

Journal of Engineering Research

HYBRID SOFTWARE DEVELOPMENT, CHALLENGES AND BEST WORK PRACTICES FROM SMES: A SYSTEMATIC LITERATURE REVIEW

Alcides Alvear-Suárez

UAGM: Computer Engineering
Gurabo, Puerto Rico

Felipe Alvear-Otálora

UAGM: Computer Engineering
Gurabo, Puerto Rico

All content in this magazine is licensed under a Creative Commons Attribution License. Attribution-Non-Commercial-Non-Derivatives 4.0 International (CC BY-NC-ND 4.0).



Abstract — The Systematic Literature Review (SLR) objective was to search for evidence and investigate the state of the art of hybrid software development methodologies and answer the following research questions. RQ1: What hybrid software development methods have been reported in the literature in the context of Small and Medium Enterprises (SMEs)? RQ2: What challenges are faced by software development SMEs in the adoption, execution, and governance of projects carried out with hybrid methodologies? RQ3. What are the best practices reported in the literature to successfully adopt hybrid methods in SMEs? To justify the SRL, the “Information and Software Technology” database was used, the search string was tested: Systematic Literature Review AND Hybrid Software Development AND Small and Medium Enterprise. The search yielded 43 articles, of which 6 analyzed study topics like those proposed for this research but did not focus on the context of SMEs. Considering the limitations of any SLR, the analysis carried out allows us to understand what the evolution of the methodologies in the last decade has been and extends the frontiers of understanding to conclude that hybrid methodologies constitute a field of software engineering, which can still mature to give more scientific support to this branch of knowledge.

Keywords - Software development, Hybrid methods.

INTRODUCTION

Responding to user needs is a constant challenge for software developers; there is no doubt that we live in an era in which digital technology is transforming societies at an unprecedented speed and, at the same time, poses new and profound challenges. An aspect that undoubtedly impacts software developers, who have seen the need to opt for different methodologies or frameworks to

structure, plan and control the development of new products [1]. Currently, selecting an appropriate framework to develop software continues to be a critical aspect, a predictor for the success or failure of a company, especially if it is small or medium-sized companies [2], [3], [4]. Although the most common approaches to software development are traditional and agile, both have been evaluated for their strengths and criticized for their weaknesses [5], [6],[4].

According to [7], although there is no perfect method that responds to all needs; Currently, there is a strong trend among developers towards hybrid methods, which take up the advantages of the traditional and the agile; being a combination of the best existing practices within these methodologies. Going deeper into the aspects of hybrid methods constitutes an exciting challenge if one considers that this trend has facilitated the intersection of methodologies, domains, processes, organizational structures, techniques, technologies, and attributes. That is, it has significantly impacted the software development cycle.

Currently, many Small and Medium Enterprises (SMEs) use agile methods as they are considered more suitable for working with small teams and on projects with varying requirements that must be developed and commercialized quickly [8].. Although many companies continue to transition to agile, others have realized that pure agile methods have deficiencies due to lack of documentation, the impossibility of implementing assessment metrics, and organizational rigidity of companies. In addition, as software development is vital in all industrial sectors, it also requires basic guidelines, standards, norms, and regulations for execution. As small as the development companies may be, planning is necessary. For this reason, they have adopted some type of

traditional methodology or have resorted to the experience of the developers to collect evidence of their processes.

The main objective of this Systematic Literature Review (SLR) is to know the state of the art of the subject, collect evidence or update the existing ones on hybrid methodologies having SMEs as application domain, know the impact that hybrid methods have had on these companies; what advantages or limitations have been reported in the implementation processes of hybrid methods and, discover similar areas of interest, which allow detecting how aspects related to software development have been improved or advanced. The most important contribution of this article is to offer a detailed understanding of hybrid methodologies gather empirical cases to discern what practices have been adopted and what advantages or challenges have been reported by companies that use this work strategy. In order to achieve the proposed objective, the research questions are answered, which are the central axis for the development of an SLR, since they offer the basic methodological guidelines to identify the primary studies, extract the relevant data, synthesize and analyze them. The questions that guided the development of this SLR were:

Q1: What evidence exists in the literature on hybrid software development methodologies applicable to SMEs?

Q2: What challenges do software development SMEs face in the adoption, execution, and governance of projects carried out with hybrid methodologies?

Q3. What are the best and most successful hybrid practices reported by industrial software developers working in SMEs?

PREVIOUS WORK

A software process can be considered a set of tools, methods, and practices to make a software product [9]. Due to the

changing nature of software, researchers, and professionals in this area of knowledge focus on improving processes, making the software life cycle a permanent and dynamic research area. In the last decade, numerous investigations have been published both in the academic and industrial sectors on the best practices and methodologies for developing software; for example, the similarities and differences between agile techniques and traditional methods such as the waterfall method have been analyzed [5], [10]; developers have improved their knowledge regarding how and when a particular methodology can be applied [11], and the environments and contexts for which traditional or agile techniques are more suitable have been reviewed. Adequate [12], [13], [14].

As stated by [7], hybrid methodologies are considered as: “A hybrid software development approach is any combination of agile and traditional (plan-driven or rich) approaches that an organizational unit adopts and customizes to its own context to needs (e.g., application domain, culture, processes, project, organizational structure, techniques, technologies, etc.)” From this perspective, what began as an integration between the traditional and the agile, today is assumed as an adaptive mix of all the processes that have to do with software development and that guarantee that a group of workers in an agile, coordinated, and productive manner, achieving the objectives of a project in the shortest possible time and consuming the least amount of resources.

Due to the integration line between traditional and agile methods, some publications focus on the administrative process, such as [15]. They designed a catalog with the best planning practices for developing projects software. The proposed emphasis considered hybrid solutions, abstract models, and a set of traditional and

agile best practices. The researchers' idea was to reduce unproductive tasks and provide tools to decrease execution time and guarantee product quality. However, according to what was established by [16], As of 2015, there are still no practical and quantitative studies on the use of hybrid methods; the only thing that is recognized is that currently, the two basic approaches are used in combination.

