Research on Humanities and Social Sciences
ISSN 2224-5766 (Paper) ISSN 2225-0484 (Online)
Vol.12, No.14, 2022

Maturity Models for Managing People in Software Development Teams: A Systematic Literature Review

Patrícia Cristina Moser* Hermano Perrelli de Moura
Centro de Informática, Universidade Federal de Pernambuco, University City, Recife 50.740-560, Brazil
* E-mail of the corresponding author: pcm3@cin.ufpe.br

Abstract
Human factors are fundamental to software development, hence the need to understand people-oriented maturity models in development teams. This article aims to identify the maturity models for people management in software development teams cited in the literature, in order to identify evidence about their use, benefits and limitations, and the human aspects involved. A systematic literature review (SLR) was carried out, where automatic searches were done in four search engines, congresses in the area of human factors within software engineering, in addition to a manual search. Evidence indicates that there are numerous models aimed at developing people in development teams, but few are applied. Models based on observation and informal discussion were found, as well as non-validated models, indicating the scarcity of models applicable to people in software engineering. However, complete but complex models were also found, indicating that a more transparent, dynamic and simple process is needed for people's development. In the observed human factors, emphasis is placed on communication, collaboration, knowledge, learning, self-management, motivation and skills in general.
Keywords: Maturity models; Human Factors; Software Development Teams; Teams; People
DOI: 10.7176/RHSS/12-14-02
Publication date: July 31st 2022

1. Introduction

The growth in the use of software products caused a greater demand for the quality of these items, inducing developers to follow some standards in the development of these products [1]. In this search for a quality standard with the created artifacts, software supplier companies began to adapt and certify their processes within standards taken as reference, such as the maturity models aimed at software development. As an example, we can mention the CMMI (Capability Maturity Model Integration), considered an international standard [2, 3] in software development process and quality management. The goal is for the quality of the software to be directly linked to the quality of the process used to develop it.

In parallel with this approach, it is necessary to understand that software development is an intellectual activity that depends on people, who usually build teams and work together. Thus, it is impossible to exclude human factors during software development because software is developed by people and for them [4]. Moe and Dingsoyr [5] comment that software development is complex, especially due to the interference of human issues in the final result. According to the authors, the effectiveness of software construction depends on the team's performance, which is influenced by interpersonal relationships, motivation, satisfaction, skills, competencies, communication, and teamwork. Thus, such aspects, known as human aspects or human factors, have become a critical factor for the success of software projects. [6].

Whether in traditional or agile environments, problems related to human aspects are increasingly perceived in the software development environment [7]. Recent studies [8]; [9]; [10] point out that problems identified in software development may be related to these aspects. The human side of the teams is still a challenge, and building these teams takes time and resources [11] and requires more soft skills [12] and experience [13].

Unsuitable people management is one of the most significant contributors to failure in software project. Software development depends on skills, motivation and interaction of people throughout the project. Without good people management, the project outcome is generally inadequate [14], [15]. Thus, people management is of the utmost relevance in software projects, and can be seen as one of the new trends that are emerging in management of human resources within organizations.

So, in addition to the search for standards for software development, companies also started to look for standards for the development of people in development companies. The goals of these models are to attract, train, organize, motivate and retain talent within the organization, both in traditional and agile environments. As for the use in traditional environments, despite these models being very well-structured, the formality, sometimes excessive, has made the adoption and the continuous improvement of its processes a complex task [4]; [16]; [17]. In agile environments, the existing maturity models are more oriented to product development, dealing with the matter of people in a more implicit way [18]; [19]; [20], requiring reformulation.

Therefore, the aim of this study is to identify the maturity models for people management in software development teams mentioned in the literature, in order to identify evidence about their use, benefits and

limitations and human aspects involved.

This paper is organized as follows: Section 2 presents the background of this study; section 3 addresses the applied methodology; Section 4 shows the obtained results; section 5 brings forth the discussions for this study and, finally, the conclusions are presented in Section 6.

2. Background

The background that supports this study is based on three distinct themes: Human Factors in Software Engineering, Software Development Teams, and Maturity Models for People

2.1 Human Factors in Software Engineering

The intangible nature of a software has made it a product difficult to create successfully. A close examination on the reasons for significant software system failures shows that several of these reasons are related to human problems [21].

However, such questions remain a neglected area of research, and the possible reasons could be the complex relationships between human psychology and software development processes, lack of awareness of the impact of human factors on software engineering, and, possibly, the lack of confidence in empirical studies on human factors (21).

Despite this scenario, it is important to highlight the efforts being made to reduce this oversight. The academy, for example, presents several national and international forums that seek to analyze human factors and how they can impact software development: Workshop on Social, Human, and Economic Aspects of Software (WASHES), Brazilian Symposium on Information Systems (SBSI), Brazilian Symposium on Collaborative Systems (SBSC), Brazilian Symposium on Human Factors in Computational Systems (IHC), Revista Brasileira Systems (iSys), International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), and Computers in Human Behavior Journal.

The realization of the socio-technical nature of software engineering is not new. Since the early 1970s, researchers have sought to look at classic software development problems, such as delays, budget overruns, and customer dissatisfaction, with a focus on people and their involvement in development activities. However, the milestone of social research in software engineering came about with the work of Gerald Weinberg [22] and Bem Shneiderman [23]. Weinberd (1998) demonstrated that software production is more than technology, it is a social activity, and that these elements are interrelated. For Shneiderman (1980), software psychology, which studies human factors in computing and information systems, addresses a range of issues that include environmental, managerial and personality factors that affect the performance of software teams. By the end of the 1980s, the work of De Marco and Lister [24] explained several factors that affect the performance of software teams, with emphasis on the interaction between the personality of the components, the work environment, the composition (diversity, complementarity etc.) and the organization (structure, roles, relationships etc.) of the teams.

In this scenario, large organizations and companies are now realizing the relevance of the physical, emotional and psychological well-being of their employees and the mutual benefits provided to the team. Studies show that human resources interfere in the development of a project and can create a competitive advantage in the market, depending on the motivation of those involved [7]. Work collaboration during project development can be crucial to the ultimate success or failure of the project. Thus, the investment in human resources and the understanding of the employees' culture should also be used as a management strategy, aiming at efficiency and productivity [9].

In the understanding of Cockburn [25], the human factor in software development influences team formation, with emphasis on the interaction of people and individuals [58]. Human factors such as communication, cordiality, talent and skills are part of software development [26], interacting with the organizational structure. Furthermore, these factors intertwine in the group of individuals when working together as a team.

Within this context, Silva et al. [27] carried out a systematic mapping of the literature to investigate the influence of human, technical and cultural factors on agile software development projects. The results indicate that the human aspects are part of the ten critical success factors in agile projects.

Bullen [28] defines the critical success factors as investments in areas that ensure the good performance and competitiveness of organizations. According to Nielsen [29], critical success factors are areas of the project that must work perfectly, in order to not compromise the result and the quality of the implementation. Moreover, human and organizational factors, such as personal characteristics and the culture of an organization, are factors that act directly on the results obtained in the short term, and directly influence the productivity of software development teams [30].

Among the critical success factors found in the study by Silva [27], the following stand out: adoption of agile techniques; support/commitment from senior management; meeting the specified requirements; appropriate

project schedule; appropriate project cost; clear definition of business processes; team composed of specialists; experienced project manager; active customer participation in the project and timely decision-making. Furthermore, the study consolidates existing concepts about human factors in the development, execution, and success of agile software.

Capretz and Ahmed [9] mapped social skills and psychological traits that interfere in life cycle of the software regarding the personality aspect. These authors believe that personality traits are more suitable for the success of a project. Mourmant and Gallivan [31] believe that the profile of IT professionals has a direct impact on job satisfaction and job changes because individual profiles are decisive in choosing work processes, which also guarantees turnover, of functions within a professional environment.

Faced with this reality, companies need to be prepared to develop the skills of their employees as an option to boost their performances.

2.2 Software Development Teams

Teams are a fundamental factor in the social and organizational sphere and are subject of studies since the late 1920s [32]. In recent decades, given the recognition of the importance of teamwork for the success of organizations, this has become one of the main topics of academic work, focusing on different organizational contexts, mostly seeking to understand how to build high performance teams. Therefore, it is important to present a historical context regarding this topic.

Within the arsenal of knowledge already produced on the subject of teams, it is necessary to have an understanding of the concept of team that used in this work. Over the years, the term “team” has come to be used indiscriminately by many authors, and it is common to find the terms “team” and “group” referring to the same concept, while others tend to make distinctions about them.

Hackman [33] uses the terms “group” and “team” interchangeably and considers the group as a social system with the following characteristics: it is recognized as an entity by its members and by non-members who relate to it; its members have some degree of interdependence; there is a clear differentiation of roles and duties among its members. Guzzo [34] shares this same understanding, considering the terms “group” and “team” as synonymous, when related to the context of an organization.

Katzenbach and Smith [35] considers “group” and “team” as different concepts and raised the main points that differ the concept of work groups from the concept of teams, from the point of view of final performance: work groups share knowledge and understanding about determining subject matter, and decision-making is geared towards helping each other achieve their individual goals. A team is a set of people with complementary skills, committed to the same purpose, sharing goals and objectives, and holding each other accountable for their final performance. For them, the essence of a team is the common commitment to the achievement of goals and satisfactory performance.

In a more comprehensive study, Kozlowski and Ilgen [36] synthesized results of researches on teamwork. They arrived at the definition that a team can (a) be formed by two or more individuals; (b) socially interact; (c) have one or more common objectives; (d) jointly perform organizationally relevant tasks; (e) have interdependencies regarding workflow, objectives and results; (f) have different roles and responsibilities; and (g) be involved in an overarching organizational system, with boundaries and links to a broader context in a task-oriented environment.

Over the years, the growing and constant technological evolution has led to the emergence of new types of teams, including software development teams. Based on his experience with this type of team, Jr [37] highlighted in his work his mistakes and successes in the management of a software team — it is considered the first work to deal with the human factor in software engineering. Years later, Demarco [24] studied software teams, with an emphasis on the human factor, within a universe that is still so concerned with the hardware and software of software engineering applications.

Since then, many other authors have started to study teams in the context of software engineering. According to Lettice and McCracken [38], between 1995 and 2006, these studies doubled in number and covered the following themes: team performance, system of rewards and compensation for its members, team and teamwork in Information Technology (IT), team composition, leadership and project management, types of teams.

