JASDG
JOURNAL FOR THE ADVANCEMENT OF SUSTAINABLE DEVELOPMENT GOALS
Advancing Knowledge. Inspiring Innovation. Achieving the SDGs.
AN INTERNATIONAL, PEER-REVIEWED, OPEN ACCESS JOURNAL
www.jasdg.com
Volume 1 Issue 3 Pages 1221-1247

GLOBAL RESEARCH TRENDS IN QUALITY ASSURANCE, SOFTWARE TESTING, AND SOFTWARE QUALITY: A BIBLIOMETRIC ANALYSIS

Angel Cylo G. Real
Faculty
Pateros Technological College
Metro Manila, Philippines
Email: cylo11388@gmail.com

Geraldine Mae H. Tamboong
Faculty
Pateros Technological College
Metro Manila, Philippines
Email: ghtamboong@paterostechnologicalcollege.edu.ph

Mary Grace M. Belgar
Faculty
Pateros Technological College
Metro Manila, Philippines
Email: mmbelgar@paterostechnologicalcollege.edu.ph

Michael Noel B. Tolentino
Faculty
Pateros Technological College
Metro Manila, Philippines
Email: mbtolentino@paterostechnologicalcollege.edu.ph

Gloria R. Garcia
Faculty
Pateros Technological College
Metro Manila, Philippines
Email: grgarcia@paterostechnologicalcollege.edu.ph

Jeffrey P. Oberas
Faculty
Pateros Technological College
Metro Manila, Philippines
Email: jpoberas@paterostechnologicalcollege.edu.ph

Joshua D. Mendoza
Faculty
Pateros Technological College
Metro Manila, Philippines
Email: jmendoza@paterostechnologicalcollege.edu.ph

© 2026 Advancing SDG Impact. All rights reserved.

JASDG
JOURNAL FOR THE ADVANCEMENT OF SUSTAINABLE DEVELOPMENT GOALS
Advancing Knowledge. Inspiring Innovation. Achieving the SDGs.
AN INTERNATIONAL, PEER-REVIEWED, OPEN ACCESS JOURNAL
www.jasdg.com
Volume 1 Issue 3 Pages 1221-1247

Jayson A. Batoon
Faculty
La Consolacion University
Malolos, Bulacan, Philippines
Email: jayson.batoon@email.lcup.edu.ph

Corresponding Author

Angel Cylo G. Real
Faculty
Pateros Technological College
Metro Manila, Philippines
Email: cylo11388@gmail.com

Abstract

This study conducts a bibliometric analysis of research trends in Quality Assurance, Software Testing, Software Quality, and Quality Engineering to examine the development, intellectual structure, influential contributions, and emerging thematic directions within the field. Bibliographic data retrieved from the Scopus database were analyzed using VOSviewer to provide a systematic representation of the research landscape across multiple publication years. Three major bibliometric techniques were employed: citation analysis, co-citation analysis, and co-word analysis. Citation analysis was used to identify influential documents, publication sources, authors, and countries, while co-citation analysis examined relationships among frequently cited references to reveal the intellectual foundations and major research clusters of Software Quality Assurance. Co-word analysis was applied to keyword relationships to identify prominent and emerging research themes within the literature. The findings demonstrate that contemporary Software Quality Assurance research is supported by several interconnected areas, including software testing, test automation, software defect prediction, software metrics, model validation, machine learning, code review, DevOps, and intelligent quality engineering. The co-citation analysis further reveals that traditional software testing and measurement principles remain important intellectual foundations, while machine learning-based defect prediction, automated testing, and intelligent software analysis represent increasingly significant research directions. Overall, the findings indicate a continuing transition from conventional testing and quality control toward more automated, continuous, predictive, data-driven, and AI-assisted Quality Engineering approaches. The study provides a structured understanding of the evolving Software Quality Assurance research landscape and highlights the growing integration of established software engineering practices with emerging intelligent technologies.

Keywords: *software quality assurance, software testing, defect prediction, large language models, bibliometric analysis, DevOps.*

© 2026 Advancing SDG Impact. All rights reserved.

JASDG
JOURNAL FOR THE ADVANCEMENT OF SUSTAINABLE DEVELOPMENT GOALS
Advancing Knowledge. Inspiring Innovation. Achieving the SDGs.
AN INTERNATIONAL, PEER-REVIEWED, OPEN ACCESS JOURNAL
www.jasdg.com
Volume 1 Issue 3 Pages 1221-1247

I. Introduction

Software Quality Assurance (QA) is a fundamental area of software engineering that ensures software products meet established quality standards, functional requirements, and user expectations through activities such as software testing, test case management, defect and bug tracking, quality measurement, test automation, and quality analytics (Parveen et al., 2007; Sultan et al., 2013; López et al., 2021). As software systems become increasingly complex, QA has evolved from conventional testing practices toward a broader Quality Engineering approach that integrates quality throughout the software development lifecycle, including modern Agile and DevOps environments (Rafi et al., 2025; Olcay et al., 2025). The increasing adoption of automated testing, machine learning, deep learning, and Artificial Intelligence (AI) has further transformed how software quality is evaluated, monitored, and improved, particularly through techniques such as software defect prediction and intelligent quality assessment (Hai et al., 2022; Parija et al., 2025). More recently, Large Language Models (LLMs) have emerged as an important research direction, with studies investigating their application to automated test case generation, vulnerability-focused unit testing, code analysis, bug report generation, and other software testing activities (Antal et al., 2025; Antonio et al., 2025; Yang, 2025). Despite these technological developments, traditional foundations such as software testing techniques, test management, defect management, software metrics, and quality assessment remain essential components of contemporary Software QA research (Alshamaa & Saleem, 2025; Parveen et al., 2007).

Given the continuing development of the field, this study conducts a bibliometric analysis of Quality Assurance, Software Testing, Software Quality, and Quality Engineering research using bibliographic records retrieved from the Scopus database, following established bibliometric principles for examining scholarly performance and intellectual structures (Donthu et al., 2021). Using VOSviewer, the study applies citation analysis to identify influential documents, publication sources, authors, and countries; co-citation analysis to examine relationships among frequently cited scholarly works; and co-word analysis to identify major research themes and emerging topics based on keyword relationships. Through these analyses, the study aims to provide a systematic understanding of the Software Quality Assurance research landscape and examine how established areas such as software testing, defect management, and software quality are increasingly connected with emerging approaches involving automation, DevOps, predictive analytics, machine learning, and artificial intelligence.

Related Literature

Software Quality Assurance (SQA) has developed into a broad research domain that integrates software testing, defect management, quality measurement, process improvement, analytics, and increasingly, artificial intelligence. Traditional QA research emphasizes systematic testing and test management as essential mechanisms for verifying software behavior and maintaining traceability throughout the software development lifecycle. Parveen et al. (2007) highlighted the importance of structured test management practices, while Bhagwat (2009) emphasized treating test cases as engineered artifacts that should be designed, maintained, and managed systematically. More recent work continues to recognize the importance of established testing techniques, including black-box testing, as fundamental approaches for validating software functionality and identifying defects without requiring knowledge of internal code structures (Alshamaa & Saleem, 2025). These studies demonstrate that, despite the emergence of new technologies, software testing remains a central foundation of Quality Assurance research.

Defect management and software defect prediction constitute another major area of the literature. Early studies focused on structuring and classifying defect reports to improve software quality control and support more efficient defect resolution processes (Sultan et al., 2013). As software repositories and bug-tracking systems have become richer sources of development data, research has increasingly applied machine learning and deep learning techniques to predict software defects and identify fault-prone components. Hai et al. (2022) demonstrated the use

© 2026 Advancing SDG Impact. All rights reserved.

JASDG
JOURNAL FOR THE ADVANCEMENT OF SUSTAINABLE DEVELOPMENT GOALS
Advancing Knowledge. Inspiring Innovation. Achieving the SDGs.
AN INTERNATIONAL, PEER-REVIEWED, OPEN ACCESS JOURNAL
www.jasdg.com
Volume 1 Issue 3 Pages 1221-1247

of deep learning for cloud-based bug tracking and software defect analysis, while Parija et al. (2025) proposed a stacking ensemble framework, combined with advanced feature engineering, to improve software defect prediction performance. These developments show a transition from reactive defect handling toward predictive Quality Assurance, where historical software data and analytical models are used to identify potential quality risks before defects become more costly to resolve.

Quality analytics has also become increasingly important in modern software engineering because development teams generate substantial amounts of data related to defects, code quality, testing progress, software metrics, and release readiness. Vollmer et al. (2019) examined the practical value of software analytics for managing quality, demonstrating how software data can support evidence-based decision-making. Similarly, López et al. (2021) introduced the Quality-aware Strategic Dashboard (QaSD), which integrates quality-related indicators into strategic decision support for Agile software development. Such approaches illustrate how QA has expanded beyond test execution toward continuous monitoring and quality intelligence, enabling managers, product owners, and development teams to evaluate quality trends and make more informed decisions throughout the development lifecycle.

The adoption of Agile and DevOps practices has further transformed the role of Quality Assurance. Modern software development emphasizes short development cycles, continuous integration, rapid deployment, and frequent feedback, requiring testing activities to be embedded continuously rather than performed as a separate final phase. Rafi et al. (2025) examined software product quality in DevOps environments from an ISO 25010:2023 perspective, highlighting the need to maintain recognized software quality characteristics while supporting rapid development and deployment. Olcay et al. (2025) likewise examined corporate QA test processes, illustrating how organizational testing practices must adapt to modern software delivery environments. These studies suggest that contemporary Quality Assurance increasingly operates as a continuous engineering function integrated across development, operations, and release management activities.

