Management framework of software development: a systematic review

Marcos Espinoza-Mina, Alejandra Colina Vargas and Natalia Montero-Ramos

Universidad Ecotec, Samborondón, Ecuador

Abstract

Software development is a complex process since it synthesizes the theoretical and practical knowledge of all those involved in satisfying a business need at the technology level. This study proposes a systematic review of the literature on the administrative framework of software development. The main objective is to map the current literature concerning the management framework of software development. Methodologically, it was adjusted to the three phases proposed for a systematic review of the Kitchenham and Charter guidelines, which are: planning, review, and results. The review was limited to open-access scientific articles published in Spanish-speaking countries between 2013 and 2022. We obtained 62 in WoS and 153 in Scopus. According to the results, the components of the administrative framework are control, organization, people, planning, technologies, and technique. The most widely used methodology is the agile methodology. Among the trends that the development administrative framework must respond to are: Global Software Development, Secure Software Development, and Integrating the business intelligence framework. This study identifies that there are no unique criteria within the factors of the administrative framework.

Keywords

Development Methodology, Management, Projects, Software Engineering

1. Introduction

As information technologies (IT) continue to permeate companies, industries, and other sectors, they trigger software development initiatives that have an impact on modern organizations [1]. This development has become more popular over time, due to today's dynamic and complex business environment, which forces organizations to adapt quickly at the level of their structures, strategies, policies, and processes [2].

The demands for the construction of high-performance software products and services are increasing, since they must be able to solve problems in different areas such as business, industrial, agricultural, aeronautical, information, and communication technologies themselves to meet the needs of the surroundings [3, 4].

The importance of software is undeniable [4]. It is fundamental in all sectors of society since the computer program performs simple or complex activities, provides support, and causes dependency on said tools and other related technologies [5].

ICAIW 2023: Workshops at the 6th International Conference on Applied Informatics 2023, October 26–28, 2023, Guayaquil, Ecuador

mespinoza@ecotec.edu.ec (M. Espinoza-Mina); acolina@ecotec.edu.ec (A. Colina Vargas)

0000-0003-1530-7243 (M. Espinoza-Mina); 0000-0003-1514-8852 (A. Colina Vargas)

CEUR Workshop Proceedings (CEUR-WS.org)

From the described context and given the importance of software in today's world, its development is considered as a key activity that requires complex management scenarios. Based on three basic pillars: processes, technologies, and people. Managing this triangle is full of decisions and other management issues [6].

From one perspective, software development involves a range of activities, including the design and coding of software [7, 8], creatively, from an initial concept guided by a method or work system [9, 10].

On the other hand, it encompasses a set of common tasks, highlighted by requirements gathering and specification, design, code or unit testing, integration, testing and user support, and troubleshooting [11, 12].

According to the ISO/IEC 12207 standard, three processes are involved in software development: agreement processes, administration or technical management processes, and technical processes. Being relevant for this document, the technical management processes, since they involve the application of technical and administrative resources to plan, organize, and control engineering functions [13, 14].

The administrative or management framework carries with it activities to direct, control and continuously improve the organization within appropriate structures, a software development project is an example. They include the act, manner, or practice of organizing, managing, directing, supervising, and controlling the necessary resources. It ranges from one person in a small company to management hierarchies made up of many people in large companies [15].

Within the management of software projects, a set of administrative factors is established, starting from the planning and execution of software development activities, forming elements of the administrative framework. It requires today, in these projects, to plan shorter iterations and frequent changes in customer requirements. The development of process flow metrics helps to control and monitor the process, and in turn, adapt it to the context [6, 16].

This study aims to map the current literature on the subject concerning software development in Spanish-speaking countries. Revealing a series of theoretical and practical issues in recent publications, emphasizing the administrative framework, at the expense of any of the methods used for development and the technologies used [17, 18, 19]. In addition, a set of trends and success factors associated with software development are identified.

The organization of the remainder of the paper is described below. In Section 2, the methodology that guided this study was discussed. Section 3 was focused on the presentation and description of the results. Section 4 focused on the discussion of these results. Finally, Section 5 will present the conclusions.

2. Methodology

The study of the administrative framework of software development was guided by the systematic review methodology of software engineering literature proposed by [20]. According to the guide suggested by the authors, the work process is structured in three main stages (see Figure 1): a) plan the review, find related works, and determine the need for the review and the research question; (b) conduct the review, choose data sources and extract data and synthesis, and (c) results found [7].

Figure 1: Research method

2.1. Review planning

The Kitchenham and Charter guide proposes exploring related work, determining research needs, and creating a research protocol as the main tasks related to the planning phase.

2.1.1. Related work and needs for review

From the bibliographic review carried out, no scientific article was found related to the administrative framework of software development. Most of the identified studies focused on systematic reviews of administration or management from various contexts within software development. They highlight the management of assumptions, management of benefits, management of requirements, risk management, and management.

Firstly, the article presented by [21] in which they make an exploration and analysis of the state of the art on assumptions and their management in software development. Because stakeholders constantly make assumptions during software development, however, these are related to artifacts in requirements engineering and software design, which shows that there must be management of assumptions from the earliest phases of software development.

Regarding the work carried out by [22], point out that, currently, considerable resources are wasted in software development that delivers fewer benefits than planned, due to not having an administration or management. For their part, they recommend good management that includes identification and structuring, planning for the realization of benefits, and having people responsible for administration.

On the other hand, in the context of software development, requirements engineering is one of the crucial phases that leads to the success or failure of the software project. Hence [23] from the systematic review, they propose an adaptable and applicable conceptual framework to requirements management, which incorporates new ways of obtaining requirements according to the effects of communication, culture, competition and interested parties factors, incorporating tools, processes, methods, and techniques to solve problems integrally.

In another identified review study, the requirements dependency process is addressed, developed by [24] they emphasize that improper handling of requirement dependencies can lead to failures in software development. The authors state that requirements dependency has significant impacts on management, and the existence of various types of requirements dependency, and techniques to address requisite dependency problems with their corresponding limitations.

Likewise, a key element in the success of a software development project is effective risk management, the article presented by [25] as a review identified performance gaps; team

participation; attention to faults; identification of tools for decision making; and business strategy. To do this, they propose a set of new strategies and perspectives for risk management in software development, showing the importance of the current technology and innovation sector.

Finally, within the related works is the investigation of [1], which explores the factors involved in software development. He points out that technical and human resource factors have limited influence on software development projects. It highlights that the use of development methods eliminates ambiguities in software development.

