QUBAHAN ACADEMIC JOURNAL
VOL. 6, NO. 1, January 2026
https://doi.org/10.48161/qaj.v6n1a1599

Certification of Computer Applications in Customer Satisfaction: An Analysis of the Developer's Role

Marcelino P. Braga [1]* and Miguel A. Brito [2]

[1] ALGORITMI Center, School of Engineering, University of Minho, Guimarães 4804-533, Portugal;
[2] Department of Information Systems, School of Engineering, University of Minho, Guimarães 4804-533, Portugal.

* **Corresponding author:** id11087@uminho.pt.

ABSTRACT: This study examines the certification of software applications and the role of programmers in customer satisfaction. It seeks to evaluate the relationship between these two factors. Twenty-two articles were examined in order to identify trends and conclusions regarding the connection between software application certification and customer satisfaction. The results demonstrate that programmers play an essential role in ensuring the integrity of systems and user security, and that their qualifications and training are vital for the effectiveness of certification and customer satisfaction. The article concludes that certifications are formal processes that foster customer confidence and that collaboration between software development companies and customers is vital to make sure the quality of software applications.

Keywords: software application, customer satisfaction, programmers, certifications, customer confidence.

I. INTRODUCTION

The relentless advancement of technology and the intensifying cybersecurity threats necessitate agile and adaptable responses from software application programmers. The rapid evolution of the digital environment, emphasized on advancements in cloud computing, artificial intelligence, and the Internet of things, compels programmers to continuously update their skills and knowledge to align with market demands. The complexity of security threats, such as attacks of ransomware, phishing, and social engineering, underscores the urgent need for robust and effective certification processes for software applications [1]. Certification of software applications is deemed essential for customer satisfaction as it ensures product quality, security, and reliability [2]. This process, which is achieved through rigorous evaluation, presupposes that the application complies with established standards of quality and security. These standards are fundamental aspects for both developers and users [1, 3]. The certification process is overseen by programmers of software applications, whose role is to create and maintain products that meet the necessary requirements for certification [4, 5]. This implies that development should be quality-oriented from the early stages, implementing integrated security measures and a commitment to continuous application updates to maintain certification and protect against vulnerabilities [6-8].

Certification serves as a trust element for customers, ensuring that the software application undergoes a detailed evaluation and complies with recognized standards of quality and security [9]. It also facilitates comparisons between different software programs and assists customers in making informed choices based on objective and accepted standards. In numerous situations, certification serves to authenticate the application that complies with specific code of practice, a crucial aspect for customers in critical sectors of community, such as government, finance, and healthcare where adherence to regulatory standards is

imperative. However, developers face significant challenges due to the dynamic nature of technological evolution and security threats [10]. It is therefore imperative that certification processes are agile and adaptable to keep pace with these changes. In the context of software application certification, customer satisfaction is a primary goal, as a quality application that meets customers' needs and expectations is essential to address them [11].

The integration of agile development practices with certification requirements presents a challenge, yet it is essential for continuous delivery of value to customers [3, 6, 10]. In this context, this review article aligns with the presented context and aims to conduct an initial survey of the main concerns of the scientific community in this area, namely, to conduct a rigorous review of existing information on software application certification. The motivation for this research lies in the potential to contribute to knowledge about software application certification in customer satisfaction and to understand the role of in this process. Furthermore, the research aims to provide information for software application development companies and professionals in the field, emphasizing the necessity for developers' training and qualification for certification success and customer satisfaction. The objective of this investigation is to present a synthesis of relevant studies on software application certification in customer satisfaction in the medium and long term and to understand the developer's influence in this process. To achieve the proposed objective in this study, the following tasks were outlined:

- Select the database for searching relevant studies in the context of software application certification.
- Present the process of consulting relevant studies using inclusion and exclusion criteria.
- Describe the selected studies for the research.
- Analyze the studies in focus.
- Present the final conclusions as well as references for future research.

1. THEORETICAL BACKGROUND

To strengthen the theoretical foundation of this study, a multidisciplinary framework is adopted that integrates key management and technology adoption models. Institutional theory discusses how external forces such as regulations, industry standards, and societal values influence the certification processes and explain why organizations seek certification for legitimacy.

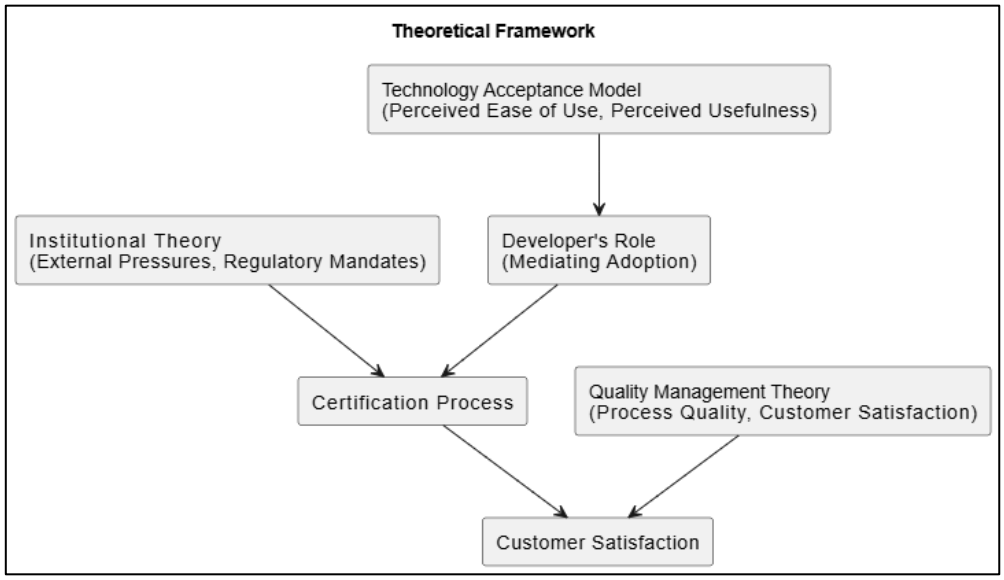

FIGURE 1. Theoretical framework.

The Technology Acceptance Model (TAM) is used to analyses how ease of use and usefulness, like other motivational factors, influence a developer's perception and adoption of certification technologies and practices. Quality Management Theory explain the outcomes that these adoption processes yield, which include product quality and customer satisfaction. All these theorists focus on different aspects but agree on the fact that systematic certification processes not only meet the required technical standards but also build the customers' trust and satisfaction. Together, these theoretical perspectives provide a robust framework to examine the developer's role in software certification and its subsequent impact on customer satisfaction (Figure 1).

2. CERTIFICATION CONCEPT

Certification is a formal process in which an independent entity attests that a service, product, system, or process fulfill specific requirements or established standards [12]. It is a measure carried out, as mentioned, by independent entities that apply standardized methodologies to ensure the objectivity and accuracy of the evaluation. This process involves an objective and impartial assessment to verify if the object in question complies with predefined criteria [2]. Study [8] in their article "Basic Concepts of Software Certification" explore the complexity and multiplicity of interpretations of the certification concept. These authors underline the variations in the understanding of this term, influenced by cultural differences, individual interests, and personal experiences. The presence of diverse opinions, both positive and negative, regarding certification is highlighted, particularly in the context of processes and products of software applications, where the use of this concept can be particularly complex. For instance, except for its association with quality management systems like ISO 9000 certification, terms related to certification were cautiously used in advertising and technical literature. This caution stemmed from suppliers' concerns regarding warranty and liability implications [2]. However, there is currently a growing interest in exploring certifications, certificates, and certification schemes for products and processes, within the scope of software applications [8].

