study of six significant corporations found that workforce empowerment was one of the top reasons for adopting Citizen Development [9]. This indicates that businesses recognize the importance of empowering their personnel through Citizen Development initiatives. Citizen Development empowers individuals within an organization to actively contribute to software development and fuel innovation, which is consistent with its broader impact beyond addressing manpower issues.

Moreover, according to a report by Kintone (2017), Citizen Developers have emerged as a key factor in the rise of digital enterprises [26]. This highlights the growing recognition of non-programmers' contributions to digital transformation and innovation. Citizen Development as a strategic approach capitalizes on the unrealized potential of the workforce and maintains organizations' competitiveness in the swiftly evolving digital age. Citizen Development is therefore viewed as an essential solution to the aforementioned software development issues.

2.5 Citizen Development Method: Low-Code Development

Arora et al. (2020) discussed Sagitec Software Studio (S3), which is a technology developed by Sagitec Solutions LLC as a low-code development platform that enables users to build applications using 'drag and drop' and very little coding [27]. S3 is currently being used to develop enterprise-level applications for healthcare, insurance, and pension use cases. In the meantime, Waszkowski (2019) describes Aurea Business Process Modelling (BPM) as a low-code development platform that has been proved to offer solutions that enable the automation of business processes in the manufacturing industry [4]. On the other hand, Sahinaslan et al. (2021) analyzed the SetXRM low-code development platform, which promised practicable and adaptable solutions in the short term [28]. In addition, Gurcan and Taentzer (2021) analyzed three well-known low-code development technologies, namely Microsoft PowerApps, Mendix, and OutSystems [29]. Microsoft's PowerApps is a cloud-based low-code platform that enables the creation of business applications with minimal programming, with a primary emphasis on data, logic, user interface, and visualization. Mendix is a mobile and web application low-code development platform that provides a collaborative integrated development environment. OutSystems, meanwhile, is comparable to PowerApps, a cloud-based low-code development platform that offers both web and mobile app development. However, it is argued that the majority of low-code developments are limited in terms of customizability because the design of the developed product is highly predetermined, or also described as 'What You See is What You Get' (WYSIWYG) [29]. Low-code development platforms may not be adaptable to new technical requirements [5] and may have critical security issues [21], according to another argument.

2.6 Citizen Development Method: No-Code Development

In contrast to low-code development, it has been identified that AppSheet as a well-known and prospective no-code development platform where no programming is required during the development process [30]. The research and testing conducted by Nurharjadmo et al. (2022) on micro, small, and medium-sized businesses utilizing AppSheet suggests that the technology has positive influences and effects, particularly in terms of reducing costs and time by a considerable amount. However, numerous publications have used the terms 'no-code' and 'low-code' interchangeably, or in some cases as 'low/no-code' [31, 32]. However, Di Ruscio et al.

(2022) have argued that the term 'no-code development platform' is used for platforms that require zero programming of any visual languages, configurations, or graphical user interfaces; however, due to opposition by market analysis firms to identify it as a distinct market segment, 'low-code development platform' is widely and to a certain extent, the only term used [30]. Therefore, for the purposes of this study, we only considered literatures that explicitly defined the specific technology as 'no-code development'.

2.7 Citizen Development Method: Spreadsheet-Driven Development

Benson et al. (2014) describe 'Quilt', a responsive read-write system for constructing a basic Web application using spreadsheets as servers and HTML to describe the client-side UI [33]. Similarly, Chang and Myers (2017) have proposed a method for the development of a lightweight data management system using a requirements table in spreadsheets that parses the table model and implements the objective system [34]. They have developed the 'Gneiss System', which combines a web interface builder with a spreadsheet editor to enable end-users to bind Web GUI elements and spreadsheet cell attributes for operating the actual service data of the application [34]. However, Yang et al. (2019) it has been identified that the critical issue of the spreadsheet-driven development, namely the lack of a clear and convincing method to connect the spreadsheet to the Internet, preventing the development of an effective data management system [35].