This last scenario implies that the construction of software follows different paths, perspectives, or strategies [25], sometimes full of obstacles due to lack of knowledge for the selection and application of an adequate reference model to guide the development [26].

All of the above supports the need to carry out the proposed review, since there is a lack of standardization or unification of criteria when considering the administrative framework in a software development project [26].

In this sense, it emerges from the review that software development organizations are currently looking for tools and methods that help them maintain their competitiveness, focusing actions on the successful implementation of software process improvement [22].

Adjusting and making changes within the development procedure due to the agile nature of the process, which can be unpredictable and affect the cost and time of the software [27].

Once the need for the study is determined, the goals, objectives, and research questions are established visibly [28].

The purpose of this review is to map the current literature on the management framework of software development.

The research questions are formulated related to the objectives of the study, they guide and direct the development and fulfillment of the specific review criteria for the study [4, 29].

- **Q1:** What are the components that make up the administrative framework of software development?
- **Q2:** How do the methodologies or methods present these components of the administrative framework of software development?
- **Q3:** What are the new development perspectives or trends that should be incorporated into the formulation of the software development management framework?

2.1.2. Definition and evaluation of the research protocol

With the definition of the research protocol, the set of instructions that guides the study was provided, describing the source of information, the search and filtering strategies, the quality assessment, the data extraction strategies, and the way of communicating the results. It is emphasized that many activities are initiated during the protocol development stage and are refined when the review itself is carried out [20]. The elements of the study protocol are described below.

- **Source of information**
  Two databases with a high level of international scientific and academic prestige were included for the search of the articles considered in this review, such as Web of Science

(WoS) and Scopus. These databases make it possible to retrieve the largest number of relevant articles related to the administrative framework of development and cover most of the journals with an impact factor [28].

- **Search strategy**
  Given the nature of this review, certain terms related to the topic under study were extracted from the ERIC Thesaurus, to serve as limiters to help narrow the search [30]. The key terms were chosen to focus the search on the most relevant scientific articles for the topic under study, the combination of the words “software development”, “software construction”, “software programming”, “management organization”, “administration”, and “methodology”.
  To ensure the concordance of the search results in the selected scientific databases, the search strings (("software development" or "software construction" or "software programming") and ("management" or "organization" or "administration" or "methodology")).
  The search was carried out in WoS and Scopus, on March 26, 2023, filtering the results for the last ten years, from January 2013 to December 2022.
- **Filter Strategy**
  It is necessary to define criteria for inclusion and exclusion, which favors the filtering of search results [31], is proposed as inclusion criteria (see Table 1) and exclusion criteria (see Table 2)

Table 1
List of inclusion criteria

Nº	Criterion	Description
C1	Basic scope	Application of technical and administrative resources to plan, organize, and control software development in any type of software project [13]
IC2	Language	Only English language, since most important studies are published in that language.
IC3	Document type	Scientific journal articles
IC4	Area of knowledge	Articles must be framed in the field of computer science
IC5	Access type	All open access
IC6	Country	Spanish-speaking countries: Argentina, Bolivia, Chile, Colombia, Costa Rica, Cuba, the Dominican Republic, Ecuador, Equatorial Guinea, Guatemala, Mexico, Nicaragua, Panama, Paraguay, Peru, Puerto Rico, Spain, Uruguay and Venezuela

The quality assessment in this study is considered an important element of the systematic review of the literature (SRL), since it recognizes those works that adequately respond to the research questions and therefore to the proposed objective [32].
Based on the recommendations outlined in the guide for [20] a quality evaluation checklist was adapted, made up of seven questions (See Table 3), so that with the score obtained, the relevance of the selected documents can be evaluated at a general level. Then, the analysis was carried out by reading the full text [33, 34].

Table 2
List of exclusion criteria.

Nº	Criterion	Description
EC1	Irrelevant studies	Those studies outside the scope specified above.
EC2	Language	Studies that are not in the English language.
EC3	Duplicate studies	Repeated studies in more than one scientific database.

Following the proposal of [35] three scales from 0 to 1, coded according to (No = 0, Partially = 0.5, Yes = 1), were used to measure the quality of the selected studies. The results for each article ranged from 0 (very poor) to 5 (very good).

Table 3
Quality Assessment Checklist

Nº	Items
QC1	Is the purpose of the research clearly explained?
QC2	Is the research methodology clearly explained?
QC3	Is the specific subject area used well defined?
QC4	Have the results of the experiments carried out been identified and communicated?
QC5	Have the researchers provided sufficient data to support their results and conclusions?

The quality of the papers was assessed throughout the filtering and data extraction phase of this SRL, considering and cataloging as a "complete reading article" when the sum of criteria was greater than 3 points.

2.2. Perform the review

In the second phase, three main activities of the review are carried out, such as the search, the filtering, and the result of the quality evaluation, taking into account the research questions and the protocol raised in the previous phase.

- **The search**
  The search operation was carried out using the search string of the research protocol and the source of information, the search string was applied to the title, abstract, and keyword metadata of the WoS and Scopus databases.
  First, the data was retrieved from the bibliographic databases using the option to export in CSV (Comma Separated Values) format to be processed in a spreadsheet application (Microsoft Excel) and in R Studio software. The results obtained are shown in Table 4.
- **The filtering**
  The filtering procedure was performed in two stages, using the inclusion and exclusion criteria specified in the study research protocol. The exclusion codes (EC1-EC3) assigned to justify the reason for eliminating an article were considered. Papers that appeared in different sources, articles not relevant to the topics, and those not written in English were excluded.

Table 4
Extraction results related articles administrative framework of software development

Database used	Nº articles extracted	Nº repeated articles	First result
WoS	62	46	16
Scopus	153	-	153
Total	-	-	169

The first phase of filtering is to read each article's title, keywords, and abstracts to determine if they link to the research questions. An attempt was made to keep those items from the first chain that at least complied with the planning, organization, or control of software development.
At the end of the first round, 58 studies were selected for the first data set for full-text filtering, while 115 studies were excluded.

- **Result of the quality evaluation**
  The quality of the selected main research papers was evaluated based on the research design, conduct, data analysis, and conclusions of the study papers. Once the 58 complete documents were evaluated, it was determined that 91.37% had met the adapted criteria, thus showing that they are of good quality.

3. Results

3.1. Overview of studies

The articles used in the study were retrieved during March 2023, from the Web of Science (WoS) and Scopus databases. The chronology of the publications of the two scientific databases covers the years from 2013 to 2022 (see Figure 2). With an annual percentage increase of 14.31% and 19.58, in WoS and Scopus, respectively. Between 2014 and 2015 there was the lowest number of investigations in both WoS and Scopus with 2 and 7 articles, respectively.

