West Science Information System and Technology
Vol. 4, No. 01, April 2026, pp. 135~145

Software Development Methodologies (2000–2026): A Scopus-Based Bibliometric Mapping of Scholarly Output and Thematic Evolution

Loso Judijanto
IPOSS Jakarta, Indonesia

Article Info

Article history:

Received Apr, 2026
Revised Apr, 2026
Accepted Apr, 2026

Keywords:

Software Development Methodologies
Bibliometric Analysis
Scopus
VOSviewer
Thematic Evolution

ABSTRACT

In this study, an analysis will be conducted on the intellectual structure and thematic development of the literature concerning software development methodologies during the time frame of 2000 to 2026 through the bibliometric method. The data were retrieved from the Scopus database and analyzed using VOSviewer to create various visualization networks, including co-authorship, citations, and keyword co-occurrences. According to the findings, there was a substantial increase in the amount of literature produced within the study domain, demonstrating the rising significance of software development methodologies in both theoretical and practical fields. Through the co-authorship analysis, it was observed that research collaboration tends to occur mainly between several dominant countries and organizations, but possibilities for collaboration across the globe still exist. In terms of citation analysis, the most cited sources were mostly related to the creation of software development tools, which highlighted the applicability of the domain. In addition, through the keyword co-occurrence and overlay analysis, the thematic development of the literature shifted from classic software engineering techniques to adaptable and smart methods, including agile development, artificial intelligence, and machine learning. Moreover, other emerging themes, such as sustainability, indicated the wide range of applications for the study field.

Corresponding Author:

Name: Loso Judijanto
Institution: IPOSS Jakarta, Indonesia
Email: losojudijantobumn@gmail.com

1. INTRODUCTION

Software development methodologies have proven themselves to be an integral part of software engineering in today's environment, as they serve as structured ways to plan, design, develop, test, and deploy software applications. Software development methodologies represent an important element of what is called the Software Development Life Cycle (SDLC). This cycle makes sure that software applications are developed in a systematic way. With the increase in complexity and demand for quick development of software over the last few decades, there have been various changes in software development methods. Consequently, several different methodologies have been developed and continue to evolve today [1].

The early days of software development were characterized by informal and unorganized approaches, particularly in the middle of the twentieth century, when computing was still very new. However, the complexity of software systems made it necessary to develop formalized procedures that could ensure systematic software development processes, and thus the emergence of the structured approaches in the 1970s in the form of the Waterfall model. The process-oriented model was characterized by the fact that each stage of the development had to be successfully completed first before proceeding to the next one, and it was quite efficient for the projects with stable requirements [2], [3].

However, the shortcomings of conventional techniques ultimately led to the development of techniques that were both more adaptable and responsive. These techniques included Agile methodology, which came about at the turn of the millennium in response to the problems associated with conventional methods of software development. Agile methodology places emphasis on iterative development, frequent communication, and cooperation. Rather than focusing on delivering the software as one finished product, agile advocates for the gradual delivery of products through shorter development periods called sprints [4], [5].

Apart from Agile, other methodologies like Scrum, Kanban, Lean Software Development, and DevOps have continued to revolutionize software development practices. The methodologies incorporate Agile methods while tackling some problems. For example, while Scrum has clearly defined roles and events that help in handling iterative development, DevOps facilitates a continuous development process by combining software development with IT infrastructure operations. Overall, these methodologies illustrate how the world of software development is increasingly embracing automation, collaboration, and quick deployment [2].

The period between 2000 and 2026 can be singled out as one in which the changes in software development methodologies were especially profound. This is due to globalization processes, digitization, and the emergence of cloud computing and artificial intelligence technologies. It became common practice for firms to use hybrid software development methodologies, as well as develop new methodologies that could meet their individual demands. The increasing number of articles devoted to this topic in scholarly publications speaks about the interest of the scientific community in studying these changes. Using bibliometric approaches, including those based on large databases like Scopus, allows gaining a better understanding of the publications' dynamics, key authors, collaboration patterns, and topics under investigation.

