DOI: 10.15439/2022F143
ISSN 2300-5963 ACSIS, Vol. 30

Scrum, Kanban or a Mix of Both? A Systematic Literature Review

Necmettin Ozkan
Information Technologies Research and Development Center
Kuveyt Turk Participation Bank
Kocaeli, Turkey
necmettin.ozkan@kuveytturk.com.tr

Sevval Bal
Information Systems Engineering Dept
Sakarya University,
Sakarya, Turkey
sevval.bal@ogr.sakarya.edu.tr

Tugba Gurgen Erdogan
Computer Engineering Dept.,
Hacettepe University,
Ankara, Turkey
tugba@cs.hacettepe.edu.tr

Mehmet Şahin Gök
Department of Business
Gebze Technical University
Kocaeli, Turkey
sahingok@gtu.edu.tr

***Abstract*—Among the Agile methods, Scrum and Kanban are widely used in software development and they are considered the two most effective ones influencing the direct results of projects. Despite the importance of knowing their relative strengths and advantages and integrating them to achieve better results than individual use, none of the secondary studies provide extensive knowledge on the topic. In this paper, we performed a Systematic Literature Review (SLR) study to investigate the characteristics of the empirical studies which involve Scrum and Kanban by comparing or integrating them. Our final set includes 38 studies posing primary information on the advantages of each method over another one, the properties including artifacts, roles, and events from Scrum and Kanban in combining them in a hybrid way, and the properties of transitions from one to another such as transition directions (Scrum to Kanban, Kanban to Scrum or Scrum/Kanban to Hybrid), transition years, and transition reasons. The outputs can be interesting for both industry and researchers. For example, nearly all of the transitioning organizations are moving from Scrum to Kanban or to hybrid method. Among the reasons for the transitions, the problems experienced with Scrum are remarkable. In comparison, Kanban stands out clearly in a positive way. Almost all of the teams combining both use flow instead of a sprint.**

***Index Terms*—Agile, software development, systematic literature review, SLR, agility, project management.**

I. Introduction

IT IS a fact that due to the evolutionary nature of software development, changing market needs, and evolving technology [1], software projects inevitably change in many aspects including requirements, circumstances, and stakeholders [2], which require agility in complex domains. Based on this natural need for agility, people have invented varying agile approaches and methods to meet the need to be compatible in the market, have shorter development cycles, fewer costs, and have the ability to move and change quickly [3, 4]. Among the agile methods, Scrum and Kanban are common in the software industry [5] and they are considered the two effective agile methods that handle and manage the progress of software development [4, 6, 7].

Deciding on a development approach is one of the critical factors influencing the direct results of projects [8]. There has already been a debate for years about which of these methods (Scrum and Kanban) are preferred [7, 9]. These cases call for a proper and deep understanding of possible methods, recognizing their strengths and weaknesses, limitations, relative advantages compared to others, context constraints, and so on. For instance, Scrum has limitations directly affecting application results, such as lack of work visibility, local optimization, large-scale implementations, and changing task priorities [10, 11, 12]. Similarly, as all other methods do, Kanban has some problems and challenges as well [13, 14]. Considering the individual limitations and challenges of each method, there are also some views that blending more than one method will yield better results than individual use. For instance, there are views that the limitations in Scrum can be mitigated by using Kanban alongside Scrum and they can complement each other [9, 15, 16, 17].

Some literature review studies have revealed the state of Scrum and Kanban separately. Despite the importance of knowing Agile methods closely, comparing them with each other, and using them together as a possible next step, none of the secondary studies provide extensive knowledge on the topic of comparing and/or integrating Scrum and Kanban to structure information available in the literature and highlight the opportunities for further research and practice.

In this paper, we performed a comprehensive literature review study to investigate the characteristics of the empirical studies which involve Scrum and Kanban by comparing or integrating them. We used a systematic literature review (SLR) as a research methodology which is a common method to conduct a literature review study. We manually searched for the studies published until March 2022 in the electronic digital libraries of scientific literature listed in Table I, and reached and deeply investigated 38 sources that compared or used Scrum and Kanban together.

The remainder of this paper is organized as follows. Section 2 summarizes the related works. In Section 3, we describe the overview research design with research questions and selection process. Section 4 delivers the results of the literature review. In Section 5, we discuss our findings and observations and in Section 6, we deliver limitations of the study and propose suggestions for future work.

II. Related Works

There are plenty of studies reviewing the Agile methods, comparing them with their characteristics, strengths, weaknesses, similarities, and differences, providing criteria to choose them according to the context of development including those covered in this study. Apart from those included in our study, there are some other non-empirical studies comparing Scrum and Kanban such as study [18] and books such as [19] (However, as we focus only on empirical studies in our study, we excluded such unempirical ones).

There are some secondary studies close to our scope and systematically review the literature. For instance, study [20] systematically reviews the literature to identify the studies providing practices in requirements specification incorporated into the Agile development methods and delivers a comparison among Scrum, Extreme Programming, and Lean in this regard. From the systematic literature review, they found a total of 12 relevant studies and eight variability and commonality practices between these three methods. Study [21] systematically reviews the literature in order to analyze the current trends of Kanban usage in software development and to identify obtained benefits and involved challenges. Similarly, study [13] conducts a systematic mapping of Kanban literature in software engineering between 2006 and 2016 resulting in 23 primary relevant papers. It provides benefits, challenges and recommended practices of Kanban applications in software development and state-of-the-art opportunities for Kanban research. Similarly, there are many similar studies on Scrum as well. However, we have not found any study that systematically reviews the literature related to comparing and/or combining Scrum and Kanban.

III. RESEARCH DESIGN

This research process has been undertaken as an SLR based on the guidelines proposed by Kitchenham et al. [22]. The following section delivers the method used to conduct this SLR.

The research process starts with defining research goals and questions. After defining search queries and searching in the major digital libraries, we gathered 391 potentially relevant publications. For scanning the retrieved studies, we developed and applied inclusion/exclusion criteria and obtained a final pool of 38 sources. After extracting the data from the sources, the results of SLR are analyzed and the findings are discussed. The remainder of the section concerns the research questions, publication selection process, data extraction and synthesis, quality assessment, and potential threats to validity.

A Research Questions

This study aims to review and classify studies that compare and/or combine Scrum and Kanban. Thus, we set the main goals related to our research 1) identify the studies which compare and/or combine Scrum and Kanban and 2) analyze and synthesize the studies' results. Based on our study's goals, we raise and investigate four main and seven sub-research questions (RQs):

RQ1. Contribution types and research methods:

RQ1.1. Contribution types: How many sources have represented a new software process model, method, metric, or process/workflow in the software development domain using Kanban and Scrum together?

RQ1.2. Research methods: What types of research methods were used in the empirical studies by the authors?

RQ2. Advantages of each method over another one:

RQ2.1. Benefits of Kanban against Scrum: What benefits (quantitative or qualitative) have been reported as a result of applying Kanban against Scrum?

RQ2.2. Benefits of Scrum against Kanban: What benefits (quantitative or qualitative) have been reported as a result of applying Scrum against Kanban?

RQ3. What properties are used from Scrum and Kanban in combining them? What artifacts, roles, and events are combined and customized while combining. We aim to reveal some patterns through the consolidated list of the elements integrated in combinations of Scrum and Kanban.

RQ4. What are properties of the transitions from one method to another?

RQ4.1. What are transitions' directions (From Scrum to Kanban, Kanban to Scrum or Scrum/Kanban to Hybrid)

RQ4.2. Which years did transitions happened?

RQ4.3. What are the transitions' reasons?

