# A Data-driven Framework to Support Team Formation in Software Projects

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

Thiago Rique
thiago.rique@virtus.ufcg.edu.br
Federal University of Campina Grande
Campina Grande, Paraíba, Brazil

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

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

Hyggo Almeida
hyggo@virtus.ufcg.edu.br
Federal University of Campina Grande
Campina Grande, Paraíba, Brazil

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

## ABSTRACT

**Context** Software analytics approaches have supported managers in making informed decisions regarding several software engineering problems. Team formation is a challenging one, being investigated by the research community as new approaches and tools have been proposed. However, the existing studies do not appropriately address the aspects and procedures to be adopted in the development of tools to meet most scenarios, besides not providing a concrete solution useful from a practical perspective. **Aims** This study provides a framework to support the development of solutions that can help managers form software teams. **Method** We interviewed a key practitioner from a software organization and analyzed the collected data to understand how the team formation problem is currently handled, identifying underlying aspects and challenges faced by the organization. **Results** We presented an overview of the proposed framework and the results of a preliminary evaluation performed by integrating a prototype into an enterprise system. **Conclusions** Our results provide a concrete solution for the team formation problem that can be integrated not only into a project management tool, but also into software analytics tools.

## CCS CONCEPTS

• **Software and its engineering → Software development process management**; • ;

## KEYWORDS

Team Formation Problem, Intelligent Software Engineering, Decision Support System

## 1 INTRODUCTION

Practitioners have used data-driven approaches (e.g., software analytics) to solve several software engineering problems. A challenging one is the Team Formation Problem (TFP), as people directly affect all software development activities. Solving the TFP means finding the most suitable configuration of a team to properly perform development activities and be compliant with the project's constraints [11].

Many approaches in the literature related to the topic use automatic selection to form teams and several criteria and characteristics, such as effort and cost estimates, professional role and performance, and hard and soft skills [12]. Existing studies have made significant efforts to provide a useful approach but they do not appropriately address the aspects and procedures to be adopted in the development of tools to meet most scenarios, making it difficult to build more accurate solutions.

Given this gap, we designed an integrated and data-driven solution for the TFP aiming to provide managers with a more useful approach from a practical perspective. The following research question (RQ) guided our study: *How could a data-driven approach provide more assertive information to help managers form software teams?* Based on the perceptions of a key practitioner from a software organization, this paper presents a framework to support forming teams in software projects, detailing each one of its components.

*ISE 2022, October 4, 2022, Virtual Event, Brazil. 6 pages.*


## 2 BACKGROUND

### 2.1 Software analytics frameworks and tools

Due to the availability of large volumes of software engineering data, practitioners and researchers have turned into software analytics, i.e., the use of analysis, data, and systematic reasoning to help managers and software engineers gain and share valuable insights to make informed decisions regarding several aspects of a project. Obtaining such insights requires some degree of automation combined with human involvement [20].

The research community has developed useful approaches for software analytics. For example, Czerwonka et al. [13] explain the process to develop CODEMINE, a software analytics platform for engineering process data collection and analysis. The Q-Rapids tool is also a great exemplar. It is part of the Q-Rapids project [17], offering software analytics capabilities by integrating quality models to improve quality in agile software development.

## 2.2 Team formation approaches

Software team formation approaches have focused on the use of soft skills [16, 21] and hard skills [4, 5] attributes. Other studies consider personality [15] and social interactions [19]. Also, approaches in Search-Based Software Engineering (SBSE) have used optimization techniques to provide solutions such as Genetic Algorithms [11].

However, obtaining criteria is still a predominantly manual task, which leads to the loss of important historical information of professionals. Given this, recent approaches have proposed software tools to support obtaining attributes from previous project history [2, 8] and source code platforms [1] in an automated manner. Others provide different functionalities such as modifying the team after allocation [22] and showing modification history [2, 8, 22, 3].

# 3 METHOD

## 3.1 Data Collection

We elicited the requirements of our solution by conducting a semi-structured interview [1] with one industry practitioner with over ten years of experience as a manager. The interviewee is the primary person accountable for forming all the software teams in collaboration with the project managers in the organization under study. It executes approximately fifty projects per year in several technological domains (e.g., Web systems, mobile systems, AI, augmented reality, embedded systems, and hardware), focusing on diverse market segments (e.g., security, biometry, and business intelligence). The interview's goal was threefold. First, we aimed to identify the most critical challenges faced by the interviewee in forming software teams. Second, we focused on eliciting attributes he judged most relevant when selecting the team members. Finally, we elicited the desired features of a tool that could support such a task.