In spite of the considerable number of software development methodologies created and applied to date, there is still not enough information available to assess how scholarly production and trends in this area have developed over time. In other words, previous research has paid attention to particular software development methodologies or a certain period of time, failing to consider an integrated approach and observe the whole process of their development. Moreover, the growing number of papers published in the field poses problems for identifying important contributions, prominent trends, and new issues. The problem under discussion calls for a bibliometric analysis of software development methodologies during 2000-2026 via the Scopus database.

The purpose of this study is to undertake a bibliometric mapping analysis of academic literature related to the methodology of software development between 2000 and 2026 by relying on data available in Scopus. More precisely, the goal is to assess publication trends, identify prominent scholars and organizations, examine the nature of cooperation, as well as analyze the evolution of themes. The contribution of this study to academic literature will consist of a detailed map that sheds light on the evolution of software development methodology.

2. METHODS

The present study applies the bibliometric research method for systematically examining the literature related to software development methodologies from the period 2000 to 2026. Bibliometrics refers to the quantitative methodology typically employed in the discipline of Library and Information Science to examine the pattern and trend of academic research. The major data source in the present study will be the Scopus database because it comprises a broad range of peer-reviewed journals, conference papers, and scholarly articles. The data collection procedure will involve searching the Scopus database for relevant articles by employing search terms like “software development methodologies,” “Agile,” “Scrum,” “DevOps,” and “SDLC.” The search will cover articles published within the stated period and pertaining to specific subjects, such as computer science and engineering.

After collecting data, this study will use bibliometric methods that include both descriptive and network-oriented analysis approaches. Citation analysis is used to identify influential sources and the intellectual structure of the field under consideration. Co-authorship, co-citation analysis, and co-word analysis will also be applied to explore relationships within the collected data. These methods help understand collaboration structures, seminal sources, and topical groups. VOSviewer will be applied for creating visualization maps of the relationships between authors, institutions, and topics.

For the assessment of thematic evolution, this study has divided the publication era into several sub-epochs, providing a longitudinal insight into the evolution of research themes. The keyword analysis method is employed to determine rising trends, waning trends, and stable trends in the research domain of software development methodologies. The thematic mapping process is utilized for the classification of themes according to their importance and evolution, which allows the identification of important themes, niche themes, and emerging themes in the subject of interest. Thus, by employing both quantitative measures and visual mapping methods, this methodology will offer valuable information about the evolution and interaction of different themes, including Agile approaches, continuous integration, and DevOps.

3. RESULTS AND DISCUSSION

3.1 Co-Authorship Analysis

Author co-analysis is used in order to investigate the patterns of collaboration that exist in the sphere of studies of software development methodologies. Author co-analysis allows identifying patterns of scientific collaboration by studying connections between authors, institutions, and countries. This helps identify important authors who have an essential role in creating new knowledge in the area under discussion.

1. Author-level Visualization

Figure 1 displays the co-authorship network visualization for the research area on software development methodology approaches. With the help of VOSviewer, the co-authorship network is created where nodes represent authors and links represent collaborations. The network also provides insight into different clusters of research through color coding. Through this network visualization, it becomes possible to recognize the most common collaboration groups, strength of collaboration, and placement of authors in the research world.