B Publication Selection Process

The search process was a manual search of peer-reviewed studies in well-known digital libraries without any specific filter in the year range. Based on the scope of this study, the search string including "Kanban and Scrum" as "intervention" was developed by following the SLR protocol [22, 23]. We did not add "population" related keyword in the string referring to the application area which is software in our study to access the largest possible set, rather, this search was done manually by the authors in the papers. Regarding the search location, we anticipated and were satisfied with the effectiveness of searching in meta-data instead of the full text as the main aim of the paper is a comparison and combination of two methods in the software development domain in different contexts, then it is expected the authors locate the relevant terms in the papers' meta-data. Finally, the search process was done in March 2022 as shown in Table I.

TABLE I. NUMBER OF SOURCES RETRIEVED AND SELECTED BY SEARCH KEYWORDS

Search Library	Place	Search String	#Initially Retrieved	#Selected
Web of Science	Meta-data	Kanban AND Scrum	119	14
Wiley			2	1
Science Direct			6	3
Scopus			179	35
IEEE			62	14
Emerald			2	0
ACM DL			21	3
Total			**391**	**70**
Total in Distinct			**216**	**38**

After defining the keywords and libraries, a pilot search was done by the first two authors to make sure the search process that would be applied is standard across the research. Based on the scope and context of our study, for the selection of papers, the following propositions of inclusion criteria (IC) and exclusion criteria (EC) in Table II were specified and applied to the papers, by focusing on empirical studies which applied Scrum and Kanban to compare or integrate into the software development domain. As can be seen in the list, studies suggesting only the use of the Kanban board, which is similar to the Scrum board, and studies examining the use of Scrumban [16] which is also defined as a specific method, are excluded.

During the application of inclusion/exclusion criteria, the papers were examined through their titles and, where necessary, abstracts in order to identify whether they are within our scope. If the abstracts were not sufficient to decide to include or exclude the papers, then, scanning through the full texts of the papers was done to identify potentially relevant ones. After the first two authors of this study identified their selected papers, the paper lists obtained from these authors were then compared to each other until reaching a consensus between them. In this step, exclusions from and inclusions into the list were made, resulting in the final agreed-upon list. The whole search process was coordinated by the third author and reviewed by the fourth author to propose improvements if needed.

In the process, a total number of 391 peer-reviewed studies were returned from the search results as seen in Table I. This initial list included duplicate records since databases we covered perform meta-data indexing of publishing databases we included directly. After removing the duplicate records, the list included 216 distinct records. Out of these 216 papers, 12 papers were investigated only through their titles, 47 of them through their titles and abstracts, and 157 of them through their titles, abstracts, and full texts.

According to the ICs and ECs, 178 papers were excluded as seen in Table II. It is noted that five papers are not accessible by the authors, even though at the first glance, they seem relevant to our study scope. Specific to these five papers, a clear decision could not be made to include or exclude them because the title and summary information were not sufficient to decide and, then, they are excluded because there is no access to the full texts by the authors. We applied EC11 (papers not available in English) either by filtering via the library relevant features allowing eliminating non-English study forehand or otherwise via manual investigations by the authors. After the manual investigation, the full texts of the five studies were not in English, and then, they were excluded. Consequently, 38 papers that compare and/or combine Scrum and Kanban within the scope of our study were identified. For the whole list, the spreadsheet containing the search results, together with inclusion and exclusion decisions is available online [24].

C. Data Extraction and Synthesis

A data collection form holding information was designed to record the collected information from the identified studies. The collected information ranges from general information about each study such as Author, Title, Year, Venue, Author Affiliation, and Author Country, as well as specific information to answer the research questions.

For the information that is subjective and open to evaluation, first, the relevant information was taken as it is from the relevant study and copied to the Excel file. Such information that is relatively difficult to quantify is such data from the parts about how the methods are combined and how they are superior to each other. Additionally, the parts of the studies including this particular information were highlighted in the original paper for any future references. For this type of data, it was aimed for researchers to have a bias as little as possible and comment on the original data in this way and develop his/her comments gradually. In addition, four randomly selected studies were independently reviewed by the first three authors of this study and jointly evaluated until consensus was reached to ensure a common understanding and the data extraction step is applied in the agreed standard way. The rest of 34 remaining papers were allocated randomly to the first two researchers. The first and second researchers run separate sessions in order to extract the data that serves to answer the research questions by applying detailed and thorough examinations of the relevant studies. The third author coordinated the data extraction process. Consequently, a thorough reading of each of the 38 identified papers was performed to extract relevant information. Once data extraction was complete, the extracted data were closely synchronized and analyzed with the first two authors.

TABLE II. INCLUSION CRITERIA AND EXCLUSION CRITERIA

ID	Criterion	# of Eliminated Studies
IC1	Papers empirically validated and applied research methods like a case study, survey, action research, interview, simulation, literature review, experiment, focus group, pilot study, and statistical analysis	-
IC2	Papers comparing and/or combining Scrum and Kanban	-
IC3	Papers in software development domain	-
IC4	Conference, workshop, journal or book-chapter papers	-
EC1	Not empiric studies such as idea or opinion papers	26
EC2	Papers not related to comparing or combining Scrum and Kanban	84
EC3	Papers providing only a usage of Kanban board within Scrum	5
EC4	Papers providing only a usage ratio for Scrum and Kanban in any field	6
EC5	Papers providing only usage of Scrumban	1
EC6	Papers in other than the software development domain (for example healthcare, construction, supply chain, education, and teaching)	33
EC7	Papers published in non-peer-reviewed sources such as thesis, web pages, workshop proposals, tutorials, panels, proceeding information and, books.	13
EC8	Secondary studies such as systematic reviews or mapping studies	0
EC9	Papers not accessible by the authors	5
EC10	Duplicate studies	0
EC11	Articles not in English	5

D. Quality assessment

The entire process relies on a search procedure that calls for explicit criteria to validate the quality of the selected candidate papers by ensuring each candidate paper is of adequate standard [22]. Accordingly, a custom quality-assessment-criteria-list and item descriptions were established as shown in Table III.

Each paper has been then assessed against this given set of questions by the assigned authors. A manual inspection was done through the full text investigation carried out to identify

TABLE III. CRITERIA FOR QUALITY ASSESSMENT

Criteria- Statement	Descriptions
QA1- Are contributions of method clear?	Clarity and robustness of the method applied in the study is satisfactory
QA2- Are outcomes as results clear?	Outcomes are clearly delivered and relevant to the method applied
QA3- Is discussion on results clear?	Discussion of the results is satisfactory and based on the results objectively. Validity threats are delivered.
QA4- Does paper have a citation?	Papers published at least two years ago have at least one citation.

each selected paper's quality assessment score. We set a score-weight based on the two values; Satisfactory (1) and Not Satisfactory (0). Accordingly, the evaluations of the papers have been made based on the predefined two values to set their scores yielding a score of four at maximum. It was decided that the studies with a score below two points would be eliminated. After applying the determined quality criteria, the quality scores of each study are as in the Excel Spreadsheet "Demographics & QA" available online at the previously shared Excel. As seen, there existed no studies assigned the lower than the threshold score and no elimination regarding the quality assessment was done.

IV. RESULTS

We present the results and findings of this SLR study concerning RQ1-RQ4. All sources included in this study are listed in a publicly accessible repository available at the previously shared online sheet named "Demographics & QA".

