

# A Decision Support System for Multiple Team Formation

Felipe Cunha  
Federal University of Campina Grande  
Campina Grande Paraíba Brazil  
felipe.cunha@virtus.ufcg.edu.br

Mirko Perkusich  
VIRTUS  
Campina Grande Paraíba Brazil  
mirko@virtus.ufcg.edu.br

Hygo Almeida  
Federal University of Campina Grande  
Campina Grande Paraíba Brazil  
hygo@virtus.ufcg.edu.br

Angelo Perkusich  
Federal University of Campina Grande  
Campina Grande Paraíba Brazil  
perkusic@virtus.ufcg.edu.br

Kyller Gorgônio  
Federal University of Campina Grande  
Campina Grande Paraíba Brazil  
kyller@computacao.ufcg.edu.br

## ABSTRACT

This work presents a Decision Support System to assist multiple team formation in the context of software development. After analysis of recent works in the literature, it was found that the approaches are still unable to reflect the real needs of the industry, which makes their practical application difficult. Our findings confirm the benefits of our prototype developed to researchers who are interested in comprehending the team formation problem and industry practitioners who may be interested in understanding how Decision Support Systems can support the teams formation.

## CCS CONCEPTS

•Software and its engineering → Software development process management.

## KEYWORDS

Software Teams, Multiple Team Formation Problem, Intelligent Software Engineering.

## ACM Reference format:

Felipe Cunha, Mirko Perkusich, Hygo Almeida, Angelo Perkusich, and Kyller Gorgônio. 2021. A Decision Support System for Multiple Team Formation. In *Anais do 1 Workshop Brasileiro de Engenharia de Software Inteligente, setembro 28, 2021, Evento Online, Brasil. SBC, Porto Alegre, Brasil, 13-28*. DOI: <https://doi.org/10.5753/ise.2021.17280>.

## 1 Introduction

Team Formation Problem consists of a set coverage problem [1] in which a group is selected from defined criteria to perform a task. It can be considered an NP-hard problem [1], [2], [3], in diverse applications such as academic collaborations [4], medical and sports teams [5] [6].

In the context of Software Engineering, the Multiple Team Formation Problem objective is to allocate multiple developers to different projects, which is a challenge because the demands conflict causes the dispute for human resources [7].

Different characteristics group the members on teams. The literature points out the dimensions to help to compose teams. In some cases, the choice of members involves availability, individual cost, and budget constraints [8], [9]. Studies focus on soft skills, that consider behavior, personality type, ability to teamwork, know how to negotiate, proactivity, autonomy, and

control ability [9], [10], [11]. However, technical skills can be efficient to form a software team, due to their technical nature present in the information technology area.

Usually, the teams are allocated manually and based on the experience and intuition of the managers, which can result in a slow process, subject to errors, subjective, tendentious, and high human dependence [12]. Furthermore, the diversity of attributes produces a large number of possible combinations, making it a repetitive process to achieve the best result. Therefore, team formation tasks require the support of tools to automate, to minimize time and errors [2].

To use information from software project management platforms is a way to automate the team-building process. These systems produce data that require use of artificial intelligence techniques, such as the use of Machine Learning, to develop intelligent solutions [13].

The literature points out approaches to team-building. Schall [14] developed a model to analyze software ecosystems by optimization with genetic algorithms. Gilal *et al.* [10] developed a model to compose software teams using the attributes: team role, personality type, and gender. Paredes-Valverde *et al.* [15] used the participation and experiences in previous projects to suggest teams. Costa *et al.* [7] used genetic algorithms to suggest teams for agile projects based on hard skills.

Although alternatives strive to form teams properly, attempts are not a smart way to deal with a scenario of practical application, on an increased scale, and in a real context. What is clear is that the proposals are to define attributes, test composition techniques, and create conceptual models, but human dependence is still strong.