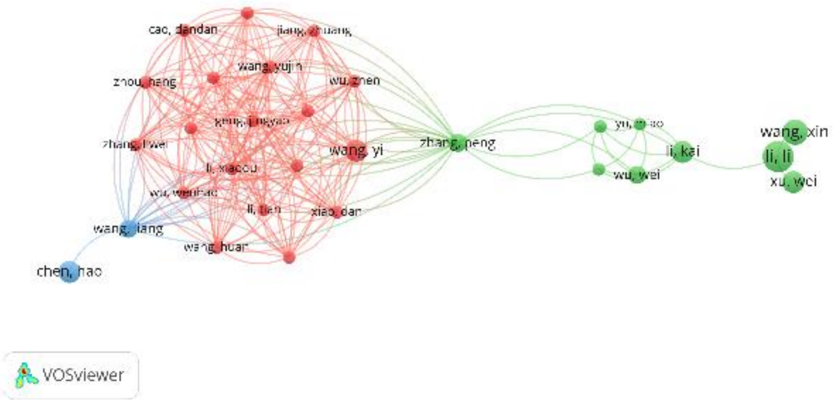

Figure 1. Author-level Visualization
Source: Data Analysis

It is clearly evident from the diagram that the existence of a very densely packed cluster (depicted in red) characterizes the network. This means there are very intense interactions among a few scholars who seem to be collaborating actively. There is a high probability that a research community exists with high frequency of interactions in terms of joint publication activities. These authors have shown high degree of interaction among themselves due to academic collaboration. On the other hand, the blue colored cluster is comparatively smaller and less interacting.

Moreover, the green cluster is somewhat isolated from the core red cluster but indirectly linked with each other. It means that new research groups or research groups localized within some regions are forming and trying to link up with the major academic group. Spatial isolation of clusters represents lower collaboration, thus, possible fragmentation within the research field.

2. Institution-level Visualization

In the Figure 2 below, the citation network visualization shows how research on software development methods is structured intellectually through visualizing linkages among sources which have been cited. In order to achieve this, citation linkages among various sources are visualized using the VOS viewer, where nodes denote sources or documents, while edges refer to citation linkage among the sources.

Figure 2. Institution-level Visualization
Source: Data Analysis

From the diagram, we can see that it is a network which is not very dense, comprising only two primary nodes and one link between them, signifying that there is some degree of citation connection in place. In one corner of the network, we have the node of the Institute for Mental Health (red cluster), which stands alone aside from its linkage with another node, hinting at the idea that it may act as a special or niche reference source within the data set. The relatively poor connection among these nodes points to the fact that the citation network of this part of the data set has not become too inter-connected, possibly implying either a fragmented body of literature or a body of literature that specializes into many research streams. The poor connection among these two clusters could imply that there are interdisciplinary relations here.

3. Country-level Visualization

The network of authorship collaboration in the country level, as depicted in Figure 3, is the best way to understand the patterns of collaboration on a global scale for software development methodology. This network visualization created by the help of VOS viewer depicts nodes as countries, and links between the nodes show the collaborative efforts between countries based on co-publications. The size of the nodes depicts the magnitude of involvement, whereas clusters show the groupings of countries with mutual collaboration in the research field.

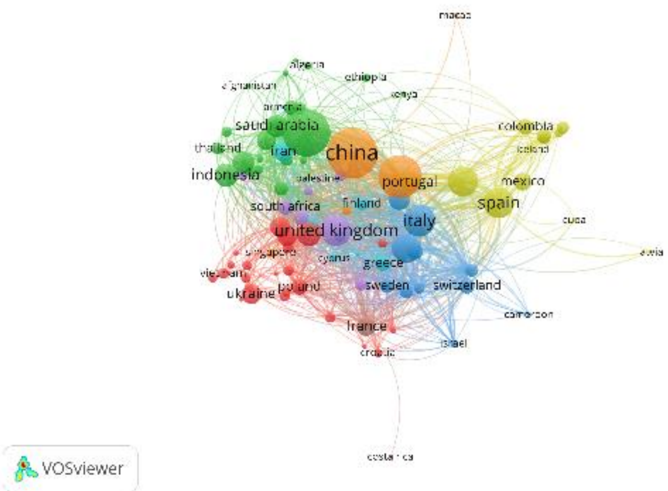

Figure 3. Country-level Visualization

Source: Data Analysis

As can be seen from the above diagram, China, the UK, Italy, and Spain have very significant positions in this network, signifying their importance and dominant nature when it comes to global research collaboration. China is one of the most significant nodes in the network, meaning that it is one of the largest contributors when it comes to number of publications and collaboration with different nations. Likewise, the UK and Italy have been identified as two of the countries having very good connections in the European research cluster.

Moreover, the fact that there are clusters of countries, such as the European cluster (in red and blue colors), the Asian and Middle Eastern cluster (in green color), and the Latin American cluster (in yellow color), implies that collaboration among nations is done along geographic or geopolitical lines. Nevertheless, certain linkages exist between the clusters, which point to an increase in the degree of globalization of scientific efforts. Portugal and Spain serve as bridges between Europe and Latin America; China connects to both the Western countries and the developing countries.