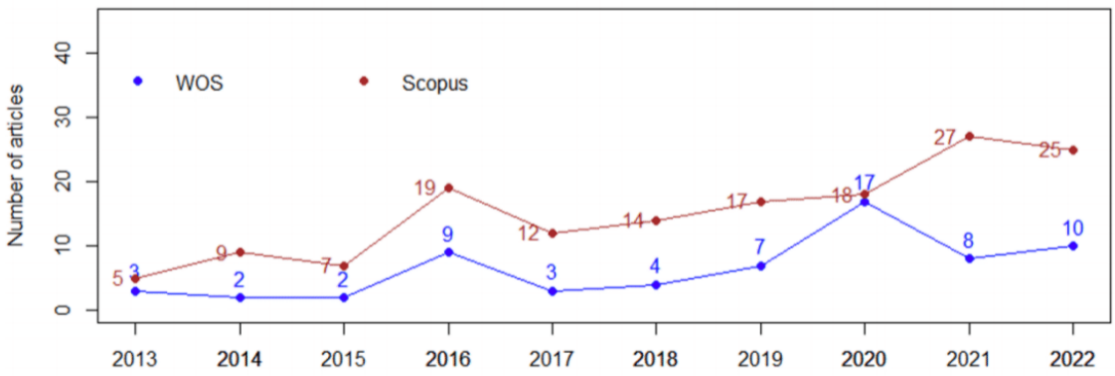

Figure 2: Chronology articles

Figure 3 shows the articles extracted from WoS for the SRL grouped by country. It is noted that Spain is the country that publishes the most articles, with 40 which is equivalent to 60.11%, followed by Mexico with 7 articles that represent 10.61%.

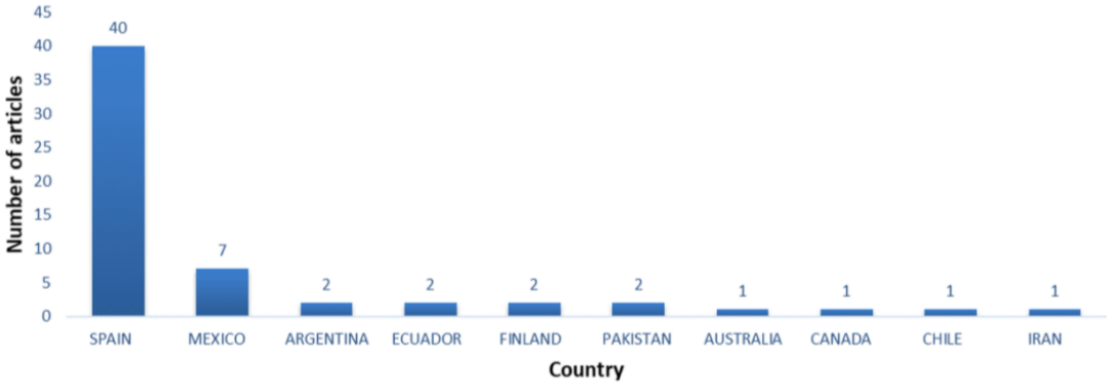

Figure 3: Articles by country retrieved in WoS

For its part, Figure 4 contains the articles retrieved from Scopus, highlighting Spain as the country with the most articles, it has 46, which represents 51.11%, then Mexico comes with 11 articles representing 12.22%.

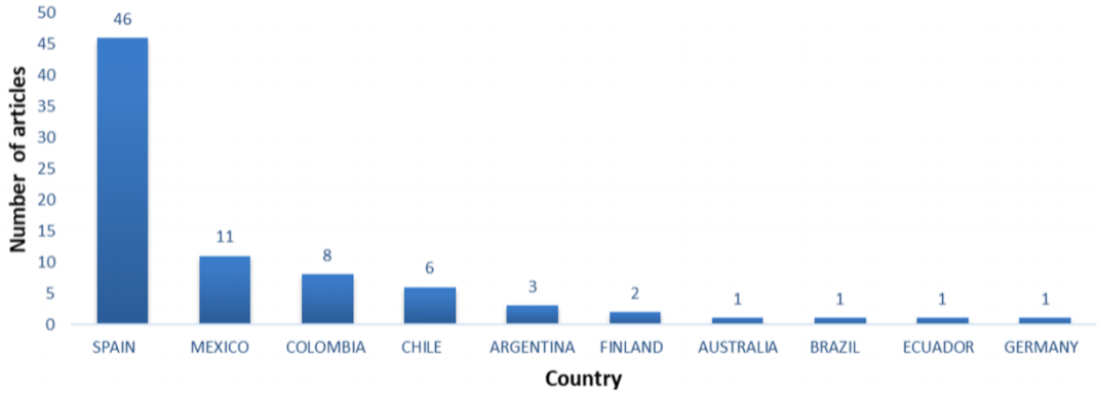

Figure 4: Articles by country retrieved in Scopus

3.2. Results of the research questions

3.2.1. What are the components that make up the management framework of software development?

To answer this research question, a careful analysis of software development as a process and not as another phase of the software life cycle was carried out.

Recognition of the components of the software development management framework is looked for in the scientific articles selected for the study.

A list of six components was used that group together the coincidences in the aspects of management in the organization, planning, control, and resources that are followed in the development of software taken from theoretical references consulted[9, 13, 15, 17] (see Table 5).

It is observed that 54% of the selected articles comply with more than three of the listed components of the software development management framework, while the rest consider at least one. The "Control" component is the most frequent with 37 articles, followed by "Organizational" with 34 articles.

These results confirm that in software development projects, the estimation, planning, and management processes are decisive since the results must be related to various business strategies [36]. It is also established that successful management practice depends on a series of non-technical issues that are of a managerial, cultural, and organizational nature.

Each software product is considered a representation of the knowledge of all the people involved [37]. In this sense, software companies have made many efforts to improve software development practices, using different tools, techniques, and models to achieve the quality of the software they produce, being demanded by customers [38].

Since software is a product, it not only needs the most appropriate development methodology but also an efficient project management strategy that is capable of estimating and managing development times, and managing potential risks in the best possible way. to deliver high-quality software products and services on time [39].

In the case of distributed software development, planning and managing a fair and transparent task assignment is critical and challenging [40].