Artificial Intelligence has become one of the most significant emerging themes in recent QA research. AI-based methods are being applied to test automation, defect prediction, software analysis, and quality assessment, reducing reliance on purely manual testing processes. In particular, Large Language Models (LLMs) are creating new opportunities for automated and intelligent QA activities. Antonio et al. (2025) investigated prompt-orchestrated LLM workflows for automated test case generation in Agile environments, demonstrating the potential of generative AI to translate development information into structured testing artifacts. Antal et al. (2025) explored the use of GPT-4 for generating vulnerability-witnessing unit tests, showing how LLMs may support both functional and security-related testing activities. These studies indicate that generative AI is increasingly being positioned as an assistant for software testers and developers rather than merely as a general-purpose text-generation technology.

LLM applications in QA extend beyond test case generation. Yang (2025) proposed a reinforcement learning framework for generating reliable bug reports using open-source LLMs, illustrating how AI can support defect documentation and issue management. Eagal et al. (2025) examined the dependability of LLMs for code clone generation, raising important questions regarding the reliability and consistency of AI-generated software artifacts. Together, these studies demonstrate that while LLMs offer significant potential for automating QA tasks, their outputs must still be evaluated carefully for correctness, dependability, and practical usefulness. This emerging research direction suggests that future Quality Assurance environments may increasingly combine human expertise with AI-supported generation, analysis, and recommendation capabilities.

Software architecture and static analysis also remain important components of the broader quality research landscape. Esposito et al. (2025) investigated the relationship between architectural smells and static analysis warnings, demonstrating that structural software problems and code-level quality indicators may be

© 2026 Advancing SDG Impact. All rights reserved.

JASDG
JOURNAL FOR THE ADVANCEMENT OF SUSTAINABLE DEVELOPMENT GOALS
Advancing Knowledge. Inspiring Innovation. Achieving the SDGs.
AN INTERNATIONAL, PEER-REVIEWED, OPEN ACCESS JOURNAL
www.jasdg.com
Volume 1 Issue 3 Pages 1221-1247

interconnected. Such findings reinforce the importance of preventive quality practices that identify potential maintainability and design problems before they lead to more serious software failures. In this context, QA extends beyond dynamic testing and includes static analysis, architectural evaluation, code review, and software maintainability assessment as complementary mechanisms for improving overall product quality.

Software Quality Assurance research has also expanded into specialized industrial and embedded-system environments. Bogorin-Predescu et al. (2025) examined a cost-effective hybrid remote testing approach for improving ECU firmware quality in the automotive domain. This type of research illustrates how QA practices must be adapted to systems in which software interacts directly with hardware and where failures may have significant operational or safety consequences. Embedded systems, automotive software, and hardware-in-the-loop testing, therefore, represent specialized but increasingly important areas within Quality Assurance, particularly as software becomes more deeply integrated into physical products and intelligent systems.

Research Objectives

This study aims to conduct a comprehensive bibliometric analysis of the academic literature on Quality Assurance, Software Testing, and Software Quality Engineering. As organizations increasingly rely on software systems to support digital transformation, business operations, and technological innovation, ensuring software quality has become a fundamental aspect of software development. Quality assurance practices, including software testing, quality engineering, automated testing, and continuous quality improvement, play a critical role in delivering reliable, secure, and high-performing software applications. Despite the substantial growth of research in this field, there remains a need to systematically map the intellectual structure, research trends, and emerging themes that characterize quality assurance and software testing research. Specifically, the study sought to:

1. Identify the most influential publications, journals, authors, institutions, and countries contributing to research on Quality Assurance, Software Testing, and Software Quality Engineering through citation analysis;
2. Analyze the intellectual structure of the field by examining relationships among highly cited publications and identifying the major research clusters that have shaped the development of software quality assurance, software testing methodologies, quality engineering, and software reliability through co-citation analysis; and
3. Explore the thematic structure of the field by analyzing keyword co-occurrence patterns and identifying the major research topics, technological trends, and emerging areas in Quality Assurance, including test automation, artificial intelligence in software testing, DevOps, continuous integration and continuous deployment (CI/CD), software quality metrics, defect prediction, quality engineering, and software reliability through co-word analysis.

© 2026 Advancing SDG Impact. All rights reserved.

JASDG
JOURNAL FOR THE ADVANCEMENT OF SUSTAINABLE DEVELOPMENT GOALS
Advancing Knowledge. Inspiring Innovation. Achieving the SDGs.
AN INTERNATIONAL, PEER-REVIEWED, OPEN ACCESS JOURNAL
www.jasdg.com
Volume 1 Issue 3 Pages 1221-1247

II. Methods

Research Design

This study employs a quantitative bibliometric approach to examine the intellectual structure, research trends, and thematic evolution of scholarly publications on Quality Assurance, Software Testing, Software Quality, and Quality Engineering. Bibliometric analysis is a widely recognized research method for evaluating scientific productivity, identifying influential publications, mapping collaboration networks, and exploring the conceptual development of a research field through bibliographic metadata such as citations, authors, keywords, affiliations, and publication sources (Donthu et al., 2021). By applying bibliometric techniques, this study provides a systematic overview of the current state of quality assurance research and identifies emerging research directions that have shaped software engineering in recent years.

The methodology follows established bibliometric procedures, including database selection, search strategy formulation, data collection, publication screening, and bibliometric mapping with VOSviewer. The analysis incorporates citation, co-citation, and co-word (keyword co-occurrence) analyses to identify influential publications, productive authors, leading institutions, contributing countries, and major thematic clusters within the Quality Assurance research landscape. These complementary techniques enable the visualization of intellectual relationships, collaboration patterns, and emerging topics that characterize the evolution of software quality assurance research (Donthu et al., 2021). The bibliometric maps generated through VOSviewer facilitate the graphical representation of citation networks, co-citation relationships, and keyword co-occurrence patterns, providing a comprehensive understanding of the field's knowledge structure. Citation analysis is used to identify influential publications and contributors, while co-citation analysis examines relationships among frequently cited references to reveal the intellectual foundations of Software QA research. Co-word analysis examines keyword co-occurrence patterns to identify major thematic clusters, relationships among research concepts, and emerging areas such as test automation, artificial intelligence, Large Language Models, defect prediction, DevOps, and quality engineering.

Data Collection

The primary data for this research were retrieved from the Scopus database, which is widely recognized as one of the most comprehensive and reputable sources of peer-reviewed scholarly literature for bibliometric studies (Baas et al., 2020). A targeted search strategy was developed to identify publications related to Quality Assurance, Software Testing, Software Quality, and Quality Engineering within the field of computer science. The search string used was TITLE-ABS-KEY((“quality assurance” AND (“software testing” OR “software quality” OR “quality engineering”))) AND PUBYEAR > 2020 AND PUBYEAR < 2026 AND (LIMIT-TO(LANGUAGE, “English”)) AND (LIMIT-TO(SUBJAREA, “COMP”)). This search strategy was designed to retrieve studies in which quality assurance is associated with software testing, software quality, or quality engineering, thereby capturing relevant research on software testing methodologies, defect management and prediction, test automation, software quality assessment, DevOps quality practices, and AI-assisted testing. The search was limited to publications from 2021 to 2025 to ensure that the dataset reflects recent developments and contemporary research trends in software quality assurance. Furthermore, only publications written in English and categorized under the Computer Science subject area were included to maintain the relevance and consistency of the retrieved literature. The resulting Scopus records contained bibliographic information such as titles, authors, publication years, sources, citations, affiliations, keywords, and cited references, which were subsequently used to conduct the citation analysis, co-citation analysis, and co-word analysis of the study.

Table 1 presents the search keywords and their corresponding justifications used to retrieve publications from the Scopus database. The selected keywords were designed to comprehensively capture scholarly literature related to Quality Assurance, Software Testing, Software Quality, and Quality Engineering within the software engineering

© 2026 Advancing SDG Impact. All rights reserved.

JASDG
JOURNAL FOR THE ADVANCEMENT OF SUSTAINABLE DEVELOPMENT GOALS
Advancing Knowledge. Inspiring Innovation. Achieving the SDGs.
AN INTERNATIONAL, PEER-REVIEWED, OPEN ACCESS JOURNAL
www.jasdg.com
Volume 1 Issue 3 Pages 1221-1247

domain. The keyword "quality assurance" was included to identify studies focusing on systematic processes and practices for ensuring software quality throughout the software development lifecycle, while the keywords "software testing," "software quality," and "quality engineering" were incorporated to broaden the search by including research on software verification, validation, defect management, quality assessment, and emerging technologies such as AI-assisted testing and DevOps.

Table 1. ***Search String in Scopus Database***

No.	Keywords	Justification
1	"quality assurance"	To capture a broad range of publications related to software quality assurance — the systematic processes, methodologies, and practices used to ensure that software products meet specified quality standards throughout the software development lifecycle, encompassing quality planning, quality control, process improvement, and continuous monitoring.
2	"software testing" OR "software quality" OR "quality engineering"	To capture publications related to software verification, validation, and quality management, including functional and non-functional testing, test automation, defect management, quality engineering, software reliability, and quality assessment, as well as emerging areas such as AI-assisted testing, DevOps quality assurance, continuous testing, and software quality analytics.

