# People Management Problems and Practices in Software Development Projects: A Systematic Literature Review

Marcelo Falkowski Burkard and Lisandra Manzoni Fontoura
*Programa de Pós-Graduação em Ciência da Computação, Federal University of Santa Maria (UFSM), Brazil*

Keywords: People Management, Problems, Practices, Systematic Literature Review.

Abstract: In the literature related to software engineering, it is possible to observe a large volume of work related to technology and processes. Comparatively, little has been done in people management, making it difficult for project managers to manage teams effectively and solve people-related problems. This study conducts a systematic literature review (SLR) to survey the people management problems faced in software development. In addition, we identify the solutions to these problems. We searched four major bibliographic databases and identified 2736 primary studies between 2016 and 2022, resulting in 35 selected articles. So, we have grouped problems and solutions by similarity to facilitate analysis. We identified 9 groups of problems and 11 groups of solutions. Communication, motivation, technical skills, and knowledge problems are most frequently reported. Regarding the solutions, the most cited are team building, feedback, and training practices. The problems and practices identified consolidate the knowledge and experience obtained in several software projects and can help managers in people management activities in software projects.

## 1 INTRODUCTION

People are one of the fundamental elements for the execution of software development projects. Their importance is recognized by both traditional and agile approaches to project management.

Many works about technologies and processes have been published, but few empirical studies explore people management.

According to Mishra and Misra (2010), the success of projects depends mainly on the individuals who make up the team, so organizations must seek and employ the best practices in people management to achieve excellence.

Best practices lead organizations to ever higher performance. According to Kerzner (2006), best practices guide continuous improvements, leading to the adoption of new best practices. So, the practice's applicability depends on the project's characteristics to which it will be applied. Therefore, it is up to the manager to assess when and where to use it.

The main objective of this SLR is to explore studies related to people management in software development to identify problems and practices used to solve them. So, this work seeks to create a new artifact for use by software project managers in their activities, compiling recent literature and facilitating the search for practices that fit the work context of each manager.

This article presents the results of an SLR in studies published from 2016 to 2022 and was conducted following a predefined protocol detailed in the following sections. First, we describe the methodology, detailing the steps followed in the study in Section 2. Then, in Section 3, we discuss the results. Finally, we present our conclusions and ideas for future work in Section 4.

## 2 METHODOLOGY

We follow the review protocol proposed by Kitchenham and Charters (2007) to perform the SLR. This protocol is based on research questions, search string, inclusion and exclusion criteria, primary search process, and study selection process, which are detailed below.

This work conducts an SLR to map people management problems and practices in software development projects, seeking recent literature to answer the questions:

- RQ1: What people management problems were cited by the authors of the primary study that af-



Burkard, M. and Fontoura, L.
People Management Problems and Practices in Software Development Projects: A Systematic Literature Review.
DOI: 10.5220/0011985300003467
In *Proceedings of the 25th International Conference on Enterprise Information Systems (ICEIS 2023) - Volume 2*, pages 179-186
ISBN: 978-989-758-648-4; ISSN: 2184-4992

fect software development projects?

- RQ2: What practices related to people management were cited by the authors of the primary study and that tend to solve problems in software development projects?

Based on these questions and the work objectives, we have defined the search string shown in 1.

Table 1: Search string also used on engines.

| Search string |
|---|
| *(“software development” OR “software project” OR “software engineering”) AND (“human resource” OR “human resources” OR “people factor”) AND (“productivity” OR “performance”)* |

The inclusion and exclusion criteria were based on those proposed by Kitchenham and Charters (2007) with adjustments according to the research questions of this RSL. Table 2 presents the adopted criteria.

Table 2: Inclusion and exclusion criteria.

| ID | Inclusion criteria | ID | Exclusion criteria |
|---|---|---|---|
| IC1 | primary study | EC1 | not primary study |
| IC2 | related to software development | EC2 | not answer research questions |
| IC3 | related to human factors | EC3 | presented as a book |
| IC4 | published between 2016 and 2022 | EC4 | published before 2016 |
| IC5 | published in English | EC5 | not published in English |
| | | EC6 | duplicate study |
| | | EC7 | with less than 15 score points |

We performed the primary search in four databases: ACM Digital Library, IEEE Xplore, Scielo, and ScienceDirect. Only libraries that could export files in the *BibTeX* format with a summary included were considered. The results were exported in files in the *BibTeX* format, including abstracts and all metadata. ACM found 969, IEEE found 328, Scielo found 2, and ScienceDirect found 1437 articles.

