The Diversity of Approaches to Support Software Project Management in the Agile Context: Trends, Comparisons and Gaps

Elielton da Costa Carvalho[a] and Sandro Ronaldo Bezerra Oliveira[b]
Graduate Program in Computer Sciense, Institute of Exact and Natural Sciences, Federal University of Pará, Belém, Pará, Brazil

Keywords: Software Project Management, Project Management Approaches, Systematic Review of Literature.

Abstract: The literature has reported a significant number of approaches to support software development teams. Due to this diversity of approaches, the process of choosing which one is the most suitable for a given project becomes a complex task as new approaches emerge. In addition, there has been no comprehensive work on what these project management approaches are, nor what their main advantages and limitations are. Therefore, it becomes important to research the characteristics of these approaches and map them to minimize decision challenges. Therefore, the objective of this work is to identify primary studies in software engineering that present approaches to support the management of software projects in the agile context. To achieve this goal, we conducted a systematic review study and selected 65 studies on software project management approaches. From these studies, we identified eight different types of approaches, with software being the most used approaches in this context. In addition, nine project areas were identified in which these approaches focus, among which "Schedule" and "Quality" stand out as the areas with the greatest focus by the approaches. Finally, we identified that the "Planning" is the phase of the project in which the approaches place more emphasis.

1 INTRODUCTION

Alignment between project management and business strategy can significantly increase the chances of organizations achieving their goals, as well as improving their performance (Gomes and Romão, 2016). In addition, agile development favoured the software industry, but required approaches that meet this new development paradigm. In this way, exploring new approaches can help organizations adapt more quickly to the new realities of the economy (Parsi, 2021).

The literature has reported a significant number of approaches (software, methodology, method, model, tool, technique, framework, practice) to support software development teams. Due to this diversity of approaches, the process of choosing which one is the most suitable for a given project becomes a complex task as new approaches emerge. It is observed, therefore, that the challenges in agile development have been and continue to be explored, but there has not been any comprehensive work on what these project management approaches are, nor what are their main advantages and limitations, which indicates the need for a further research in this area (Shrivastava and Rathod, 2017).

It is therefore important to research the characteristics of these approaches and map them to minimize decision challenges. Additionally, it is also necessary to investigate how they are used and the lessons learned from their use, in order to identify opportunities for improvement (Varajão et al., 2017). Thus, the objective of this work is to identify primary studies in software engineering that present approaches to support the software projects management in the agile context.

To achieve this goal, we performed a systematic review study and selected 65 studies on approaches to software project management. In our results, we identified 8 types of distinct approaches, 9 areas that these approaches focus on, as well as 5 project phases that they emphasize.

[a] https://orcid.org/0000-0002-9819-5357
[b] https://orcid.org/0000-0002-8929-5145

Carvalho, E. and Oliveira, S.
The Diversity of Approaches to Support Software Project Management in the Agile Context: Trends, Comparisons and Gaps.
DOI: 10.5220/0011063500003176
In *Proceedings of the 17th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2022)*, pages 138-149
ISBN: 978-989-758-568-5; ISSN: 2184-4895

In addition to this introductory section, this work is structured as follows: Section 2 presents some concepts on the topic of this research, Section 3 details the study design, Section 4 presents the results, Section 5 presents the discussions, Section 6 addresses some threats to the validity of this work, Section 7 brings some related works and Section 8 closes this work by presenting the conclusions.

2 BACKGROUND

This section introduces concepts related to software project management and software project management approaches.

The PMBOK – Project Management Body of Knowledge (2017) defines a project as a temporary effort undertaken to create a unique product, service or result. In turn, project management is defined as the application of knowledge, skills, tools and techniques to project activities in order to fulfill project requirements. Banica et al. (2018) state that the main objective of project management is to provide expected results with quality, planned time, and approved costs, in addition to being essential to take care of risks. The PMBOK Guide (2017) also mentions that several approaches can be used to assist the project manager and support team collaboration.

Project management approaches are means that aim to support the whole process or part of the management process. These approaches can be software, methodologies, practices, methods, frameworks, models, among others (Kostalova et al., 2015). These approaches contribute to a significant gain in the efficiency and effectiveness of project management. In addition, they allow a change in the way of managing projects and give professionals greater decision-making power, especially project managers (Retnowardhani and Suroso, 2019). The authors also reinforce that organizations should use them to obtain better results in their projects.

3 STUDY DESIGN

This section presents the objectives of the work, the research questions and the method used.

3.1 Goal and Research Question

This study aims to identify the approaches used to support the software projects management in agile context. We are interested in identifying the type of approach, where it is applied within the project, its main contributions and limitations, as well as the form of evaluation used by its developers.

To formalize the objective of this study, the Goal-Question-Metric (GQM) defined by Basili (1992) was used. Thus, this study seeks to:

- Analyze: primary studies, through a Systematic Literature Review (SLR),
- In order to: identify the approaches used in software project management (SPM) that are reported in the specialized literature,
- Regarding: the definition, use and evaluation of these approaches,
- From the point of view of: researchers, organizations and professionals in software project management,
- In context: industrial and academic agile software development,

Thus, we propose the following research questions (RQ):

- *RQ1: What is the name and type of approach?* – The objective is to identify the type of SPM approach that was used in the work, eg method, software, technique, etc., as well as the name of the approach,
- *RQ2: What are the strengths and limitations of the approach?* – The objective is to identify the main points that deal with the advantages and disadvantages of using the identified approach,
- *RQ3: How was the approach evaluated?* – The objective is to identify the method used to evaluate the use of the approach.

