

# Towards Designing of Recommendation System for Recruiting of Software Development Teams

Karina Melnyk, Natalia Borysova, Zoia Kochuieva and Dina Huliieva

*National Technical University "Kharkiv Polytechnic Institute", Kirpichova, 2, Kharkiv, 61002, Ukraine*

## Abstract

The problem of recruiting software development teams based on the choice of potential candidates is considered. An analytical review of methods for creating recommendation systems has been conducted. A suitable similarity measure based on the type of input data of the domain area has been chosen. The recruiting task of software development teams in an IT-company is formalized. The functional model of business-process of recruiting of software development teams has been developed. The method of resolving the recruiting task has been proposed. The algorithm for solving the issue is given. The proposed microservice architecture of the information system, which has been used for solving the problem of automation of the personnel selection process, has been improved at the expense of creating and adding a new microservice to solve the recruiting task of software development teams. The architecture of recommendation system in the form of deployment diagram and database model are developed. The team building process according to technical appraisal is demonstrated. The evaluation process of the quality of the team building is described. The set of quality metrics and the convolution criterion are proposed.

## Keywords 1

Recommendation system, team building, software development teams, content-based filtering approach, similarity measure, quality assessment

## 1. Introduction

Recommender systems (RS) long ago have taken their rightful place in various spheres of human life and activity. They are used to provide the user with goods, services, information, etc. The conceptual framework of such actions are most relevant search requests, the analysis of user preferences, the analysis of the goods, services, information, etc. However, the potential of recommendation systems in the field of project management has not yet been fully disclosed. There is no doubt that now there are many software products, which allow automating some of the routine actions of a project-manager (PM) and head resource manager (HR). Some of them, such as Applicant Tracking Systems, are used to search for candidates for vacant company positions [1, 2], others are aimed directly at the project management process when a software development team has already been formed [3, 4]. Nevertheless, the recruiting task of teams remains without support of RS.

Today, team building problems are mostly solved by HR-managers manually. The research [1] proposes a solution to the task of personnel selecting for team building according to the technical appraisal. As a result, a list of candidates with suitable technical skills for each vacant position is formed without taking into account psychological characteristics. Further, HR and PM conduct face-to-face interviews with each potential applicant to assess the candidate's communication, leadership and personal qualities. However, it is a time-consuming process. Therefore, some HR-managers use

---

CMIS-2021: The 4th International Workshop on Computer Modeling and Intelligent Systems, April 27, 2021, Zaporizhzhia, Ukraine  
EMAIL: karina.v.melnyk@gmail.com (K. Melnyk); borysova.n.v@gmail.com (N. Borysova); aliseiko@gmail.com (Z. Kochuieva);  
dgulieva@ukr.net (D. Huliieva)  
ORCID: 0000-0001-9642-5414 (K. Melnyk); 0000-0002-8834-2536 (N. Borysova); 0000-0002-4300-3370 (Z. Kochuieva);  
0000-0001-8310-745X (D. Huliieva);



© 2021 Copyright for this paper by its authors.  
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).  
CEUR Workshop Proceedings (CEUR-WS.org)

psychological techniques for recruiting software development teams as additional stage of appraisal. This approach allows realizing the intellectual and creative potential of employees in accordance with the strategic goals of the IT-company. Many authors have researched the problem of team building based on psychotypes, for example, Robert Hogan, Bruce Tuckman, Dr. Meredith Belbin, D. Keirse [5-7]. The task of automation of the process of recruiting software development team based on their members' psychological compatibility is highly relevant. There are many benefits of this approach. Firstly, it will speed up the decision-making process on this issue, because a computer takes much less time to process large amounts of information than a human. Secondly, automating this process will free up the PM's and HR-manager's time for other tasks. Therefore, the main point of this research is dedicated to the development of a recommendation system with particular attention to the function of the recruiting of software development team based on their psychological compatibility.

## 2. Formal problem statement

Let's consider the formulation of the recruiting task of software development teams. Input data for this task is information about the psychotypes of potential candidates. Designate  $C = \{C_1, \dots, C_n\}$  as a set of potential candidates with suitable technical skills for the software development teams from employees and non-employees. The set has formed as result of resolving the personnel selection task according to [1]. The psychological portrait of a potential candidate is a set of competencies that was built based on the model for superior performance developed by L. M. Spencer, such as result orientation, analytical thinking, search for information, etc. [8].