Screening Process

To ensure the relevance and consistency of the dataset, a systematic screening process was applied to the publications retrieved from the Scopus database. The search was limited to publications published between 2021 and 2025, written in English, and classified under the Computer Science subject area. Publications were considered relevant when they addressed Quality Assurance in relation to Software Testing, Software Quality, or Quality Engineering, including related topics such as test automation, defect prediction, DevOps quality practices, software quality assessment, and AI-assisted testing. Records that did not satisfy the established search criteria or were outside the defined publication period, language, and subject area were excluded from the dataset. After applying the inclusion and exclusion criteria, the remaining publications formed the final dataset used for citation analysis, co-citation analysis, and co-word analysis, ensuring that the bibliometric findings were based on literature directly relevant to the scope and objectives of the study.

Data Analysis and Visualization

The final dataset was analyzed using VOSviewer to examine and visualize the intellectual and thematic structure of research related to Quality Assurance, Software Testing, Software Quality, and Quality Engineering. VOSviewer enables the construction of bibliometric networks based on citation relationships, co-citation patterns, and keyword co-occurrences, allowing large volumes of bibliographic information to be represented through network maps and thematic clusters. In this study, three primary bibliometric techniques were employed and interpreted as follows.

© 2026 Advancing SDG Impact. All rights reserved.

JASDG
JOURNAL FOR THE ADVANCEMENT OF SUSTAINABLE DEVELOPMENT GOALS
Advancing Knowledge. Inspiring Innovation. Achieving the SDGs.
AN INTERNATIONAL, PEER-REVIEWED, OPEN ACCESS JOURNAL
www.jasdg.com
Volume 1 Issue 3 Pages 1221-1247

Citation Analysis

Citation analysis was conducted to identify the most influential publications, authors, sources, institutions, and countries contributing to research on Quality Assurance and Software Testing. Citation counts were used as indicators of the scholarly influence and visibility of publications within the dataset. The analysis also examined the number of documents and total link strength generated by VOSviewer to determine the relationships among influential contributors. Publications with higher citation counts were considered to have greater scholarly impact within the defined research period, although differences in publication year were considered when interpreting the results because older publications have had more time to accumulate citations.

Co-citation Analysis

Co-citation analysis was conducted to examine the intellectual structure and knowledge foundations of Quality Assurance research by identifying references that are frequently cited together by publications in the dataset. In VOSviewer, cited references were analyzed according to their frequency of co-citation, with strongly related references grouped into clusters representing common theoretical foundations, methodologies, or research perspectives. The size of each node represents the relative influence of a cited reference, while the connections and total link strength indicate the degree of intellectual relationship between references.

Co-word Analysis

Co-word analysis was performed to examine the conceptual and thematic structure of Quality Assurance research through the co-occurrence of keywords across publications. Author keywords and other relevant keyword information contained in the Scopus records were analyzed using VOSviewer to identify frequently occurring terms and determine how different research concepts are interconnected. Keywords appearing together across multiple publications were grouped into thematic clusters, allowing the identification of established and emerging areas such as software testing, quality assurance, test automation, artificial intelligence, machine learning, Large Language Models, defect prediction, DevOps, software quality, and quality engineering. The frequency of keyword occurrence indicates the prominence of a research topic, while total link strength reflects the extent to which a keyword is connected with other concepts within the literature.

© 2026 Advancing SDG Impact. All rights reserved.

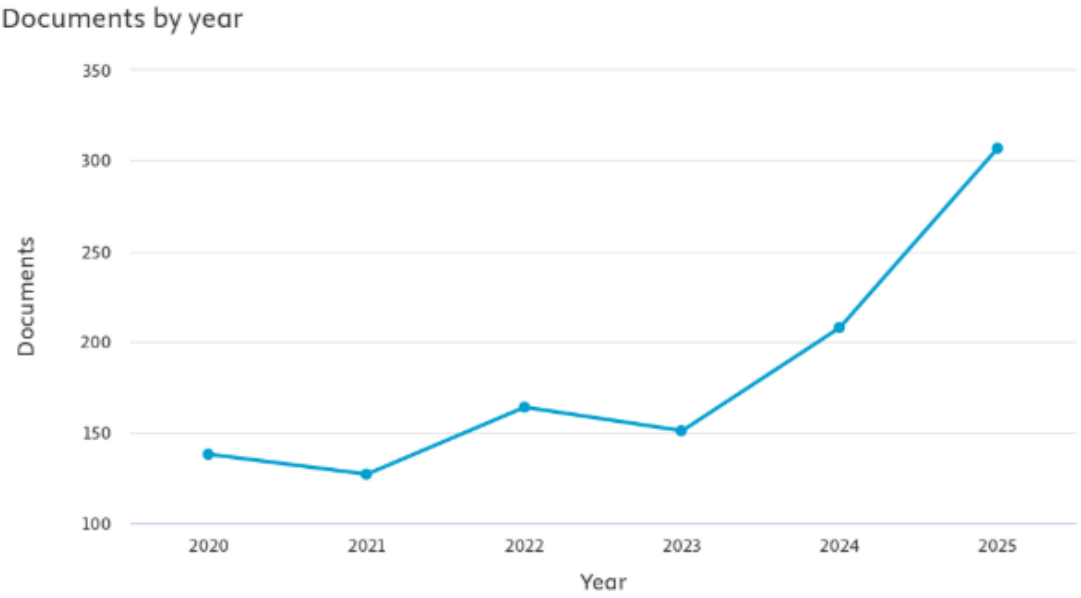
JASDG
JOURNAL FOR THE ADVANCEMENT OF SUSTAINABLE DEVELOPMENT GOALS
Advancing Knowledge. Inspiring Innovation. Achieving the SDGs.
AN INTERNATIONAL, PEER-REVIEWED, OPEN ACCESS JOURNAL
www.jasdg.com
Volume 1 Issue 3 Pages 1221-1247

III. Results

This section presents the major patterns identified in the bibliometric dataset on Quality Assurance, Software Testing, Software Quality, and Quality Engineering. The results are organized according to publication trends and the intellectual and thematic structure of the research field, as revealed through citation analysis, co-citation analysis, and co-word analysis.

Figure 1 illustrates the annual distribution of scholarly publications related to Quality Assurance, Software Testing, Software Quality, and Quality Engineering from 2020 to 2025. Research output decreased slightly from approximately 138 publications in 2020 to 127 in 2021, before increasing to around 163 publications in 2022. A moderate decline occurred in 2023, with approximately 150 publications, followed by substantial growth to around 208 publications in 2024. The most significant increase occurred in 2025, reaching approximately 307 publications, representing the highest research output within the six-year period. Overall, the trend demonstrates a considerable increase in scholarly attention toward Software Quality Assurance in recent years, particularly from 2023 onward, indicating the growing importance of software quality, testing automation, quality engineering, and emerging technologies within contemporary software development.

Figure 1. ***Annual Publication Trend on Quality Assurance, Software Testing, and Software Quality (2020–2025)***



Citation Analysis

Table 2 presents the top 10 most cited documents identified in the citation analysis of research related to Quality Assurance, Software Testing, Software Quality, and Quality Engineering. The results demonstrate the scholarly influence of publications addressing both established software quality practices and emerging technological approaches, including automated testing, artificial intelligence, large language models, defect prediction, DevOps, and software quality assessment. Documents with higher citation counts represent studies that have received greater scholarly attention and have contributed more substantially to the development of the research area within the period examined.

© 2026 Advancing SDG Impact. All rights reserved.

JASDG
JOURNAL FOR THE ADVANCEMENT OF SUSTAINABLE DEVELOPMENT GOALS
Advancing Knowledge. Inspiring Innovation. Achieving the SDGs.
AN INTERNATIONAL, PEER-REVIEWED, OPEN ACCESS JOURNAL
www.jasdg.com
Volume 1 Issue 3 Pages 1221-1247

Table 2. ***Top 10 Document Citation Analysis***

Rank	Author(s)	Title	Citations
1	Thota, M. K., Shajin, F. H., & Rajesh, P.	Survey on Software Defect Prediction Techniques	323
2	Wang, S., Liu, T., Nam, J., & Tan, L.	Deep Semantic Feature Learning for Software Defect Prediction	286
3	Zonnenshain, A., & Kenett, R. S.	Quality 4.0—The Challenging Future of Quality Engineering	272
4	Tantithamthavorn, C., Hassan, A. E., & Matsumoto, K.	The Impact of Class Rebalancing Techniques on the Performance and Interpretation of Defect Prediction Models	236
5	Riccio, V., Jahangirova, G., Stocco, A., Humbatova, N., Weiss, M., & Tonella, P.	Testing Machine Learning Based Systems: A Systematic Mapping	231
6	Feng, Y., Shi, Q., Gao, X., Wan, J., Fang, C., & Chen, Z.	DeepGini: Prioritizing Massive Tests to Enhance the Robustness of Deep Neural Networks	226
7	Pornprasit, C., & Tantithamthavorn, C. K.	JITLine: A Simpler, Better, Faster, Finer-Grained Just-in-Time Defect Prediction	127
8	Jiarpakdee, J., Tantithamthavorn, C. K., Dam, H. K., & Grundy, J.	An Empirical Study of Model-Agnostic Techniques for Defect Prediction Models	119
9	Lu, J., Yu, L., Li, X., Yang, L., & Zuo, C.	LLaMA-Reviewer: Advancing Code Review Automation with Large Language Models Through Parameter-Efficient Fine-Tuning	118
10	Pornprasit, C., & Tantithamthavorn, C. K.	DeepLineDP: Towards a Deep Learning Approach for Line-Level Defect Prediction	104

Table 3 presents the top 10 publication sources identified through the citation analysis. IEEE Transactions on Software Engineering demonstrates the greatest citation influence, with 1,362 citations from 28 documents, followed by Empirical Software Engineering with 417 citations from 16 documents and the Proceedings of the International Conference on Software Engineering with 253 citations from 17 documents. In terms of publication volume, the ACM International Conference Proceeding Series contributed the largest number of documents with 39 publications, although its citation count of 106 is substantially lower than that of the leading journal sources. The results demonstrate that both established software engineering journals and conference proceedings contribute

© 2026 Advancing SDG Impact. All rights reserved.