According to the results, 63% (24/38) of the studies are conference papers, 34% are journal articles, and 3% are workshop papers. In terms of venues for the selected 38 papers, "International Conference on Agile Software Development (XP)" is at the top with five papers, followed by "Hawaii International Conference on System Sciences (HICSS)" and "Agile Conference" with three papers for each. Regarding the journal articles, "Journal of Software: Evolution and Process" is the top one with two papers. In terms of the authors' affiliation types, 55% (21/38) of the papers are from academia, 24% from industry, and 21% from industry and academia collaboration. Related to the authors' country distribution, the USA is at the top with nine authors, followed by Norway, New Zealand, Brazil, the UK and Italy with three papers for each, among twenty-two different countries in total. The contexts of the studies include software development (not specified) with seven papers at the top, software maintenance with three papers, and web-based content management system with two papers respectively. Other contexts include finance and insurance web services, automotive production software, content management system, library information system, data science, software project management, cloud-based software development, video game development, mobile software development and so on, among others. Figure 1 shows the cumulative number of sources per year by considering type (comparing vs. integrating).

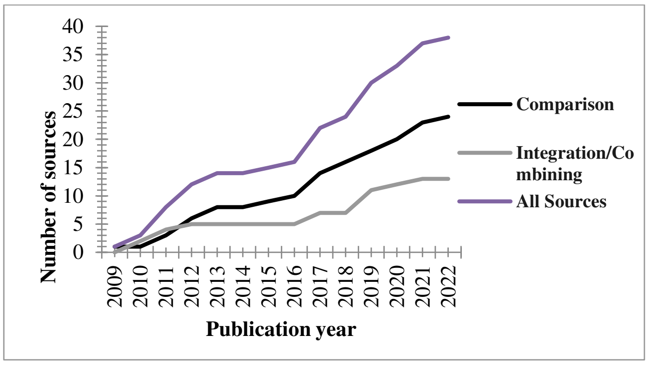

Fig. 1: Number of sources per year

Regarding RQ1.1 (Contribution types), Figure 2 depicts the contribution types of the primary sources. As shown, 22 studies do not provide any new model, method, tool, metric or process, as they rather mainly deliver a comparison between the methods or apply them. We see a considerable amount of new model proposals.

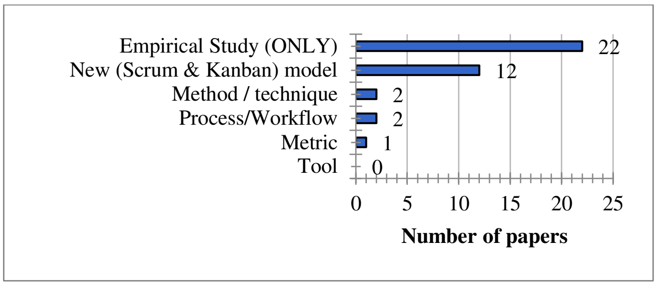

Fig. 2: Contribution types

In terms of the research methods used in the empirical studies by the authors (regarding RQ1.2), as seen in Figure 3, the majority of the studies use a case study method, followed by a survey, interview, and simulation.

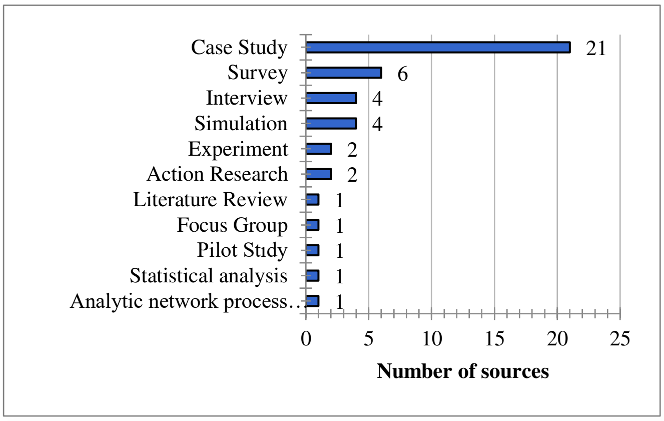

Fig. 3: Research methods

When it comes to RQ2.1 (Benefits of Kanban against Scrum), Table IV delivers the results. Similarly, Table V depicts results for Scrum Benefits against Kanban (RQ2.2), ordered by "Number of sources". We have added the domain information of each source by using abbreviations; 3D Animation Production as 3D, Agile Application (General) as Agile G., Agile Testing as Test, Automotive as Auto, Broadcasting Software Development as Broadcast, Cloud-based Software Development as Cloud, Content Management System as Content, Data Science Project Management as Data, Finance and Insurance Web Services as Finance,

TABLE IV: KANBAN BENEFITS AGAINST SCRUM

Category	Kanban Benefits against Scrum	Domain	Number of sources - Sources
Process	Flexible and adaptive	Mob,Web, Lrn, Lib, Agile,Game	**7-** [9], [30], [33], [36], [37], [4], [43]
	Easy for transition and use	Data, Lrn, Agile, Soft	**7-** [9], [35], [36], [4], [42], [45], [47]
	Efficiency	Soft, Broadcast, Cloud, UX	**4-** [48], [57], [58], [14]
	Focus on work	Lrn, Soft, Maintenance, Broadcast	**4-** [36], [54], [55], [57]
	Quality	Agile, Game, Soft P.	**4-** [9], [4], [43], [53]
	Delivery time	Content, Game	**3-** [9], [38], [43]
	Visibility	Maintenance, Broadcast, Soft P.	**3-** [7], [55], [57]
	Performance	Agile, Soft, Maintenance	**3-** [25], [4], [46]
	Controlling the flow	Web, Cloud	**2-**[40], [58]
	Delivering value	Lib, Soft	**2-**[37], [54]
	Project schedule management	Soft P.	**2-**[7], [53]
	Project resources management	Soft P.	**1-** [53]
	Project risk management	Soft P.	**1-** [53]
	Consistency in project management	Soft P.	**1-** [7]
	Being reliable	Agile	**1-** [4]
	Better for development with no certain deadline	Lrn	**1-** [36]
	Better for small and not complex feature development	-	**1-** [9]
	Closer contacts with users	UX	**1-** [14]
	Cost-effective	-	**1-** [9]
	Eliminating excessive and unnecessary meetings	Mob	**1-** [30]
	Less rework	Lib	**1-** [37]
	Getting fast feedbacks	-	**1-** [9]
	Traceability	Maintenance	**1-** [55]
People	Teamwork	Maintenance, Data	**2-**[35], [55]
	Collective understanding	Maintenance, Broadcast	**2-**[55], [57]
	Respect for people and their current states	Soft	**1-** [42]
	Satisfaction of individual team members	Data	**1-** [35]
	Less stress	Mob	**1-** [27]

TABLE V: SCRUM BENEFITS AGAINST KANBAN

Category	Scrum Benefits over Kanban	Domain	Number of sources - Sources
Process	Path clarity, being well-defined	Soft, Video	**3-** [42], [43], [47]
	Better for development with batched and a block of works	Agile	**2-** [9], [4]
	Delivery time	Maintenace, Soft	**2-** [25], [47]
	Better for bigger teams	UX	**1-** [14]
	Better for development with certain deadline	Lrn	**1-** [36]
	Better for development within high uncertainty environments	Agile	**1-** [4]
	Better for the testing process	Test	**1-** [31]
	Project cost management	Soft	**1-** [53]
	Detailed tracking and overview of projects	Lrn	**1-** [36]
	Predictability	Soft	**1-** [54]
	Specification of customer requirements	Soft	**1-** [47]
	Easy to manage	Soft	**1-** [54]
People	Teamwork	Web, Soft	**3-** [33], [42], [47]
	Empowering	Mob	**1-** [27]
	Communication	Video	**1-** [43]

Learning Management System as Lrn, Library Information System as Lib, Mobile Software Development as Mob, Open Source as Open, Software Development (General) as Soft, Software Development Project Management as Soft P., Software Maintenance as Maintenance, System Admin as System, Telecommunication Software Development as Telco, User Experience as UX, Video Game Development as Game, and Web-based Content Management system as Web. In column "Category", the classified information of each item including process, organization and human dimension is given. The "Organization" category is about making decisions related to software engineering and development in a business context and aligning software technical decisions with the business goals of the organization [13].