3.2 Method

To achieve the objective of this work, an SLR was conducted. SLR is an evidence-based secondary study method to systematically identify, analyze and interpret all relevant documents related to a specific research question (Kitchenham and Charters, 2007).

We performed the SLR from May/2021 to January/2022. The study was organized in four stages, adapted from (Kitchenham and Charters, 2007; Petersen et al., 2015), as follows:

- Step 1 – Definition of research questions: in this step, three research questions were defined based on the objective of the study (Subsection 3.1),
- Step 2 – Search: in this stage, based on the research questions, a replicable process was defined for carrying out the search for studies in selected scientific bases (Subsection 3.3),
- Step 3 – Study selection: in this step a replicable process was defined and applied to select only

the relevant studies according to the objective of this work (Subsection 3.4),

- Step 4 – Study classification and data extraction: in this step, based on the research questions, a strategy was defined to: (i) map the relevant data from the primary studies (Subsection 3.5) and (ii) present the results of the work (Section 4).

Two researchers participated in the planning and execution of the work: a graduate student in Computer Science and a professor / researcher with a PhD in Software Engineering.

3.3 Search Strategy

The search took place in an automated way through a string formed by a series of keywords and their respective synonyms. These keywords were defined based on the research questions, from the PICOC (Population, Intervention, Comparison, Outcomes and Context) structure suggested by Kitchenham and Charters (2007).

However, "Comparison" and "Intervention" are not part of the scope of this work. In this way, the string was formulated with terms related to (i) population, (ii) outcome and (iii) context. The terms used were:

- Population: project management,
- Result: tool, method, technique, model, technology, practice, standard, guide, work product, methodology, framework, process, principle, theme and profile,
- Context: software and agile.

Thus, we got the following search string: *(“project management”) AND (“tool” OR “method” OR “technique” OR “model” OR “technolog*” OR “practice” OR “pattern” OR “standard” OR “guide” OR “work product” OR “methodolog*” OR “framework” OR “process” OR “principle” OR “theme” OR “profile”) AND (“software”) AND (“agile”).*

The search string was applied to the IEEE Xplore and ACM DL databases. We did not search the EI COMPENDEX and SCOPUS because, according to the results of Souza et al. (2018), there was a high rate of redundancy.

3.4 Study Selection

In this step of the work, inclusion (IC) and exclusion (EC) criteria were applied in order to select only the relevant works that answered our research questions. The IC and EC are presented below.

- IC: Studies that present some software project management approach applied in the agile context and studies that evaluated the approach used,
- EC: Studies that are not written in English, studies not available for download openly or through the institutional IP of the researchers, studies such as workshop reports, posters, presentations, speaker keynotes, books, theses and dissertations.

Each of the studies underwent a selection process consisting of four steps: (i) two researchers read the titles and abstracts of all studies and applied the exclusion criteria, this step was defined as pre-selection, (ii) the same researchers discussed differences in the application of exclusion criteria to reach a consensus, (iii) the researchers read the title and abstract, and the full text if necessary, of the studies selected in the first step to apply the inclusion criteria, (iv) the researchers discussed differences in the application of exclusion criteria to reach a consensus.

Regarding the IEEE, a total of 1339 studies were returned, among which 1235 were excluded after pre-selection, leaving 104 studies. About ACM, 1486 studies were returned when running the search string. Of this total, 1233 were excluded in the pre-selection. Thus, 253 studies remained. After pre-selection, the remaining 357 primary study were read in full and submitted to the IC and EC.

The process resulted in 65 primary studies (25 from IEEE and 40 from ACM) (available at *https://zenodo.org/record/5876282#.YefGov7MLIV*).

3.5 Study Classification and Data Extraction

To collect the necessary data that answer the research questions defined for this work, a researcher was responsible for reading the 65 selected studies.

Data analysis aims to classify the studies according to the proposed research questions. Therefore, the result of this SLR should map and classify studies regarding: the presence of software project management approaches, the project area and phase approach was used, the advantages and disadvantages of its use, and the form of evaluation of the approach.

4 RESULTS

This section presents the results of SLR. Subsection 4.1 presents an overview of the results. Subsections 4.2, 4.3, and 4.4 describe the results for RQ1, RQ2, and RQ3, respectively. In these subsections, the primary studies will be referenced and identified by codes and are available at the URL presented in Subsection 3.4.

4.1 Overview

SLR looked for works between the years 2001 and 2021. However, the 65 selected primary studies are distributed between the years 2004 and 2021, as shown in Figure 1. Still based on Figure 1, it is possible to notice that, despite a decrease in the number of studies after 2008, the trend is for a growth in the number of publications related to the subject of this work, with a sharper peak from 2015. However, it is also noted that the years 2020 and 2021 showed a decrease significant in the number of publications. According to Yanow and Good (2020), there was a reduction in the number of publications in 2020 as many universities and companies had to reduce their research activities, as the laboratories were closed. It should also be taken into account that the number of studies in 2021 is lower as a result of the period of execution of this work.

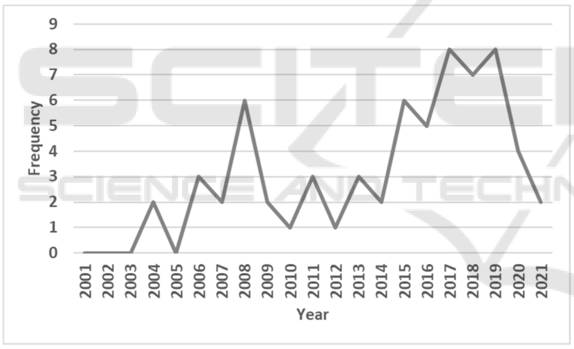

Figure 1: Distribution of studies per year.