Let  $T = \{T_1, \dots, T_m\}$  is a set of possible teams. Therefore, the task of team building based on psychotypes is a mapping of one set to another  $f: C \rightarrow T$ . If cardinality of the  $\|T\| = 1$ , then the candidate can belong to several variations of one team; if cardinality of the  $\|T\| \geq 2$ , then the candidate can belong to several variations of every team. The final decision depends from the goals of the IT-company. For instance, the team  $T_m$  is more important for the IT-company compared to the team  $T_{m-2}$ , that means the best candidate, who fits both teams, has to belong to the team  $T_m$ .

So, the purpose of the article is to resolve the recruiting task of software development teams using the recommendation system.

To achieve the goal of the given research, it is necessary to:

- conduct analytical review of methods for creating recommendation systems;
- choose a method for recommendation system for team building;
- develop a method for the recruiting task of software development teams;
- conduct numerical researches;
- assess the proposed approach of team building.

## 3. Literature review

There are some traditional approaches to creating of RS [9, 10]:

- content-based filtering approach;
- collaborative filtering approach;
- hybrid approach.

Content-based filtering approach provides recommendations based on comparing information about the user and the object of recommendations. Firstly, a profile of users and recommendation objects are built using the same criteria and the similarity measure of these profiles is estimated; then the closest objects to the user's profile are selected for the recommendation.

Collaborative filtering approach provides recommendations based on comparing information about the preferences of different users with different characteristics and makes recommendations to new users with similar characteristics. The characteristics of recommendation objects are not taken into account in this approach.

Hybrid approach combines both of the previous approaches, trying to use their advantages and neutralize disadvantages.

To solve the given problem, it is efficient to use a content-based filtering approach. A fully formed profile of the employee is compared with the profile of the “ideal” team member for this project or with a set of requirements for a vacant position. Such problem statement moreover allows avoiding the disadvantages of the content-based filtering approach: the limited content analysis and over-specialization, because the profile of the “ideal” team member and real candidates should not contain any gaps.

An analysis of various sources of information has shown the methods used for content-based filtering have been divided into two main groups [9, 10]. The first group of methods calculates different similarity measures between the considered objects. The second group of methods is data mining methods, which involves the use of different machine learning methods. The choice of the method in this case depends on the domain area, the available statistical material and the power of the computing system. Input data for solving the task of recruiting of software development teams is the psychological portrait of some potential candidate to appropriate job. To resolve the given task, it is necessary to compare “ideal” portrait and information about a real candidate. It means to calculate the similarity between two objects. Thus, the next stage is choosing the similarity metric corresponding to the input data of the considered domain. There are many such metrics [11-12]. To find a degree of similarity between quantitative data, there are many metrics: the Euclidean measure, the Manhattan measure or City block distance, the Mahalanobis measure, the Chebyshev measure, and others. The presented metrics can also be used for interval data. For qualitative information, we can use the Hemming similarity measure, the measure of Rogers-Tanimoto, Rao coefficient. The distance between objects described by binary or dichotomous data can be measured using the Pearson metric or square contingency coefficient, the Jacquard coefficient, and the metrics used for qualitative data: the Hamming measure and the Rogers-Tanimoto measure. For mixed data, it is recommended to use Gower’s coefficient, Zhuravlev’s metric, Voronin’s similarity measure, Mirkin’s metric.

According to the L. M. Spencer’s model, each characteristic from the psychological portrait of some potential candidate is measured from one to ten, depending on the degree of manifestation of a particular characteristic. Therefore, it is proposed to calculate only the similarity measure for quantitative data. Let’s consider the most commonly used metrics [11-12], where  $C_k = \{x_1, \dots, x_n\}$  is a  $k$ -th candidate with psychological portrait  $\{x_1, \dots, x_n\}$  and  $C_{ideal} = \{x_1', \dots, x_n'\}$  is a “ideal” portrait accordingly:

1. Euclidean distance. It is the most common distance function, that represents geometric distance in multidimensional space:

$$\rho(C_k, C_{ideal}) = \sqrt{\sum_{i=1}^n (x_i - x_i')^2} \quad (1)$$

2. Squared Euclidean distance. It is used to give more weight to objects that are more distant from each other:

$$\rho(C_k, C_{ideal}) = \sum_{i=1}^n (x_i - x_i')^2 \quad (2)$$

3. Manhattan distance. The distance between two points measured along axes at right angles. For this measure, the influence of individual large differences (outliers) decreases (since they are not squared):

$$\rho(C_k, C_{ideal}) = \sum_{i=1}^n |x_i - x_i'| \quad (3)$$

4. Chebyshev distance. This distance can be useful when we need to define two objects as “different” if they differ in any one coordinate:

$$\rho(C_k, C_{ideal}) = \max_i |x_i - x_i'| \quad (4)$$

An analytical review of the proposed similarity metrics (1)-(4) from the entire arsenal of existing metrics and features of the domain area allows choosing the Euclidean distance as a measure for this research.