The importance of teamwork in software development is undeniable. In traditional development, still in the 2000s, the study by Faraj and Sproull [39] showed a strong relationship between experience management and team performance. Since then, some studies have analyzed teamwork using team performance models in agile development, such as the one found in Moe, Dingsøyr, and Dybå [11]. Pikkarainen et al. [40] focused on how agile development methods improve communication and stated that Scrum and XP practices improve formal and informal communication. A survey of success factors on agile development found that team capability was one factor [41]. Maruping et al. [42] demonstrated that XP practices of collective ownership of codes and coding standards could lead to an increase in the technical quality of software products. Sharpe and Robinson [43]

described how agile development teams enable collaboration, coordination, and communication.

Dingsøyr and Dybå [44], focusing on human and cooperative aspects in software development teams, discuss the use of team effectiveness models for better theoretical understanding for future teamwork studies. The authors describe the pros and cons of these models and discuss priorities for future studies on team effectiveness in software development.

Lindsjorn et al. [45] studied the quality of work in agile software development teams — in terms of performance, learning, and job satisfaction — and whether this effect differs from the effect of traditional software teams. Despite claims about the importance of teamwork in agile teams, the study did not find a higher quality of teamwork than similar research on traditional teams.

Most current agile development methods argue that teams should manage themselves. Moe, Dingsøyr, and Dybå [11] studied the challenges faced by these teams, particularly in teams using Scrum. They identified that the transition from individual work to self-managed teams requires a reorientation of developers and management. Making such changes takes time and resources, but it is a prerequisite for the success of any kind of agile self-management method.

Barbosa et al. [46] carried out a study that aimed to understand the perception of interdependence at work and its relationship with trust among software engineers who work with agile methods. The study shows a direct connection between factors such as feelings, maturity, and technical level with confidence and their impact on interdependence. Preliminary findings show that trust is an important psychological factor that can be influenced by the maturity level and technical level of an individual.

Although much has already been discovered on the various facets of teamwork, many gaps are found in the literature and in the questions in need of answers.

2.3 Maturity Models for People

A maturity model contains the crucial process elements for one or more areas of interest, describing an evolutionary improvement path from informal processes to mature processes, with improved quality and effectiveness [47]. They are divided into categories, levels or stages, which attest, in turn, to the degree of evolution in which an organization is at a given moment.

The idea of maturity is associated with the concept of process stability, that is, stability occurs when the organization is at a level of excellence with its projects, in which its processes are stable and free from variations, their executions in a consistently homogeneous way [47].

In this context, the need arises to establish maturity models that provide bases for identifying the maturity present in organizations. Such models are responsible for numerically quantifying the maturity of an organization, and, based on the results obtained, it is possible to develop processes by applying the best practices, which in turn conduct the continuous development of processes [48].

By the end of the assessment of the maturity level of the organization and knowing at what level of maturity it is, it is possible to outline a strategy with the objective of raising the levels of knowledge and management in dealing with its projects. Currently, discussions about project management maturity are constant in the literature, so to better understand what happens within organizations.

The increase in the adoption rate of maturity models can be attributed to the success of these methodologies, and studies have documented this phenomenon [44]. Several maturity models currently aim to improve the software process, focusing on optimizing time, cost, and quality of management and engineering practices in software development organizations [49]. This is the case of the Capability Maturity Model (CMM).

While companies used the CMM for software development, some felt difficulties when implementing the model to improve processes, regarding the dimension of people management in organizations. It was found that improvements in software development processes resulted in some changes in how to manage people, and the CMM did not foresee such changes.

Seeking to solve this need, the most complete and ambitious theoretical proposal was developed by Software Engineering Institute (SEI), from Carnegie-Mellon University. Sponsored by the aforementioned institution, a group of researchers in the area of Software Engineering and Human Resource Management developed a set of models to help organizations manage and grow their intellectual capital [50]. The People Capability Maturity Model (P-CMM), based on the most successful practices in the field of human resources, knowledge management, and organizational development, is a guide for the establishment of conduct and standards to achieve the continuous improvement of the work of the organization's workforce. The ultimate goal is to improve the organizations' ability to attract, train, motivate, organize and retain their human resources.

The P-CMM has a structure with five maturity levels (Figure 1), enabling the establishment and evolution of practices for attracting, improving, motivating, and retaining people [50]. Each of the model's five maturity levels represents a different level of organizational capability to manage and develop the workforce. The levels are as follows: Level 1 – Initial; Level 2 – Managed; Level 3 – Defined; Level 4 – Predictable and Level 5 – Optimized.

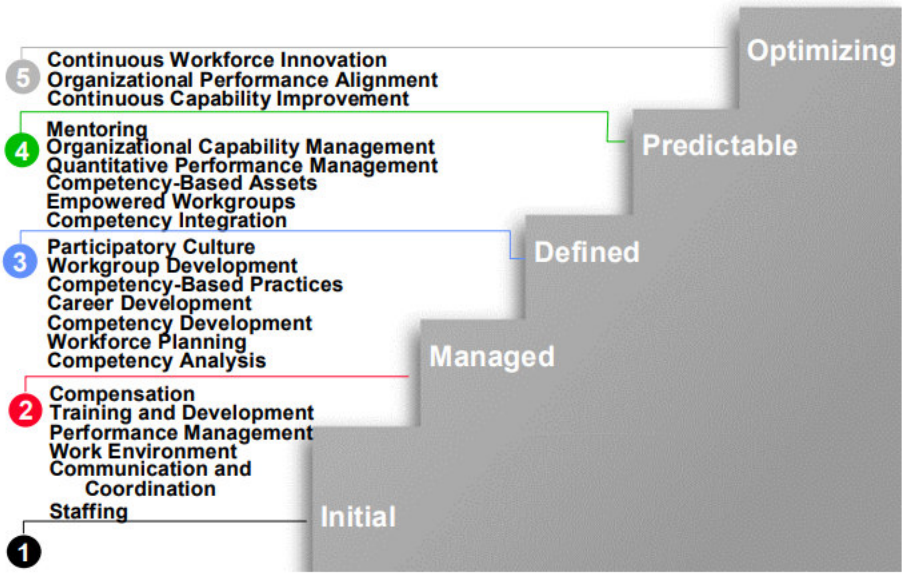

Figure 1. Five P-CMM maturity levels

Each maturity level, except level 1, is composed of a set of process areas (PA's), each representing an important organizational process for managing people. Each PA is composed of goals — requirements — that an organization must satisfy to establish the ability to act on the workforce's capability. A level is reached only when all the goals of each PA are satisfied. By adopting and institutionalizing the implementation practices of each process area of the P-CMM, the goals of the process areas are achieved. As this occurs, the organization reaches a higher level of maturity, which positively influences organizational capability [50].

While P-CMM was primarily developed for software companies, it can be implemented for any organization that cares about its workforce. According to Gamal [51], the Indian Confederation of Industry has developed a program to implement P-CMM in all industries in India. The SEI, the provider of P-CMM, stated that the kinds of organization using P-CMM include: Business Process Outsourcing, Hospitality, Construction, Insurance, US Government Agencies, Banks, Financial Services, Information Technology, Consulting, Defense Contracting, Pharmaceuticals, US Department of Defense, Software Development and Information Management.

Due to a need to map traditional concepts to agile teams, several models that call themselves agile maturity models have emerged, seeking to address the needs and characteristics inherent to this methodology and incorporate them into a maturity model, like the models [19] and [20].

More details on the maturity models for people are covered in section 4 (Results). The following section will address the method used in this study.

3. Materials and Methods

This study adopts the guidelines for performing systematic literature reviews in software engineering [52], and its phases can be observed in the following sections.

3.1. Research Questions

According to Gil [53], all research begins with some kind of problem or matter, and it must be formulated as a question. To understand the phenomenon studied, the following research questions were developed:

QP1: What are the maturity models aimed at managing people in software development teams mentioned in the literature? (What are the main features of these models? Are they in use? Are they validated?)

QP2: What are the benefits and limitations of the models identified in the literature? (What are the best practices adopted for people management?)

QP3: What factors related to human factors are addressed in these models?

The answers to these research questions are presented in Section 5.

3.2 Search Strategy

The process to define the search strategy is divided into: defining the search terms; creating the search string; listing the search sources; reporting the selection criteria; and describing the study selection process. The rigor adopted in the search process is a factor that distinguishes SLRs from traditional reviews [52].

One of the search strategies for primary studies can be automatic and/or manual [54]. This research initially carried out an automatic search using search engines, followed by a manual search ranging from 2001 to 2020.

This range was due to the P-CMM maturity model — one of the references for this study —, launched in 2001. The selection process for this study will be described in Section 3.4.

3.3 Study Selection Criteria

The methods proposed by Kitchenham and Charters [52] guided the selection of studies. To reduce the likelihood of bias, selection criteria should be decided during protocol definition — although they could be refined during the search process. The inclusion (IC) and exclusion (EC) criteria are based on the research questions and are as follows:

IC1: Studies related to people management in a software development environment;
IC2: Studies published between 2001 and 2020;
EC1: Studies out of scope;
EC2: Studies that do not present data in scientific format;
EC3: Studies not written in English;
EC4: Lecture summaries, tutorials, presentation slides, or incomplete documents;
EC5: Books, theses, or dissertations;
EC6: Studies that are not accessible;
EC7: Secondary or tertiary studies;
EC8: Studies that have only lessons learned or experience reports;
EC9: Studies that are merely based on expert opinion.

3.4 Selection process

Initially, an automatic search was performed. For this type of search, Dybå and Dingsøyr [55] recommend the search terms to be comprehensive so to include the most significant number of studies and avoid missing relevant searches. Therefore, we tried to group the most significant number of words that could be related to the purpose of this review. The search terms, their synonyms, or related words are presented in Table 1.

Table 1. Search Terms

Search Terms	Synonyms or Related Words
Maturity Model	Framework
People Management	Human Factors, Human Aspects, People, Team, Group

With the search terms defined, the search string was constructed. The Boolean OR was used to incorporate the alternative spellings for the main terms and the Boolean AND to compose the junctions: “Maturity Model OR Framework AND People Management OR Human Factors OR Human Aspects OR People OR Team OR Group”.

The automatic search sources selected for this review are the main electronic databases of relevance [56]. Those used for this study are in Table 2.

Table 2. Automatic Search Sources

Electronic Database	Website
Compendex (Engineering Village)	http://www.engineeringvillage.com
ACM	http://dl.acm.org
Scopus – Elsevier	http://www.scopus.com
Web of science	https://apps.webofknowledge.com

In order to complement the automatic search, a manual search was performed in the events about human factors in software engineering: Workshop on Cooperative and Human Aspects of Software Engineering (CHASE); International Conference on Software Engineering (ICSE) and Conference on Human Factors in Computing Systems (CHI).