The selection of studies was divided into three stages. In the first stage, we imported the *BibTeX* files into the tool StArt[1], which automatically identified 241 duplicate articles. They were rejected according to the EC6, resulting in 2495 non-duplicated articles. 239 articles were published before 2016 or did not have a publication date, all were rejected according to the EC4. The StArt tool assigns a score to each article according to the number of occurrences of keywords in the title (5 points), abstract (3 points), and keywords (2 points); 2051 articles with less than 15 points were rejected according to the EC7.

In the second stage, we read the title and abstract of the selected articles. 134 articles were rejected according to the EC2, resulting in 71 articles were selected at the end of this stage.

In the third stage, we read all articles and excluded 36 according to criteria EC1, EC2, EC3, and EC5. At the end of the process, 35 articles were selected.

[1]State of the Art through SLR - available at http://lapes.dc.ufscar.br

# 3 RESULTS

This section presents the results found for each of the research questions.

## 3.1 Research Question 1

*RQ1: What people management problems were cited by the authors of the primary study that affect software development projects?*

The problems cited in the primary studies were analyzed and categorized into 9 groups according to their similarities to facilitate the analysis.

### 3.1.1 Communication

Communication refers to how one person relates to another. Shameem et al. (2020) and Margareth and Mulyanto (2021) cite that the incorrect choice of communication tools can cause communication breakdowns and misunderstanding between the sent message and the receiver’s understanding.

de Magalhães (2017), Wang et al. (2018), Bass (2016), Machuca-Villegas et al. (2022) and Shameem et al. (2020) argue that a big team makes communication difficult and, thus, this occurs with low frequency. And, for communication to occur with reasonable frequency, Stylianou and Andreou (2016) mention that there is an increase in communication overhead, that is, an increase in time spent organizing communication instead of performing productive work.

de Magalhães (2017) shows that the lack of adequate feedback is one of the leading causes of demotivation and burnout in the development team.

### 3.1.2 Motivation

According to Machuca-Villegas et al. (2022), motivation moves a person towards action. França et al.

(2020) argue that motivation refers to the desire to work and is signaled by people's attitude towards their work, directly influencing their performance. Lack of motivation is a threat to both the development and the management team (García et al., 2017). Low motivation levels also affect productivity, effectiveness, and team learning (Fatema and Sakib, 2017). de Magalhães (2017) mentions little autonomy and lack of adequate feedback as one of the reasons for low motivation. Bass et al. (2018) complement: employment policies, work-life balance, common technical challenges, innovation, number of hours, rewards, good management, adequate working conditions, work involvement with others, and quality of work generated.

In addition to being one of the causes, low quality of work and generated products is also pointed out by Bass et al. (2018) as one of the effects of low motivation. Bass et al. (2018) also presents a correlation between motivation and turnover in organizations - motivated professionals tend to remain in their current jobs; in contrast, professionals with low motivation tend to generate attrition and dismissal.

Team motivation depends on choosing appropriate strategies. Shameem et al. (2020) comment that the lack of motivational strategies impacts the work, making it impossible to escalate the use of agile methodologies in software development. Bass et al. (2018) propose using gamification, mechanics, and characteristics typical of games for monitoring and managing software development professionals to improve their motivation and engagement.

#### 3.1.3 Technical Skills and Knowledge

Regarding technical factors that influence productivity in software development, Meyer et al. (2017) list: the domain of programming language and tools, software size, complexity, and product quality.

Fatema and Sakib (2017) claim that selecting people with the right technical skills for a project is one of the most complex activities in project management.

Nigar (2017) and da Cunha et al. (2016) mention that it is not enough to have technical knowledge. The lack of skill (experience or practice) in applying that knowledge also negatively impacts software delivery.

Regarding the evolution of knowledge and skills, Kula et al. (2021) argue that greater team stability is positively related to the development of skills and technical knowledge.