Regarding the use of hybrid methodologies in SMEs, the study carried out by [17] indicates how these companies have changed over time, in what areas of knowledge they have focused, and what empirical evidence exists on their development. As is known, SMEs are companies that work with low resources; they are made up of small work teams; they offer innovative software products but are not backed by a prescriptive methodology; they do not have specific clients; there is no documentation, and the software is only validated after its release. Furthermore, these companies by nature repudiate the notion of repeatable and controlled processes, betting instead to take advantage of unpredictable, reactive, and low-precision engineering [18]. Therefore, attempts to adopt structured work schemes, whether traditional or agile, cause profound rejection. This lack of rigor makes it challenging to transfer experiences, affects the trust and credibility of their processes, and may indicate the enormous rate of business failures reported in the literature.

RESEARCH METHOD

In the field of SI, numerous protocols have been proposed to plan, develop, and publish the results obtained in SLRs. To carry out the review on hybrid methodologies applied to SMEs, the guidelines proposed by [19], [20], [21] were followed. According to [13] SLRs are useful for inquiring about state of the art, exploring best practices, and updating or collecting better evidence on a particular

topic. The template proposed by [20] was used to perform the SLR, shown in Figure 1. The procedure included: a justification for the study; the application of the pilot study (to refine the search criteria, select the keywords, apply the PICOC method, construct the research questions and design the search strings); selection of inclusion and exclusion criteria; the mechanisms for extracting information and the application of techniques to summarize the results.

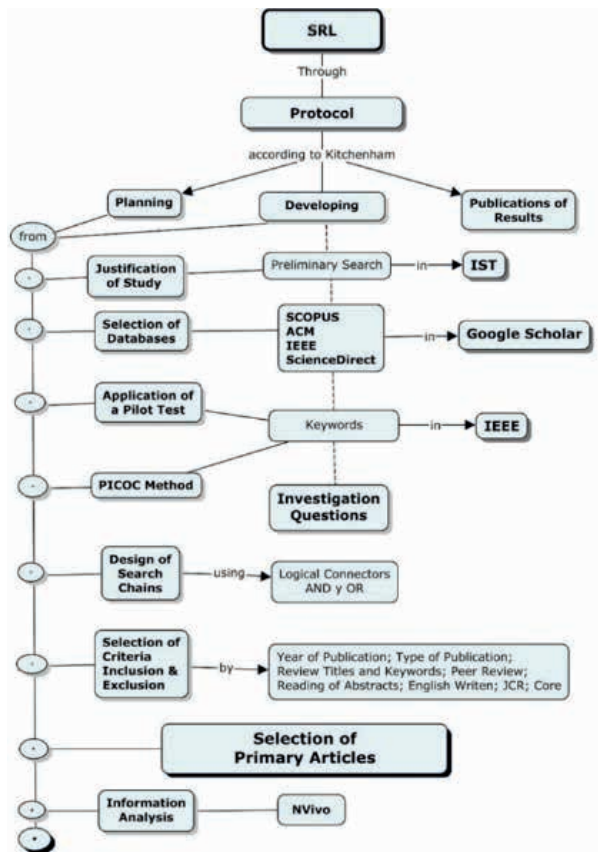


Figure 1. Protocol for the development of the SLR

A. STUDY JUSTIFICATION

As seen in section II, several authors have addressed issues related to the implementation of hybrid development methodologies; however, there is still no solid evidence that SMEs are the domain of application; therefore, the impact that these methods have had on these companies is unknown, what advantages

or limitations have been reported and what practices can be successful in this context. As previously argued, the SMEs dedicated to the software industry; typically lack economic resources and accurate software development models, presenting problems to build or reuse their models. Therefore, flexible models are required that adapt efficiently to the needs of the developers. However, no matter how small the development companies may be, planning is necessary.

To justify the SRL, the database of the magazine “Information and Software Technology” was used, the search string was tested: (Systematic Literature Review) AND (Hybrid Software Development) AND (Small and Medium Enterprise). The search yielded a total of 43 articles, of which 6 analyzed study topics like those proposed for this research but did not focus on the context of SMEs.

According to what is established by [22] in an SLR, it is important to carry out a Pilot Study or preliminary information search, having as dependent variable the terms Hybrid software development and as independent variables, the terms Traditional methodologies, Agile methods, and SMEs – Startup. As shown in Table 1, in this phase of the procedure, the Springer database was used, basic search strings were built, the titles of 122 articles were reviewed, and of these, 37 publications were selected to review the keywords and know related terms related to the topic. One hundred sixty-five keywords were obtained; these were organized in an Excel template. Repeated terms and those unrelated to the topic (Data no show) were eliminated.

B. PILOT STUDY: PILOT STUDY SEARCH CRITERIA

As the articles that publish experiments in IS are not very rigorous in the use of terminology, it is difficult to locate the relevant experiments associated with the variables that

are to be tested in an SRL, hence the need for the Pilot Study to help recognize related terms that the authors use interchangeably to refer to the same processes.

Keywords from the pilot study were distributed in the PICOC format; whose name is derived from the acronym in Population, Intervention, Comparison, Outcome, and Context; which, according to [23], improves the specificity and conceptual clarity of the problem and helps to design research questions more efficiently.

| Dependent Variable | Independent Variable | Springer | Articles for Keyword Review |
|------------------------------------|--|----------|-----------------------------|
| <i>Hybrid software development</i> | <i>Traditional methodology-methods</i> | 30 | 12 |
| | <i>Agile methods</i> | 71 | 17 |
| | <i>SMEs - Startup</i> | 23 | 8 |
| <i>Total</i> | | 122 | 37 |

Table 1. Pilot study search criteria

As the articles that publish experiments in IS are not very rigorous in their use of terminology, it is challenging to locate the relevant experiments associated with the variables that are to be tested in an SRL, hence the need for the Pilot Study to help recognize related terms that the authors use interchangeably to refer to the same processes. Table 2 shows the five original terms and related terms derived from the Pilot Study.