3.5 Primary Studies Selection Process

This phase aims to identify the central primary studies of this study. Automatic searches were carried out, applying the search string presented in Table 2, in Scopus (850 studies), Compendex (1017 studies), ACM (29 studies), and Web of Science (277 studies). After the automatic ones, a manual search was performed (15 studies). With the sum of the bases, 2188 studies were obtained. Table 3 illustrates the primary study selection process.

Table 3. Primary Studies Selection Process

	ACM	Compendex	Scopus	Web of Science	Manual	Total
Found	29	1017	850	277	15	2188
Duplicates	19	631	333	155	1	1139
Total valid	10	386	517	122	14	1049
After phase 2 (title, abstract, keyword)	1	12	5	1	13	32

Of the 2188 studies, 1139 were duplicates, leaving 1049 studies. After the first reading (title, abstract, and keywords) and with the inclusion and exclusion criteria, 32 papers remained. A new refinement was applied — which acted as a filter —, with the researchers' evaluation for 32 studies. Each researcher read the study assigning 0, 0.5, or 1 as a grade so that the maximum grade for that particular study should be 2. Studies with a grade between 1.5 and 2.0 were kept, leaving 19 at the end.

Figure 2 shows the Preferred Reporting Items for Systematic Reviews and Meta-Analysis (PRISMA) diagram [57] for the search methodology. The diagram depicts the number of records identified, included, and excluded. To manage the studies used in this SLR, the Parsifal tool was used.

It is important to indicate that the process of selecting primary studies may contain biases, not only in the identification and screening phase, but also in the phase of quality analysis of the studies.

Publication bias refers to the matter that positive results are more likely to be published than negative results. The concept of positive or negative results sometimes depends on the researcher's point of view.

Publication bias is even more of a problem when methods/techniques are sponsored by influential groups in the software industry. For example, the US MoD is an extremely important and influential organization which sponsored the development of the Capability Maturity Model and used its influence to encourage the industry to adopt CMM. In such circumstances, few companies would want to publish negative results — and there is a strong incentive to publish papers supporting the new method/technique [52].

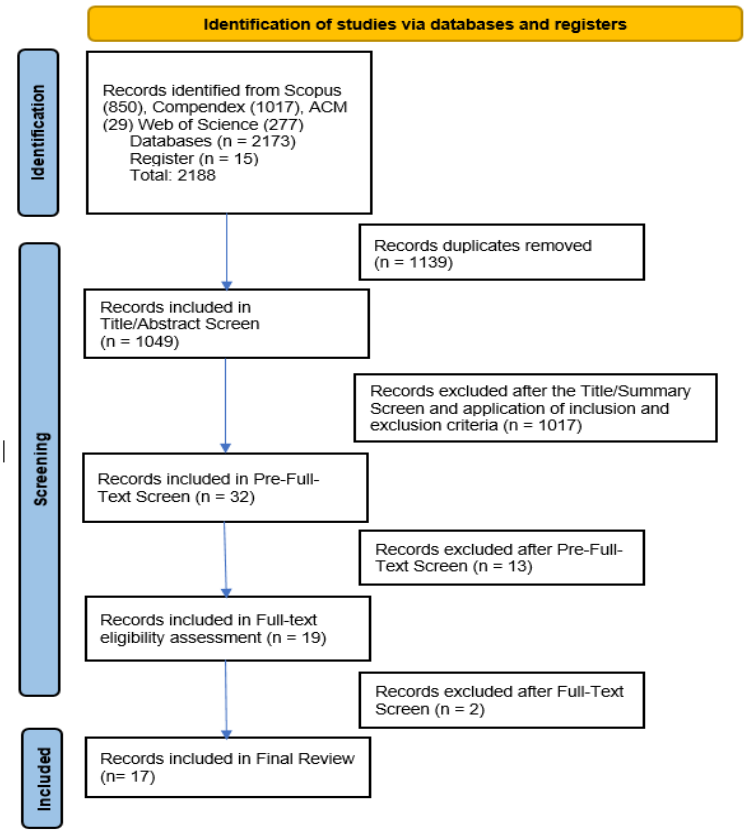

Figure 2. Search Strategy

Publication bias can lead to systematic bias in systematic reviews unless special efforts are made to address this issue. One such effort is the search for papers in conference proceedings. But the very search and inclusion of studies published in conferences can be considered a bias, which is the case of this study.

3.6 Study Quality Assessment

In addition to general inclusion/exclusion criteria, the assessment of the "quality" of primary studies is considered critical [52]. For this study, ten quality criteria were considered:

Is there an adequate description of the context in which the research was carried out?
Is the research objective clearly described?
Do the authors present conclusions clearly?
Do the authors clearly describe the methodology used?
Does the study clearly identify a maturity model/framework/ that can be used for people?
Does the study signal which factors related to human aspects are being investigated?
Does the study clearly identify the limitations of the cited models?
Does the study clearly identify the benefits of the cited models?
Are threats to validity mentioned?
Do the authors describe the limitations of the study?

The identified studies were analyzed using a checklist defined with the above-mentioned quality criteria. For the responses, sets were defined, considering 1 for "yes"; 0.5 for "partially" and 0 for "no". Studies that scored above 7 were considered.

Checklists are also developed by considering bias and validity problems that can occur at the different stages in an empirical study: Design, Conduct, Analysis and Conclusions. And it was no different with this study.

There are many quality checklists for different types of empirical study already published. The medical guidelines all provide checklists aimed at assisting the quality assessment undertaken during a systematic literature review as do Fink [58] and Petticrew & Roberts [59]. However, each source identifies a slightly different set of questions and there is no standard agreed set of questions. Furthermore, Kitchenham and Charters [52] mention that the lists can be tuned for studies in software engineering.

The criteria for a quality checklist are not exhaustive [52] and may depend on the context of the study itself, where the researcher can select the most appropriate quality assessment questions for their specific research questions. Kitchenham and Charters suggest a checklist for qualitative studies in [33]. Considering the context of this study, only ten of the eighteen suggested criteria were considered, which could be considered a bias.

It is noteworthy that, as the aim of the article is to identify maturity models for people management in software development teams mentioned in the literature, in order to identify their use, the number of pages of the studies was not one of the evaluation criteria. The authors defined the strategy to include all identified studies, even those not validated, not evaluated or those that still need more evidence. This strategy was used as a way to signal incipient models.

3.7 Data extraction

Data extraction was performed in a Microsoft Excel® spreadsheet, where the information extracted from the studies was defined. This information can be seen in Table 4.

Table 4. Data Extraction Table

ID	Unique study identifier
Title	Study title
Authors	Study authors
Year	Year of publication
Context	Study Context
Problem	The problem addressed in the study
Maturity Model/Framework	The maturity model or framework addressed in the study
Features	Characteristics of maturity/framework models
Benefits	Benefits of the identified model/framework
Limitations	Limitations of the identified model/framework
Best Practices	Best practices implemented by the framework model
Human Factors	Observed human factors
Considerations	Considerations that emphasize some evidence

3.8 Data synthesis

Data synthesis involves gathering and summarizing the results of included primary studies. In this review, the qualitative approach was used, and the line of argumentative synthesis technique was used [60], where issues of importance are identified, and each study's approach to each question is documented and tabulated.

The line of argumentative synthesis approach is used when researchers are concerned about what they can infer about a topic as a whole from a set of selective studies that look at a piece of the issue. This analysis is a two-part one. First, the individual studies are analyzed, then an attempt is made to analyze the set of studies as a whole. Issues of importance are identified and the approach to each issue taken by each study is documented and tabulated.

4. Results

Seventeen studies published between 2001 and 2020 were analyzed. Table 5 summarizes these studies, ordered by an identifier (ID), and followed by information on the author and year of publication.

Table 5. Selected Studies

ID	Author(s)	Year
M1793*	Curtis, B., Hefley, B., & Miller	2001
A0247*	Srinivasa, G., & Ganesan, P	2002
M1789	Packlick, J.	2007
M1790	Sidky, A., Arthur, J., & Bohner, S.	2007
A1782*	Gamal, A. M.	2008
M1791	Qumer, A., & Henderson-Sellers, B.	2008
M1792	Patel, C., & Ramachandran, M.	2009
A0171*	Lu, X., Xu, D., & Han, J.	2010
A1854	Richardson, I., Casey, V., McCaffery, F., Burton, J., & Beecham, S.	2012
A2116*	Colomo-Palacios, R., Casado-Lumbreras, C., Soto-Acosta, P., Misra, S., & García-Peñalvo, F. J.	2012
M1786	Horita, F. E., & Barros, R. M.	2012
M1788	Fontana, R. M., Meyer Jr, V., Reinehr, S., & Malucelli, A.	2015
A1902	Friedrich, R., Computing, C. I. T., Bleimann, U., Sengel, I., & Walsch, P.	2016
A1783*	Zhang, C.	2016
M1778	Marsicano, G., Pereira, D. V., da Silva, F. Q., & França, C.	2017
M0001*	Dutta, K., Baruah, N., & Baruah, J. B.	2018
A1763	Walter, B., Marović, B., Garnizov, I., Wolski, M., & Todosijevic, A.	2020

* Studies marked with an asterisk (*) refer to the same maturity model, applied to different contexts

From these studies, we sought to answer the following research questions: QP1: What are the maturity models aimed at managing people in software development teams mentioned in the literature?; QP2: What are the benefits and limitations of the models identified in the literature?; QP3: What factors related to human factors are being addressed in these models?

The answers to these questions are presented below and divided into sections to further favor the understanding. At the end, a table containing the main characteristics of the models found will be presented.

Eleven maturity models focused on people management used by software development companies were identified. Seven (7) of the seventeen (17) studies identified mention the same model, but since they apply to different contexts, they were also considered for this review. Table 6 identifies the maturity models and their authors, with the respective years.

Table 6. Maturity Models and authors

Maturity Model	Author(s)	Year
Agile Adoption Framework	Sidky, A., Arthur, J., & Bohner, S.	2007
AGILE Maturity Map (AMM)	Packlick, J.	2007
Agile Software Solution Framework (ASSF)	Qumer, A., & Henderson-Sellers, B.	2008
Agile Maturity Model (AMM)	Patel, C., & Ramachandran, M.	2009
Global Teaming (GT)	Richardson, I., Casey, V., McCaffery, F., Burton, J., & Beecham, S.	2012
GAIA Human Resources	Horita, F. E., & Barros, R. M.	2012
Progressive Outcomes Framework	Fontana, R. M., Reinehr, S., & Malucelli, A.	2015
Virtual Team Maturity Model (VTMM)	Friedrich, R., Computing, C. I. T., Bleimann, U., Sengel, I., & Walsch, P.	2016
Software Engineers Team Maturity	Marsicano, G., Pereira, D. V., da Silva, F. Q., & França, C.	2017
G'EANT Software Maturity Model (GSMM)	Walter, B., Marović, B., Garnizov, I., Wolski, M., & Todosijevic, A.	2020
People Capability Maturity Model (People-CMM)	Curtis, B., Hefley, B., & Miller	2001
	Srinivasa, G., & Ganesan, P	2002
	Gamal, A. M.	2008
	Lu, X., Xu, D., & Han, J.	2010
	Colomo-Palacios, R., Casado-Lumbreras, C., Soto-Acosta, P., Misra, S., & García-Peñalvo, F. J.	2012
	Zhang, C.	2016
	Dutta, K., Baruah, N., & Baruah, J. B.	2018

In the following sections, the models found will be presented.