## 3.2 Data Analysis

We transcribed the interview and analyzed the resulting text, i.e., corpus, using the qualitative analysis tool IRaMuTeQ [6]. The corpus had 1677 words, i.e., occurrences, and 47 sentences, i.e., text segments (TS). IRaMuTeQ automatically identified four primary themes, i.e., classes discussed during the interview:

- **Class 1:** *The criteria*, with occurrence in 11 TS (22.58%), relates to the importance of combining social and technical characteristics to achieve the project's success, converging with the literature [16] [14], [3];
- **Class 2:** *The granularity of skills*, with occurrence in 12 TS (25.81%), exposes the preference for a high granularity level of the technology to be used by the tool, e.g., programming language.
- **Class 3:** *The goal*, with occurrence in 12 TS (25.81%), refers to the importance of having the tool focusing on the project level and providing flexibility to change the technology needs and team members even after the project has started.
- **Class 4:** *The data organization*, with occurrence in 12 TS (25.81%), refers to the need for integrating with the organization's database to have reliable and consistent information about the projects and professionals.

[1] https://tinyurl.com/yff49dv2

# 4 FRAMEWORK

This section presents an overview of the framework and details each one of its components.

## 4.1 Framework overview

The framework consists of the main components that a data-driven solution architecture needs to support the development of tools for TFP. Figure 1 shows framework's components, namely: Data Sources, Data Collector, Team Formation Engine, and Frontend.

## 4.2 Data Sources

Data Sources are the data origin location such as a project management system in an organization that manages information from people, projects, and tasks. It is responsible for providing an Application Programming Interface (API) to allow integration from third applications to access their data.

## 4.3 Data Collector

Data Collector is an interface, e.g., a Web-based API, for communication between a data source and an external system. Also, the Data Collector is responsible to extract data from Data Sources, and saving a copy of the data as a Historical Database (Figure 1). A historical Database consists of previous project data to help build the developer profile which is composed of capabilities and information about their performance.

## 4.4 Team Formation Engine

Team Formation Engine applies a Genetic Algorithm to optimize team suggestions according to the input settings.

## 4.5 Frontend

The Frontend is responsible for receiving the team settings and showing suggestions. Our TeamPlus Frontend component is presented in the Subsection 5.4.

# 5 PRELIMINARY EVALUATION

This section details a preliminary evaluation performed by integrating a prototype into an enterprise system.

## 5.1 Data Source

For the Data Source, we used the project management system in the organization under study. However, we emphasize that the data source can be any other project management system such as Git[2], Clickup[3], Jira[4], Trello[5], and Jenkins[6].

## 5.2 Data Collector - TeamPlus API

For data collection, we developed an API that uses REST (Representational State Transfer) model to send requests to the Data Source server. First, a request using the system credentials authorizes data collection. Then, the API requests data to the Data Source receives the data, and saves a copy. Figure 2 shows an example of users' data

[2] https://git-scm.com/
[3] https://clickup.com/
[4] https://www.atlassian.com/software/jira
[5] https://trello.com/
[6] https://www.jenkins.io/