| Original Term | Terms derived from the Pilot Study |
|-----------------------------|---|
| Hybrid software development | Hybrid development approach, Hybrid development method, Hybrid development system, Hybrid Software Development, Hybrid Software Development Method, Hybrid, Hybrid methods, Hybrid organization. |
| Traditional methodology | Traditional methodologies, Formal development, Waterfall, Waterfall model, Conventional software development methods, Traditional development approach |
| Agile methodology | Agile, Agile methods, Agile Modeling, Agile models Agile practices, Agile software, Agile software development, Agile software engineering, Agility, Scrum, Extreme Programming, Feature Driven Development(FDD), Lean software development, Test-driven development (TDD), Feature-driven development (FDD), eXtreme Programming (XP), Agile data method, Lean Development, Microsoft Solutions Framework (MSF) for Agile, Adaptive Software Development (ASD), Crystal, Behavior-driven development (BDD), Six Sigma, Dynamic systems development method |
| SMEs | Startup Company, Small Organizations, Small Project, Small and medium size enterprise, Product innovation enterprise, Small medium bussines |
| Experiment | Experimental study, experimental comparison, experimental analysis, experimental evidence, y experimental setting |

Table 2. Terms related to the research variables from the application of the Pilot Study.

The keywords of the Pilot Study were distributed in the PICOC format; whose name is derived from the acronym in English: Population (population), Intervention (intervention), Comparison, Outcome, and Context; which, according to [23], improves the specificity and conceptual clarity of the problem and helps to design the research questions more efficiently.

The research questions are of the utmost importance for developing an SLR as they offer the basic methodological guidelines to identify the primary studies, extract the relevant data, and synthesize and analyze them. The questions that guided the development of this SLR were:

RQ1: What evidence exists in the literature on hybrid software development

methodologies applicable to SMEs?

RQ2: What challenges do software developer SMEs face in the adoption, execution, and governance of projects carried out with hybrid methodologies?

RQ3. What are the hybrid best practices and the most successful ones reported by industrial software developers working in SMEs?

Keywords from the pilot study were also used to design and test different search strings, considering that the general words help retrieve less precise articles. In contrast, the more specific ones allow refining the search processes. Under this perspective and following the model the chains were structured, as shown in Figure 2, where the white circle indicates an “OR” and the black ones indicate an “AND”, the words separated by commas (,) indicate an “OR.”

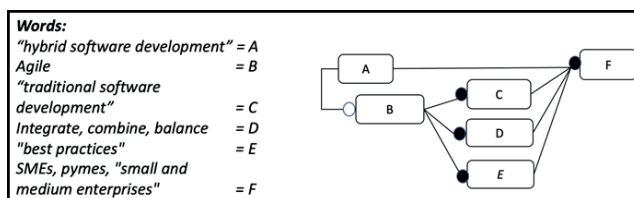


Figure 2. Search string structure

C. INFORMATION SOURCES AND INCLUSION/EXCLUSION CRITERIA

A preliminary search in Google Scholar detected that the articles related to the variables of interest were published mainly in journals and conference proceedings, so the digital databases were selected, considering criteria such as: coverage, ease search and delivery result; that is, if the article could be consulted completely or only the abstract. Since there are no databases specialized in experiments in the SI area, 5 databases were used for this SLR: Google Scholar, IEEE, ScienceDirect, SpringerLink and ACM.

According to what was established by [24], [25] and by [10], the first publications after the

agile manifesto were the basis for the adoption of these new methodologies and to understand the characteristics of software development. From this perspective. Therefore, hybrid methodologies are considered a postadoption issue of agile methodologies and therefore, an exclusion criterion for this SLR was articles published before 2008. Documents that were not written in English were also excluded, that had not been peer-reviewed and those that were considered duplicate or similar investigations. After reading the titles and/or abstracts, other publications were excluded. Finally, the JCR and CORE impact factor that evaluates the quality of the conferences was considered, the remaining publications were considered primary articles. Subsequently, an exhaustive reading was carried out to corroborate the relationship of the primary articles with the research questions and with the objectives of the SLR.

RESULTS

Like [26] states, an SLR is a rigorous methodology whose objective is to compile the existing evidence on a research topic and provide probable solutions within the field of software engineering to problems and contexts. The protocol used to carry out the SLR is shown in figure 1, in which the recommendations made by [26], [27], and [28] were included; the implementation of this protocol allowed the selection of 68 primary articles.

To justify the relevance of the study, the IST database was used, and the search string was tested: “Systematic Review” AND “Software Development” AND “Waterfall” AND “Hybrid Methods” AND “Agile Practices” AND “SMEs.” This chain returned 45 articles, of which 6 were SLR. Within this subset, only the investigations of [29], [19], and [20] were based on agile developments, and the investigations [21], [22], and [23], [25]

focused on aspects related to engineering, of the software. It was determined that none of these publications integrated the variables proposed for this SLR.

The search strings were iteratively and incrementally tested on the four databases using the Boolean OR and AND connectors. The process started from associating the dependent and independent variables, yielding 368,693 articles. Refining the search by year and type of publication reduced the number to 517, as shown in Table 2.

| Query | Scopus | ACM | IEEE | Science Direct | Total |
|--------------|------------|-----------|-----------|----------------|------------|
| Q1 | 54 | 25 | 22 | 91 | 193 |
| Q2 | 44 | 25 | 20 | 23 | 114 |
| Q3 | 5 | 37 | 4 | 88 | 137 |
| Q4 | 2 | 1 | 1 | 65 | 73 |
| Total | 105 | 88 | 47 | 267 | 517 |

Table 3. Results obtained when testing the search strings in the different databases.

The inclusion and exclusion criteria were then applied (Excluding articles by reading the abstract), and the 68 primary studies were finally obtained.

A. RESULTS OF THE CATEGORIZATION PROCESS

As [30], to develop knowledge within software engineering (SE), it is necessary to accumulate empirical evidence that helps summarize, integrate, combine, and compare the results of different studies on a specific topic or research question research. Although the primary studies are heterogeneous, in the present SLR, the categorization allowed applying some statistics to determine, as of 2008, the annual frequency of publications on the subject, the most used databases, and the most frequent types of study within the field research.