4.1 Agile Maturity Map (AMM)

The model presented by Packlick [61] is based on a single experience in Saber Airline Solutions, where the author describes an approach based on the observation of real teams. The proposal is to use a goal-oriented approach since, according to the author, the teams were more motivated to discover their ways of doing their jobs.

The AMM comprises five maturity levels representing different stages of learning that an agile team must fulfill (Awareness, Transformation, Breakthrough, Optimizing, and Mentoring). The objectives are related to the acronym AGILE: Acceptance Criteria; Green-Bar Tests and Builds; Iterative Planning, Learning and Adapting; and Engineering Excellence. Each goal is detailed as a user story describing what teams must accomplish, and the agile roadmap is defined by the intersection of those objectives and maturity levels.

The experience with the Agile Maturity Map is that the improvements resulting from a goal-oriented approach for a large mature agile organization are much faster and more sustainable than using practice-oriented approaches that are more traditional. The authors found that team members value and respect a process that works and perform more when they participate in its development, rather than having it imposed on them. Teams can overcome biases against specific practices by focusing on goals. As a result, team members understand the logic behind the practice much better.

Mentoring, focusing on learning, is the human aspect best emphasized in this model, including being one of the five maturity levels.

4.2 Agile Adoption Framework

The model proposed by Sidky et al. [19] is part of a whole framework to guide and assist in the adoption of Agile. The framework comprises a Sidky Agile Measurement Index (SAMI) and a four-step process that guides and supports organizations' agile adoption efforts.

More specifically, SAMI encompasses five agile levels that are used to identify the agile potential of projects and organizations (Collaborative, Evolving, Effective, Adaptive and Comprehensive), where they are not associated with any business value. Instead, they are based on the qualities and values of agility.

Agile levels can be defined as a set of agile practices that are related and, when adopted according to agile principles, result in significant improvements in the software development process, leading to the realization of a core value of agility. Thus, agile levels enumerate the different degrees of agility possible for a project or organization. The agile potential of a project or organization is expressed in terms of the highest agile level it can achieve.

The four-step process, on the other hand, helps to determine (a) whether organizations are ready for agile adoption; and (b) if guided by their potential, what set of agile practices can and should be introduced. This four-step assessment process is the “backbone” of the framework. First, it provides an assessment component that helps to determine if (or when) an organization is ready to move towards agility. And second, the process guides and assists the agile coach in the process of identifying which agile practices the organization should adopt. The model was submitted for evaluation by the agile community. Some of the evaluators suggested that agility levels could be prioritized according to the business values an organization hopes to realize. Other evaluators suggested a reorganization of agile practices based on experimental successes. In other words, they argue that the types of projects and the experiences gained from past adoption efforts can and should serve as a basis for formulating a better arrangement of practices within agile levels. Based on the rationalizations above, an adaptive measurement index is both desirable and beneficial.

Feedback from model evaluators and subsequent insight led the authors to recognize the usefulness and need for flexibility in adapting the SAMI to meet (a) individual experiences and, perhaps, (b) business objectives.

Within the human factors observed, the article addresses trust in people and the interaction between them. SAMI clarifies the following factors: Ideal agile physical setup, self-organizing teams, frequent face-to-face communication, collaborative teams, and empowered and motivated teams.

4.3 Agile Software Solution Framework (ASSF) e Agile Adoption and Improvement Model (AAIM)

Qumer and Henderson-Sellers [20] describe several approaches to help the transition from traditional to agile. The Agile Software Solution Framework (ASSF) provides a general context for exploring agile methods. It is tied to the business aspects of software development so that business value and agile process are well aligned. All elements of ASSF are intended to guide the behavior of self-organizing agile teams in large and complex project development environments. It includes templates to keep development and teams aligned and driven with a shared mental model, which is a distinguishing feature of ASSF, compared to traditional agile methods.

The ASSF, in relation to the agile conceptual aspect, encompasses a series of elements. Knowledge, Governance, Method Core are all linked to the Business element through the business-agile alignment bridge or business value. This bridge has an impact on governance, which in turn configures an agile method of software

development (building and application), in terms of the business value it delivers.

The Agile Toolkit (another element of the framework) provides an application of these ideas in practice (building or adapting software processes), while the embedded analytical tool (4-DAT) is especially used as a quality assessment measure to assess the degree of agility in software development practices.

Furthermore, the Method Core represents the main aspects of an agile method: agility, people, process, product and tools, which can be combined to build agile methods specific to each situation, in order to achieve the desired business value.

While ASSF elements provide new and highly useful information about the agility characteristics of process elements, none of them support the methodology of process adoption. To fill this gap, the authors conceived, based on industry analysis and a grounded theory research methodology, the Agile Adoption and Improvement Model (AAIM), which was developed to assist in the introduction, evaluation, and improvement of agile software development (processes or methods) in a software development organization.

The model is composed of three agile blocks (prompt, crux and apex) in which six levels are embedded. The first block — Prompt — has level 1) Agile Childhood. The second block — Crux — comprises levels 2) Agile Initial, 3) Agile Realization and 4) Agile Value. The last block — Apex — has the last two levels: 5) Agile Smart and 6) Agile Progress. AAIM Level 1 focuses on introducing basic agile properties (speed, flexibility, and responsiveness). In AAIM Level 2, the focus is on enabling communication and collaboration. The next, AAIM Level 3 is represented by the use of executable artifacts and minimal documentation. At AAIM Level 4, the agile foundation is established and the focus is on valuing people and not ignoring tools and processes. AAIM Level 5 focuses on learning, and lastly, AAIM Level 6 establishes a lean environment, with high quality and minimal resources, sustaining agility). In each block, the degree of agility of an agile process is measured quantitatively using the agile measurement modeling approach (one of the ASSF tools).

As main features, there are flexibility, speed, responsiveness, communication orientation, people orientation, executable artifacts, learning, and leanness. Through this model, different organizations can adopt agile practices in different ways. AAIM lays the groundwork for implementing agility, and the software development organization can adapt or customize AAIM according to its local organizational structure, culture, size, and development environment.

On the plus side, AAIM helps assess how and how well an agile process/method is being followed within a software development organization. Also, it helps to assess the current agile level of an organization and guides measurement and evaluation (quantitatively) ofe the degree of agility of a software development process. Furthermore, it combines the concepts of theory and practice (data and feedback from researchers and the software industry) and has in its phases 5 and 6 specific guidelines for people in this software development environment.

However, regarding the use of the model, the authors identified that the success of the transition to an agile environment depends substantially on the leadership role of the CIO and the executive, since the model has many elements. A step-by-step approach can be considered reasonable for a gradual and successful transition.

4.4 Agile Maturity Model (AMM)

Patel and Ramachandran [62] presented a proposal based on the CMMI-DEV framework, but the process and practice areas focus on agile principles. For the authors, there was a need for a model of software process improvement to adapt to agile software development environments, identify and define agile practices for each maturity level, and relate the problem of agile practices to the improvement objectives of those practices.

The Agile Maturity Model (AMM) has five maturity levels and shows how agile software development practices mature based on agile principles and practices (Initial, Explored, Defined, Improved, and Mature). Each level has a predefined goal to help the professional or organization focus on their improvement activities. Also, the gain in maturity, as in CMMI-DEV, is related to the increase in the definition and control of processes through metrics based on agile practices.

At the Initial level, the development environment is unstable. The next level, Explored, means implementing project planning; story card-driven development; on-site customer; and introduction to test-driven development. The third level, called Defined, focuses on customer satisfaction;, communication improvements; software quality; and improving coding practices. At the fourth level, Improved, the goals are: to measure the software process; to be able to empower teams and rewards; to implement project management; to assess project risks and work with simplicity and with no prolonged work. The highest-maturity team, at the Mature level, continually improves the software process; manages uncertainties; fine-tunes project performance; and prevents software defects.

It is worth noting that the AMM model links agile software development practices to maturity levels but is not an exhaustive representation of agile software development practices. The model is based on agile software development's values, practices, and principles and is designed to improve and enhance agile development methodology and drive agile principles and goals.

Regarding the human factors observed in the model, communication, collaboration, staff training, rewards system, staff retention, and people-orientation (with a focus on pair programming) stand out.

However, the conclusions of the model are temporary, where verification and evaluation are still necessary.

4.5 Global Teaming (GT)

Taking the basic structure of Capability Maturity Model Integration (CMMI), Richardson et al. [63] developed Global Teaming (GT), a framework that sets specific goals and sub-practices for Global Software Engineering Teams (GSE). The authors proposed the GT taking into account technical activities and human, social and cultural implications, with factors considered important in the configuration of global software development teams. This framework can be used as a supplement to the CMMI, but can also be used in conjunction with other processes that a global software project team may implement.

The WG has two specific goals (SGs): "Define Global Project Management'' and ''Define Management Between Locations". SG1 represents the practices required at the beginning of the project; while SG2 classifies the practices required when the project is operational. And each SG has Specific Practices (SPs) and sub-practices.

Twenty-five GSE factors were identified, which can be grouped into four major headings: 1. Distance (Communication, Language, Culture, Temporal Issues); 2. Infrastructure (Process Management, Tools, Technical Support, Communication Tools); 3. Management (True Cost, Project Management, Risk Management, Defined Roles and Responsibilities, Team Selection, Effective Partitioning, Skills Management, Knowledge Transfer, Coordination, Visibility, Reporting Requirement, Information Management, Teamness) and 4. Human Factors (Fear, Motivation, Trust, Cooperation).

When implementing GT, tangible results must be achieved in a reasonable period. This is particularly important to sustain the effort required for improvement to occur. Practitioners should therefore find the practical guidelines provided by the GT process area particularly useful as they address key aspects of the GSE and discuss threats, if specific practices are not implemented.

The study by Richardson et al. [63] considers that the proposed framework is in its initial version, and that it is necessary to validate the model. However, the first results indicate that all the practices of the model are relevant for a global software organization.

