International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 10-No.4 (2024) pp. 1242-1248
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

The Use of Agile Models in Software Engineering: Emerging and Declining Themes

Sevinç AY*

Firat University, Department of Software Engineering, 23119, Elazığ, Turkey
* **Corresponding Author Email:** say@firat.edu.tr - **ORCID:** 0009-0001-6309-0889

Article Info:

DOI: 10.22399/ijcesen.703
Received : 29 November 2024
Accepted : 04 December 2024

Keywords:

Software Engineering,
Agile Software,
Agile Approaches,
SCRUM.

Abstract:

Software engineering is a systematic discipline that covers the creation of a software product at the end of software development processes. Many different models are used in software development processes. The most widely used model today is the Agile Approaches model. Methods that focus on increasing customer orientation and speed in software development projects are considered agile approaches. Agile approaches are used to overcome the limitations of traditional methods such as the waterfall approach and to adapt to constantly changing requirements. Agile approaches increase collaboration in a project team established to develop a software product and contribute to product success by increasing customer satisfaction. Agile approaches are highly preferred for distributed software development environments and remote teams. The reviewed literature shows that agile approaches increase flexibility and speed of delivery in software engineering projects, but they also bring many difficulties when they are not applied with an expert team in large-scale projects or when they are applied incorrectly. Agile approaches have transformed software engineering processes, prioritizing customer requirements and supporting high quality software development processes. However, factors such as the experience of teams, organizational culture and process management skills are critical for the successful implementation of agile methods. Within the scope of the study, bibliometric analysis was conducted on 1372 data obtained from the Web of Science database. While the prominent fields of study in the study were “software development”, “information-systems” and “decision-making”, the themes that are about to disappear were “management” and “framework”. The study provides a roadmap for future applications of agile methods in software engineering.

1. Introduction

Software engineering is a systematic and measurable engineering discipline in the development, design and maintenance of software [1]. Software engineering encompasses the process of developing high quality software through the software development models used [2]. The concept of software engineering first emerged in 1968 to meet the software demands of the US military. For about 10 years, computer programmers were referred to as software engineers. A few universities started to offer master's degrees in software engineering. Between 1995 and 2005, software engineering became the fastest growing career field in the US. This growth coincided with the increased use of personal computers, the proliferation of the Internet, and the rise of the commercial software industry. In addition to the undergraduate and graduate education options offered by many universities, some universities also offered online software engineering graduate education options [3]. Software engineering processes are guidelines or instructions that examine the stages that programs go through in order to manage a computer system and program different hardware units. At the core of software engineering, following the software process is critical in the design, development, testing and maintenance of high-level, complex and large software. Improper or incomplete execution of one or more of these processes may cause the resulting software product to be weak [2,3]. The steps followed as software development process are called Software Development Life Cycle (SDLC). The basic steps of the software life cycle are planning, analysis, design development, integration and

testing, and implementation [4]. There are many different models in the software life cycle. The size, scope and user diversity of the software cause the diversity of models. Therefore, choosing the model to be used in a software project plays a very important role [5]. As a result of the increasing demand for software applications, the development of the software industry has gained momentum. Customer satisfaction, market demand, and business prosperity are important principles that the software industry considers when developing successful software. The increasing number of software project implementations has led to many failed software products. This is due to improperly selected software models, methods or processes [6, 7].

Traditional software development models focus on a series of defined phases. The requirements phase is fully specified before coding. In this respect, it is easy to monitor the process and plan the tasks. Due to the lack of customer involvement in the process, new models have been sought. The most popular traditional model is the waterfall model. The Waterfall model consists of seven development phases: requirements analysis, design, coding and implementation, testing, operation and deployment, and maintenance. Each of these phases must be fully completed before moving on to the next phase. In the first three phases all planning is done and only when all coding and testing is complete can the project be deployed. The last phase does not end because perfecting the product and improving its functionalities will continue as long as the product is used [3]. Agile approaches are preferred over traditional models in order to solve their limitations. Especially after the publication of the agile manifesto, the development and use of agile models accelerated [8]. Agile approaches prioritize customer requirements and are itrative models that respond quickly to changes. In this respect, they are superior to traditional methods. The most widely used agile approaches are listed as follows: eXtreme Programming Development Process (XP), Scrum Development Process, Crystal Family of Methodologies (CM) and Dynamic System Development Method (DSDM) [9].
The aim of this paper is to present a systematic review showing the structure of agile approaches. The prominent topics about agile approach models and the themes that have lost their importance are examined in detail. The article is organized as follows: Section 2 contains all the related work where these methods are discussed. Section 3 presents an analysis of the agile approaches introduced through a systematic review. Section 4 includes the findings. Section 5 presents conclusions and discussion.