It is necessary, within the elements of the development administrative framework, to cover at least two requirements: (1) integrate a culture of process control and continuous improvement and (2) provide high-quality personnel with the knowledge and skills to work successfully under the models and standards used in organizations [41, 42].

Another element that emerges from the review, as a component of the organization, is communication, key in the exchange of information between team members, to which are added, in some cases, additional obstacles such as differences in time zones and cultural, IT infrastructure and other delays in the act of communication and leadership [40, 43, 44].

Regarding productivity in software development, there is a specific interest in identifying its influence factors. It is classified into technical factors, organizational factors, product factors, project factors and personal factors. However, these approaches tend to focus on technical factors to the detriment of social and human or personal factors [45, 46].

Others appeal to design models that allow for the reconciliation of software development management and organizational management through the relationship of factors associated with production in software factories and administrative factors, aligning operational metrics with objectives. factories strategic [47].

Given the importance of software in today's world, development is a key activity that requires complex management scenarios. It requires that the implications of difficult decisions be explored in the context of software development projects, emphasizing the emotional consequences, and the human factor in decision-making in IT organizations [6, 48, 49].

Likewise, it was identified in the review that the lack of unified theoretical and methodological

criteria from software engineering, to address human aspects in the administrative framework of software development [50].

These results are a reflection of the management needs that exist within software development, since they must increasingly adapt to teams that work together, but are located in different geographical locations, contexts, which is having a profound impact on the way to conceive, design, build and test the products [51, 52].

Table 5
Components of the software development management framework

Components	Item	No. Items
Control	37	[53, 54, 39, 46, 55, 56, 57, 52, 26, 41, 58, 59, 60, 51, 61, 62, 63, 36, 64, 65, 66, 45, 67, 68, 37, 69, 48, 70, 71, 72, 47, 73, 74, 75, 76, 77, 78]
Organizational	34	[54, 39, 46, 56, 57, 26, 41, 58, 59, 38, 79, 61, 62, 36, 64, 65, 68, 37, 69, 43, 70, 42, 49, 44, 80, 72, 47, 73, 81, 74, 75, 76, 77, 78]
People	32	[39, 56, 82, 26, 41, 59, 38, 79, 51, 61, 83, 63, 36, 45, 67, 68, 37, 40, 69, 48, 70, 42, 49, 44, 80, 50, 84, 75, 76, 78]
Planning	20	[54, 39, 46, 56, 57, 52, 26, 41, 59, 36, 64, 68, 49, 72, 81, 74, 75, 76, 77, 78]
Technologies	20	[54, 39, 56, 82, 41, 85, 83, 63, 36, 66, 67, 68, 37, 42, 49, 71, 84, 76, 78]
Technical	17	[39, 56, 82, 41, 63, 36, 66, 45, 43, 72, 47, 73, 74, 76, 77, 78]

3.2.2. How do the methodologies raise these components of the administrative framework of software development?

This question seeks to know what the main software development methodologies are and how they present the administrative framework of software development in the set of selected scientific articles.

Software development methodologies are considered to provide a framework for organizing, planning, estimating, controlling, monitoring, and measuring the processes and activities of a software development project [36].

The literature reviewed suggests two marked trends in the types of development methodology: the traditional or plan-based approach and agile. This classification is subject to key elements to the nature of the software to be developed, the competencies of the development team, and the culture of the organization that is designing the system are key elements to consider [9].

Table 6 presents the scientific articles reviewed associated with the methodologies, identifying aspects related to the administrative framework of each of the methodologies. The agile methodology is the most referenced with 57.58%, followed by other methodologies with 39.39%.

The studies analyzed show a significant improvement in the number of successful projects and the optimization of the use of the necessary resources in software development, through the use of continuously evolving agile methodologies. Software development management under a traditional approach has become insufficient due to the high variability of requirements and the need to change the organization of project work teams [28].

In agile software development, knowledge is prone to vanish, due to what is established in the agile manifesto, documentation is not a priority, it is shared in the code phase and it constitutes a way of linking the resources used by developers in the project and the experts [37].

The rise of agile methodologies and practices has provided some useful tools that, combined with Web Engineering techniques, can help establish a framework for estimating, managing, and planning Web development projects [36]. Agile approaches tend to focus solely on scope and simplicity rather than problem-solving and discovery [63].

In this sense, software development processes increasingly incorporate tools and systems that support and contribute to the challenge of seeking optimal use of human and technical resources of the project. In this regard, agile methodologies, such as Scrum and extreme programming (XP) methods, have been key factors in understanding what software projects face [26].

In this particular case, the relevance of Scrum, as a benchmark for agile methodology, is based on the fact that it allows identifying the problems of the process, and exploring various approaches, but not their fundamental causes. Defines project schedules based largely on the speed of sprints [75].

Likewise, within other emerging methodologies is the "InterMod Methodology", a combination of accepted fundamentals of agile methods, whose development is based on models and user-centered design, allowing the development of high-quality interactive applications. As a main characteristic, it plans and organizes the software project as a series of iterations guided by the User Objectives in an agile and user-centered way [76].

A series of models are presented as results of the review to reconcile software development management and organizational management through the relationship of factors associated with production in software developers and administrative factors, aligning operational metrics with the strategic goals of developer organizations.

Relying in this case, the Capability of Integrated Maturity (CMMI), the Personal Software Processes (PSP), or the Software for Teams (TSP), to support the continuous improvement of software factories, reduce quality costs and the number of defects, increase project profitability and customer satisfaction [47, 48].

On the other hand, in the case of small and medium development companies (SMEs) they cannot afford to carry out tool evaluations for each project that require a lot of time and personnel, and instead adopt sets of fixed tools, called evaluation tools. application lifecycle management (ALM), which allows them to make multi-criteria decisions (MCDM) in solving problems, given the little flexibility, one for each ALM domain [86].

Several adaptations or extensions of the agile methodology were found in this research, one of them is the hybrid development model, with the integration of User-Centered Design (UCD), whose body of knowledge of software development is given through the evaluation of the participation of stakeholders and users, providing more human-centered methodologies and with an overview of requirements management to offer competitive products with an adequate User Experience (UX) [79].

Table 6
Methodologies used in the administrative framework of software development

Name of methodologies	Item	No. Items
Agile methodology	19	[46, 87, 56, 52, 60, 83, 62, 63, 36, 65, 68, 37, 40, 69, 44, 73, 84, 75, 78]
Traditional methodologies	1	[68]
Other Methodologies	13	[55, 85, 88, 89, 90, 68, 40, 69, 70, 42, 76, 77]

3.2.3. What are the new perspectives or trends in the formulation of the administrative framework of software development?