Kanban is regarded as more flexible and adaptive than Scrum without strict time-boxes, roles, rules and sprint constraints. Kanban teams feel more powerful to address their internal processes and respond quickly with its continuous flow management allowing constant re-planning to uncertainty and frequent changes and faster responses. They can deliver results earlier with frequent releases and division of the work in very small-time chunks [37], [48]. Kanban is also simpler and paves the way to an easier transition with less cost, time, chaos and turbulence. It does not touch titles or positions of people, requiring a lower “patience point” [42]. Thanks to its more “sequential” nature and WIP limits, Kanban seems to be more efficient in distributed context [58]. Using Kanban over its precursor Scrum practices brings advantages in terms of efficiency and focus with WIP limits, stopping context-switching, providing granularity of visualization, gaining a collective understanding of the whole process, less paperwork, and reducing batch size through the pipeline [57], [14].

As reported quantitatively and/or qualitatively by some studies, Kanban provides better quality, project schedule, risk, and resource management, reliability, lead time results, performance, visibility, traceability, teamwork, the satisfaction of individual team members, regular feedback faster, and closer contacts with users. It does not require haste like in Scrum and, therefore, resulted in better quality [4] and less stress [27]. Kanban works better within certain contexts like development with no certain deadline and small and not complex feature development. It allows for identifying reworks at early stages [37]. It has been found that Scrum supports collectivist teamwork, more empowered team members, shared goals and intense communication across the team. Although Scrum has a rigid and routine structure, it provides a path clarity and easiness to manage the processes thanks to its being a well-defined method. According to some results, Scrum provides a better cycle time. Scrum also seems to be more suitable for bigger teams, batched works and developments within high uncertainty environments like for new product development and within certain deadlines. Some of the results quantitatively put forward that it is better than Kanban in terms of project cost management and agile testing processes. Scrum provides better predictability and detailed tracking and overview of projects.

Regarding RQ3 (properties of the proposed combining models), Table VI depicts the results by providing what properties are used from Scrum and Kanban in combining them (M stands for Method). It shows in particular what artifacts, roles, and events are combined and customized while combining Scrum and Kanban. In the Scrum and Kanban integration, numerically speaking, the most used elements are PO (Product Owner), Daily, Sprint Review, Sprint Retrospective, Scrum Team, User Story, Definition of Done, and Sizing from Scrum, and, Pull system, Continuous flow, WIP, and Kanban boards from Kanban. Besides, there are some custom-made elements in their integrations. Notable ones can be listed as follows: simultaneously used number and size-based WIP, iterative movement of the items on the flow, work/hypothesis/experiment/release-based iterations, calendar-based regular or on-demand ceremonies, daily planning, team formations that break the “standard” cross and self-organizing team structures including floating teams, sub-groups/roles and supervisory authority, same/similar-size work items, and product owner teams.

TABLE VI: ELEMENTS USED IN COMBINING

M	Element/Source	Number of sources - Sources
Scrum	PO	**11** - [26], [28], [29], [32], [33], [34], [39]-Case1, [40], [41], [49], [56]
	Daily	**9** - [26], [28], [33], [38], [41], [44], [49], [51], [52]
	Scrum Team	**7** - [26], [28], [39]-Case1, [49], [51], [52], [56]
	Product Backlog	**6** - [28], [32], [34], [44], [49], [56]
	User Story	**5** - [26], [28], [32], [38], [51]
	Definition of Done	**5** - [28], [34], [38], [44], [51]
	Sizing (T-Shirt size, effort-based, planning poker etc.)	**5** - [39]-Case2, [40], [49], [51], [56]
	Sprint Review	**5** - [28], [34], [38], [44], [49]
	Sprint Retrospective	**5** - [28], [34], [38], [44], [49]
	Grooming	**3** - [34], [49], [56]
	SM (Scrum Master)	**3** - [28], [33], [49]
	Definition of Ready	**2** - [28], [51]
	Sprint Planning	**1** - [26]
	Minimal Marketable Feature-Sets / Minimum Viable Product	**1** - [41]
	Sprint	**1** - [26]
Kanban	Kanban boards	**16** - [26], [28], [29], [32], [33], [34], [38], [39]-Case1, [39]-Case2, [40], [41], [44], [49], [51], [52], [56]
	Work in progress (WIP)	**10** - [28], [33], [34], [39]-Case1, [40], [44], [49], [51], [52], [56]
	Pull system	**7 –** [34], [38], [40], [41], [44], [51], [56]
	Continuous flow	**5** - [33], [39]-Case1, [40], [41], [56]
	Metrics	**3** - [39]-Case2, [40], [44]
	Value-stream/chain	**2** - [28], [41]
	Service level agreement/expectation	**2** - [28], [39]-Case2
	Size of task is not limited	**1** - [56]
	Replenishment	**1** - [52]
	Kanban coach	**1** - [52]
Custom	Calendar-based regular ceremonies	**4** - [40], [49], [52], [56]
	Work/Experiment/Hypothesis/Release-based iteration planning	**4** - [29], [32], [49], [51]
	Back and forward movement of the items on the flow	**3** - [26], [40], [56]
	Product Owner Team	**3** - [44], [51], [52]
	Supervisory authority	**2** - [29], [32]
	Same/similar-size work items	**2** - [40], [41]
	WIP (number and size based)	**1** - [26]
	Metrics: Cycle and Lead time integrated with story point (pseudo-velocity)	**1** - [26]
	Buffer in capacity	**1** - [26]
	Backlog clean-up actions	**1** - [29]
	Not multi-functional teams	**1** - [32]
	Ceremonies scheduled on demand	**1** - [32]
	Sub-group/role-based board columns	**1** - [32]
	Leader responsible for tasks administration on the board	**1** - [32]
	’Wiki’ to keep track of project’s progress rather than the Scrum events	**1** - [32]
	Daily Planning – a combination of Sprint Planning and Daily Scrum	**1** - [34]
	A person to take care of processes and team development	**1** - [34]
	Review for finished items on the board	**1** - [34]

	Floating teams (distribution of team members between different floating teams)	**1** - [38]
	Each feature owned by a developer	**1** - [38]
	Longer [than the maximum in Scrum] iterations	**1** - [40]
	Team Coaches	**1** - [44]

Related to RQ4 (Properties of the transitions from one method to another), the transition directions are from Scrum to Hybrid (11/18), from Scrum to Kanban (5/18), from Waterfall to Hybrid (1), and from Custom to Hybrid (1/18). It can be seen that most of the transitions are to the hybrid methods and predominantly from Scrum. Additionally, Figure 4 and Table VII present transition years, and transition reasons respectively (the studies that explicitly express the transition year form the cumulative A curve, and after adding the five studies that do not explicitly express the transition year from the cumulative B curve when the year of publication is accepted as the transition year).

For the transition reasons, regarding the lack of flexibility and predictability during the fixed sprints and timeboxes, the studies report that sprints are not adequate for frequent, constant, and unexpected changes and (especially external) dependencies in hectic and unstable environments that require quick responses, frequent re-planning, re-prioritization, and a vast amount of mid-iteration updates for dynamism or resulting in non-predictive sprints. These issues make Scrum unsuitable for maintenance teams in which the dynamism is high and estimation is time-consuming and often incorrect. Under these circumstances, it becomes unclear how much teams would deliver in a sprint [40].