It was found that hybrid software development methodologies constitute an

emerging and growing topic within the IS field, with the highest frequency of publication being observed in the 2013-2015 interval; the results are shown in figure 3.

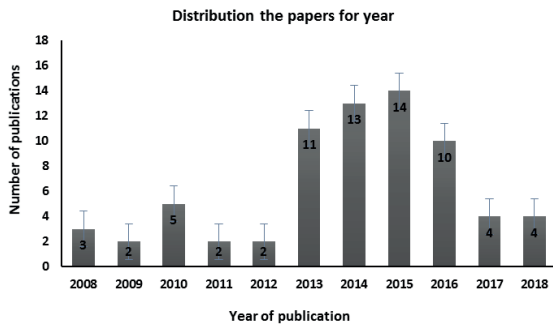


Figure 3. Number of articles per year of search

According to [11] in the first decade of the 21st century, many methods, tools, techniques, and practices were developed to improve software manufacturing. However, time and experience have been needed to take on the new constructs and make them fit reality and the needs of both developers and users. To date, many companies have adopted agile proposals while others continue in the process of critical evaluation, trying to detect the benefits and difficulties of this work scheme.

It was also observed that 100% of the publications used in this SLR were searched with the help of electronic databases, these results are seen in figure 4. It was also observed that 100% of the publications used in this SLR were searched with the help of electronic databases, these results are seen in figure 4.

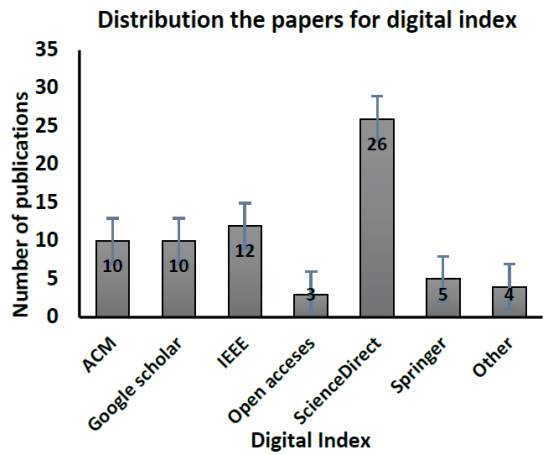


Figure 4. Number of articles per digital indexes

This indicates a strong tendency of researchers to publish their findings in recognized and prestigious databases in the scientific-technical field; therefore, it is not surprising that ScienceDirect and IEEE are the most widely used dissemination sources in SI.

The objective of an SLR is to analyze the relevant information on a research topic to discover what is being done, which topics are the most addressed, and what is the trend followed by researchers in a particular field of knowledge. Hence the need to seek reliable and up-to-date information; to help software developers make better decisions in an area as active and controversial as IS. In this regard and according to what is established by [25], [31], not all scientific documents, nor all journals have the same value, nor do they contribute in the same way to the investigative process and, although the categorization tries to offer a standardized metric, in any case the procedure retains a high degree of subjectivity.

In this SLR, the metrics provided by JCR and CORE were used as criteria to analyze the scientific quality of the primary articles; In this regard, it was determined that 82.9% of the publications were indexed and were subjected to an exhaustive peer review process and, of these; As can be seen in

figure 5, 41.2% (28 articles) were Journals, and the other 41.2% were publications made at specialized conferences. The review in the JCR determined that of this amount, 75.9% of the articles were published in high-impact journals, quartile 1 (Q1), and the remaining 24.1% were located in quartile 2 (Q2). The conference publications were also indexed, and the CORE determined that 27.6% were classified as type A* conference, 51.7% as type A, and the remaining 20.7% as type B.

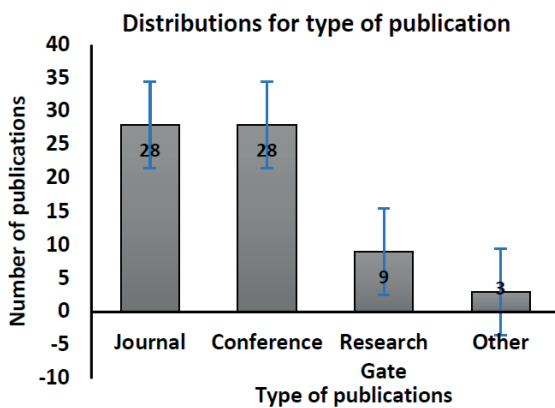


Figure 5. Number of articles per type of publication

In addition, to measure the rigor and relevance of the publications for the SLR, a Likert scale was used, and the results of this process are observed in Figure 6. In this regard, it was determined that 81.4% of the articles (mean scale and high) showed a rigor consistent with the quality required for a journal or a specialized conference; however, the level of contribution of these publications was only 47.1%. In other words, the quality of a publication is only sometimes directly related to the contribution that it can offer to solve a research problem. It was also determined that the least rigorous publications are the most relevant; this is possibly due to the fact that hybrid methodologies, as said before, are an emerging issue within the SE, and therefore, the researchers are still in the experimental

phase of the process and are beginning to publicize their contributions in forums or specialized conferences. In the coming years, the number of journals is expected to be much higher than the number observed up to the present.

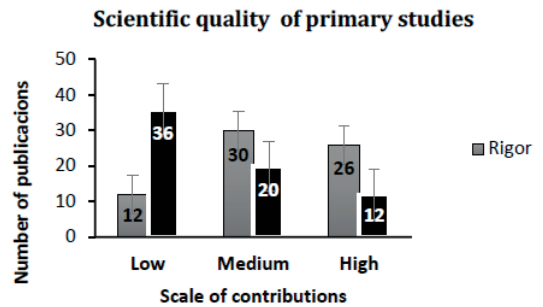


Figure 6. Contribution of primary studies to the SLR

After reading the 68 primary articles, the degree of contribution of each publication was determined in relation to the objectives and research questions. An extended Likert scale was used, and a score was calculated from the dependent and independent variables. The results of this part of the process are summarized in Figure 7. As can be seen, 52% of the articles had little or low contribution since they addressed only one or two of the variables proposed for this SLR or did not explicitly propose methodological integration processes for software development. Only 11% of the articles consulted dealt with hybrid work frameworks, some applicable in the context of SMEs.