From the review carried out in response to the question, there are different trends or perspectives of the software development management framework, showing as relevant to "Global Software Development", "Secure Software Development" and "Integrate Business Intelligence Framework (BI) applied to software development", mainly.

Regarding the trend toward Global Software Development (GSD), it is conceived as a well-established field within software engineering with the benefits of a global environment. Software project management plays a key role in the success of the GSD, as it provides tools to be used for coordination, planning, and supervision, along with estimation techniques that can be used to better fit a distributed project [74].

For its part, the accelerated growth in the exploitation of vulnerabilities due to errors or failures in the software development process has become a recurring concern for the Software Industry. Today there is a diversity of methodologies, models, and tools with specific objectives in each stage of secure software development [53].

Another element that stands out in the study is the increase in the number of cyberattacks in recent years, as well as their sophistication and impact. Therefore, they demand new emerging models of software development that help develop secure software by default. A new secure software methodology is proposed, adapted to all current environments in terms of security and software quality, providing more secure software [90].

The exploitation of data related to software development from intensive software development organizations to support tactical and strategic decision-making is a challenge since it supports guides and tools to exploit data related to software development and knowledge. expert to improve your decision-making [88].

In the last decade, agile methods have changed the software development process in unprecedented ways, and with the growing popularity of Big Data, optimizing development cycles through data analytics is becoming a commodity. essential [84].

Business Intelligence techniques improve GSD management beyond the information provided by traditional tools. They allow information to be integrated and presented in a single place, thus enabling easy comparisons between multiple projects and factories and providing support for informed decision-making in GSD management [58].

Another trend line in the software process is potentially suitable for building software development methodologies through the reuse of basic assets. However, adopting this approach without prior evaluation of its suitability can lead to failure [91].

For this reason, software development must be based on the reuse of old hardware, software, and non-proprietary free code (open source), as well as the virtualization of servers and machines, to create software that can be useful for more than a decade [55].

4. Discussion

This study has been carried out to map the current literature on the subject concerning the administrative framework of software development in Spanish-speaking countries, identifying a total of 58 relevant studies that were analyzed following the research protocol defined in the review.

First, the results of the review reflect the achievement of the main objective of this study by answering the research questions posed. The interest and importance of the topic under study are confirmed, evidencing an increase in publications from 2013 to 2022, with a slight decrease in 2020, possibly attributable to the global pandemic.

In addition, the diversity of aspects addressed in the articles reviewed demonstrates that this is a complex field of research with numerous cross-functional influences, given its close relationship with the business context in general, which is analyzed in most of the studies.

Concerning the first research question, six components characterizing the administrative framework of software development are proposed, which encompass the key processes and technical and managerial activities to be considered in software development. The "control" and "organizational" components are the most frequent due to their significant impact on the software development process.

Studies indicate that there is no standard the definition of the management framework; studies indicate that there is no standard in defining the management framework; however, it is generally suggested that organizations need to thoroughly assess their readiness before embarking on a development project, regardless of the methodology to be followed. Identifying these components in a specific software project for a given organization could increase the likelihood of success by improving the relationship with other factors that influence development.

It is essential to involve all stakeholders to validate all assumptions during development, as the "People" component plays a critical role in the success of the development. This conditions the planning, monitoring, organization, and control processes, with the objective that the results are aligned with the various business strategies.

Regarding the second research question, studies suggest that agile methodologies facilitate the integration of a culture of processes and continuous improvement in organizations, due to their adaptability to changing and complex contexts.

Some of the emerging challenges that the administrative framework for development needs to address include Global Software Development, Secure Software Development, and the integration of the business intelligence framework.

5. Conclusions

In the systematic review, two internationally recognized scientific and academic databases, Web of Science and SCOPUS, were used as primary sources of consultation.

The review process was guided by the Kitchenham & Charters methodology, identifying, verifying, and validating the quality and contribution of the selected articles according to the subject of study. The search was divided into two groups, one for the administrative framework with 62 and 153 articles, from WoS and SCOPUS, respectively.

The chronology of the selected publications was established for the years 2013 to 2022; showing an annual percentage increase of 14.31% and 19.58, in WoS and Scopus, respectively. The results were shown by country, with Spain standing out in the two databases, with the largest number of articles.

The results were shown by country, standing out in the two databases Spain, with the largest number of articles.

The systematic review of the literature presented was aimed at providing a complete mapping of the studies focused on the recognition of the administrative framework of software development aimed at Spanish-speaking countries and the identification of trends.

Regarding the identification of related works according to the research protocol, the inexistence of scientific articles that deal with a review of the literature referring to the administrative framework of software development stands out. However, some relevant works of reviews of different management contexts related to software development were taken into account.

The results identify a diversity of elements that make up the administrative framework within which coincidence in the studies stands out concerning six key components: control, organization, people, planning, technologies, and technique. These aspects are significant to successfully manage software development regardless of the methodology applied.

It was possible to identify within the results at the element level a new management tool, called "Expert Coaching", a specialist role that guides and motivates the development team in meeting objectives and goals.

Regarding the relevant methodology in the administrative framework of software development, there is a growing increase in publications that use agile methodologies, with their different methods and adaptations, which arises in response to the urgent need to undertake software development projects, with the highest quality, in the shortest possible time and with minimal change work, and after it is put into production.

In addition, there was evidence of diversity in the publications regarding the management of software development using other methodologies. However, the increase in scientific publications is confirmed by the implementation of good practices guided by the ISO/IEC 29110 standards, with his family. This is a result of the fact that small companies are leading the development market worldwide, there has been an interest in recent decades to reinforce the management of development projects, with recognized reference frameworks or good practices.

It stands out within the new perspectives or trends in the formulation of the administrative framework of software development, such as proposals oriented towards "Global Software Development", "Secure Software Development" and "Integrating the business intelligence framework applied to the software development".

These promote that not only value quality but incorporate software security, from the development phase; given that currently, it is incorporated only at the end of it and the high value provided by the projects is not taken into account.

A significant trend is the integration of data analysis business intelligence tools and Big Data into software development, and in this way, the development cycles or phases are optimized, at

the code level, tests, and project management in general.

This study contributed with an acknowledgment and update of the different aspects to be considered in software development, at the administrative level, which are benchmarks for success in good software project management. This knowledge obtained serves as a basis for future research.

References

[1] S. J. Dubey, Key success factors in software development projects, Available at SSRN 1952935 (2011).