3.2 Citation Analysis

Citation analysis is conducted to evaluate the intellectual impact and influence of scholarly publications within the domain of software development methodologies.

Table 1. The Most Impactful Literatures

Citations	Authors and year	Title
41760	[6]	Research electronic data capture (REDCap)-A metadata-driven methodology and workflow process for providing translational research informatics support
31080	[7]	Software news and update AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading
23454	[8]	Features and development of Coot
20468	[9]	PHENIX: A comprehensive Python-based system for macromolecular structure solution
20032	[10]	WGCNA: An R package for weighted correlation network analysis
17112	[11]	Geneious Basic: An integrated and extendable desktop software platform for the organization and analysis of sequence data
11541	[12]	Advances in functional and structural MR image analysis and implementation as FSL
11444	[13]	BEAST: Bayesian evolutionary analysis by sampling trees
10896	[14]	Overview of the CCP4 suite and current developments
10773	[15]	MEGA3: Integrated software for Molecular Evolutionary Genetics Analysis and sequence alignment.

Source: Scopus, 2026

From the table above, one can observe that the most cited literature within the selected data set mainly comprises software-related studies aimed at enhancing computational capabilities for science, especially bioinformatics, computational biology, and data analysis. The highly cited papers include Harris et al.'s (2009) discussion on the use of REDCap software for data management and Trott and Olson's (2010) discussion on the application of AutoDock Vina software for simulation. In addition, other papers related to Coot, PHENIX, WGCNA, and Geneious Basic indicate the significance of software programs in conducting scientific activities. Overall, the table above suggests that the most cited literature involves more than the traditional approaches to software development and includes innovative methods of software applications.

3.3 Keyword Co-Occurrence Analysis

Analysis of keyword co-occurrence is applied to investigate the structure and evolution of knowledge within the research area of software development methodologies. Through examining the co-occurrence of keywords, this method determines research themes, trends, and interconnections among various areas of research. This method gives a clear picture of the way research trends have developed throughout the years and what possible future directions may be taken.

1. Network Visualization

The co-occurrence network map presented in Figure 4 demonstrates the basic framework of the software development methodologies research domain. This co-occurrence network map is generated through the VOSviewer application, where each node is defined by a keyword, and the links show the number of times that the keywords co-occur in the same papers. The larger size of the node shows the importance of particular concepts, while the different colors of nodes show the clustering of research themes around related topics.

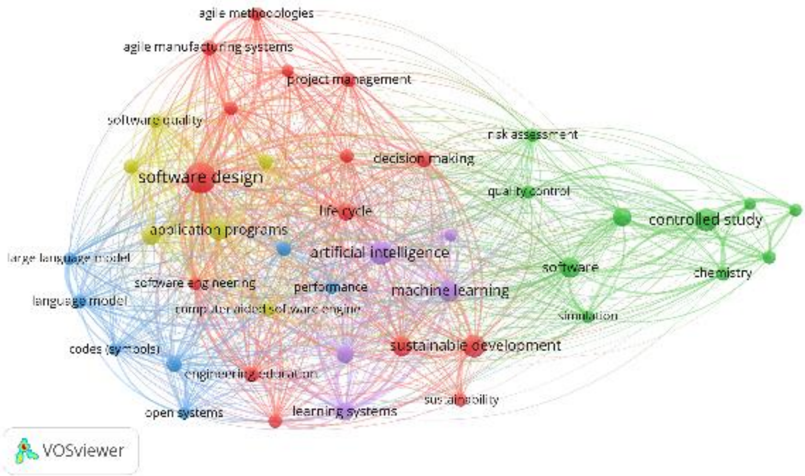

Figure 4. Network Visualization

Source: Data Analysis

According to the figure above, "software design" has emerged as one of the central keywords with the greatest dominance, appearing at the center of the map with many links to other words. The appearance of software design at the center suggests that software design is still one of the fundamental topics in this literature review, with strong ties to different aspects of development methods. Strong links between "software design," "application programs," "life cycle," and "project management" suggest that principles of software engineering still have a place in methodology research.