2.8 Citizen Development Method: Automatic Code Generator

Automatic code generators are essential for a rapid application development that simplify the creation of common application features, such as CRUD-based software functionality. Benouda et al. (2018) proposed a technique for automated code generation based on Model Driven Architecture and the 'Develop Once, Run Everywhere' principle, streamlining and accelerating the development process [36]. Inayatullah et al. (2019) proposed a model-driven framework for cross-platform web and mobile applications that generates scaffolding code for CRUD operations features. They developed a Domain Specific Modeling Language (DSML) based on the Unified Modeling Language (UML) with multiple data types and stereotypes [37].

A comprehensive open-source transformation engine was developed to autonomously generate scaffolding CRUD code for cross-platform applications from the class diagram of high-level models. Anuar et al. (2022) developed a code automation tool called Re-CRUD, which generates a comprehensive and integrated electronic records management system for web application development [38]. They tested a small-scale web application, the Electronic Database Management System (EDMS), and found that Re-CRUD provides more efficient and pure code automation than traditional frameworks like CakePHP, Laravel, Symphony, and FuelPHP[39].

3. Methodology

To fulfill the purpose of this study, a research methodology has been established and implemented. This methodology follows a systematic series of steps to ensure the production of credible findings. The following Figure 1 provides an illustration of the methodology.

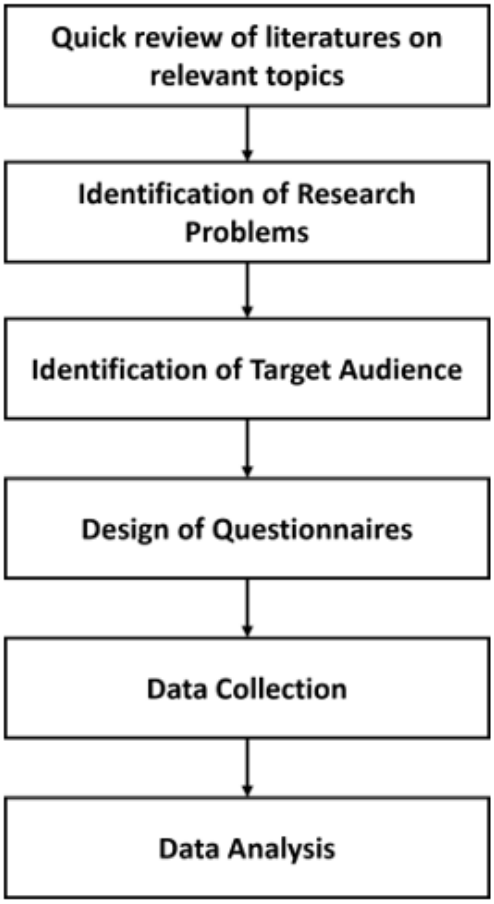

Figure 1. Research Methodology

The study initiates with an extensive examination of relevant literature, concentrating on key issues. This review provides a critical analysis of important subjects, such as the definition of Citizen Developer. It delves into the conceptual framework associated with this term, elucidating its traits and characteristics as outlined in academic discussions. Furthermore, the article explores the creation of Citizen Developers, providing a comprehensive analysis of past and present patterns that provide light on their development within the realm of software development. Moreover, this research examines the substantial importance of Citizen Developers within the realm of software development, placing emphasis on their responsibilities, contributions, and overall influence. Finally, the paper critically examines several methods and approaches related to Citizen Development activities, providing a clear explanation of their fundamental ideas and processes.

Secondly, within the scope of this study, a crucial research issue has been uncovered and subsequently elucidated. The research problem suggests that the shortage of Software En-

gineers in the industry might potentially be mitigated by incorporating Citizen Development approaches. In this approach, Citizen Developers would play a collaborative and supporting role inside development teams. Therefore, this study aims to assess the levels of preparedness and acceptance among Software Engineers with regards to this unique proposal.

3.1 Respondent

Moreover, the research is focused on individuals who describe themselves as Software Engineers, based on their professional work experience or formal academic degrees. The implementation of this specific criterion guarantees that the research encompasses the viewpoints and knowledge of individuals who possess a deep understanding of software development processes.

This survey serves as an initial investigation to support the hypothesis of our subsequent overall study that Citizen Developers should not work in isolation but rather in collaboration with software development experts, namely software engineers for effective and quality end-products. While we aimed to gather as many respondents as possible by distributing the survey randomly through various channels, we established a minimum acceptable sample size to ensure the study's progression within the broader research framework, which will be outlined briefly in the 'future works' section.