#### 3.1.4 Geographic Aspects

Especially when it comes to large global organizations or companies using outsourcing, geographic aspects play an essential role in the productivity of development teams. Outsourcing is the process of acquiring products or services from a third-party supplier. Bass (2016) divides it into two categories: onshore, in which the supplier is located in the same territory as the customer, and offshore, which involves a geographically remote supplier, usually separated from the customer by a significant temporal distance (time zone) and culture differences. In contrast, some organizations create their structures in an offshore format which helps build a presence in emerging markets while benefiting from lower costs. However, Qahtani (2020) and Shameem et al. (2020) warn that cultural differences can have a negative impact on team performance.

Related to the lack of rapport between teams and the difficulty of collaboration, Bass et al. (2018) argue that language differences and the reduction of overlapping work schedules caused by differences in time zones generates communication delays.

Communication difficulty due to cultural differences generates a cascade effect that impacts other aspects, according to Britto et al. (2019). They exemplify that low cultural adequacy can even make training difficult.

#### 3.1.5 Commitment

Commitment is related to engagement and perseverance. Machuca-Villegas et al. (2022) characterize it as the level of individual responsibility that a person assumes to carry out the activities that make up the delivery of a team. Kula et al. (2021) identify that commitment is one of the main factors that promote the team's effectiveness and, therefore, helps deliveries to occur on time. Professionals with a higher level of commitment and technical expertise tend to identify better and assess risks, which increases the chances of project success.

Regarding productivity, Oliveira et al. (2016) identified that managers tend to perceive greater productivity in professionals who combine focus, commitment, and proactivity with tasks delivered without delay and with good quality. Everyone should take responsibility for the results obtained, fulfill their duties, and, eventually, admit their own mistakes to improve the software development process Machuca-Villegas et al. (2022).

Tam et al. (2020) show that in projects that follow the waterfall methodology, customer involvement occurs mainly at the beginning and end of the project. In contrast, in agile projects, customer involvement must be high throughout the project's life cycle - the customer must also be "agile".

#### 3.1.6 Satisfaction

According to Machuca-Villegas et al. (2022), job satisfaction can be determined by the difference between what people want and what they have in their work. Therefore, we have dissatisfied professionals when expectations regarding certain aspects are not met.

França et al. (2020) argue that satisfaction refers to pleasant emotions concerning work and directly influences the performance of software development teams as it increases their desire to stay in the organization and be present at work. de Magalhães (2017) cite aspects that generate job satisfaction: personal and professional growth, recognition, opportunities, salary, and relationships with colleagues.

#### 3.1.7 Team Stability

Analyzing factors perceived as contributing to on-time deliveries, Kula et al. (2021) list: team stability, represented by low turnover and team familiarity, that is, the amount of time that members work with each other. The authors argue that both factors are related to improving coordination and adaptability in case of environmental changes (Kula et al., 2021). Therefore, project managers should focus on keeping teams stable and supporting the team in development to achieve better deliveries.

One challenge of keeping teams stable for long periods is employee turnover. Since job changes are a natural phenomenon in the market and high staff turnover has a negative impact on productivity and quality during software development, it is up to the manager to mitigate turnover risks. Bass et al. (2018) correlate low motivation and high turnover rates.

Another contributing factor to turnover rates presented by Bass et al. (2018) is the nature of the work performed - companies in the outsourcing sector have a higher turnover expectation for the development team than companies in other segments.

#### 3.1.8 Focus

One of the developer profiles featured in the work of Meyer et al. (2017) is called a focused developer, who feels more productive when he has worked concentrated on a single task at a time.

Last-minute meetings are characterized as a cause of constant change of focus, which, as identified by Oliveira et al. (2016), impacts achieving the state of "flow" desired by developers. Kohl et al. (2020) interviewed developers and concluded that small context changes lasting less than 3 minutes do not disconnect the developer from the previous task. For example, running a script that takes 15 seconds does not generate impacts related to loss of focus.

#### 3.1.9 Autonomy

Especially about the team's autonomy in defining and assigning responsibilities, Machuca-Villegas et al. (2022) mention this as a factor with a high impact on the performance of individuals during software development: the greater the team's autonomy in self-management and defining who will be responsible for a task, the greater the team's motivation and productivity. In contrast, several authors argue that task assignment is a resource allocation problem, and therefore an optimized solution should be sought through mathematical models (Song et al., 2020)(Chiang and Lin, 2020).

### 3.2 Research Question 2

*RQ2: What practices related to people management were cited by the authors of the primary study and that tend to solve problems in software development projects?*

The practices cited in the primary studies were analyzed and categorized into 12 groups according to their similarities to facilitate the analysis.