Is fundamental to consider the managers preferences regarding the most effective technical skills about the used project technologies. In addition, the team objectives are diverse, such as starting a new project, training, solve a specific problem, among others. Therefore, skills must be mapped to attend most team purposes. It is also important to develop and validate a tool with high-level information that meets the real needs of managers, is viable and effective to suggest teams. With this, it will be possible to present a concrete solution, as close as possible to the real

world and allow adjustments with minimal change in the quality of teamwork.

In our previous study, we proposed a multiple-team formation model based on the use of a genetic algorithm that uses information from professionals in previous projects to optimize the composition of teams. As a result, our solution achieved an average of 86.4% accuracy against the teams chosen by managers and an average of 75% acceptance against the recommended teams and much faster when compared to the manual allocation process [16].

To enable the applicability of the proposed team recommendation system in industrial environments, we have developed a tool based on a Decision Support System, called Teamplus.

In this article, we focus on key features identified and how information should be presented in a team-building support system. In addition, we focus to demonstrate the Teamplus tool and report a comparison against other systems, aim to contribute to the literature in the area, and for future acceptance in a real industrial environment.

The article is organized as follows. Section 2 presents the materials and methods. Section 3 presents the results and discussions. Section 4 presents the final remarks as well as the main future works opportunities.

## 2 Materials and Methods

This work investigates the design of a tool to support software team formation. The proposed solution is a software tool to assist project managers to compose teams of software developers. The design of this study follows the life cycle proposed by Kontio [17], positing this research into different phases. The present study was developed in the informational and propositional phases [17].

The objective is develop a Decision Support System to automate on level 2 [18] the multiple software teams formation based on technical skills, from gaps and trends identified of literature about software teams formation. To achieve this, we formulated the following research questions (RQs): **RQ1:** What is the current scenario in the literature on solutions, strategy, criteria, and characteristics to the software multiple teams formation problem? **RQ2:** What are the trends and gaps in recent approaches to support the build task of multiple teams in a software development context?

We found that many approaches use automatic selection to compose teams and varied criteria and diverse characteristics, such as effort and cost estimates, professional role and performance, technical and non-technical skills (RQ1).

We identified still that team formation is a critical step in software project management. Most of the studies analyzed highlight that to compose manually teams is a problem that the industry has not yet overcome. The inadequate teams and professionals' dissatisfaction results in missed deadlines, increased costs, and projects with compromised quality (Table 1), in addition to the loss of professionals' important skills. We also

verified that the team formation algorithms and tools are a solution that should be widely discussed. They can present practical results to reduce the companies' problems in the team formation task (RQ2). Table 1 illustrates a review of approaches to support team formation.

**Table 1: Approaches or decision support systems for team formation.**

Approach/ DSS/	Problems	Method	Reference
Singh	Members manual selection to form team breaks deadlines, increase budgets, and is a slow process	Algorithm for automatic select members based on team estimates	[21]
Latorre	Critical impact on projects, but the team formation is treated in the background	Socio-technical framework	[22]
Jana	Attribution of members to form teams is ineffective	Binary mathematical model, based on cost and effort targets	[23]
Gharote	Friction and dissatisfaction in the manual allocation made by managers	Scattered search algorithm	[24]
Arias	Team recommendation is little explored; manual and non-comparative process; it does not analyze individual and collective performance; punctual allocation	Allocation based on capacity, function and historical performance (e.g. frequency, duration, quality and cost)	[25]
Paredes-Valverde	Manual team formation is complex in medium and large companies	DSS based on the semantic similarity between the requirements document and previous projects	[15]
Almeida	The process of assigning software engineers to agile teams is carried out empirically based on the experience and maturity of software professionals or using a self assignment approach	DSS to assigning an agile team considering the technical and social skills of employees	[26]

After analysis of strategies, problems, challenges in literature, and talking to researchers, specialists, and practitioners of the industry, we developed the proposition of a practical and efficient approach.

We analyze strategies in the literature to create the Decision Support System prototype. We identify attribute categories, related to hard skills and teams compose rules.

