Repository data-based algorithm for selection of product teams of IT specialists

Alexey Zhelepov
Computer Science Department
Ulyanovsk State Technical University
Ulyanovsk, Russia
ORCID: 0000-0003-1197-4401

Nadezhda Yarushkina
Computer Science Department
Ulyanovsk State Technical University
Ulyanovsk, Russia
ORCID: 0000-0002-5718-8732

***Abstract*—Due to the lack of qualified personnel in the IT sector, companies provide their employees with the opportunity to work remotely. That helps even a small company to stand as a global player on the market and recruit new professionals from all over the world. The companies involved in the development of product solutions are interested in hiring cohesive teams of developers who were working together for a long time. However, the HR processes of the company should be restructured and added by additional tools that will help to analyze the entire team's activity and created artifacts. The article contains a detailed description of the MVP that implements searching, selection of project teams based on data from open-source code repositories, and related artifacts. The report describes the algorithm for selecting the main team from the entire set of developers who took part in the development of the project.**

Keywords—repository, remote team, metrics, search, filtering

I. Introduction

In spite of the constant global growth, the Russian IT sector still lacks highly qualified personnel [1]. RUSSOFT, the Russian development company community, has prepared a study that reveals this problem especially for IT companies from regions [2]. Due to the lack of employees, the companies change their work model and give them the opportunity to work remotely. The transition opens up the possibility to hire new developers on the global market.

At the same time, the product development based companies follow the tendency to look for not only individual specialists but the entire teams of professionals [3, 4, 5]. Within the article, such teams are called cohesive teams. The concept of such a collaborative model implies that the group of developers has already been working together on projects, their internal processes and relations were established. The HR hunting of such teams is determined by the features of modern project development such as rapid hypothesis check, MVP development and etc.

The article describes the data source that was applied in research and can be used as a source for the global search of product teams, the basic architecture of components, and the algorithm for filtering the main team and finally the results derived by the algorithm.

II. The Architecture of the Product Team Search System

GitHub is widely used by developers (the service is also well known as their social network) from all over the world to store their projects. It was used as a source of data for the research. The resource contains more than 37 million of users, 100 million of project repositories. The interaction layer between the developed system and the data source was built upon the opened GitHub API [6].

The figure 1 shows the architecture of the developed system.

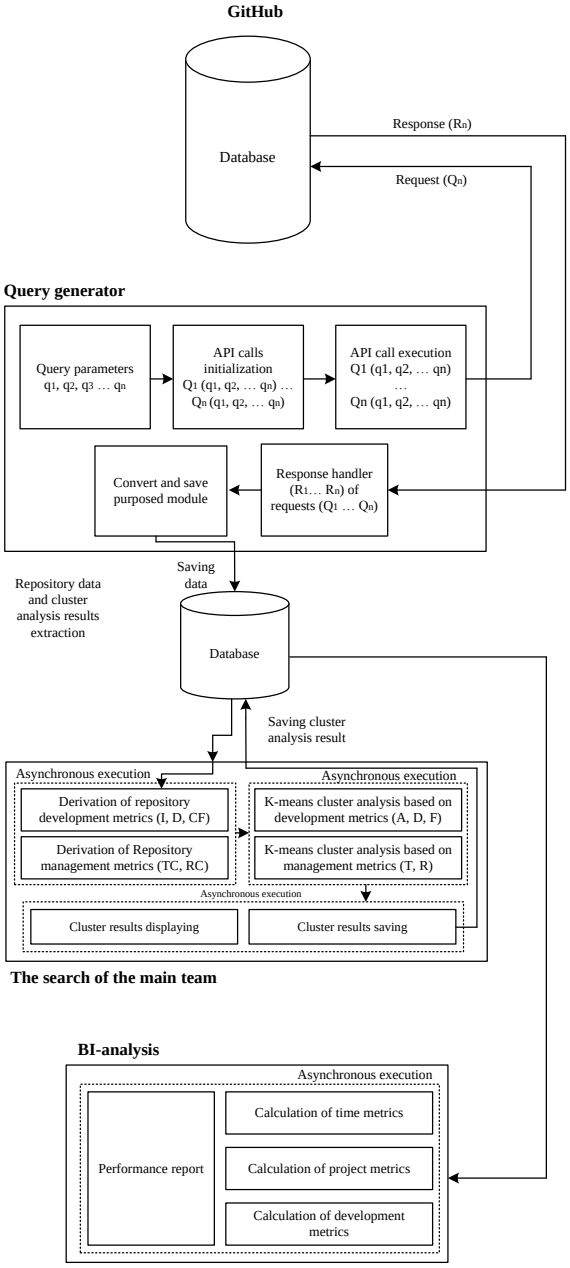

Fig. 1. Architecture of the project team search system.