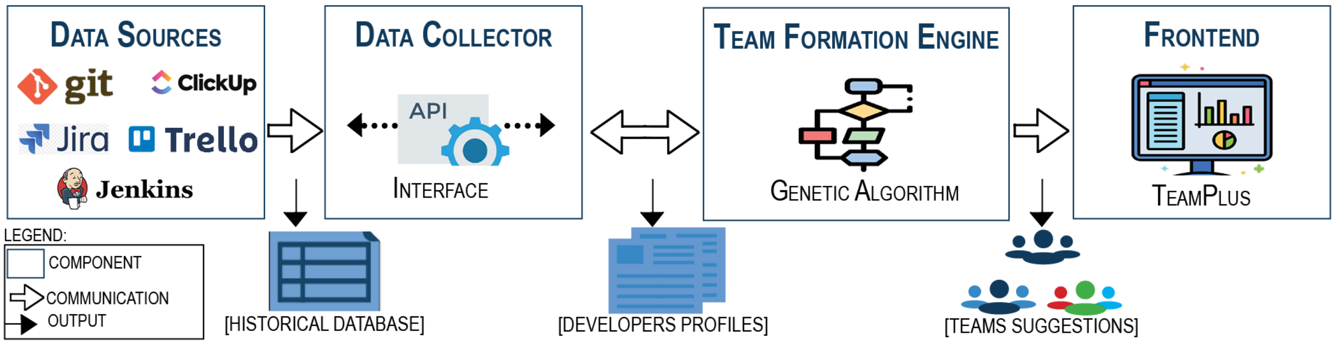


**Figure 1: Data-driven Framework developed.**

received (A) and adjusted (B) in JSON format. Then, we access user items: identifier, profile, level, role, contract id, and project id. This information will be used to model the professional's profile in the format suitable for use by the Engine.

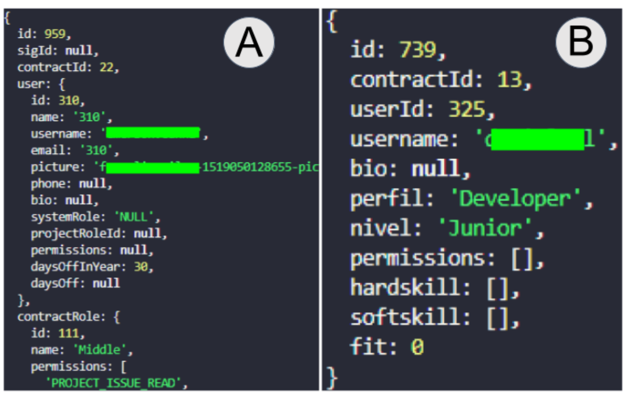


**Figure 2: Sample of data in JSON format.**

Next, the Data Collector component performs the modeling of the professionals' profiles. Based on Costa et al. [10]. The modeling quantifies the technical attributes of professionals from the analysis of tasks. These attributes have the following elements: an id; a key and a value to represent the attribute's tag and its weight.

A vector of n characteristics is used to structure the developer's attributes. Then, the vector of each candidate will be compared to the vector of technologies demanded, which is defined in the Frontend. This similarity comprises the developer aptitude, which represents how fit the candidate is to compose the project team. We used Manhattan Similarity (Equation 1) for presenting good results in the tests performed.

$$ManhattanSimilarity = 1 - \frac{\sum_{i=1}^{k} |u[i] - v[i]|}{k} \quad (1)$$

### 5.3 Team Formation Engine - Genetic Algorithm

GA have been demonstrated the efficiency for team formation [18, 7]. We solution implements the GA based on Costa et al. [9] to operationalize the input data and generate team suggestions. The GA is based on natural selection and genetic reproduction to search for an optimal solution among a set of candidate solutions, using genetic operators of selection, evaluation, crossover, mutation, and population recombination. In the implemented genetic algorithm, each professional has an id, name, profile (e.g. developer), level, and skills vector. Each chromosome is a candidate solution, and, in its structure, each individual m is formed by n ids of professionals, and the set of m individuals forms the population. Figure 3 shows the pseudocode of the GA implemented.

```
Initialize random population
Evaluate the population
While termination criterion is not satisfied{
Perform Elitism
Select chromosomes by roulette procedure
Perform crossover
Perform mutation
Generate new population
Evaluate the population
}
Output the best individuals
```

**Figure 3: Pseudocode of GA.**

The fitness function is the GA mechanism that measures how close a candidate solution is to an optimal one. It calculates the average competence of developers, aiming to obtain the highest possible score based on the competencies required by the project. Our fitness function $f_1(X)$ is presented in Equation 2, where: **X** is the chromosome to be evaluated; **n** is the number of genes of the chromosome; **i** is the gene position in the chromosome structure; $C(d_i, p_i)$ is the competence between the developer and the project in index $i$.

$$f_1(X) = \frac{\sum_{i=1}^{n} C(d_i, p_i)}{n} \quad (2)$$

For the evaluation function, we defined as 10 generations or 80% of the population with the same fitness value. For the elitism, the two best individuals are kept in the next generation (e.g., k=2).