Our approach seeks to automate the team formation on level 2, which is when the system offers a set of alternatives and humans can accept or ignore to make a decision [18]. The approach aims to optimize the team formation with hard skills (e.g., Java language) of professionals (e.g., senior developer), as a similarity or diversity grouping strategy and varied size. This proposal

defines the basic setting which was used in the prototype (Figure 1).

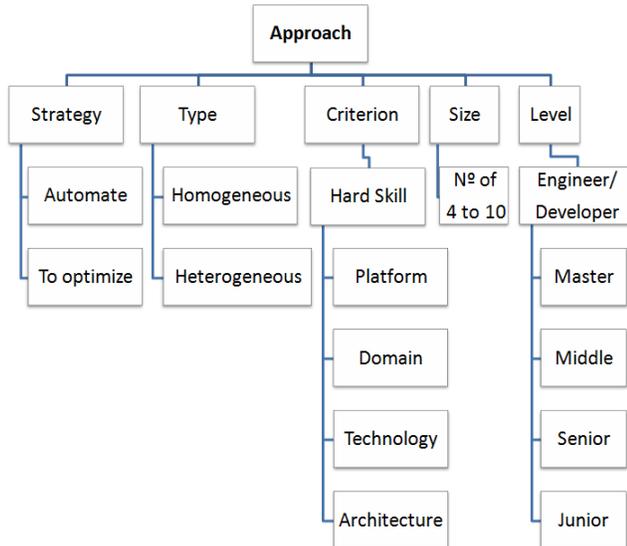
Furthermore, we analyze a database of historical data, created in our previous work [16], with real information about a company's projects, developers, and technologies used at the level of technical tasks originated from Scrum artifacts, a framework for management and agile development of projects software, focused on team concept [19].

Table 2 shows the analysis results with category, description, and tag related to the skills. The tags for each category were used as textual search terms and will be selected as managers' preferences.

**Table 2: Categorization of tags.**

Category	Description	Tag
Platform	Application platform	Web, Mobile, Virtual reality, Embedded, Desktop
Domain	Application domain	Educational, E-commerce, Bank, TV
Technology	The technology used in the Frontend and Backend of the application (e.g. programming languages, database)	Angular, Java, Android, JavaScript, MongoDB, RFID, Typescript, Node, NodeJS, Mongoose, Bootstrap, Express, Bcrypt, JSPDF, Firebase
Architecture	Architecture used in the application	MVC, Client-server, Event-Based, Multi Layer

This analysis contributed to the present study to identify criteria and characteristics to build our approach (Figure 1).

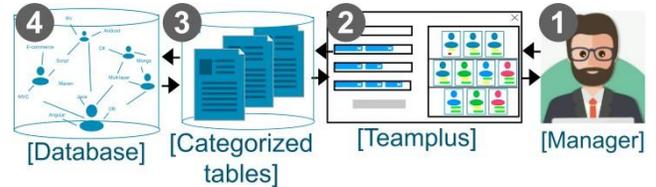


**Figure 1: Decision Support System basic scenario to team formation.**

Therefore, the proposed solution consists of a system linked to the company project management system, so that requests to your database feed categorized tables with main information

professionals. To the user, the tool presents the information contained in these tables in an appropriate format. A Decision Support System must be interactive; sufficiently friendly; present information in a format and terminology familiar to users; and have a selective quantity of information [20].

Figure 2 illustrates a diagram of the proposed solution. The manager (1) adds search tags (2). Then the tool algorithm searches for profiles of professionals related to the tags in tables (3) with organized information from the database (4) and shows the teams suggested (2).



**Figure 2: Scheme of the proposed solution.**

### 3 Results and Discussion

#### Modeling of the proposed system

In software companies, it is common for the board and the most experienced managers to define the final configuration of the teams. Therefore, it was decided that the tool is aimed at them. With the teams' suggestions, reach a consensus on the best possible scenario. This avoids team conflicts.