The other notable cluster (red) focuses on topics related to agile methods, decision-making, and project management. It is pertinent to note that this cluster highlights the increasing significance of flexibility and iteration in software development. The inclusion of words such as performance and risk evaluation in this particular cluster is indicative of the fact that recent researches have become increasingly focused on measuring the efficiency of Agile methods.

In the case of the green cluster, one can observe numerous interrelations with other sciences like controlled experiments, quality control, simulations, and chemistry. The implication is clear: software development practices are not just considered in terms of software engineering, but they are also implemented in other scientific spheres. Thus, one can say that software practices are used not only in IT projects but in experimental or simulated settings.

Blue cluster covers computational issues, such as artificial intelligence, machine learning, computer-aided software engineering, and large language models. Such a concentration denotes the ongoing fusion between software engineering approaches and digital trends. The numerous mentions of artificial intelligence indicate that intelligent systems and automation processes have an impact on modern software development practices.

2. Overlay Visualization

The keyword co-occurrence overlay visualized in Figure 5 shows the timeline of research trends within software development methodology. The keywords on the map are categorized according to their average year of publication with lighter shades of blue indicating studies conducted in earlier years and yellow representing the recent ones. This method is useful in tracking changes in the subject matter of interest over time.

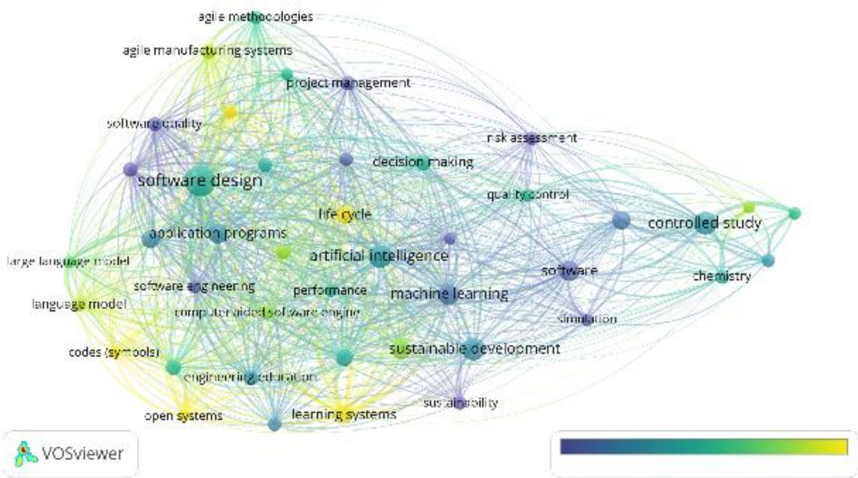

Figure 5. Overlay Visualization
Source: Data Analysis

The diagram shows that previous research areas, signified by the blue and green hues, have mostly focused on basic notions, including software design, software systems, application programs, and computer-aided software engineering. This is because these concepts characterize the conventional emphasis in software engineering research that involves development processes and design methodologies. The dominant placement of these terms shows that they have formed the very foundations of this area of study, thus acting as the pillars for future methods to be developed.

With the growth of the field, it is clear that there is an increasing tendency towards integration and intelligent applications. Words like artificial intelligence, machine learning, decision making, and risk assessment have been colored greenish to signify that these were only recently coined. This is indicative of the trend where there has been increased emphasis on data-driven and adaptive systems within software engineering through the use of automated, predictive, and intelligent mechanisms for decision-making purposes.

Trends from more recent research papers, marked in yellow, consist of themes like sustainable development, sustainability, agile methodologies, and engineering education. These new trends show that modern research is not limited to technical aspects but also covers other areas. For instance, the increased presence of the word Agile reflects its importance in recent times, while the occurrence of terms related to sustainability shows the new trend toward software development that takes into account ecological and moral aspects of the process.

3. Density Visualization

The density visualization presented in Figure 6 provides a heatmap representation of the concentration and intensity of research topics in software development methodologies. In this VOSviewer-generated map, areas with brighter colors (yellow) indicate higher keyword density and greater research attention, while darker regions (green to blue) represent less frequently studied topics. This visualization is particularly useful for identifying core research hotspots as well as peripheral or emerging areas within the field.