The selection consists of the process of choosing two parents to carry out the crossover. We adopted The Proportional Selection method, where the selection is carried out based on the relative value in the evaluation of individuals in relation to the others. In roulette selection, each individual is associated with a portion of a roulette, proportional to its fitness value. Then, the cumulative probability of each individual is computed so that individuals can be selected by lot. Therefore, a piece of a roulette wheel is assigned to each individual proportional to its fitness. Individuals who are more adapted have a greater chance of being chosen, but randomness is respected in order to maintain the diversity of the population.

The crossover operator consists of a cross of two parent chromosomes that produces a child chromosome. The one-point method was used. A random cut-off point is drawn and the first child comprises the part of the first parent before the cut-off point summed part of the second parent after the cut-off point. The second child will be formed by the remaining parts 4.

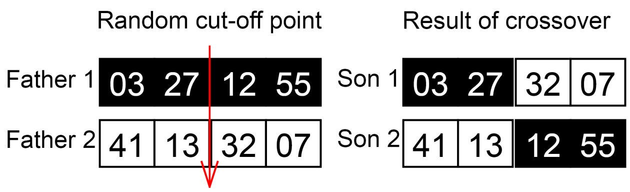


**Figure 4: Crossover.**

For the mutation, the Random Resetting method was adopted. To perform the mutation of two new genes, we followed steps: 1. A number between 0 and 1 is drawn for each gene; 2. If the number is less than or equal to the predetermined probability of 0.5% (0.005), then the operator acts on the gene in question and will be randomly changed by a gene that does not belong to the chromosome, and; 3. the process is repeated for all the genes of the two children.

### 5.4 TeamPlus Frontend

TeamPlus is a web-based tool for automatic software team formation. Its objective is to assist the project manager in allocating teams. TeamPlus' primary requirement is to create, update, and delete a project. However, in contrast to traditional project management tools, its primary capabilities are related to giving support in creating the project team in light of the available professionals in the organization.

Figure 5 shows the use case for creating a new project, that includes the use case "select team". To create a new project, first, the manager must add descriptive information about the project, including its title, nature, and important dates. Next, the manager must add information about the project's needs, including technical skills, soft skills, and the number of team members and their level (Figure 6). Given this, TeamPlus allows the manager to visualize a collection of projects similar to the one created. Further, TeamPlus suggests members of the team (Figure 7) by using a GA. Notice that the manager is free to accept or ignore suggestions or select other team members given whatever criteria he finds valuable. Finally, the manager selects the team members, and the project is recorded in the system's database.

TeamPlus must be accessed via a web browser. On its home screen, the manager enters his credentials, which are the same as those of the people management system (i.e., the organization's

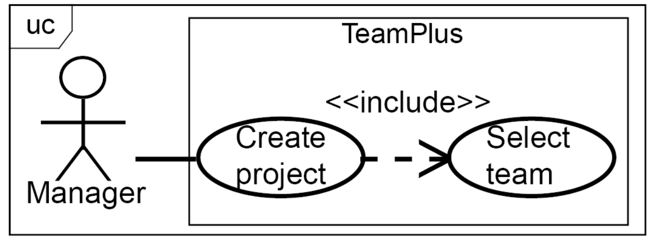


**Figure 5: Use Case - Create a Project**

system with information about human resources). By clicking on the “Enter” button, the application retrieves data from professionals in the enterprise system.

When creating a new project, TeamPlus shows a form to collect general project information and team attributes. Figure 6 shows an example in which the manager enters the search tags “Javascript, MVC, Web, Communication, and Leadership”, and selects the number of developers by level. After clicking on "view suggestions", the tool will deliver suggestion 1 with 66.7% and suggestion 2 with 58.3% in relation to the tags, as illustrated in Figure 7.

The manager can move people between suggestions to analyze different compositions. The team chosen by the manager can be allocated to the project by clicking on "create project". On the project screen, the manager clicks on a project to view its information (1). Next, he clicks on "History" to see the history of project updates (2), and a graph is displayed that can help to compare the number of changes in the team (3), according to Figure 8.

## 6 VALIDATION

We validated the TeamPlus Frontend with one industry practitioner with over ten years of experience as a software development manager. The interviewee is the primary person accountable for forming all the software teams in the organization in collaboration with the project managers. We validated the following functionalities: screens, the granularity of input settings, presentation format, and drag-and-drop function.