The processes associated with certifying software applications continue to be transformed by new technologies, as well as the greater sophistication of system software. To rectify these issues, there is a movement to automate certification, create certification procedures for intelligent systems, and integrate the processes of certification into agile software development methodologies [12, 13]. The last three decades have seen remarkable transformation of artificial intelligence, which has led to heated discussions within almost all aspects of society. The document, "Trust in AI: Global Insights 2023" from [14] dives into the discourse revolving around Artificial Intelligence (AI) and addresses the matters of trust, ethics, and responsibility concerning the creating and usage of this tool. There is deepening apprehension toward AI technologies because of their biased, discriminatory, manipulative, or even human rights-violating employments. For the acceptance or implementation of AI, public trust is crucial. Moreover, AI ethics has been the subject of many discussions, from the need to make the development and deployment of AI algorithms transparent and accountable to the need to ensure fairness in their use. The debate on AI algorithmic bias also raises the important issue of reproducing discrimination and prejudice in automated systems. The algorithms used within AI technologies are frequently scrutinised, and their design and operational logic remain opaque. Calls for governance and legislative regulation of AI technologies have been equally urgent to create boundaries around the use of AI technologies to make sure they are deployed safely and responsibly.

While the literature offers valuable insights on software application certification and customer satisfaction, it largely remains descriptive rather than critically analytical. Many studies summarize certification practices without integrating key theories like Institutional Theory, the Technology Acceptance Model, and Quality Management Theory, and they often overlook the critical role of developers in mediating these processes. Moreover, there is a heavy reliance on secondary data, with few studies incorporating primary research to validate their findings. This study addresses these gaps by proposing an integrated theoretical framework that emphasizes the developer's mediating role and by employing a mixed-methods approach, including surveys and expert interviews to provide a more nuanced and empirically grounded understanding of how certification practices impact customer satisfaction.

II. METHOLOGY

This research is based on a study following the methodology suggested by Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA). In addition to the systematic review following the PRISMA guidelines, this study adopts a mixed-methods approach to enhance methodological rigor. While the systematic review provides a comprehensive analysis of secondary sources retrieved from the SCOPUS database, primary research has been incorporated to validate and enrich these findings.

1. RESEARCH DESIGN

This study is designed as a systematic review using PRISMA guidelines. After applying inclusion and exclusion criteria, 22 articles of high quality covering the subjects of computer applications, customer satisfaction, and developers' perspectives were selected. This approach integrates customer satisfaction literature, allowing for examination of the implications of certification practices on customer satisfaction. The review was aimed at understanding the implications of certification for customer satisfaction, combining theoretical literature with empirical evidence. These conclusions rest on systematic analysis of models, issues, and strategies, which interrelate with the objectives of the study.

2. SAMPLING STRATEGY

A systematic review was performed in the Scopus database for software certification and customer satisfaction alongside the role of a developer. The timeframe was set to the most recent 14 years. Based on the publication's reputation, only Q1 and Q2 journals were selected so that quality research is available. After performing a search, 309 articles were found and subsequently scanned through to check for reliable titles and abstracts, duplicates, and full texts. These articles were then filtered through inclusion and exclusion criteria, which provided a sample of 22 articles with the highest relevance to the research objectives. The quality of the articles ensured the credibility of the published literature.

3. ELIGIBILITY CRITERIA

This research commenced in November 2023, focusing on the certification of software applications in customer satisfaction and the assessment of the role played by developers in this context. The study utilized the Scopus database as the primary source for bibliographic data collection, employing a selection and filtering protocol based on carefully defined search strings. To construct the search strings, relevant keywords and themes were identified (see Table 1), resulting in the formulation of four distinct queries:

- Query 1: Yielded 36 documents, with a string centered on the intersection of software certification, developer role, ISO quality, and expanded to include customer satisfaction.
- Query 2: Accumulated 69 documents, aimed at studying customer satisfaction, developer role, ISO quality, and certification of software applications.
- Query 3: Resulted in 95 documents, emphasizing software application certification, developer role, ISO quality standards, and customer satisfaction.
- Query 4: Identified 100 scientific articles focused on the study of software application certification.
- Query 5: Provided a return of 309 scientific articles, using string interconnecting concepts such as software certification, software application development, ISO quality standards, and customer satisfaction.

A comparative analysis of the queries revealed that Queries 4 and 5 were the most significant. This indicated that the studies retrieved were more representative and had a broader scope, which was relevant to the research topic.

Table 1. Keywords with their representing strings.

Nº	Key words	String	Outcome
Search #1	Role	TITLE-ABS-KEY (software AND, certification) OR TITLE-ABS-KEY (software, AND developer, AND role) OR TITLE-ABS-KEY (iso, AND quality, AND software, AND product) AND TITLE-ABS-KEY (developer, AND software, AND customer, AND satisfaction, AND role))	36
Search #2	Software Developers Certification Quality Satisfaction Customer	(TITLE-ABS-KEY (developer, AND software, AND customer, AND satisfaction, AND role) OR TITLE- ABS-KEY (software, AND developer, AND role) OR TITLE-ABS-KEY (iso, AND quality, AND software, AND product) AND TITLE-ABS-KEY (software, AND certification, AND))	69
Search #3	Product	(TITLE-ABS-KEY (software AND, certification) AND TITLE-ABS-KEY (software, AND developer, AND role) OR TITLE-ABS-KEY (iso, AND quality, AND software, AND product) OR TITLE-ABS-KEY (developer, AND software, AND customer, AND satisfaction, AND role))	95
Search #4	Software Certification	TITLE-ABS-KEY (software AND certification) AND (LIMIT-TO (LANGUAGE, "English") OR LIMIT-TO (LANGUAGE, "Spanish") OR LIMIT-TO (LANGUAGE, "Portuguese")) AND (LIMIT-TO (EXACTKEYWORD, "Computer Software") OR LIMIT-TO (EXACTKEYWORD, "Certification") OR LIMIT-TO (EXACTKEYWORD, "Software Testing") OR LIMIT-TO (EXACTKEYWORD, "Software Engineering") OR LIMIT-TO (EXACTKEYWORD, "Safety Engineering") OR LIMIT-TO (EXACTKEYWORD, "Software Certification") OR LIMIT-TO (EXACTKEYWORD, "Certification Process") OR LIMIT-TO (EXACTKEYWORD, "Quality Assurance") OR LIMIT-TO (EXACTKEYWORD, "Testing")) AND (LIMIT-TO (SUBJAREA, "COMP")) AND (LIMIT-TO (DOCTYPE, "ar")) AND (LIMIT-TO (OA, "all"))	100
Search #5	Software Developer Certification Quality Satisfaction Customer	(TITLE-ABS-KEY (software AND certification) OR TITLE-ABS-KEY (software AND developer) OR TITLE-ABS-KEY (iso AND quality) AND TITLE-ABS-KEY (developer AND software AND customer AND satisfaction))	309

For the selection of relevant literature derived from Queries 4 and 5, a set of criteria was implemented, including (i) identification, (ii) selection, (iii) eligibility, and (iv) inclusion, allowing for the exclusion of studies not aligned with the research objectives. Documents that did not fit the defined publication types, showed data inconsistencies, were duplicates, or lacked a direct relationship with the research objectives were excluded. Additionally, articles of low methodological quality or reputation, unavailable, or unpublished in peer-reviewed journal of scientific nature in Spanish, Portuguese, or English were

disregarded. The inclusion criteria focused on scientific articles specifically addressing software application certification and its influence on customer satisfaction, as well as the role of the developer, with a preference for publications of higher scientific reputation, classified in the Q1 and Q2 quartiles of Scimago and published in the last 14 years.

