A Systematic Mapping Study on Scrum and Kanban in Software Development

Arif Hossain

Master's Thesis

Supervisor: Adnan Ashraf

The Faculty of Science and Engineering

Åbo Akademi University

2023

Abstract

Background: Agile methodologies, such as Scrum and Kanban, have gained significant popularity in software development organizations. However, there is a need to compare and contrast these methodologies to determine their effectiveness and suitability in specific conditions.

Objective: The objective of this systematic mapping study is to compare Scrum and Kanban in software development organizations and identify their methodological differences, benefits, drawbacks, and current/future trends.

Method: A comprehensive literature review was conducted, analyzing 47 primary studies. Data synthesis and analysis were performed to extract relevant information on the characteristics of Scrum and Kanban.

Results: The study identified several methodological differences between Scrum and Kanban, highlighting their unique characteristics and implementation considerations. The study presents a detailed breakdown of the reported differences, benefits, drawbacks, and trends associated with these methodologies.

Conclusions: Choosing between Scrum and Kanban depends on the specific needs, context, and goals of the organization. Scrum excels in areas such as path clarity, delivery time, and teamwork, while Kanban offers advantages in flexibility, easy transition, and focus on work. The findings emphasize the importance of understanding requirements, team dynamics, project characteristics, and customer expectations when selecting an agile methodology. This systematic mapping study contributes to the understanding of Scrum and Kanban in software development organizations. By considering the findings, organizations can make informed decisions and optimize their agile practices to enhance productivity, efficiency, and quality in software development projects.

Table of Contents

List of Figures

List of Tables

1. Introduction

Agile methodologies are increasingly popular in software development due to their ability to improve project efficiency, productivity, and quality. Scrum and Kanban are two of the most widely used and studied methods. While Scrum has been popular for some time and is widely used globally, Kanban has gained popularity in Europe and is rapidly gaining worldwide recognition.

The aim of this systematic mapping study is to identify and analyze existing research articles on Scrum and Kanban in software development to compare and contrast their effectiveness in specific conditions. The study answers four research questions related to the differences between Scrum and Kanban in approach and methodology, their benefits and drawbacks, the influence of their practices and principles on project efficiency, productivity, and quality, and the current state of research and future trends in comparing Scrum and Kanban in software development.

To answer the research questions, the study uses a systematic mapping study approach. This methodology involves a rigorous search strategy using specific inclusion and exclusion criteria to identify relevant research articles. The study conducts the search process using various electronic databases, including IEEE Xplore, ACM, and ScienceDirect, to ensure the inclusion of a wide range of studies. The identified studies are then evaluated based on predefined inclusion and exclusion criteria to select the most relevant studies for the review.

The results of the systematic mapping study are presented in a structured and organized manner, including tables and figures to provide a clear overview of the data analyzed. The study also provides recommendations for practitioners and researchers in software development on the most appropriate Agile methodology to use in specific conditions based on the findings of the review.

The study contributes to the body of knowledge in software development by providing a comprehensive comparison of Scrum and Kanban in Agile methodologies. Additionally, the study provides practitioners and researchers with valuable insights into the strengths and weaknesses of Scrum and Kanban and their applicability in specific conditions. The findings of this study also help to identify gaps in the literature and highlight areas for future research on comparing Scrum and Kanban in software development.

2. Background and Goal

Agile development has been widely adopted in the software development industry due to its effectiveness in managing complex and dynamic projects. Numerous agile methodologies have been developed over time, with Scrum and Kanban being the most popular ones. Scrum, a framework for managing and completing complex projects, has been widely adopted by software development teams worldwide. Kanban, a method for managing knowledge work with an emphasis on just-in-time delivery, is rapidly gaining popularity, especially in Europe. Both Scrum and Kanban offer different approaches and methodologies for managing software development projects, and understanding their differences and similarities is crucial for project success.

Despite the growing popularity of Scrum and Kanban, there is still a lack of understanding about their benefits and drawbacks in software development organizations. Moreover, there is a need to compare the two methodologies to understand which one is best suited for specific project conditions. Therefore, this systematic mapping study aims to identify and analyze relevant research articles that explore the use of Scrum and Kanban in software development projects.

2.1 History of Agile Methods

The origins of agile methods can be traced back to the 1990s when a group of software development practitioners gathered in Utah to discuss lightweight development methods. This meeting led to the creation of the Agile Manifesto in 2001, which outlined a set of principles for agile software development. The manifesto emphasized individuals and interactions, working software, customer collaboration, and responding to change over processes and tools, comprehensive documentation, contract negotiation, and following a plan (Beck et al., 2001).

One of the first agile methods to be developed was Extreme Programming (XP), which was created by Kent Beck in the late 1990s. XP is a software development methodology that emphasizes customer satisfaction, continuous delivery, and the use of technical practices such as pair programming, test-driven development, and continuous integration. XP was designed to be a lightweight alternative to traditional software development methods and to address the shortcomings of the Waterfall model (Beck, 1999).

Another agile method that gained popularity in the early 2000s is Scrum, which was initially introduced by Ken Schwaber and Jeff Sutherland in 1995. Scrum is a framework for managing and completing complex projects, particularly software development projects. Scrum emphasizes team collaboration, iterative development, and customer feedback, and it provides a set of practices and roles that help teams to organize their work and deliver high-quality software (Schwaber & Sutherland, 2011).

Another important agile method is Kanban, which was developed by David J. Anderson in the early 2000s. Kanban is a lean manufacturing method that has been adapted for software development. Kanban emphasizes the flow of work through a system and visualizing the work to identify and eliminate bottlenecks. Kanban uses a pull-based system to manage work and limits work in progress to improve flow and reduce lead time (Anderson, 2010).

The agile movement has grown significantly over the past two decades, and a wide range of agile methods and practices have emerged. These methods and practices include Lean Software Development, Crystal, Feature-Driven Development, Dynamic Systems Development Method (DSDM), and Adaptive Software Development (ASD) (Highsmith, 2009). Despite their differences, all of these methods share a common set of values and principles that prioritize flexibility, collaboration, and continuous improvement.

In summary, agile methods emerged in response to the limitations of traditional software development methods, such as the Waterfall model. The Agile Manifesto outlined a set of principles for agile software development that emphasized individuals and interactions, working software, customer collaboration, and responding to change over processes and tools, comprehensive documentation, contract negotiation, and following a plan. Over the past two decades, a wide range of agile methods and practices have emerged, including Extreme Programming, Scrum, and Kanban. These methods prioritize flexibility, collaboration, and continuous improvement and have been widely adopted in the software development industry.

3. Research Methodology

In this section, we will discuss the research methodology used in this systematic mapping study on Scrum and Kanban in software development. The methodology will be divided into four main sections, namely Planning, Conducting, Reporting, and Research goal and questions. Further, we will discuss the search and selection process, classification framework, data extraction, and data synthesis.

3.1 Planning

The planning phase of the research methodology involved defining the research questions and objectives, identifying the scope of the study, and determining the search strategy. The research questions were formulated based on the goal of the study. The objective of the study is to identify and compare Scrum and Kanban in terms of their approach and methodology for software development projects. The scope of the study is limited to the most commonly reported benefits and drawbacks of using Scrum and Kanban in software development organizations. The search strategy was defined by selecting the relevant databases and defining the search string.

3.2 Conducting

The conducting phase is where the research plan is put into action to carry out the systematic mapping study. This phase involves several steps that help to extract the data needed to answer the research questions. The conducting phase will follow the steps described below:

- Search and selection: In this phase, the search strings mentioned in the research protocol will be implemented in selected digital libraries, including IEEE Xplore, ACM, and ScienceDirect, to retrieve relevant publications related to Scrum and Kanban comparison in software development. The search strings will be modified to fit the formatting of each digital library, if necessary. Duplicates and irrelevant findings will be removed, and the selection criteria will be applied to shortlist primary studies that are relevant to the research questions. Backward snowballing will be performed to find additional relevant publications based on the reference lists of the primary studies.

- Data extraction: In this phase, the data extraction strategy as outlined in the research protocol will be followed to extract the required data from the selected primary studies. The data extraction will include details such as the year of publication, research questions, research methodology, software development context, Scrum or Kanban application, comparison criteria, benefits, drawbacks, and findings.
- Data synthesis: In this phase, the extracted data will be analyzed and summarized to provide the answers to the research questions. The findings from the primary studies will be synthesized to identify common themes, trends, and patterns in the comparison of Scrum and Kanban in software development.

3.3 Reporting

This section will describe the process of reporting the findings of the systematic mapping study. It will detail the format and structure of the report, including the use of tables and graphs to present data, and the methods used to analyze and interpret the findings.

3.4 Research questions and goal

The goal of our systematic mapping study is to compare Scrum and Kanban in software development projects. Our research questions are as follows:

RQ1. What are the key differences between Scrum and Kanban in terms of their approach and methodology for software development projects?

Rationale: To identify the fundamental differences between Scrum and Kanban methodologies in software development, in order to help software development organizations choose the most suitable approach for their projects.

Outcomes: A comprehensive analysis of the key differences between Scrum and Kanban methodologies in software development.

RQ2. What are the most commonly reported benefits and drawbacks of using Scrum and Kanban in software development organizations?

Rationale: To identify the potential benefits and drawbacks of using Scrum and Kanban methodologies in software development organizations, in order to help software development organizations make informed decisions about adopting these methodologies.

Outcomes: A synthesis of the most commonly reported benefits and drawbacks of using Scrum and Kanban in software development organizations.

RQ3. How do the practices and principles of Scrum and Kanban influence the efficiency, productivity, and quality of software development projects?

Rationale: To understand how the practices and principles of Scrum and Kanban methodologies influence the efficiency, productivity, and quality of software development projects, in order to help software development organizations improve their project outcomes.

Outcomes: An analysis of how the practices and principles of Scrum and Kanban methodologies influence the efficiency, productivity, and quality of software development projects.

RQ4. What is the current state of research on the comparison of Scrum and Kanban in software development, and what are the future trends and directions in this area?

Rationale: To understand the current state of research on the comparison of Scrum and Kanban methodologies in software development, and to identify future trends and directions in this area, in order to provide guidance for future research.

Outcomes: A systematic review of the current state of research on the comparison of Scrum and Kanban methodologies in software development, and identification of future trends and directions in this area.