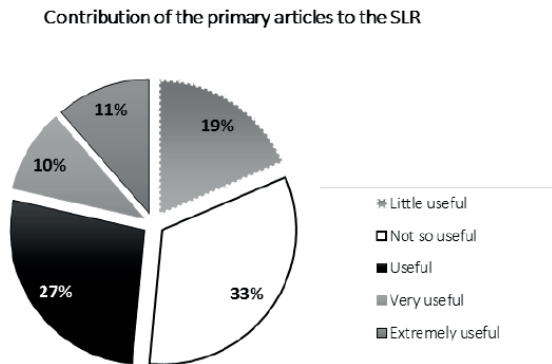


Figure 7. Degree of contribution of the primary articles to the SLR

B. QUALITATIVE ANALYSIS

An analysis within each topic determined that there are no recent publications on pure traditional methodologies, and those cited in this SLR are articles where the use or characteristics of said methodologies are tangentially discussed and compared with more suitable forms and/or agile ways of manufacturing a software product. In other publications, the characteristics of the workgroup are analyzed, and it is determined that the skills of an agile team are different from those required to work under traditional methodological schemes.

Traditional methodologies are also mentioned in experimental case studies, where it has been proposed to include usability techniques in companies that use the waterfall development model or that opt for some type of hybrid between traditional and agile. Other articles discuss the need to migrate from traditional to agile in the context of cloud computing, open-source software, and distributed software. Finally, one of the articles determines how to transform startups with traditional work schemes into agile companies that better adapt to changing market requirements and specific customer needs.

Regarding agile methodologies, 29 articles

were detected on the subject, distributed as follows: seven SLRs, two tertiary studies, a retrospective publication that addresses agile development from its origins to 2012, publications that propose specific agile frameworks for fill some gaps observed in the adoption processes, lessons learned with the implementation of case studies, articles with basic conceptual foundations to guide the reader in the selection process of an agile methodology, in particular, articles focused on the importance of including the factor human as a key to the success of agile projects and finally, publications that are based on more recent action topics such as cloud computing.

Currently, the methodologies are assumed as a set of practices applied differently depending on the need and the type of projects. According to what was observed by [7] it is determined that neither the pure traditional approach nor the pure agile approach is used in practice by developers.

Regarding usability, seven publications were found: an SLR that summarized several selected works, a case study, two frameworks, two publications with conceptual foundations, and one related to open software. Only 10.3% of the publications of this SLR discuss aspects related to usability, indicating that the subject continues to be little analyzed and, as some authors point out, the gap between HCI and SE remains.

The last search topic in this SLR was software development in SMEs apparently only five articles were detected by the search strings; however, two studies of SMEs, an SLR, na article on the conceptual foundations of software development in startups, and a framework to facilitate decision-making and achieve business agility were found. Few publications discuss how software is developed in startups, detecting 54 practices and 15 primary areas of knowledge, some covered by SWEBOK.

DISCUSSION OF RESEARCH QUESTIONS

A. RQ1 WHAT TYPES OF METHODOLOGIES FOR SOFTWARE DEVELOPMENT HAVE BEEN THE MOST USED IN THE LAST DECADE?

To contextualize RQ1, we will start from the contributions established by [32]; in this retrospective analysis, after a decade of implementation of agile methodologies, the authors indicate that even though some agile methodologies had begun to be structured in the 90s, only after the Agile Manifesto, all of them (XP, Scrum, Lean, FDD or the different Crystal methodologies) tried to address the fundamental principles of the Manifesto, trying to bring coherence to the Agile discourse. That is to say; the first years are considered a stage of adoption of agile principles based on: collaborative development, which gave importance to people and therefore to the work team; work optimization, eliminating tasks that did not add value or functionality to the process; the inclusion of the client as an active element in the development process and the acceptance that uncertainty was an essential part of software development.

An SLR with empirical studies published up to 2005 [33] found that most agile publications were based on XP development and very few on Scrum. In addition, a trend of agile methodologies in small projects and newly created companies (startups or SMEs) was evidenced; hence, the authors expressed the need to apply agile methodologies in larger companies and more significant projects complexity.

Between 2008 and 2015, 19 of the 29 selected articles were written on Agile Software Development (ASD); of them, three are SLR and a Mapping Study. In these publications, the authors selected 174 primary studies extracted from an obtainable larger universe if one considers that the inclusion and exclusion

criteria only allow the selection of a maximum of 5% of the studies related to the topic.

Although there are still publications related to adoption stages or the selection of a particular agile methodology, enough time has also passed to dispassionately evaluate the process and propose useful frameworks to implement the benefits or overcome the difficulties of ASD; In this context, there are also the case studies, and the lessons learned [6] that, without a doubt, focus on the coverage of the process and make it advance significantly.

From 2015 to the present, four SLRs and one Mapping Study have reviewed the progress of different ASD topics; in 2017, in their tertiary study, [29] reviewed the research areas and determined that the topics of analysis increased in parallel with the topics related to SE. They detected ten research areas within ASD: adoption, methods, practices, human and social aspects, CMMI, usability, global software engineering (GSE), organizational agility, embedded systems, and online software products. The authors also noted significant progress in administrative management (SPM) by including usability, organizational maturity degrees (CMMI), and GSE within the agile field. In this context, Scrum emerges as the most widely used agile methodology at an industrial level. It is not surprising that several articles and an SLR focus on discussing the importance of including the human factor as a key to the success of agile projects[34].