4. ETHICAL CONSIDERATION

Since this study is based solely on a systematic review of published literature, no direct involvement of human participants was required, and ethical approval was not necessary. Nonetheless, rigorous ethical standards were maintained throughout the research process by ensuring all sources were appropriately cited and data were handled with integrity and transparency.

III. RESEARCH OUTCOMES

During the consultation phase of this research focused on the impact of certification of computer applications on customer satisfaction and the specific analysis of the role of programmers, a total of 309 articles were identified as being significantly relevant. This initial selection was made following the application of a research protocol on the Scopus database. We then proceeded with a careful evaluation of the 309 articles, conducting a detailed analysis of their titles, abstracts, and particularly study types. Special attention was given to empirical studies, given their relevance and ability to provide practical and verifiable data about the investigation. This process ensured that only works strictly aligned with the objectives of our investigation were considered for subsequent analysis. After this detailed analysis, 22 articles were considered eligible for inclusion in the research. The selection of these articles was based on their direct relevance to the study topic, namely the effect of certification of computer applications on customer satisfaction, analyzed from the perspective of the role played by programmers. These articles not only aligned precisely with our area of interest but were also published in high-reputation journals, specifically in quartiles Q1 and Q2 of Scimago, as evidenced by a high number of references and citations.

According to the reviewed studies, several key limitations were identified with varying frequencies. Generalizability issues were most prominent, noted in 13 studies. Within this category, 8 studies were limited by their reliance on single case studies, 4 cited small sample sizes, and 1 mentioned context restrictions that hinder broader applicability. Technical detail shortcomings were less frequent, reported in 3 studies 2 highlighted the need for more rigorous formal verification and 1 pointed out insufficient technical specifications. Methodological concerns were present in 4 studies, with 3 studies criticizing the lack of comparative evaluation and 1 study relying too heavily on self-reported data, which may introduce bias. Lastly, issues related to scalability and validation were mentioned in 2 studies, with one study noting limited external validation and another expressing concerns about scalability. This detailed breakdown underscores the diversity of limitations across literature, highlighting critical areas for future research improvement (Table 2). The table clearly breaks down the limitations by category, with a visual representation (using ASCII bar indicators) showing the frequency and count of each limitation, providing a comprehensive view of the issues identified across literature.

Table 2. Detailed frequency of key limitations across reviewed studies.

Category	Sub-category	Frequency	Count
Generalizability	Single case studies	########	8 studies
	Limited sample sizes	####	4 studies
	Context restrictions	#	1 study
	Total Generalizability		13 studies
Technical Details	Need for formal verification	##	2 studies
	Insufficient technical specs	#	1 study
	Total Technical Details		3 studies

	Lack of comparative evaluation	###	3 studies
Methodological	Over-reliance on self-reports	#	1 study
	Total Methodological		4 studies
	Limited external validation	#	1 study
Scalability/Validation	Scalability concerns	#	1 study
	Total Scalability/Validation		2 studies

1. DESCRIPTION OF SELECTED ARTICLES FOR ANALYSIS

The study by [10] presents a model for certifying software application products aimed at providing a more precise and reliable assessment of application quality. The model covers various areas of software application products, such as context description, user requirements, high-level design, detailed design, implementation, and testing. Certification is based on specific criteria for each product area, with varying levels of achievement. Despite being comprehensive and flexible, allowing for certification of software application products at all stages of the lifecycle, the model has gaps in the need for a higher effort in formal verification, especially in products derived from other software artifacts.

Study [15] discuss software quality certification in Master Data Management (MDM)-based applications. The research proposes certifying the functional adequacy of MDM applications based on ISO 8000 standards. The study presents significant results in identifying functional requirements based on ISO 8000 for master data applications. However, the research lacks a generic environment for evaluating and certifying MDM-based systems, in addition to the absence of specific standards addressing a set of software functional requirements, complicating the assessment of functional adequacy in master data applications. Study [16] address the importance of user participation in software application development projects. The research emphasizes the need to balance user participation levels to avoid conflicts and ensure satisfaction for both developers and users. The study highlights a significant discrepancy in definitions and perceptions of success concerning project between developers and industry of software, represented by customers or users. Contradictory interests between developers and users can adversely impact the performance of software development. The research has limitations, such as being conducted in a single environment, limiting result generalization, and subjectivity in satisfaction perceptions.

Study [17] conducted a global perspective analysis of IoT device certification and proposed a model to establish a security certification scheme for these devices compatible with the EU Cybersecurity Law. The proposed model met the specific qualification criteria of the ENISA system for a potential IoT device certification scheme. However, the model lacks specific technical details on the selection of evaluation methods, relying on a general approach and needing adaptation to different IoT device profiles. It also does not consider certifications from non-EU bodies that may be applicable to devices marketed in the EU, in addition to the evaluation of potential IoT device security certifiers needing to consider contextual factors. Study [18] introduce a method for improving the quality in the development of vital security software for the airline industry by applying agile methodologies and cost and certification effort optimisation during the software development lifecycle. The research employs an industrial case to demonstrate the effectiveness and practicality of the suggested method. Among the limitations of the work are the gaps for performing validation of the framework by certification bodies, the lack of studies to measure the impact of the proposed solution in other domains, and the shortcomings related to the security of IoT devices that AutoCert currently ignores.

Study [19] developed a certification scheme for IoT devices using public key infrastructure and remote attestation that solves the Time-Of-Check to Time-Of-Use (TOCTOU) issue by verifying device authentication and application state integrity. Authentication and application state verification can be completed in about 8 seconds. This means that the solution addresses integrity and verification problems from IoT devices in a secure and feasible manner. The research has limitations, as with almost all research works; system security is vulnerable to Denial of Service (DoS) attacks at both application and network layers, which affects system availability and responsiveness. Study [20] address the management of

vulnerabilities in open-source components distributed by vendors and the need for periodic recertification by medium and small enterprises. The research presents an application of a solution that assesses the differences between versions of an application to provide evidence to the analyst to evaluate the level of risk and determine if recertification is necessary. The results showed that using automated rules can reduce analysis time without affecting assessment accuracy. One limitation is that delta-certification can only detect known and registered vulnerabilities, in addition to the difficulty of applying the solution to larger and more complex projects, as the level of detail of evidence required may impact application stability and performance. The work does not address how vulnerabilities can be resolved, only how they can be detected.