The tool must cross information to make suggestions. Due to software companies that can be made up of hundreds of professionals, the tool will dispense manual registration. The tool must have useful data tables, linked to the company's resource management system.

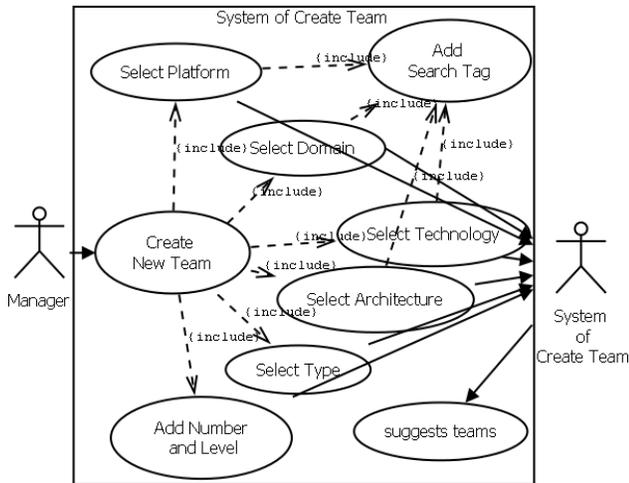
The tool must receive information about projects and professionals. This information is classified about the platform, domain, technology, and architecture. Professionals should be classified in terms of level, function, and others, which may be a process or a new technology.

About the team size, the quantity can be entered individually with levels considered: engineer, developer, senior, master, junior, and middle. It is believed that, in this way, the tool can meet the companies' strategies diversity. Thus it is possible to adopt the configuration of standard teams, for example, a team of five members consists of one senior, two masters, and two juniors.

Staff time varies, with projects typically lasting twelve months. Regarding the team purpose, it was decided not to include it in the approach, because it is not interesting to distinguish the purpose at this moment.

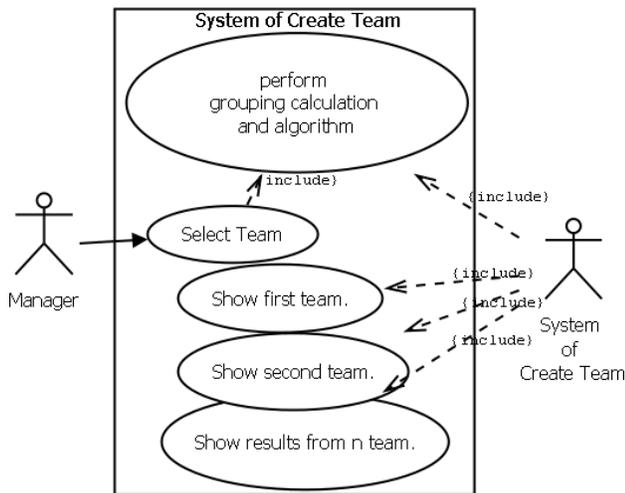
In an application of the Create New Team use case (Figure 3), a project manager starts adding search tags, related to skills about the platform, domain, technology, and architecture. Then, add the number and level of engineers and developers. Finally, select the grouping type: homogeneous or heterogeneous.

Then, the system will search for members that have the requested attributes and will suggest the teams. Figure 4 shows the use-case diagram with this process.



**Figure 3: Use case of the process of creating a new team of the proposed solution.**

The system then will search for members, calculate the grouping function and generate teams. The compositions that have the best results for the function used in the algorithm in relation to the managers' preferences will be presented as a suggestion. The manager chooses the work team that best suits his needs (Figure 4).