## 4. Materials and methods

### 4.1. Formalization of the team building process

The task of recruiting of software development team for the particular project is a multi-criteria task with a large number of restrictions and requirements. Any client always wants to receive the end-product as soon as possible with maximum benefit and with limited resources, despite the requirements of the stakeholders for the product can be contradictory. That is why the project planning and recruiting task of team for this project are very important components of the successful implementation of a quality product. To implement an effective business process in order to achieve the maximum result in team recruiting, it is necessary to create a model of this process. There are many techniques for modelling different business process, such as methodology IDEF0, BPMN Specification, Flowcharts diagrams, Data Flow diagram, notation eEPC. To formalize given business process, it is proposed to use Business Process Modeling Notation (BPMN). The BPMN-diagrams allow describing business processes in a unified standardized language that is understandable to all stakeholders, regardless of their level of technical knowledge. Let's consider the process of team recruiting in more detail way (Fig. 1).

The team recruiting process begins with the formation of a set of requirements that apply to the qualitative and quantitative composition of the team. In order to draw up a list of requirements for potential candidates for the software development team, it is necessary to hold a kickoff meeting. Project Scope and Project charter should be formed during this meeting. These documents are the basis not only for team requirements, but also for software requirement specification. The requirements to technical skill for the competence profiles of IT-specialists for this project are drawn up according to the wishes of the PO and the existing working conditions in the IT-company. The next step is the recruiting process of potential members of software development teams. The HR-manager firstly checks the resumes of those employees who are currently in the personnel reserve.

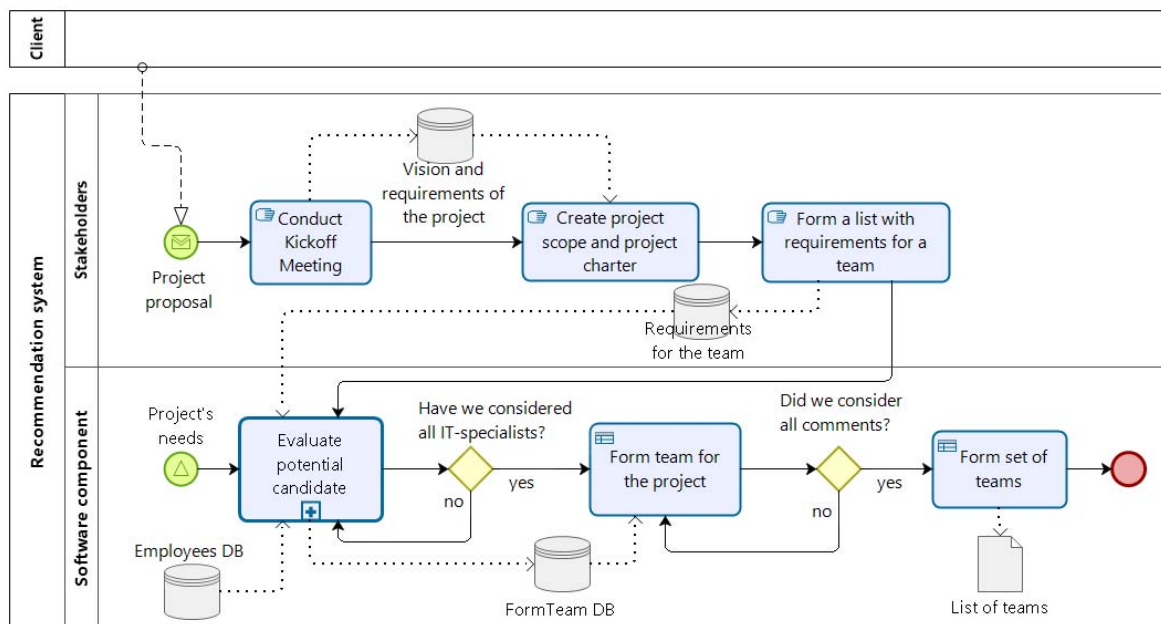


Figure 1: The functional model of the team building process

Next step is checking the employees who are currently included in projects from clients. Sometimes there is a need for a particular IT-specialist who has been already allocated to another project. In this case, this employee can be hired like a consultant. If a proper candidate were not find in the company, HR-manager would search for potential candidates in the Internet. The process of evaluating an IT-specialist can also be initiated due to the immediate needs of the project, for example, in case of dismissal or illness of one of the team members. The task of evaluating potential candidates for the software development teams is considered in [1, 13-15]. As a result of this stage,

potential candidates are screened and set of candidates are formed for each position in the project team. The next step is to solve the problem of team recruiting for new project. Depending on the needs of the PO, it could be recruited one or more teams with different team members.