#### 3.2.1 Team Assembly

According to Zaouga et al. (2019) and Meyer et al. (2017), the manager must ensure the assignment of members to a project considering the project's technical specificities and the individual's capacity.

Angelis (2019) argues that professionals whose qualification just fits to project's needs should be sought for the correct assembly of the team, rather than overqualified professionals, for example. This makes the project flow, and professionals have an adequate level of challenge, avoiding problems of satisfaction and motivation.

In addition to the importance of technical skills, Caulo et al. (2021) and da Cunha et al. (2016) argue that the personality traits of individuals also impact the performance of software development teams.

Identifying the most suitable professionals for the project begins in the recruitment phase. Nastiti and Setyohadi (2020) state that unclear job descriptions are one of the leading causes of incorrect hires, as the attracted candidate profile is not very adherent to the project's actual needs. Therefore, it is up to the manager to ensure adequate job descriptions. Regarding the selection process, Nastiti and Setyohadi (2020) reinforce that professional selection should focus on educational aspects and candidates' experience, which

can be evaluated during interviews. Paredes-Valverde et al. (2018) propose an ontology that covers aspects that should be evaluated during selection.

Ideally, the team should be heterogeneous, meaning that no one person should have the same expertise as another. Thus, it adds diversity to discussions and improves the quality of decisions. Schloegel et al. (2016) argue that the team must have age diversity among its members and technical breadth.

Finally, Fatema and Sakib (2017), Bass et al. (2018), Wang et al. (2018), and Stylianou and Andreou (2016) emphasize the importance of maximizing delivery while minimizing communication overhead. It is up to the manager to find the balance point from which adding new members reduces the amount of software delivered.

#### 3.2.2 Feedback

Feedback is information an agent provides regarding someone's performance or understanding. Hattie and Timperley (2007). Zaouga et al. (2019) characterize the activity of providing feedback as part of the process of managing the project team.

According to Bass et al. (2018), software developers consider the practice of feedback as a form of recognition, and the lack of feedback is a communication problem that directly impacts motivation and, consequently, the quality of the team's delivery de Magalhães (2017). Fatema and Sakib (2017) and Machuca-Villegas et al. (2022) agree that adequate feedback is among the main factors that influence the productivity of software development teams.

#### 3.2.3 Training

As identified by Shahzad et al. (2017), employees expect the organization to be committed to constantly training its professionals. In contrast, Bass et al. (2018) add that the manager should challenge team members to learn new skills.

The first step to guarantee the team's qualification is the effective management of competencies, according to Song et al. (2020) and Angelis (2019). With this, gaps in knowledge and professionals who can become eventual technical disseminators can be identified. Training should not be limited to hard skills. Cárdenas-Castro et al. (2019) argue that psychological training improves human skills and well-being within the team, and its results are more useful in the early stages of the project than when compared to the production stages.

Nicolaescu et al. (2020) suggest training in conjunction with career monitoring and succession planning, to keep professionals motivated.

#### 3.2.4 Resource Allocation Optimization

During the software project planning stage, the project manager must solve a "puzzle": the allocation problem in software projects.

Aiming to create an optimized minimum schedule, Ángel Vega-Velázquez et al. (2018) present models that balance time and costs. Furthermore, Chiang and Lin (2020) add to their model the skills of the people involved to improve the accuracy of the estimates. Finally, Nigar (2017) introduces a mathematical model that considers five factors: project duration, task fragmentation, robustness, cost, and stability.

As an alternative to models that consider a limited number of constraints, Shen et al. (2020) propose to increase the number of variables considered. Shen et al. (2018) propose an algorithm based on machine learning to dynamically allocate individuals, considering aspects such as the ability and motivation to learn and the evolution of technical proficiency over time.

Finally, Song et al. (2020) propose a model for optimizing resource allocation considering specifics of the corporate culture of Chinese state-owned companies.

#### 3.2.5 Performance Monitoring

According to Nicolaescu et al. (2020), it is a key activity for managers to measure and monitor performance in software development. Oliveira et al. (2016) argue that the measures can be used to compare the efficiency of different developers in the same organization, helping to assemble and adjust teams.

According to Zaouga et al. (2019), performance metrics provide good-quality feedback to improve the professional's performance.