**Figure 4: Use cases of the process of selecting a team.**

### Prototype

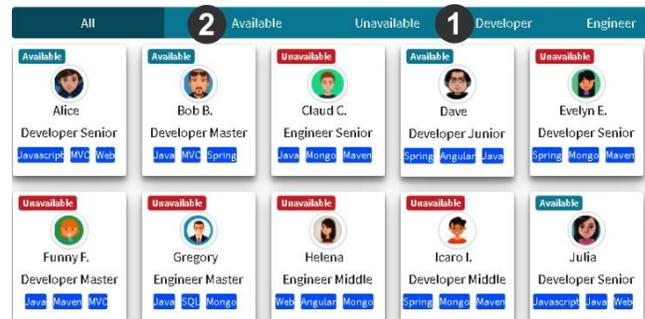
The proposed solution, which is in the prototype stage, is a software tool capable of to suggest teams based on project demands with the available resources in an optimized way. In view of the existence of several scenarios, this proposal seeks to define the basic scenario. A tool prototype was created to predict how the artifact of the proposed solution will interact with the problem context to achieve the stakeholders' goals [27].

In order to make the presentation of professionals in the tool as efficient as possible [20], it was decided to gather the main information in card format, as shown in Figure 5, where you can view information on status (1), name (2), level (3), and skills of the professional (4).



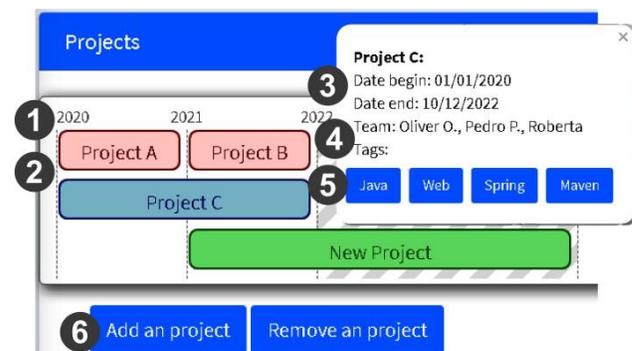
**Figure 5: Sample card with information on the professional's status, level, and skills.**

Figure 6 shows the People screen, where all professionals' cards are presented and viewed according to professional level (1) and availability status (2).



**Figure 6: People screen.**

Figure 7 shows the Projects screen, where all the company's projects are presented in a timeline (1), allowing the manager to view information on execution status (2), dates (3), allocated team (4), and technologies (5) used in the projects, by a popup window. The manager can add new projects (6) and see suggestions from multiple teams to be allocated.



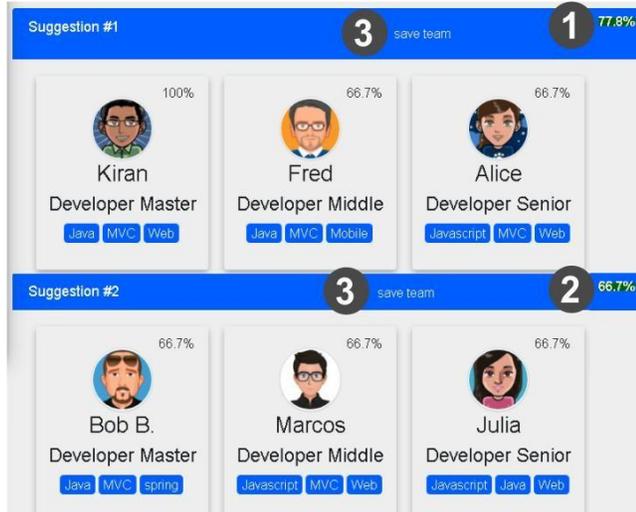
**Figure 7: Projects screen.**

Figure 8 shows the Form Teams screen, where the manager can view suggested teams, modify them, and, when preferred, allocate them to the created projects. On this screen, the manager inserts search tags (1) related to the projects' technologies, which are also related to the professionals' technical skills. After that, the manager defines the number of professionals to compose the teams (2), being possible to enter by level. Finally, it confirms the grouping type (3) according to needs and clicks on Form teams (4).



**Figure 8: Entry attribute fields screen.**

As a result, illustrated in Figure 9, the solution delivers accurate team suggestions that will facilitate the management decision process. With transparency in the technical compatibility found by the algorithm, the idea is to give managers the ability to quantitatively assess the impact of different circumstances. The manager, when necessary, can manipulate multiple parameters and analyze alternative paths to the decision, seeking to increase the performance of teams and the quality of projects.