Let's consider the task of team recruiting in more detailed way.

## 4.2. The method of resolving the recruiting task of software development teams

The team building task begin with defining the team structure. The management of the IT-company assign the PM for a particular project. The PM and the Product Owner (PO) further determine the number of people in the team and the distribution of roles between them according to the project budget.

Let's denote  $R$  as a set of different roles in a team, for example,  $r_i$  – development leader,  $r_k$  – back-end developer,  $r_l$  – quality assurance engineer. Then  $R_i, i \in R$  is the number of team members of  $r_i$ -th role in the team. So, all values of  $R_i$  are the input data from PM and PO.

The result of solving the personnel recruiting task problem is group of sets with potential candidates for each role in the project team [1]. If  $C = \{C_1, \dots, C_n\}$  is a set of potential candidates for the software development teams from employees and non-employees, than  $C_i$  – is a set of potential candidates for the  $r_i$ -th role in the team.

Let denote  $K$  as set of competencies for the psychological portrait of an IT-specialist, then  $x_k$  – is a value of the  $k$ -th competence. This variable has view  $x_k^{pi}$  for a particular  $p$ -th ( $p \in C_i$ ) specialist of  $r_i$ -th role in the team.

The algorithm for solving the recruiting task of software development teams is a step-by-step filling out the template of the hierarchical structure of the team based on proper candidates according to the results of calculating the similarity measure. Therefore, it is necessary to compute the similarity metric between potential candidates. Designate  $s_{ij}^{pq}$  as the similarity measure of  $p$ -th and  $q$ -th potential candidates between the  $r_i$ -th and the  $r_j$ -th roles in the team,  $p \in C_i, q \in C_j$ . If  $i = j$ , than  $s_{ii}^{pq}$  is a similarity measure between people with similar role in the team. The Euclidean measure as a similarity measure between candidates has been proposed in this study. It means the similarity measure can be found according to (1) as follows:

$$s_{ij}^{pq} = \sqrt{\sum_{k \in K} (x_k^{pi} - x_k^{qj})^2}, p \in C_i, q \in C_j. \quad (5)$$

Let's consider the common method of team building for a project under sufficient financial, human and time resources.

Step 1. The choosing of PM.

The Project Manager is the most important and responsible specialist for the project, who is the connecting link between the client and the team members. Therefore, he/she is the basis for the selection of the rest of the team. The formation of the team is carried out according to the principle of choosing the candidates who are closest to the Project Manager in terms of psychological characteristics.

Step 2. Appointment of specialists of the next management level: Development Leader, Quality Assurance Leader, Architect.

There are three variants:

- There are proper IT-professionals who are immediately assigned to selected roles.
- This option provides several potential specialists for one position. At the same time, the PM is ready to work with any of them because of the past positive mutual experience and communication. In this case, there is no need to calculate the similarity measure. The decision about the appointment to the position is made after the possible teams have been selected and analyzed.

- If employees and non-employees apply for the position, then the choice is based on calculating the similarity measure between the PM and possible candidates using the formula (5). The obtained set of values of the metric  $s_{ij}^{pq}$  is sorted in ascending order. The best candidate is determined as follows:

$$candidate_{best} = \min_{C_j} \{s_{ij}^{pq}\} \quad (6)$$

where  $p$  is the Project Manager in this case,  $q$  is any person from the set  $C_j$ .

In the face of uncertainty, three lists of candidates are formed: the first list corresponds to the group of people applying for the Development Leader role, where the top names from the list belong to people with the highest priority, the second list being candidates for Quality Assurance Leader, and the third one for Architect.

Step 3. Assessment of the set  $C_k$  ( $C_k \subset C$ ).

Let's consider the set of potential candidates  $C_k$  for the  $r_k$ -th role in the team. All following actions depend on the results of comparing the cardinality of the set with the number of required employees:

$$action = \begin{cases} |C_k| < R_k \Rightarrow go\ to\ HR - manager \\ |C_k| = R_k \Rightarrow C_k \subset \forall T_d (d = \overline{1, m}) \\ |C_k| > R_k \Rightarrow go\ to\ step\ 4 \end{cases} \quad (7)$$

The number of potential candidates is not enough to fill the vacancy in the first case, so it is necessary to return to the previous stage to solve the problem of personnel selection. The second option describes a situation when the candidates' set of  $r_k$ -th role is a part of all variations of the future team. The third case implies the resolving of the recruiting task.