2. Related Work

Kaluza et. al [8] presented a hybrid approach to software engineering education. A realistic project environment was created to prepare students for their roles in software teams. With the hybrid model presented, students' attitudes are addressed. It was observed that students adapted to the approach that was expected to cope with changes in requirements. Hamed et. al. [9] conducted a systematic review of popular software development methods. . The most widely used agile software approaches are introduced and analyzed comparatively. In particular, the shortcomings of agile approaches used by SMEs are shown. In addition, solutions to these limitations are presented. Edison et. al. [10] present a systematic literature review of studies on Agile approaches. In addition, based on the findings of these studies, problems in the literature are identified. It is also emphasized how specific methods are selected for commercial applications. Limitations and success factors for the use of agile approaches, especially in large-scale projects, are discussed. The study also addresses the research gaps that have emerged in this area. Aleida et. al. [11] prepared a study in which they demonstrate the benefits of using Agile and DevOps approaches together in terms of managing customer requirements. It shows that these two paradigms may have incompatible aspects, but when properly structured, they can benefit organizations. Limitations of the study include the fact that some of the case studies included in the study do not provide sufficient information and that these analyses may not be unbiased because they are taken from commercial companies.

3. Material and Methods

Web of Science database, the keywords “agile models” and “agile software” were used with the conjunction “OR” and the keyword “software engineering” was used with the conjunction “AND”. “All fields” were selected in the search process. 1372 documents were reached. Bibliometric analysis was performed on the data obtained within the scope of the study. R Studio and Bibliometrix R-Package programs were used for bibliometric analysis. The concept of bibliometrics means the examination of scientific studies through statistical methods and numerical analysis [12]. Bibliometric analysis is used to examine, follow and identify studies in any field [13]. It is one of the qualitative and quantitative analysis methods used to evaluate the authors, study groups, countries, and the effects of journals [14]. Within the scope of the study, the following questions were tried to be answered:

- What is the annual scientific production of the studies?
- How are the links between authors, exhibitions and keywords in the studies?
- How is the thematic analysis of the studies?

4. Findings

The findings of the bibliometric analysis on the application of agile models and agile software in the field of software engineering are given in this section. It is seen that English language is mostly used in the studies obtained from the data set. Other study languages include Turkish, Spanish, Portugee, Estonian and German. Proceeding Paper (1078), Article (300), Review Article (27), Book Chapters (11) and Early Access (1) were identified as document types. The findings obtained from the data set show that the first studies were produced in 1998. Figure 1 shows the graph produced by the R program for annual scientific production. When annual scientific production is analyzed, studies gained momentum for the first time in 2001, when the Agile manifesto was published. Then, it is seen that 2010 stands out as another breaking point. In the first years when the annual number of publications progressed between 20-30, 87 studies were carried out in 2010. After this date, the studies conducted in the field have followed a fluctuating course. In the analysis, where the years 2018-2021 are accepted as the year range with the highest efficiency in terms of studies, the highest number was reached in 2020 with 129 studies. One of the