3.5 Search and selection process

This section will describe the process of selecting relevant studies for inclusion in the systematic mapping study. It will outline the search string used to query databases, the databases searched, and the inclusion and exclusion criteria used to select studies.

We initiate our search and selection process by conducting an initial search across the digital libraries we have chosen. The following step includes the merging of publications and elimination of impurities, as well as the application of selection criteria to evaluate the pertinence of the publications. To complement this process, we employ backward snowballing as a supplementary strategy to identify relevant literature. Eventually, the data extraction concludes the entire process. The complete search and selection process, along with the size of our corpus at each of the five stages, is illustrated in Figure 1.

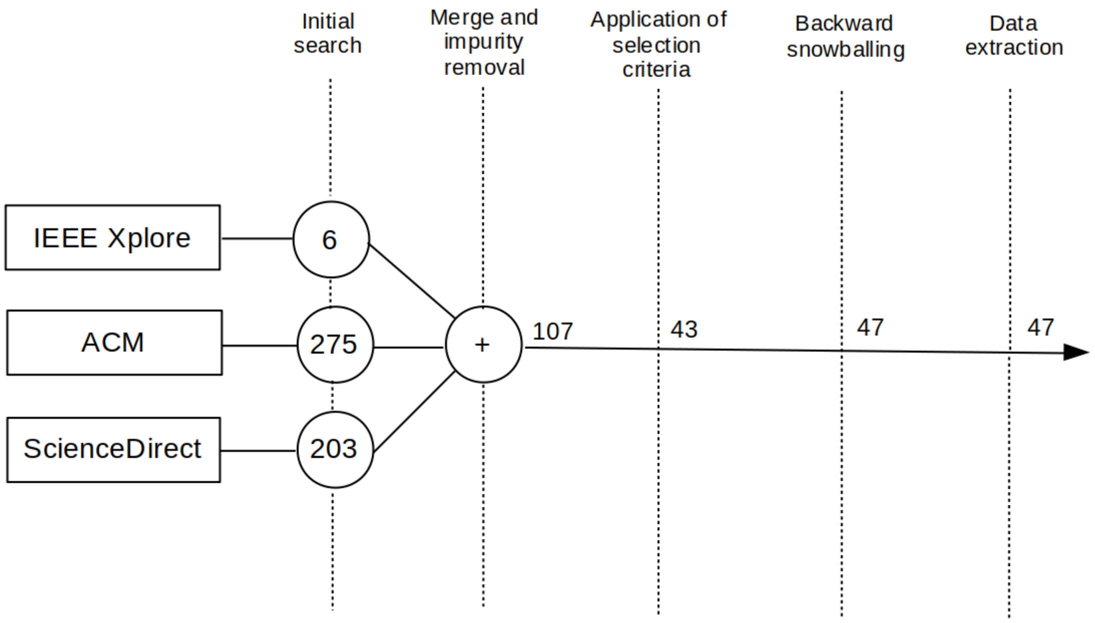

Figure 1: Selection Process

3.5.1 Initial search

The initial search for this study was conducted on three widely-used academic databases: IEEE Xplore Digital Library, ScienceDirect, and ACM Digital Library. The search string used was **"(Scrum AND Kanban) AND comparison"**, which allowed us to search the databases for papers containing relevant keywords in the titles and abstracts.

In total, 6 papers were found on IEEE Xplore, 275 on ACM, and 203 on ScienceDirect. The advanced search features on these databases were utilized to fully optimize the search string and extract the relevant results.

Table 1: Studies after first search

Source	Number
IEEE	6
ACM	275
ScienceDirect	203

3.5.2 Merging and impurity removal

In this sub-section, we will describe the process used to merge duplicate studies and remove irrelevant studies. We will explain the software used and the criteria used to identify duplicate studies and irrelevant studies.

The study selection procedure will be applied on the search results to remove false positives. It comprises two phases:

1. Title and abstract level screening
2. Full-text level screening

Title and Abstract Level Screening: Screen the titles and abstracts of the retrieved primary studies to determine if they meet the relevance criteria and if they provide evidence on the comparison between Scrum and Kanban in software development.

Full-text Level Screening: Evaluate the full-text articles of the primary studies that passed the title and abstract screening, and apply the inclusion and exclusion criteria to determine if the primary studies meet all of the required criteria for inclusion in the systematic literature review.

3.5.3 Application of selection criteria

We applied two sets of criteria, inclusion criteria and exclusion criteria, to select the relevant studies.

Inclusion criteria:

- The primary studies must be relevant to the research questions, and they must provide information on the comparison between Scrum and Kanban in software development.

- The studies must be based on empirical research methods, such as case studies, experiments, or surveys, which provide evidence on the benefits and drawbacks of Scrum and Kanban.
- The studies must be published within a specified time frame, such as the past 10 years or the past 5 years, to ensure that the information is up to date and relevant.
- The studies must be written in English

Exclusion criteria:

- The study is not relevant to the research questions, and does not provide information on the comparison between Scrum and Kanban.
- The study does not present an approach or method or algorithm or technique or framework
- The study was published more than 10 years ago.
- The study was not written in English

3.5.4 Backward Snowballing

Backward snowballing is another technique that will be used in this study to identify additional relevant studies. This technique involves reviewing the reference lists of the selected studies and examining them for potential inclusion. Backward snowballing is particularly useful for identifying older studies that may not have been captured in the initial search.

In this study, backward snowballing will be conducted on the selected studies using the same inclusion and exclusion criteria as the initial search. Any additional studies identified through backward snowballing will undergo the same screening and selection process as the initially identified studies. Figure 2 demonstrates the complete search and selection process.

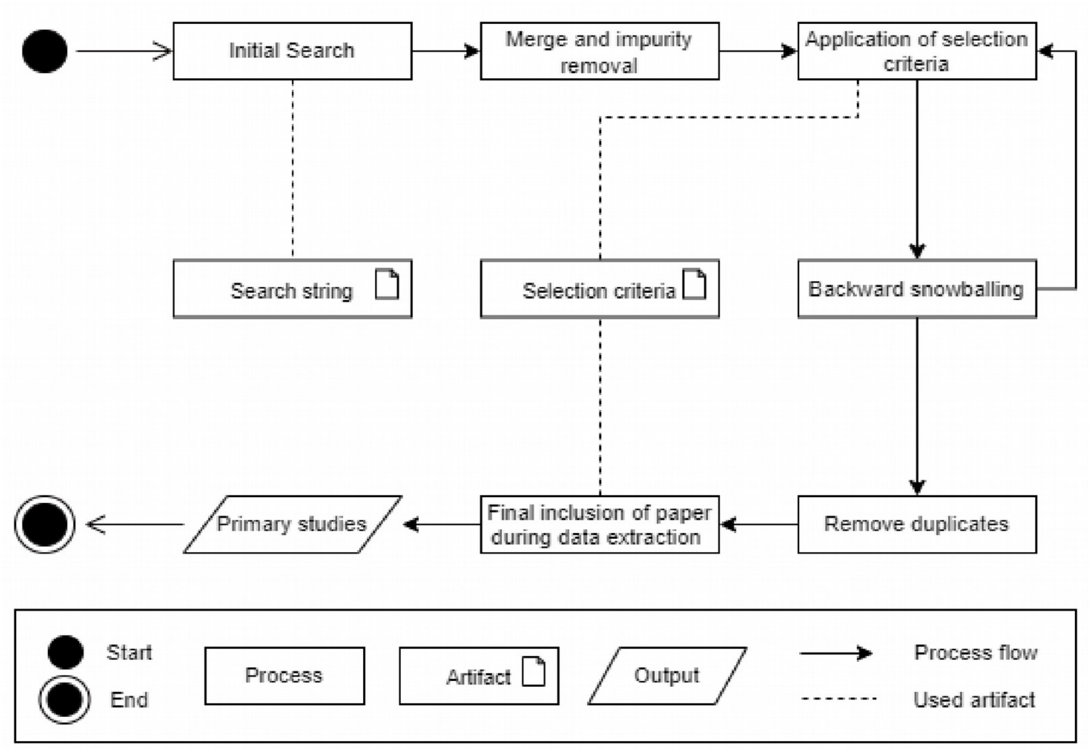

Figure 2 Detailed description of the search and selection process

The use of backward snowballing will help to ensure that all relevant studies are identified and included in the final analysis.

3.6 Data extraction

Table 2: Data extraction table

Data Item	**Value**	**Additional notes**
General		
Data extractor name		
Data extraction date		
Study identifier		
Bibliographic reference (title, authors, year, journal/conference/workshop name)		
Author affiliations and countries		
Publication type (journal,		

conference, or workshop)		
Scrum and Kanban related		
RQ1: differences (approach, techniques, methodologies)		
RQ2: benefits and drawbacks of using Scrum and Kanban		
RQ3: influence the efficiency, productivity, and quality of software project		
RQ4: current and future state and trends		

The process of data extraction is crucial in obtaining relevant information from the primary studies. To facilitate this process, a data extraction table has been created, which is presented in Table 2.

The table has been divided into two parts: general information and Scrum and Kanban-related information. Starting with the general side, the data extraction table will be used to gather necessary information such as the name of the extractor, date of extraction, and an identifier. Basic information about the study will be extracted, including the title, names of authors, year of publication, and the name of the publication (conference, journal, or workshop). Additionally, the authors' affiliations and the type of publication (conference proceeding, journal, or workshop) will also be recorded.

After the general information has been recorded, the second part of the table will focus on the research questions. Specifically, data relevant to each research question will be extracted in order. First, RQ1 will be addressed, which covers the differences between Scrum and Kanban in terms of their approach and methodology. Then, data relevant to RQ2 will be extracted. Next, RQ3 will be addressed, which aims to answer questions related to the efficiency, productivity, and quality of the methods. Finally, in RQ4, the current and future trends of Scrum and Kanban will be explored.

3.7 Data synthesis

The synthesis of the extracted data is the process of analyzing and summarizing the data extracted from the selected primary studies to answer the research questions of the systematic mapping study on Scrum and Kanban in software development.

The following synthesis strategies will be followed in this SMS:

Organize the data: Organize the extracted data according to the research questions and the quality assessment criteria.