Study [21] present an analysis of the relationship between software engineering issues and customer learning and satisfaction in software Capstone projects carried out by engineering students. The study was conducted through a survey with 127 students and their industry clients. It was found that students in management roles achieved higher learning outcomes than those working as developers, although this varied considerably among teams and their members. Common issues encountered were related to testing, technology skills and efforts estimation and task management. Despite the significant results presented, the study has limitations, such as the use of student self-assessment, which may introduce bias and inaccuracy, a sample confined to a specific course that restricts the generalization of results, lack of objective data to validate conclusions, omission of external factors that may influence learning and customer satisfaction, absence of a control group limiting causal attribution, and analysis conducted at a single time point, preventing assessment of learning and satisfaction evolution over time.

In the article "Investigation of temporal dissociation and focused immersion as moderators of satisfaction continuance intention relationship: Smartphone as an example," the relationship between customer satisfaction and continuance intention is investigated using smartphones as an example. The authors explore how temporal dissociation and focused immersion moderate this relationship, providing valuable insights into customer loyalty towards smartphone applications. The results highlight understanding user behavior towards smartphone applications and the influence of factors such as temporal dissociation and focused immersion on the relationship between satisfaction and continuance intention [22]. The research faces limitations such as a small and homogeneous sample, predominance of young participants, convenience sampling, and limited focus on smartphone usage, constraining the generalization of results.

The study by [23] highlights the need for privacy issues in software engineering and its accompanying findings from an empirical study performed in Brazil. The research underscores the increasing worry for the privacy of users and how Software Development Life Cycle (SDLC) methodologies frequently ignore non-functional aspects such as privacy. The study indicates that many computer programmers do not have adequate skills to construct software programs that can enable users to exercise privacy. There is a gap between announcing a privacy regulation and putting it into force where it is believed that concern for privacy grows, which poses a challenge for many organisations to try and comply with the new laws. The authors carried out a survey with Brazilian software developers, considering personal, behavioural, and situational variables that determine how the development team perceives privacy. Nine personal factors, five behavioural factors, and seven external environmental factors were found to be significant in the privacy control decisions of software developers.

The study "Towards a Theory of Software Developer Job Satisfaction and Perceived Productivity" by [24] outlines an empirical study that attempts to analyse the determinants of well-being and productivity of a software developer. The authors studied a large software firm, did an ethnographic study of the workplace, and spoke to some of the programmers. Therefore, they created a theory that focuses on the relations between social and technical factors, productivity, and satisfaction of a developer. The research gives additional information on programmers' satisfaction and productivity issues in software firms and how to make their developers more satisfied and productive. The authors of the theory claim that there is an interdependent relationship between the satisfaction of software developers and their perceived productivity in the presence of a relevant work environment, social context, and technical competencies. The research presents significant limitations related to its implementation being restricted to a single organization, compromising the generalizability of the results to broader contexts. The methodology adopted, focused solely on questionnaire

surveys, raises concerns about potential sample bias. Additionally, data collection relied on developers' self-reports, rising issues of subjectivity and potential data distortions. The correlational nature of the factor analysis precludes direct causal inference. Furthermore, the operational definitions of the factors and variables studied may reflect a limited perspective, based on the authors' assumptions and existing theoretical framework, neglecting the inclusion of potentially relevant variables regarding gaps in research on understanding causal relationships between perceived productivity, and satisfaction of job along with contextual variations affecting these relationships. There is also a debate among researchers about the validity of job satisfaction and perceived productivity as performance evaluation metrics for developers. Additionally, there is a lack of consensus on which performance indicators are most suitable for assessing developers' productivity.

The research "Recommender System for Configuration Management Process of Entrepreneurial Software Designing Firms," by [11], examines a recommendation system for configuration management processes in entrepreneurial software development firms. The system aims to assist developers and clients in making better decisions when choosing the type of software to be developed, selecting resources, and estimating costs. The proposed system is based on data from previously completed projects and prioritizes resources in a way to increase productivity and product quality while reducing effort, complexity, and failures. The research demonstrates that the proposed system is effective and enhances customer satisfaction. Limitations of this research include the lack of evaluation of other recommendation systems, the absence of scalability evaluation of the proposed recommendation system, and the failure to consider non-measurable variables in its assessment, such as technical expertise and knowledge of software developers. Additionally, the selection of personalized resources is influenced by many uncontrollable factors, such as changes in client requirements, and the proposed recommendation system does not account for all these factors. The authors also acknowledge that more research is required to measure the effectiveness and efficiency of the recommendation system in additional software development projects.

The study "Customer reviews as the measure the quality of software," [25], examines users' perceptions of open-source software quality and determines the effectiveness of customer reviews in relation to popularity and quality characteristics, such as programming languages and problem domains. The results indicate that reviews and user analyses of popular open-source projects exhibit very positive feelings of joy, anticipation, and trust as more frequent compared to other feelings like disgust, fear, and surprise. Furthermore, it was revealed that the effectiveness of reviews does not correlate with the number of downloads, suggesting that only a small number of users review the software. The authors conclude that customer reviews are an effective measure of software quality in open-source projects. Limitations of the study include its focus on software projects hosted on SourceForge, exclusion of other sources of open-source software, analysis limitation to popular projects, neglecting lesser-known ones, and the use of a single sentiment analysis tool specific to software without exploring other available options.

The article titled "Mutual development in mass collaboration: Identifying interaction patterns in customer-initiated software product development" investigates the phenomenon of customer-initiated software development in a business entity dedicated to creating project management tools for the oil and gas sector. This research sought to explain the areas of interaction and overlap between End-User Development (EUD) and professional software development. The research focuses on the role of customers in the development process using concepts from co-configuration theory, meta-design, and modding. Practical illustrations are given of the inclusion of customer input in five subprocesses of integration: adaptation, generalization, improvement requests, specialization, and personalization. The findings support that co-configuration, meta-programming, and modding emerge as key processes of collaboration between professional and end-user developers. However, this is limited by using one case study, which makes the findings less generalizable.

The paper by [26] entitled "Mutual development: The software engineering context of end-user development" describes a case study concerning the evolution of software products initiated by clients, which was conducted over two years in a commercial software firm that specializes in developing project management applications for the oil and gas industry. Throughout this period, the researchers not only

observed but actively participated in the company's development activities, with a particular interest in understanding how clients and professional developers engage in a collaborative development process facilitated by the shared use of software tools, including products and support systems. To collect data, the researchers conducted interviews with both developers and clients to identify the contributions of clients to product development. From these interviews emerged five fundamental sub-processes - adaptation, generalization, requests for improvement, specialization, and personalization - which serve to describe and compare the different stages of development that unfolded during the interaction between developers and clients. By employing an empirical approach based on direct observation and active participation in the company's development activities, the researchers were able to analyze the collected data and subsequently propose a theoretical model of mutual development. The identified results consist of the client's contribution as solution designers and generators of new ideas. The main limitation of this research includes the specific focus on a single case study rather than a broader sample, which may limit the generalizability of the results.