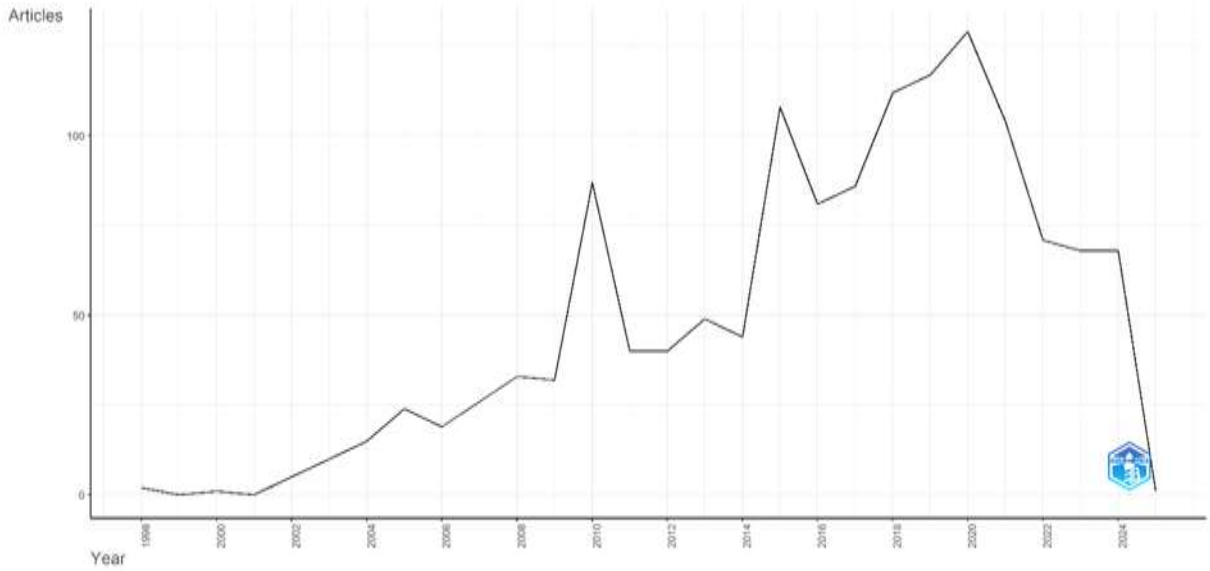

Figure1. *Annual Scientific Production*

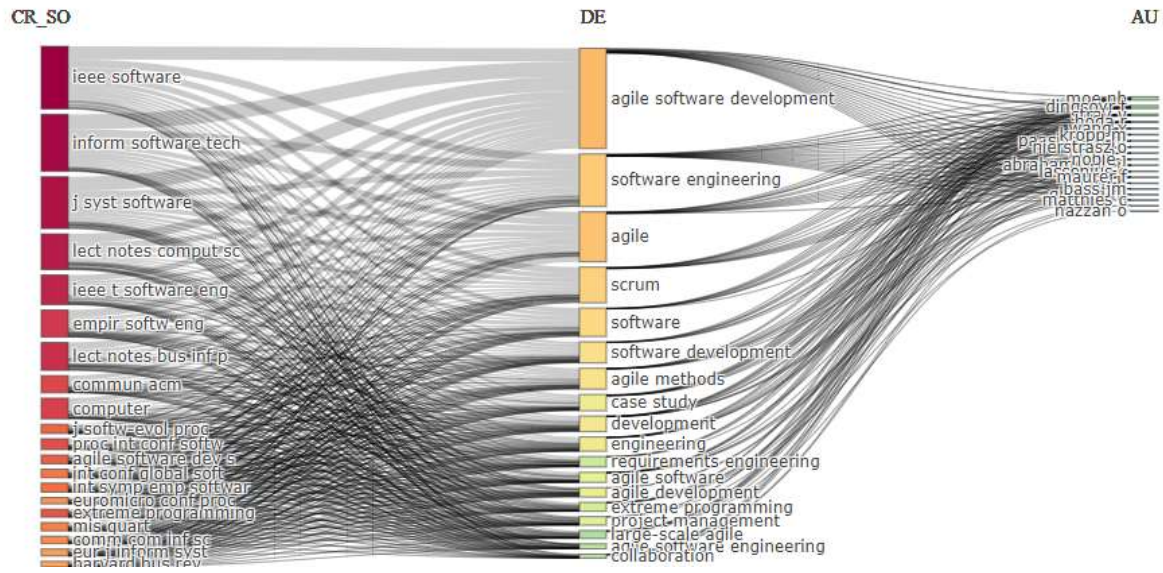

Figure 2. *Three-field plot*

important findings of the bibliometric analysis is The Three-Field Plot obtained from the R program. In Figure 2, the graph showing the links between the prominent authors, keywords and journals named as sources in the field is presented. In the figure, the left column represents countries, while the middle column represents prominent keywords. The right column represents journals. Grey colored connecting lines between variables visualize the relationships. The size of the rectangles representing the fields indicates that there are a large number of articles associated with that particular item. It was observed that the most frequently used keywords in publications were used in all of the journals that were prominent in citation. In addition, the top 20 most frequently used keywords by authors are presented in the graph. The gray links in between represent strong connections. It is seen that the first three keywords are the most used words and used in the top 20 journals. It was also found that all authors included the first three words in their studies. Another area examined within the scope of bibliometric analysis was the trending topics discussed in the studies obtained. Trending topics include the most commonly used words and fields of study. Figure 3 shows the trend topics obtained by the R program.