**Figure 9: Result sample with suggested multiple teams.**

In the example given in Figure 8, see the situation in which the manager, responsible to form teams for a new project, inserts as input the search tags “Java, Web, MVC”, related to the technologies to develop the project. Selects by level three developers: one Master, one Middle, and one Senior so that the teams are composed of members as homogeneous as possible, that is, experts in relation to technologies. After, (Figure 9) the tool will deliver suggestion 1 with 77.8% compatibility (1) and team

suggestion 2 with 66.7% team compatibility (2), regarding to the tags, and allow save (3) the suggestions to new projects.

People can be moved between teams to better fit and preferences can be adjusted before making the best decision. The suggestions chosen by the managers can be added to the projects or, in the case of adaptation to the company's new technologies, be recommended for training.

Therefore, it is clear that the present study presents itself as an important contribution to academic literature, as it enables a better understanding of the real needs of software project team managers. Our solution maintains the independence and freedom for managers to make the right decisions regarding the task of allocating teams. This increases the confidence of our proposal with stakeholders as to its effective use in the industry. This system will be implemented as a web application and, for future case studies, will be linked to the project management system that is already in use in a company. We intend to implement the Teamplus tool using the genetic algorithm because it is a powerful search mechanism and one of the most suitable methods for combinatorial optimization problems [28]. We intend still in future study add non-technical categories.

Table 3 illustrates a comparison of our solution to the works described, the last work corresponding to this research. In addition to the fact that few of the works described are focused on proposing a decision support system, many of them fall into the problem of a solution with high human dependence.

**Table 3: Comparison of approaches to support team building decisions.**

Reference	Solve the problem?	Hard Skill	Low human dependence	friendly [20]
[21]	Yes	No	No	No
[22]	Yes	No	No	No
[23]	Yes	No	No	No
[24]	Yes	No	No	No
[21]	Yes	Yes	No	No
[15]	Yes	Yes	Yes	No
[26]	Yes	Yes	Yes	No
<b>Present study</b>	Yes	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>

## 4 Conclusions

This study presents and describes the process of prototype modeling of Decision Support System to assist managers in the software multiple team formation.

The built prototype helps and improves the process of assigning professionals to teams, taking into account the more

efficient hard skills, low human dependence to use the solution, and shows information at a high level to project managers. To that end, it was considered gaps in the literature on team formation in the software development context, managers preferences, and historical data about a company's projects.

The most important functionalities of the prototype are: to present the main information in a practical format, suitable and familiar to the users, transparency in the technical compatibility between the professionals of the teams, giving the managers the ability to assess the impact of different circumstances given the suggested teams.

The future works, we intend to implement and integrate it in a real industry environment. In addition, we intend to perform a study case to validate the practical use. With that, contribute to the academy-industry collaborations.