Figure 2 presents the project areas where the identified approaches are used. Based on this figure, it is possible to verify that the focus of the approaches is mainly focused on the areas of schedule (22 studies), quality (19 studies) and communications (17 studies), as is the case of the following primary studies: [PS61], [PS59] and [PS30], respectively.

Figure 3 illustrates the project phases where the approaches identified in this work are used. It is possible to note that the approaches give more emphasis to the project planning phase (36 studies), in addition to the monitoring and control (28 studies) of the same, as can be exemplified by [PS64] and [PS10], respectively.

It is important to note that some studies focus on more than one phase of the project at the same time, as is the case, for example, of the following studies: [PS07, PS12, PS28, PS54, PS65]. The same applies to the project areas, because, as in the phases, there are studies that are focused on more than one area simultaneously, for example: [PS04, PS34, PS39, PS47, PS57].

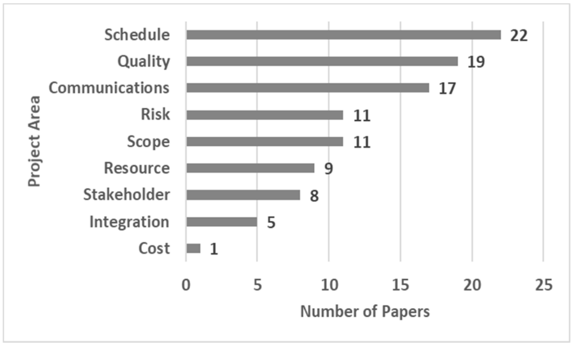

Figure 2: Project areas where identified approaches are used.

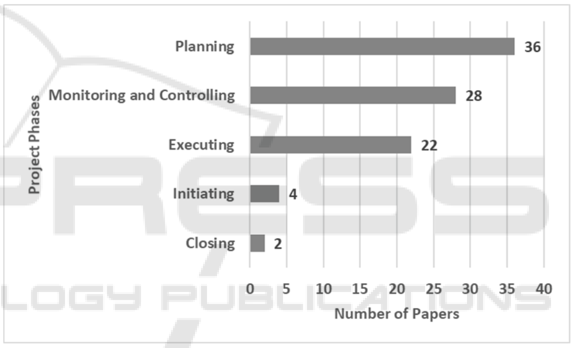

Figure 3: Project phases where identified approaches are used.

4.2 Type of Approach

This subsection presents results referring to RQ1 (“What is the name and type of approach?”).

From the selected studies, it was possible to identify eight types of approaches, they are: software, methodology, method, model, tool, technique, framework and practice. Figure 4 illustrates the types of approaches identified, as well as the number of studies that address each of them.

Still based on Figure 4, it is possible to verify that “software” is the most common type of approach used to support the software projects management, this approach being identified in 15 studies. Soon after, approaches of the “methodology” type are the ones that stand out the most, developed and / or used in 12 studies. Finally, among the types of approaches that are most used, “method” appears in third place, with this type of approach being identified in 11 primary studies.

It is worth noting that "framework" and "practice" approaches are the least used, according to data extracted from primary studies. Such approaches were observed only in four and two studies, respectively.

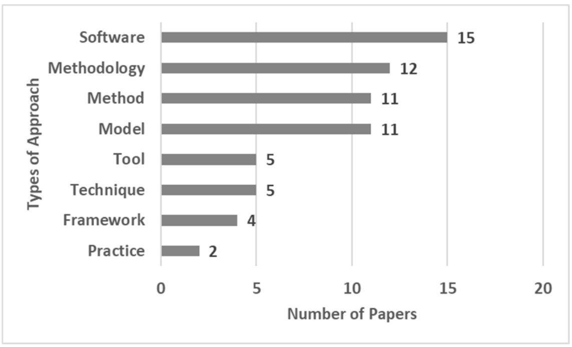

Figure 4: Types of approaches identified.

4.2.1 Software

As mentioned earlier, software is the most common type of approach used to support software project management. Table 1 presents the studies that present this type of approach.

Table 1: Studies that present a "Software" approach.

Studies that present a "Software" approach
[PS01], [PS02], [PS06], [PS07], [PS10], [PS14], [PS16], [PS21], [PS30], [PS40], [PS43], [PS44], [PS52], [PS61], [PS62]

Morgan and Maurer (2006) [PS44] report in their study the use of MasePlanner. MasePlanner is software aimed at planning communication in the project. It is software that, according to the authors, supports interactions, facilitating non-verbal communication. MasePlanner provides teams with a digital environment that supports information management in addition to natural interactions. With respect to support for natural interaction, MasePlaner allows planning work products to be created, edited and organized in a similar way to paper planning meetings.

More recently, Alhazmi and Huang (2018) [PS01] developed the Sprint Planning Decision Support System (SPESS). This software aims to assist managers in sprint planning. SPESS is primarily based on three factors: developer competence, developer seniority, and task dependency. This tool aims to assign the tasks of each Sprint to the developers ensuring that each team member contributes to the maximum of their potential, and the project planning is optimized for the shortest possible time.