JASDG
JOURNAL FOR THE ADVANCEMENT OF SUSTAINABLE DEVELOPMENT GOALS
Advancing Knowledge. Inspiring Innovation. Achieving the SDGs.
AN INTERNATIONAL, PEER-REVIEWED, OPEN ACCESS JOURNAL
www.jasdg.com
Volume 1 Issue 3 Pages 1221-1247

significantly to the dissemination of Quality Assurance research, while their citation impact varies considerably across publication venues.

Table 3. ***Top 10 Sources — Citation Analysis***

Rank	Publication / Source	Documents
1	IEEE Transactions on Software Engineering	28
2	Empirical Software Engineering	16
3	Proceedings – International Conference on Software Engineering	17
4	IEEE Access	14
5	ACM International Conference Proceeding Series	39
6	Communications in Computer and Information Science	27
7	Lecture Notes in Computer Science	18
8	Lecture Notes in Networks and Systems	29
9	CEUR Workshop Proceedings	22
10	Proceedings of SPIE — The International Society for Optical Engineering	11

Table 4 presents the top 10 authors identified through citation analysis. Chakkrit Tantithamthavorn ranked first with 636 citations across 11 documents, indicating the strongest overall citation influence among the authors included in the analysis. Jirayus Jiarpakdee ranked second with 412 citations from six documents, followed by Patanamon Thongtanunam with 296 citations from eight documents. Other influential contributors include Meng Yan, Zhiqiang Li, Xiao-Yuan Jing, Michael Felderer, Foutse Khomh, and David Lo, demonstrating that research influence is distributed among scholars working across different areas of software quality and testing.

Table 4. ***Top 10 Authors — Citation Analysis***

Rank	Author	Documents
1	Tantithamthavorn, Chakkrit	11
2	Jiarpakdee, Jirayus	6
3	Thongtanunam, Patanamon	8
4	Yan, Meng	8

© 2026 Advancing SDG Impact. All rights reserved.

JASDG
JOURNAL FOR THE ADVANCEMENT OF SUSTAINABLE DEVELOPMENT GOALS
Advancing Knowledge. Inspiring Innovation. Achieving the SDGs.
AN INTERNATIONAL, PEER-REVIEWED, OPEN ACCESS JOURNAL
www.jasdg.com
Volume 1 Issue 3 Pages 1221-1247

Rank	Author	Documents
5	Li, Zhiqiang	7
6	Jing, Xiao-Yuan	6
7	Felderer, Michael	7
8	Khomh, Foutse	6
9	Lo, David	7
10	Staegemann, Daniel	9

Table 5 presents the top 10 countries contributing to research on Quality Assurance, Software Testing, Software Quality, and Quality Engineering, based on publication output and citation performance. China ranked first with 187 documents and 1,608 citations, followed by India with 162 documents and 998 citations, and the United States with 157 documents and 878 citations. Although Australia ranked seventh in publication output with only 50 documents, it accumulated 1,592 citations, indicating a particularly high level of scholarly impact relative to its number of publications. Canada and Japan also demonstrated substantial citation influence, recording 1,036 and 831 citations, respectively. Overall, the results indicate that Software Quality Assurance research has a broad international presence, with Asian, North American, European, and Australian researchers making substantial contributions to the development of the field.

Table 5. ***Top 10 Countries — Citation Analysis***

Rank	Country	Documents	Citations
1	China	187	1,608
2	India	162	998
3	United States	157	878
4	Germany	99	591
5	Canada	53	1,036
6	Japan	51	831
7	Australia	50	1,592
8	Brazil	41	200
9	Sweden	35	251

© 2026 Advancing SDG Impact. All rights reserved.

JASDG
JOURNAL FOR THE ADVANCEMENT OF SUSTAINABLE DEVELOPMENT GOALS
Advancing Knowledge. Inspiring Innovation. Achieving the SDGs.
AN INTERNATIONAL, PEER-REVIEWED, OPEN ACCESS JOURNAL
www.jasdg.com
Volume 1 Issue 3 Pages 1221-1247

Rank	Country	Documents	Citations
10	Austria	34	283

Co-citation Analysis

Table 6 presents the top 10 documents with the highest co-citation frequency and total link strength identified through the VOSviewer analysis. The Art of Software Testing recorded the highest co-citation frequency with 17 citations, followed by Experimentation in Software Engineering with 16 citations and Introduction to Software Testing with 12 citations, demonstrating the continuing influence of foundational software testing literature. In terms of network connectivity, the studies associated with static-code defect prediction and empirical validation of defect prediction models achieved the highest total link strength of 13, indicating strong relationships with other cited references in the network.

Table 6. ***Top 10 Documents with the Highest Co-citation and Total Link Strength***

Document	Citations	Total Link Strength
Badgett, T. (2011). The Art of Software Testing.	17	5
Wesslén, A. (2012). Experimentation in Software Engineering.	16	2
Offutt, J. (2016). Introduction to Software Testing.	12	2
Pietsch, S. (2008). Benchmarking Classification Models for Software Defect Prediction: A Proposed Framework and Novel Findings.	11	12
Counsell, S. (2012). A Systematic Literature Review on Fault Prediction Performance in Software Engineering.	11	7
Frank, A. (2007). Data Mining Static Code Attributes to Learn Defect Predictors.	10	13
Malhotra, R. (2015). A Systematic Review of Machine Learning Techniques for Software Fault Prediction.	10	5
Zhu, L. (2017). Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices.	10	1
Matsumoto, K. (2017). An Empirical Comparison of Model Validation Techniques for Defect Prediction Models.	9	13
Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-Sampling Technique.	9	11

© 2026 Advancing SDG Impact. All rights reserved.

JASDG
JOURNAL FOR THE ADVANCEMENT OF SUSTAINABLE DEVELOPMENT GOALS
Advancing Knowledge. Inspiring Innovation. Achieving the SDGs.
AN INTERNATIONAL, PEER-REVIEWED, OPEN ACCESS JOURNAL
www.jasdg.com
Volume 1 Issue 3 Pages 1221-1247

Source: Author interpretation based on VOSviewer analysis

Figure 2 presents the co-citation network of research related to Software Quality Assurance, Software Testing, Software Quality, and Quality Engineering. The visualization reveals several interconnected clusters represented by different colors, with each cluster grouping references that have stronger co-citation relationships and potentially share similar theoretical, methodological, or research foundations. The network indicates that Software Quality Assurance research is supported by several interconnected areas, particularly software defect prediction, software testing methodologies, machine learning techniques, quality engineering, and empirical software engineering.

Figure 2. ***Co-citation Network of Quality Assurance Research***

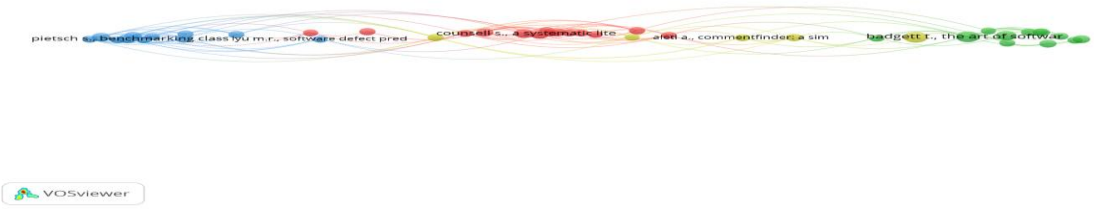


Table 7 presents the five major co-citation clusters identified through the VOSviewer analysis. A total of 43 co-cited references were distributed across five clusters, with each cluster representing a group of publications that demonstrate relatively strong intellectual relationships through their repeated co-citation within the analyzed literature. Cluster 1 (Red) represents the largest cluster with 13 references and emphasizes software testing, test automation, and continuous quality practices, while Cluster 2 (Green) contains 12 references focusing primarily on machine learning-based software defect prediction. Cluster 3 (Blue) consists of nine references addressing the methodological aspects of defect prediction, including data quality, classification, class imbalance, and model validation. Cluster 4 (Yellow) contains five references connecting code review and empirical software engineering with intelligent automation, while Cluster 5 (Purple) consists of four references representing foundational research in software measurement and defect prediction.