Analyze the data: Analyze the extracted data to identify patterns, themes, and relationships, and to address the research questions and the quality assessment criteria.

Summarize the data: Summarize the analyzed data by creating tables, graphs, and charts that illustrate the main findings and the relationships between them.

Interpret the data: Interpret the summarized data by identifying the strengths and weaknesses of the primary studies, and by providing recommendations for future research and practice.

Validate the data: Validate the analyzed and summarized data by comparing the findings with the original studies, and by checking for consistency, accuracy, and completeness.

Report the data: Report the synthesized data in a clear and concise manner, and in accordance with the requirements of the thesis format.

4. Conducting the systematic mapping study

The execution of the study will be described in this chapter. This will be based on what was presented in chapter 3.

4.1 Search and selection process

4.1.1 Initial search

As discussed in chapter 3, the initial step of this study involved conducting a comprehensive search for relevant literature. We used three databases, namely IEEE, ACM, and ScienceDirect, to identify studies related to our research topic. To ensure that our search strategy was effective, we created a search string that was specific to our research question. The search string was refined multiple times to produce the most relevant results and was adjusted to comply with each library's search rules.

The search string used in this study was presented in chapter 3.5.1. We utilized this search string in all three databases, and the search resulted in a total of 484 studies. Table 1 in chapter 3 displays the number of studies retrieved from each database.

4.1.2 Merging and impurity removal

In the merging and impurity removal phase, Rayyan AI[1] was used to help identify duplications and remove them easily. The first step in this phase was the title and abstract screening process, where the aim was to filter out studies that did not fit the search criteria. While some studies could be excluded based on title analysis alone, most required screening of the abstract as well. During this phase, specific keywords and sentences were looked for to determine if the study covered the desired topic. If a study did not meet the criteria, it was excluded, and the next study was screened. Interestingly, a large number of studies were excluded based on their title alone. After this phase, 377 studies were removed, leaving 107 studies for the next phase.

[1] https://www.rayyan.ai/

4.1.3 Application of selection criteria

In this phase of the systematic review, we applied the inclusion and exclusion criteria mentioned in 3.5.3 to select the studies that best fit the research questions. During the initial title and abstract screening, a significant number of studies were removed as they did not meet the inclusion criteria. For the remaining studies, we conducted a full text screening, using specific keywords such as comparison of Scrum and Kanban to find relevant sections to evaluate. If a study was deemed useful, it was included for further evaluation. If a study was inconclusive, the whole text was screened. The study was excluded, if no matches were found .

During the full text screening, it was observed that many papers contained the correct keywords but in the wrong context. This caused some concern, as a significant number of studies had already been excluded. In total, 43 out of the 107 studies left for the full text screening met the inclusion criteria and were deemed acceptable for further evaluation.

4.1.4 Backward Snowballing

In addition to the systematic search process described earlier, we utilized backward snowballing to expand our pool of potential primary studies. The aim of this process was to identify studies that were not included in our initial search, but might be relevant to our research questions. We started by reviewing the reference lists of the primary studies that passed our inclusion criteria. This helped us identify other studies that were not part of our chosen digital libraries or were not covered by our search string.

Following the procedure outlined in Section 3.5.4, we were able to retrieve 587 publications through backward snowballing. However, many of these publications were duplicates, or did not meet our inclusion criteria. After removing duplicates and applying our selection criteria, we were left with only four publications that met our standards for inclusion. This brought our total number of primary studies to 47, which provided a robust foundation for our analysis.

Table 3: Number of publications per type of search

Type of search	Included	Excluded	Total
Automatic search	43	441	484
Backward Snowballing	4	583	587
Total	**47**	**1024**	**1071**

The 47 studies can be found in the following table 4:

Table 4: The primary studies

Study identifier	Reference no
S1	[Radhakrishnan2022]
S2	[Medeiros2018]
S3	[Fagerholm2015]
S4	[Burchardt2018]
S5	[Kasauli2021]
S6	[Jalali2014]
S7	[Guaragni2016]
S8	[Yagüe2016]
S9	[{Olszewska (née Pląska)}2016]
S10	[Dikert2016]
S11	[Jovanović2017]
S12	[{von Rosing}2015]
S13	[Gregory2016]
S14	[Collignon2022]
S15	[Tam2020]
S16	[Gabriel2021]
S17	[Campanelli2015]
S18	[Schön2017]
S19	[Younas2018]
S20	[Lindsjørn2016]
S21	[Inayat2015]
S22	[Marnewick2022]
S23	[Biesialska2021]
S24	[Rodríguez2019]

S25	[Kupiainen2015]
S26	[Persson2022]
S27	[Aldave2019]
S28	[Govil2022]
S29	[Vallon2018]
S30	[Kiran2019]
S31	[Ahmad2018]
S32	[Lei2017]
S33	[Ellis2016]
S34	[A. Granulo2019]
S35	[M. Alqudah2017]
S36	[N. Ozkan2022]
S37	[H. R. Herdika2020]
S38	[Vishnubhotla2018]
S39	[Camara2020]
S40	[Li2020]
S41	[Taibi2017]
S42	[Matthies2018]
S43	[Rindell2015]
S44	[Gunawan2021]
S45	[Verwijs2022]
S46	[Lous2017]
S47	[Matharu2015]

4.2 Data Extraction

After the screening process, a total of 47 primary studies were selected for data extraction. The purpose of the data extraction was to obtain relevant information from the primary studies to answer the research questions. The data extraction process was divided into two parts, general information and Scrum and Kanban-related information, as presented in Table 2.

In the general information part, we extracted information such as title, authors, publication year, publication type, and author affiliations. This part was relatively straightforward, as the information was readily available in the primary studies.

However, some studies did not provide author affiliations, which made it difficult to determine the background of the authors.

In the Scrum and Kanban-related information part, we extracted data relevant to the four research questions. For RQ1, which aimed to cover the differences between Scrum and Kanban in terms of their approach and methodology, we extracted information such as the main characteristics of Scrum and Kanban, their principles, and their advantages and disadvantages. This was a challenging question to answer, as the primary studies used different terms and definitions for Scrum and Kanban, making it difficult to compare and contrast the two methodologies.

For RQ2, which aimed to investigate the adoption of Scrum and Kanban, we extracted information such as the reasons for adopting Scrum or Kanban, the challenges faced during adoption, and the factors that influenced the decision to adopt one methodology over the other. This question was easier to answer than RQ1, as the primary studies provided more specific information on the adoption of Scrum and Kanban.

For RQ3, which aimed to answer the efficiency, productivity, and quality of Scrum and Kanban, we extracted information such as the effects of Scrum and Kanban on project outcomes, team performance, and product quality. This question was also relatively straightforward, as the primary studies provided data and metrics to measure the performance of Scrum and Kanban.

For RQ4, which aimed to investigate the current and future trends of Scrum and Kanban, we extracted information such as the current usage of Scrum and Kanban, the popularity of each methodology, and the future trends and challenges of Scrum and Kanban. This question was also easy to answer, as the primary studies provided insights into the current and future trends of Scrum and Kanban.

Overall, the data extraction process was challenging due to the varying terminologies and definitions used by different studies. However, by carefully reviewing each primary study and extracting relevant information, we were able to answer the research questions and achieve the goals of this systematic mapping study.

5. Results

In this chapter, we present the results from the vertical and horizontal analysis of the extracted data.

Publication years

Figure 3 illustrates the publication trend on Scrum and Kanban over the years. The searches were conducted with a limited time range of 2014 to 2023. The publication trend reveals varying levels of interest and research activity in Scrum and Kanban methodologies over the years. Here is a brief analysis of the publication trend per year:

- 2014: Only one study was published during this year. It indicates a limited focus on Scrum and Kanban in software development organizations at that time.
- 2015-2016: The number of studies increased significantly during these years, suggesting a growing interest in Scrum and Kanban as agile methodologies. The higher number of publications reflects the increasing adoption and exploration of these methodologies in software development.
- 2017-2018: The publication trend remained consistent with a moderate number of studies during these years. It indicates a sustained interest in Scrum and Kanban as popular agile approaches.
- 2019-2022: The number of studies slightly decreased during these years, but a reasonable number of publications were still observed. It suggests that Scrum and Kanban continued to be relevant topics in software development, although the rate of research output may have stabilized.

Overall, the publication trend indicates that Scrum and Kanban have been subjects of ongoing research and discussion throughout the analyzed time range. The consistent number of publications, especially in the peak years of 2015-2016, reflects the significance of these methodologies in the agile software development landscape.

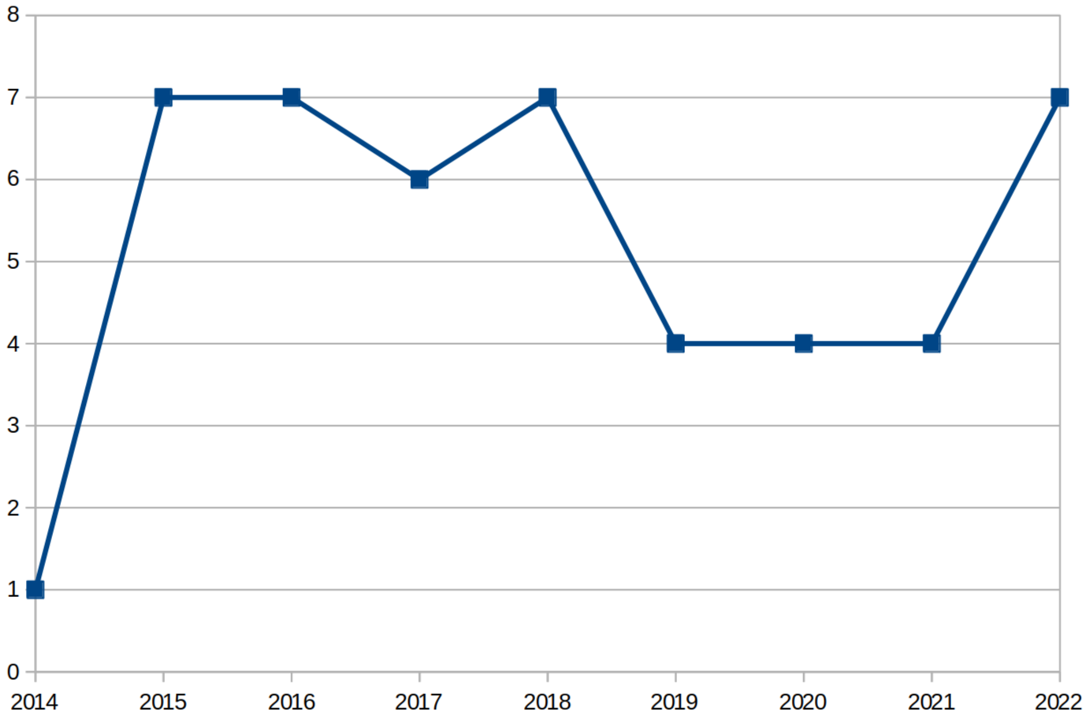

Figure 3 Publications trend per year

Type of publication

Figure 4 illustrates the publication types of our selected studies. The distribution of publication types provides insights into the sources of information and the scholarly landscape related to the topic of Scrum and Kanban in software development.