The other studies that use software are focused on communication [PS06, PS10, PS30, PS40, PS52], quality [PS02, PS07, PS14, PS16, PS62], schedule [PS07, PS14, PS43, PS61], scope [PS07] and stakeholders [PS21]. It is worth noting that software does not necessarily serve only one area of the project. Begosso et al. (2019) [PS07], for example, present SimScrumF. This software is a game that focuses on promoting student engagement in the process of learning the concepts of Scrum methodology, widely used to manage software development projects around the world. The game addresses the process of managing the scope defined for the project, aiming to deliver a product with quality, within the pre-established schedule.

4.2.2 Methodology

Behind only software, methodologies were the most used approaches to support project management. Table 2 presents the studies that focus on this type of approach.

Table 2: Studies that present a "Methodology" approach.

Studies that present a "Methodology" approach
[PS04], [PS09], [PS12], [PS13], [PS17], [PS20], [PS28], [PS37], [PS39], [PS47], [PS53], [PS54]

Castillo-Barrera et al. (2018) [PS13] state that before starting the execution of the project, it is necessary to have previously carried out an analysis and also a synthesis of the information that permeates it. To this end, the authors present BloomSoft, a methodology adapted from Bloom's Taxonomy and used in conjunction with Scrum, which aims to support teams in planning the construction, integration and testing of the software that will be developed in the project. Not only that, as the authors also emphasize that the methodology makes it possible to have an agile way of classifying the complexity of user stories based on the verbs identified in them and, concomitantly, to determine from this classification the stage of Software Development to which it belongs. Thus, the development team has the possibility to classify the stories in stages and, with that, make a better planning for each Sprint.

Bierwolf et al. (2017) [PS09] report the use of DevOps methodology in software project management. DevOps aims to mitigate risks, in particular to achieve a stable, secure and reliable production environment. Through this form of communication and collaboration, all stakeholders manage any uncertainties that arise, for example, due to changes in technology or any changes in the environment. The work also performs a comparison between the results of using traditional approaches in relation to DevOps approaches. The comparison is mainly related to aspects such as control and mitigation of uncertainties and risks.

There is also a methodology that varies from Scrum and is presented by Baptista (2008) [PS04]. uScrum, the methodology described in the aforementioned study, manages uncertainty and the unknown, allowing the team involved in the project to react quickly to changes in the most diverse conditions. uScrum allows the team to effectively prioritize regular work alongside the more difficult creative work. The methodology prioritizes tasks in monthly iterations based on their importance, urgency and timeliness and, like Scrum, they are also called sprints.

4.2.3 Method

Another type of approach that stands out are methods, as this type of approach was identified in 11 of the 65 selected studies and, together with models, is the third most used to support software project management. Table 3 presents the studies that deal with methods.

Table 3: Studies that present a "Method" approach.

Studies that present a "Method" approach
[PS03], [PS05], [PS08], [PS11], [PS15], [PS23], [PS26], [PS31], [PS50], [PS56], [PS64]

Haugen (2006) [PS26] describes the use of Planning Poker. The aim of the study is to examine whether introducing the planning poker process into user story estimation improves estimation performance compared to unstructured group estimation. In this process, the customer first explains each user story to the developer group. Developers then discuss the work involved in implementation to the point where everyone feels they have enough information to estimate the effort required. All developers then estimate the user story independently and reveal their estimates simultaneously. Next, the developers with the lowest and highest estimates justify their estimates, and the group continues the discussion to decide on a collective estimate, possibly conducting one or more additional rounds of individual estimates.

Zhang et al. (2016) [PS64] address in their study on Early Software Size Estimation (ESSE), a method that can extract semantic features from natural language requirements automatically and build size estimation models for the project. ESSE performs a semantic analysis of requirements specification documents by extracting information and disseminating activation. Then, characteristics related to complexity are extracted from the semantic analysis results. In addition, ESSE extracts local resources and global resources to do word-level semantic analysis. Then, using these features, size drivers and actual sizes from historical project data, the size estimation model can be established by regression algorithms. Finally, ESSE can estimate the size of a new project using this size estimation model.

The other studies that use some software project management methods are concerned with managing quality [PS05, PS50], schedule [PS11], resources [PS31, PS56], scope [PS03] and project risks [PS23]. In addition, these same studies are focused on the following project phases: planning [PS11, PS23, PS31], execution [PS50] and monitoring and control [PS03, PS05, PS23, PS31, PS56].

4.2.4 Model

As well as the methods, it was possible to identify, through the selected studies, 11 models that aim to support the software projects management in agile context. Table 4 presents the studies that focus on this type of approach.

Table 4: Studies that present a "Model" approach.

Studies that present a "Model" approach
[PS22], [PS25], [PS27], [PS29], [PS34], [PS36], [PS46], [PS48], [PS55], [PS58], [PS59]

Godoy et al. (2019) [PS22] present a new software development tool based on agile methodologies Scrum and Kanban and adapted to the current trend of the global software environment. The Blueprint model proposes lightweight project management that is combined with a team organization to encourage and facilitate communication between teams in different locations. Blueprint introduces important adaptations to Scrum and Kanban to reduce unwanted bureaucracy and to facilitate global software development.

Perkusich et al. (2013) [PS48] developed a Probabilistic Model with Baysian Network. As the name suggests, the model is a Bayesian network that represents a software development project managed in essence with the Scrum methodology. Scrum Masters should use it to identify project issues and guide the team to improve the project's chances of success. The model produces data with probability values that represent the current status of key project factors. It should be used to identify problems and prioritize areas for improvement. The prioritization of areas for improvement should be a collaborative activity and the model should only be used as a source of information to guide the discussion.