To determine the range of acceptable minimum number of respondents, we have benchmarked with few similar studies other than previous related works. Towards investigating how Software Engineers perceive AI-Assisted Code Reviews, Alami and Ernst (2025) has conducted a survey which collected a total of 20 respondents to support their later whole research [40]. A preliminary study from exploratory intersectional data collection among software users and developers conducted by Winchester et. al. (2022) has collected a total of 12 respondents to support their subsequent full research [41]. In fact, a related study to this research by Anuar et. al. (2023) has a total of 5 respondents to survey feedback from software engineering community on validating the proposal.

Hence, for the scope of this research at this point, we would tolerate minimum respondents between 5 to 20 participants, given that the demographic composition is considerably diverse in terms of gender, age, highest level of education, number of years of experience in software development, size of the organization, and industry in which they work.

3.2 Survey Questions

In addition, a set of questionnaires has been designed to collect pertinent data and valuable insights from the specified target demographic. This questionnaire comprises two unique components. The initial component of the study is dedicated to gathering Demographic Information, encompassing six questions that have been specifically formulated to capture fundamental demographic characteristics of the participants. The second component encompasses a set of Specific Questions, which consists of eleven inquiries designed to extract detailed ideas and impressions regarding the incorporation of Citizen Developers into development teams. The following Table 1 summarizes the Specific Questions that have been inquired in this questionnaire.

Furthermore, the data collection procedure is implemented using a carefully designed random sample mechanism, which guarantees the inclusion of individuals from the specified target population. This methodology enhances the collection of data that enhances the applicability of the results.

Finally, subsequent to the acquisition of data, an in-depth and systematic analysis is undertaken. This phase involves the interpretation and synthesis of the collected data, facilitating a comprehensive understanding of the attitudes, viewpoints, and readiness levels of Software Engineers toward the concept of Citizen Development. The outcomes of this analysis will be expounded upon in the ensuing sections, affording insights and implications that contribute to the advancement of knowledge in this domain.

4. Results and Discussions

Twenty people who self-identify as software developers took part in the survey after it was randomly distributed. Demographic data, such as gender, age, highest level of education, number of years of experience in software development, size of the organization, and industry in which they work, was gathered in order to further understand the background of the participants.

The demographic profile of the study participants reveals a gender distribution, with 75 percent being males and 25 percent females. Regarding age, the participants encompassed a range of 18 to 34 years, with the majority falling within the 25 to 34 years old demographic. In relation to the participants' educational background, it was found that 75 percent of them possessed a bachelor's degree as their highest level of educational attainment. Within the framework of software development, the participants demonstrated a spectrum of experience spanning from 1 to 10 years, with an average duration of approximately 3 years.

The participants in the study were affiliated with organizations of varying sizes, ranging from small-scale enterprises with 1-50 employees to large organizations with 501 or more employees. It is worth noting that there was a significant presence of small-sized organizations among the participants. Furthermore, the majority of participants were employed in the Technology or IT sector, accounting for a substantial proportion of 70 percent.

5. Questionnaire Findings and Analysis

This survey consisted of eleven (11) targeted inquiries pertaining to Citizen Development, aiming to acquire perspectives on this subject based on respondents' software development expertise. We embarked on this research endeavor to gain a deeper understanding of the readiness and acceptance of software engineers towards the concept of Citizen Development, a phenomenon that holds the promise of addressing the pressing shortage of software engineers by integrating non-developer personnel into the development process.

Table 1. List of Specific Questions

Question ID	Specific Question	Insights provided
SQ1	At this point, how familiar are you with the concept of 'Citizen Development'?	Familiarity with Citizen Development
SQ2	Have you ever worked with or collaborated with non-programmers (Citizen Developers) during software development process?	Experience with Non-Programmers in Software Development
SQ3	In your current organization, are there any active Citizen Development initiative?	Presence of Active Citizen Development Initiatives
SQ4	In your opinion, to what extent Citizen Developers should be involved in software development?	Extent of Involvement of Citizen Developers
SQ5	In your opinion, how helpful do you think it is for Citizen Developers to assist software developers at any stage of the software development process?	Perceived Helpfulness of Citizen Developers
SQ6	In your opinion, what types of applications do you think are suitable for Citizen Developers to be involved in?	Types of Applications Suitable for Citizen Developers
SQ7	Do you believe that involving Citizen Developers in the software development team can free up software developers' time to focus on more complex projects and tasks?	Impact on Software Developers' Time
SQ8	In your opinion, how would you rate the effectiveness of having Citizen Developers assist software developers in the software development process?	Effectiveness of Citizen Developers' Assistance
SQ9	In your opinion, what would be the ideal combination between software developers and citizen developers in a software development team?	Ideal Combination in a Software Development Team
SQ10	In your opinion, should Citizen Development have a model, framework or guideline to effectively develop or assist in developing software?	Need for a Model or Framework
SQ11	Overall, do you think that the concept of Citizen Development would promote manpower optimisation in an organisation or company?	Promotion of Manpower Optimization