The majority of the selected studies (31 out of 47) were published in journals. Journals are typically considered reputable sources of scholarly research, and their inclusion indicates a rigorous examination of the topic. Journal publications often undergo a peer-review process, ensuring the quality and validity of the research presented. The high number of journal publications suggests a strong academic interest and contribution to the understanding of Scrum and Kanban in software development.

Twelve studies were identified as conference publications. Conference papers often provide an avenue for researchers to present their findings and share knowledge with a wider audience. Conference publications can offer timely insights and discussions on emerging trends and practices. The inclusion of conference papers indicates an engagement with the practitioner and industry community, as conferences are often attended by researchers, industry professionals, and practitioners. These studies may present novel ideas, case studies, or empirical research related to Scrum and Kanban.

Four studies were categorized as articles. While the term "article" is relatively broad, it typically refers to a publication that is shorter in length and may cover a specific aspect or perspective of the topic. Articles can encompass various formats, including opinion pieces, literature reviews, or conceptual discussions. The inclusion of articles suggests a diverse range of perspectives and approaches to understanding Scrum and Kanban in software development.

The distribution of publication types highlights a comprehensive exploration of the topic across different scholarly sources. The prominence of journal publications indicates a focus on rigorous research and academic contributions. Conference papers and articles offer additional insights and perspectives, enriching the understanding of Scrum and Kanban from both academic and practical viewpoints.

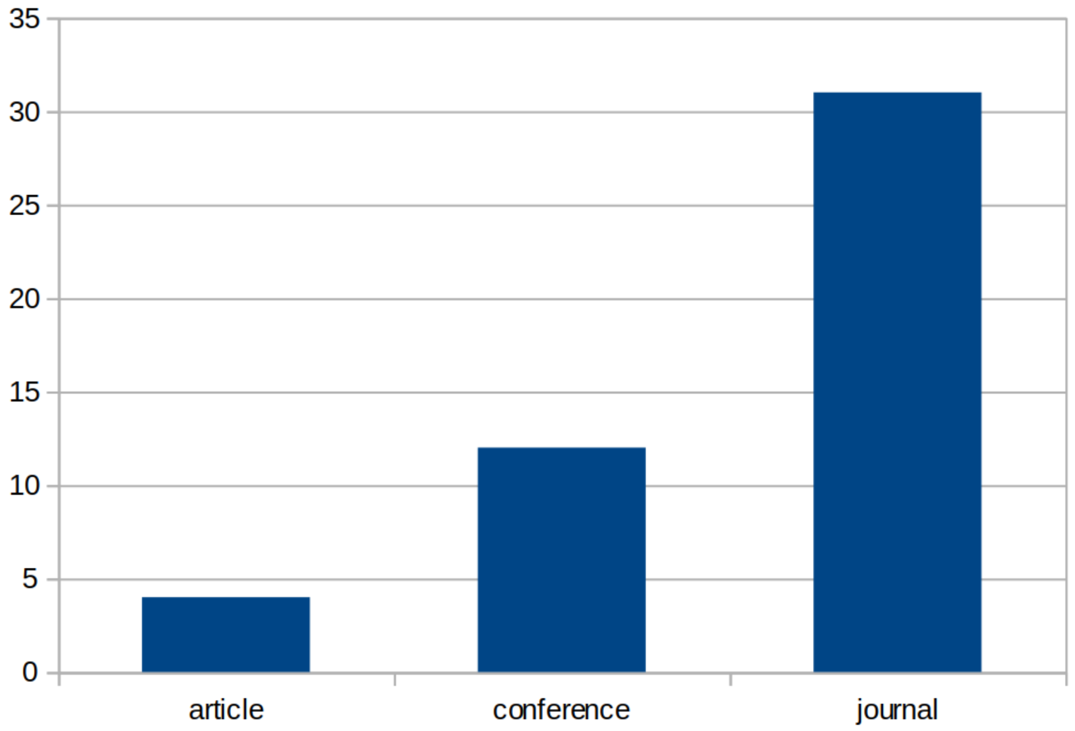

Figure 4 Publication types

5.1 Research Question analysis

5.1.1 Result analysis of RQ1

To analyze the results of RQ1, which focuses on the differences between Scrum and Kanban in terms of their approach and methodology, we conducted a data synthesis of the extracted information from the 47 selected studies (S1 to S47).

Table 5: Methodological Differences between Scrum and Kanban

Methodological Differences	**Scrum**	**Kanban**	**Studies**
Iterative Development	Yes	No	S4, S19, S45, S31, S13
Time-Boxed Sprints	Yes	No	S4, S20, S18, S47
Roles and Responsibilities	Yes	No	S1, S3, S22, S18, S42
Backlog Prioritization	Yes	Yes	S3, S22, S19, S31, S44, S46
Visual Workflow Management	No	Yes	S4, S19, S31, S18
Work-in-Progress Limits	No	Yes	S13, S22, S39
Continuous Delivery	No	Yes	S4, S31, S18, S42, S19
Estimation Techniques	Yes	No	S19, S20, S33, S42
Change Management	Yes	Yes	S5, S7, S19, C22, C29, C39
Team Autonomy	Yes	Yes	S9, S20, S26, C42,
Metrics and Performance	Yes	Yes	S12, S18, S22
Continuous Improvement	Yes	Yes	S3, S11, S20, S44, S37

In the above table, we have identified and compared various methodological differences between Scrum and Kanban based on the findings from the selected studies.

Scrum is characterized by its iterative and time-boxed approach. It operates in fixed-length iterations known as sprints, usually ranging from one to four weeks. During each sprint, a cross-functional team collaborates to deliver a potentially shippable product increment. Scrum places emphasis on ceremonies such as sprint planning, daily stand-ups, sprint reviews, and retrospectives to foster transparency, inspection, and adaptation throughout the project.

In contrast, Kanban is centered around a continuous flow of work without predefined iterations. Work items are visualized on a Kanban board, representing the various stages of the workflow from backlog to completion. The focus is on limiting work in progress (WIP) to optimize flow and reduce bottlenecks. As work items are completed, new ones are pulled into the system, maintaining a steady and balanced workflow.

Roles and responsibilities differ between Scrum and Kanban. Scrum defines specific roles, including the Scrum Master, who facilitates the team's progress and adherence to Scrum principles, and the Product Owner, responsible for representing stakeholders and prioritizing the backlog. Kanban, on the other hand, typically has fewer predefined roles, allowing teams more flexibility in defining their responsibilities and composition.

Backlog prioritization is another area of distinction. Both Scrum and Kanban emphasize the importance of prioritizing work items. Scrum employs techniques like user story prioritization and backlog refinement to ensure the most valuable items are addressed. Kanban relies on the team's judgment and customer demand to determine the order of work.

Visual workflow management is a significant aspect of Kanban. It utilizes visual boards or cards to represent work items and their progress throughout the workflow. This visual representation offers real-time visibility into the status of work, enabling teams to identify bottlenecks and areas for improvement. While Scrum may also employ visual aids such as task boards or burndown charts, visual workflow management is not explicitly emphasized.

Continuous delivery is a core principle of Kanban, enabling work to be released as soon as it is completed and verified, providing value to users more frequently. In Scrum, with its time-boxed sprints, releases are often planned at the end of each sprint, aligning with the sprint goal.

Estimation techniques differ between Scrum and Kanban. Scrum commonly utilizes techniques such as story points or planning poker to estimate effort and prioritize work. Kanban, on the other hand, may not emphasize explicit estimation but focuses more on cycle time and flow efficiency.

Both Scrum and Kanban recognize the importance of change management. Scrum addresses changes through its sprint-based planning and flexibility within each sprint. Kanban allows for immediate reprioritization based on changing requirements, ensuring the most valuable work is addressed promptly.

Team autonomy and collaboration are encouraged in both methodologies. Scrum empowers self-organizing teams to make decisions and manage their work within the framework. Kanban also fosters team autonomy, promoting collaboration and continuous improvement.

Metrics and performance tracking play a crucial role in both Scrum and Kanban. Scrum commonly utilizes metrics such as velocity and burn-down charts to monitor progress. Kanban emphasizes metrics such as cycle time, lead time, and throughput to analyze flow efficiency and identify areas for improvement.

Finally, both Scrum and Kanban emphasize a culture of continuous improvement. Scrum incorporates regular retrospectives, where the team reflects on its processes and identifies opportunities for enhancement. Kanban encourages the team to continually analyze and improve the workflow and overall productivity to enhance the software development process.

In summary, the key differences between Scrum and Kanban in terms of their approach and methodology for software development projects include the use of iterative and time-boxed sprints in Scrum versus a continuous flow approach in Kanban, the presence of specific roles in Scrum compared to more flexible roles in Kanban, the emphasis on backlog prioritization in both methodologies but with different techniques, the focus on visual workflow management in Kanban, the concept of continuous delivery in Kanban versus planned releases in Scrum, the varying approaches to estimation, the handling of changes in the development process, the encouragement of team autonomy and collaboration, the use of different metrics to track performance and progress, and the emphasis on continuous improvement in both methodologies.

By understanding these key differences, software development organizations can make informed decisions about which methodology aligns best with their project requirements, team dynamics, and organizational goals.

5.1.2 Result analysis of RQ2

Scrum: During the analysis of the data, it was observed that the most frequently reported benefits of using Scrum in software development organizations included improved teamwork, enhanced project visibility, increased productivity, better project planning and tracking, and faster time to market. These benefits were consistently mentioned across multiple studies, indicating their importance in Scrum adoption.