The remaining studies that deal with some model focus on project schedule management [PS34, PS36, PS58]. In addition to the aforementioned studies, there are those focused on risk management [PS27, PS46], communication [PS25], scope [PS29], stakeholders [PS55] and quality [PS59]. These same studies still focus on different phases of the project, such as: planning [PS25, PS46, PS58, PS59], execution [PS36, PS55] and monitoring and control [PS27, PS29].

4.2.5 Tool

Based on the selected studies, five tools were identified that support the software projects management. Table 5 presents these studies.

Table 5: Studies that present a “Tool” approach.

Studies that present a “Tool” approach
[PS18], [PS38], [PS42], [PS49], [PS63]

Vivian et al. (2015) [PS63] expose in their study a Panel to View Online Teamwork Discussions. The tool is a dashboard that extracts and communicates the distribution of team roles and members' emotional information in real time. The panel is composed of a series of elements: team participation and distribution of roles, team and individual sentiment analysis and team and individual emotions. It provides real-time analysis of teamwork discussions and visualizes team members' emotions, the roles they have taken, and the overall team sentiment during the course of a collaborative project.

Mateescu et al. (2015) [PS42] present a tool called aWall. It is an agile team collaboration tool for large multi-touch wall systems. aWall was designed based on empirical user research using new concepts of interaction and visualization to support and promote the highly collaborative and communicative agile work style. The tool is based on web technology and can be used both in co-located and distributed environments. According to the study authors, the tool can be crucial for agile teams, since the agile process depends on intense interaction, collaboration and constant open communication between team members. The remaining studies that deal with tools turned their attention to the monitoring and control of the project schedule, and to the planning of the scope. Fehlmann and Kranich (2017) [PS18], for example, through a Bayesian Approach Burn-up Chart, were able to provide estimates of how much additional time is needed to complete the planned work.

4.2.6 Technique

Following what was verified regarding the tools, from the selected studies, five techniques were also identified that aim to support the software projects management, as shown in Table 6

Table 6: Studies that present a “Technical” approach.

Studies that present a “Technique” approach
[PS33], [PS35], [PS41], [PS51], [PS60]

Stapel et al. (2011) [PS60] present Flow Mapping, a technique of the FLOW Method to plan and guide communication in distributed development projects. To achieve these goals, the technique is centered on the visualization of a FLOW map. A FLOW map is a special FLOW model (i.e. visualization of project participants, documents and information flows) extended by features to improve awareness in distributed teams. According to the authors, when using the Flow Mapping approach, the communication of a distributed project can be planned in a working day.

Kroll et al. (2017) [PS35] used a Genetic Algorithm-Based Assignment technique. The technique was used to assign tasks in a global software development (GSD) project. The technique uses a queue-based GSD simulator to evaluate the fitness function. Results based on a multiple case study (applying the technique to data from three real-world projects) show that the approach could be better than project managers' task assignments.

4.2.7 Framework

Four studies present some framework that supports the software projects management in agile context, as can be seen in Table 7.

Table 7: Studies that present a “Framework” approach.

Studies that present a “Framework” approach
[PS19], [PS24], [PS32], [PS45]

Silva and Oliveira (2016) [PS19] developed an Agile Project Portfolio Management Framework. The framework refers to a flexible approach to portfolio management, suggesting faster and more dynamic meetings, focusing mainly on the interaction and commitment of those involved in the process. Some agile practices that can be used in each activity of this framework are: planning portfolio management, identifying new proposals, analyzing candidate projects, composing the project portfolio and monitoring portfolio.

Guerrero et al. (2019) [PS24] present Eagle, a framework that supports a systematic way to define, measure and visualize the practices of members of software development teams following agile principles. Specifically, the framework provides microservices architecture based on the "Governify" ecosystem for managing accordingly. The framework provides an ecosystem of tools for organizations to define their best practices to follow and track the adherence of their teams and members in order to learn about their pitfalls and improve the project over time.

Jain and Suman (2018) [PS32] expose in their study on a project management framework for global software development (GSD). The GSD Project Management Framework, as the authors call the approach, assimilates the knowledge areas of the PMBOK with the knowledge areas necessary for an effective management of the GSD. It would guide the project manager on aspects to consider when executing distributed projects. The presented framework covers feasibility and risk management, virtual team management, knowledge management, scope and resource management, performance management and GSD integration management.

4.2.8 Practice

As shown in Figure 4, practices were the least identified types of approaches in the selected studies, with a total of two studies that only address some of them, as shown in Table 8.

Table 8: Studies that present a "Practice" approach.

Studies that present a "Practice" approach
[PS57], [PS65]

Schreiber et al. (2017) [PS57] discuss a practice called Metrics Driven Research Collaboration (MEDIATION). The practice aims to ensure that all project participants have an ongoing common objective: the success of the project. As per established practice, the project team should focus on the most important requirements and continually verify that the software product conforms to the defined scope and corresponding metrics. The practice further establishes that the status, challenges and progress of the project be transparent to all team members at all times.

Zhang et al. (2020) [PS65] implemented a practice called Fireteam, a practice that focuses on small teams in the software industry. Fireteam is nothing more than a teamwork style. The practice defines two to five members to handle the division of labor and coordination issues in traditional development teams. Briefly, the practice aims to reduce project management overheads and improve productivity, through the institutionalization of the practice of small teams throughout the organization, to solve problems arising from human and social aspects, such as friendship, talent, skill and communications.