In this section, we delve into the insights and opinions shared by the professionals who generously participated in this survey. Their valuable feedback provides critical insights into the perception of Citizen Development within the software development community and paves the way for a meaningful discussion on its potential implications and challenges. Figure 2 summarizes the questionnaire findings, whereby the detailed analyses are described as follows.

5.1 Findings

The following subsections discusses the detailed findings based on each of the specific questions.

SQ1: Familiarity with Citizen Development

Seventy (70) percent of the respondents are aware of the concept of citizen development. The results suggest that the software developers may have a considerable level of familiarity with the idea being studied or at least we can conclude that software development by Citizen Developers a familiar concept and practice.

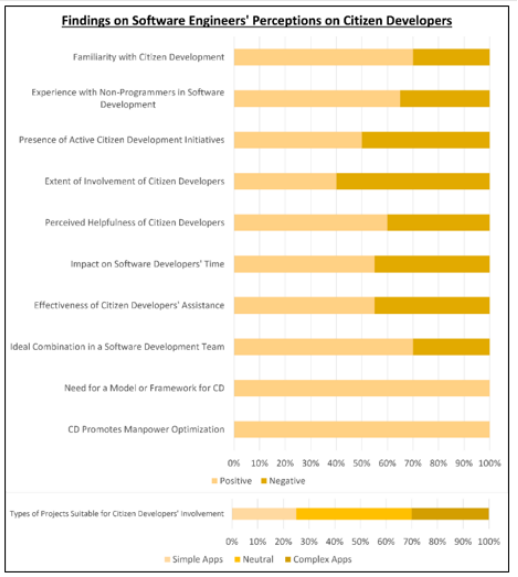

Figure 2. Questionnaire Findings Summary

SQ2: Experience with Non-Programmers in Software Development

A significant proportion of the participants, specifically sixty-five (65) percent, reported having prior involvement in working or collaborating with individuals who lack programming expertise, commonly referred to as Citizen Developers, within the context of software development. This discovery underscores a prominent pattern in which software development teams are progressively integrating individuals from varied backgrounds into their projects, potentially indicating a rising recognition of Citizen Developers within the industry.

SQ3: Presence of Active Citizen Development Initiatives

Fifty (50) percent of the participants indicated the presence of active Citizen Development initiatives in their respective organizations. The findings suggest that there is a moderate level of implementation of Citizen Development practices in the surveyed workplaces. This indicates that a significant number of organizations are actively exploring and adopting this approach.

SQ4: Extent of Involvement of Citizen Developers

On the inquiry related the involvement extent of Citizen Developers in software development, forty (40) of the respondents believe that Citizen Developers should be maximally involved to assist the software development team. This perspective implies a significant level of confidence to integrate Citizen Developers into core software development activities.

SQ5: Perceived Helpfulness of Citizen Developers

Sixty (60) percent, of the participants believe that the involvement of Citizen Developers in the software development process is beneficial, irrespective of the stage at which they provide assistance to professional software developers. This suggests the perceived significance of Citizen Developers in the context of software development projects, with the potential to improve efficiency and productivity.

SQ6: Types of Projects Suitable for Citizen Developers

There are differences in opinion among respondents concerning the types of applications that are considered suitable for Citizen Developers. Twenty-five percent (25) of respondents agreed that Citizen Development is suitable for creating simple applications with basic functionalities. Meanwhile, a slightly higher percentage of respondents of thirty (30) percent, believe that Citizen Developers can contribute to the development of complex applications. Despite the slight differences, however it can be concluded that a considerably significant portion of software developers agreed that Citizen Developers are ready for any scale of projects.