Step 4. Recruitment of specialists for the  $r_k$ -th role in the team.

Similarly to step 2, the similarity measure between the top-level manager (Development Leader, Quality Assurance Leader, Architect) and the corresponding candidates from the set  $C_k$  is calculated using formula (5). The choice is also carried out from a ranked list according to the formula (6). If there is a need to select several candidates, then they are selected with the smallest values of the similarity measure.

Step 5. Checking the team.

If the team is not formed, then go to step 3. If the team is fully completed, then go to step 2 to form an alternative team. If all possible options are considered, then RS has to provide a list of possible teams for evaluation and analysis to PM.

Thus, the method of forming of project's team based on the psychological portrait of candidates has been considered.

### 4.3. Architectural solution of the recruiting task of software development teams

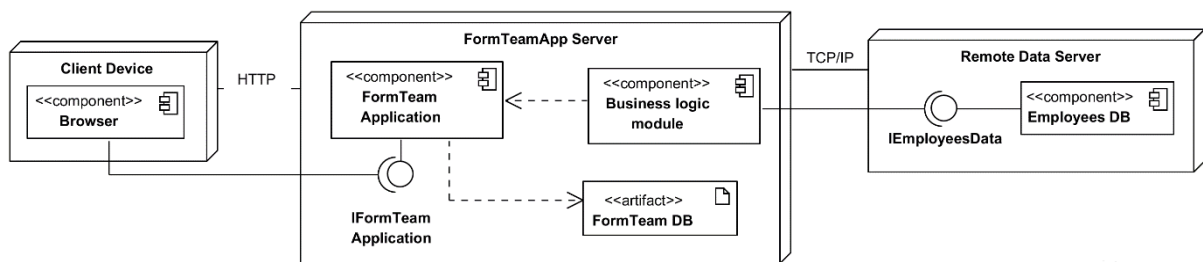
The software design always starts from the defining of the functional and non-functional requirements. Let us look more closely at the functional requirements for RS. The main user of RS is PM. He or she has the ability to view the lists of candidates for each vacant position in the project. These lists are formed by IS after solving the task of personnel selection [1]. PM can view the information about each candidate available in the system; can find a group of people according to proximity based on psychological characteristics. There is a possibility to assign a Development Leader, a Quality Assurance Leader and an Architect. The RS allows recruiting or assigning candidates to teams for the selected Development Leader, Quality Assurance Leader and Architect. The PM can assign a particular team from the list to a project. The RS can create a report with list of generated teams. Non-functional requirements include an intuitive and user-friendly interface, reliability of data transfer and storage, usability, and high performance. An important requirement is

the fact that the forming task of the software development teams uses the results of solving the recruiting task [1]. Thus, RS has to coordinate its actions with IS from [1] or be a component of it.

The research [1] has shown the feasibility of using the microservice architecture as a distributed system architecture based on the analysis of existing IS for the solving various tasks related to the assessment and selection of personnel. This approach shows how to build the software as a set of small, asynchronously interacting services, which implements one of the tasks from the list of ones, which are formulated in the analytical review of the subject area and solved using the developed models. The architecture of IS from [1] consists of several microservices:

1. Microservice Employee – this component allows processing information about employees in the IT-company.
2. Microservice Candidate serves for finding and monitoring potential candidates for vacant positions.
3. Microservice Estimation allows assessing the competencies of employees and candidates.
4. Microservice Identify – a database that contains authentication information about each user of the system.

To solve the problem of forming the software development team, it is necessary to add other microservice, which will form teams with potential candidates. Microservice architecture allows using different technologies to solve each problem. Therefore, the following architecture of the microservice for the recruiting task has been proposed. It is presented in the form of a deployment diagram (Fig. 2).



**Figure 2:** Deployment diagram for the software component

A three-tier architecture to deploy the software component has been chosen. The advantages of the chosen architecture are scalability, high level of security and reliability, low requirements for network speed between clients and the application server, and low requirements for performance and technical characteristics of clients. Let's consider the presented architecture in more detail.

The Client node includes the following components: Web browser and JQuery – JavaScript framework. The FormTeamApp Server node is an Internet Information Services server, which allow the application to work. There are many components on this node: FormTeam Application is a web application, Business logic module is a component for implementing the logic of work between domain objects, FormTeam DB is a component that contains information about formed teams with potential candidates, IFormTeam Application is an application interface. The Remote Data Server node is a system database location node that uses the SQL Server database. This node deploys the database, which is used by the IIS. The FormTeam DB database for storing, changing and processing information about the subject area, that is, about employees, potential candidates, projects, etc., was developed. The model of this base is represented using the IDEF1x notation (Fig. 3).