The study "Handshaking with Implementation Proposals: Negotiating Requirements Understanding," by [27], addresses the use of a negotiation process called handshaking with implementation proposals to effectively communicate requirements, even in situations with limited written requirements and distance between clients and developers. This handshaking process helps create value, establish a stable foundation for the project, and improve communication between product management and development organizations. Among the key results, there is an improvement in requirements understanding, enabling the identification of critical issues and the correction of inconsistent decisions. There was an increase in the quality of information for decision-making, aiding in precise planning and reducing the need for change management. Practitioners assessed that the improved understanding of requirements resulted in a reduction of about 40% in costs related to defects. One of the limitations identified by the authors is that not all product managers achieved the desired results in the handshaking process. Some development teams refused to create implementation proposals, stating that the provided requirements were not valid. The need for further empirical research in different contexts and cultures to validate and enhance the handshaking process with implementation proposals is one of the main gaps in the research.

The study by [28], explores the factors influencing customers' purchase intention in e-commerce recommendation systems. The study presents a model that analyzes the importance of elements such as novelty, diversity, accuracy, recommendation quality, satisfaction, trust, usefulness, ease of use, and familiarity in consumers' purchase decisions. The research was performed among users who had experience with online shopping at the Lazada.com website, along with other relevant data that was gathered to assess how these elements affected consumers' purchase intention. The questionnaire that was designed for this research comprised sections dealing with demographic information, online shopping experience, use of recommendation systems, and factors affecting customers' purchase intention. Using a five-point Likert scale, the participants rated their e-commerce recommendation systems' novelty, diversity, ease of use, usefulness, satisfaction, quality of recommendations, customer trust, and familiarity. The results of this study showed that the recommendation systems' purchase intention was affected by the accuracy, diversity, ease of use, satisfaction, trust, and even recommendation quality.

The research titled "Introducing knowledge redundancy practice in software development: Experiences with job rotation in support work" [29] investigated the impact of employee rotation on knowledge sharing and collaboration in IT application support teams. The study sought to understand the pros and cons of employee rotation practices in the Norwegian firm MapIT, focusing on how rotation, in turn, aids in knowledge redundancy at the team and organizational levels. It describes in detail the different stages and outcomes of the study, including a theoretical part that discusses the use of rotation in software companies and the role of knowledge redundancy in software engineering. The findings indicate that while employee rotation has the potential to improve knowledge sharing and collaboration among application development teams, careful consideration must be given to how the rotation is structured to mitigate efficiency loss and redundancy. The main limitation of this research is that it was done within one organization, which reduces the possibility of extrapolating the findings to different organizational settings.

The fuzzy-AHP-based technique for design feature selection in Massively Multiplayer SQL server games: A case study" by [30] tackles a fuzzy-AHP (Analytic Hierarchy Process)-based technique to aid MMORPG (Massively Multiplayer Online Role-Playing Game) designers in the decision-making process regarding design feature selection. The study suggests a hierarchical system framework consisting of components such as a character system, game environment, user interfaces, virtual item stores, and sound/light effects. In this research, design features were analyzed based on users' importance ratings using a hybrid fuzzy logic and analytic hierarchy process methodology. The outcomes of the study presented a summary of the importance of design features in online gaming and empowered developers to make tactical decisions that counterbalance the design features during the development stages of the products. Limitations include the absence of the ability to generalize the results of the study to other countries or cultures, as the study was only limited to participants from Taiwan.

Study [31] conducted a study titled "Hybrid model for evaluation of quality aware DevOps" on a hybrid model for assessing quality in DevOps. The approach involves developing software applications that promote collaboration between development and operations teams to deliver applications quickly and reliably. The study discusses the research findings compared to existing approaches and concludes that the TDMBD can significantly enhance the quality of delivered software. As a limitation of the research, the authors argue that the approach still lacks a quantifiable perspective, as there is no defined metric to measure performance through software attributes. Other limitations such as lack of external validation, generalization of results, and concerns regarding data privacy and security in the approach can also be noted.

The article "A Comprehensive MCDM-Based Approach for Object-Orientated Metrics Selection Problems" deals with the problem of selection of metrics in Object-Orientated Programming (OOP) with the aim of measuring design and code quality deficiencies. The researchers tried to address ways to identify defects as early as possible in the life cycle of software development to maintain high-quality software and reduce costs while improving customer satisfaction. In this study, the authors aimed to prune the metric combinations to make the rules for defect identification relevant and precise using a multicriteria decision-making approach based on DEMATEL. The application of fuzzy DEMATEL techniques in this research was shown to improve the accuracy and effectiveness of the defect detection rules for design defects in object-orientated programs. Some of these research limitations include the inability to generalize results over different domains of software, having a small sample size for external validity, difficulty in the selection of metrics for defect detection, assuming that fuzzy DEMATEL techniques will be executed correctly, and the expectation that external validation will affirm the robustness of the results. Moreover, the validation of the study was done exclusively based on a single previous study, and the approach taken does not incorporate the agile aspect of coding within information systems, which are perpetually updated and modified [32].

In the research titled "Cloud services composition through cloud patterns: a semantic-based approach," [33] develops a method for composing cloud services to satisfy client needs from a semantic perspective. The goal of this research was to develop a methodology for an automated cloud application construction process backed with cloud patterns and Semantic Web technologies such as SPARQL, OWL, and OWL-S. The methodology intended to solve the known mismatches problem between service interfaces, operations, and parameters within the automation of cloud service composition. This work captured the service composition problem and its challenges, noting the necessity of semantic techniques in the composition process to achieve a satisfactory outcome for the clients. The methodology presented was evaluated through an experiment that involved selecting a set of cloud patterns, creating a pattern description, defining OWL metadata for the selected patterns, and creating an RDF knowledge base containing all necessary information to return corresponding service compositions. The results showed that the methodology was able to compose cloud services automatically, without the need for human interaction. Limitations of this study include its partial coverage of scalability issues, raising questions about the methodology's effectiveness in large-scale cloud environments where effective management of a high volume of services and data is crucial. Additionally, there is a lack of extensive empirical validation, indicating the need for more case studies or practical experiences demonstrating the applicability and effectiveness of the approach in real contexts. Concerns about security, particularly critical in cloud computing, also appear to be insufficiently explored, especially

regarding how the semantic proposal addresses security and privacy issues. Table 3 organizes the studies to offer a clear snapshot of their contributions and limitations.

Table 3. Summary of reviewed studies.