SQ7: Impact on Software Developers' Time

Fifty-five (55) percent of the respondents agreed that the inclusion of Citizen Developers to a software development team can significantly reduce the workload of software developers which allow them to focus on more complex tasks. Thus, this finding supports the notion that Citizen Developers can significantly improve the distribution of resources among software development teams that could result to time effectiveness.

SQ8: Effectiveness of Citizen Developers' Assistance

Fifty-five (55) percent of the participants believe that the involvement of Citizen Developers in the software development process is potentially beneficial and effective when working alongside professional software developers. This finding suggests that a significant proportion of the surveyed population believes that the collaboration between software developers and Citizen Developers leads to favorable outcomes.

SQ9: Ideal Combination in a Software Development Team

A significant majority of participants of seventy (70) percent believe that the optimal composition of a software development team should be majority number of software developers versus a minority of Citizen Developers. This perspective suggests an approval for a mixed-team configuration that capitalizes on the proficiencies of both professional developers and Citizen Developers.

SQ10: Need for a Model or Framework

According to the survey participants, a consensus of a hundred (100) percent was reached, indicating unanimous agreement that Citizen Development necessitates the presence of a well-defined model, framework, or guideline in order to facilitate the efficient development or support of software. The aforementioned consensus underscores the perceived significance of implementing well-defined guidelines to facilitate the seamless integration of Citizen Developers within software development processes.

SQ11: Promotion of Manpower Optimization

A consensus is evident, as all respondents unanimously agree that the implementation of Citizen Development would enhance workforce optimization within an organization or company. The aforementioned consensus highlights the notion that Citizen Development has the capacity to improve workforce efficiency and resource allocation in organizational settings.

5.2 Analysis (Statistical Significance)

In order to verify and analyse the data obtained through the survey responses, a statistical significance analysis has been conducted where the t-test method was applied. The following Table 2 summarises the results for the statistical significance analysis through the T-Test Method.

The analysis of software engineers' perceptions toward Citizen Developers resulted in the following statistical outcomes: a mean positive perception of 0.665 (66.5%) and a mean negative perception of 0.335 (33.5%). Additionally, the one-tailed p-value is 0.0138, and the two-tailed p-value

Table 2. T-Test Analysis Results

Analysis Parameters / Perceived	Positive	Negative
Mean	0.665	0.335
Variance	0.039472	0.039472
P(T<=t) one-tail	0.013763407	
t Critical one-tail	1.833112933	
P(T<=t) two-tail	0.027526815	
t Critical two-tail	2.262157163	

is 0.0275, indicating a statistically significant difference between positive and negative perceptions. The mean positive perception (66.5%) is nearly twice as high as the mean negative perception (33.5%). This suggests that software engineers generally hold a favorable view of Citizen Developers.

However, the remaining skepticism (33.5%) highlights key challenges that could be addressed through well-defined models, governance mechanisms, and enhanced collaboration strategies.

6. Proposed Solution

Cooperation between professional developers and end users or domain experts is vital to address significant issues in software requirement fulfilment. The end users or domain experts, who can be empowered as Citizen Developers, could be a strategic game-changer to the project team. Allowing them to participate in the development of the software products directly and actively, hence require a conducive and effective collaboration environment. The idea of effective collaboration environment can be enabled through the development processes, which often understood in the form of software development model (SDM) or also known as software development lifecycle (SDLC).

In conjunction to the issues discussed, a software development model was proposed with the aim to provide a structured process flow for collaborative development of software products. Most importantly, the model shall address the concerns manage the complexities of software requirements by bridging both technical and domain expertise.