As can be seen in the figure, the trending topics with the highest frequency in the research area are presented by year. The horizontal axis represents the years of the studies. The vertical axis represents trending topics. Dots represent term frequencies. Software-development (63 frequencies) came to the forefront in 2018 and continued to be the trend topic with the highest frequency until 2021. Software, design, scrum, agile software are other prominent trending topics. While the oldest

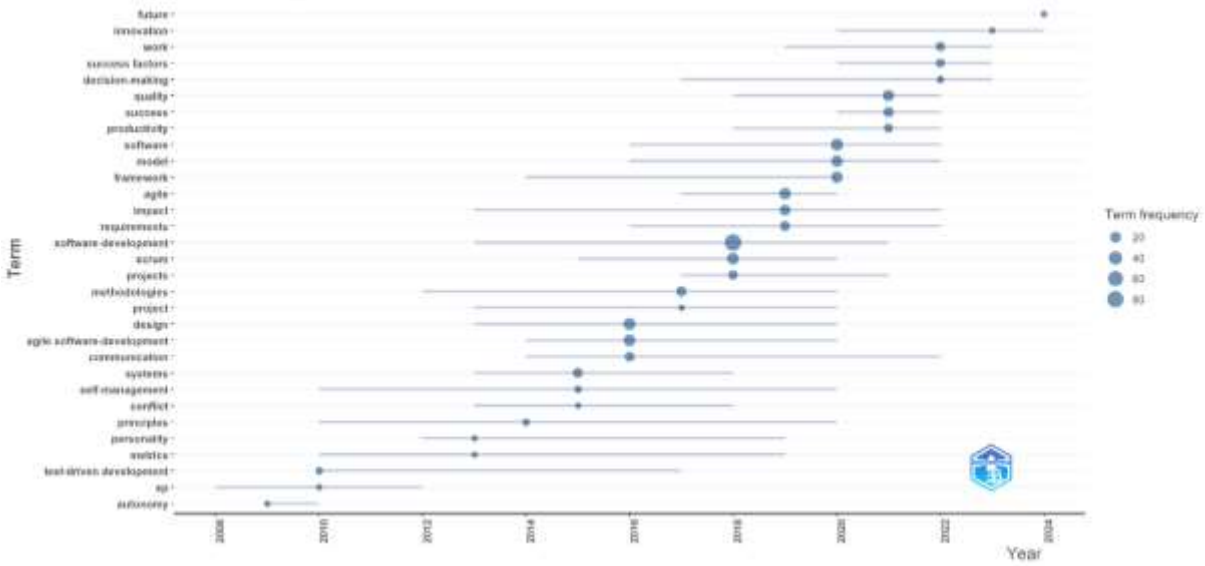

Figure 3. *Trend-topics*

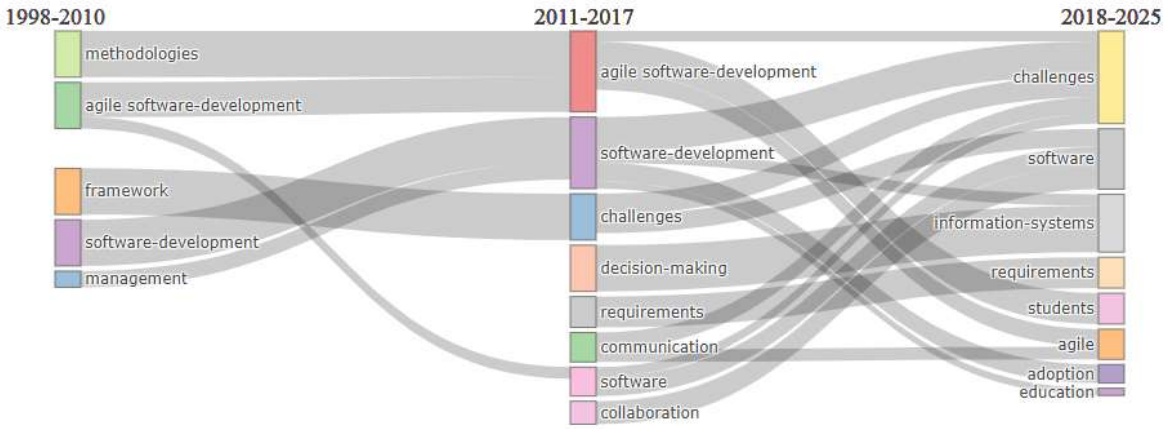

Figure 4. *Thematic evolution*

trending topics were autonomy and xp, these topics are outdated today. In recent years, innovation, future, success factors are new trending topics.