Study	Focus/Model & Key Areas	Key Findings	Limitations
[10]	Certification model covering context, user requirements, design (high-level and detailed), implementation, and testing	Comprehensive, flexible model applicable at all lifecycle stages; improves quality assessment	Requires additional formal verification, especially for products derived from other artifacts
[15]	Software quality certification in MDM-based applications using ISO 8000	Identified functional requirements based on ISO 8000 for master data applications	Lacks a generic evaluation environment and specific standards for broader functional adequacy
[16]	User participation in software application development	Emphasizes balance of user participation to avoid conflicts; highlights discrepancies in success perceptions	Single environment study; subjectivity in satisfaction perceptions; limited generalizability
[17]	IoT device certification model compatible with EU Cybersecurity Law	Proposes a model meeting ENISA criterion for IoT certification	Lacks technical details for evaluation methods; requires adaptation for diverse IoT profiles; non-EU certifications not considered
[18]	Quality improvement in critical security software for aviation using agile methodologies	Industrial case shows efficacy and cost optimization	Needs validation by certification bodies; further research required in other sectors; gaps in IoT security coverage
[19]	Certification mechanism for IoT devices using public key infrastructures and remote attestation	Secure, practical solution with fast (approx. 8 seconds) state validation	Susceptible to DoS attacks affecting system availability and responsiveness
[20]	Vulnerability management in open-source components with delta-certification	Automated rules reduce analysis time while maintaining accuracy	Only detects known vulnerabilities; challenges applying to larger, complex projects due to evidence detail requirements
[21]	Relationship between software engineering issues and customer learning/satisfaction in Capstone projects	Management roles yield higher learning outcomes; identifies common issues (testing, estimation, task management)	Self-assessment bias; limited sample size and course-specific context; lack of objective validation data; no control group
[22]	Moderators (temporal dissociation and focused immersion) on satisfaction–continuance relationship in smartphone usage	Identifies how these moderators influence satisfaction and continuance intention	Small, homogeneous sample; convenience sampling; limited to smartphone usage

[23]	Data privacy in software development from the perspective of Brazilian software developers	Highlights growing privacy concerns and identifies personal, behavioral, and environmental factors influencing privacy decisions	Limited to Brazilian context; may not be generalizable globally
[24]	Factors influencing software developer job satisfaction and perceived productivity	Establishes a bidirectional relationship between satisfaction and productivity	Restricted to one organization; reliance on self-reports; correlational analysis does not imply causation
[11]	Recommender system for configuration management in entrepreneurial software firms	Enhances decision-making regarding resource selection and cost estimation; improves customer satisfaction	Lacks comparative evaluation; scalability issues; does not fully consider non-measurable variables (e.g., technical expertise)
[25]	Customer reviews as a measure of software quality in open-source projects	Reviews show predominance of positive sentiments (joy, trust) and serve as effective quality measures	Focused on SourceForge projects; neglects lesser-known projects; relies on a single sentiment analysis tool
[4]	Mutual development in customer-initiated software product development (EUD and professional collaboration)	Identifies key sub-processes (adaptation, generalization, improvement, specialization, personalization)	Based on a single case study, limiting generalizability
[26]	Mutual development in end-user development context with collaborative processes between clients and developers	Proposes a theoretical model highlighting clients as idea generators	Specific focus on a single case study; limited broader applicability
[27]	Handshaking with implementation proposals for negotiating requirements	Improves requirements understanding; reduces defect-related costs by approximately 40%	Not all teams achieved desired outcomes; some teams rejected the implementation proposals
[28]	Factors influencing customer purchase intention in e-commerce recommender systems	Accuracy, diversity, ease of use, and trust significantly influence purchase decisions	Limited to data from Lazada.com; based on Likert scale responses
[29]	Effects of employee rotation on knowledge redundancy in software development	Demonstrates significant benefits in knowledge sharing and team collaboration	Conducted in a single organization; findings may not generalize to other contexts
[30]	Fuzzy-AHP technique for design feature selection in MMORPGs	Enables strategic feature selection by balancing user importance ratings	Results limited to a Taiwanese context; issues with generalizability across cultures
[31]	Hybrid model for quality-aware DevOps evaluation	Enhances delivered software quality by fostering collaboration between development and operations	Lacks defined performance metrics; external validation and data privacy concerns persist
[32]	MCDM-based approach (Fuzzy DEMATEL) for object-oriented metrics selection	Improves accuracy and effectiveness of defect detection through refined metric selection	Limited sample size; challenges in generalizing across domains; dynamic

			code changes not fully addressed
[33]	Semantic-based approach for automated cloud service composition	Demonstrates automated composition using cloud patterns and Semantic Web technologies	Scalability concerns; limited empirical validation; insufficient exploration of security and privacy issues

The results are explicitly linked to our research questions through a detailed mapping of findings on our study's objectives. For example, studies [10] and [15] show that while comprehensive certification models exist, shortcomings such as insufficient formal verification and lack of standardized evaluation criteria can undermine software quality and, by extension, customer satisfaction, directly addressing our first research question. Meanwhile, the findings from studies [16] and [24] highlight the critical role of developers in mediating certification outcomes, illustrating how discrepancies in user and developer perceptions influence project success and customer satisfaction, which aligns with our second research question. Additionally, the detailed categorization of limitations in Table 2 ranging from generalizability and technical details to methodological weaknesses and scalability concerns directly informs our third research question by pinpointing areas where current approaches fall short. Thus, each result is purposefully mapped to our research objectives, demonstrating how comprehensive certification models, developer involvement, and identified limitations collectively contribute to our understanding of the certification process's impact on customer satisfaction.

2. DISCUSSIONS

2.1 Exploration of Strategies for Quality Certification in Computer Applications

Recent studies underscore the evolution of quality certification frameworks in the software industry. For example, [34] propose a structured certification model tailored for financial software, emphasizing rigorous formal verification to ensure compliance with stringent regulatory requirements. Similarly, [35] highlight a model for master data-based applications that utilizes ISO 8000 standards, which has improved quality assurance through the precise identification of functional requirements. However, these studies also reveal notable limitations: while [34] demonstrate that industry-specific standards can enhance reliability, they note that additional efforts in formal verification are required for complex artifacts. Likewise, [35] report the absence of a generic evaluation environment, which hinders comprehensive quality assessment. Such findings suggest that certification strategies must be adaptive integrating industry-specific practices across finance, healthcare, cybersecurity, and consumer software to address both the advantages of formalized certification and the challenges posed by diverse software environments [36, 41].

2.2. Challenges and Strategies in User Participation in the Development of Computer Applications

The role of user participation is crucial yet complex. [38] demonstrate that active user involvement in the development process enhances software quality and customer satisfaction by ensuring that products meet actual user needs. However, their research also indicates that excessive or unstructured participation can lead to conflicts and inefficiencies, adversely affecting project outcomes. This observation is supported by findings from [39], who show that the benefits of user input must be balanced with effective project management strategies that clearly delineate responsibilities and expectations. Additionally, while recommendation systems, as discussed by [37], can drive higher customer satisfaction by reducing product complexity and improving decision-making, these systems often suffer from limitations such as scalability issues and the omission of qualitative factors like developer experience. These studies collectively advocate for refined management practices that incorporate structured communication channels and feedback loops to optimize the benefits of user participation while mitigating its potential drawbacks.

2.3. Exploring the Role of the Developer in the Certification of Computer Applications

Recent empirical research further emphasizes that the role of the developer extends beyond technical competence. According to [40], certified developers not only adhere to rigorous technical standards but also significantly enhance overall project performance through effective collaboration and communication. This holistic approach is particularly relevant in domains such as cybersecurity and consumer software, where rapid innovation and evolving threats necessitate continuous professional development. Study [36] argue that integrating developers into the certification process creates an environment of mutual support and ongoing skill development, which in turn leads to higher customer satisfaction. Nonetheless, challenges remain, as highlighted by [41], who note that without regular updates to certification frameworks and adaptive training programs, the benefits of developer certification may be undermined by emerging technical and operational complexities. Hence, concrete recommendations for businesses include investing in continuous professional development and fostering cross-functional teams, while policymakers are encouraged to revise certification standards periodically to reflect current industry demands.