Database consists of 13 entities:

- the entity «Employees» contains information about all employees of the company;
- the entity «PMs» contains information about all employees of the company in the position of PM; these employees are placed in a separate table, because of the company's management should be able to choose particular PM from the list to manage a new project;
- the entity «IT\_specialists» contains information about all employees in the positions of IT-specialists, except for project managers; these employees are placed in a separate table, because there are many employees who are not involved in the implementation of IT-projects;
- the entity «Candidates» contains information about employees and non-employees who took a part in a job interviews;

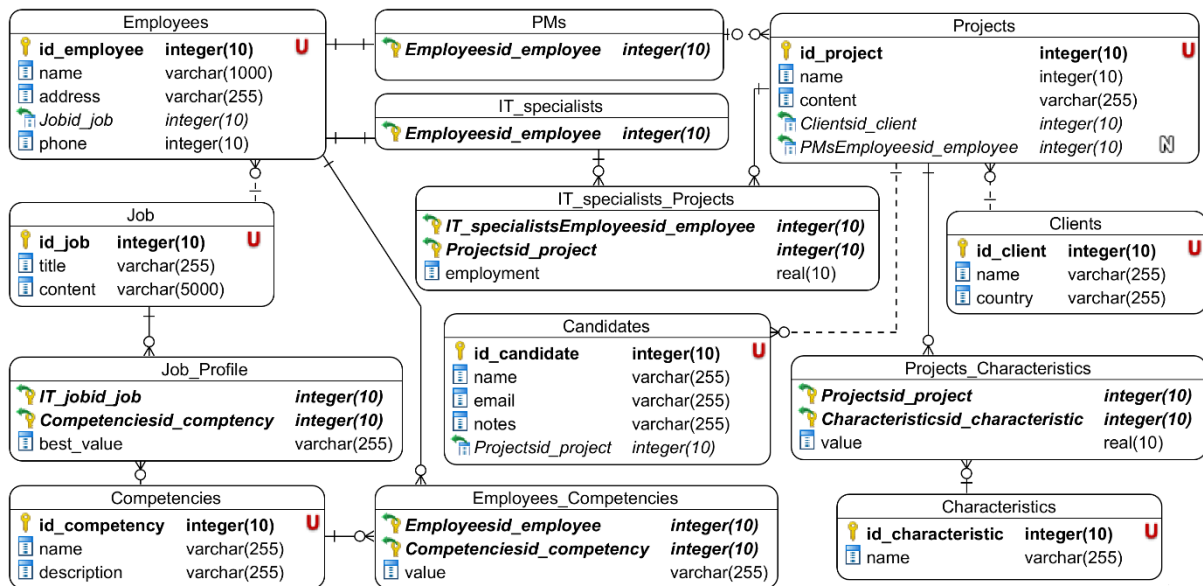


Figure 3: Model of database

- the entity «Job» contains information about all positions in the IT-company;
- the entity «Competencies» contains the competencies from job competency profiles;
- the entity «Projects» contains information about all projects of the IT-company;
- the entity «Clients» contains information about all clients of the IT-company including the country and the time zone;
- the entity «Characteristics» contains information about all characteristics of the projects;
- the associative entity «Job\_Profile» shows the competencies corresponding to a particular position; it contains a description of the ideal profile of each position;
- the associative entity «IT\_specialists\_Projects» shows IT-specialists who were involved or are currently working on the company's projects;
- the associative entity «Projects\_Characteristics» shows the characteristics corresponding to each project of the company;
- the associative entity «Employees\_Competencies» shows the levels of competence of the employees.

Thus, the RS architecture has been presented to solve the task of forming of software development teams.

#### 4.4. Results and Discussion

Let's consider the using of the developed RS for recruiting of software development teams. One foreign company has ordered a software from an IT-company. The names of the candidates, name of project and IT-company have been hidden. Management of the IT-company has assigned particular PM to this project. According to the proposed method of the team building, first step is done. Project Manager and HR-manager with Product Owner have chosen the following structure of the team from 10 IT-specialists: Project Manager (1 person), Development Leader (1 person), Quality Assurance Leader (1 person), Architect (1 person), Middle Front-end Developer (2 persons), Middle Back-end Developer (2 persons), Middle QA Engineer (1 person) and Junior QA Engineer (1 person). Next step is an appointment of specialists of the next management level: Development Leader, Quality Assurance Leader, and Architect. The input data is the information about employees of the IT-company and non-employees with suitable technical appraisal. The result of solving the task of selecting potential candidates according to [1] is following: five people claim for the vacant position of Development Leader in the project; four people claim for two vacant positions of Middle Front-end Developers, and three people for two vacant positions of Middle Back-end Developers. Each from group of five Development Leaders can be assigned on this role. So, final decision depends from



different variants of possible teams. The Table 1 presents the input data of potential candidates based on the L. M. Spencer model for superior performance.