On the form screen, the interviewee approved the functionality of searching for skills and technologies through tags, claiming to be more intuitive when compared to checkboxes. Regarding the granularity of the skills, in his view, it is not necessary to give the weighted input of the skills, as it is the algorithm's task. He stated that knowing the nature of the project helps in decision-making with the analysis of teams on similar projects if that is the case. Thus, he approved the functionality of exhibiting projects with similar technologies. Regarding the granularity level of professionals, the participant stated that just considering the levels is enough, and it is not necessary to consider the sub-levels. He requested to add the tester role and project methodology field to increase accuracy.

Regarding the suggestions, the participant approved the format of the presentation of the professionals through cards, which he considered "very good". About the manager moving the cards to adjust the suggestions, for some reason, restriction or particular interest, he approved and stated that "there will always have to be human interaction, it's never an objective thing". The function that shows the percentage of technical compatibility of the suggested people and teams was approved and considered interesting.

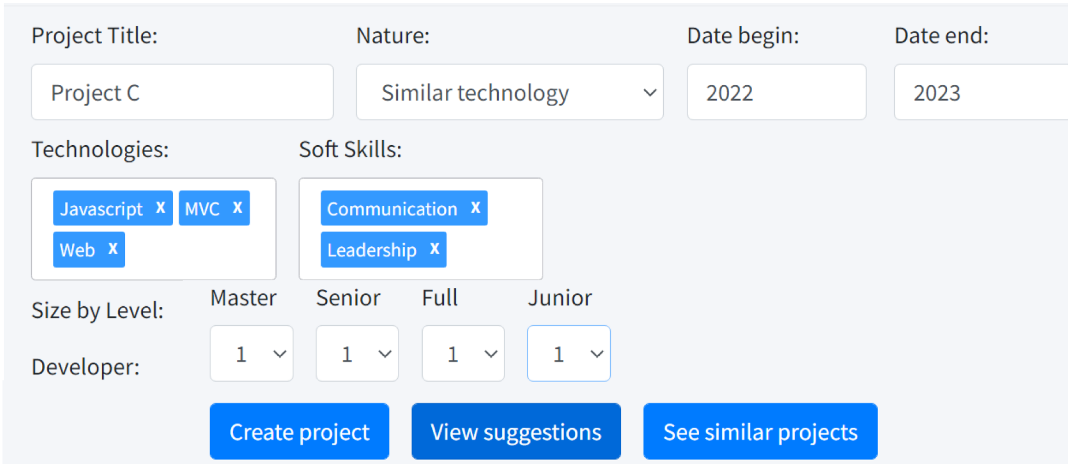


**Figure 6: Entry attribute fields screen.**

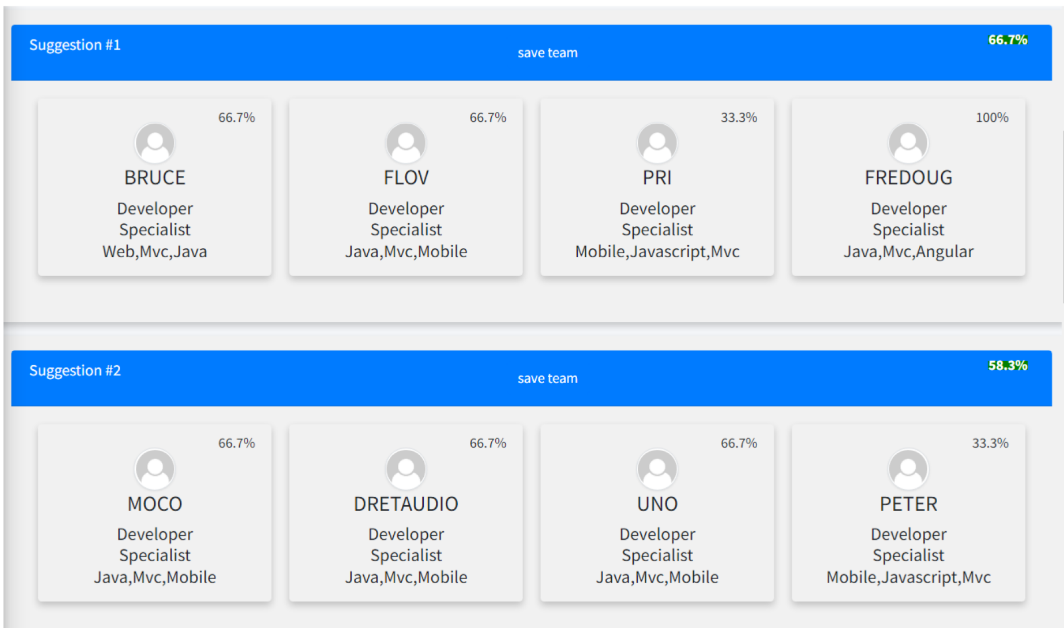


**Figure 7: Result sample with suggestions in format of card**

On the project management screen, when we asked about the functionality of saving change history, he confirmed that it helps to make future decisions to complement the profiles. When we asked about how to know if the team was chosen correctly, he stated that the team suggestion given by the tool must appear as a proposal and, in a second moment, the project manager confirms the total or partial suggestion and the accuracy must be evaluated by the tool. In general, the interviewee stated: "From the point of view of doing it manually, the tool is already solving 60% of doing it manually".

## 7 THREATS TO VALIDITY

Although the proposed solution was evaluated on real-world information and with expert support, it is possible that it does not reflect all the characteristics of the industry. To mitigate this, we intend will carry out an evaluation with other project managers.