Table 7. ***Co-citation Clusters on Quality Assurance***

Cluster	Theme	Count	Representative Publications
1 (Red)	Software Testing, Test Automation, and Continuous Quality	13	Badgett (2011); Offutt (2016); Zhu et al. (2017); Fraser & Arcuri (2011); Barr et al. (2015); and related studies on software testing, test automation, test generation, and continuous quality
2 (Green)	Machine Learning-Based Software Defect Prediction	12	Counsell et al. (2012). Malhotra (2015) and related studies on machine learning, classification, feature selection, ensemble learning, software metrics, and defect prediction

© 2026 Advancing SDG Impact. All rights reserved.

JASDG
JOURNAL FOR THE ADVANCEMENT OF SUSTAINABLE DEVELOPMENT GOALS
Advancing Knowledge. Inspiring Innovation. Achieving the SDGs.
AN INTERNATIONAL, PEER-REVIEWED, OPEN ACCESS JOURNAL
www.jasdg.com
Volume 1 Issue 3 Pages 1221-1247

Cluster	Theme	Count	Representative Publications
3 (Blue)	Defect Prediction Models, Data Quality, and Model Validation	9	Pietsch et al. (2008); Frank et al. (2007); Matsumoto et al. (2017); Chawla et al. (2002); and related studies on classification benchmarking, defect datasets, class imbalance, and model validation
4 (Yellow)	Code Review, Intelligent Automation, and Empirical Software Engineering	5	Wesslén et al. (2012); CommentFinder-related research; Kingma & Ba (2015); and related studies on code review, intelligent software analysis, machine learning, and empirical software engineering
5 (Purple)	Software Metrics and Defect Prediction Foundations	4	Halstead (1977); studies on object-oriented software metrics; defect prediction reviews; and machine learning-based defect prediction research

Cluster 1 (Red): Software Testing, Test Automation, and Continuous Quality

This cluster contains 13 references and represents the largest intellectual cluster identified in the co-citation network. Its primary focus is the development and application of systematic approaches to software testing, including fundamental testing principles, automated test generation, test adequacy, continuous testing, and the integration of Quality Assurance activities throughout the software development lifecycle. The size of this cluster indicates that software testing continues to serve as one of the central intellectual foundations of Software Quality Assurance research, even as newer technologies and methodologies are incorporated into the field. Badgett (2011) and Offutt (2016) represent important testing-oriented references within this intellectual structure, providing concepts that continue to support subsequent research into more advanced testing techniques. The cluster also demonstrates the increasing importance of test automation, reflecting efforts to reduce the manual effort required to design, execute, and maintain test cases as software applications increase in size, complexity, and release frequency. Another important aspect represented within this cluster is the relationship between testing and continuous software development practices: research related to continuous integration, delivery, and deployment demonstrates that software testing is increasingly performed throughout the development process rather than only at the end of implementation, enabling earlier defect identification and faster feedback to developers.

Cluster 2 (Green): Machine Learning-Based Software Defect Prediction

This cluster contains 12 references and represents one of the most prominent data-driven research streams within the co-citation network. Its central concern is the use of historical software data, software metrics, and computational learning techniques to predict which components of a software system are more likely to contain defects—an important development because it enables organizations to move from primarily reactive defect detection toward more proactive and predictive quality management. Counsell et al. (2012) contribute an important examination of fault prediction performance, providing an intellectual foundation for evaluating the accuracy and reliability of predictive Quality Assurance approaches, while Malhotra (2015) examines machine learning techniques for software fault prediction based on characteristics such as code complexity, source-code changes, development history, and previous defect information. The cluster further incorporates research on classification algorithms, feature selection, ensemble learning, and software metrics, demonstrating the increasing sophistication of defect prediction research. From a practical Quality Assurance perspective, defect prediction can

© 2026 Advancing SDG Impact. All rights reserved.

JASDG
JOURNAL FOR THE ADVANCEMENT OF SUSTAINABLE DEVELOPMENT GOALS
Advancing Knowledge. Inspiring Innovation. Achieving the SDGs.
AN INTERNATIONAL, PEER-REVIEWED, OPEN ACCESS JOURNAL
www.jasdg.com
Volume 1 Issue 3 Pages 1221-1247

help testing teams prioritize limited resources by identifying modules that require greater attention, potentially improving test planning, risk-based testing, regression testing, and resource allocation.

Cluster 3 (Blue): Defect Prediction Models, Data Quality, and Model Validation

This cluster contains nine references and focuses on the methodological reliability of predictive models used in Software Quality Assurance. While Cluster 2 emphasizes the use of machine learning for defect prediction, Cluster 3 concentrates more specifically on whether these predictive models are constructed, evaluated, and validated appropriately—an important distinction, since a sophisticated algorithm does not necessarily produce reliable results when the underlying dataset, evaluation procedure, or experimental design contains weaknesses. Pietsch et al. (2008) contribute research on benchmarking classification models, enabling comparison of predictive techniques under common experimental conditions, while Frank et al. (2007) examine the use of static code attributes for learning defect predictors, demonstrating how traditional software measurement can be integrated with machine learning. Matsumoto et al. (2017) contribute a methodological perspective on model-validation techniques, which is necessary to determine whether a predictive model can generalize beyond the data on which it was developed. The cluster also contains Chawla et al. (2002) and the SMOTE technique, reflecting the importance of addressing class imbalance, since software defect datasets frequently contain considerably more non-defective than defective components.

Cluster 4 (Yellow): Code Review, Intelligent Automation, and Empirical Software Engineering

This cluster contains five references and represents a research stream connecting traditional software review practices with modern computational and empirical approaches. Although smaller than the first three clusters, its position within the co-citation network indicates an important intellectual relationship between empirical software engineering methods, code review, and intelligent Quality Assurance technologies. Empirical software engineering emphasizes the systematic use of experiments, observations, and data to evaluate software engineering practices, providing methodological foundations for determining whether proposed testing and QA techniques produce measurable, reproducible improvements. The cluster also contains research on modern code review, an important QA activity in which developers or reviewers examine source-code changes to identify defects, maintainability problems, security concerns, and deviations from development standards; the presence of CommentFinder-related research illustrates the movement toward intelligent assistance in review activities. The inclusion of methodological machine-learning research, including Kingma and Ba (2015), further illustrates the relationship between contemporary AI techniques and software engineering applications such as defect prediction and semantic software analysis.

Cluster 5 (Purple): Software Metrics and Defect Prediction Foundations

This cluster contains four references and represents the quantitative measurement foundations underlying several contemporary Software Quality Assurance approaches. Although it is the smallest cluster identified in the analysis, its publications provide concepts that support the measurement, assessment, and prediction of software quality. Halstead (1977) represents an important foundation through the development of quantitative software metrics, which are important because QA decisions become more systematic when software characteristics can be quantified rather than evaluated entirely through subjective judgment. The cluster also includes research on object-oriented software metrics, reflecting the evolution of software measurement as programming paradigms have changed; measures relating to complexity, coupling, cohesion, and inheritance provide insight into maintainability and defect-proneness and have subsequently become important features in many defect prediction models. A particularly significant relationship within this cluster is the connection between traditional software metrics and machine learning-based defect prediction: modern prediction algorithms frequently depend on measurable software characteristics as input variables, meaning that AI-based Quality Assurance often builds upon rather than replaces earlier software measurement research.

© 2026 Advancing SDG Impact. All rights reserved.

JASDG
JOURNAL FOR THE ADVANCEMENT OF SUSTAINABLE DEVELOPMENT GOALS
Advancing Knowledge. Inspiring Innovation. Achieving the SDGs.
AN INTERNATIONAL, PEER-REVIEWED, OPEN ACCESS JOURNAL
www.jasdg.com
Volume 1 Issue 3 Pages 1221-1247

Co-word Analysis

Table 8 presents the top 15 keywords identified through the co-word analysis. Among the 83 keywords included in the VOSviewer network, "software quality assurance" recorded the highest frequency with 171 occurrences and a total link strength of 209, demonstrating its central position within the analyzed research landscape. This was closely followed by "quality assurance" with 170 occurrences and a total link strength of 272, which recorded the highest total link strength among the top keywords, indicating particularly strong connections with other research concepts. "Software testing" ranked third with 149 occurrences and a total link strength of 225, further demonstrating that testing remains one of the fundamental areas within Software Quality Assurance research. Emerging technological topics are also strongly represented, particularly machine learning (98 occurrences), deep learning (50), artificial intelligence (34), and large language models (25), indicating the increasing integration of intelligent technologies into software quality activities.

Table 8. ***Top 15 Keywords in the Co-occurrence Analysis***

Rank	Keyword	Occurrences	Total Link Strength
1	Software Quality Assurance	171	209
2	Quality Assurance	170	272
3	Software Testing	149	225
4	Machine Learning	98	199
5	Software Quality	87	116
6	Software Defect Prediction	72	90
7	Software Engineering	52	111
8	Deep Learning	50	83
9	Test Automation	35	54
10	Artificial Intelligence	34	87
11	Testing	32	70
12	Defect Prediction	26	35
13	Large Language Models	25	32
14	Continuous Integration	21	44
15	Test Case Generation	19	25

© 2026 Advancing SDG Impact. All rights reserved.