Finally, it can be concluded that in the decade 2008 to 2021, different kinds of agile methodologies have coexisted and, rather than adopting a particular way of working, developers have used the so-called “agile practices,” which, as [35] indicates, do not have a standard definition in the literature. However, they result from the fusion of different practices used by ASD, which the authors call “universal.” In practice, it

can be said that the agile methodological universe has been contextualized by software manufacturers who, according to their needs, opt for a style or work practice without marrying a particular methodology. A specific approach between SE and HCI is also observed; since it has begun to be recognized that making software is a human activity, in which a workgroup is required, and all the parties involved contribute with their inputs to manufacture a better-quality product.

B. RQ2 AT WHAT HISTORICAL MOMENT AND UNDER WHAT CIRCUMSTANCES DO HYBRID METHODOLOGIES EMERGE?

Agile methodologies arise in a historical context in which traditional methodologies were inoperative, forcing developers to reflect on which parts of the process could be ignored or manufactured differently to save time, avoid costs, and guarantee quality. Agile began to be a reality at the end of the nineties; its principles were organized in the Agile Manifesto of 2002. Since then, several years have passed, trying first to understand and then apply the different orientations to specific work contexts. At times the ASD has focused more on the administrative area (Scrum or Kanban), and other times it has focused better on the software design or construction process (XP). Nevertheless, as [15] indicates, life is not made in black and white: and just as many traditional processes did not fully respond to the historical moment, something similar happened with agile methodologies.

As early as 2002, [32] found that software developers and project managers had difficulty evaluating the suitability of agile processes in their work environments. The authors affirm that in the “Time of the Internet,” organizations seek competitive advantages through the timely offer of Internet-based services, forcing developers to produce or

improve their applications in record time. In this context, agile approaches offer technical and management processes that try to adapt to change, generating software versions that meet the aggressive demand of the consumer society.

In the beginning, agile processes were designed to quickly support the production of working code (as in XP), and the concepts of iterative (as a functional deliverable) and incremental (with improved functionality) reached great value to the detriment of other phases. In the same publication [36], the Agile Manifesto principles were analyzed, and the practical limitations that existed at the time for its fulfillment were determined. They concluded that some aspects of a software development project could benefit from an agile approach. In contrast, others could do better than a predictive model essential aspects such as documentation. Important aspect for large companies with complex developments that require good documentation or models to support the evolution of their software.

In 2005 and still, in 2017 [4], they determined that the selection of the work methodology was a critical point within the software development process and indicated that although agile methodologies were prevalent in recent years, only 50% of the projects implemented with this way of working were considered successful; that is, selecting a method implies taking a drastic risk, which is affected by variables such as the nature of the project, the skills of the development team, the limitations of the project, the participation of the client and the organizational culture. The risk that not all work teams want to assume.

Indeed, many companies still do not have the necessary training and organizational maturity to migrate to agile. In 2010, a survey conducted by [37] determined that approximately 39% of 1,023 professionals in the IT sector claimed to follow an agile

method. However, the reality is that the approach of many organizations is limited by their organizational culture and the government requirements to which they must adjust [26]. Therefore, as [27] [28] states, the adoption of agility requires the integration of agile and nonagile development elements. Greater agility at the level of people (work teams), processes, products, and tools, and better management through traditional administrative schemes.

On the other hand, the defenders of agile practices themselves affirm that each practice is helpful on its own. However, several practices provided better value, promoting the need to opt for hybrid schemes within ASD. Indeed, a survey whose results are analyzed by [29] shows that in at least 20% of XP projects, developers use three or more agile practices simultaneously. In addition, it has also been reported in the literature that work teams evolve as they work together. Hence, the experience is a significant factor in evaluating the gains or limitations of a particular way of working. Similar to what was reported by [30], the experience of the developers was crucial for the transition from traditional to agile and will also be so short when new ways of developing software are needed.

As is recognized, the IT sector handles a large and complex compendium of critical information, which requires defined design and development processes. To meet these requirements, traditional plan-driven approaches are necessary, but at the same time, it is necessary to increase coverage through agile management of all processes. In addition, as [31] states, agile methods were designed to be executed in projects with small work teams; however, larger companies, attracted by the possibility of bringing their products to market more quickly, saw a new way of working in agile.

It is concluded that in this decade, neither

the pure traditional approach nor the pure agile approach is used in practice by developers, who in real life opt for a much more ad-hoc process, where work methodologies become tools that can be used interchangeably according to your needs. Apparently, agile methods speed up software development, improve quality, and increase customer satisfaction. However, as [23] states, in practice, it is known that the world is not entirely “streamlined” since there are environments in which agile methods are not fully applicable or cannot show their strengths. It is chosen to identify different combinations between the traditional and the agile [26].

C. RQ 3 WHAT TYPES OF HYBRID METHODS ARE MOST EFFICIENT FOR SMES?

In this study, five publications were detected that related the SMEs with the variables of interest [34], [35],[15], [36], some of these were included in the mapping prepared by Berg [17]. For this work, the researchers selected 74 studies and answered three research questions that led them to detect how startup software developers have changed over time, what areas of knowledge they have focused on, and what empirical evidence exists on their development. The article provides evidence on the particularities of start-ups: typically, innovative organizations, shaped by the market, with little or no operating history; Established to respond to the needs of the consumer society, they face high volatility in technologies and marketing processes. As [15] states, the startup environment is dynamic, unpredictable, and even chaotic, which forces entrepreneurs to act quickly, since 60% of these companies only survive 5 years.

Software developer startups present a combination of characteristics that become challenges when implementing na application [4]. As is typically known, SMEs are companies

that work with low resources; they are made up of small work teams; they offer innovative software products but are not backed by a prescriptive methodology; they do not have specific clients; there is no documentation, and the software is only validated after its release. Under these circumstances its high self-destructive power is not uncommon.

As [36] indicates, the lack of rigor of these companies makes the transfer of experiences difficult or even dangerous, since, for example, the choice of a good methodology can be useful only within its context; In addition, the lack of documentation seriously affects the trust and credibility of your processes. Therefore, like what was concluded by [18] Existing studies, which address software engineering in startups, are insufficient to analyze all aspects of engineering and do not create a solid body of knowledge.