4.6 GAIA Human Resources

Horita and Barros [64] propose the GAIA Human Resources, a model that gradually increases the quality of people management in software projects. Its main goal is to increase the quality of the software development process through the institutionalization of good practices to improve the management of the human resources available in the organization. In this context, the monitoring and evaluation of human factors, planning of needs and skills, training and performance analysis were important for the proposal of the framework.

GAIA is composed of three main structures: maturity levels (Initial, Repeated, Defined, Managed and Optimized); a set of services; and a Diagnostic Assessment Questionnaire (DAQ). Each framework maturity level consists of a set of services composed of five basic components: Tools and Techniques, Indicators, Workflow, Vocabulary and Models. The Diagnostic Assessment Questionnaire (DAQ) aims to identify the maturity level of the organization. It is a process carried out with stakeholders, and its objective is to identify and define which services can be implemented to meet and solve the identified problems.

Although there is a need to apply the model in new software development projects, so that new factors are highlighted, it is important that improvements are proposed and the framework can be consolidated and improved, needed to define an efficient and transparent process for its implementation.

Additionally, the development of a tool that helps in the application of this model is proposed.

Finally, although the model aims to improve people management, it still needs more evidence to be consolidated better.

4.7 Progressive Outcomes Framework

The aim of the study by Fontana et al. [65] was to investigate how agile teams evolve to maturity. As a result, the Progressive Outcomes Framework was proposed, a model that describes the process of maturing agile software development. It is a framework where people play a central role, ambition is a skill to maturity, and improvement is driven by the results that agile teams seek, rather than pre-written practices.

In a training of process improvement, management should consider an agile software development team for self-organizing software and should focus on training a team rather than implementing control tools.

The authors believe, in comparison with other agile improvement guides available in industry and academia, the Progressive Outcomes Framework to be the innovative approach to maturation process as it brings a tool of context-specific methods.

Likewise, the maturation process of intelligent software development does not seem to be composed of a

linear sequence of practices adoption. In fact, it seems to be a discontinuous process, where the team seeks the results simultaneously for a behavior improvement in the behavior itself, in deliveries, requirements, the final product, and in the relationship with the customer.

In short, how improvements in agile software development can improve team performance.

4.8 Virtual Team Maturity Model (VTMM)

Friedrich and Bleimann [66] developed the Virtual Team Maturity Model (VTMM) based on literature review on virtual teams and team performance. The model has 3 (three) main components: 1. A process model with 11 processes; 2. Key performance indicators (KPIs); and 3. Four maturity levels (undefined, basic, advanced, mastery). The purpose of the model is to assess the level of competence of virtual teamwork in project teams, and propose clear steps for the improvement of team performance.

The model focuses on the internal processes of the project team, which are necessary to compensate for critical factors such as lack of face-to-face interactions, challenges in tacit communication, building trust, providing feedback, establishing work rules, and offer of rewards and recognition.

As VTMM is designed flexibly, in terms of defining levels and processes, the model can be adapted to the needs of the organization. The goal is to implement a pattern. VTMM in its standard version should cover 80% of the requirements for good virtual communication processes for any team.

However, to implement the processes in a virtual team, the team leader needs to understand the processes and define them in conjunction with the team members.

The model was validated through a Delphi process with an expert panel of more than 80 participants, mainly from the IT project management area, and subsequent pilot implementation in a virtual team.

4.9 Software Engineers Team Maturity

In an attempt to understand the concept of mature teams in the context of software engineering, Marsicano et al. [67] proposed a model formed by twelve categories, representing the maturity of the team: 1. Interpersonal understanding; 2. Team identity; 3. Team culture; 4. Intra-team relationship; 5. Communication within the team; 6. Extra-team relationship; 7. Extra-team communication; 8. Team learning; 9. Shared knowledge; 10. Mutual support; 11. Team management; 12. Team Technical process.

Thus, the authors propose the model and reveal three distinct dimensions of team maturity: learning, relational and technical maturity. The model also displays definitions and relationships between dimensions. The authors state that the maturity of the software engineering team is perceived as the balance of these dimensions. When the three dimensions are balanced, they catalyze team performance.

As a result, the authors proposed the following definition for mature software engineering teams: “A mature software engineering team (i) has members that know and are aware of each other, share goals, values, thoughts, and feel that they are long for the team; (ii) has sustainable relationships between members, customers, external leadership and other stakeholders; (iii) promotes collective and individual learning through knowledge share and support; (iv) can adapt its organization and processes to focus on delivery; and (v) generates constant results, meeting all the needs of the interested parties."

Finally, it is important to highlight that some points identified are crucial for the maturity of the team, which are individual maturity, the team's training time and external agents. They were not the focus of the research, but they appeared repeatedly in the data. Therefore, the authors believe it is important to pay attention to them, as they can strengthen or weaken the team's maturity.

The observed human factors are understanding, relationship, communication, learning, culture and knowledge.

The focus of the Software Engineers Team Maturity is the development of people, it does not have any technical aspect in its composition.

4.10 GÉANT

Walter et al. [68] proposed a maturity model designed for the software development teams of GÉANT, a large organization with around 30 software projects and around 20 development teams. Such a model addresses issues related to the geographic distribution of the team, percentage of the dispersed workforce, and parallel involvement in other project priorities. The authors opted for a prescriptive approach based on the collective needs of development teams, emphasizing continuous representation rather than a staged model, providing a more flexible representation in order to allow changes to be facilitated in the future.

The GÉANT model is composed of components of processes, goals, parameters and a scoring system, covering the areas of Requirements Engineering, Design and Implementation, Quality Assurance, Team Organization and Software Maintenance. Each target area (TA) is composed of specific objectives (GS) that capture sub-goals and related activities. As a result, the model identifies 29 Specific Objectives (SGs) grouped into five Target Areas (TAs), which are essential for effective software development.

The implementation of this model, in practice, faces several difficulties. Firstly, the distributed structure of the GÉANT teams promotes their independence and self-organization. This makes it difficult to compare results, but it also makes teams reluctant to be evaluated. Therefore, the authors proposed steps for an evaluation plan to overcome these problems and obtain valuable feedback. Before implementing the maturity model, it needs to be evaluated and approved.

As can be seen, the model is not limited to activities directly related to software development, but also to organizational, communication and human issues. Furthermore, it is a model that is not viable for local teams, since it was specifically designed for the context of companies belonging to the GÉANT project, taking into account the distributed nature of the teams.

4.11 People Capability Maturity Model (P-CMM)

The model proposed by Curtis, Hefley, Miller [50] was mentioned earlier in section 2.3., where its main features were presented. In this section, several authors demonstrate studies on the P-CMM, observed in different contexts. To aid in the understanding, cited studies will be presented in the following subsections.

4.11.1 Srinivasa e Ganesan (2002)

One year after the release of version 2.0 of the P-CMM by the SEI, Srinivasa and Ganesan [69] carried out a study recognizing that the employees of a software company have a strong influence on the quality of their products. The authors formalize the relationship between pair programming, which is one of the techniques suggested by the P-CMM, and the P-CMM process areas, to improve teamwork, communication, and knowledge levels. Specifically, Srinivasa and Ganesan's study outlines the advantages and effects of adopting pair programming if an organization wants to reach a higher level within the P-CMM framework.

This technique improves the work environment, offers alternative learning opportunities, and makes mentoring a part of daily activities. The authors point out that these benefits are achieved with no more than 15% extra development time while achieving a 15% reduction in software defects.

Srinivasa and Ganesan (2002) evidenced the team's delegation of authority and responsibility in the study. The authors conclude that pair programming is a convenient, pleasant, and low-cost technique for managing some of the P-CMM KPAs.

4.11.2 Gamal (2008)

In his study, Gamal [51] mentions that Indian companies are experienced in the implementation of P-CMM level 2 (the level where the standards of training of the workforce are found). Also, the reasons for this comes from the high rates of employee turnover during the late 1990s and the belief that its highly skilled workforce is its greatest natural asset, leading to a system of practices that builds a workforce capable of achieving the performance levels most beneficial to the organization.

The author addresses the importance of the human factor in the software industry, providing the main actions to design and implement a framework for workforce competency management based on the P-CMM Level 3 specifications.

In addition to the benefits achieved by a company certified at level 2, Gamal observes the benefits to be obtained by an organization that obtains the P-CMM level 3. According to the author, this level, if implemented, has a direct positive impact on many essential workforce-related activities, such as retention, training, recruitment, and deployment. And to reach this level, organizations must: 1. Implement competency analysis; 2. Develop a culture of professionalism based on well-known competencies in the workforce; 3. Develop a policy to manage their workforce as a strategic asset; 4. Develop processes for career development and skills development; and 5. Develop a competency-based workforce planning policy. For this, the main steps to implement the workforce training program by PCMM 3, according to Gamal, are the formation of the working group, competency analysis, collection and definition of competency profiles, and skills development plans.

4.11.3 Lu, Xu and Han (2010)

Lu, Xu, and Han [70] discuss the use of P-CMM in the Chinese software industry, specifically in the city of Hangzhou, whose economic aggregate of the software industry ranks 4th in the country. Based on the literature review and the investigation of companies, they studied 6 (six) process areas managed within the P-CMM level 2 model, which compose the set of critical factors that lead to the organizational effectiveness of software companies. The attributes of level 2 of the model are as follows: Human Resources, Communication and Coordination, Work Environment, Performance Management, Training and Development, and Compensation.

The authors surveyed to estimate the maturity of teams in software companies in Hangzhou, and how P-CMM level 2 attributes can influence organizational effectiveness. According to the results obtained, the study identified that software companies in Hangzhou are basically at the medium level of P-CMM level 2. The most implemented attributes (from highest to lowest) are human resources, Communication and Coordination, Training and Development, Work Environment, Performance Management, and Compensation.

Through correlation analysis and regression analysis of six process areas and organizational effectiveness, the authors found a significant correlation between improved organizational effectiveness and the degree of

maturity of some areas (product quality, communication between members, compensation, physical environment, staff stabilization).

4.11.4 Colomo, Casado-Lumbreras, Soto-Acosta, Misra, and García-Peñalvo (2012)

Colomo-Palacios et al. [71] studied the use of the P-CMM model in organizations that adopt globally distributed teams (Global Software Development – GSD) and how the P-CMM model can improve management in this context. Although there are challenges to overcome (cultural and communication problems), the study by Colomo et al. sheds light on human resource management, by studying which practices defined in the People-CMM are most important in GSD scenarios.