## REFERENCES

- [1] Kamal Z. Zamli, Md. Abdul Kader, Saiful Azad and Bestoun S. Ahmed. 2021. Hybrid Henry gas solubility optimization algorithm with dynamic cluster-to-algorithm mapping. *Neural Comput & Applic* 33, (Jan. 2021), 8389–8416. DOI: <https://doi.org/10.1007/s00521-020-05594-z>.
- [2] Theodoros Lappas, Kun Liu and Evimaria Terzi. 2009. Finding a team of experts in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, New York, NY, USA: ACM, 467–476. <https://doi.org/10.1145/1557019.1557074>.
- [3] Said Tkatek, Saadia Bahti, YounesLmzouari and Jaafar Abouchabaka. 2020. Artificial Intelligence for Improving the Optimization of NP-Hard Problems: A Review. In *International Journal of Advanced Trends in Computer Science and Engineering*, 9(5), September - October 2020, 7411 – 7420. <https://doi.org/10.30534/ijatcse/2020/73952020>.
- [4] Paul Bergey and Mark King. 2014. Team machine: A decision support system for team formation. *Decision Sciences Journal of Innovative Education*, 12(2), (Apr. 2014), 109–130. DOI: <https://doi.org/10.1111/dsji.12027>.
- [5] Mohammad Z. Z. Abidin, Mohd K. M. Nawawi and Maznah M. Kasim. 2016. Conceptual Framework Of Decision Support System For Team Sports. *Journal Of Engineering And Applied Sciences*, 11(8), 1788 - 1791. DOI: 10.36478/jeasci.2016.1788.1791.
- [6] Mason Wright and Yevgeniy Vorobeychik. 2015. Mechanism design for team formation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 1050-1056. <http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9902>.
- [7] Alexandre Costa, Felipe Ramos, Mirko Perkusich, Arthur Freire, Hyggo Almeida and Angelo Perkusich. 2018. In *Proceedings of the International Conference on Software Engineering and Knowledge Engineering (SEKE '18)*, Vol. 2018-July, no. June, pp. 474–479. <https://doi.org/10.18293/seke2018-108>.
- [8] Hui Yi Chiang and Bertrand M. T. Lin. 2020. A decision model for human resource allocation in project management of software development. *IEEE Access*, vol. 8, pp. 38073-38081. DOI: <https://doi.org/10.1109/ACCESS.2020.2975829>.
- [9] Fabio Q.B. da Silva, A. Cesar C. Franca, Tatiana B. Gouveia, Cleviton V.F. Monteiro, Elisa S.F. Cardozo and Marcos Suassuna. 2011. An empirical study on the use of team building criteria in software projects. In *Proc. Int. Symp. Empirical Softw. Eng. Meas., Sep.,* 58-67. <https://doi.org/10.1109/ESEM.2011.14>.
- [10] Abdul Rehman Gilal, Jafreezal Jaafar, Mazni Omar, Shuib Basri and Ahmad Waqas. 2016. A rule-based model for software development team composition: Team leader role with personality types and gender classification. *Information and Software Technology*, Vol. 74, (Jun. 2016), 105-113. DOI: <https://doi.org/10.1016/j.infsof.2016.02.007>.
- [11] Margarita André, María G. Baldoquín and Silvia T. Acuña. 2011. Formal model for assigning human resources to teams in software projects. *Information and Software Technology*, Vol. 53, 3 (Mar. 2011), 259-275. DOI: <https://doi.org/10.1016/j.infsof.2010.11.011>.
- [12] Silvia T. Acuña and Natalia Juristo. 2004. Assigning people to roles in software projects. *Softw., Pract. Exper.*, vol. 34, 7 (Jun. 2004), 675-696. DOI: <https://doi.org/10.1002/spe.586>.
- [13] Mirko Perkusich, Lenardo Chaves e Silva, Alexandre Costa, Felipe Ramos, Renata Saraiva, Arthur Freire, Edinaldo Dilonzo, Emanuel Dantas, Danilo Santos, Kyller Gorgonio, Hyggo Almeida and Angelo Perkusich. 2019. *Intelligent Software Engineering in the Context of Agile Software Development: a Systematic Literature Review*. *Information and Software Technology*, Vol. 119, (Mar. 2020), 106241. DOI: <https://doi.org/10.1016/j.infsof.2019.106241>