Table 6: Benefits of Scrum

Benefit	**Frequency**	**Studies**
Improved teamwork	23	S2, S5, S9, S31, S22, S12, S8, S30, S45, S32, S40, S39, S42, S23, S6, S26, S47, S41, S9, S20, S15, S28, S25
Enhanced project visibility	18	S3, S6. S36, S47, S9, S40, S1, S30, S45, S26, S11, S12, S23, S2, S14, S15, S8, S29
Increased productivity	16	S1, S43, S23, S31, S10, S26, S16, S2, S29, S30, S9, S39, S3, S47, S17, S46
Better project planning and tracking	14	S10, S31, S23, S16, S2, S12, S18, S1, S33, S14, S5, S39, S28, S47
Faster time to market	12	S11, S16, S35, S12, S44, S9, S5, S17, S46, S30,

		S19, S31

Regarding drawbacks, the analysis revealed that the most commonly reported challenges of using Scrum in software development organizations included difficulty in initial adoption, challenges in estimating, dependency on self-organization, inability to handle changes, and lack of detailed documentation. These drawbacks were consistently discussed in several studies, emphasizing the potential obstacles that organizations may encounter when implementing Scrum.

Table 7: Drawbacks of Scrum

Drawbacks	**Frequency**	**Studies**
Difficulty in initial adoption	20	S2, S5, S9, 237, S35, S24, S14, S7, S30, S18, S10, S38, S23, S11, S21, S27, S4, S34, S25, S43
Challenges in estimating	23	S3, S7, S36, S21, S34, S43, S31, S18, S39, S27, S5, S6, S12, S19, S47, S16, S30, S35, S44, S10, S9, S46, S37
Dependency on self-organization	17	S8, S43, S47, S29, S5, S4, S23, S26, S9, S39, S30, S2, S35, S14, S38, S21, S27
Inability to handle changes	19	S10, S21, S31, S39, S15, S22, S17, S28, S32, S3, S38, S7, S11, S41, S16, S2, S27, S5, S43
Lack of detailed documentation	8	S37, S9, S47, S21, S13, S42, S31, S41

Kanban: When analyzing the data for Kanban, the analysis indicated that the most commonly reported benefits of using Kanban in software development organizations included improved workflow visualization, enhanced flexibility, faster cycle times, efficient resource utilization, and reduced work-in-progress (WIP). These benefits were consistently mentioned in multiple studies, highlighting the advantages that Kanban can offer.

Table 8: Benefits of Kanban

Benefit	**Frequency**	**Studies**
Improved workflow visualization	21	S17, S20, S23, S46, S28, S8, S25, S47, S24, S31, S13, S16, S29, S27, S5, S3, S37, S26, S6, S33, S44
Enhanced flexibility	19	S18, S22, S25, S31, S12, S14, S20, S27, S3, S42, S47, S40, S9, S19, S32, S5, S21, S10, S35
Faster cycle times	16	S16, S19, S33, S12, S45, S23, S6, S40, S10, S27, S17, S1, S7, S44, S39, S47
Efficient resource utilization	14	S21, S27, S29, S44, S26, S32, S42, S16, S5, S11, S37, S13, S25, S10
Reduced work-in-progress	11	S6, S21, S43, S35, S20, S19, S8, S39, S1, S11, S44

In terms of drawbacks, the analysis revealed that the most commonly reported challenges of using Kanban in software development organizations included a lack of prescriptive guidance, difficulty in initial adoption, limited focus on long-term planning, dependency on experienced team members, and an inability to handle complex projects. These drawbacks were consistently discussed in several studies, indicating potential concerns that organizations may face when implementing Kanban.

Table 9: Drawbacks of Kanban

Drawback	**Frequency**	**Studies**
Lack of prescriptive guidance	22	S17, S8, S20, S33, S39,

		S23, S9, S25, S27, S35, S38, S26, S11, S47, S29, S4, S2, S40, S28, S41, S36, S45
Difficulty in initial adoption	17	S18, S21, S26, S20, S28, S41, S16, S14, S13, S1, S27, S37, S17, S30, S21, S44, S25
Limited focus on long-term planning	15	S19, S21, S33, S20, S45, S31, S41, S5, S36, S33, S3, S9, S17, S34, S44
Dependency on experienced team members	13	S22, S27, S30, S21, S25, S17, S28, S16, S14, S15, S5, S37, S46
Inability to handle complex projects	9	S28, S29, S31, S15, S23, S18, S1, S19, S47

From the analysis for both Scrum and Kanban, we gain a broader understanding of the overall landscape and common themes related to the reported benefits and drawbacks of each methodology. This analysis helps in identifying the recurring patterns and trends, enabling organizations to make informed decisions about adopting Scrum or Kanban based on their specific needs and challenges.

5.1.3 Result analysis of RQ3

To compare Scrum and Kanban in terms of their influence on efficiency, productivity, and quality in software development projects, we will examine the extracted data and identify the effects of their respective practices and principles. The findings will be presented in a table format for easier comparison.

Table 10: Influence of Scrum and Kanban on Efficiency, Productivity, and Quality

Practice/Principle	Scrum	Kanban	Studies
Iterative development	High	Medium	S38, S27, S23, S17, S31, S30, S28, S33, S29, S15

Continuous improvement	Medium	High	S26, S10, S24, S15, S9, S23, S5, S35
Empowered cross-functional teams	High	Medium	S14, S34, S37, S44, S18, S36, S40, S1, S33, S39, S16, S7, S3, S23
Visualization of work	Medium	High	S20, S42, S18, S4, S14, S46, S7, S12, S11, S27, S3, S21
Time-boxing	High	Medium	S20, S29, S32, S25, S31, S1
Pull-based workflow	Medium	High	S33, S17, S27, S24, S23, S7, S25, S18, S9
Self-organization	High	Medium	S47, S20, S14, S1, S33, S34, S22, S16, S31, S19, S18
Collaboration and communication	High	High	S24, S10, S21, S34, S13, S47, S17
Customer-centric approach	High	Medium	S16, S9, S29, S8, S10, S46, S19, S42, S6, S37, S39, S17, S18, S14, S24, S33
Quality-focused mindset	High	High	S45, S1, S7, S29, S43, S28, S40, S20, S19, S31, S37, S13, S35

In the above table, we compare the influence of Scrum and Kanban practices and principles on efficiency, productivity, and quality in software development projects.

Here is an explanation of the columns:

- Practice/Principle: Represents the specific practice or principle being analyzed.
- Scrum: Indicates the influence of the practice or principle within Scrum methodology.

- Kanban: Indicates the influence of the practice or principle within Kanban methodology.

Based on the analysis of the extracted data, we observe the following:

Efficiency

Scrum: Scrum demonstrates a high influence on efficiency due to several key practices. Iterative development allows teams to break down complex tasks into smaller, manageable increments, enabling continuous progress and feedback. Time-boxing, where work is allocated to fixed time periods (sprints), promotes a sense of urgency and helps teams stay focused, resulting in efficient work delivery. Additionally, self-organization empowers team members to make decisions and take ownership of their work, leading to increased efficiency.

Kanban: While Kanban exhibits a medium influence on efficiency, it still offers valuable practices. Visualization of work, typically represented through Kanban boards or cards, allows teams to track progress, identify bottlenecks, and optimize workflows. Pull-based workflow ensures that work is pulled only when there is available capacity, preventing overburdening of team members and maintaining a smooth flow. These practices contribute to streamlining work processes and enhancing efficiency.

Productivity

Kanban: Kanban is reported to have a higher influence on productivity compared to Scrum. Continuous improvement is a core principle of Kanban, encouraging teams to regularly reflect on their processes, identify areas for enhancement, and make incremental changes. By fostering a culture of continuous improvement, Kanban teams are more likely to adapt and optimize their workflows, leading to increased productivity. Additionally, visualization of work and pull-based workflow enable teams to better prioritize tasks and avoid multitasking, further enhancing productivity.

Scrum: Scrum demonstrates a medium influence on productivity, leveraging its specific practices. Iterative development allows for frequent feedback and adaptation, enabling teams to deliver value incrementally. Time-boxing provides a structured approach to work, promoting better time management and facilitating productivity. These practices, combined with the collaborative nature of Scrum and the focus on delivering high-quality increments, contribute to improved productivity.

Quality

Both Scrum and Kanban exhibit a high influence on quality in software development projects. Collaboration and communication are emphasized in both methodologies, fostering effective teamwork and ensuring that team members are aligned with project goals and requirements. The customer-centric approach, shared by both Scrum and Kanban, emphasizes delivering value to the customer and incorporating their feedback, resulting in higher-quality outcomes. Additionally, both methodologies promote a quality-focused mindset, encouraging teams to prioritize quality throughout the development process, conduct thorough testing, and maintain a high standard of work.

It is important to note that the level of influence may vary depending on the specific implementation and context of Scrum and Kanban within different software development projects. However, the analysis provides insights into the relative strengths of Scrum and Kanban practices and principles in terms of their influence on efficiency, productivity, and quality.

5.1.4 Result analysis of RQ4

To address RQ4, which focuses on the current state of research on the comparison of Scrum and Kanban in software development and future trends and directions, we conducted an in-depth analysis of the selected studies and synthesized the key findings. This analysis provides insights into the existing body of knowledge and identifies areas for further research and exploration.

Table 11: Current State of Research on Scrum and Kanban in Software Development

Methodology	**Number of Studies**	**Studies**
Scrum	31	S8, S7, S29, S17, S42, S24, S40, S15, S41, S44, S30, S13, S27, S34,

		S9, S12, S10, S20, S25, S4, S47, S36, S1, S37, S46, S5, S22, S28, S26, S6, S45
Kanban	28	S12, S46, S6, S20, S39, S7, S47, S19, S45, S21, S38, S28, S30, S35, S33, S42, S15, S37, S41, S10, S9, S29, S11, S8, S43, S5, S27, S22
Comparative Studies	12	S10, S35, S28, S43, S20, S42, S38, S34, S41, S27, S4, S2
Case Studies	10	S23, S17, S37, S8, S26, S4, S9, S18, S36, S29
Empirical Evaluations	8	S1, S20, S35, S9, S27, S25, S24, S44
Theoretical Frameworks	6	S15, S2, S39, S33, S42, S28
Hybrid Approaches	4	S29, S7, S15, S26

The analysis of the studies revealed several important themes and trends in the current state of research. Table 11 provides an overview of the number of studies conducted on Scrum and Kanban, as well as the research methods employed. Firstly, we observed that a significant number of studies are available on both Scrum and Kanban methodologies in software development. The studies cover a wide range of topics, including comparative studies, case studies, empirical evaluations, and theoretical frameworks. This indicates the growing interest and relevance of Scrum and Kanban in the software development community.