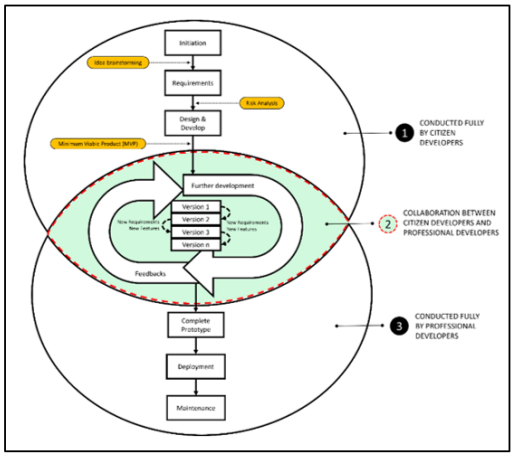

Figure 3. Proposed Model

The proposed software development model is constructed on the foundational steps of the software development process as outlined in the SWEBOK Guide V4.0 [42] and addresses the gaps identified in Microsoft's Agile v2 Model for Citizen Development [43] and PMI's Hyperagile Model [6]. Standard SDMs generally consist of six sequential stages: initiation, requirements, design, coding, testing, deployment, and maintenance [44]. These six stages form the structural framework of the proposed model. Additionally, since the central focus of Microsoft Agile v2 and PMI Hyperagile is on prototyping methodology, the proposed model incorporates the Prototype SDM. The prototype model emphasizes the quick identification of primary user requirements by developers and users, enabling the rapid creation of a software prototype. This prototype is then refined iteratively based on user evaluations and feedback until it meets user satisfaction [45].

The proposed model is divided into three main components: citizen development, collaboration, and conventional development. Within the citizen development component, the requirements phase involves citizen developers initiating the process by defining initial ideas and gathering high-level requirements for analysis. These requirements are categorized and prioritized to assess their complexity, facilitating the definition of responsibilities. Subsequently, a risk analysis is conducted to evaluate the feasibility of proceeding with the software development model. The design phase within the citizen development component focuses on creating and developing the minimum viable product (MVP).

Following the design phase, the development phase begins, involving software development professionals, referred to as 'conventional developers.' These experts address any technical challenges beyond the capability of citizen developers and ensure that all defined requirements are met. Conventional developers work closely with the citizen developer team to provide technical support and guidance throughout the development process. This iterative and collaborative effort between citizen developers and conventional developers facilitates a smooth transition from design to development, culminating in the delivery of a robust application.

7. Future Works

Future works for this study is firstly, to conduct further research into specific concerns of software engineers towards citizen developers that may help refine best practices for integrating Citizen Developers into the software development ecosystem. Ultimately, it is vital to explore the use cases and processes required to implement the proposed software development model. Most importantly, the next step is to validate the practicality of the model by conducting thorough testing and evaluation in diverse variables. Hence, upon the testing and evaluation of the purpose model, definitely corrective actions and enhancements may be required.

8. Conclusion

In conclusion, the analysis of the conducted survey highlights a notable level of optimism among Software Engineers re-

garding the inclusion of Citizen Developers in the software development process, emphasizing the potential for a collaborative approach to software development. Based on these findings and addressing the identified gaps, a proposed software development model has been developed. This model integrates the strengths of both Citizen Developers and conventional developers, providing a structured and inclusive framework for collaborative software creation. While this study has limitations that may influence the precision of its conclusions, the insights gained are instrumental in understanding the perceptions of Software Engineers and the potential of Citizen Development. The proposed model offers a practical solution to workforce challenges and serves as a blueprint for fostering innovation and efficiency in software development methodologies. Future works will focus on detailing the use cases and processes required for implementing the proposed model and conducting rigorous validation to ensure its practicality and adaptability in real-world scenarios. This ongoing research aims to contribute significantly to advancing software development practices, addressing workforce deficiencies, and shaping the future trajectory of the industry.

Acknowledgment

This work was supported/funded by the Ministry of Higher Education of Malaysia under Fundamental Research Grant Scheme (FRGS/1/2020/ICT01/UTM/02/1) (Vote No. R.K130000.7856.5F415).

References

[1] I. H. Azmy, A. Azmi, N. Kama, H. Mohd Rusli, and A. W. Anuar, "Towards a semantic-based searching in a web application framework for an improved development productivity and usability," *Journal of Web Semantics*, vol. 7, no. 28, pp. 12–23, 2022.

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

[3] C. Silva, J. Vieira, J. C. Campos, R. Couto, and A. N. Ribeiro, "Development and validation of a descriptive cognitive model for predicting usability issues in a low-code development platform," *Human factors*, vol. 63, no. 6, pp. 1012–1032, 2021.

[4] R. Waszkowski, "Low-code platform for automating business processes in manufacturing," *IFAC-PapersOnLine*, vol. 52, no. 10, pp. 376–381, 2019.