Nastiti and Setyohadi (2020) suggest that performance measurement can be done through KPIs (key performance indicators) previously agreed upon with the team. Traditionally, performance is related to efficiency and productivity. Cárdenas-Castro et al. (2019) argue that it's essential that the performance indicators agreed upon also assess individual soft skills, thus promoting more collaboration.

The indicators can be made available on a board or online tool visible to the entire team so that a team trained in interpreting the indicators can monitor each other's performance (Fatema and Sakib, 2017).

#### 3.2.6 Use of Agile Practices

Machuca-Villegas et al. (2022) state that team members should be empowered to make decisions and organize themselves to establish and achieve goals. To

have self-organization maturity related to attribution, autonomy, and definition of work methods, de Magalhães (2017) suggests using methods such as Extreme Programming and Scrum.

In organizations whose teams do not employ such agile methods and there is a strong dependence on the manager's role, Shahzad et al. (2017) suggest that the latter constantly questions the team if there is a better way of doing things.

Pair programming is an agile practice consisting of two programmers sharing the same workstation, one writing the code while the other analyzing the work done. Caulo et al. (2021) and Fatema and Sakib (2017) list pair programming as a practice perceived as related to increased productivity.

#### 3.2.7 Job Rotation

Before the 1950s, most work strategies in traditional industries were characterized by simplification, specialization, and repetition. Currently, in knowledge industries, such as software development, these same characteristics are recognized for generating monotony, fatigue, and, consequently, a decrease in the performance of individuals de Magalhães (2017).

Job rotation is moving individuals on software projects within the same organization. Santos (2017) characterizes movements in two formats: job-to-job (playing a new role within the organization) and project-to-project (moving to another project or team).

França et al. (2020) point out that this can be a positive practice in terms of motivation and satisfaction of individuals, making professionals constantly challenged to learn new skills.

In contrast, Kula et al. (2021) indicate that managers should try to keep stable teams where members can build long-term familiarity with on-time deliveries and performance optimization. Therefore, it is up to the manager to find the balance between motivation generated by new opportunities and performance optimization resulting from maintaining a stable team.

#### 3.2.8 Personality Assessments

Angelis (2019) presents empirical results that support the claim that personality traits impact the productivity of software development teams. The project manager can use psychometric tests with the support of professionals in the field of psychology to identify the profiles of individuals.

The Big Five categorization model divides human profiles into five major areas: extroversion, agreeableness, conscientiousness, neuroticism, and openness. Caulo et al. (2021) propose that profiles with a higher degree of kindness and conscientiousness correspond to people with greater productivity in software development.

Furthermore, Meyer et al. (2017) proposes a new categorization of software development professionals using six categories: social developer, solitary developer, focused developer, balanced developer, lead developer, and goal-oriented developer. The author presents suggestions on how to work with each professional.

#### 3.2.9 Communication

The impacts of inefficient communication are widely discussed in the literature; more is needed regarding good practices to ensure good communication.

Bass (2016) suggests online tools for the virtual representation of Kanban boards, this being a practice highly related to monitoring practices.

As for the frequency of communication, Margareth and Mulyanto (2021) indicate that the communication frequency positively impacts productivity. Therefore, the team must maintain fluid communication and avoid static communication protocols.

#### 3.2.10 Rewards

Related to rewards and incentives for the development team, Bass et al. (2018) cites practices that can happen in conjunction with constant performance evaluations: salary increases and promotions of competent personnel.

In addition, Fatema and Sakib (2017) mentions that at times when there is a need for overtime, small recognitions are appropriate for the team.

#### 3.2.11 Social Encounters

Shahzad et al. (2017) states that companies with a culture that has routines of social encounters where everyone gathers promote a culture of innovation and collaboration.

#### 3.2.12 Onboarding

Onboarding members to globally distributed project teams can take longer than when the entire team is in one location. Britto et al. (2019) mention, as an example, projects of high complexity and legacy code for which remote mentoring during a period of 4-6 months may be insufficient. The authors suggest prioritizing the hiring of mentors who can pass on knowledge face-to-face and synchronously to increase the speed of the team's progress and look for mentors with good communication skills.

# 4 CONCLUSIONS

This study aimed to conduct an SLR to identify problems arising from people management in software development projects and what practices can be employed to solve them.

Frequent problems included communication, motivation, technical skills, knowledge, and geography.