The designed system consists of 4 modules:

- the query generator $Q(q_1, q_2, \dots q_n)$ that creates API calls for providing further data exchange between GitHub and the system. q_1, q_2, … q_n are query parameters which describe project technologies, count of participants and etc.;

- the search of the main project team. The GitHub repositories are specific due to their public access. That makes them a perfect field for the participation of side developers besides the core team. The main function of the module is filtering and selecting the main team;
- the BI module that allows calculating teamwork performance. It helps to measure teamwork. In this paper, the BI module is only mentioned as a part of a system and its algorithms and visualizations will be presented in future research results;
- the recommendation system for improving teamwork. The module's purpose is to analyze core team members and propose some improvements to their work. This module is still under development.

The query module supports these input parameters:

- q_1 describes technologies that are used in the development of the project. This parameter allows partitioning repositories by technological stacks, which is primarily important for an HR specialist when he looks for the potential product teams. The parameter is represented by a set of string values indicating programming and DevOps technologies: $q_1 \in$ [.NET, Ruby on Rails, Kubernetes …];
- q_2 is a "copy index" of the repository by other developers or project teams. The parameter describes how the project is used by side teams. The range of values is $[0; +\infty]$;
- q_3 describes the index of project's popularity. It is a social parameter of the repository because any developer can leave feedback on the project. The parameter's value bounds within $[0; +\infty]$ and reflects the number of positive reviews left by side developers;
- q_4 contains the name of searched repository;
- q_5 means the name of development team;
- q_6 assumes the number of project's participants and the value lies in range of $[0; +\infty]$.

In the course of further research, it is planned to expand the search capabilities by adding a number of additional parameters. Search is implemented in the Python programming language. It interacts with GitHub services and retrieves necessary data.

III. TEAMWORK AND REPOSITORY MODELS

The project team distinctively clarifies its roles:

- Analysts (A) are team members who identify the needs of project's users and other communities' developers. The analyst decides the way of the project's development;
- Developers (D) are team members who solve the project's problems;
- QA specialists (Q) are team members who are responsible for its quality and workability.
- Team leaders (T) are team members who manage the development process and provide task assessment and distribution between the developers.

It is assumed that one member of a team can fulfill several roles. Thus, the model of the project team can be represented as:

$$M = \{A, D, Q, T\} \quad (1)$$

Each of the roles interacts with the project repository in a certain way.

The analyst role can be accomplished by the side developers or community who are interested in project development. The interaction between them and the core team can be tracked by:

- Tasks (T) that are created by non-core team members;
- Project changes (PR) that are proposed by community members to correct errors for the final project's improvement;
- Social activity of participants, expressed in their comments to the tasks (C), recognition of the significance of the project in the form of positive reviews (R) and their own repository copies represented by forks (F).

The developer role can be separated between the core team and side developers who are interested in project's improvement. The repository describes the role's activities via the next set of characteristics:

- Commits (Cmt) are contributions to the project in the form of lines of program code, documentation and etc.;
- Project branches (B) that reflect the activity of the assigned project tasks execution. The branches are created in order to simplify development synchronization.

QA role is also partly assigned to the community. The repository describes the role via the set of characteristics:

- Tasks (T) that are created to fix errors in the project;
- Merged branches (MB). Before adding to the main branch of the project repository, the completed task should be tested for errors and compliance with the requirements of the task.

The role of the leader can be performed only by the core developer. The distinctive metrics are:

- Released versions (R). While creating a release, the team leader combines many branches (MB) of other developers which is a quite complex task;
- Task assignment (T) for the project's developers.

Finally, the project repository model can be expressed through many relationships of project roles and its artifacts: Rep = { A → (T, PR, C, R, F); D → (Cmt, B); Q → (T, MB); L → (R, T, MB) }.

The model determines the relationship between the interaction of individual project artifacts and the roles of the project team. Based on the presented model description, an

algorithm for identifying the main project team is developed and further presented.

IV. The Algorithm for Selecting the Main Team

The selection algorithm is designed to search for the main development team among all developers that previously made their contributions to the project. The algorithm operates with the data of chosen repository (commits, contributors and etc.). The search is based on the k-means clustering method [7] because of its manual setup of the number of clusters.