According to the authors, the People-CMM is one of the few methods to obtain quality in managing human resources within organizations. However, its application in distributed environments is not easy, as the model was designed to be implemented in a single organization. The challenges reported in the GSD create several constraints regarding the implementation of the model. Experts consider that fourteen of the nineteen variables of the P-CMM model are highly or significantly affected in GSD environments. Over 70% of processes are affected within the GSD to a large extent, with three of them (Communication and Coordination, Participatory Culture, and Empowered Working Groups) being affected at the highest level.

According to the authors, the more complex a process and the more level of maturity it requires, the more likely it is to be affected within the GSD. These results are consistent with those obtained for adapting processes to the GSD: processes belonging to high maturity levels are difficult to adapt to the GSD. This is the case of the processes: Communication and Coordination, Participatory Culture, and Empowered Working Groups. The only exception is the Mentoring process which, despite belonging to a high maturity level (level 4), is considered to have a mid-level GSD impact offered and is easy to adapt.

4.11.5 Zhang (2016)

The study by Zhang [17] considers that the growth of small and medium-sized enterprises (SMEs) depends mainly on the competitiveness generated by an efficient management system, highlighting the performance management system. According to the author, most private SMEs in China has not established a complete performance management system or are just at a stage equivalent to level 1 of the P-CMM model (which is the unmanaged one).

Zhang's work addresses the steps to have good performance management for SMEs in China, according to Level 2 of the P-CMM. For this, it suggests a sequence of phases: 1. Preparation phase; 2. Diagnosis Phase; 3. Design Phase; 4. Demonstration Phase; 5. Implementation phase; 6. Usage Phase; and 7. Summary. This model is well suited to the needs of small and medium-sized private companies and can help companies implement effective performance management.

4.11.6 Dutta, Baruah e Baruah (2018)

Finally, in the study by Dutta et al. [72], the P-CMM model is approached in the context of the information technology industry in India, and the explicit implementation of the model is important to reach the highest level of maturity, fundamental for the success of organizations.

According to the authors, it is known that most of the Indian tier 5 companies are improving day by day. However, lower-tier companies do not perform consistently because they do not follow PCMM extensively. At maturity levels 2 and 3, companies tend not to follow model training programs and prefer experienced employees trained by other higher-level companies. In these companies, workforce planning is very traditional, leading to lower strategic performance, lower development of human assets, lower management improvement, and dispersed team practices.

Thus, the authors conclude that the explicit implementation of the P-CMM is important to reach the highest level of maturity, which is fundamental for the success of organizations.

5. Discussions

In this section, the results found will be discussed, comparing the maturity models identified in relation to their characteristics (applications, areas and objectives), use, benefits and limitations, human factors and how they add value in software development environments.

Table 7 presents a summary of the maturity models identified, considering which types of teams are targeted, which aspects they have in their structure (technical or human) and which goals they intend to achieve.

Considering the nomenclature of the models can be interesting, since the nomenclature varies between the identified models. It is observed that the terminologies of the models found to vary between "maturity model" and "framework". Of the eleven models, seven are considered maturity models, three are called frameworks, one is a process area and one is considered a checklist. However, in all of them, levels are observed, which are characteristics of maturity models. Taking into account all the aspects observed, it can be seen that the differences in nomenclature between them (framework or maturity model) do not interfere with the objectives of the models. All have levels (or stages) of maturity, ensuring that the best practices can be implemented over time.

While this study sought for people-oriented models of maturity, evidence shows that of the eleven ones, only four of them are specifically people-oriented. The technical aspects can be observed in most of the analyzed models (eight), which may indicate that the human aspects are still little explored within the existing models, or even indicate that the processes are still above the people.

Regarding the application, most are aimed at agile teams, which may invite a concern in the formalization of some practices, even in agile environments.

Table 7. Synthesis of Maturity Models for People

Model	Type	Application	Areas	Objective	Maturity Levels
Agile Adoption Framework	Framework	Agile teams	Technical and human processes	Guide and assist in the adoption of Agile	5 levels
AGILE Maturity Map (AMM)	Maturity Model	Unique experience (Sabre Airline Solutions)	Technical and human processes	Increase adoption of agile practices	5 levels
Agile Software Solution Framework	Framework	Agile teams	Technical and human processes	Customize the agile approach in the organization	5 levels
Agile Maturity Model (AMM)	Maturity Model	Agile teams	Technical and human processes	Pushing agile principles	5 levels
Global Teaming (GT)	Process Area	GSD teams	Technical and human processes	Guidelines for global teams	2 Specific Objectives
GAIA Human Resources	Maturity Model	Development teams	People	Best practices to improve people management	5 levels
Progressive Outcomes Framework	Framework	Agile teams	People play a central role	Maturity of agile software development	6 categories
Virtual Team Maturity Model	Maturity Model	Virtual teams	People	Improve the efficiency of virtual teams	4 levels
Software Engineers Team Maturity	Maturity Model	Agile teams	Human aspects	Reach team maturity	12 categories
GÉANT Software Maturity Model	Maturity Model	Developed for G'EANT	Technical and human processes	Determine improvements for software teams	5 target areas
P-CMM	Maturity Model	Organizations	People	Attract, train, organize, motivate and retain talent	5 levels

The benefits and limitations of maturity models were also analyzed. To aid in the understanding of the comparison between them, the models are positioned in different Tables and classified according to their applications (Table 8: models that help in the adoption of agile practices; Table 9: models aimed at virtual teams; Table 10: models aimed at exclusively for people).

At this point, it is important to highlight that Tables 8, 9 and 10 summarize the maturity models from different perspectives, not just from the perspective of human factors. This is due to the fact that, despite these factors being present in the structure of the mentioned models, some of the models do not expose human issues as benefits or limitations.

According to what is being presented in Table 8, it can be seen that the models are based on the values of agility, but linking development practices to maturity levels. This may indicate the concern of the models to maintain agility, even using a maturity model. In addition, it can be said that, with the participation of the team in the development of goals, as is the case of the Agile Maturity Map and Progressive Outcomes Framework, including allowing the emergence of specific practices to the context, there is a concern of the models regarding human questions.

On the other hand, it is important to analyze the limitations of the models. Of the six models described in Table 8, four of them have problems related to validation: Progressive Outcomes Framework has not been applied in the industry; Agile Maturity Model has temporary conclusions; Agile Software Solution Framework and Agile Maturity Map have local conclusions and cannot be generalized. GÉANT does not provide evidence regarding validation, however, it is a short-range model, difficult to implement in internally diversified and independent organizations. The Agile Adoption Framework also does not provide evidence regarding validation; however, as a limitation, it is a model with many elements, not having reached its full potential yet.

It can be concluded that, of the maturity models aimed at the adoption of agile practices, none can be considered finalized, and not ready to be used effectively.

Table 8. Benefits and Limitations of models that guide the use of agile practices

Model	Benefits	Limitations
Agile Adoption Framework	• *Framework levels are not associated with any business value; they are based on the qualities and values of agility.* • *The levels are intended to provide a framework to guide the adoption process.* • *The framework is the first step in addressing the need to provide organizations with a structured approach to guide them in the movement towards agility.*	• *Need for flexibility to adapt SAMI (Agility Measurement Index) to individual experiences and business objectives.* • *SAMI's 4-stage process has not yet reached its full potential.*
Agile Maturity Map (AMM)	• *Improvements resulting from a goal-oriented approach are much faster and more sustainable than using more traditional practice-oriented approaches.* • *Team members value and respect a process that works, and they do much more when they participate in its development.*	• *Tendency for some teams to allow one team member to do most of the work, resulting in much less effective adoption than when multiple team members implement tasks.* • *A practice might work well for one team, and it might work poorly for another.* • *Based on a single experience (Sabre Airline Solutions).*
Agile Software Solution Framework (ASSF)	• *Provides flexibility, speed, responsiveness, communication and people orientation, executable artifacts, learning.* • *The organization can adapt or customize the AAIM (Agile Adoption and Improvement Model) according to its local organizational structure, culture, size, and development environment.*	• *The conclusions of the case studies are local and cannot be generalized.* • *The success of the transition to an agile environment depends on the leadership role of the CIO and the management executive.* • *A step-by-step approach to a gradual and successful transition is indicated..*
Agile Maturity Model (AMM)	• *It is based on the values, practices, and principles of agile software development.* • *Links agile development practices to maturity levels.* • *It is designed to improve and enhance agile software development methodology and drive agile principles forward.*	• *It is important that the improvements are applied in the main process areas that will provide a visible return in a short period, at the risk of the trust and support for the AMM program being eroded.* • *Conclusions are tentative and based on observation and informal discussion.* • *Verification and evaluation are still needed.*
Progressive Outcomes Framework	• *Allows the emergence of context-specific practices.* • *Considers people as agents who play a fundamental role in the maturation process.* • *Perceives ambidexterity as a fundamental skill for maturity.* • *Does not prescribe practices but describes the results that teams seek.*	• *Challenge to discover subjective ways of evaluating agile teams.* • *Not applied in the industry.*
GÉANT Software Maturity Model (GSMM)*	• *Aligns the effort to improve development processes with the individual governance structures of each entity participating in the GÉANT.* • *Extracted practices can be shared, adapted, and applied by teams to streamline and align software development and governance processes.*	• *Model difficult to implement in internally diversified and independent organizations.* • *It is a short-range model.*

The benefits and limitations of models that focus on global and distributed teams are presented in Table 9. Both GT and VTMM allow talent management around the world. On the plus side, the GT provides a guide to

understanding how to manage technical talent around the world, and the prescribed practices can be embedded in any model in use by the organization. However, the model still needs validation. VTMM serves as a reference model against which virtual teams can be evaluated and through which gaps in performance can be identified and remedied. The point of attention is that the effective participation of the team leader is important to understand the processes.

Table 10 compares three of the models found focused exclusively on human aspects for development teams (for local teams). The GAIA model was not applied in the industry and, in preliminary validation, there was a need to use a more transparent and dynamic process. As for the application of Software Engineering Team Maturity, it is important to observe individual maturity, the training time of the team, and external agents.

Analyzing the aspects found, despite being a complex model, the P-CMM is the one that has a complete approach, allowing to improve the quality of human resources management. Within this SLR, it was the most discussed model, with evidence of its application in six different situations — already mentioned in section 4. However, it is considered a complex model, not being suitable for small companies.