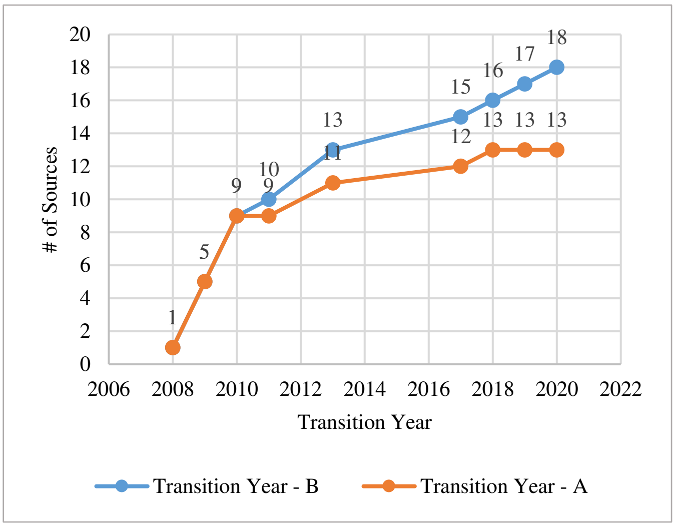

Fig. 5: Transition Year

The combination of inaccurate estimates and timeboxes leads to non-sensing estimation activities and, thus, waste and reduced productivity [38], [46]. Some studies indicate that Scrum causes constant strain and quality issues such as increased technical debt, decreased system maintainability, untested implementations, and unmet production maintenance just to meet business time targets leaving little time to "others" including technical and administrative tasks with "no business value" or "little urgency". Such tasks mostly remain in the background of the business. Study [38] states that developers did not pay enough attention to creating good architectural designs, rather mainly focused on the short-term goals of the sprints. Short-term sprints also decrease visibility beyond sprints. Some teams regard Scrum as difficult to accept with its revolutionary transformation change proposition [45], "aggressive", "materialized" and "rigid" structure [42], [46].

TABLE VII. TRANSITION REASONS

Cate gory	Transition reason	Domain	Number of sources
Scrum			
Process	Lack of flexibility and predictability during sprints for frequent/unexpected changes	3D, Telco, Mob, System, Web, Maintenance, Open, Broadcast	**11-** [28], [29], [32], [33], [34], [40], [44], [45], [55], [56], [57]
	Quality decrease	Telco, Content, Web, Auto, Soft	**5-** [29], [38], [39], [40], [46]
	Unsuitable for maintenance	System, Maintenance, Soft	**4-** [34], [44], [46], [55]
	Estimation is time-consuming, non-sensing, or causing delays	Web, Content, Soft	**3-** [38], [40], [46]
	Scrum ceremonies cost	Mob, Finance	**2-** [32], [41]
	Lack of work visibility	Maintenance, Finance	**2-** [41], [55]
	Focusing on short-term goals of sprints	Content	**1** - [38]
	Challenges in Sprint-based delivery	Web	**1** - [40]
	Increased length of feedback loops	Finance	**1** - [41]
	Being rigid	Soft	**1** - [46]
	Failed sprints	Soft	**1** - [54]
	Reduced productivity	Soft	**1** - [46]
	Work items rarely correlated	Web	**1** - [40]
	Lack of overall status beyond sprints	3D	**1** - [28]
	External dependency	Open	**1** - [56]
	Ineffective Scrum events	Content	**1** - [38]
	Waste	Soft	**1** - [46]
Organization	Challenges in large scale projects	Mob, Content, Soft	**3-** [32], [38], [46]
	Fragmentation	Web, Content, Telco	**3-** [29], [38], [40]
	Revolutionary change	Telco	**1** - [45]
	Lack of co-location	Web	**1** - [40]
	Lack of communication and collaboration	Maintenance	**1** - [55]
	Challenges in having a common language with business	Content	**1** - [38]
People	Need for deep specialty in teams	Web, Open	**2-** [40], [56]
	Constant strain	Telco	**1** - [29]
	Challenges in being self-organizing teams	Telco	**1** - [29]
	Challenges in having cross-functional teams	Web	**1-** [40]
	Lack of team lead resulting in a stagnation of team development	Telco	**1** - [29]
Kanban			
Proc ess	Lack of control	System	**1** - [34]
	Lack of deadlines	System	**1** - [34]
Peop le	Not having a feel progressing	System	**1** - [34]

This strict position of Scrum also seems problematic when a lack of co-location is present [40]. We have seen some scaling challenges with Scrum including issues in the distributed scenarios, breaking large items down into smaller ones to be squeezed into one sprint, lack of synchronization, and fragmentation caused by isolated and localized Scrum teams. Among other items, specialization in certain parts of systems, dependency on deep specialists, and rarely correlated work items to bunch for/in a sprint are counted. Scrum ceremonies can cause rework, high cost, increasing length of feedback loops, lack of end-to-end work visibility, and lack of visibility within sprints. They, then, are done more and more seldom, shorter and shorter in their duration to finally “die out” [38] or end up with failed sprints [54]. Scrum teams may result in stagnation and having dominant team members and group formations within the team when team members are unchanged [29] or some other issues when teams have a number of part-time resources with varying availability [40]. On the other hand, in Kanban, with the lack of boundaries, teams are prone to drive into “guerilla-style” working and feel a lack of deadlines and no actual sense of progress.

V. Discussion

The fact that the majority of sources were produced by industry-oriented authors, especially as experience and case study papers, indicates that the subject of the study can be considered mainly as a practical-led area and a need by the sector. This shows that the industry is ahead of academia in this regard. The data about a cumulative number of sources per year indicates that the first studies start just after the years when the first Kanban documents were published, the two methods are mostly compared in the first years, and the sources on the integration gained speed later on but have slowed down in recent years, which needs a further investigation. In general, after 2016, a noticeable acceleration in the number of sources draws attention.

Static Entities: It seems that there are some (almost) static entities in Scrum including sprint, time-boxes, and team formations that are found problematic in terms of providing flexibility and agility. Scrum plans and starts the sprint with a prediction for the future, but where the external environment change is high, the sprint predictions become incapable. This indicates that the sprint structure provides low agility in a high change environment. With this level of inflexibility, it seems that one of the areas where Scrum is most incapable is maintenance work. The main reason for Scrum's inadequacy in maintenance works is that these environments have a higher level of change than other environments. With a sprint-based maneuverability, Scrum works more comfortably in environments with low/medium uncertainty. Where the uncertainty is high, the estimations made to fit into this artificial and such static sprints become more meaningless and seen as a waste. All these are mainly why people regard Kanban and continuous flow along with its metrics designed for dynamic work management in the hybrid solutions as more flexible, adaptive, and powerful compared to Scrum.

Packaging work items: Sprint also focuses on packaging work items with a development-oriented perspective so-called the artificial world of development, rather than the real needs of the business world, so-called business-oriented development. In this sense, sprint is also (for example) a method of fragmentation. Even if things are not in a nature to be broken up or to be joined, they are compelled to do so.

Deadlines: Sprint includes deadlines, even measured in seconds; the end of a sprint. With a production-based logic, Scrum locates people as production muscles and expects them to comply with the strict lines, stressing and tiring them out. Estimates for the works to be done within the deadlines continue to be made in Scrum with the classical method logic; The main thing changing is the unit (man/day versus story point). With these concrete "lines" and the "glorification" of business orientation, parts that look inward (to technical sides and systems) may also stay in the background. To assign a static end rather than an end according to the nature of the works creates a meaningless and dangerous dichotomy between the nature of the work and these static ends. It is dangerous because the developers may not always choose the right way (e.g. even if things are not “on the way”, do tests properly). This dilemma paves the way for the sacrifice of more "abstract" concepts such as quality. In order to solve this problem, it is seen that some hybrid solutions are adopting business-work/experiment/hypothesis-based iteration planning solutions, not clock-time-based ones.