## 8 CONCLUSIONS

In this paper, we proposed a data-driven framework to support the development of solutions for software team formation based on needs and experience of one industry practitioner accountable for forming all the software teams in collaboration with the project managers. Our framework comprises four components, namely:

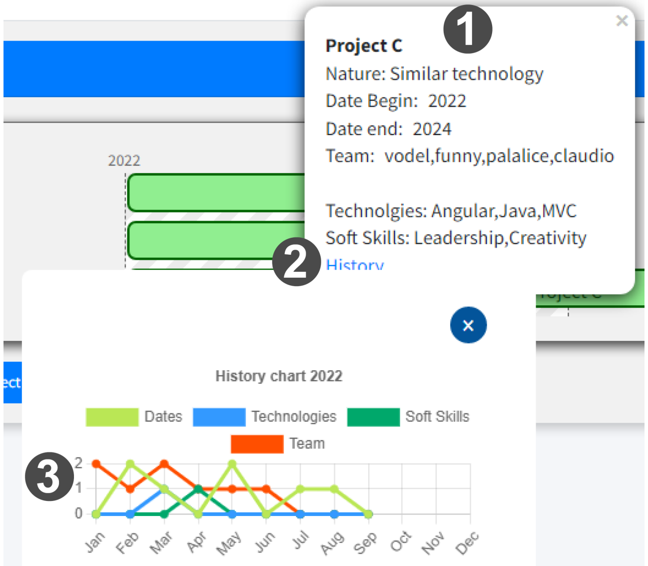


**Figure 8: History chart.**

Data Sources, Data Collector, Team Formation Engine, and Frontend. We performed a preliminary evaluation by integrating a prototype into a project management system of the collaborating organization and implementing a Genetic Algorithm into our Engine. Integration with software analytics tools are also enabled. We validated the features presented by Frontend, which showed that our solution can support project managers in forming teams with less effort and more aspects not commonly found in manual processes. Additionally, our solution helps to increase team accuracy by communicating with the enterprise system and using professionals' relevant data.

Our future work involves adding new features to the proposed solution. We also intend to carry out an evaluation with other project managers.

## ACKNOWLEDGMENTS

This work is supported by grant 1169/2021, Paraíba State Research Foundation (FAPESQ).

## REFERENCES

[1] Shohreh Ajoudanian and Maryam Nooraei Abadeh. 2019. Recommending human resources to project leaders using a collaborative filtering-based recommender system: case study of github. *IET Software*, 13, 5, 379–385. DOI: 10.1049/iet-sen.2018.5261.

[2] Fernando Almeida, Diogo Adão, and Catarina Martins. 2019. Decision support system for assigning members to agile teams. *International Journal of Information Technologies and Systems Approach*, 12, 2, 43–60. DOI: 10.4018/IJITSA.2019070103.

[3] 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*, 53, 3, 259–275. DOI: 10.1016/j.infsof.2010.11.011.

[4] Michael Arias, Jorge Munoz-Gama, and Marcos Sepúlveda. 2017. A multicriteria approach for team recommendation. In *Business Process Management Workshops*. Springer International Publishing, Cham, 384–396. DOI: 10.1007/978-3-319-58457-7_28.

[5] A. Arunachalam, N. P. Nagarajan, V. Mohan, M. Reddy, and C. Arumugam. 2016. Resolving team selection in agile development using nsga-ii algorithm. *CSI Transactions on ICT*, 4, 2, 83–86.