We also look for practices that are frequently used to solve problems related to human resource management. Practices related to team assembly, feedback, training, resource allocation optimization, and performance monitoring were the most cited.

In future work, we can identify which, among the practices mapped in this work, are characterized as best practices. In addition, best practices can be validated with specialists in the field, and a catalog detailing each of these practices and related problems they are intended to solve can be drawn up, to facilitate managers' search for practices that fit their context of work.

# ACKNOWLEDGEMENTS

We thank the Brazilian Army and its Army Strategic Program ASTROS for the financial support through the SIS-ASTROS GMF project (898347/2020).

# REFERENCES

Angelis, L. (2019). Statistical analysis in the research of human factor in software engineering: Invited talk paper. In *Proceedings of the 9th Balkan Conference on Informatics*, BCI'19, New York, NY, USA. Association for Computing Machinery.

Bass, J. M. (2016). Artefacts and agile method tailoring in large-scale offshore software development programmes. *Information and Software Technology*, 75:1–16.

Bass, J. M., Sarah, B., Razzak, M. A., and Noll, J. (2018). Employee retention and turnover in global software development: Comparing in-house offshoring and offshore outsourcing. In *2018 IEEE/ACM 13th International Conference on Global Software Engineering (ICGSE)*, pages 77–86.

Britto, R., Smite, D., Damm, L.-O., and Börstler, J. (2019). Performance evolution of newcomers in large-scale distributed software projects: An industrial case study. In *2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)*, pages 1–11.

Caulo, M., Francese, R., Scanniello, G., and Tortora, G. (2021). Relationships between personality traits and productivity in a multi-platform development context. In *Evaluation and Assessment in Software Engineering*, EASE 2021, page 70–79, New York, NY, USA. Association for Computing Machinery.

Chiang, H. Y. and Lin, B. M. T. (2020). A decision model for human resource allocation in project management of software development. *IEEE Access*, 8:38073–38081.

Cárdenas-Castro, C., Gil Julio, J. C., and Rodríguez, P. (2019). Soft skills training: Performance psychology applied to software development. In *2019 IEEE/ACM 12th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, pages 115–116.

da Cunha, J. A. O., da Silva, F. Q., de Moura, H. P., and Vasconcellos, F. J. (2016). Towards a substantive theory of decision-making in software project management: Preliminary findings from a qualitative study. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 1–10.

de Magalhães, C. V. (2017). Toward understanding work characteristics in software engineering. *ACM SIGSOFT Software Engineering Notes*, 41(6):1–6.

Fatema, I. and Sakib, K. (2017). Factors influencing productivity of agile software development teamwork: A qualitative system dynamics approach. In *2017 24th Asia-Pacific Software Engineering Conference (APSEC)*, pages 737–742.

França, C., da Silva, F. Q. B., and Sharp, H. (2020). Motivation and satisfaction of software engineers. *IEEE Transactions on Software Engineering*, 46(2):118–140.

García, F., Pedreira, O., Piattini, M., Cerdeira-Pena, A., and Penabad, M. (2017). A framework for gamification in software engineering. *Journal of Systems and Software*, 132:21–40.

Hattie, J. and Timperley, H. (2007). The power of feedback. *Review of educational research*, 77(1):81–112.

Kerzner, H. (2006). *Gestão de Projetos-: As Melhores Práticas*. Bookman editora.

Kitchenham, B. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report. http://www.dur.ac.uk/ebse/resources/Systematic-reviews-5-8.pdf.

Kohl, K., Vasilescu, B., and Prikladnicki, R. (2020). *Multitasking Across Industry Projects: A Replication Study*, page 93–100. Association for Computing Machinery, New York, NY, USA.

Kula, E., Greuter, E., Van Deursen, A., and Georgios, G. (2021). Factors affecting on-time delivery in large-scale agile software development. *IEEE Transactions on Software Engineering*, pages 1–1.

Machuca-Villegas, L., Gasca-Hurtado, G. P., Puente, S. M., and Tamayo, L. M. R. (2022). Perceptions of the human and social factors that influence the productivity of software development teams in colombia: A statistical analysis. *Journal of Systems and Software*, 192:111408.

Margareth, A. R. and Mulyanto, A. (2021). Designing project management application to support developer productivity using goal directed design approach. In *2021 International Conference on Data and Software Engineering (ICoDSE)*, pages 1–6.