An attempt to remedy the shortcomings of startups was proposed by [17]. According to the authors, the lean-startup scheme could be helpful for these companies since they speed up business, turn ideas into products, measure customer satisfaction, are based on the BML (Build-Measure-Learn) process, they build a minimum usable product (MVP), and eliminate any waste from the process. Under these circumstances, the construction of a software product is like the application of the scientific method within the research field; that is, during the process, questions are raised that are possibly solved with the development of a software product, hypotheses are built from the creation of an MVP, the product is preliminarily validated, and the customer's inputs on the benefits or deficiencies of the system are collected. Customer feedback is used to learn and select efforts that create value, eliminating unnecessary efforts from the system. In a complementary way, the authors have detected four phases that startups go through initiation phase or moment from

when the idea is conceived until the first sale is made; product stabilization phase that implies maintaining the business long enough to obtain a new client: a growth phase, established by the size of the market, and phase of organizational maturity that enables the company to work on new ventures. Ignoring these practices or ignoring the moments that a small company goes through may be indicative of the tremendous rate of failures reported in the literature.

To know which hybrid methods are most useful for the context of SMEs, it is enough to remember that agile methods such as XP and Scrum were initially designed for projects with small teams [30]. Without falling into the false dichotomy that what is traditional is bad or what is agile is good, as indicates [18], building software in a small company means "choosing a coherent set of policies, organizational structures, technologies, procedures and artifacts that allow conceiving, develop, implement, and maintain a software product. Unlike plandriven software development, agile software addresses uncertainty and quickly adapts to change; in the process, developers learn while creating perceived or requested customer value.

To date, attempts have been made to introduce various models to promote software development in startups; however, these companies are creative and flexible by nature and do not want to introduce bureaucratic measures into their processes. In addition, as indicated, these are companies that work with limited resources and are not willing to invest in establishing rigorous and inoperative processes. Therefore, the attempts to adopt structured work schemes, whether traditional or agile, have caused a profound rejection by startups that by nature repudiate the notion of repeatable and controlled processes, betting instead on taking advantage of unpredictable, reactive engineering and low precision.

CONCLUSIONS

and structure of this article.

A statistical review of this research shows that the number of studies on software development methodologies has increased significantly from a quantitative and qualitative point of view. It is observed that 100% of the publications used in this SLR were selected from electronic databases such as ScienceDirect, IEEE, or ACM. Discussion forums and specialized conferences have also increased, indicating that this topic arouses much interest within the SE community.

In this literature review, it was found that in the last decade, software engineering has been powerfully influenced by the development of the Internet, a technology that in turn has permeated people's lives has changed the way of thinking and doing things stuff. It is recognized that manufacturing software is a difficult task, and as with most technological processes, a structured scheme is required to guide the developer and facilitate the achievement of the stipulated goals. To develop software, there are different process models, from completely traditional to completely agile. However, in this SLR, it was glimpsed that none of them fully responded to the needs of developers, presenting shortcomings that the diferente investigations propose to solve through case studies, lessons learned, or specific frameworks.

Faced with an increasing number of methodological possibilities, it is logical to think that developers select those that best fit the problem they need to solve; Using their experience and knowledge acquired over time, many do not even ask themselves if their work scheme is traditional or agile, they simply choose their tools intuitively and make them work for a particular purpose.

ACKNOWLEDGMENT

To Dr. Graciela E. Quintero from UAGM, for her valuable collaboration in the writing

REFERENCES

- [1] Roger. S. Pressman, *Software Engineering: A Practitioner's Approach*. Pressman & Associates, Inc, 2010.
- [2] J. Molina-Ríos and N. Pedreira-Souto, "Comparison of development methodologies in web applications," *Information and Software Technology*, vol. 119, p. 106238, 2020, doi: <https://doi.org/10.1016/j.infsof.2019.106238>.
- [3] D. R. Luna *et al.*, "Requirement Specification in Government {IT} Procurement," *Journal of Systems and Software*, vol. 86, no. 4, pp. 166–181, 2013, doi: <http://dx.doi.org/10.1016/j.eswa.2013.09.034>.
- [4] M. Alqudah and A. Abdulsalam, "Implementing computer-aided language learning tool using hybrid agile method: A case study," *Proceedings - 2013 International Conference on Informatics and Creative Multimedia, ICICM 2013*, pp. 174–180, 2013, doi: 10.1109/ICICM.2013.37.
- [5] Vijayarathy, L. R. and Butler, C. W., "Choice of software development methodologies: Do organizational, project, and team characteristics matter?," 2015, pp. 86–94.
- [6] A. A. and Ilnago, V., "A Comparative Analysis of Traditional," in *Software Engineering and Agile Software Development*, Wailea, Maui, HI, 2013, pp. 4751–4760.
- [7] M. Kuhrmann *et al.*, "Hybrid Software and System Development in Practice: Waterfall, Scrum, and Beyond," in *Proceedings of the 2017 International Conference on Software and System Process*, in ICSSP 2017. New York, NY, USA: ACM, 2017, pp. 30–39. doi: 10.1145/3084100.3084104.
- [8] A. Schmid, G. Jacobs, M. Löwer, T. Katzwinkel, W. Schmidt, and J. Siebrecht, "A Functional Platform Strategy for Integrated Machine Tools," *Procedia CIRP*, vol. 60, pp. 296–301, 2017, doi: <https://doi.org/10.1016/j.procir.2017.02.042>.
- [9] M. Humphrey *et al.*, "How the Repository Driven Test Automation (RDTA) Will Make Test Automation More Efficient, Easier & Maintainable," in *Companion Proceedings of the 36th International Conference on Software Engineering*, in ICSE Companion 2014, vol. 16. New York, NY, USA: ACM, Mar. 2010, pp. 5–6. doi: 10.1145/2399016.2399037.
- [10] George. Nerur, Sridhar.; Mahapatra, RadhaKanta.; Mangalaraj, "Challenges of migrating to agile methodologies," vol. 48, no. 5, pp. 72–78, 2005.
- [11] T. Dyb and T. Dingsøy, "Empirical studies of agile software development: A systematic review," *Information and Software Technology*, pp. 833–859, 2008.
- [12] B. Boehm, "Get Ready for Agile Methods, with Care," *Computer*, vol. 35, pp. 64–69, 2002, doi: 10.1109/2.976920.
- [13] B. Kitchenham *et al.*, "Systematic {Literature} {Reviews} in {Software} {Engineering} - {A} {Tertiary} {Study}," *Inf. Softw. Technol.*, vol. 52, no. 8, pp. 792–805, 2010, doi: 10.1016/j.infsof.2010.03.006.
- [14] A. C. J. Highsmith, "Agile Software Development: The People Factor," 2004.
- [15] G. P. G.-H. Bustamante Andrés Felipe, Jesús Andrés Hincapié, "Structure of a Multi-model Catalog for Software Projects Management Including Agile and Traditional Practices," in *Trends and Applications in Software Engineering*, Springer, Ed., 2015, pp. 87–97.
- [16] Theocharis Georgios, Kuhrmann Marco, Münch Jürgen, Diebold Philipp, "Is Water-Scrum-Fall Reality? On the Use of Agile and Traditional Development Practices," *Product-Focused Software Process Improvement*, 2015.
- [17] V. Berg, J. Birkeland, A. Nguyen-Duc, I. O. Pappas, and L. Jaccheri, "Software startup engineering: A systematic mapping study," *Journal of Systems and Software*, vol. 144, pp. 255–274, 2018, doi: <https://doi.org/10.1016/j.jss.2018.06.043>.
- [18] N. Paternoster, C. Giardino, M. Unterkalmsteiner, T. Gorschek, and P. Abrahamsson, "Software development in startup companies: A systematic mapping study," *Information and Software Technology*, vol. 56, no. 10, pp.1200–1218, 2014, doi: 10.1016/j.infsof.2014.04.014.
- [19] K. Dikert, M. Paasivaara, and C. Lassenius, "Challenges and success factors for large-scale agile transformations: A systematic literature review," *Journal of Systems and Software*, vol. 119, pp. 87–108, 2016, doi: <https://doi.org/10.1016/j.jss.2016.06.013>.