[6] Brigido Vizeu Camargo and Ana Maria Justo. 2018. Iramuteq: um software gratuito para análise de dados textuais. *Temas em Psicologia*, 21, 2, (May 2018), 513–518. DOI: 10.9788/TP2013.2-16.

[7] Hui Yi Chiang and Bertrand M. T. Lin. 2020. A decision model for human resource allocation in project management of software development. *IEEE Access*, 8, 38073–38081. DOI: 10.1109/ACCESS.2020.2975829.

[8] Filipe D. Coelho, Rodrigo Q. Reis, and Cleidson R. B. de Souza. 2019. A genetic algorithm for human resource allocation in software projects. In *2019 XLV Latin American Computing Conference* (CLEI '19). IEEE. DOI: 10.1109/CLEI47609.2019.235055.

[9] Alexandre Costa, Felipe Ramos, Mirko Perkusich, Arthur Freire, Hyggo Almeida, and Angelo Perkusich. 2018. A search-based software engineering approach to support multiple team formation for scrum projects. In *In Proceedings of the International Conference on Software Engineering and Knowledge Engineering* (SEKE '18), 474–479. DOI: 10.18293/seke2018-108.

[10] Alexandre Costa, Felipe Ramos, Mirko Perkusich, Ademar De Sousa Neto, Luiz Silva, Felipe Cunha, Thiago Rique, Hyggo Almeida, and Angelo Perkusich. 2022. A genetic algorithm-based approach to support forming multiple scrum project teams. *IEEE Access*, 10, 68981–68994. DOI: 10.1109/ACCESS.2022.3186347.

[11] Alexandre Costa et al. 2020. Team formation in software engineering: a systematic mapping study. *Ieee Access*, 8, 145687–145712.

[12] Felipe Cunha, Mirko Perkusich, Hyggo Almeida, Angelo Perkusich, and Kyller Gorgônio. 2021. A decision support system for multiple team formation. In *Anais do I Workshop Brasileiro de Engenharia de Software Inteligente*. SBC, 13–28.

[13] Jacek Czerwonka, Nachiappan Nagappan, Wolfram Schulte, and Brendan Murphy. 2013. Codemine: building a software development data analytics platform at microsoft. *IEEE software*, 30, 4, 64–71.

[14] 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 *In Proc. Int. Symp. Empirical Softw. Eng. Meas.* (ESEM '11). (Sept. 2011), 58–67. DOI: 10.1109/ESEM.2011.14.

[15] A. R. Gilal, J. Jaafar, S. Basri, M. Omar, and M. Z. Tunio. 2015. Making programmer suitable for team-leader: software team composition based on personality types. In *Mathematical sciences and computing research (ismsc), international symposium on*, 78–82.

[16] 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*, 74, (June 2016), 105–113. DOI: 10.1016/j.infsof.2016.02.007.

[17] Liliana Guzmán, Marc Oriol, Pilar Rodrıguez, Xavier Franch, Andreas Jedlitschka, and Markku Oivo. 2017. How can quality awareness support rapid software development?–a research preview. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*. Springer, 167–173.

[18] Theodoros Lappas, Kun Liu, and Evimaria Terzi. 2009. Finding a team of experts in social networks. In *In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*. ACM, New York, NY, USA, 467–476. DOI: 10.1145/1557019.1557074.

[19] R. Latorre and J. Suárez. 2017. Measuring social networks when forming information system project teams. *Journal of Systems and Software*, 134, 304–323.

[20] Tim Menzies and Thomas Zimmermann. 2013. Software analytics: so what? *IEEE Software*, 30, 4, 31–37.

[21] Mazni Omar, Bikhtiyar Hasan, Mazida Ahmad, Azman Yasin, Fauziah Baharom, Haslina Mohd, and Norida Mohd Darus. 2016. Applying fuzzy technique in software team formation based on belbin team role. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 8, 8, 109–113.

[22] Constantinos Stylianou, Simos Gerasimou, and Andreas S. Andreou. 2012. A novel prototype tool for intelligent software project scheduling and staffing enhanced with personality factors. In *24th International Conference on Tools with Artificial Intelligence* (ICTAI '12). IEEE, 277–284. DOI: 10.1109/ICTAI.2012.45.