Push System: In Scrum, the flow is partially “push” based, where tasks would move to the next step as soon as they completed the previous step [59]. Also, Sprint Planning works for how much work to push (into the sprint) [12]. Scrum further “pushes” the teams to produce shippable products at the end of the sprint [12]. This push and time-box approach in Scrum [39] along with artificial hot deadlines of the sprints may lead to more stress, pressure, and fatigue on the teams and enforce them to compromise on quality. Kanban WIP limits aim to reach a near real-time balance between demands and capacity in a way free of stress for developers. Using a persistent, balanced, and thus, stress-free flow in Kanban facilitates achieving a more sustainable and reliable flow [59] that positively affects the final product and its quality. The pull system applying WIP limits stands as a choice of many hybrid solutions reported in our study.

Inside of the Sprint: In Scrum, the inside of the sprint is like a single station of which WIP is sprint-based. It makes the default, fundamental and granular level of work management level sprint-based. Thus, work management inside a sprint is less trackable. For this reason, congestion and bottlenecks may also occur in certain parts of a sprint. Kanban offers a more effective control mechanism with WIP limits that seem to be widely integrated into the hybrid solutions over controlling capacity and scope of work under high variability in Scrum [40]. In Kanban, each workstation is designed separately and traceability and visibility are provided to all teams and stakeholders through a continuous workstation-based flow on the Kanban board (even not reset at the end of the sprint). Kanban board can be considered as an advanced version of Scrum board in this sense and it can be seen that it is widely preferred among the hybrid solutions.

Scaling, with Isolation: The original Scrum setup is team and sprint-based, posing challenges in large-scale projects. This issue is manifested in our results as fragmentation, challenges in large scale projects, focusing on short-term goals of sprints, lack of overall status beyond sprints, and lack of communication and collaboration issues as shown in our “Transition Reasons” list. On the other hand, Kanban aims to gather the teams around the value streams for customers and address the whole organization by adopting system thinking and Fit-For-Purpose principles [60] on the horizontal axis that starts with the customer and ends with the customer delivery.

The indication of this aim in the hybrid solutions is the use of value-stream/chain.

A sprint backlog is owned by one specific team unlike a Kanban board shared by multiple teams [14]. Similarly, when sprint (specific to a particular team), *self-* organizing and cross-functional team promises come together, an encapsulation, division, and, to an extent, isolation issues occur for the teams. Due to its nature and instinct, sprint is a kind of planning within a particular team, while inter-team planning should be done according to emergent principles and globally. Despite the scaling models, doing this alignment within the teams at longer intervals emerges as a problem in Scrum.

Moreover, In Scrum, breaking the work down into small items regarding the time constraint and reductionist slicing of work can disrupt the holistic view and loss of track of item dependency [14]. Thus, sprint breeds an instinct for a short-term approach. Kanban supports better project management results and team building around common and collective understanding beyond sprints. Work items can be delivered earlier, as Kanban implements WIP in the entire flow and items can behave independently, eliminating [unnecessary] dependency within work items (bundling items in a sprint is a kind of dependency).

Necessities(!): With its flexible structure, Kanban allows teams to be more self-organizing. Thus, activities that are not necessary and done at specific times just to comply with the "necessities" of the framework can be avoided (The calendar-based, not sprint-based events, in the hybrid solutions are a good example of this). Moreover, teams can develop actions in a more sensitive, lively and instantaneous (synchronous) manner in addition to the a-synchronous actions offered by the framework. For example, teams should not wait for the end of the sprint for feedbacks or for the beginning of the sprint for new work to take in.

Scrum provides a form with its “rigid” structure for cultures that need order that is relatively missing in Kanban. In contrast, each "game rule" in Scrum is hand-binding, and Scrum's predictable, rigid, artificial boundaries promise an average-level-performing goal and may hinder average teams from performing higher. “Increasing the length of feedback loop” is an example of this; Because it recommends that feedback is *mainly* managed through an artificial structure, not emergently. Generally speaking, Scrum's addiction to asynchronous communication can be a challenging factor where synchronous communication is the remedy. Scrum events are a sort of prediction. Their ability to manage a different flow is low. On the other hand, since Kanban does not come with a relatively rigid form, it seems more challenging for it to take a shape. Moreover, Kanban teams are expected to be internally motivated. The driving force is internal, not external [as in Scrum]. In teams that do not have this intrinsic strength, therefore, applying Kanban can make the situation worse. Even for motivated teams, being in a homogeneous and flat flow in Kanban may not support the feeling of progress. Against the lack of control, not having a feeling of progress, and lack of deadline matters in Kanban, Scrum’s sprint structure comes with solutions.

Transition: The fact that Kanban is closer to Lean as root than Scrum seems to have made it simpler. Scrum, rather, is like a luxury framework with its add-ons at a high cost. This luxury also applies to the transition to it. Even at the entry level, Scrum requires a threshold to endure, and teams must be accompanied until they pass this threshold in a healthy way. Because the transition steps are bulk, teams suddenly find a more tough challenge and have to deal with it even in the first place. Radical transitions to this rigid and costly structure seem to threaten and wear out the teams.

In Kanban where progress rather than transition is essential, the transformation is easier. Kanban focuses on the flow (process) rather than providing agility through teams' (re)organizations for transitions. Kanban does not also require any strictly pre-defined roles or processes adopted in a revolutionary way like in Scrum, rather recommends an iterative and incremental (shortly agile) way of implementation and an easy way for transition and use. It does not have a special preference and compulsion regarding the setup of the teams. In this sense, with its principle “start with what you do now” and “initially, respect current processes, roles, responsibilities and job titles” [59], it can align and coordinate existing teams at upper levels. However, as reported by study [13], it similarly seems from our results that current studies covered in our work are mostly restricted to process and people levels and have not yet been scaled to the organization level including such as portfolio project levels or being used as a tool for decision-making by management.

Miscellaneous (Unclassified): Kanban does not deal with the types of works in the flow, they are regarded as homogeneous. Scrum, on the other hand, considers the works as complex by default and allows to handle such works according to their nature. Scrum, thus, seems to be more advantageous than Kanban in complex projects.

Scrum is better at cycle time while Kanban is better at lead time. Scrum encourages more collaboration within the team. While Scrum is better in intra-team communication, it seems problematic in inter-team communication. Scrum teams do not always go on the “happy path”, the other side of the coin comes to the surface and it is seen that the problems between people (experienced in hierarchical structures) are present to them as well.

Depending on the situation, specialties still seem necessary by considering a balance to strike between specialization and generalization. Although some issues with the PO role generating an extra layer and dependency have been witnessed, the hybrid solutions including the PO role have been commonly observed. It means there seems to be a need for a mediator, the PO role, which acts as a bridge between the customers and the development teams. Having the PO in the process may also ensure that a representative of the customer is involved. The PO can also bring simplicity to the structure.

Both Scrum and Kanban seem to be good in delivery time, tracking and overview of projects, and teamwork. Where one is weak, the other one can be strong. These and similar situations still make these two methods preferable. However, the reported disadvantages of the particular methods (especially Scrum) open gates to integrating the most advantageous part of the methods and eliminating their disadvantages at the same time by hybridizing them.