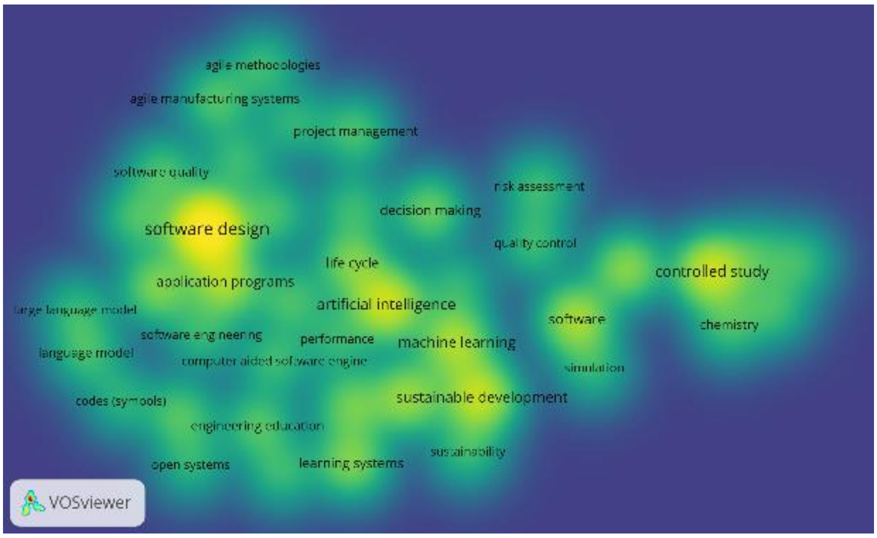

Figure 6. Density Visualization
Source: Data Analysis

This visualization shows that the keyword "software design" is the most dominant and concentrated keyword, as represented by the brightest yellow area in the middle of the graph. This indicates that software design continues to be the main concern of researchers and acts as an overarching theme linking different topics like application program, software engineering, and lifecycle. In addition, the green to yellow color gradients around the brightest part represent other closely related subjects like artificial intelligence, machine learning, and decision-making. All of these confirm the deep integration between conventional software engineering practices and computation.

On the contrary, disciplines like controlled study, chemistry, and simulation are found in isolated locations with lower density towards the right part of the map. This implies that even though these subjects have a presence in the research environment, they are specialized and not extensively linked to the main aspects of software development techniques. Likewise, newer disciplines like sustainable development and sustainability are evident but at medium density, implying increasing interest but still not dominating the research scope.

Discussion

From the findings of the bibliometric analysis, it can be observed that research into software development methodologies has become a very dynamic and multidisciplinary area of study during the years 2000-2026. The emphasis on core topics like software design and system development suggests that fundamental aspects of software engineering remain relevant in defining the scope of research. Nevertheless, the inclusion of emerging concepts, especially the Agile method of software development, signifies a change in approach towards a more adaptive, iterative, and responsive methodology.

As a result of the analysis of co-authorships, it is possible to point out the presence of significant clusters of collaboration between some leading countries including China, the United Kingdom, and certain European nations. Thus, it is safe to say that scientific knowledge is being produced through effective collaboration within established academic networks. On the other hand, there are also some small clusters indicating unequal collaboration worldwide as there are some parts of the world which are not sufficiently represented in the sphere.

Based on citation, the findings reveal that very influential papers are mostly linked to the creation of software instruments that are broadly used by researchers. In other words, the usefulness of certain applications that help researchers conduct analyses, simulations, and streamline processes is more likely to have an impact in academia than a paper that lacks any practical utility. Such findings make sense because the field is mainly concerned with applied issues, where the use and scalability of software instruments are important criteria in assessing academic significance.

As observed from the analysis of keywords through co-occurrence and overlay, there is evident evolution in themes within this particular topic. Initially, the works done by authors were mainly concerned with the methodology in software engineering and the technicalities surrounding software engineering. However, current literature highlights the incorporation of artificial intelligence and machine learning in the software development process. The evolution to topics such as sustainability and engineering education highlights an expansion of software engineering beyond just technical proficiency to other issues affecting society and the environment.