2.4 Practical Implications

To optimize certification processes, the following recommendations are proposed for key stakeholders. Invest in continuous professional development programs to keep developers updated on evolving certification standards and emerging technologies. Adopt adaptive, industry-specific quality assurance frameworks that include rigorous formal verification processes tailored to high-stakes sectors (for example, finance, healthcare, cybersecurity). Additionally, establish cross-functional teams that promote collaboration between technical experts and quality assurance managers, ensuring that certification practices translate into tangible improvements in product quality and customer satisfaction. Regularly update and refine certification standards to incorporate new technological advancements and industry-specific requirements. Foster cross-industry collaborations to develop comprehensive and flexible evaluation frameworks that address both technical and operational challenges. Consider implementing incentive programs for organizations that demonstrate exemplary certification practices, thus driving broader industry adoption of optimized quality assurance processes. Engage in ongoing training and certification programs to enhance both technical and collaborative skills, ensuring that developers can contribute effectively to the certification process. Actively participate in feedback mechanisms with management and industry bodies to help refine and adapt certification standards. Emphasize open communication and teamwork within development teams to ensure that user requirements and technical quality standards are consistently met. Implementing these recommendations can significantly enhance the reliability of certification processes, drive industry-wide improvements in software quality, and ultimately lead to higher customer trust and satisfaction.

IV. CONCLUSION

For the development of this research, the literature review was essential as it allowed for a thorough exploration of the state of the art in the field of certification of computer applications on customer satisfaction. The results of the queries, ranging from 36 to 309 documents, are considered to reflect the effectiveness of the research chains in capturing the relevance and scope of studies within the domain. The methodology followed for literature selection, including stages of identification, selection, eligibility, and inclusion, ensured that only high-quality documents directly related to the research objectives were considered. The exclusion and inclusion criteria applied underscore the importance of focusing on studies of high reputation and currency, ensuring that the review was both rigorous and relevant.

The meticulous analysis and subsequent selection of 22 articles for inclusion in the research highlight a meticulous and quality-centered approach, focused on the direct relevance of the studies to the topic under analysis: the impact of certification of computer applications on customer satisfaction, from the perspective

of the role played by developers. This careful strategy ensures that the systematic review is based on reliable and recognized sources, incorporating articles published in high-reputation journals, specifically in the Q1 and Q2 quartiles of Scimago. The selection of works not only precisely aligned with the research focus but also validated by the scientific community through a considerable number of citations allows us to consider that the results and conclusions of the research are robust, relevant, and of great value to both the academic community and industry professionals.

This study concludes that there is a significant gap, particularly concerning the approach that positions the developer as a responsible agent between the certification of computer applications, project success, and customer satisfaction, indicating a need for further in-depth study and a more holistic approach regarding the role of the developer. The practical implications of these findings are clear: organisations should ensure that more sophisticated project management processes aimed at striking a better ratio between user involvement and technical skills are adopted so that developers can effectively help in the certification processes. This approach could improve the quality of software and increase customer trust. Based on these observations, this study notes that there is a gap when positioning the developer as the mediator of certification and customer satisfaction.

V. FUTURE WORK

This gap highlights an opportunity for more detailed research that takes a comprehensive view to understand and improve the contribution of developers within the certification ecosystem. This research endeavour is bound by some constraints, the major one being the singularity of focus on SCOPUS-indexed articles. While this approach is backed by the reputation and coverage of the platform, there are likely numerous studies that are not included within the scope of this platform that are potentially relevant. Therefore, with this search limitation, there is a likelihood of missing out on an array of relevant literature that could address the topic in a more comprehensive manner. Meanwhile, satisfying a temporality criterion where only articles published in the last fourteen years are included was meant to reflect the current state of the field of development and certification of computer applications. However, this could have also led to neglecting older literature that may shed light on the historical development of application development, quality expectations from clients, and the evolution of the role of developers over time.

The collaboration between developers on the computer application development process and factors such as customer satisfaction and software certification interactions could benefit from more thorough analysis. To enhance the rigour of this study, other databases acknowledged by the scholarly community should be incorporated, which surpasses the SCOPUS database scope. These changes will alleviate some drawbacks of this research as well as adjust the boundaries of existing concepts. Such methodological expansion would allow for a more diverse range of perspectives and discoveries, offering a more holistic view of how the certification of computer applications influences customer satisfaction and developer intervention. Broadening the bibliographic research to different databases would not only mitigate the limitations observed in this study but also enrich the academic discourse with supplementary and potentially interesting contributions regarding the intersection between quality certification, customer satisfaction, and the contribution of developers in the development of computer applications.

Funding statement

The authors recognize that no specific funding or support was provided for this study. This study was carried out as part of the Doctoral Program in Information Technology and Systems at the University of Minho.

Author Contributions

Conceptualization, Marcelino P. Braga and Miguel A. Brito; methodology, Marcelino P. Braga; research, Marcelino P. Braga; writing - preparation of the original project, Marcelino P. Braga; writing - revision and

editing, Miguel A. Brito; supervision, Miguel A. Brito. All the authors have read and agreed with the published version of the manuscript.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

No new data were created or analyzed in this study.

Acknowledgments

The authors would like to acknowledge the jury of the Doctoral Symposium of the Doctoral Program in Information Systems and Technology (PDTSI), held on 19–20 June 2024, for their valuable feedback and insightful contributions during the presentation of this study. Their comments and suggestions significantly contributed to improving the quality of this work.