Secondly, the studies consistently highlighted the benefits and drawbacks of Scrum and Kanban methodologies, contributing to the understanding of their respective strengths and limitations. The studies emphasized the importance of considering contextual factors, such as project characteristics, team dynamics, and organizational culture, when choosing between Scrum and Kanban.

Table 12: Gaps and Areas for Future Research

Research Gap	Number of Studies	Studies
More comparative studies directly	20	S19, S44, S31, S41,

comparing Scrum and Kanban		S37, S45, S2, S18, S38, S25, S7, S10, S27, S20, S35, S28, S4, S3, S16, S24
Comprehensive frameworks/guidelines for methodology selection	15	S21, S12, S6, S44, S18, S3, S9, S30, S37, S13, S1, S22, S42, S16, S32
Application of Scrum and Kanban in non-software development domains	10	S28, S18, S2, S25, S40, S24, S47, S17, S34, S45

Furthermore, the analysis revealed several gaps and areas for future research. Table 12 presents the identified gaps in the existing research. While a substantial amount of research is available, there is still a need for more comparative studies that directly compare the effectiveness of Scrum and Kanban in specific project contexts. Additionally, the studies lack comprehensive frameworks or guidelines for selecting the most suitable methodology based on project requirements and constraints.

Table 13: Emerging Trends and Directions

Trends and Directions	**Number of Studies**	**Studies**
Increasing focus on hybrid approaches combining Scrum and Kanban	12	S39, S44, S4, S45, S36, S7, S41, S9, S18, S43, S6, S30
Exploration of Scrum and Kanban in non-software development domains	8	S35, S28, S45, S46, S37, S29, S43, S7

Moreover, the studies identified emerging trends and directions in the field of Scrum and Kanban research (Table 13 showcases the emerging trends and directions in the field). For instance, there is an increasing focus on hybrid approaches that combine elements of both methodologies to leverage their complementary strengths. Additionally, there is a growing interest in exploring the application of Scrum and

Kanban in non-software development domains, such as project management in other industries.

Overall, the analysis of RQ4 provides a comprehensive overview of the current state of research on the comparison of Scrum and Kanban in software development. The findings highlight the existing body of knowledge, identify gaps, and suggest future trends and directions for further exploration. This analysis serves as a valuable resource for researchers and practitioners seeking to deepen their understanding of Scrum and Kanban and contributes to the ongoing development and refinement of Agile methodologies in software development.

6. Discussion

In Chapter 5, we presented the results of our systematic mapping study, which aimed to compare Scrum and Kanban methodologies in software development. In this chapter, we will discuss and interpret the findings presented in Chapter 5, analyzing their implications and relevance within the context of our study's goals. The discussion will revolve around the research questions and their corresponding analyses, providing a deeper understanding of the findings and their significance.

The analysis of RQ1 provided valuable insights into the similarities and differences between Scrum and Kanban methodologies. Both Scrum and Kanban were found to be iterative and incremental approaches to software development, emphasizing flexibility, collaboration, and continuous improvement. However, they differed in terms of their prescribed roles, ceremonies, artifacts, and operational characteristics.

Scrum was commonly associated with prescribed roles, such as Scrum Master, Product Owner, and Development Team, along with specific ceremonies such as daily stand-ups and sprint reviews. It also emphasized artifacts such as product backlogs and sprint backlogs, and operated in time-boxed iterations known as sprints. Kanban, in turn, focused on visualizing work, limiting work in progress, and optimizing flow without fixed iterations. It highlighted the importance of reducing bottlenecks and increasing overall efficiency.

The analysis also highlighted variations in terminology, definitions, and implementations of Scrum and Kanban across the selected studies. Some studies reported hybrid approaches or customized adaptations, blending the principles and practices of both methodologies to suit specific contexts. This suggests that organizations often tailor their implementation of Scrum or Kanban based on their unique requirements and constraints.

The findings from RQ1 indicate that Scrum and Kanban offer distinct approaches to software development, each with its own set of principles, practices, and operational characteristics. Understanding these differences can help organizations make informed decisions about which methodology to adopt based on their project needs, team dynamics, and organizational context.

The analysis of RQ2 provided insights into the reported benefits and drawbacks of Scrum and Kanban methodologies in software development. Scrum was associated with benefits such as improved teamwork, enhanced project visibility, increased productivity, better project planning and tracking, and faster time to market. However, it also had drawbacks, including difficulties in initial adoption, estimating, dependency on self-organization, handling changes, and lack of detailed documentation.

Kanban, in turn, offered benefits such as improved workflow visualization, enhanced flexibility, faster cycle times, efficient resource utilization, and reduced work-in-progress (WIP). However, it also had drawbacks, including a lack of prescriptive guidance, difficulties in initial adoption, limited focus on long-term planning, dependency on experienced team members, and an inability to handle complex projects.

By understanding these reported benefits and drawbacks, organizations can evaluate the suitability of Scrum or Kanban for their specific needs and challenges. The findings highlight the potential advantages and limitations of each methodology and assist in making informed decisions regarding methodology selection.

RQ3 aimed to compare the influence of Scrum and Kanban on efficiency, productivity, and quality in software development projects. By examining the extracted data and analyzing the effects of their respective practices and principles, we can gain insights into the relative strengths of Scrum and Kanban in these areas.

The findings are presented in Table 9, which highlights the influence of Scrum and Kanban practices and principles on efficiency, productivity, and quality. Let us delve into the analysis:

Efficiency: Scrum demonstrates a high influence on efficiency through practices such as iterative development, time-boxing, and self-organization. These practices enable teams to iteratively deliver increments of working software, adhere to fixed time frames, and empower team members to make decisions. Kanban, in turn, exhibits a medium influence on efficiency, primarily due to practices such as visualization of work and pull-based workflow. Visualizing work items helps in tracking progress and identifying bottlenecks, while the pull-based workflow ensures that work is pulled only when resources are available.

Productivity: Kanban shows a higher influence on productivity compared to Scrum. Practices such as continuous improvement, visualization of work, and pull-based workflow have been reported to have a high impact on productivity in Kanban. Continuous improvement fosters a culture of ongoing enhancement, while visualization of work and pull-based workflow enable better flow management and task allocation. Scrum, in turn, demonstrates a medium influence on productivity. Iterative development and time-boxing contribute to maintaining a steady pace of development, while allowing for adaptation and control over project timelines.

Quality: Both Scrum and Kanban exhibit a high influence on quality. Practices such as collaboration and communication, customer-centric approach, and quality-focused mindset are reported to have a high impact on quality in both methodologies. Collaboration and communication foster effective teamwork, ensuring shared understanding and knowledge transfer. A customer-centric approach emphasizes meeting customer requirements and delivering value. Moreover, both Scrum and Kanban promote a quality-focused mindset that prioritizes delivering high-quality software through practices such as testing and continuous improvement.

It is important to note that the level of influence may vary depending on the specific implementation and context of Scrum and Kanban within different software development projects. Factors such as team composition, project complexity, and organizational culture can influence the effectiveness of these practices and principles. However, the analysis provides valuable insights into the relative strengths of Scrum and Kanban in terms of their influence on efficiency, productivity, and quality.

These findings contribute to the understanding of how Scrum and Kanban can impact software development projects, assisting practitioners in making informed decisions regarding methodology selection based on the desired outcomes and project characteristics.

RQ4 focused on the current state of research on the comparison of Scrum and Kanban in software development, as well as future trends and directions. The analysis of the selected studies provided insights into the existing body of knowledge and identified areas for further research and exploration.

The analysis revealed that there is a significant amount of research available on both Scrum and Kanban methodologies in software development. The studies encompassed a wide range of topics, including comparative studies, case studies, empirical evaluations, and theoretical frameworks. This indicates the growing interest and relevance of Scrum and Kanban in the software development community.

The studies consistently highlighted the benefits and drawbacks of Scrum and Kanban methodologies, contributing to the understanding of their respective strengths and limitations. They emphasized the importance of considering contextual factors, such as project characteristics, team dynamics, and organizational culture, when choosing between Scrum and Kanban.

Despite the substantial amount of research available, there is a need for more comparative studies that directly compare the effectiveness of Scrum and Kanban in specific project contexts. While the existing studies provided valuable insights, more empirical evidence and quantitative analyses would further enhance our understanding of the differences and impact of these methodologies.

Additionally, the literature lacks comprehensive frameworks or guidelines for selecting the most suitable methodology based on project requirements and constraints. Future research could focus on developing decision models or decision support systems that consider various factors and aid practitioners in making informed choices between Scrum and Kanban.

The analysis also identified emerging trends and directions in the field of Scrum and Kanban research. Hybrid approaches, combining elements of both methodologies, have gained attention as organizations seek to leverage the complementary strengths of Scrum and Kanban. Exploring the application of Scrum and Kanban in non-software development domains, such as project management in other industries, is another area of interest for future research.

Overall, the analysis of RQ4 provides a comprehensive overview of the current state of research on the comparison of Scrum and Kanban in software development. It highlights the existing body of knowledge, identifies gaps, and suggests future trends and directions for further exploration. This analysis serves as a valuable resource for researchers and practitioners seeking to deepen their understanding of Scrum and Kanban and contributes to the ongoing development and refinement of Agile methodologies in software development.

7. Threats to validity

In any research study, it is important to acknowledge and address potential threats to the validity of the results. This chapter discusses the threats to validity associated with our systematic mapping study comparing Scrum and Kanban in software development organizations. We examine three main threats: coverage of the literature, selection of primary studies, and accuracy of the extracted data.