Another area that was analyzed bibliometrically was thematic evolution. Within the scope of the study examining agile approaches in the field of software engineering, the thematic evolution of author keywords was examined in order to see the development course of sub-themes from past to present in the literature. While examining the thematic development of the studies, it is divided into three periods. The years 2010 and 2017, in which significant leaps were made in the studies, were selected as the cutting year. "Walktrap" was selected as the clustering algorithm. The number of elements in each cluster was set as 3 and the minimum cluster frequency was set as 5. Similarities and differences between the prominent themes in these years were analyzed. Figure 4 shows the thematic evolution showing the prominent themes over the years.

The themes of "methodologies, agile software-development", "framework", "software development", "management" were prominent until 2010, when the number of articles started to rise from 1998 and reached the highest number. After 2010, it is seen that the themes of "methodologies" and "agile software-development" have merged under the theme of "agile software-development". In the same way, the themes of "software development" and "management" merged and continued to exist as "software development." It is seen that the theme of "framework", another prominent theme, feeds the theme of "challenges" and forms the basis for its emergence. Between the periods of '018-2025, it is seen that the themes have become clearer and the concepts have become clearer. It is understood that the newly added themes "education" and "students" come from publications in the field of software engineering education. Other findings of the analysis are that one of the strengthened themes is "challenges" and "decision-making" is replaced by "information-systems".

The last area examined as a bibliometric analysis finding was the thematic map. Louvain was used as the clustering algorithm in the thematic map drawn using author keywords. A thematic map of 250 keywords was created by selecting the minimum cluster frequency as 5. Figure 5 shows the thematic map created according to author keywords. In the thematic map, the y-axis represents density and the x-axis represents centrality. Centrality expresses the importance of the research topic in the thematic map. Intensity reflects the development of the relevant theme [15,16]. Thematic map has four different areas that describe the nature of the themes [17].