JASDG
JOURNAL FOR THE ADVANCEMENT OF SUSTAINABLE DEVELOPMENT GOALS
Advancing Knowledge. Inspiring Innovation. Achieving the SDGs.
AN INTERNATIONAL, PEER-REVIEWED, OPEN ACCESS JOURNAL
www.jasdg.com
Volume 1 Issue 3 Pages 1221-1247

Figure 3 presents the co-word network visualization of Software Quality Assurance research, illustrating the relationships among frequently occurring keywords identified in the analyzed literature. In the VOSviewer network, the size of each node represents the prominence or frequency of a keyword, while the connecting lines indicate co-occurrence relationships between keywords appearing together in publications. Different colors represent clusters of keywords that form related research themes, allowing the major conceptual areas within Software Quality Assurance research to be identified.

Figure 3. ***Co-word Network of Quality Assurance Research***

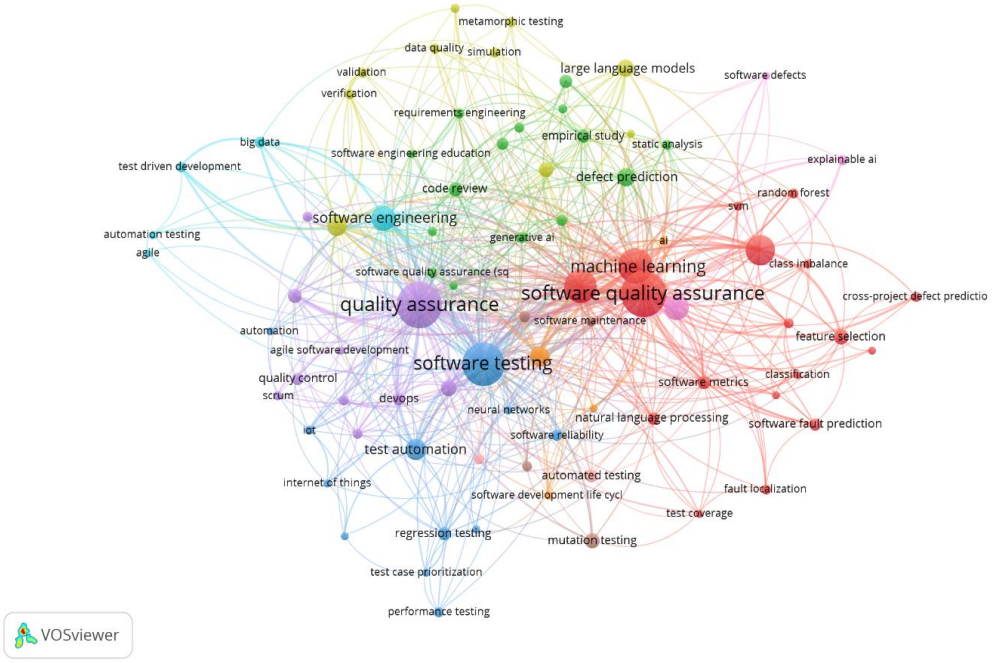


Table 9 presents the results of the co-word analysis, identifying 83 keywords distributed across 10 thematic clusters in Software Quality Assurance research. The clustering demonstrates the multidimensional nature of the field, combining established areas such as software testing, quality assurance, software development, and quality management with emerging areas involving machine learning, deep learning, generative AI, Large Language Models, and reinforcement learning. Cluster 1 is the largest with 18 keywords, followed by Cluster 2 with 15 keywords and Cluster 3 with 12 keywords, indicating particularly strong research concentration around software defect prediction, intelligent software engineering, and software testing. Clusters 4 and 5 each contain 10 keywords and highlight the growing importance of AI-assisted testing, test case generation, DevOps, continuous integration, and Agile Quality Assurance.

Table 9. ***Co-word Analysis Clusters on Quality Assurance***

Cluster	Theme	Keywords	Representative Terms
1 (Red)	Software Quality Assurance and Machine Learning-Based Defect Prediction	18	software quality assurance, machine learning, software quality, software defect prediction, feature selection, software metrics, natural

© 2026 Advancing SDG Impact. All rights reserved.

JASDG
JOURNAL FOR THE ADVANCEMENT OF SUSTAINABLE DEVELOPMENT GOALS
Advancing Knowledge. Inspiring Innovation. Achieving the SDGs.
AN INTERNATIONAL, PEER-REVIEWED, OPEN ACCESS JOURNAL
www.jasdg.com
Volume 1 Issue 3 Pages 1221-1247

Cluster	Theme	Keywords	Representative Terms
			language processing, software fault prediction, ensemble learning, random forest
2 (Green)	Intelligent Software Engineering, Code Review, and Generative AI	15	defect prediction, code review, empirical study, modern code review, generative AI, large language models (LLMs), static analysis, mining software repositories, requirements engineering
3 (Blue)	Software Testing, Test Automation, and Reliability Engineering	12	software testing, test automation, regression testing, software reliability, performance testing, IoT, automation, neural networks, model-based testing, test case prioritization
4 (Yellow)	AI-Assisted Testing, Test Case Generation, and Software Verification	10	testing, large language models, test case generation, verification, validation, data quality, simulation, metamorphic testing, quantum computing, quality
5 (Violet)	Quality Assurance, DevOps, and Agile Quality Management	10	quality assurance, continuous integration, software development, DevOps, quality control, quality management, security, Agile development, Scrum

Cluster 1 (Red): Software Quality Assurance and Machine Learning-Based Defect Prediction

This is the largest thematic cluster, containing 18 keywords related primarily to Software Quality Assurance, machine learning, software quality, software defect prediction, feature selection, software metrics, and fault prediction. The prominence of these terms indicates that predictive approaches have become an important component of contemporary Software Quality Assurance research, enabling researchers to analyze historical software data and metrics to identify patterns associated with defective components. Hai et al. (2022), for example, demonstrated the application of deep learning to cloud-based bug-tracking and software defect analysis, while Parija et al. (2025) proposed a stacking ensemble framework with advanced feature engineering for software defect prediction. The cluster, therefore, suggests that Software Quality Assurance is increasingly moving beyond reactive defect identification toward predictive and data-driven quality management (Hai et al., 2022; Parija et al., 2025).

Cluster 2 (Green): Intelligent Software Engineering, Code Review, and Generative AI

This cluster contains 15 keywords associated with intelligent software engineering, including defect prediction, code review, empirical studies, generative AI, Large Language Models, static analysis, requirements engineering, and mining software repositories. This cluster demonstrates the growing integration of AI technologies into software engineering activities that traditionally depended heavily on human analysis and expertise. Eagal et al. (2025) examined the dependability of LLMs for code clone generation, demonstrating both the potential and the reliability concerns associated with AI-generated software artifacts, while Esposito et al. (2025) examined relationships between architectural smells and static analysis warnings, emphasizing the continuing importance

© 2026 Advancing SDG Impact. All rights reserved.

JASDG
JOURNAL FOR THE ADVANCEMENT OF SUSTAINABLE DEVELOPMENT GOALS
Advancing Knowledge. Inspiring Innovation. Achieving the SDGs.
AN INTERNATIONAL, PEER-REVIEWED, OPEN ACCESS JOURNAL
www.jasdg.com
Volume 1 Issue 3 Pages 1221-1247

of automated source-code and architecture analysis. Together, these developments indicate that modern QA increasingly integrates static analysis, empirical software engineering, and generative AI techniques (Eagal et al., 2025; Esposito et al., 2025).

Cluster 3 (Blue): Software Testing, Test Automation, and Reliability Engineering

This cluster consists of 12 keywords focusing on software testing, test automation, regression testing, software reliability, performance testing, model-based testing, and test case prioritization. Software testing remains one of the primary mechanisms through which software quality is verified and defects are detected before deployment. Alshamaa and Saleem (2025) emphasize the continuing relevance of established software testing techniques, particularly black-box testing, while Parveen et al. (2007) demonstrate the importance of effective test management in organizing and controlling testing activities. This cluster demonstrates that traditional software testing remains fundamental while increasingly being combined with automation, prioritization, and reliability-oriented approaches (Parveen et al., 2007; Alshamaa & Saleem, 2025).

Cluster 4 (Yellow): AI-Assisted Testing, Test Case Generation, and Software Verification

This cluster contains 10 keywords and represents a rapidly developing research direction involving testing, Large Language Models, test case generation, verification, validation, simulation, and data quality. Antonio et al. (2025) investigated prompt-orchestrated LLM workflows for automated test case generation in Agile environments, while Antal et al. (2025) examined GPT-4 for vulnerability-witnessing unit test generation. These studies illustrate how generative AI may reduce manual effort in test design while enabling the generation of additional testing scenarios, reflecting the emergence of AI-assisted software verification and automated test generation as an important direction within contemporary QA research (Antal et al., 2025; Antonio et al., 2025).

Cluster 5 (Violet): Quality Assurance, DevOps, and Agile Quality Management

This cluster contains 10 keywords centered on quality assurance, continuous integration, DevOps, software development, quality control, quality management, Agile development, and Scrum. The relationship among these keywords illustrates how Quality Assurance has become increasingly integrated into the entire software development lifecycle, with testing and quality activities performed continuously rather than postponed until the final stages of development. Rafi et al. (2025) examined software product quality within DevOps from an ISO 25010:2023 perspective, while Olcay et al. (2025) examined corporate QA test processes. This cluster represents the transition from isolated testing toward continuous and organizationally integrated Quality Assurance (Rafi et al., 2025; Olcay et al., 2025).

© 2026 Advancing SDG Impact. All rights reserved.