Firstly, it was supposed to organize clusters according to JMS model described in the study [8]. The clusters would have been chosen as Junior, Middle, Senior, and indicated the skill level of specialists.

However, a thorough review has shown that such a strategy could not be applied in this field. The repository provides the development statistics: timelines, code, commits and etc. These data perfectly describes only developers who actually make the final contributions. However, the work of their leaders (communications, task and issue management, release management and etc.) stays in shadow. Ultimately, the analysis based only on quantitative metrics and the JMS model will cause a serious deviation.

Considering the features above, the following clusters were chosen:

- Contributor (C), a participant who makes a relatively small contribution to the project;
- Participant (P), a specialist who periodically takes part in project improvement;
- A prospective member of the team (PM), a developer who actively takes part in the project;
- Main developer (MD), a specialist who makes the greatest contributions to the project.

The clustering consists of two stages and based both on development and management metrics [9].

The separate stages help to cover not only plain development metrics dedicated to code writing but to take into account management processes as well.

The development features that were chosen as metrics:

- Insertions and deletions (I, D). The metric describes the number of code lines that were added and removed by each of project's team member;
- Changed files (CF). This characteristic helps to cover the operational range of each team member.

The management features selected as metrics:

- Task Count (TC). This value helps to analyze the ability of team members to correctly understand and determine the project development vector.
- Release Count (RC). It shows the number of prepared released versions made by each of the developers. The metric describes the ability of team members to correlate all contributions into the stable version of the software.

The algorithm consists of the stages:

- [STAGE 1] provides pre-processing and preparation of input data;
- [STAGE 2] conducts clustering procedures for project contributors;
- [STAGE 3] determines the affiliation of each contributor to a particular cluster
- [STAGE 4] visualizes the results.

Before the clustering, it assumes the handling of input data. Thus, the developers are grouped by a number of commits, changed files, created releases, and task quantity.

The input data can be represented via the structures that are described in figure 2.

```
struct label: {
user_id: string
}

struct management {
task_count: int
release_count: int
}

struct development {
additions_count: int
deletions_count: int
changed_files_count: int
}
```

Fig. 2. Structures that describe the input data.

Label structure describes the assignment of values to the contributor. Structures of management and development contain the characteristics for both of clustering analysis (based on management and developer metrics).

The k-means clustering algorithm assumes a predetermined number of clusters. In the case, this number is 4, according to the number of presented clusters: "C", "P", "PM" and "MD".

Another parameter of the clustering algorithm is the maximum number of iterations. In the present case, the value of 10,000 is chosen (obtained experimentally - starting from about 1000 iterations, the centres of the clusters change slightly).

The output of the clustering procedures is presented as a tuple like {label; cluster name}.

V. The Experiment and Results

The algorithm was tested on 10 repositories that describe these well-known projects:

- ClickHouse, a column-oriented database management system created by Yandex. The project's history has more than 30 000 contributions made by 350 developers, including the main team. The team is not remote and sits in the same office.
- Yii 2 Framework which is widely used by PHP developers all over the world. The quantity of commits is about 20 000, the number of participants is 950. The international team is remote and works from all over the world.
- Albumentations and Catalyst. Most of the machine learners favor these frameworks despite their relatively young age. Albumentations is extremely

used by X5 Retail Group in the computer vision projects, Catalyst is mostly an initiative project of several developers.

- 6 other projects of the international product team named Evil Martians: PostCss, BrowsersList, AutoPrefixer, NanoId, Gon, ImgProxy. The repositories are widely used by web developers and have a high rating on GitHub.

The choice of these repositories is explained that their core developers took part in the IT Conference Stachka. One of the article's authors organized this conference. Hence it became possible to make a correct expert assessment of the algorithm results.

The analysis of the projects presented in Table 1. The analysis is based both on development and management metrics.