4.3 Strengths and Limitations of Approaches

This section gives an overview of the main advantages and disadvantages / limitations of the approaches identified from the selected studies.

4.3.1 Strengths

When a researcher / developer proposes to develop a certain approach, he is looking for some means that help him to break specific barriers of the context in which he is inserted. Regarding the approaches that aim to support the software projects management, identified from the selected studies, it is no different. They all have some advantages and even though they are, in some cases, different from each other, at the end of the day they all have the same objective: to efficiently support software development teams.

Some approaches, despite being virtual, try to be as close as possible to the real world [PS38], as with the tool developed by Liskin and Schneider (2012) [PS38]. Likewise, the approaches identified always try to reduce the effort required by the team to complete a project task [PS34, PS56, PS58, PS59]. These approaches are considered robust enough to identify problems in the project and correct them in time, avoiding further damage to resources, schedule and, consequently, to the project [PS48].

Also noteworthy are those approaches that provide great ease in terms of planning communications, as is the case of Flow Mapping presented by Stapel et al. (2011) [PS60] that allows communication planning in up to one day. In addition, it has an approach that allows the reduction of Sprint time, when working with Scrum, without the project losing quality, even if the reduction is minimal, as reported by Alhazmi and Huang (2018) [PS01], and Kroll et al. (2017) [PS35].

4.3.2 Limitations

One of the points to take into account when we are talking about any type of approach, whatever the purpose it was developed for, are its limitations (now called disadvantages). Limitations are inherent to any work, especially when it comes to team-focused approaches. Regarding software project management approaches, there are a number of limitations identified in some selected studies and that end up extending to the others.

Initially, it was possible to observe that there is an approach that focuses, among other things, on the constant repair of the project and the work products generated by it. However, the act of constantly fixing defects can bring additional efforts to developers, as reported by Tang (2008) [PS34]. There are also approaches that rely entirely on solid and equal commitment from all team members. If the team is not 100% focused and committed, the project naturally tends to fail [PS12, PS28, PS47, PS54], as less engaged employees could perform activities inappropriately, even with the help of the approach, as Godoy et al. (2019) [PS22].

Some approaches are not so recommended for planning a small number of tasks, as they were developed to deal with large volumes of data, as with the Sprint Planning Decision Support System (SPESS), developed by Alhazmi and Huang (2018) [PS01]. Other approaches do not constantly monitor the project and this limits the team's view of the progress of what is being developed, as Stapel et al. (2011) [PS60].

With regard specifically to software, many plugins that are developed to compose them end up falling into disuse quickly, as is the case of plugins developed for Redmine, reported in the study by Dowling and McGrath (2015) [PS16]. For the authors, while there is an active community of developers working on Redmine and creating valuable plugins, this is something of a double-edged sword. That's because, plugins are not always compatible with the latest versions, causing headaches during the update and failures in functionality of features that teams previously used.

4.4 Evaluating Approaches

This section presents the main ways used by the authors of the selected studies to evaluate and validate their approaches.

The way most used by the authors of the works to evaluate their approaches was the case study, with a total of 55 studies that used this method. Trapa and Rao (2006) [PS61], for example, applied their approach to a project called Cronos, where user stories were divided into tasks and inserted into the software. Each task was assigned to two developers who were responsible for assigning the time estimate for each task they were responsible for. After that, reports were generated with information about the project completion time and the impacts of the additions of new functionalities that came to exist.

Some studies used questionnaires to obtain feedback on the approach developed, five studies more precisely. Bastarrica et al. (2017) [PS05] formulated two research questions to identify the advantages and disadvantages of using their approach. Data collection involved a structured questionnaire that was made available to the CEOs and a technical professional from each company. The purpose of the questionnaire was to capture actual and desired practices in relation to project management.

Regarding observational assessments, a total of three studies used this assessment method. We can highlight the study by Mateescu et al. (2015) [PS42] who, from the observation of the use of the proposed approach, were able to identify positive points and points for improvement, as well as conclusions about how efficient the proposed approach can be in a project. The authors verified that aWall takes advantage of the resolution of web technology and thereby surpasses the possibilities of existing desktop tools. They noted that each agile meeting has its main task and specific objective, but also needs various supporting information and artifacts.

Finally, two studies conducted internal evaluations, that is, the developers of the approaches themselves evaluated them based on their own metrics, without necessarily having made the approaches available to third parties. Bruegge et al. (2009) [PS11] evaluated the feasibility and performance of their approach from two classification experiments. The first experiment classified the tasks according to the activity to which they belonged. In the second experiment, the approach was employed to classify the status of tasks, that is, the machine learning engine predicts whether these tasks have already been completed or not.

5 DISCUSSION

This section presents our main conclusions and impressions of the results presented in Section 4.

The first item to note refers to the results on which areas and phases of project management the approaches emphasize. As illustrated in Figure 2, the areas of schedule, quality and communication are the areas that received the most attention from the authors. Something similar can be observed in (Carvalho et al., 2020). The authors identified, through an opinion survey with software project managers, that the areas of schedule and communication are the two areas that these professionals are most concerned about.

Specifically on the schedule, the area most emphasized in the studies, it was possible to observe that the focus in this area is given by the fact that estimating the time and the project schedule are crucial tasks and extremely influence the project results (Zhang and Jin, 2020). Regarding the phases, it was observed that the authors give greater emphasis to project planning. That is because planning is crucial to the success of a project. Tasks need to be allocated to team members, taking into account task precedence, balancing the workload of team members, and ensuring quality (Lin et al., 2014).