JASDG
JOURNAL FOR THE ADVANCEMENT OF SUSTAINABLE DEVELOPMENT GOALS
Advancing Knowledge. Inspiring Innovation. Achieving the SDGs.
AN INTERNATIONAL, PEER-REVIEWED, OPEN ACCESS JOURNAL
www.jasdg.com
Volume 1 Issue 3 Pages 1221-1247

IV. Discussion

The results of this bibliometric analysis provide an overview of the evolving research landscape in Software Quality Assurance, Software Testing, Software Quality, and Quality Engineering. The findings indicate that Software Quality Assurance has developed beyond conventional software testing and defect detection toward a broader discipline involving test automation, software analytics, machine learning, continuous integration, DevOps, and artificial intelligence. The citation, co-citation, and co-word analyses demonstrate that traditional QA concepts remain strongly represented within the literature while increasingly interacting with emerging intelligent and data-driven technologies. This development reflects the growing complexity of modern software systems and the need for organizations to ensure software reliability, maintainability, performance, and overall quality throughout increasingly rapid development cycles, consistent with prior research on the continuing importance of systematic test management and software quality analytics in supporting effective quality-related decisions (Parveen et al., 2007; Vollmer et al., 2019; López et al., 2021).

One of the most significant themes identified in the analysis is the growing relationship between Software Quality Assurance and machine learning-based software defect prediction. The co-word analysis shows that terms such as machine learning, software defect prediction, defect prediction, deep learning, feature selection, and software metrics occupy important positions within the research landscape, suggesting that researchers are increasingly exploring predictive approaches capable of identifying software components that are more likely to contain defects before failures occur during testing or production. Hai et al. (2022) demonstrated the application of deep learning to software defect analysis using bug-tracking information, while Parija et al. (2025) investigated an ensemble-based framework incorporating advanced feature engineering for improving software defect prediction. These developments indicate a transition from predominantly reactive QA practices toward predictive and risk-oriented Quality Assurance, where historical software data can support more informed testing decisions and resource allocation.

Another important finding concerns the continued significance of software testing and test automation. Software testing remains one of the dominant themes within the co-word network and is strongly connected with regression testing, test automation, model-based testing, test case prioritization, performance testing, and continuous integration, indicating that although intelligent technologies are becoming increasingly important, systematic software testing continues to provide a fundamental foundation for Quality Assurance. Alshamaa and Saleem (2025) emphasize the continuing importance of established software testing techniques, while Parveen et al. (2007) demonstrate the importance of effective test management. The increasing connection between software testing and automation further reflects the demands of modern development environments, where frequent software changes require tests to be executed rapidly and repeatedly.

The results also reveal the growing influence of Artificial Intelligence and Large Language Models (LLMs) in Software Quality Assurance research. Keywords related to artificial intelligence, generative AI, Large Language Models, test case generation, natural language processing, and intelligent software engineering appear within the co-word network, demonstrating the emergence of AI-assisted Quality Assurance as an important research direction. Antonio et al. (2025) investigated prompt-orchestrated LLM workflows for automated test case generation in Agile environments, while Antal et al. (2025) explored the use of GPT-4 for generating vulnerability-witnessing unit tests. Yang (2025) further demonstrated the potential of reinforcement learning and open-source LLMs for reliable bug report generation. Together, these studies indicate that generative AI is beginning to support multiple QA activities, including test design, defect documentation, code analysis, and vulnerability-oriented testing.

The co-citation analysis further demonstrates that contemporary Software Quality Assurance research is grounded in several interconnected intellectual foundations, encompassing software testing and automation, machine

© 2026 Advancing SDG Impact. All rights reserved.

JASDG
JOURNAL FOR THE ADVANCEMENT OF SUSTAINABLE DEVELOPMENT GOALS
Advancing Knowledge. Inspiring Innovation. Achieving the SDGs.
AN INTERNATIONAL, PEER-REVIEWED, OPEN ACCESS JOURNAL
www.jasdg.com
Volume 1 Issue 3 Pages 1221-1247

learning-based defect prediction, model validation and data quality, intelligent software analysis, and software metrics. The prominence of defect prediction literature indicates that predictive software quality represents an important intellectual component of the field, while the continued presence of foundational testing and software measurement research demonstrates that newer intelligent techniques build upon established software engineering principles. The relationships among these clusters suggest that modern QA research does not represent a complete replacement of traditional methodologies but rather their integration with automated, analytical, and intelligent techniques.

Another important development reflected in the results is the integration of QA with Agile and DevOps practices. Keywords including continuous integration, DevOps, Agile software development, Scrum, and quality management indicate that software quality is increasingly considered throughout the software development lifecycle rather than only during a dedicated testing phase. Rafi et al. (2025) examined software product quality within DevOps using the ISO 25010:2023 perspective, while Olcay et al. (2025) examined corporate QA testing processes. These studies support the broader finding that QA is increasingly evolving toward continuous Quality Engineering, where development, testing, deployment, monitoring, and quality management activities are closely integrated. Overall, the bibliometric findings demonstrate that Software Quality Assurance is becoming increasingly multidisciplinary: traditional areas such as test management, software testing, defect tracking, software metrics, and quality management continue to provide important foundations, while emerging research increasingly incorporates machine learning, deep learning, predictive analytics, test automation, generative AI, and Large Language Models (Sultan et al., 2013; Hai et al., 2022; Antal et al., 2025; Antonio et al., 2025). The integration of these themes suggests that the future of Software Quality Assurance is likely to involve greater collaboration between human QA professionals and intelligent technologies, with automation and AI supporting rather than entirely replacing established quality engineering practices.

Implications

The findings of this bibliometric analysis provide several important implications for Software Quality Assurance researchers, QA practitioners, software developers, test managers, and organizations involved in software development. First, the results demonstrate that Quality Assurance should increasingly be viewed as a continuous activity integrated throughout the Software Development Life Cycle rather than as a separate testing phase performed only before software release. The strong presence of software testing, continuous integration, DevOps, Agile development, and test automation within the identified research themes supports the adoption of continuous quality practices in which testing and quality evaluation occur throughout development, enabling organizations to identify defects earlier, provide faster feedback to development teams, and reduce the risks associated with discovering critical problems late in the development lifecycle (Rafi et al., 2025; Olcay et al., 2025).

The findings also emphasize the potential importance of machine learning and predictive analytics in Quality Assurance. Software defect prediction can help QA teams identify components that may require greater testing attention based on software metrics, historical defects, and other development information, complementing conventional testing by enabling more risk-based allocation of testing resources, particularly for large and complex software systems where exhaustive testing may be impractical (Hai et al., 2022; Parija et al., 2025).

Another important implication involves the increasing application of Artificial Intelligence and Large Language Models to QA activities. Recent research demonstrates opportunities for AI-assisted test case generation, unit test generation, bug report generation, and other software engineering activities (Antal et al., 2025; Antonio et al., 2025; Yang, 2025), which may help reduce repetitive manual activities and support QA professionals in generating, analyzing, and maintaining testing artifacts. However, the dependability of AI-generated software artifacts remains an important consideration, as demonstrated by research examining the reliability of LLM-generated outputs (Eagal et al., 2025). Organizations adopting generative AI for QA should therefore maintain

© 2026 Advancing SDG Impact. All rights reserved.

appropriate human review, validation, traceability, and quality controls rather than assuming that AI-generated results are automatically correct.

The findings also have implications for QA workforce development. As Quality Assurance becomes increasingly automated and data-driven, QA professionals may require competencies beyond conventional manual testing, including test automation, API testing, continuous integration, software analytics, machine learning concepts, and effective use of AI-assisted tools, while foundational knowledge of test design, defect management, requirements analysis, and software quality principles remains important. Finally, this study contributes to Software Quality Assurance research by mapping the relationships among established and emerging research areas, providing opportunities for future interdisciplinary studies examining how intelligent technologies can improve QA efficiency without compromising reliability, explainability, maintainability, and human oversight.

Conclusion

This study presents a bibliometric analysis of research on Quality Assurance, Software Testing, Software Quality, and Quality Engineering, providing an overview of the intellectual structure, influential contributors, and major thematic developments within the field. Using bibliographic records retrieved from the Scopus database and analyzed through VOSviewer, the study applied citation analysis, co-citation analysis, and co-word analysis to examine publication patterns and relationships within the scholarly literature. The results demonstrate that Software Quality Assurance has developed into a broad and increasingly multidisciplinary research field that combines traditional testing and quality management principles with emerging approaches involving automation, predictive analytics, machine learning, deep learning, DevOps, and Artificial Intelligence.

The findings indicate that established concepts such as software quality assurance, quality assurance, and software testing remain among the most prominent themes within the research landscape. At the same time, machine learning and software defect prediction demonstrate substantial thematic importance, indicating increasing scholarly attention toward predictive approaches to software quality. The co-citation results further reveal intellectual foundations associated with software testing, defect prediction, software metrics, data quality, and model validation. Emerging keywords involving Large Language Models, generative AI, explainable AI, reinforcement learning, and automated test case generation suggest that intelligent Quality Assurance represents a developing research direction with significant potential for future investigation. Taken together, the study demonstrates that Software Quality Assurance is evolving from a predominantly reactive discipline toward a more continuous, automated, predictive, and intelligent Quality Engineering approach, with traditional testing remaining essential but increasingly complemented by automation, software analytics, machine learning, and AI-assisted techniques, integrated with Agile development, DevOps, and continuous integration throughout the development lifecycle.