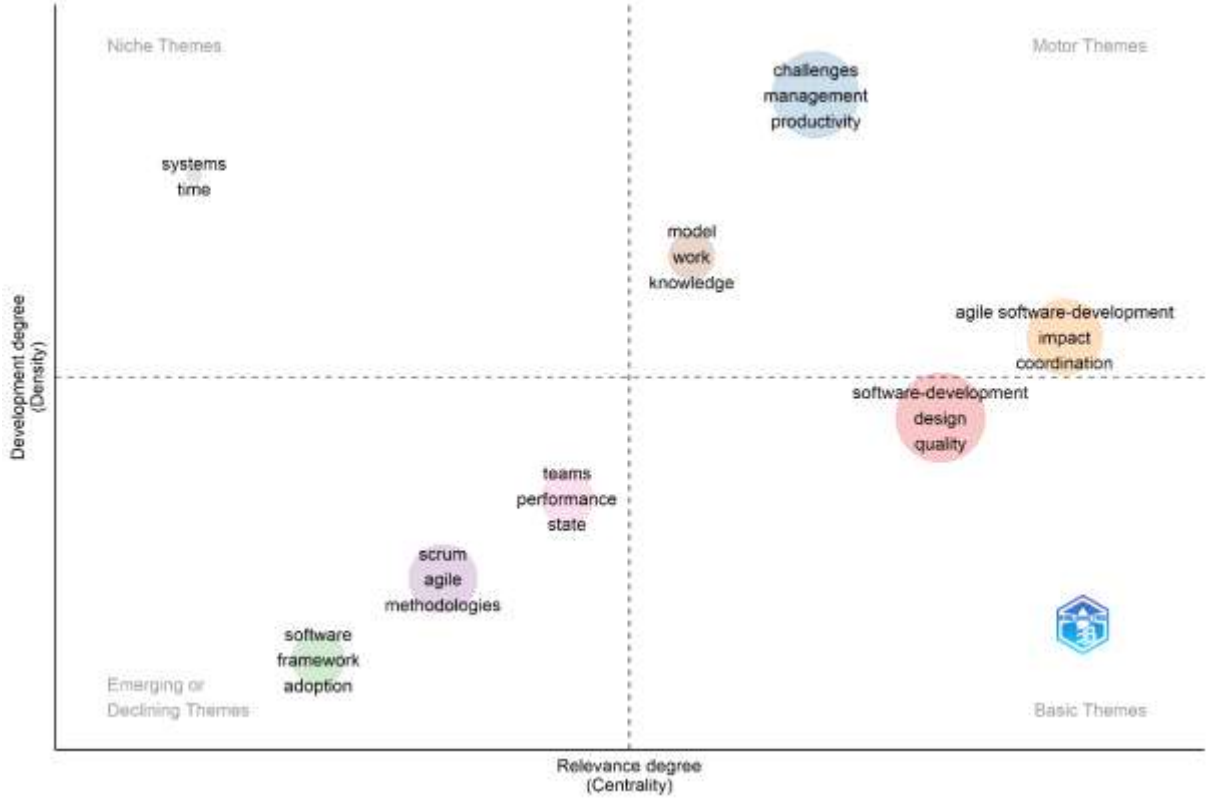

Figure 5. *Thematic map*

The themes in the upper right area of the diagram are advanced themes and are important in terms of structuring the research area. The themes in this area have high density and high centrality. They are called motor themes. The themes in the upper left area are highly developed but isolated themes. They have high density and low centrality. These themes are only of marginal importance because they have weak external links. Themes in this area are characterized as both very special and peripheral. They are known to be of limited importance to the field. Themes in the left sub-domain have low centrality and low intensity. This means that they represent emerging themes or themes in the process of disappearing. The themes in the right sub-section are characterized by high centrality and low intensity. This indicates that the themes are important for a research area and that the area is related to topics in different research areas. However, they are not sufficiently developed. A great deal of research has been conducted on these themes. When the diagram is analyzed, it is seen that a total of eight clusters are formed in four sections. Three different keywords were used to represent each cluster in the graph [15].

The motor themes in the diagram consist of three clusters. The first cluster consists of the keywords "challenges", "management", "productivity", the second cluster consists of the keywords "agile software-development", "impact", "coordination" and the third and smallest cluster consists of the keywords "model", "work", "knowledge". It can be said that these themes are the prominent focus topics in the field of agile approaches in software engineering. In addition, these themes are of great importance in the development of the field. Niche themes consist of a single cluster and two keywords. These words are "systems" and "time". These themes have been relatively less studied in research and therefore have not developed sufficiently. Emerging or declining themes consist of three clusters. The first cluster that stands out in this area consists of the words "scrum", "agile", "methodoligies". The other clusters seem to have equal weight. One of these clusters consists of the words "software", "framework", "adoption". The last cluster consists of the words "teams", "performance", "state". These themes represent topics that have recently emerged or are losing their importance in the field of study. Basic themes are represented in a single cluster. This cluster consists of the words "software-development", "design", "quality". The themes in this cluster are important themes that are frequently used in the field of study but are still in the process of development.

5. Conclusion and Discussions

Software engineering, which is the main discipline that includes software product development and maintenance processes, is a field that is developing day by day. Many different software development models are used in the software developed in this field. The software development model that emerged in order to eliminate the limitations of traditional models is agile models. These models, which have been developing since the day the agile manifesto emerged, are widely preferred. These models, which prioritize customer requirements and can be responded quickly, are also frequently preferred in the business sector. In this study, a bibliometric analysis study is presented in which the developing and prominent issues about agile models, which are widely used by software engineers, are investigated. Within the scope of the study, niche, basic and prominent themes were examined. Links between authors, journals and prominent subject areas are explained. Researchers who want to work in this field are provided with a lot of important information about the field.

In the study, 1372 data were obtained in the Web of Science database by using the keywords "agile models" and "agile software" with the conjunction "OR" and the keyword "software engineering" with the conjunction "AND" in the bottom line. It has been determined that Proceedings Paper stands out among the documets types obtained. The studies were conducted between 1998 and 2025. The study languages include Turkish, Spanish, Portugee, Estonian and German.

In the study, Engine themes consist of "challenges", "agile software-development", "impact", "model" themes. Niche themes are listed as "systems" and "time". Emerging or declining themes "scrum", "software", "framework", "teams", "performance" can be listed as some prominent themes.

It is suggested that the study can be extended by comparing it with traditional software development models. In the future, it is thought that the scope of the study can be expanded by examining the thesis studies conducted in this field. In addition, adding data not only from Web of Science database but also from Scopus, ScienceDirect database will increase the effectiveness of the study.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.

- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] Bourque, P., Robert, F. , Lavoie, J. M. , Lee, A. , Trudel, S., & Lethbridge, T. C. (2002). Guide to the software engineering body of knowledge (swebok) and the software engineering education knowledge (seek)-a preliminary mapping, *10th International Workshop on Software Technology and Engineering Practice*, pp. 8-23.

[2] Faruk, M. J. H., Subramanian, S., Shahriar, H., Valero, M., Li, X. , & Tasnim, M. (2022). Software Engineering Process and Methodology in Blockchain-Oriented Software Development: A Systematic Study, 2022 IEEE/ACIS *20th International Conference on Software Engineering Research, Management and Applications (SERA)*, Las Vegas, NV, USA, pp. 120-127, doi: 10.1109/SERA54885.2022.9806817.

[3] Rico, D. F., & Sayani, H. H. (2009). Use of Agile Methods in Software Engineering Education, *2009 Agile Conference, Chicago, IL, USA*, pp. 174-179, doi: 10.1109/AGILE.2009.13.

[4] Seker, S. E. (2015). Yazılım geliştirme modelleri ve sistem/yazılım yaşam döngüsü. Software development models and system/software lifecycle", YBS Ansiklopedi, 2(3), 18-29.

[5] Coşkan, D., & Onay Durdu, P. (2014). Kullanılabilirlik ve yazılım yaşam döngüsü: *Türkiye'deki yazılım organizasyonlarındaki durum. 8. Ulusal Yazılım Mühendisliği Sempozyumu*, 55-66.

[6] Caffery, F., Taylor, P., & Coleman, G. (2007). Adept: A Unified Assessment Method for Small Software Companies, *IEEE Computer Society*. 24(1);24-31, doi: 10.1109/MS.2007.3.

[7] Taylor, P., Greer, D., Coleman, G., McDaid, K., & Keenan, F. (2007). Preparing Small Software Companies for Tailored Agile Method Adoption: Minimally Intrusive Risk Assessment. *Software Process Improvement and Practice* 13(5):421 – 437 DOI:10.1002/spip.358

[8] Kaluza, M., Candrlic, S., & Katic, M. A. (2023). Hybrid Agile Approach in Software Engineering Education – A Case Study, *2023 46th MIPRO ICT and Electronics Convention (MIPRO), Opatija, Croatia*, pp. 1543-1548, doi: 10.23919/MIPRO57284.2023.10159921.

[9] Hamed, A. M. M., & Abushama, H. (2013). Popular agile approaches in software development: Review and analysis, *2013 International Conference On Computing, Electrical And Electronic Engineering (Icceee), Khartoum, Sudan*, pp. 160-166, doi: 10.1109/ICCEEE.2013.6633925.

[10] Edison, H., Wang, X., & Conboy, K. (2022). Comparing Methods for Large-Scale Agile Software Development: A Systematic Literature Review, *IEEE Transactions on Software Engineering*, 48(8);2709-2731, doi: 10.1109/TSE.2021.3069039.

[11] Almeida, F., Simões, J., & Lopes, S. (2022). Exploring the benefits of combining DevOps and agile. *Future Internet*, 14(2), 63.

[12] Al, U., & Tonta, Y. (2004). Citation Analysis: Sources Cited in Dissertations Completed at Hacettepe University Department of Librarianship. Information Word, 5(1), 19-47.

[13] Zupic, I., & Čater, T. (2015). Bibliometric methods in management and organization. *Organizational research methods*, 18(3), 429-472.

[14] Krauskopf, E. (2018). A bibiliometric analysis of the Journal of Infection and Public Health: 2008-2016. *J Infect Public Health* 11(2):224-229. doi: 10.1016/j.jiph.2017.12.011.

[15] Aria, M., Alterisio, A., Scandurra, A., Pinelli, C., & D'Aniello, B. (2021). The scholar's best friend: research trends in dog cognitive and behavioral studies. *Anim Cogn.* 24, 541–553.

[16] Cahlik, T. (2000). Comparison of the Maps of Science. *Scientometrics*. 49, 373–387. https://doi.org/10.1023/A:1010581421990

[17] Cobo, M. J., Martínez, M. A., Gutiérrez-Salcedo, M., Fujita, H., & Herrera-Viedma, E. E. (2015). 25 years at Knowledge-Based Systems. *Knowledge Based Systems*. 80, 3-13. https://doi.org/10.1016/j.knosys.2014.12.035