Regarding the hybrid models, sprint is almost non-existent, instead, flow is extensively used. In relation to this, it is seen that the SM role is also positioned very few. Mostly, the roles and events are integrated from Scrum. One of the reasons can be an endeavor to use the ready-roles of Scrum that are not defined in Kanban or not abandoning already-used Scrum roles after the transition to a hybrid one. We have seen that Daily is used as a common communication manner in the hybrid models. Nevertheless, it seems that some teams do not prefer to feed a rigid and relatively bloated framework, like Scrum, just for the sake of it. They tend to avoid the high cost of events of Scrum and (seldomly) want to make them when necessary or (mostly) on a calendar base. However, there is a tendency to do the events as routine (calendar/iteration/release) based rather than only as needed. Additionally, there is still a desire to predict the future by the teams, but they do not want to limit themselves to a deadline. In providing forecasting/estimations, there are certain studies that consider not only size but quantity and size criteria together. Some solutions break Scrum's rigid team structure and redesign it according to the needs. Similarly, the solution that converts Kanban's one-way flow to a two-way flow direction is also observed.

We have seen that Scrum has advantages in requirements management phases and many solutions are trying to address these phases with Scrum particles. Especially in this area, there is a preference for the product-based structuring that may be easier to manage with Scrum. In general, while Scrum seems good at defining the projects, Kanban stands out in the development part of it.

In our study results, it has been observed that Kanban provides an advantage over Scrum in general. The results of the study also exhibit a considerable percentage of transitions to hybrid models (from Scrum). Most organizations adopted Scrum before Kanban [24]. This may be one reason why the transitions are mostly from Scrum. On the other hand, studies [38, 46, 55] report that an increasing number of companies previously using Scrum is switching to Kanban due to the issues that Scrum bears.

Considering Kanban's linear and Scrum's deterministic approaches with the effect of the manufacturing sector to a certain extent, one of the recommendations of our work would be to use Scrum and Kanban with the “AND” operator, not “OR” as we have seen how they complement each other and how they produce better solutions together. We recommend providing varying hybrid models relevant to different contexts of organizations. When choosing a method, it is more appropriate to adopt it in a way that will be beneficial to the organizations themselves [14], rather than fully complying with that particular method; it is recommended to blend the methods according to the needs instead of blindly adhering to one method. For such an integration, it is reminded that Kanban is methodology independent; it can be applied to teams implementing Scrum, Extreme Programming, or Waterfall at the operational levels [39]. It is also proposed that articles that are going to be focusing on the integration of multiple Agile methods might be providing more insightful information, especially for the practice.

VI. Limitations and Future Work

The procedures used in our study have limitations in several ways. More likely, we may have missed some relevant studies as we did not include all possible libraries. In particular, we have missed the studies published in non-peer-reviewed resources. Although the studies included and excluded were checked by other researchers, for most of the studies, a single researcher extracted the data from the included studies. The values of the quality assessment criteria are somewhat subjective. Also, the primary studies' results are context dependent and their generalizability may be low and problematic when ignored.

We are planning a further study that will evaluate the combining models to grab their common patterns from different views and we will recommend our own new model(s) with an eclectic approach. The proposed models will be suited best for certain contexts and be selectable accordingly with guidance to know the cost, trade-offs and (dis)advantages of the selections. Before finalizing the proposed models, we will conduct evaluations with experts in the field of Agile Software Development to develop the model iteratively and incrementally.

References

[1] Noureddine, A. A., Meledath, D., & Samira, Y., "A Framework for Harnessing the Best of Both Worlds in Software Project Management: Agile and Traditional", Information Systems Education Conference, 2009.

[2] Henderson, P., "Why large IT projects fail", ACM Transactions on Programming Languages and Systems, vol. 15, no.5, pp.795–825, 2006.

[3] Conn, S., "A New Teaching Paradigm in Information Systems Education: An Investigation and Report on the Origins, Significance, and Efficacy of the Agile Development Movement", Information Systems Education Journal, vol. 2, no. 15, pp.3 – 18 2004.

[4] Zayat, W., & Senvar, O., "Framework study for agile software development via scrum and Kanban", International journal of innovation and technology management, vol. 17, no.4, 2020.

[5] Weflen, E., MacKenzie, C. A., & Rivero, I. V., "An influence diagram approach to automating lead time estimation in Agile Kanban project management", Expert Systems with Applications, 187, 2022.

[6] Alaidaros, H., & Omar, M., "Software project management approaches for monitoring work-in-progress: A review", Journal of Engineering and Applied Sciences, vol. 12, no.15, pp. 3851-3857, 2017.

[7] Lei, H., Ganjeizadeh, F., Jayachandran, P. K., & Ozcan, P., "A statistical analysis of the effects of Scrum and Kanban on software development projects", Robotics and Computer-Integrated Manufacturing, vol. 43, pp.59-67, 2017.

[8] Aurisch, R., Ahmed, M., & Barkat, A., "An outlook at Agile methodologies for the independent games developer", International Journal of Computers and Applications, vol. 43, no.8, pp. 812-818, 2021.

[9] Alqudah, M., & Razali, R., "An empirical study of Scrumban formation based on the selection of scrum and Kanban practices", Int. J. Adv. Sci. Eng. Inf. Technol, vol.8, no.6, pp.2315-2322, 2018.

[10] Tripathi, N., Rodriguez, P., Ahmad, M. O., and Oivo, M., "Scaling kanban for software development in a multisite organization: Challenges and potential solutions", Agile Processes, in Software Engineering, and Extreme Programming, pp. 178–190, 2015.

[11] Rodriguez, P., Partanen, J., Kuvaja, P., and Oivo, M., "Combining lean thinking and agile methods for software development: a case study of a Finnish provider of wireless embedded systems detailed", 47th Hawaii International Conference on System Sciences (HICSS), pp. 4770–4779, IEEE, 2014.

[12] Ozkan, N., & Gök, M. Ş., "How Scrum Inhibits Agility", 15th Turkish National Software Engineering Symposium (UYMS) pp. 1-6, IEEE, 2021.

[13] Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M., "Kanban in software engineering: A systematic mapping study", Journal of Systems and Software, vo.137, pp.96-113, 2018.

[14] Law, E. L. C., & Lárusdóttir, M. K., "Whose experience do we care about? Analysis of the fitness of scrum and kanban to user experience", International Journal of Human-Computer Interaction, vol.31, no.9, pp.584-602, 2015.

[15] Banijamali, A., Dawadi, R., Ahmad, M. O., Similä, J., Oivo, M., & Liukkunen, K., "An empirical study on the impact of Scrumban on geographically distributed software development", 4th international

conference on model-driven engineering and software development (MODELSWARD), pp. 567-577, IEEE, 2016.

[16] Ladas, Cc. Scrumban-essays on kanban systems for lean software development: Modus Cooperandi Press, 2009.

[18] Alqudah, M., & Razali, R., "A comparison of scrum and Kanban for identifying their selection factors", 6th International Conference on Electrical Engineering and Informatics (ICEEI) pp. 1-6, IEEE, 2017.

[19] Kniberg, H., & Skarin, M., Kanban and Scrum-making the most of both: Lulu. com, 2010.

[20] Herdika, H. R., & Budiardjo, E. K., "Variability and commonality requirement specification on agile software development: Scrum, xp, lean, and kanban", 3rd International Conference on Computer and Informatics Engineering (IC2IE), pp.323-329. IEEE, 2020.

[21] Ahmad, M. O., Markkula, J., & Oivo, M., "Kanban in software development: A systematic literature review", 39th Euromicro conference on software engineering and advanced applications, pp. 9-16, IEEE, 2013.

[22] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering--a systematic literature review," Inf. Softw. Technol., vol. 51, no. 1, pp. 7–15, 2009.

[23] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3," Engineering, vol. 45, no. 4ve, p. 1051, 2007.