**Table 1**  
The psychological picture of candidates to software development team

Employees	Focus on achieving results	Analytical thinking	Conceptual thinking	Customer service orientation	Development of others	Influence	Information search	Leader skills	Teamwork and cooperation	Self-confidence
DevLead 1	8	10	7	8	9	8	7	8	9	10
DevLead 2	9	9	6	9	8	7	6	9	8	7
DevLead 3	8	6	10	9	8	7	8	9	5	6
DevLead 4	6	8	8	6	7	8	9	10	6	7
DevLead 5	8	6	7	7	6	7	8	9	7	8
MidFront-end Dev 1	7	8	9	8	7	9	9	9	10	9
MidFront-end Dev 2	8	7	9	9	8	7	8	9	8	9
MidFront-end Dev 3	9	8	9	10	6	8	7	9	8	9
MidFront-end Dev 4	6	5	8	9	7	6	8	9	8	7
MidBack-end Dev 1	10	9	6	8	7	5	6	5	9	8
MidBack-end Dev 2	9	8	6	9	8	7	6	8	9	9
MidBack-end Dev 3	9	10	9	7	8	10	10	9	8	7

Next step is calculating the matching between candidates. The results of calculating the similarity measure for five employees from the Development Leaders' list (designated DevLead 1-5) are presented in the Table 2. It is the base for forming the development teams. The best candidates for the team are highlighted in bold.

**Table 2**  
An example of calculating the similarity measure for recruiting of software development teams

Employees from second and third lists	DevLead 1	DevLead 2	DevLead 3	DevLead 4	DevLead 5
MidFront-end Dev 1	<b>4,582575695</b>	6,082762530	6,855654600	<b>5,291502622</b>	5,099019514
MidFront-end Dev 2	<b>4,472135955</b>	<b>4,690415760</b>	<b>4,472135955</b>	<b>5,196152423</b>	<b>3,872983346</b>
MidFront-end Dev 3	5,000000000	<b>4,582575695</b>	5,567764363	6,324555320	4,690415760
MidFront-end Dev 4	7,141428429	5,916079783	<b>4,582575695</b>	<b>5,291502622</b>	<b>3,741657387</b>
MidBack-end Dev 1	5,744562647	<b>5,000000000</b>	8,660254038	8,831760866	6,633249581
MidBack-end Dev 2	<b>3,316624790</b>	<b>2,645751311</b>	<b>7,141428429</b>	<b>7,071067812</b>	<b>4,898979486</b>
MidBack-end Dev 3	<b>5,567764363</b>	6,244997998	<b>6,708203932</b>	<b>5,099019514</b>	<b>6,324555320</b>

The analysis of the Table 2 allows seeing three candidates from the second list instead of the two required one for DevLead 4. The reason is equal similarity measure of two candidates. Therefore, RS has formed two alternative teams for this employee. Thus, the result of using the RS is six alternative software development subteams.

The rest of the team is formed in a similar way: the similarity measures between the Quality Assurance Leader and his subordinates are calculated, and the Architect is selected. The RS forms a set of recommendations. It is a framework for the PM, who forms the final structure of the team for the project.

For checking the adequacy of working of RS with the opinion of experts, it is necessary to choose special metrics. The Recall and Precision metrics are inapplicable to assess the efficiency of the work of the developed RS, since the RS does not produce irrelevant responses. The metrics will always be equal to 100%. However, there is another useful metric. It is the *AP* metric – Average Precision. It takes into account the order of the recommendations of proposed software teams. It can be used for evaluating the efficiency of RS if the recommendation task is considered as a ranking task [16]. *AP* is calculated using the following formula:

$$AP@N = \frac{1}{m} \sum_{k=1}^N (P(k) \text{ if } k^{\text{th}} \text{ item was relevant}) = \frac{1}{m} \sum_{k=1}^N P(k) \cdot rel(k), \quad (8)$$

where  $rel(k)$  is an indicator that shows when  $k^{\text{th}}$  item was relevant ( $rel(k)=1$ ) or not ( $rel(k)=0$ );  $N$  is a number of recommendations made by RS;  $m$  is the number of relevant items from list of recommendations made by RS;  $P(k)$  is Precision at cutoff  $k$ , that is simply the precision calculated by considering only the subset of RS' recommendations from rank 1 through  $k$ .