Meyer, A. N., Zimmermann, T., and Fritz, T. (2017). Characterizing software developers by perceptions of productivity. In *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 105–110.

Mishra, A. and Misra, S. (2010). People management in software industry: The key to success. *SIGSOFT Softw. Eng. Notes*, 35(6):1–4.

Nastiti, P. and Setyohadi, D. B. (2020). Identification of critical factors on structure and conduct using scp framework: Evidence from indonesian ict industry. In *2020 17th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 24–29.

Ángel Vega-Velázquez, M., García-Nájera, A., and Cervantes, H. (2018). A survey on the software project scheduling problem. *International Journal of Production Economics*, 202:145–161.

Nicolaescu, S. S., Florea, A., Kifor, C. V., Fiore, U., Cocan, N., Receu, I., and Zanetti, P. (2020). Human capital evaluation in knowledge-based organizations based on big data analytics. *Future Generation Computer Systems*, 111:654–667.

Nigar, N. (2017). Model-based dynamic software project scheduling. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, ESEC/FSE 2017, page 1042–1045, New York, NY, USA. Association for Computing Machinery.

Oliveira, E., Conte, T., Cristo, M., and Mendes, E. (2016). Software project managers' perceptions of productivity factors: Findings from a qualitative study. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '16, New York, NY, USA. Association for Computing Machinery.

Paredes-Valverde, M. A., del Pilar Salas-Zárate, M., Colomo-Palacios, R., Gómez-Berbís, J. M., and Valencia-García, R. (2018). An ontology-based approach with which to assign human resources to software projects. *Science of Computer Programming*, 156:90–103.

Qahtani, A. M. (2020). An empirical study of agile testing in a distributed software development project. In *Proceedings of the 2020 3rd International Conference on Geoinformatics and Data Analysis*, ICGDA 2020, page 110–114, New York, NY, USA. Association for Computing Machinery.

Santos, R. E. (2017). Uses, benefits, and limitations of job rotation in software engineering. *ACM SIGSOFT Software Engineering Notes*, 41(6):1–6.

Schloegel, U., Stegmann, S., Maedche, A., and van Dick, R. (2016). Reducing age stereotypes in software development: The effects of awareness- and cooperation-based diversity interventions. *Journal of Systems and Software*, 121:1–15.

Shahzad, F., Xiu, G., and Shahbaz, M. (2017). Organizational culture and innovation performance in pakistan's software industry. *Technology in Society*, 51:66–73.

Shameem, M., Kumar, R. R., Nadeem, M., and Khan, A. A. (2020). Taxonomical classification of barriers for scaling agile methods in global software development environment using fuzzy analytic hierarchy process. *Applied Soft Computing*, 90:106122.

Shen, X., Guo, Y., and Li, A. (2020). Cooperative coevolution with an improved resource allocation for large-scale multi-objective software project scheduling. *Applied Soft Computing*, 88:106059.

Shen, X.-N., Minku, L. L., Marturi, N., Guo, Y.-N., and Han, Y. (2018). A q-learning-based memetic algorithm for multi-objective dynamic software project scheduling. *Information Sciences*, 428:1–29.

Song, X., Liu, J., Cao, Y., and Long, H. (2020). Optimization of human resource allocation problem considering costs reduction and balance in chinese state-owned enterprise. In *Proceedings of the 2020 4th International Conference on Management Engineering, Software Engineering and Service Sciences*, ICMSS 2020, page 211–215, New York, NY, USA. Association for Computing Machinery.

Stylianou, C. and Andreou, A. S. (2016). Investigating the impact of developer productivity, task interdependence type and communication overhead in a multi-objective optimization approach for software project planning. *Advances in Engineering Software*, 98:79–96.

Tam, C., da Costa Moura, E. J., Oliveira, T., and Varajão, J. (2020). The factors influencing the success of on-going agile software development projects. *International Journal of Project Management*, 38(3):165–176.

Wang, L., Huang, M., and Liu, M. (2018). How founders' social capital affects the success of open-source projects: A resource-based view of project teams. *Electronic Commerce Research and Applications*, 30:51–61.

Zaouga, W., Rabai, L. B. A., and Alalyani, W. R. (2019). Towards an ontology based-approach for human resource management. *Procedia Computer Science*, 151:417–424.