Limitations and Future Research Directions

Several limitations should be considered when interpreting the findings of this study. First, the analysis relies primarily on Scopus-indexed publications, meaning that relevant literature indexed exclusively in other academic databases may not be represented. Second, the results are influenced by the specific search terms, inclusion criteria, publication period, and bibliographic metadata used to construct the dataset; different search strategies may therefore produce different publication and thematic patterns. Third, bibliometric measures such as citation counts and total link strength provide quantitative indicators of scholarly relationships but do not necessarily represent the practical importance, methodological quality, or real-world effectiveness of individual studies. Finally, keyword variations, synonymous terminology, author-selected keywords, and differences in bibliographic metadata may influence the clustering produced by VOSviewer. Future studies may address these limitations by incorporating additional databases, applying systematic keyword normalization, extending the publication period, and combining bibliometric methods with systematic literature review or qualitative content analysis.

© 2026 Advancing SDG Impact. All rights reserved.

JASDG
JOURNAL FOR THE ADVANCEMENT OF SUSTAINABLE DEVELOPMENT GOALS
Advancing Knowledge. Inspiring Innovation. Achieving the SDGs.
AN INTERNATIONAL, PEER-REVIEWED, OPEN ACCESS JOURNAL
www.jasdg.com
Volume 1 Issue 3 Pages 1221-1247

Recommendations

Based on the findings of this bibliometric analysis, future Software Quality Assurance research should continue investigating the integration of Artificial Intelligence, machine learning, deep learning, and Large Language Models with established testing and quality engineering practices. Particular attention should be given to evaluating the reliability, accuracy, explainability, and practical usefulness of AI-generated test cases, defect reports, code analyses, and other testing artifacts. Although generative AI provides significant opportunities for improving testing efficiency, future research should establish appropriate validation mechanisms and human oversight to ensure that AI-assisted QA activities produce dependable results.

Organizations are also encouraged to strengthen the integration of test automation, continuous integration, Agile development, and DevOps within their Quality Assurance processes. Testing should be incorporated throughout the software development lifecycle to support earlier defect detection and continuous feedback rather than being concentrated only before production release, while maintaining structured test management, defect management, quality metrics, and governance mechanisms to ensure that faster development processes do not compromise software quality.

Further research should investigate machine learning-based software defect prediction, particularly issues involving dataset quality, feature selection, model validation, class imbalance, explainability, and cross-project generalizability, with predictive models evaluated not only according to statistical performance but also according to their practical usefulness to QA teams in improving test prioritization, defect detection, resource allocation, and software release decisions in real-world development environments.

Future bibliometric studies should also expand the evidence base by incorporating databases beyond Scopus, extending the publication period where appropriate, and applying standardized keyword-cleaning and thesaurus procedures before conducting network analysis. Combining bibliometric analysis with systematic literature reviews, qualitative content analysis, case studies, and empirical industry research may provide a more comprehensive understanding of how emerging Quality Assurance techniques are actually applied in practice. Finally, researchers and practitioners should continue examining the relationship between human expertise and intelligent automation—rather than treating AI as a complete replacement for QA professionals, future approaches should investigate how testers, developers, and intelligent tools can work together effectively, with particular attention to human validation of AI-generated testing artifacts, transparency of automated decisions, maintainability of automated tests, data privacy, software security, and responsible use of AI.

Declarations

Ethics Declaration

Not applicable. This study is a bibliometric analysis based exclusively on secondary data retrieved from publicly available scientific databases. It does not involve human participants, animals, or any form of intervention requiring ethical approval.

Consent to Publish Declaration

Not applicable.

Consent to Participate Declaration

Not applicable.

Funding

This research received no external funding.

© 2026 Advancing SDG Impact. All rights reserved.

JASDG
JOURNAL FOR THE ADVANCEMENT OF SUSTAINABLE DEVELOPMENT GOALS
Advancing Knowledge. Inspiring Innovation. Achieving the SDGs.
AN INTERNATIONAL, PEER-REVIEWED, OPEN ACCESS JOURNAL
www.jasdg.com
Volume 1 Issue 3 Pages 1221-1247

Data Availability Statement

Not applicable. This study is a bibliometric analysis based exclusively on secondary data retrieved from publicly available scientific databases.

Acknowledgements

The authors express their gratitude to their colleagues and academic mentors who provided valuable insights into bibliometric research and software development education. Special thanks are extended to institutional staff and collaborators for their support during data collection and manuscript preparation.

Conflict of Interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

© 2026 Advancing SDG Impact. All rights reserved.

JASDG
JOURNAL FOR THE ADVANCEMENT OF SUSTAINABLE DEVELOPMENT GOALS
Advancing Knowledge. Inspiring Innovation. Achieving the SDGs.
AN INTERNATIONAL, PEER-REVIEWED, OPEN ACCESS JOURNAL
www.jasdg.com
Volume 1 Issue 3 Pages 1221-1247

References

Alshamaa, Z. M., & Saleem, N. N. (2025). Black box software testing techniques: A literature review. Passer Journal of Basic and Applied Sciences, 7(2), 1001–1011. https://doi.org/10.24271/psr.2025.508216.1963

Antal, G., Bán, D., Isztin, M., Ferenc, R., & Hegedűs, P. (2025). Leveraging GPT-4 for vulnerability-witnessing unit test generation. Proceedings of the 29th International Conference on Evaluation and Assessment in Software Engineering (EASE 2025), 1056–1065. https://doi.org/10.1145/3756681.3757075

Antonio, A. A. F., Alfredo, B. P., & Gustavo, S. G. R. (2025). Engineering prompt-orchestrated LLM workflows for automated test case generation in agile environments. International Journal of Advanced Computer Science and Applications, 16(12), 771–781. https://doi.org/10.14569/IJACSA.2025.0161272

Bhagwat, A. (2009, December 28). Software test case engineering: Treating test cases as a product. StickyMinds.

Bogorin-Predescu, A., Titu, S., Titu, A. M., Marcu, D. F., Dragomir, D. C., & Dragomir, M. (2025). Innovative cost-effective embedded system to enhance ECU firmware quality through remote hybrid testing in automotive domain. Applied Sciences (Switzerland), 15(23), Article 12736. https://doi.org/10.3390/app152312736

Donthu, N., Kumar, S., Mukherjee, D., Pandey, N., & Lim, W. M. (2021). How to conduct a bibliometric analysis: An overview and guidelines. Journal of Business Research, 133, 285–296.

Eagal, A., Stolee, K. T., & Ore, J.-P. (2025). Analyzing the dependability of large language models for code clone generation. Journal of Systems and Software, 230, Article 112548. https://doi.org/10.1016/j.jss.2025.112548

Esposito, M., Robredo, M., Arcelli Fontana, F., & Lenarduzzi, V. (2025). On the correlation between architectural smells and static analysis warnings. Software Quality Journal, 33(4), Article 33. https://doi.org/10.1007/s11219-025-09730-7

Hai, T., Zhou, J., Li, N., Jain, S. K., Agrawal, S., & Ben Dhaou, I. (2022). Cloud-based bug tracking software defects analysis using deep learning. Journal of Cloud Computing, 11, 32. https://doi.org/10.1186/s13677-022-00311-8

López, L., Manzano, M., Gómez, C., Oriol, M., Farré, C., Franch, X., Martínez-Fernández, S., & Vollmer, A. M. (2021). QaSD: A quality-aware strategic dashboard for supporting decision makers in agile software development. Science of Computer Programming.

Olcay, H. H., Belen, E., & Hacıoğlu, E. (2025). The review of test processes on corporate quality assurance (QA) at SeturTech. Proceedings of the 29th International Conference on Evaluation and Assessment in Software Engineering (EASE 2025), 826–829. https://doi.org/10.1145/3756681.3757008

Parija, S., Muduli, D., Shameem, M., & Sharma, S. K. (2025). Enhancing software defect prediction: A stacking ensemble framework with advanced feature engineering. Proceedings of the 29th International Conference on Evaluation and Assessment in Software Engineering Companion (EASE Companion 2025), 28–34. https://doi.org/10.1145/3727967.3756848

Parveen, T., Tilley, S., & Gonzalez, G. (2007). A case study in test management. Proceedings of the 45th Annual ACM Southeast Conference (ACM-SE 45), 82–87.

Rafi, S., Akbar, M. A., & Khan, A. A. (2025). Assessing software product quality in DevOps: An ISO 25010:2023 perspective. Proceedings of the 29th International Conference on Evaluation and Assessment in Software Engineering Companion (EASE Companion 2025), 10–16. https://doi.org/10.1145/3727967.3756847

© 2026 Advancing SDG Impact. All rights reserved.

Sultan, T., Khedr, A. E., & Sayed, M. (2013). A proposed defect tracking model for classifying the inserted defect reports to enhance software quality control. International Journal of Computer Applications, 67(14).

Vollmer, A. M., Martínez-Fernández, S., Bagnato, A., Partanen, J., López, L., & Rodríguez, P. (2019). Practical experiences and value of applying software analytics to manage quality. Proceedings of ESEM 2019. arXiv:1910.11007.

Yang, G. (2025). CTQRS-based reinforcement learning framework for reliable bug report generation using open-source large language models. Applied Sciences (Switzerland), 15(23), Article 12545. https://doi.org/10.3390/app152312545

© 2026 Advancing SDG Impact. All rights reserved.