- [20] H. Chen and K. Damevski, "A Teaching Model for Development of Sensor-driven Mobile Applications," in *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*, in ITiCSE '14. New York, NY, USA: ACM, 2014, pp. 147–152. doi: 10.1145/2591708.2591719.
- [21] M. R. de A. Souza, L. Veadó, R. T. Moreira, E. Figueiredo, and H. Costa, "A systematic mapping study on gamerelevant methods for software engineering education," *Information and Software Technology*, p., 2017, doi: <https://doi.org/10.1016/j.infsof.2017.09.014>.
- [22] J. Pernstål, R. Feldt, and T. Gorschek, "The lean gap: A review of lean approaches to large-scale software systems development," *Journal of Systems and Software*, vol. 86, no. 11, pp. 2797–2821, Nov. 2013, doi: <https://doi.org/10.1016/j.jss.2013.06.035>.
- [23] S. Ghobadi, "What drives knowledge sharing in software development teams: A literature review and classification framework," *Information & Management*, vol. 52, no. 1, pp. 82–97, Jan. 2015, doi: <https://doi.org/10.1016/j.im.2014.10.008>.
- [24] B. Boehm, "A View of 20th and 21st Century Software Engineering," ICSE, 2006.
- [25] V. P. Guerrero-Bote and F. Moya-Anegón, "A further step forward in measuring journals' scientific prestige: The SJR2 indicator," *Journal of Informetrics*, vol. 6, no. 4, pp. 674–688, 2012, doi: <https://doi.org/10.1016/j.joi.2012.07.001>.
- [26] M. Kuhrmann and G. Kalus, "Criteria for software process tailoring: a systematic review," *Proceedings of the 2013 International Conference on Software and System Process - ICSSP 2013*, p. 171, 2013, doi: 10.1145/2486046.2486078.
- [27] B. A. Kitchenham, *Procedures for performing systematic reviews*. Keele University TR/SE-0401, 2004.
- [28] A. Beelmann, M. Petticrew, and H. Roberts, "Systematic reviews in the social sciences. A practical guide," *European Psychologist*, vol. 11, pp. 244–245, 2006.
- [29] R. Hoda, N. Salleh, J. Grundy, and H. M. Tee, "Systematic literature reviews in agile software development: A tertiary study," *Information and Software Technology*, 2017.
- [30] D. S. Cruzes and T. Dyb, "Research synthesis in software engineering: A tertiary study," *Information and Software Technology*, vol. 53, no. 5, pp. 440–455, 2011, doi: 10.1016/j.infsof.2011.01.004.
- [31] D. Avison, M. Cole, and Guy Fitzgerald, "Reflections on Teaching Information Systems Analysis and Design: From Then to Now!," *Journal of Information Systems Education*, vol. 17, no. 3, pp. 253–256, 2006.
- [32] D. Turk, R. France, B. Rumpe, and G. Georg, "Model-Driven Approaches to Software Development," *Springer*, vol. 2426, pp. 229–230, 2002.
- [33] J. Erickson, K. Lyytinen, and K. Siau, "Agile Modeling, Agile Software Development, and Extreme Programming: The State of Research," *Journal of Database Management (JDM)*, vol. 14, no. 4, p. 13, 2005.
- [34] F. Selleri Silva *et al.*, "Using CMMI together with agile software development: A systematic review," *Information and Software Technology*, vol. 58, pp. 20–43, 2015, doi: 10.1016/j.infsof.2014.09.012.
- [35] H. Ayed, B. Vanderose, and N. Habra, "Supported approach for agile methods adaptation: an adoption study," *Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering - RCoSE 2014*, no. March 2015, pp. 36–41, 2014, doi: 10.1145/2593812.2593820.
- [36] D. Gupta and R. Dwivedi, "Configurable Method Model of Agile Methods -for Creating Project-Specific Methods," pp. 86–91, 2016.
- [37] D. West, "Water-Scrum-Fall Is The Reality Of Agile For Most Organizations Today," *Forrester*, p. 17, 2011.