[2] S. Nerur, R. Mahapatra, G. Mangalaraj, Challenges of migrating to agile methodologies, Communications of the ACM 48 (2005) 72–78.

[3] M. Muñoz, M. Negrete, M. Arcilla-Cobián, Using a platform based on the basic profile of iso/iec 29110 to reinforce devops environments., J. Univers. Comput. Sci. 27 (2021) 91–110.

[4] M. A. Akbar, J. Sang, A. A. Khan, M. Shafiq, S. Hussain, H. Hu, M. Elahi, H. Xiang, et al., Improving the quality of software development process by introducing a new methodology–az-model, IEEE Access 6 (2017) 4811–4823.

[5] M. E. Bogopa, C. Marnewick, Critical success factors in software development projects, South African Computer Journal 34 (2022) 1–34.

[6] R. Colomo-Palacios, C. Casado-Lumbreras, P. Soto-Acosta, Á. García-Crespo, Decisions in software development projects management. an exploratory study, Behaviour & Information Technology 32 (2013) 1077–1085.

[7] M. Zhang, S. Gu, Y. Shi, The use of deep learning methods in low-dose computed tomography image reconstruction: a systematic review, Complex & intelligent systems 8 (2022) 5545–5561.

[8] M. Elmezain, W. H. W. Baduruzzaman, M. A. Khoiry, The impact of project manager's skills and age on project success, Brazilian Journal of Operations & Production Management 18 (2021) 1–16.

[9] I. Sommerville, Ingeniería del software, Pearson educación, 2005.

[10] H. Florez, M. Leon, Model driven engineering approach to configure software reusable components, in: Applied Informatics: First International Conference, ICAI 2018, Bogotá, Colombia, November 1-3, 2018, Proceedings 1, Springer, 2018, pp. 352–363.

[11] B. Davis, Mastering Software Project Requirements: A Framework for Successful Planning, Development & Alignment, J. Ross Publishing, 2013.

[12] F. Tsui, O. Karam, B. Bernal, Essentials of software engineering, Jones & Bartlett Learning, 2022.

[13] I. ISO, Iec 12207 systems and software engineering-software life cycle processes, International Organization for Standardization: Geneva (2017).

[14] O. A. Fonseca-Herrera, A. E. Rojas, H. Florez, A model of an information security management system based on ntc-iso/iec 27001 standard, IAENG Int. J. Comput. Sci 48 (2021) 213–222.

[15] I. ISO, Information technology — security techniques — information security management systems, International Organization for Standardization (2018).

[16] K. S. Bitla, S. S. Veesamsetty, Measuring process flow using metrics in agile software development, 2019.
[17] H. Edison, X. Wang, K. Conboy, Comparing methods for large-scale agile software development: A systematic literature review, IEEE Transactions on Software Engineering 48 (2021) 2709–2731.
[18] H. Florez, E. Garcia, D. Muñoz, Automatic code generation system for transactional web applications, in: Computational Science and Its Applications–ICCSA 2019: 19th International Conference, Saint Petersburg, Russia, July 1–4, 2019, Proceedings, Part V 19, Springer, 2019, pp. 436–451.
[19] D. Sanchez, H. Florez, Model driven engineering approach to manage peripherals in mobile devices, in: Computational Science and Its Applications–ICCSA 2018: 18th International Conference, Melbourne, VIC, Australia, July 2–5, 2018, Proceedings, Part IV 18, Springer, 2018, pp. 353–364.
[20] B. Kitchenham, S. Charters, et al., Guidelines for performing systematic literature reviews in software engineering version 2.3, Engineering 45 (2007) 1051.
[21] C. Yang, P. Liang, P. Avgeriou, Assumptions and their management in software development: A systematic mapping study, Information and Software Technology 94 (2018) 82–110.
[22] W. Wysocki, I. Miciuła, M. Mastalerz, Classification of task types in software development projects, Electronics 11 (2022) 3827.
[23] A. Chaipunyathat, N. Bhumpenpein, Communication, culture, competency, and stakeholder that contribute to requirement elicitation effectiveness., International Journal of Electrical & Computer Engineering (2088-8708) 12 (2022).
[24] F. Noviyanto, R. Razali, M. Z. Ahmad Nazri, Understanding requirements dependency in requirements prioritization: a systematic literature review., International Journal of Advances in Intelligent Informatics 9 (2023).
[25] K. G. O. Valério, C. E. S. da Silva, S. M. Neves, Risk management in software development projects: systematic review of the state of the art literature, International Journal of Open Source Software and Processes (IJOSSP) 11 (2020) 1–22.
[26] B. L. Flores-Rios, F. J. Pino, J. E. Ibarra-Esquer, F. F. González-Navarro, O. M. Rodríguez-Elías, Análisis de flujos de conocimiento en proyectos de mejora de procesos software bajo una perspectiva multi-enfoque/analysis of knowledge flows in software process improvement projects under a multi-perspective approach, Revista Ibérica de Sistemas e Tecnologias de Informação (2014) 51.
[27] B. Habib, R. Romli, M. Zulkifli, Identifying components existing in agile software development for achieving "light but sufficient" documentation, Journal of Engineering and Applied Science 70 (2023) 75.
[28] L. Mendoza-Pitti, H. Calderón-Gómez, M. Vargas-Lombardo, J. M. Gómez-Pulido, J. L. Castillo-Sequera, Towards a service-oriented architecture for the energy efficiency of buildings: a systematic review, IEEE Access 9 (2021) 26119–26137.
[29] C. Lockwood, Z. Munn, K. Porritt, Qualitative research synthesis: methodological guidance for systematic reviewers utilizing meta-aggregation, JBI Evidence Implementation 13 (2015) 179–187.
[30] ERIC, Education resources information center, Institute of Education Sciences (2023).