[5] M. Woo, "The rise of no/low code software development—no experience needed?" *Engineering (Beijing, China)*, vol. 6, no. 9, p. 960, 2020.

[6] P. M. Institute, "Citizen development: The handbook for creators and change makers." Project Management Institute, 2021.

[7] N. Carroll and M. Maher, "How shell fueled digital transformation by establishing diy software development." *MIS Q. Executive*, vol. 22, no. 2, p. 3, 2023.

[8] K. Ahmed *et al.*, "How can it empower citizen developers?" 2017.

[9] D. Hoogsteen and H. Borgman, "Empower the workforce, empower the company? citizen development adoption," 2022.

[10] D. Thacker, V. Berardi, V. Kaur, and G. Blundell, "Business students as citizen developers: Assessing technological self-conception and readiness." *Information Systems Education Journal*, vol. 19, no. 5, pp. 15–30, 2021.

[11] J. Traxler and H. Crompton, *Critical mobile pedagogy: Cases of digital technologies and learners at the margins*. Routledge, 2020.

[12] A. C. Bock and U. Frank, "Low-code platform," *Business & Information Systems Engineering*, vol. 63, pp. 733–740, 2021.

[13] R. U. B. Salgueiro, "The impact of microsoft power platform in streamlining end-to-end business solutions: Internship report at microsoft portugal, specialist team unit," Master's thesis, Universidade NOVA de Lisboa (Portugal), 2020.

[14] L. Tang, "Erp low-code cloud development," in *2022 IEEE 13th International Conference on Software Engineering and Service Science (ICSESS)*. IEEE, 2022, pp. 319–323.

[15] N. Prinz, C. Rentrop, and M. Huber, "Low-code development platforms-a literature review." in *AMCIS*, 2021.

[16] M. K. Fojcik and M. Fojcik, "Teachers experience with introducing programming in different courses for non-computer science students."

[17] B. Adrian, S. Hinrichsen, and A. Nikolenko, "App development via low-code programming as part of modern industrial engineering education," in *Advances in Human Factors and Systems Interaction: Proceedings of the AHFE 2020 Virtual Conference on Human Factors and Systems Interaction, July 16-20, 2020, USA*. Springer, 2020, pp. 45–51.

[18] E. Martinez and L. Pfister, "Benefits and limitations of using low-code development to support digitalization in the construction industry," *Automation in Construction*, vol. 152, p. 104909, 2023.

[19] J. Wong, M. Driver, and S. Ray, "The future of apps must include citizen development," *Gartner, Stamford*, 2019.

[20] A. Namoun, A. Owrak, and N. Mehandjiev, "Non-programmers composing software services: a confirmatory study of the mental models and design challenges," *Applied Sciences*, vol. 9, no. 24, p. 5558, 2019.

[21] M. Oltrogge, E. Derr, C. Stransky, Y. Acar, S. Fahl, C. Rossow, G. Pellegrino, S. Bugiel, and M. Backes, "The rise of the citizen developer: Assessing the security impact of online app generators," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 634–647.

[22] M. Tisi, J.-M. Mottu, D. S. Kolovos, J. De Lara, E. M. Guerra, D. Di Ruscio, A. Pierantonio, and M. Wimmer, "Lowcomote: Training the next generation of experts in scalable low-code engineering platforms," in *STAF 2019 Co-Located Events Joint Proceedings: 1st Junior Researcher Community Event, 2nd International Workshop on Model-Driven Engineering for Design-Runtime Interaction in Complex Systems, and 1st Research Project Showcase Workshop co-located with Software Technologies: Applications and Foundations (STAF 2019)*, 2019.

[23] R. Alt, J. M. Leimeister, T. Priemuth, S. Sachse, N. Urbach, and N. Wunderlich, "Software-defined business: Implications for it management," *Business & Information Systems Engineering*, vol. 62, pp. 609–621, 2020.

[24] M. Lebens, R. J. Finnegan, S. C. Sorsen, and J. Shah, "Rise of the citizen developer," *Muma Business Review*, vol. 5, no. 12, pp. 101–111, 2021.

[25] P. Vincent, K. Iijima, M. Driver, J. Wong, and Y. Natis, "Magic quadrant for enterprise low-code application platforms," *Gartner report*, vol. 120, 2019.