TABLE I. THE RESULT OF CLUSTER ANALYSIS FOR THE PROJECTS' REPOSITORIES

№	Repository	Number of contributors	Development metrics (K-means)	Management metrics (K-means)
1	Albumenta-tions	46	C: 26 P: 16 PM: 2 MD: 2	C: 1 P: 1 PM: 2 MD: 0
2	PostCss	288	C: 208 P: 77 PM: 2 MD: 1	C: 1 P: 0 PM: 0 MD: 0
3	Yii2	983	C: 875 P: 101 PM: 4 MD: 2	C: 1 P: 1 PM: 1 MD: 1
4	ClickHouse	350	C: 311 P: 32 PM: 6 MD: 1	C: 14 P: 8 PM: 1 MD: 0
5	Catalyst	36	C: 27 P: 7 PM: 1 MD: 1	C: 1 P: 1 PM: 0 MD: 0
6	Browsers List	100	C: 89 P: 8 PM: 2 MD: 1	C: 1 P: 1 PM: 0 MD: 0
7	Auto Prefixer	155	C: 140 P: 11 PM: 3 MD: 1	C: 1 P: 1 PM: 0 MD: 0
8	NanoId	55	C: 41 P: 13 PM: 1 MD: 0	C: 1 P: 0 PM: 0 MD: 0
9	Gon	59	C: 52 P: 4 PM: 2 MD: 1	C: 1 P: 0 PM: 0 MD: 0
10	ImgProxy	31	C: 22 P: 7 PM: 1 MD: 1	C: 1 P: 0 PM: 0 MD: 0

As it is shown in table 1 the distribution of project developers based on management metrics is smaller than on the other ones. Such behavior can be explained as a small percentage of specialists are those who are responsible for crucial decisions in project development (tasks creation, preparing releases and etc.). According to GitHub nature, these developers are certainly the members of core teams who are most interested in the development of project. The figure 3 shows the example of visualization for clustering based on development and management metrics.

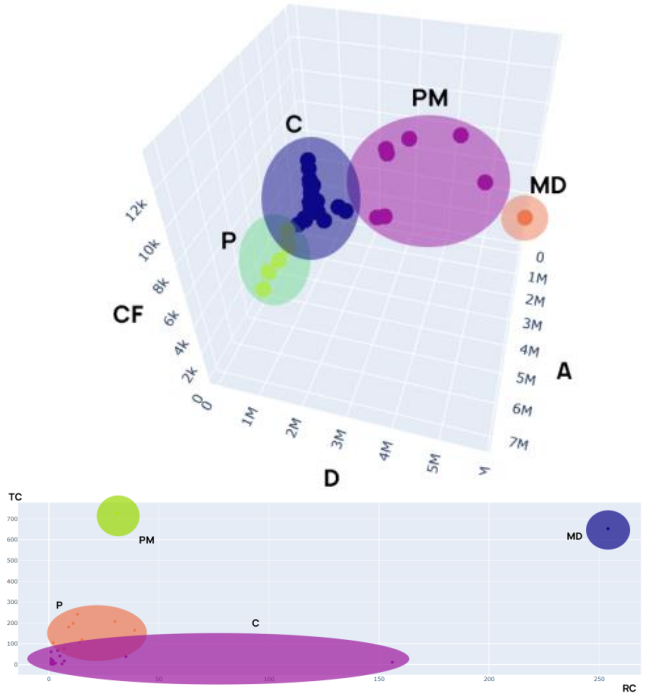

Fig. 3. The development and management metrics based cluster analysis for the repository of ClickHouse.

The core developers of the ClickHouse, Yii 2, Catalyst, Albumentations and members of the Evil Martians Team assessed the practical significance of the algorithm. The results of the intersection of core members identified by the algorithm (P1) and real members (P2) were approved by the experts and shown in Tables 2, 3. The gray color identifies the cases when algorithm was correct.

TABLE II. ALGORITHM GIVEN RESULT (P1) AND EXPERT ASSESSMENT BASED CHOICE (P2) FOR PROJECTS CLICKHOUSE, YII 2, CATALYST, ALBUMENTATIONS

	ClickHouse	Yii 2	Catalyst	Albumentations
P1 (only members of PM and MD clusters)	BayoNet, Ivan Blinkov, Nikolai Kochetov, Vitaliy Zakaznikov, alexey-milovidov, chenxing-x, proller	samdark, NabiKAZ, SilverFire, 十巡洋艦㊣, Carsten Brandt, Qiangxue	Tezikov Roman, Sergey Kolesnikov	Alexander Buslaev, Eugene Khvedchenya, Alex Parinov, Vladimir Iglovikov
P2 (expert assessment)	alesapin, bayoNet, blinkov, 4ertus2, KochetovNicolai den-crane, proller, alexey-milovidov	Qiangxue, samdark, SilverFire, cebe	Tezikov Roman, Sergey Kolesnikov	Alexander Buslaev, Eugene Khvedchenya, Alex Parinov, Vladimir Iglovikov
Intersection of $P_1 \cap P_2$	5 / 8	3 / 4	2 / 2	4 / 4