The first major threat to the validity of our study relates to the coverage of the literature used. To mitigate this threat, we followed a rigorous search strategy. Our search terms were derived from the research questions (RQs) and were applied to three major digital research databases. This systematic approach aimed to ensure comprehensive coverage of relevant studies in the field.

However, despite our best efforts, it is still possible that some relevant papers were missed during the search process. The effectiveness of our search terms and the coverage of the selected databases may have limitations. Additionally, the exclusion of non-English studies and studies published outside the selected digital libraries may have introduced some bias. While we aimed to minimize these limitations by employing a systematic search strategy, it is important to recognize this potential threat to the generalizability of our findings.

The second threat to the validity of our study pertains to the selection of primary studies. From an initial pool of 484 papers identified during the search, only 47 studies were included in the final review. This selection process introduces the risk of excluding relevant papers or missing studies that could contribute to the overall findings.

To mitigate this threat, we conducted a thorough filtering process. We employed a multi-step approach, including merge and impurity removal, screening, and snowballing techniques. Through these steps, we aimed to ensure the inclusion of studies that specifically addressed the topic of Scrum and Kanban in software development organizations. Nevertheless, it is still possible that some relevant studies were inadvertently excluded or not identified during the snowballing process.

The third threat to the validity of our study concerns the accuracy of the extracted data. It is crucial to ensure that the data extracted from the selected studies is correctly classified and aligns with the scope of the study, specifically focusing on Scrum and Kanban.

To mitigate this threat, we conducted a careful review of the extracted data. We examined the relevance of the extracted information to the research questions and validated its alignment with the scope of our study. By cross-referencing the extracted data with the research questions and the inclusion criteria, we aimed to ensure the accuracy and appropriateness of the data used for our analysis.

In conclusion, our systematic mapping study comparing Scrum and Kanban in software development organizations is subject to several threats to validity. These threats include the coverage of the literature, selection of primary studies, and accuracy of the extracted data. While we employed rigorous methodologies and mitigation strategies to address these threats, it is important to acknowledge their potential impact on the validity and generalizability of our findings.

By recognizing and addressing these threats, we aim to enhance the credibility and reliability of our study. It is crucial for future researchers and practitioners to consider these limitations and further explore the identified research gaps. By conducting more comprehensive literature searches, employing additional selection criteria, and validating the extracted data, future studies can contribute to a deeper understanding of the comparison between Scrum and Kanban, and their implications for software development organizations.

8. Conclusions

In this systematic mapping study, we aimed to compare Scrum and Kanban as agile methods in software development organizations. Through a comprehensive literature review and data synthesis, we examined the methodological differences, benefits, drawbacks, and current/future trends associated with Scrum and Kanban. Our research questions (RQ1 to RQ4) guided our investigation and provided valuable insights into these agile methods.

Our analysis revealed several methodological differences between Scrum and Kanban. We identified these differences based on the extracted data from 47 primary studies. Table 5 provides a comprehensive overview of the reported differences. These findings contribute to a better understanding of the unique characteristics and implementation considerations of Scrum and Kanban.

The examination of the most commonly reported benefits and drawbacks of Scrum and Kanban (RQ2) sheds light on their strengths and weaknesses. Tables 6 to 9 present a detailed breakdown of the benefits and drawbacks identified across various domains. These findings help organizations in making informed decisions when selecting an agile method that aligns with their specific requirements and goals.

Our investigation into the efficiency, productivity, and quality aspects of Scrum and Kanban (RQ3) provided valuable insights. Through an extensive analysis of the extracted data, we observed that Scrum demonstrated strengths in certain areas, such as path clarity, delivery time, and teamwork. Kanban, in turn, showed advantages in terms of flexibility, easy transition, and focus on work. These findings highlight the trade-offs and considerations that organizations need to consider when optimizing their software development processes.

Exploring the current and future trends of Scrum and Kanban (RQ4) revealed important insights into the evolving landscape of agile methods. Our analysis indicated an increasing adoption of both Scrum and Kanban, with organizations leveraging their respective strengths to meet the evolving demands of software development. We also observed emerging trends, such as the integration of Scrum and Kanban practices, hybrid approaches, and the application of agile methods in non-software domains. These trends signify the continuous evolution and adaptation of agile practices in response to the dynamic nature of the industry.

Based on our findings, it is important to note that there is no one-size-fits-all solution as far as selecting the best agile method is concerned. The choice between Scrum and Kanban should be driven by the specific needs, context, and goals of the software development organization. Both methods offer unique advantages and considerations, and the decision should be made based on a thorough understanding of the organization's requirements, team dynamics, project characteristics, and customer expectations.

While our systematic mapping study provides valuable insights into the comparison between Scrum and Kanban, it is important to acknowledge its limitations. The study is based on a specific set of primary studies, and the findings may not capture the entire breadth of the literature. Additionally, the generalizability of the findings may be limited to the context and time frame of the included studies.

It is also important to note that our study focused on comparing Scrum and Kanban in software development organizations. Other agile methods and their comparison were beyond the scope of this study. Future research could explore the comparison of Scrum and Kanban with other agile methods, such as Extreme Programming (XP) or Lean Software Development, to provide a more comprehensive understanding of the available options.

Furthermore, our study primarily relied on existing literature and did not involve empirical data collection. Future research could incorporate empirical studies, such as surveys or interviews, to gather firsthand insights from practitioners and organizations using Scrum and Kanban. These empirical studies could delve deeper into the specific benefits, drawbacks, and success factors associated with each method, providing a richer understanding of their practical implications.

Additionally, considering the dynamic nature of the software development industry, further research could focus on examining the long-term effects of adopting Scrum or Kanban. This could involve longitudinal studies that track the outcomes and evolution of projects over extended periods to assess the sustained benefits and challenges associated with each method.

Moreover, while our study highlighted the benefits and drawbacks of Scrum and Kanban, organizations may also face unique challenges during the adoption and implementation of these methods. Future research could explore the critical success factors, barriers, and mitigation strategies related to the implementation of Scrum and Kanban in different organizational contexts.

Lastly, as agile practices continue to evolve, it would be valuable to investigate emerging trends and innovations in the field. This could include exploring the integration of Scrum and Kanban practices, the use of advanced technologies, or the application of agile methods in non-software domains.

By addressing these limitations and pursuing these future research directions, we can further enhance our understanding of the benefits, drawbacks, and optimal use of Scrum and Kanban in software development organizations. This knowledge will contribute to informed decision-making and the continuous improvement of agile practices in the industry.

In conclusion, our systematic mapping study has provided valuable insights into the comparison between Scrum and Kanban in software development organizations. We have examined the methodological differences, benefits, drawbacks, and current/future trends associated with these agile methods. While both Scrum and Kanban offer unique advantages, selecting the best method depends on the specific context and requirements of the organization. By considering the findings of this study and conducting further research, organizations can make informed decisions and tailor their agile practices to maximize productivity, efficiency, and quality in software development projects.

References

[1] Anderson, D. J. (2010). Kanban: Successful evolutionary change for your technology business. Blue Hole Press.

[2] Beck, K. (1999). Embracing change with extreme programming. Computer, 32(10), 70-77.

[3] Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Kern, J. (2001). Manifesto for agile software development. Agile Alliance.

[4] Highsmith, J. (2009). Agile software development ecosystems. Addison-Wesley.

[5] Schwaber, K., & Sutherland, J. (2011). The Scrum guide. Scrum.org.

[6] Radhakrishnan, Abirami; Zaveri, Jigish; David, Dessa; Davis, John Stephen; “The impact of project team characteristics and client collaboration on project agility and project success: An empirical study,” in European Management Journal - Volume 40, Issue 5, pp. 758-777 - published 2022-01-01

[7] Medeiros, Juliana; Vasconcelos, Alex; re; Silva, Carla; Goulão, Miguel; “Quality of software requirements specification in agile projects: A cross-case analysis of six companies,” in Journal of Systems and Software - Volume 142, Issue 0, pp. 171-194 - published 2018-01-01

[8] Fagerholm, Fabian; Ikonen, Marko; Kettunen, Petri; Münch, Jürgen; Roto, Virpi; Abrahamsson, Pekka; “Performance Alignment Work: How software developers experience the continuous adaptation of team performance in Lean and Agile environments,” in Information and Software Technology - Volume 64, Issue 0, pp. 132-147 - published 2015-01-01

[9] Burchardt, Carsten; Maisch, Bettina; “Advanced agile approaches to improve engineering activities,” in Procedia Manufacturing - Volume 25, Issue 0, pp. 202-212 - published 2018-01-01

[10] Kasauli, Rashidah; Knauss, Eric; Horkoff, Jennifer; Liebel, Grischa; {de Oliveira Neto}, Francisco Gomes; “Requirements engineering challenges and practices in large-scale agile system development” in Journal of Systems and Software - Volume 172, Issue 0, pp. 110851 - published 2021-01-01

[11] Jalali, Samireh; Wohlin, Claes; Angelis, Lefteris; “Investigating the applicability of Agility assessment surveys: A case study,” in Journal of Systems and Software - Volume 98, Issue 0, pp. 172-190 - published 2014-01-01

[12] Guaragni, Fausto; Schmidt, Tobias; Paetzold, Kristin; “Traditional and Agile Product Development in a Hyperconnected World: Turning Weaknesses into Strengths,” in Procedia CIRP - Volume 52, Issue 0, pp. 62-67 - published 2016-01-01

[13] Yagüe, Agustin; Garbajosa, Juan; Díaz, Jessica; González, Eloy; “An exploratory study in communication in Agile Global Software Development,” in Computer Standards & Interfaces - Volume 48, Issue 0, pp. 184-197 - published 2016-01-01

[14] {Olszewska (née Pląska)}, Marta; Heidenberg, Jeanette; Weijola, Max; Mikkonen, Kirsi; Porres, Ivan; “Quantitatively measuring a large-scale agile transformation,” in Journal of Systems and Software - Volume 117, Issue 0, pp. 258-273 - published 2016-01-01

[15] Dikert, Kim; Paasivaara, Maria; Lassenius, Casper; “Challenges and success factors for large-scale agile transformations: A systematic literature review,” in Journal of Systems and Software - Volume 119, Issue 0, pp. 87-108 - published 2016-01-01