[31] B. Kitchenham, P. Brereton, A systematic review of systematic review process research in software engineering, Information and software technology 55 (2013) 2049–2075.
[32] A. Amores-Valencia, D. Burgos, J. W. Branch-Bedoya, Influence of motivation and academic performance in the use of augmented reality in education. a systematic review, Frontiers in Psychology 13 (2022) 1011409.
[33] M. Azzeh, A. B. Nassif, I. B. Attili, Predicting software effort from use case points: A systematic review, Science of Computer Programming 204 (2021) 102596.
[34] R. Isnin, A. A. Bakar, N. S. Sani, et al., Does artificial intelligence prevail in poverty measurement?, 2020.
[35] S. O. Oruma, M. Sánchez-Gordón, R. Colomo-Palacios, V. Gkioulos, J. K. Hansen, A systematic review on social robots in public spaces: Threat landscape and attack surface, Computers 11 (2022) 181.
[36] C. J. Torrecilla-Salinas, J. Sedeño, M. Escalona, M. Mejías, Estimating, planning and managing agile web development projects under a value-based perspective, Information and Software Technology 61 (2015) 124–144.
[37] M. A. P. Pardo, H. A. O. Erazo, C. A. C. Lozada, Documenting and implementing devops good practices with test automation and continuous deployment tools through software refinement, Periodicals of Engineering and Natural Sciences 9 (2021) 854–863.
[38] N. Morales-Aguiar, V. Vega-Zepeda, Factores humanos y la mejora de procesos de software. propuesta inicial de un catálogo que guíe su gestión, Revista Ibérica de Sistemas e Tecnologias de Informação (2018) 30–42.
[39] S. Alcaraz-Corona, J. L. Cantú-Mata, F. Torres-Castillo, Exploratory factor analysis for software development projects in mexico, Statistics, Optimization & Information Computing 7 (2019) 85–96.
[40] A. Yagüe, J. Garbajosa, J. Díaz, E. González, An exploratory study in communication in agile global software development, Computer Standards & Interfaces 48 (2016) 184–197.
[41] M. Muñoz, J. Mejia, C. Y. Laporte, Implementación del estándar iso/iec 29110 en centros de desarrollo de software de universidades mexicanas: Experiencia del estado de zacatecas, RISTI-Revista Ibérica de Sistemas e Tecnologias de Informação,(29) (2018) 43–54.
[42] M.-L. Sánchez-Gordón, R. V. O'Connor, Understanding the gap between software process practices and actual practice in very small companies, Software Quality Journal 24 (2016) 549–570.
[43] C. Gupta, V. Gupta, A decentralized framework for managing task allocation in distributed software engineering, Applied Sciences 11 (2021) 10633.
[44] G. Gutiérrez, M. T. G. De Lena, J. Garzás, J. M. Moguerza, Leadership styles in agile teams: An analysis based on experience, IEEE access 10 (2022) 19232–19241.
[45] L. Machuca-Villegas, G. G. Hurtado, S. M. Puente, L. M. R. Tamayo, An instrument for measuring perception about social and human factors that influence software development productivity., J. Univers. Comput. Sci. 27 (2021) 111–134.
[46] S. Galvan-Cruz, M. Mora, R. V. O'Connor, F. Acosta, F. Álvarez, An objective compliance analysis of project management process in main agile methodologies with the iso/iec 29110 entry profile, International Journal of Information Technologies and Systems Approach (IJITSA) 10 (2017) 75–106.
[47] K. J. Cerón, H. F. Arboleda, Modelo de relacionamiento estratégico entre factores organi-

zacionales y técnicos en fábricas de software colombianas, Información tecnológica 29 (2018) 29–38.
[48] A. E. Chavarría, S. B. Oré, C. Pastor, Aseguramiento de la calidad en el proceso de desarrollo de software utilizando cmmi, tsp y psp/quality assurance in the software development process using cmmi, tsp and psp, Revista Ibérica de Sistemas e Tecnologias de Informação (2016) 62.
[49] B. Adenso-Díaz, S. Lozano, E. Gutiérrez, L. Calzada, S. Garcia, Assessing individual performance based on the efficiency of projects, Computers & Industrial Engineering 107 (2017) 280–288.
[50] R. Juárez-Ramírez, C. X. Navarro, G. Licea, S. Jiménez, V. Tapia-Ibarra, C. Guerra-García, H. G. Perez-Gonzalez, How covid-19 pandemic affects software developers' wellbeing, and the necessity to strengthen soft skills, Programming and Computer Software 48 (2022) 614–631.
[51] R. Colomo-Palacios, L. J. López-Cuadrado, I. González-Carrasco, J. F. García-Peñalvo, Sabumo-dtest: Design and evaluation of an intelligent collaborative distributed testing framework, Computer Science and Information Systems 11 (2014) 29–45.
[52] J. L. C. Valdez, H. F. Medina, G. A. V. Contreras, G. C. Morales, A. H. G. González, Collaborative integrated model in agile software development (mdsic/mdsic−m)-case study and practical advice, International Journal of Advanced Computer Science and Applications 10 (2019).
[53] H. Nina, J. A. Pow-Sang, M. Villavicencio, Systematic mapping of the literature on secure software development, IEEE Access 9 (2021) 36852–36867.
[54] G. Baños, K. Melendez, A. Dávila, Prácticas adoptadas de la iso/iec 20000 en pequeñas organizaciones desarrolladoras de software que ofrecen mesa de servicios. un estudio de caso/iso/iec 20000 practices adopted in small software development organizations that offer services desk. a case study, Revista Iberica de Sistemas e Tecnologias de Informação (2016) 1.
[55] E. Barra, J. Morato, Early knowledge organization assisted by aspects, Science of Computer Programming 121 (2016) 34–54.
[56] C. Gupta, J. M. Fernandez-Crehuet, V. Gupta, Measuring impact of cloud computing and knowledge management in software development and innovation, Systems 10 (2022) 151.
[57] A. Meidan, J. A. García-García, M. Escalona, I. Ramos, A survey on business processes management suites, Computer Standards & Interfaces 51 (2017) 71–86.
[58] A. Maté, J. Trujillo, F. García, M. Serrano, M. Piattini, Empowering global software development with business intelligence, Information and Software Technology 76 (2016) 81–91.
[59] A. Lluís Mesquida, A. Mas, T. San Feliu, M. Arcilla, Integración de estándares de gestión de ti mediante min-its., RISTI: Iberian Journal on Information Systems & Technologies/Revista Ibérica de Sistemas e Tecnologias de Informação (2014).
[60] M. Choraś, T. Springer, R. Kozik, L. López, S. Martínez-Fernández, P. Ram, P. Rodriguez, X. Franch, Measuring and improving agile processes in a small-size software development company, IEEE access 8 (2020) 78452–78466.
[61] J. Nicolás, J. M. C. De Gea, B. Nicolas, J. L. Fernandez-Aleman, A. Toval, On the risks and safeguards for requirements engineering in global software development: Systematic