TABLE III. ALGORITHM GIVEN RESULT (P1) AND EXPERT ASSESSMENT BASED CHOICE (P2) FOR PROJECTS POSTCSS, BROWSERSLIST, AUTOPREFIXER, NANOID, GON, IMGPROXY

	Post Css	**Browsers List**	**Auto Prefixer**	**Nano Id**	**Gon**	**Img Proxy**
P1 (only members of PM and MD clusters)	ai, ben-eb, jedmao	ai, AleshaOleg, akx, An-Tu	ai, bogdan0083, yepninja, Semigradsky	ai	gazay, torbjon, john-bai	DarthSim, koenpunt
P2 (expert assessment)	ai	ai, akx	ai, Semigradsky	ai	gazay	DarthSim
Intersection of $P_1 \cap P_2$	1 / 3	2 / 4	2 / 4	1 / 1	1 / 3	1 / 2

There is no coincidence that results are divided between tables 2 and 3. Table 2 contains projects of high complexity. Table 3 has results for simpler libraries that simplify the development process rather than provide a full-scale solution like projects presented in table 2. However, the algorithm did not include some core developers. That is why the further research will take into account the social component of the development process: soft skills. Its numerical metric will be presented as the number of messages and related discussion sub-messages about proposed solutions to repository issues. Such discussions are commonly leaded by core developers.

The algorithm did not identify some of the core team members and put them into C and P clusters. That occurred because these developers do not make enough changes for the project because of their other kind of activity: support functions, community work and even contribution for other team's projects.

VI. CONCLUSION

This paper describes the architecture of the project team search system, the model of the project team, and its members' roles, the algorithm which provides the search of the core team.

The practical significance of the approach is that it helps to automate the search not only of specialists but of development teams. The HR manager is able to analyze the team's activity and make a decision on whether to hire the found team or not.

Further study will be associated with the development of a recommendation module of the system, which is planned to be built on the basis of the fuzzy logic paradigm [10].

ACKNOWLEDGMENTS

The authors of the study thank the Russian Fund for Fundamental Research for supporting work under grant № 18-47-730019 p_a.

REFERENCES

[1] N. Yarushkina, T. Afanaseva and O. Shiniaeva, "Research of IT-Cluster of Ulyanovsk District," Ulyanovsk, UlSTU Publ., pp. 137-138, 2013.

[2] N. Solovyova, "Russian IT-sphere expects the extreme employee shortage," IT World, 2018 ["Online]. URL: https://www.it-world.ru/it-news/it/140881.html.

[3] D. Sandy Staples, "A Study of Remote Workers and Their Differences from Non-Remote Workers," Organizational and End User Computing, vol. 13, no. 2, pp. 3-14, 2001. DOI: 10.4018/joeuc.2001040101.

[4] N. Yarushkina, G. Guskov, P. Dudarin and V. Stuchebnikov, "An Approach to Similar Software Projects Searching and Architecture Analysis Based on Artifcial Intelligence Methods," Proceedings of the Third International Scientifc Conference Intelligent Information Technologies for Industry (IITI). Advances in Intelligent Systems and Computing. Springer, Cham, vol. 1, pp. 341-352, 2018.

[5] E. Gil, "High Growth Handbook: scaling startups from 10 to 10.000 people," Stripe Press, pp. 95-97, 2018.

[6] Y. Vasin and Y. Yasakov, "Distributed data management system for integrated handling of geolocation data," Computer Optics, vol. 40, no. 6, pp. 919-928, 2016. DOI: 10.18287/2412-6179-2016-40-6-919-928.

[7] K. Kaur, K. Minhas, N. Mehan and N. Kakkar, "Static and Dynamic Complexity Analysis of Software Metrics," World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering, vol. 3, no 8. pp. 1936-1938, 2009.

[8] T. Afanasyeva, A. Zhelepov and I. Zagaichuk, "Framework for Accessing Professional Growth of Software Developers, ICCTA, 5th International Conference on Computer and Technology Applications, 2019.

[9] Y. Li and H. Wu, "A Clustering Method Based on K-Means Algorithm," International Conference on Solid Devices and Materials Science, 2012.

[10] N. Yarushkina, "Methods for Fuzzy Expert Systems in Intellectual CAD-Systems," Saratov, 1997, 106 p.