[24] https://tinyurl.com/2p89fvex.

[25] Anderson, D. J., Concas, G., Lunesu, M. I., Marchesi, M., & Zhang, H., "A comparative study of Scrum and Kanban approaches on a real case study using simulation", International Conference on Agile Software Development, pp. 123-137, Springer, Berlin, Heidelberg, 2012.

[26] Polk, R. "Agile and Kanban in coordination", Agile Conference, pp. 263-268, IEEE, 2011.

[27] Laanti, M., "Agile and Wellbeing--Stress, Empowerment, and Performance in Scrum and Kanban Teams, 46th Hawaii International Conference on System Sciences, pp. 4761-4770, IEEE, 2013.

[28] Gomes Filho, A. F., Alencar, D., & Toledo, R. D., "Agile in 3D: Agility in the Animation Studio", Brazilian Workshop on Agile Methods, pp. 63-76, Springer, Cham, 2017.

[29] Hirner, H., Lavicka, M., Schefer-Wenzl, S., & Miladinovic, I., "Agile Software Integration in Telecommunications—a Case Study", 27th Telecommunications Forum (TELFOR), pp. 1-4, IEEE, 2019.

[30] Cruz, E. F. C. D., Fernandes Junior, F. E., & Sardinha, E. D., "An experience in the use of SCRUM and KANBAN for project development in a Waterfall environment", XX Brazilian Symposium on Software Quality, pp. 1-7, 2021.

[31] Srivastava, A., Mehrotra, D., Kapur, P. K., & Aggarwal, A. G., "Analytical evaluation of agile success factors influencing quality in software industry", International Journal of System Assurance Engineering and Management, vol.11, no.2, pp.247-257, 2020.

[32] da Cruz, A. F. et. al., "Blueprint model: An agile-oriented methodology for tackling global software development challenges", Advances in Science Technology and Engineering Systems Journal, vol.5, no.6, pp.353-362, 2020.

[33] Gulliksen Stray, V., Moe, N. B., & Dingsøyr, T., "Challenges to teamwork: a multiple case study of two agile teams", International conference on agile software development, pp. 146-161, Springer, Berlin, Heidelberg, 2011.

[34] Terlecka, K., "Combining Kanban and Scrum--Lessons from a team of sysadmins" Agile Conference, pp. 99-102, IEEE, 2012.

[35] Saltz, J., & Crowston, K., "Comparing data science project management methodologies via a controlled experiment", Hawaii International Conference on System Sciences, 2017.

[36] Granulo, A., & Tanović, A., "Comparison of SCRUM and KANBAN in the Learning Management System implementation process", 27th Telecommunications Forum (TELFOR), pp. 1-4, IEEE, 2019.

[37] Shimoda, A., & Yabuki, T., "Cost and value analysis of software development method focused on individual function", Eighth International Conference on Intelligent Computing and Information Systems (ICICIS), pp. 237-243, IEEE, 2017.

[38] Nikitina, N., & Kajko-Mattsson, M., "Developer-driven big-bang process transition from Scrum to Kanban., International conference on software and systems process, pp. 159-168, 2011.

[39] Majchrzak, M., & Stilger, L., "Experience report: Introducing Kanban into automotive software project" e-Informatica Software Engineering Journal, vol. 11, no.1, 2017.

[40] Birkeland, J. O., "From a timebox tangle to a more flexible flow", International conference on agile software development, pp. 325-334), Springer, Berlin, Heidelberg, 2010.

[41] Rutherford, K., Shannon, P., Judson, C., & Kidd, N., "From chaos to kanban, via scrum", International Conference on Agile Software Development, pp. 344-352, Springer, Berlin, Heidelberg, 2010.

[42] Gelmis, A., Ozkan, N., Ahmad, A. J., & Guler, M. G., "Impact of Turkish National Culture on Agile Software Development in Turkey", International Conference on Lean and Agile Software Development, pp. 78-95, Springer, Cham, 2022.

[43] McKenzie, T., Morales-Trujillo, M., Lukosch, S., & Hoermann, S., "Is agile not agile enough? A study on how agile is applied and misapplied in the video game development industry", IEEE/ACM Joint 15th International Conference on Software and System Processes (ICSSP) and 16th ACM/IEEE International Conference on Global Software Engineering (ICGSE), pp. 94-105, IEEE, 2021.

[44] Seikola, M., & Loisa, H. M., "Kanban implementation in a telecom product maintenance", 37th EUROMICRO Conference on Software Engineering and Advanced Applications, pp. 321-329, IEEE, 2011.

[45] Raju, H. K., & Krishnegowda, Y. T., "Kanban Pull and Flow—A transparent workflow for improved quality and productivity in software developmet", Fifth International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom), pp. 44-51, IET, 2013.

[46] Sjøberg, D. I., Johnsen, A., & Solberg, J., "Quantifying the effect of using kanban versus scrum: A case study", IEEE software, vol. 29, no. 5, pp.47-53, 2012.

[47] Hašek, F., & Mohelská, H., "Selection of a Suitable Agile Methodology–Case Study," Hradec Economic Days, 2021.

[48] Cocco, L., Mannaro, K., Concas, G., & Marchesi, M., "Simulating kanban and scrum vs. waterfall with system dynamics", International conference on agile software development, pp. 117-131, Springer, Berlin, Heidelberg, 2011.

[49] Saltz, J., & Suthrland, A., "SKI: An agile framework for data science", IEEE International Conference on Big Data (Big Data), pp. 3468-3476, IEEE, 2019.

[50] Tudjarova, S., Chorbev, I., & Joksimoski, B., "Software Quality Metrics While Using Different Development Methodologies," International Conference on ICT Innovations, pp. 240-250, 2017.

[51] Diebold, P., Theobald, S., Wahl, J., & Rausch, Y., "Stepwise transition to agile: From three agile practices to Kanban adaptation", Journal of Software: Evolution and Process, vol.31, no.5, e2167, 2019.

[52] Senapathi, M., & Drury-Grogan, M. L., "Systems thinking approach to implementing kanban: A case study", Journal of Software: Evolution and Process, vol.33, no.4, e2322, 2021.

[53] Iqbal, J., Omar, M., & Yasin, A., "The impact of agile methodologies and cost management success factors: An empirical study", Baghdad Science Journal, vol.16, no.2, pp.496-504, 2019.

[54] Willeke, E. R., "The inkubook experience: A tale of five processes", Agile Conference, pp. 156-161, IEEE, 2009.

[55] Ahmad, M. O., Kuvaja, P., Oivo, M., & Markkula, J., "Transition of software maintenance teams from Scrum to Kanban," 49th Hawaii International Conference on System Sciences (HICSS), pp. 5427-5436, IEEE, 2016.

[56] Robinson, P. T., & Beecham, S., "TWINS-This Workflow Is Not Scrum: Agile process adaptation for open source software projects, "IEEE/ACM International Conference on Software and System Processes (ICSSP), pp. 24-33, IEEE, 2019.

[57] Senapathi, M., & Srinivasan, A., "Understanding post-adoptive agile usage: An exploratory cross-case analysis", Journal of Systems and Software, vol.85, no.6, pp.1255-1268, 2012.

[58] Lunesu, M. I., Münch, J., Marchesi, M., & Kuhrmann, M., "Using simulation for understanding and reproducing distributed software development processes in the cloud", Information and Software Technology, vol. 103, pp.226-238, 2018.

[59] Anderson, D. J. Kanban : successful evolutionary change in your technology business. 2010.

[60] Anderson, D. J., & Bozheva, T., Kanban Maturity Model: Evolving Fit-For-Purpose Organizations: Lean Kanban, 2018.