[16] Jovanović, Miloš; Mas, Antonia; Mesquida, Antoni-Lluís; Lalić, Bojan; “Transition of organizational roles in Agile transformation process: A grounded theory approach,” in Journal of Systems and Software - Volume 133, Issue 0, pp. 174-194 - published 2017-01-01

[17] {von Rosing}, Mark; {von Scheel}, Joshua; Gill, Asif Qumer; “Applying Agile Principles to BPM - The Complete Business Process Handbook,” Volume 0, Issue 0, pp. 557-581 - published 2015-01-01

[18] Gregory, Peggy; Barroca, Leonor; Sharp, Helen; Deshp; e, Advait; Taylor, Katie; “The challenges that challenge: Engaging with agile practitioners’ concerns,” in Information and Software Technology - Volume 77, Issue 0, pp. 92-104 - published 2016-01-01

[19] Collignon, Stephane Eric; Nazir, Salman; Surendra, N; a Chingleput; “Agile systems development: Privacy theoretical lens to challenge the full information disclosure paradigm.” in Information & Management - Volume 59, Issue 6, pp. 103679 - published 2022-01-01

[20] Tam, Carlos; da Costa Moura, Eduardo Jóia; Oliveira, Tiago; Varajão, João; “The factors influencing the success of on-going agile software development projects,” in International Journal of Project Management - Volume 38, Issue 3, pp. 165-176 - published 2020-01-01

[21] Gabriel, Stefan; Niewoehner, Nadine; Asmar, Laban; Kühn, Arno; Dumitrescu,

Roman; “Integration of agile practices in the product development process of intelligent technical systems” in Procedia CIRP - Volume 100, Issue 0, pp. 427-432 - published 2021-01-01

[22] Campanelli, Amadeu Silveira; Parreiras, Fern; o Silva; “Agile methods tailoring – A systematic literature review,” in Journal of Systems and Software - Volume 110, Issue 0, pp. 85-100 - published 2015-01-01

[23] Schön, Eva-Maria; Thomaschewski, Jörg; Escalona, María José; “Agile Requirements Engineering: A systematic literature review,” in Computer Standards & Interfaces - Volume 49, Issue 0, pp. 79-91 - published 2017-01-01

[24] Younas, Muhammad; Jawawi, Dayang N.A.; Ghani, Imran; Fries, Terrence; Kazmi, Rafaqut; “Agile development in the cloud computing environment: A systematic review,” in Information and Software Technology - Volume 103, Issue 0, pp. 142-158 - published 2018-01-01

[25] Lindsjørn, Yngve; Sjøberg, Dag I.K.; Dingsøyr, Torgeir; Bergersen, Gunnar R.; Dybå, Tore; “Teamwork quality and project success in software development: A survey of agile development teams,” in Journal of Systems and Software - Volume 122, Issue 0, pp. 274-286 - published 2016-01-01

[26] Inayat, Irum; Salim, Siti Salwah; Marczak, Sabrina; Daneva, Maya; Shamshirb; Shahaboddin; “A systematic literature review on agile requirements engineering practices and challenges,” in Computers in Human Behavior - Volume 51, Issue 0, pp. 915-929 - published 2015-01-01

[27] Marnewick, Carl; Marnewick, Annlizé L.; “Benefits realisation in an agile environment,” in International Journal of Project Management - Volume 40, Issue 4, pp. 454-465 - published 2022-01-01

[28] Biesialska, Katarzyna; Franch, Xavier; Muntés-Mulero, Victor; “Big Data analytics in Agile software development: A systematic mapping study,” in Information and Software Technology - Volume 132, Issue 0, pp. 106448 - published 2021-01-01

[29] Rodríguez, Pilar; Mäntylä, Mika; Oivo, Markku; Lwakatare, Lucy Ellen; Seppänen, Pertti; Kuvaja, Pasi; “Chapter Four - Advances in Using Agile and Lean Processes for Software Development,” in Volume 113, Issue 0, pp. 135-224 - published 2019-01-01

[30] Kupiainen, Eetu; Mäntylä, Mika V.; Itkonen, Juha; “Using metrics in Agile and Lean Software Development – A systematic literature review of industrial studies,” in Information and Software Technology - Volume 62, Issue 0, pp. 143-163 - published 2015-01-01

[31] Persson, John Stouby; Bruun, Anders; Lárusdóttir, Marta Kristín; Nielsen, Peter Axel; “Agile software development and UX design: A case study of integration by mutual adjustment,” in Information and Software Technology - Volume 152, Issue 0, pp. 107059 - published 2022-01-01

[32] Aldave, Ainhoa; Vara, Juan M.; Granada, David; Marcos, Esperanza; “Leveraging creativity in requirements elicitation within agile software development: A systematic literature review,” in Journal of Systems and Software - Volume 157, Issue 0, pp. 110396 - published 2019-01-01

[33] Govil, Nikhil; Sharma, Ashish; “Estimation of cost and development effort in Scrum-based software projects considering dimensional success factors,” in Advances in Engineering Software - Volume 172, Issue 0, pp. 103209 - published 2022-01-01

[34] Vallon, Raoul; {da Silva Estácio}, Bernardo José; Prikladnicki, Rafael; Grechenig, Thomas; “Systematic literature review on agile practices in global software development,” in Information and Software Technology - Volume 96, Issue 0, pp. 161-180 - published 2018-01-01

[35] Kiran, D.R.; “Chapter 26 - Just in time and kanban - Production Planning and Control,” in Volume 0, Issue 0, pp. 369-379 - published 2019-01-01

[36] Ahmad, Muhammad Ovais; Dennehy, Denis; Conboy, Kieran; Oivo, Markku; “Kanban in software engineering: A systematic mapping study,” in Journal of Systems and Software - Volume 137, Issue 0, pp. 96-113 - published 2018-01-01

[37] Lei, Howard; Ganjeizadeh, Farnaz; Jayach; ran, Pradeep Kumar; Ozcan, Pinar; “A statistical analysis of the effects of Scrum and Kanban on software development projects,” in Robotics and Computer-Integrated Manufacturing - Volume 43, Issue 0, pp. 59-67 - published 2017-01-01

[38] Ellis, George; “Chapter 8 - Agile Project Management: Scrum, eXtreme Programming, and Scrumban - Project Management in Product Development,” in Volume 0, Issue 0, pp. 223-260 - published 2016-01-01

[39] A. Granulo; A. Tanović; “Comparison of SCRUM and KANBAN in the Learning Management System implementation process,” in 2019 27th Telecommunications Forum (TELFOR) - Volume 0, Issue 0, pp. 1-4 - published 2019-01-01

[40] M. Alqudah; R. Razali; “A comparison of scrum and Kanban for identifying their selection factors,” in 2017 6th International Conference on Electrical

Engineering and Informatics (ICEEI) - Volume 0, Issue 0, pp. 1-6 - published 2017-01-01

[41] N. Ozkan; S. Bal; T. G. Erdogan; M. Ş. Gök; “Scrum, Kanban or a Mix of Both? A Systematic Literature Review,” in 2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS) - Volume 0, Issue 0, pp. 883-893 - published 2022-01-01

[42] H. R. Herdika; E. K. Budiardjo; “Variability and Commonality Requirement Specification on Agile Software Development: Scrum, XP, Lean, and Kanban,” in 2020 3rd International Conference on Computer and Informatics Engineering (IC2IE) - Volume 0, Issue 0, pp. 323-329 - published 2020-01-01

[43] Vishnubhotla, Sai Datta; Mendes, Emilia; Lundberg, Lars; “An Insight into the Capabilities of Professionals and Teams in Agile Software Development: A Systematic Literature Review - Proceedings of the 2018 7th International Conference on Software and Computer Applications,” in Volume 0, Issue 0, pp. 10–19 - published 2018-01-01

[44] Camara, Rafael; Alves, Annelyelthon; Monte, Iury; Marinho, Marcelo; “Agile Global Software Development: A Systematic Literature Review - Proceedings of the XXXIV Brazilian Symposium on Software Engineering,” in Volume 0, Issue 0, pp. 31–40 - published 2020-01-01

[45] Li, Mengting; Li, Xin; Chen, Fan; Hao, Haijiang; Li, Ran; “Implementation of Agile Development Software Based on Project Management - Proceedings of the 4th International Conference on Intelligent Information Processing,” in Volume 0, Issue 0, pp. 397–404 - published 2020-01-01

[46] Taibi, Davide; Lenarduzzi, Valentina; Ahmad, Muhammad Ovais; Liukkunen, Kari; “Comparing Communication Effort within the Scrum, Scrum with Kanban, XP, and Banana Development Processes - Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering,” in Volume 0, Issue 0, pp. 258–263 - published 2017-01-01

[47] Matthies, Christoph; “Scrum2kanban: Integrating Kanban and Scrum in a University Software Engineering Capstone Course - Proceedings of the 2nd International Workshop on Software Engineering Education for Millennials,” in Volume 0, Issue 0, pp. 48–55 - published 2018-01-01

[48] Rindell, Kalle; Hyrynsalmi, Sami; Lepp\"{a}nen, Ville; “A Comparison of Security Assurance Support of Agile Software Development Methods - Proceedings of the 16th International Conference on Computer Systems and Technologies,” in Volume 0, Issue 0, pp. 61–68 - published 2015-01-01

[49] Gunawan, F; i; K. Budiardjo, Eko; “A Quest of Software Process Improvements in DevOps and Kanban: A Case Study in Small Software Company - 2021 The 4th International Conference on Software Engineering and Information Management,” in Volume 0, Issue 0, pp. 39–45 - published 2021-01-01

[50] Verwijs, Christiaan; Russo, Daniel; “A Theory of Scrum Team Effectiveness,” in ACM Trans. Softw. Eng. Methodol. - Volume 0, Issue 0, pp. - published 2022-11-01

[51] Lous, Pernille; Kuhrmann, Marco; Tell, Paolo; “Is Scrum Fit for Global Software Engineering? - Proceedings of the 12th International Conference on Global Software Engineering,” in Volume 0, Issue 0, pp. 1–10 - published 2017-01-01

[52] Matharu, Gurpreet Singh; Mishra, Anju; Singh, Harmeet; Upadhyay, Priyanka; “Empirical Study of Agile Software Development Methodologies: A Comparative Analysis,” in SIGSOFT Softw. Eng. Notes - Volume 40, Issue 1, pp. 1–6 - published 2015-02-01