Table 9. Benefits and Limitations of Models for Virtual Teams

Model	Benefits	Limitations
Global Teaming (GT)	***• Provides a guide to understanding how to manage technical talent around the world.*** ***• It is called a process area for global teams, which can complement any other model, or be used in isolation.***	***• Framework is in its initial version (need validation).***
Virtual Team Maturity Model (VTMM)	***• Allows quick iterations on team performance improvements.*** ***• Requires a low investment.*** ***• Applicable to industry-based virtual teams.*** ***• Serves as a reference model.*** ***• Can be adapted to the organization's needs.***	***• Model only for GSD teams.*** ***• The effective participation of the team leader is important to understand the processes.*** ***• The entire team needs to be trained in communication processes.***

Table 10. Benefits and Limitations of People-oriented Models

Model	Benefits	Limitations
GAIA Human Resources	• It was developed to meet, deliver and add value to the software project through human resources.	• Not applied in the industry. • In the preliminary validation, there was a need to use a more transparent and dynamic process, both for managers and for the development teams.
Software Engineers Team Maturity	• Appropriate and assertive approaches to improve the maturity and effectiveness of software engineering teams in practice.	• For implementation, it is important to observe: 1. Individual maturity; 2. Team formation time; 3. External agents.
People Capability Maturity Model (P-CMM)	• People-oriented. • Developed for the software industry, but its focus has expanded to all types of organizations. • Provides a complete structure to improve the quality of human resources management. • It is adaptable to different types of organizations (levels can be broken down).	• Is not suitable for small companies • Complex model.

Finally, Table 11 summarizes the human factors found in the models surveyed. Regarding the factors found, it can be seen that the terms Collaboration, Communication, Culture, Empowerment, Knowledge, Learning, Motivation, Self-organization, Training/Development, and Competency, in general, are recurrent. However, there is also a concern about these models, from work environment to the employees' mental health.

Table 11. Models x Human Factors

Model/Framework	Human Factors
Agile Adoption Framework	***Ideal agile physical setup; Self-organization; Face-to-face communication; Collaboration; Empowered teams and Motivated teams.***
Agile Maturity Map (AMM)	***Awareness; Knowledge; Creative innovation; Mentoring; Productivity.***
Agile Software Solution Framework (ASSF)	***Self-assessment; Self-improvement; Empowered teams; Cooperative and Collaborative evaluation; Learning; Communication; Encourage and accommodate change; Motivation.***
Agile Maturity Model (AMM)	***Self-organization; Rewards; Respect for the co-workers; Collaboration; No overtime; Satisfaction; Staff retention; Communication; Empowered teams; Feedback.***
Progressive Outcomes Framework	***Experience; Knowledge; Collaboration; Communication; Commitment; Care; Self-organization; Practices learning; Team conduct.***
G'EANT Software Maturity Model (GSMM)	***Knowledge; Satisfaction; Collaboration; Feedback; Team organization.***
Global Teaming (GT)	***Communication; Culture; Cooperation; Coordination; Training teams; Motivation.***
Virtual Team Maturity Model (VTMM)	***Organize "Get-to-know-each-other"; Agree on Rules; Set Goals; Perform Task- Management; Give and Receive Feedback; Organize Decision-Making; Conduct Meeting-Management; Engage in Trust-building; Define Information- Management; Give Rewards and Recognitions; Arrange Ramping-Down.***
GAIA Human Resources	***Performance; Knowledge; Training; Motivation; Commitment; Change; Infrastructure.***
Software Engineers Team Maturity	***Interpersonal Understanding; Team Identity; Team Culture; Intra-Team Relationship; Intra-Team Communication; Extra-Team Relationship; Extra-Team Communication; Team Learning; Shared Knowledge; Mutual Support; Team Management; Team Technical Process.***
People Capability Maturity Model (P-CMM)	***Continuous Workforce Innovation; Organizational Performance Alignment; Continuous Capability Improvement; Mentoring; Organizational Capability Management; Quantitative Performance Management; Competency-Based Assets; Empowered Workgroups; Competency Integration; Participatory Culture; Workgroup Development; Competency-Based Practices; Career Development; Competency Development; Workforce Planning; Competency Analysis; Compensation; Training and Development; Performance Management; Work Environment; Communication and Coordination; Staffing.***

5.1 Implications for Research and Practice

Software organizations have applied and suggested several models aimed at developing people in development teams. However, there is still no consistent and comprehensive empirical evidence of the direct effects of this application. On the contrary, there are numerous limitations to implementing these models, some even not validated. Thus, it is suggested that new studies be carried out within each of the models identified by this SLR to determine and understand, with greater clarity, the impacts of the use of each model within the development teams.

For practice, the central implication of the evidence is that the models found different address aspects of human factors, despite being concerned with human relationships and issues inherent to development teams, such as communication, collaboration, and self-organization. However, due to limitations and problems encountered while implementing some of these models, it is important to have a clear understanding of the goals to be achieved and the potential interaction between the results.

5.2 Limitations of this Review

The most common limitation of systematic reviews is search coverage and possible biases introduced during study selection, data extraction, and analysis. These are also the main limitations of this review.

Potential research bias in the selection, extraction, and analysis of data was addressed by performing all steps of the peer review. Two researchers performed all the tasks independently and then merged the results. A third researcher supervised the process and participated in consensus meetings whenever necessary.

Finally, to address coverage, four mechanisms were used to search for primary studies. The content of each

paper resulting from the automatic searches was systematically read before discarding any potentially relevant paper. A manual search was performed to increase the coverage of primary studies.

An important limitation that worth mentioning is that one of the libraries that would be used as a search source (IEEE) was not available for data import during the period in which the research was carried out. As a way of increasing coverage, it was replaced by the Web of Science database.

The snowballing technique was not used, which could add pertinent new studies, but it is believed that this would not radically change the results.

6. Conclusions

This study identifies the maturity models for managing people in software development teams cited in the literature, in order to identify evidence about their use, benefits and limitations and human aspects involved. Thus, this study mentioned, in addition to models in use, non-validated studies, studies not applied in the industry, models based on local studies, as is the case of the Agile Maturity Map, VTMM, Agile Maturity Model and GT.

In the quest for answers to the research questions of this work, a systematic literature review was carried out. Four search engines were used for the automatic search, in addition to the manual search in the main events in the area. Considering the inclusion and exclusion criteria, seventeen studies were selected, covering eleven different maturity models.

Regarding QP1 and QP2, eleven models were identified: Agile Adoption Framework; Agile Maturity Map; Agile Software Solution Framework; Agile Maturity Model; Progressive Outcomes Framework; GÉANT Software Maturity Model; Global Teaming; Virtual Team Maturity Model; GAIA Human Resources; Software Engineers Team Maturity; and People Capability Maturity Model. Of these, six mention problems regarding validation, leaving five of them, in theory, available for use.

This is the case of the GÉANT model, focused on agile practices, but which was developed for application in the GÉANT project (something very specific), being difficult to implement in internally diversified and independent organizations. Another model that, in theory, does not have validation problems is the Agile Adoption Framework, also aimed at agile practices, but — because of its numerous elements in its structure — in need of a reformulation. The third one is the Virtual Team Maturity Model, that serves as a reference model for virtual teams. The limitation is in relation to communication, because the entire team needs to be trained in communication processes — something of a recurring problem in virtual teams. The fourth model is the Software Engineers Team Maturity, where the authors sought to understand the concept of mature teams in the context of software engineering. Team maturity is perceived as the balance of 3 dimensions: Learning, Relational and Technical Maturity. Although there is no explicit evidence that the model needs validation, further studies are needed. Finally, there is the P-CMM, entirely focused on people. It was developed for the software industry, but its focus has expanded to all types of organizations, providing a complete framework to improve the quality of human resources management, being adaptable to different types of organizations. A point of consideration, and seen as a limitation, is that it is a complex model, with many levels and practices, making its application difficult, particularly in agile teams.

In relation to QP3, which addresses the issue of human factors, fifty factors were identified within the eleven models surveyed. Highlight for communication, collaboration, knowledge, learning, self-management, motivation and skills in general.

Finally, this study contributes with an overview of maturity models for people, applicable to software engineering, and a finding that only the P-CMM is validated by industry practices.

References

1. Rovai, R. L. (2013). Metodologias inovadoras para gestão de projetos: modelo referencial para implantação da ITILV3 através da metodologia PRINCE2: estudo de caso. Revista de Gestão e Projetos-GeP, 4(2), 252-270.
2. Chrissis, M. B., Konrad, M., Shrum, S. (2006) CMMI: Guidelines for Process Integration and Product Improvement. 2nd Edition, Addison-Wesley.
3. Forsberg K. 2011 Visualizing Project Management (Beijing: Publishing House of Electronics Industry).
4. Miranda, R. (2011). Uma Revisão Sistemática Sobre Equipes de Desenvolvimento de Software: Tipologia, Características e Critérios de Formação. Centro de Informática (CIn), Universidade Federal de Pernambuco.
5. Moe, N. B.; Dingsoyr, T. Scrum and team effectiveness: Theory and practice.agile processes in software engineering and extreme programming. In: Lecture Notes in business Information Processing.[S.l.: s.n.], 2008
6. Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development.
7. Pirzadeh, L. Human factors in software development: A systematic literature review. In: Chalmers