Regarding the type of approach, it was possible to verify that software is the most commonly used type of approach to support the software projects management. Results like this can be considered common, considering that currently several systems that automate processes are becoming more and more commonplace. Such systems facilitate the work of people in different fields of application (Uspenskiy et al., 2019). Shaikh et al. (2018) also state that project management software is widely used because it includes several useful features that facilitate the work of the team, especially the project manager, features such as task management, real-time monitoring, chatbox, notifications and alerts.

In addition to software, project management methodologies also stood out. During the analysis of the selected studies, it was possible to identify 12 studies that deal with this type of approach. Among these methodologies, one that stands out is Scrum. This methodology was used directly in four studies [PS12, PS28, PS47, PS54] and indirectly, through adaptations, in two studies [PS04, PS07]. One of the reasons for this is that Scrum teams are the most important factors in improving the performance of a project's success. Like any agile development method, Scrum follows a collaborative and guided approach to software development, reflecting the principles of the Agile Manifesto (Michael et al., 2021).

Something that caught our attention when analyzing the studies was how the approaches focus on team commitment and engagement. However, as reported in Subection 4.3.2, some approaches impose a process of constant communication and interaction, using these items as a way of evaluating the performance of team members. However, care must be taken with this particular issue, as there are approaches that can give the false impression that a collaborator is not helping or not engaged in the project. This is because this type of approach is based on and draws its conclusions from constant interactions. However, not always someone who helps in the project interacts as often, but can contribute with more technical information [PS63].

Still on the limitations, it was found that some approaches are not recommended to plan a small number of tasks, as they were developed to deal with large volumes of data. In this case, the ideal is for the organization to invest in tools that constantly monitor it, especially when dealing with a large-scale project [PS01].

Regarding the evaluations of the approaches conducted by the authors, it was possible to notice a considerable rate of studies that used the case study as a form of evaluation, a rate that corresponds to 86% of the analyzed studies. This rate allows us to verify that this type of evaluation manages to generate more satisfactory and more reliable results, as the approach is submitted to a real environment and from its use, data are extracted that allow the planning and conduct of improvements, in addition to the reduction of possible limitations that the approach may present.

Based on the results listed and described in this paper, it is possible to state that, although there was a drop in the number of studies in the years 2020 and 2021 (the latter year justified by being the year of conducting this SLR), there is an interest in part of academia and industry to develop research aimed at managing software projects, especially research that focuses on approaches that aim to facilitate this process.

6 THREATS TO VALIDITY

This section discusses potential threats to the validity of this study and actions taken to address validity issues. We used the structure proposed by Wohlin et al. (2000).

6.1 Construct Validity

To minimize the risk that the SLR would not bring the studies that answered the research questions, a test was carried out with the search string. Four studies that proved to meet the research objectives were manually selected and then it was verified if, when running the string in the bases, these same studies would return, which in fact happened.

6.2 Internal Validity

During the extraction process, studies were ranked based on our judgment. Studies that depend on the judgment of the authors can carry with them a bias that needs to be mitigated as much as possible. With that in mind, throughout the study analysis process, weekly meetings were held to discuss and reach a consensus on which studies should really be selected.

6.3 External Validity

It is possible that SLR does not return all relevant studies on approaches that support software project management. To mitigate this risk, we identified and relied on studies similar to this one so that it was not started from scratch.

6.4 Conclusion Validity

To ensure the conclusion validity of our study, we present throughout Section 4 charts and tables exposing the results generated directly from the data and we discuss the observations and explicit trends. This ensures a high degree of traceability between data and conclusions. In addition, our corpus of studies is available to other researchers. Furthermore, the SLR process was carried out with the support of a PhD professor who has extensive experience in studies of this genre, with several publications in software engineering.

7 RELATED WORKS

This section presents similar studies that are directly or indirectly related to the investigation of the present study. Despite all efforts, we did not find secondary studies that searched the literature on approaches to support software project management in agile context. However, there are secondary studies that explore other aspects of software project management.

Couto et al. (2021) conducted research consisting of a systematic literature mapping (SLM) and a survey, with the objective of identifying visualization approaches that could help in the software projects management. As a result, the authors identified 16 visualization approaches that meet the research objectives. Finally, based on the identified approaches, the authors proposed an extension of the PMBOK, adding two more processes, namely: "Plan Data Visualization" and "Implement Data Visualization".

El Bajta et al. (2018) performed an MSL to identify and classify studies on software project management approaches in the context of global software development. The authors identified, from the analysis of 84 articles, the strengths and weaknesses of each approach and analyzed their applications in the industry. The results obtained in the work indicate that the interest in software project management for the GSD has increased since 2006. It was also found that the most frequently reported methods (40%) are those used for coordination, planning and monitoring, together with with time and cost estimation techniques.

As can be seen in the studies described above, they are not focused on the same context, nor on the same objective of this SLR. However, we believe that they serve as a basis for our work as they focus on software project management or part of it, as is the case of Vieira et al. (2020) that focus only on project risk management. Therefore, this work is unique in that it explicitly deals with software project management approaches in agile context, focusing not only on one type of approach and not only on an area or phase of the project, but covering it as a whole.

8 CONCLUSION

This study described an SLR to identify approaches that support the software projects management in agile context. We selected 65 primary studies, from 2004 to 2021. Of these, 22 studies address approaches focused on project schedule management and 36 studies are focused on the planning phase.