[26] J. McKendrick, "The rise of the empowered citizen developer," *New Providence, NJ: Unisphere Research*, 2017.
[27] R. Arora, N. Ghosh, and T. Mondal, "Sagitec software studio (s3)-a low code application development platform," in *2020 International Conference on Industry 4.0 Technology (I4Tech)*. IEEE, 2020, pp. 13–17.
[28] E. Sahinaslan, O. Sahinaslan, and M. Sabancıoglu, "Low-code application platform in meeting increasing software demands quickly: Setxrm," in *AIP Conference Proceedings*, vol. 2334, no. 1. AIP Publishing, 2021.
[29] F. Gürcan and G. Taentzer, "Using microsoft powerapps, mendix and outsystems in two development scenarios: an experience report," in *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. IEEE, 2021, pp. 67–72.
[30] D. Di Ruscio, D. Kolovos, J. de Lara, A. Pierantonio, M. Tisi, and M. Wimmer, "Low-code development and model-driven engineering: Two sides of the same coin?" *Software and Systems Modeling*, vol. 21, no. 2, pp. 437–446, 2022.
[31] S. S. Bhattacharyya and S. Kumar, "Study of deployment of "low code no code" applications toward improving digitization of supply chain management," *Journal of Science and Technology Policy Management*, vol. 14, no. 2, pp. 271–287, 2023.
[32] Z. Yan, "The impacts of low/no-code development on digital transformation and software development," *arXiv preprint arXiv:2112.14073*, 2021.
[33] E. Benson, A. X. Zhang, and D. R. Karger, "Spreadsheet driven web applications," in *Proceedings of the 27th annual ACM symposium on User interface software and technology*, 2014, pp. 97–106.
[34] K. S.-P. Chang and B. A. Myers, "Gneiss: spreadsheet programming using structured web service data," *Journal of Visual Languages & Computing*, vol. 39, pp. 41–50, 2017.
[35] C. Yang, S. Pan, R. Li, Y. Liu, and L. Peng, "A coding-free software framework for developing lightweight web data management systems," *Applied Sciences*, vol. 10, no. 3, p. 865, 2020.
[36] H. Benouda, M. Azizi, M. Moussaoui, and R. Esbai, "Automatic code generation within mda approach for cross-platform mobiles apps," in *2017 First International Conference on Embedded & Distributed Systems (EDiS)*. IEEE, 2017, pp. 1–5.
[37] M. Inayatullah, F. Azam, and M. W. Anwar, "Model-based scaffolding code generation for cross-platform applications," in *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE, 2019, pp. 1006–1012.
[38] A. W. Anuar, N. Kama, A. Azmi, and H. Mohd Rusli, "Revisiting web application development with integrated records management important aspect using re-crud," *Journal of Information and Knowledge Management (JIKM)*, vol. 12, no. 1, pp. 31–53, 2022.
[39] A. W. Anuar, N. Kama, A. Azmi, H. M. Rusli, and Y. Yahya, "Re-crud code automation framework evaluation using desmet feature analysis," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 5, 2022.
[40] A. Alami and N. A. Ernst, "Human and machine: How software engineers perceive and engage with ai-assisted code reviews compared to their peers," *arXiv preprint arXiv:2501.02092*, 2025.
[41] H. Winchester, A. E. Boyd, and B. Johnson, "An exploration of intersectionality in software development and use," in *Proceedings of the Third Workshop on Gender Equality, Diversity, and Inclusion in Software Engineering*, 2022, pp. 67–70.
[42] I. C. Society, "Guide to the software engineering body of knowledge version 4.0 (swebok guide v4.0)," 2023.
[43] Microsoft, "Differences between power apps and traditional app development approaches," 2022, https://learn.microsoft.com/en-us/power-apps/guidance/planning/app-development-approaches.
[44] D. Han, X. Li, J. Zhang, G. Wang, H. Deng, and L. Deng, "Research and application of software development model in computer software development," in *Proceedings of the 5th International Conference on Information Science and Systems*, 2022, pp. 7–12.
[45] J. Yu, "Research process on software development model," in *IOP Conference Series: Materials Science and Engineering*, vol. 394, no. 3. IOP Publishing, 2018, p. 032045.