university of technology. Göteborg, Sweden.[S.l.: s.n.], 2010.
8. Tripp, J. F., Riemenschneider, C., & Thatcher, J. B. (2016). Job satisfaction in agile development teams: Agile development as work redesign. Journal of the Association for Information Systems, 17(4), 1.
9. Capretz, L. F.; Ahmed, F. Making sense of software development and personality types. In: IT professional, 12(1). [S.l.: s.n.], 2010. p. 6–13.
10. Moser, P. C., & da Silva Araújo, J. I. (2020). Internalização de novos membros em equipes de desenvolvimento de software: uma versão detalhada. iSys-Brazilian Journal of Information Systems, 13(3), 25-54.
11. Moe, N.; Dingsoyr, T.; Dybå, T. A teamwork model for understanding an agile team: a case study of a scrum project. In: Inf. Softw. Technol. 52, 4 80–4 91.[S.l.: s.n.],2010.
12. Balijepally, V.; Mahapatra, R.; Nerur, S. P. Assessing personality profiles of software developers in agile development teams. Communications of the Association for Information Systems, v. 18, n. 1, p. 4, 2006.
13. Boehm, B.; Turner, R. Management challenges to implementing agile processes in traditional development organizations. In: Softw. IEEE 22 (5). [S.l.: s.n.], 2005. p. 30–39.
14. Roger S Pressman. 2005. Software engineering: a practitioner's approach. Palgrave Macmillan.
15. Ian Sommerville. 2010. Software engineering. Pearson.
16. Giehl, R. B. T. (2007). P-cmm: Qualidade no processo de software e gestão de pessoas.
17. Zhang, C. Design of human capability maturity analysis system online p-cm model.In: IEEE.2015 International Conference on Intelligent Transportation, Big Data and Smart City. [S.l.], 2015. p. 302–305.
18. Ambler, S. (2010). The agile maturity model (AMM). Dr. Dobbs Journal, April, 1.
19. Sidky, A., Arthur, J., & Bohner, S. (2007). A disciplined approach to adopting agile practices: the agile adoption framework. Innovations in systems and software engineering, 3(3), 203-216.
20. Qumer, A., & Henderson-Sellers, B. (2008). A framework to support the evaluation, adoption, and improvement of agile methods in practice. Journal of systems and software, 81(11), 1899-1919.
21. Capretz, L. F., Ahmed, F., & da Silva, F. Q. B. (2017). Soft sides of software. arXiv preprint arXiv:1711.07876.
22. Weinberg, G. M. (1998). The Psychology of Computer Programming, 1971. New York: von Nostrand Reinhold.
23. Shneiderman, B. (1980, June). Natural vs. precise concise languages for human operation of computers: Research issues and experimental approaches. In 18th Annual Meeting of the Association for Computational Linguistics (pp. 139-141).
24. Lister, T. R., & DeMarco, T. (1987). Peopleware: Productive projects and teams. New York: Dorset House.
25. Cockburn, A. (2002) "Agile Software Development". Boston: Addison Wesley.
26. Cockburn, A. Highsmith, J. (2001) "Agile Software Development: The People Factor," Computer, pp. 131-133.
27. Silva, R. C. d. C. Um estudo sobre a influência de fatores humanos e culturais em projetos de desenvolvimento de software ágeis. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2017.
28. Bullen, C. V., & Rockart, J. F. (1981). A primer on critical success factors.
29. Nielsen, J. L. (2002), "Critical Success Factor for Implementing an ERP system in a University Environment: A Case study from the Australia, Honours Dissertation", Brisbane, Australia, Griffith University, 2002, 201p.
30. Livermore, J. A. "Factors that Impact Implementing an Agile Software Development Methodology". SoutheastCon. Proceedings. IEEE.p. 69-72, 2007.
31. Mourmant, G.; Gallivan, M. How personality type influences decision paths in the unfolding model of voluntary job turnover: an application to is professionals. In: In Proceedings of the 2007 ACM SIGMIS CPR conference on Computer personnel research: The global information technology workforce (pp. 134-143). ACM.[S.l.: s.n.], 2007.
32. CANNON-BOWERS, J. A.; BOWERS, C. Team development and functioning. In: APA Handbook of Industrial and Organizational Psychology, Vol 1: Building and Developing the Organization, edited by S. Zedeck Copyright © 2011 American Psychological Association. [S.l.: s.n.], 2011.
33. Hackman, J. R. A set of methods for research on work teams. In: Yale Univ New Haven CT School of Organization and Management. [S.l.: s.n.], 1982.
34. Guzzo, R. A.; Shea, G. P. Group performance and intergroup relations in organizations. In:Handbook of industrial and organizational psychology. [S.l.: s.n.], 1992.
35. Katzanbach, J.; Smith, D. The discipline of teams. Boston: Harvard business school. In: Boston: Harvard Business School. [S.l.: s.n.], 1993.
36. Kozlowski, S. W. J.; Ilgen, D. R. Enhancing the effectiveness of work groups and teams. In: Psychological science in the public interest. [S.l.: s.n.], 2006.
37. Jr, F. P. B. The mythical man-month: essays on software engineering. [S.l.]: Pearson Education, 1995.

38. Lettice, F.; McCracken, M. Team performance management: a review and look forward. In: Team Performance Management: An International Journal. [S.l.: s.n.], 2007.
39. Faraj, S.; Sproull, L. Coordinating expertise in software development teams. In: Manage. Sci. 46 (12), 1554–1568.[S.l.: s.n.], 2000.
40. Pikkarainen, M.; Haikara, J.; Salo, O.; Abrahamsson, P.; Still, J. The impact of agile practices on communication in software development. In: Empir. Softw.Eng. 13, 303–337.[S.l.: s.n.], 2008.
41. Chow, T. A survey study of critical success factors in agile software projects.In: The Journal of Systems and Software. [S.l.: s.n.], 2008. p. 961–971.
42. Maruping, L.; Zhang, X.; Venkatesh, V. Role of collective ownership and coding standards in coordinating expertise in software project teams. In: Eur. J. Inf. Syst. 18, 355–371. DOI: 10.1057/ejis.2009.24.[S.l.: s.n.], 2009
43. Sharp, H.; Robinson, H. Three "c" s of agile practice: Collaboration, coordination, and communication. In: In Agile Software Development. Current research and future directions, XP 2010, Trondheim, pp. 61–85, DOI: 10.1007/978- 3- 642- 12575- 1.[S.l.:s.n.], 2010.
44. Dingsoyr, T.; Dyba, T. Team effectiveness in software development: Human and cooperative aspects in team effectiveness models and priorities for future studies. In: IEEE.2012 5th international workshop on cooperative and human aspects of software engineering (chase). [S.l.], 2012. p. 27–29.
45. Lindsjorn, Y.; Sjoberg, D. I.; Dingsoyr, T.; Bergersen, G. R.; Dyba, T. Teamwork quality and project success in software development: A survey of agile development teams. In: Journal of Systems and Software, 122, 274-286.[S.l.: s.n.], 2016.
46. Barbosa, I. F.; Oliveira, M. P.; Reis, P.; Gomes, T.; Silva, F. Q. da. Towards understanding the relationships between interdependence and trust in software development: qualitative research. In: In Proceedings of the 10th international workshop on Cooperative and Human Aspects of Software Engineering. [S.l.: s.n.], 2017.p. 66–69.
47. SEI. CMMI® for Development. Version 1.3, Technical Report, CMU/SEI-2010-TR-033, Software Engineering Institute, Nov. 2010. 482 p.
48. Proença, D., & Borbinha, J. (2018, September). Maturity models for data and information management. In International conference on theory and practice of digital libraries (pp. 81-93). Springer, Cham.
49. Aysolmaz, B., & Demirörs, O. (2011, June). A detailed software process improvement methodology: BG-SPI. In European Conference on Software Process Improvement (pp. 97-108). Springer, Berlin, Heidelberg.
50. Curtis, B., Hefley, B., & Miller, S. (2001). People Capability Maturity Model®(P-CMM®) Version 2.0, Software Engineering Institute.
51. Gamal, A. M. A blueprint for building its workforce empowerment program basedpcmm level 3 (people capability maturity model). In: In 2008 ITI 6th InternationalConference on Information Communications Technology (pp. 101-105). IEEE.[S.l.:s.n.], 2008.
52. Kitchenham, B.; Charters, S. Guidelines for Performing Systematic Literature Reviews in Software Engineering, Keele University. [S.I.], 2007.
53. Gil, A. C. (2010). Como elaborar projetos de pesquisa/–12. Reimpressão.–São Paulo: Atlas, 2009. Como elabora projetos de pesquisa./5. Ed.–São Paulo: Atlas.
54. Randolph, J. (2009). A guide to writing the dissertation literature review. Practical Assessment, Research, and Evaluation, 14(1), 13.
55. Dybå, T.; Dingsøyr, T. Empirical Studies of Agile Software Development: A Systematic Review. Information and Software Technology, Butterworth-Heinemann Newton, MA, USA, v. 50, n. 9, p. 833-859, August 2008.
56. Easterbtook, S. M.; et al. (2007). Selecting Empirical Methods for Software Engineering Research. In F. Shull, J. Singer and D. Sjøberg (eds) Guide to Advanced Empirical Software Engineering, Springer.
57. Moher, D.; Liberati, A.; Tetzlaff, J.; Altman, D.G. for the PRISMA Group. Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement. J. Clin. Epidemiol. 2009, 62, 1006–1012.
58. Fink, A. Conducting Research Literature Reviews. From the Internet to Paper, Sage Publication, Inc., 2005.
59. Petticrew, M.; Roberts, H. Systematic Reviews in the Social Sciences: A Practical Guide, Blackwell Publishing, 2005, ISBN 1405121106.
60. Noblit, G.W. and Hare, R.D. Meta-Ethnography: Synthesizing Qualitative Studies. Sage Publications, 1988.
61. Packlick, J. (2007, August). The agile maturity goal-orientated approach to agile improvement. In Agile 2007 (AGILE 2007) (pp. 266-271). IEEE
62. Patel, C., & Ramachandran, M. (2009). Agile maturity model (AMM): A Software Process Improvement framework for agile software development practices. International Journal of Software Engineering, IJSE, 2(1), 3-28.
63. Richardson, I., Casey, V., McCaffery, F., Burton, J., & Beecham, S. (2012). A process framework for global software engineering teams. Information and Software Technology, 54(11), 1175-1191.

64. Horita, F. E., & Barros, R. M. (2012). Gaia's human resources-an approach to integrating ITIL and maturity levels focused on improving the human resource management in software development. At 25th International Conference on Computer Applications in Industry and Engineering (CAINE).
65. Fontana, R. M., Meyer Jr, V., Reinehr, S., & Malucelli, A. (2015). Progressive Outcomes: A framework for maturing in agile software development. Journal of Systems and Software, 102, 88-108.
66. Friedrich, R., Bleimann, U., Stengel, I., & Walsh, P. (2016). The Virtual Team Maturity Model (VTMM) for real Virtual Project Team Performance. In The 7th International Conference on Society and Information Technologies ICSIT.
67. Marsicano, G.; Pereira, D. V.; Silva, F. Q. da; França, C. Team maturity in software engineering teams. In: IEEE.2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM). [S.l.], 2017. p. 235–240.
68. Walter, B., Marović, B., Garnizov, I., Wolski, M., & Todosijevic, A. (2020, September). Best practices for software maturity improvement: a GÉANT case study. In European Conference on Software Process Improvement (pp. 30-41). Springer, Cham.
69. Srinivasa, G., & Ganesan, P. (2002, August). Pair Programming: Addressing Key Process Areas of the People-CMM. In Conference on Extreme Programming and Agile Methods (pp. 221-230). Springer, Berlin, Heidelberg.
70. Lu, X.; Xu, D.; Han, J. Research on the staff maturity of software companies in china- take Hangzhou city as a sample. In: In 2010 International Conference on Management and Service Science (pp. 1-4). IEEE.[S.l.: s.n.], 2010.
71. Colomo-Palacios, R., Casado-Lumbreras, C., Soto-Acosta, P., Misra, S., & García-Peñalvo, F. J. (2012). Analyzing human resource management practices within the GSD context. Journal of Global Information Technology Management, 15(3), 30-54.
72. Dutta, K., Baruah, N., & Baruah, J. B. A Rigorous Study of P-CMM in Indian IT Industry.