We observed that software is the most common approach among all the others, being identified in a total of 15 studies, followed by the methodologies that appear in 12 studies. We found some positive points related to the use of these approaches, such as: reducing the effort required by the team, provide greater ease in terms of communications planning and reduce Sprint time when working with Scrum. In addition, we identified some negative points, as we can highlight: there are approaches that fully depend on an equal commitment of all team members and there are approaches that require many tasks in the initial phase of the project.

From the results, it was possible to verify how important and how project management approaches are for the development of software with more quality, in addition to being able to see some limitations that still need to be mitigated, which therefore gives opportunities to researchers in the area to develop further in the area. In this line, we plan as future works to apply a survey to software project managers who work in industry and / or academia to verify if what is being presented in the literature is being used in organizations and, finally, apply Grounded Theory to build one or more theories about the main advantages and disadvantages of using the approaches identified in the survey.

ACKNOWLEDGEMENTS

The authors would like to thank the Coordination for the Improvement of Higher Education Personnel (CAPES) in Brazil for the financial support for the granting of a scholarship for a master's degree.

REFERENCES

Banica, L., Hagiu, A., Bagescu, A., Gherghinescu, A. (2018). Designing A Website for A Recruitment Agency with Pmbok Methodology. *Scientific Bulletin-Economic Sciences*, *17*(1), 60-67.

Basili, V. R. (1992). *Software modeling and measurement: the Goal/Question/Metric paradigm.*

Carvalho, E. C., Malcher, P. R. C., & Santos, R. P. (2020). A Survey Research on the Use of Mobile Applications in Software Project Management. In *SBQS'20: 19th Brazilian Symposium on Software Quality*.

Couto J., Kroll J., Ruiz D. and Prikladnicki R. (2021). A PMBoK Extension Proposal for Data Visualization in Software Project Management.In Proceedings of the 23rd International Conference on Enterprise Information Systems - Volume 2: ICEIS, ISBN 978-989-758-509-8, pages 54-65. DOI: 10.5220/0010454 600540065.

El Bajta, M., Idri, A., Ros, J. N., Fernandez-Aleman, J. L., Carrillo de Gea, J. M., Garcia, F., & Toval, A. (2018). Software project management approaches for global software development: a systematic mapping study. Tsinghua Science and Technology, 23(6), 690–714. https://doi.org/10.26599/tst.2018.9010029.

Gomes, J., Romão, M. (2016). Improving Project Success: A Case Study Using Benefits and Project Management. *Procedia Computer Science*, *100*, 489–497.

Kitchenham, B., Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering*.

Kostalova, J., Tetrevova, L., Svedik, J. (2015). Support of Project Management Methods by Project Management Information System. *Procedia - Social and Behavioral Sciences*, *210*, 96–104.

Lin, J., Yu, H., Shen, Z., Miao, C. (2014). Studying task allocation decisions of novice agile teams with data from agile project management tools. In *ASE '14: ACM/IEEE International Conference on Automated Software Engineering*.

Michael, D., Dazki, E., Santoso, H., R. Indrajit, E. (2021). Scrum Team Ownership Maturity Analysis on Achieving Goal. In *2021 Sixth International Conference on Informatics and Computing (ICIC).*

Paasivaara, M., Lassenius, C. (2011). Scaling Scrum in a Large Distributed Project. In *2011 5th International Symposium on Empirical Software Engineering and Measurement (ESEM)*.

Parsi, N. (2021). The Next Agile Awakening: Four Agile Leaders Discuss New Possibilities in a World of Sudden Change. *PM Network, 35*(3), 36–43.

Petersen, K., Vakkalanka, S., Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, *64*, 1–18.

PMBOK Guide. (2017). Project management body of knowledge (pmbok® guide). In *Project Management Institute*.

Retnowardhani, A., Suroso, J. S. (2019). Project Management Information Systems (PMIS) for Project Management Effectiveness: Comparison of Case Studies. In *2019 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE)*. IEEE.

Shaikh, T. H., Khan, F. L., Shaikh, N. A., Shah, H. N., & Pirani, Z. (2018). Survey of Web-Based Project Management System. In *2018 International Conference on Smart Systems and Inventive Technology (ICSSIT)*.

Shrivastava, S. V., Rathod, U. (2017). A risk management framework for distributed agile projects. *Information and Software Technology*, *85*, 1–15.

Uspenskiy, M. B., Smirnov, S. V., Loginova, A. V., & Shirokova, S. V. (2019). Modelling of Complex Project Management System in the Field of Information Technologies. In *2019 III International Conference on Control in Technical Systems (CTS)*.

Varajão, J., Colomo-Palacios, R., Silva, H. (2017). ISO 21500:2012 and PMBoK 5 processes in information systems project management. *Computer Standards & Interfaces*, *50*, 216–222.

Vieira, M., C. R. Hauck, J., Matalonga, S. (2020). How Explicit Risk Management is Being Integrated Into Agile Methods: Results From a Systematic Literature Mapping. In *SBQS'20: 19th Brazilian Symposium on Software Quality*.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., Wesslén, A. (2000). *Experimentation in Software Engineering*. Springer US.

Yanow, S. K., Good, M. F. (2020). Nonessential Research in the New Normal: The Impact of COVID-19. *The American Journal of Tropical Medicine and Hygiene*, *102*(6), 1164–1165.

Zhang, S., Jin, L. (2020). Research on Software Project Schedule Management Method based on Monte Carlo Simulation. In *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*.