literature review and quantitative assessment, IEEE Access 6 (2018) 59628–59656.
[62] J. de Vicente Mohino, J. Bermejo Higuera, J. R. Bermejo Higuera, J. A. Sicilia Montalvo, The application of a new secure software development life cycle (s-sdlc) with agile methodologies, Electronics 8 (2019) 1218.
[63] A. Aldave, J. M. Vara, D. Granada, E. Marcos, Leveraging creativity in requirements elicitation within agile software development: A systematic literature review, Journal of Systems and Software 157 (2019) 110396.
[64] R. J. Martelo, I. Jiménez-Pitre, L. Moncaris González, Guía metodológica para el mejoramiento del desarrollo de software a través de la aplicación de la técnica árboles de problemas, Información tecnológica 28 (2017) 87–94.
[65] L. López, X. Burgués, S. Martínez-Fernández, A. M. Vollmer, W. Behutiye, P. Karhapää, X. Franch, P. Rodríguez, M. Oivo, Quality measurement in agile and rapid software development: A systematic mapping, Journal of Systems and Software 186 (2022) 111187.
[66] V. Muntés-Mulero, O. Ripolles, S. Gupta, J. Dominiak, E. Willeke, P. Matthews, B. Somosköi, Agile risk management for multi-cloud software development, IET Software 13 (2019) 172–181.
[67] G. P. Gasca-Hurtado, M. C. Gómez-Álvarez, L. Machuca-Villegas, M. Muñoz, Design of a gamification strategy to intervene in social and human factors associated with software process improvement change resistance, IET Software 15 (2021) 428–442.
[68] A. Alvear Suárez, V. A. Bollati, J. M. Vara Mesa, Realities and perspectives of software development in puerto rico: a new analysis, 2021.
[69] W. M. Ocampo, E. Suescun, C. J. P. Calvache, State of agile contracting in the software industry and the public sector, results of a systematic mapping of the literature, Periodicals of Engineering and Natural Sciences 9 (2021) 375–385.
[70] J. Gallardo, A. I. Molina, C. Bravo, M. A. Redondo, A model-driven and task-oriented method for the development of collaborative systems, Journal of network and computer applications 36 (2013) 1551–1565.
[71] C. A. Guerrero, J. M. Londoño, Revisión de la problemática de la calidad del software para el desarrollo de aplicaciones de computación en la nube, Información tecnológica 27 (2016) 61–80.
[72] M. Muñoz, P. Montoya-Méndez, Rutas de apoyo a las emps para la selección de herramientas que faciliten la implementación del estándar iso/iec 29110, Revista Ibérica de Sistemas e Tecnologias de Informação (2022) 3–23.
[73] M. Muñoz, J. Mejia, C. Y. Laporte, Implementing iso/iec 29110 to reinforce four very small entities of mexico under an agile approach, IET Software 14 (2020) 75–81.
[74] M. El Bajta, A. Idri, J. N. Ros, J. L. Fernández-Alemán, J. M. C. de Gea, F. García, A. Toval, Software project management approaches for global software development: A systematic mapping study, Tsinghua Science and Technology 23 (2018) 690–714.
[75] F. Albero Pomar, J. A. Calvo-Manzano, E. Caballero, M. Arcilla-Cobián, Understanding sprint velocity fluctuations for improved project plans with scrum: a case study, Journal of Software: Evolution and Process 26 (2014) 776–783.
[76] C. Fernández-Sánchez, J. Garbajosa, A. Yagüe, J. Perez, Identification and analysis of the elements required to manage technical debt by means of a systematic mapping study, Journal of Systems and Software 124 (2017) 22–38.

[77] F. Pinciroli, J. L. B. Justo, R. Forradellas, Systematic mapping study: On the coverage of aspect-oriented methodologies for the early phases of the software development life cycle, Journal of King Saud University-Computer and Information Sciences 34 (2022) 2883–2896.
[78] M. Hamid, F. Zeshan, A. Ahmad, F. Ahmad, M. A. Hamza, Z. A. Khan, S. Munawar, H. Aljuaid, An intelligent recommender and decision support system (irdss) for effective management of software projects, IEEE Access 8 (2020) 140752–140766.
[79] E.-M. Schön, J. Thomaschewski, M. J. Escalona, Agile requirements engineering: A systematic literature review, Computer standards & interfaces 49 (2017) 79–91.
[80] P. Ralph, S. Baltes, G. Adisaputri, R. Torkar, V. Kovalenko, M. Kalinowski, N. Novielli, S. Yoo, X. Devroey, X. Tan, et al., Pandemic programming: How covid-19 affects software developers and how their organizations can help, Empirical software engineering 25 (2020) 4927–4961.
[81] G. Robles, A. Capiluppi, J. M. Gonzalez-Barahona, B. Lundell, J. Gamalielsson, Development effort estimation in free/open source software from activity in version control systems, Empirical Software Engineering 27 (2022) 135.
[82] B. Bossavit, S. Parsons, From start to finish: teenagers on the autism spectrum developing their own collaborative game, Journal of Enabling Technologies 11 (2017) 31–42.
[83] A. Hinderks, F. J. D. Mayo, J. Thomaschewski, M. J. Escalona, Approaches to manage the user experience process in agile software development: A systematic literature review, Information and Software Technology 150 (2022) 106957.
[84] K. Biesialska, X. Franch, V. Muntés-Mulero, Big data analytics in agile software development: A systematic mapping study, Information and Software Technology 132 (2021) 106448.
[85] V. P. Castro-Rivera, R. A. Herrera-Acuña, M. A. Villalobos-Abarca, Desarrollo de un software web para la generación de planes de gestión de riesgos de software, Información tecnológica 31 (2020) 135–148.
[86] M. Pilar, J. Simmonds, H. Astudillo, Semi-automated tool recommender for software development processes, Electronic Notes in Theoretical Computer Science 302 (2014) 95–109.
[87] O. A. A. Galán, J. L. C. Valdéz, H. F. Medina, G. A. V. Contreras, J. L. S. Sumuano, Proposal of a sustainable agile model for software development, International Journal of Advanced Computer Science and Applications 11 (2020).
[88] M. Manzano, C. Ayala, C. Gómez, A. Abherve, X. Franch, E. Mendes, A method to estimate software strategic indicators in software development: An industrial application, Information and Software Technology 129 (2021) 106433.
[89] D. A. Franco, J. L. Perea, P. Puello, Metodología para la detección de vulnerabilidades en redes de datos, Información tecnológica 23 (2012) 113–120.
[90] J. C. S. Núñez, A. C. Lindo, P. G. Rodríguez, A preventive secure software development model for a software factory: a case study, IEEE Access 8 (2020) 77653–77665.
[91] H. Agh, F. Garcia, M. Piattini, R. Ramsin, Requirements for adopting software process lines, Journal of Systems and Software 164 (2020) 110546.