It becomes clear from the density visual analysis that although the classic themes still dominate, the new ones are slowly being recognized by researchers. The appearance of new themes with low density indicates that there is room for more research within the discipline and great opportunities exist in the future. In particular, one can mention the combination of SDM, sustainability, and intelligent systems.

4. CONCLUSION

The current bibliometric analysis shows that studies related to software development methods in the period between 2000 and 2026 have witnessed considerable changes and development, marked by the transition from classical, plan-oriented methods to new, adaptive, and technology-infused models. Results show that whereas fundamental topics like software design continue to form the basis of the studies, new topics like Agile development methods, artificial intelligence, and sustainability are gaining traction in determining the future course of research in the domain.

REFERENCES

[1] P. Kokol, M. Kokol, and S. Zagoranski, "Software development with scrum: a bibliometric analysis and profile," *Libr. Philos. Pract.*, pp. 1–21, 2020.

[2] F. Almeida, "Bibliometric analysis of agile software development," *arXiv Prepr. arXiv2004.05876*, 2020.

[3] P. N. Mata, J. M. Martins, and J. C. Ferreira, "New software product development: Bibliometric analysis," *J. Knowl. Econ.*, vol. 16, no. 1, pp. 4161–4184, 2025.

[4] F. Conceição, M. J. Dias Lousã, and J. C. Pereira de Morais, "SECURITY AND RISK IN SOFTWARE DEVELOPMENT PROJECTS: A BIBLIOMETRIC REVIEW," 2026.

[5] S. S. Maidin, N. Yahya, M. A. bin Fauri Fauzi, and N. S. A. A. Bakar, "Current and future trends for sustainable software development: Software security in agile and hybrid agile through bibliometric analysis," *J. Appl. Data Sci.*, vol. 6, no. 1, pp. 311–324, 2025.

[6] P. A. Harris, R. Taylor, R. Thielke, J. Payne, N. Gonzalez, and J. G. Conde, "Research electronic data capture (REDCap)—a metadata-driven methodology and workflow process for providing translational research informatics support," *J. Biomed. Inform.*, vol. 42, no. 2, pp. 377–381, 2009.

[7] O. Trott and A. J. Olson, "AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading," *J. Comput. Chem.*, vol. 31, no. 2, pp. 455–461, 2010.

[8] P. Emsley, B. Lohkamp, W. G. Scott, and K. Cowtan, "Features and development of Coot," *Biol. Crystallogr.*, vol. 66, no. 4, pp. 486–501, 2010.

[9] P. D. Adams *et al.*, "PHENIX: a comprehensive Python-based system for macromolecular structure solution," *Biol. Crystallogr.*, vol. 66, no. 2, pp. 213–221, 2010.

[10] P. Langfelder and S. Horvath, "WGCNA: an R package for weighted correlation network analysis," *BMC Bioinformatics*, vol. 9, no. 1, p. 559, 2008.

[11] M. Kearse *et al.*, "Geneious Basic: an integrated and extendable desktop software platform for the organization and analysis of sequence data," *Bioinformatics*, vol. 28, no. 12, pp. 1647–1649, 2012.

[12] S. M. Smith *et al.*, "Advances in functional and structural MR image analysis and implementation as FSL," *Neuroimage*, vol. 23, pp. S208–S219, 2004.

[13] A. J. Drummond and A. Rambaut, "BEAST: Bayesian evolutionary analysis by sampling trees," *BMC Evol. Biol.*, vol. 7, no. 1, p. 214, 2007.

[14] M. D. Winn *et al.*, "Overview of the CCP4 suite and current developments," *Biol. Crystallogr.*, vol. 67, no. 4, pp. 235–242, 2011.

[15] S. Kumar, K. Tamura, and M. Nei, "MEGA3: integrated software for molecular evolutionary genetics analysis and sequence alignment," *Brief. Bioinform.*, vol. 5, no. 2, pp. 150–163, 2004.