- [14] Daniel Schall. 2016. Skill-based team formation in software ecosystems. In *Proceedings of the Int. Workshop Qual. Assurance Comput. Vis. Int. Workshop Digit. Eco-Syst.* <http://ceur-ws.org/Vol-1711/paperDECOSYS2.pdf>
- [15] Mario Andrés Paredes-Valverde, María del Pilar Salas-Zárate, Ricardo Colomo-Palacios, Juan Miguel Gómez-Berbís and Rafael Valencia-García. 2018. An ontology-based approach with which to assign human resources to software projects. *Sci. Comput. Program.*, vol. 156, (May 2018), 90-103. DOI: <https://doi.org/10.1016/j.scico.2018.01.003>.
- [16] Antonio Alexandre M. Costa. 2019. *Uma Abordagem de Apoio à Decisão para Formação de Múltiplas Equipes em Projetos Ágeis de Software*. Supervisor: Dr. Hyggo O. de Almeida. Ph.D. Dissertation. Federal University of Campina Grande, Campina Grande, PB.
- [17] Jyrki Kontio. 2001. *Software Engineering Risk Management: A Method, Improvement Framework, and Empirical Evaluation*. Ph.D. Dissertation. Helsinki University of Technology, publisher: Center of Excellence, ISBN: 952-5136-22-1.
- [18] Robert Feldt, Francisco G. de Oliveira Neto and Richard Torkar. 2018. Ways of applying artificial intelligence in software engineering. In *Proceedings of the 6th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering ser. (RAISE '18)*, 27 May-3 June, 2018, Gothenburg, Sweden, 35-41. <http://doi.acm.org/10.1145/3194104.3194109>.
- [19] Ken Schwaber and Mike Beedle. 2002. *Agile software development with Scrum*, vol. 1. Prentice Hall Upper Saddle River, NJ.
- [20] Ramesh Sharda, Dursun Delen and Efraim Turban. 2014. *Business Intelligence and Analytics: Systems for Decision Support* (10th ed.). Pearson, London, UK.
- [21] Avnish Singh Jat, Purtee Kohli and Devpriya Soni. 2016. In *Proceedings of the 4th International Conference on Science, Technology and Management*, May, 2016, New Delhi, India. 584-588. <http://data.conferenceworld.in/ICSTM4/P584-588.pdf>.
- [22] Roberto Latorre and Javier Suárez. 2017. Measuring social networks when forming information system project teams, Vol. 134, (Dec. 2017), 304–323. DOI: <https://doi.org/10.1016/j.jss.2017.09.019>.
- [23] R. K. Jana, Manas K. Sanyal and Saikat Chakrabarti. 2017. Binary fuzzy goal programming for effective utilization of IT professionals. In *Mandal J., Satapathy S., Sanyal M., Bhateja V. (eds) Proceedings of the First International Conference on Intelligent Computing and Communication*, November, 395–405, 2016, *Advances in Intelligent Systems and Computing*, vol 458. Springer, Singapore. DOI: [https://doi.org/10.1007/978-981-10-2035-3\\_40](https://doi.org/10.1007/978-981-10-2035-3_40).
- [24] Mangesh Gharote, Rahul Patil and Sachin Lodha. 2017. Scatter search for trainees to software project requirements stable allocation, *Journal of Heuristics* 23, (Jun. 2017), 257–283. DOI: <https://doi.org/10.1007/s10732-017-9343-z>.
- [25] Michael Arias, Jorge Munoz-Gama and Marcos Sepúlveda. 2016. A multi-criteria approach for team recommendation. In *Dumas M., Fantinato M. (eds) Business Process Management Workshops. BPM, May 06, 2017. Lecture Notes in Business Information Processing*, vol. 281. Springer, Cham. 384–396. [https://doi.org/10.1007/978-3-319-58457-7\\_28](https://doi.org/10.1007/978-3-319-58457-7_28).
- [26] Fernando Almeida, Diogo Adão and Catarina Martins. 2019. Decision Support System for Assigning Members to Agile Teams. In *International Journal of Information Technologies and Systems Approach*, Vol. 12, Issue 2, (July-December 2019), 18 pages. DOI: <https://doi.org/10.4018/IJITSA.2019070103>.
- [27] Roel J. Wieringa. 2014. *Design science methodology for information systems and software engineering* (1st ed.). Springer, Berlin, Heidelberg.
- [28] Emanuel Falkenauer. 1998. *Genetic Algorithms and Grouping Problems*. John Wiley & Sons, Inc., New York, NY.