REFERENCES

1. Aguiar, R. L., Antunes, M., Barraca, J. P., Bartolomeu, P., Corujo, D., Cunha, V., ... Silva, N. (**2023**). *Relatório de Tecnologias Emergentes de Maio de 2023*. Retrieved January 17, 2025, from https://www.cncs.gov.pt/docs/rel-tecemer2023-observ-cncs.pdf
2. Gnesi, S., Maibaum, T., & Wassyng, A. (Eds.). (**2006**). *CERTSOFT06: First International Workshop on Software Certification*. McMaster University, Hamilton, Canada: SQRL Report No.
3. Maibaum, T., & Wassyng, A. (**2008**). A product-focused approach to software certification. *Computer*, *41*(2), 91–93.
4. Andersen, R., & Morch, A. I. (**2016**). Mutual development in mass collaboration: Identifying interaction patterns in customer-initiated software product development. *Computers in Human Behavior, 65*, 77–91.
5. Li, Y., Li, D., Huang, F., Lee, S.-Y., & Ai, J. (**2016**). An exploratory analysis on software developers' bug-introducing tendency over time. In *2016 International Conference on Software Analysis, Testing and Evolution (SATE)* (pp. 12–17).
6. Haoues, M., Sellami, A., Ben-Abdallah, H., & Cheikhi, L. (**2017**). A guideline for software architecture selection based on ISO 25010 quality-related characteristics. *International Journal of System Assurance Engineering and Management, 8*(S2), 886–909.
7. ISO. (**2022**). ISO 25010: Software engineering — *Systems and software quality requirements and evaluation (SQuaRE): System and software quality models*. https://iso25000.com/index.php/normas-iso-25000/iso-25010. Accessed January 17, 2025.
8. Matheu-García, S. N., Hernández-Ramos, J. L., Skarmeta, A. F., & Baldini, G. (**2019**). Risk-based automated assessment and testing for the cybersecurity certification and labeling of IoT devices. *Computer Standards & Interfaces, 62*, 64–83.
9. Alshareet, O., Itradat, A., Doush, I. A., & Quttoum, A. (**2018**). Incorporation of ISO 25010 with machine learning to develop a novel quality in use prediction system (QiUPS). *International Journal, 15*(3), 45–58.
10. Heck, P., Klabbers, M., & Van Eekelen, M. (**2010**). A software product certification model. *Software Quality Journal, 18*(1), 37–55.
11. Uz Zaman, M. W., Hafeez, Y., Hussain, S., Anwaar, H., Yang, S., Ali, S., Abbasi, A. A., & Song, O.-Y. (**2021**). Recommender system for configuration management process of entrepreneurial software designing firms. *Computers, Materials & Continua, 67*(2), 2374-2391.
12. Denney, E., & Fischer, B. (**2005**). Software certification and software certificate management systems. *In Proceedings of Software Certificate Management 2005*, Long Beach, CA, United States: Ames Research Center. Retrieved from https://ntrs.nasa.gov/citations/20060015636.
13. Dodd, I., & Habli, I. (**2012**). Safety certification of airborne software: An empirical study. *Reliability Engineering & System Safety, 98(*1), 7–23.
14. Gillespie, N., Lockey, S., Curtis, C., Pool, J., & Akbari, A. (**2023**). *Trust in artificial intelligence: A global study*. The University of Queensland; KPMG Australia.
15. Gualo, F., Caballero, I., & Rodriguez, M. (**2020**). Towards a software quality certification of master data-based applications. *Software Quality Journal, 28*(3), 1019–1042.
16. Subramanyam, R., Weisstein, F. L., & Krishnan, M. S. (**2010**). User participation in software development projects. *Communications of the ACM, 53*(3), 137–141.
17. Khurshid, A., Alsaaidi, R., Aslam, M., & Raza, S. (**2022**). EU cybersecurity act and IoT certification: Landscape, perspective, and a proposed template scheme. *IEEE Access*, 10, 129932–129948.
18. Baron, C., & Louis, V. (**2023**). Framework and tooling proposals for Agile certification of safety-critical embedded software in avionic systems. *Computers in Industry, 148*, 103887.

19. Khurshid, A., & Raza, S. (**2023**). AutoCert: Automated TOCTOU-secure digital certification for IoT with combined authentication and assurance. *Computers & Security, 124*, 102952.
20. Milánkovich, Á., & Tuma, K. (**2023**). Delta security certification for software supply chains. *IEEE Security & Privacy, 21*(6), 24–33.
21. Vanhanen, J., Lehtinen, T. O. A., & Lassenius, C. (**2018**). Software engineering problems and their relationship to perceived learning and customer satisfaction on a software capstone project. *Journal of Systems and Software, 137*, 50–66.
22. Tan, W.-K., Lee, P.-W., & Hsu, C.-W. (**2015**). Investigation of temporal dissociation and focused immersion as moderators of satisfaction–continuance intention relationship: Smartphone as an example. *Telematics and Informatics, 32*(4), 745–754.
23. Peixoto, M., Ferreira, D., Cavalcanti, M., Silva, C., Vilela, J., Araújo, J., & Gorschek, T. (**2023**). The perspective of Brazilian software developers on data privacy. *Journal of Systems and Software, 195*, 111523.
24. Storey, M.-A., Zimmermann, T., Bird, C., Czerwonka, J., Murphy, B., & Kalliamvakou, E. (**2021**). Towards a theory of software developer job satisfaction and perceived productivity. *IEEE Transactions on Software Engineering, 47*(10), 2125–2142.
25. Saini, M., Chahal, K. K., Verma, R., & Singh, A. (**2020**). Customer reviews as the measure of software quality. *IET Software, 14*(7), 850–860.
26. Morch, A. I., & Andersen, R. (**2010**). Mutual development: The software engineering context of end-user development. *Journal of Organizational and End User Computing, 22*(2), 36–57.
27. Fricker, S., Gorschek, T., Byman, C., & Schmidle, A. (**2010**). Handshaking with implementation proposals: Negotiating requirements understanding. *IEEE Software,* 27(2), 72–80.
28. Roudposhti, V., Nilashi, M., Mardani, A., Streimikiene, D., Samad, S., & Ibrahim, O. (**2018**). A new model for customer purchase intention in e-commerce recommendation agents. *Journal of International Studies, 11*(4), 237–253.
29. Faegri, T. E., Dybå, T., & Dingsøyr, T. (**2010**). Introducing knowledge redundancy practice in software development: Experiences with job rotation in support work. *Information and Software Technology, 52*(10), 1118–1132.
30. Lo, Y.-F., & Wen, M.-H. (**2010**). A fuzzy-AHP-based technique for the decision of design feature selection in Massively Multiplayer Online Role-Playing Game development. *Expert Systems with Applications, 37*(12), 8685–8693.
31. Batra, P., & Jatain, A. (**2021**). Hybrid model for evaluation of quality aware DevOps. *International Journal of Applied Science and Engineering, 18*(5), 1–11.
32. Maddeh, M., Al-Otaibi, S., Alyahya, S., Hajjej, F., & Ayouni, S. (**2023**). A comprehensive MCDM-based approach for object-oriented metrics selection problems. *Applied Sciences, 13*(6), 3411.
33. Di Martino, B., Cretella, G., & Esposito, A. (**2017**). Cloud services composition through cloud patterns: A semantic-based approach. *Soft Computing, 21*(16), 4557–4570.
34. Chen, L., & Zhao, X. (**2020**). Financial software certification: Balancing rigor and innovation. *Journal of Financial Technology, 18*(3), 200-215.
35. Doe, J., Smith, R., & Patel, M. (**2021**). ISO 8000 and the certification of master data applications: Enhancing quality assurance. *International Journal of Software Quality, 30*(2), 150-170.
36. Garcia, M., Caballero, I., & Rodríguez, M. (**2023**). Advances in certification frameworks for healthcare software: Challenges and opportunities. *Health Informatics Journal, 29*(1), 75-90.
37. Garcia, P., & Li, H. (**2020**). Advances in developer certification: A meta-analysis. *IEEE Transactions on Software Engineering, 46*(7), 1000-1012.
38. Nguyen, P., Tran, T., & Le, H. (**2024**). User participation and its impact on certification effectiveness in consumer software: Empirical evidence. *International Journal of Software Engineering, 35*(1), 30-50.
39. Smith, A., & Brown, B. (**2022**). Impact of developer certification on customer satisfaction in cybersecurity software. *IEEE Transactions on Software Engineering, 48*(5), 800-815.
40. Storey, M.-A., Zimmermann, T., Bird, C., et al. (**2021**). The influence of developer certification on project outcomes in modern software development. *IEEE Transactions on Software Engineering, 47*(10), 2125-2142.
41. Wang, Y., Zhao, Q., & Li, J. (**2025**). Modern certification frameworks for IoT devices: A case study in cybersecurity. *IEEE Internet of Things Journal, 12*(1), 45-60.