Let's consider an example of calculating the *AP*-metric. The RS formed six alternative development subteams and arranged them in a certain order. An expert arranged formed six teams by RS in a certain order as well. The results of the allocation of teams are presented in the Table 3.

**Table 3**  
Results of the allocation of software development teams

Number in the list	Expert	RS	Correct or not
1	Team 5	Team 5	Correct
2	Team 1	Team 1	Correct
3	Team 4	Team 4	Correct
4	Team 2	Team 3	Incorrect
5	Team 3	Team 2	Incorrect
6	Team 6	Team 6	Correct

Thus, the metric *AP* according to (8) is equal to:

$$AP@6 = \frac{1}{6} \left( \frac{1}{1} + \frac{2}{2} + \frac{3}{3} + \frac{3}{4} + \frac{3}{5} + \frac{4}{6} \right) = 0,84.$$

The obtained value characterizes the adequacy of the RS operation, since the order of the formed teams reflects the priority of the command. If the importance of the commands is approximately equal, the expert can arrange in a different order than the recommendations from the RS. Thus, the obtained results indicate the results of the RS operation reflect the stated requirements.

In addition, it is possible to assess the quality of the RS work based on a set of characteristics, which shows the degree of achieving the goal.

To assess the quality of the forming process of the development teams, it is necessary to match the results of work of the HR-manager and PM before applying the RS and after. Process quality is a set of process indicators that reflects the degree of achievement of certain goals [15, 17]. To assess the quality of team building process, the following criteria or features can be used:

- $f_1$  – difficulty level of data identification;
- $f_2$  – difficulty level of processing data;
- $f_3$  – time for data processing;
- $f_4$  – level of objectivity in team building;
- $f_5$  – level of adaptation to changing conditions;
- $f_6$  – degree of consideration of psychological compatibility.

Each feature should be evaluated by 7-point scale, where 1 is the minimal value for metrics and 7 is the maximum accordingly. To obtain the numerical values of the criteria, the experts have evaluated the complexity of solving the given task manually and using RS according to the proposed

scale. In this study, a weighted average quality metric  $PQ$  is proposed in [15] as a convolution criterion, where  $f_u'$  is a normalized value of  $f_u$ -th metric and  $w_u$  is a weight of it:

$$PQ = \frac{\sum_{u=1}^6 f_u' w_u}{\sum_{u=1}^6 w_u} \quad (9)$$

In order to normalize the selected criteria, it is necessary to determine  $f_u^+$  – set of criteria that make a positive contribution to the quality assessment, and  $f_u^-$  – negative criteria. Normalization is carried out according to the formulas (10) and (11) for negative and positive features, respectively:

$$f_u' = 1 - \frac{f_u^-}{6} \quad (10)$$

$$f_u' = \frac{f_u^+}{6} \quad (11)$$

Analysis of the domain area shows that an increasing of the values of features  $f_1 - f_3$  negatively affects the quality of the process of forming the project team, and the criteria  $f_4 - f_6$  are positive. The result of the assessment of the quality of forming the software development teams by (9)-(11) is shown below (Table 4).

**Table 4**  
The quality assessment

Criterion	Expert	RS	Weight of criterion	Normalized values	
				Expert	RS
$f_1$	6	1	0,4	0,14	0,86
$f_2$	5	1	0,8	0,29	0,86
$f_3$	7	2	0,9	0,01	0,71
$f_4$	4	5	0,5	0,57	0,71
$f_5$	6	3	0,4	0,86	0,43
$f_6$	2	7	0,9	0,29	1,00
$PQ$				0,30	0,79

The results from the Table 4 allows to notice that the quality of the process of forming the project's team based on the use of RS has increased by 49%, which confirms the research hypothesis stated earlier. Thus, the decision-making process of recruiting of software development teams can be improved using recommendations that allow taking into account the psychological profile of potential candidates.

## 5. Conclusion

In the course of this study, the approach for automation of the performance of one of the main functional responsibilities of a PM was presented. It is the formation of software development teams according to recommendation from the RS. Based on a comprehensive analysis of the domain area for the implementation of the RS, the content-based filtering approach and Euclidean distance were chosen. This measure has been used for calculation the proximity of candidates according to the psychological portrait. The algorithm for solving the task of forming of software development teams and a functional model of this business process are presented. The architecture of the RS was proposed and the database model was provided.

The scientific novelty of the given research is to improve the process of recruiting of software development teams using the proposed method, which allows taking into account the psychological compatibility of team members. The similarity measures between candidates are calculated. It allows taking into account results of calculations for forming the development team. Since the Recall and

Precision metrics are inapplicable for assessing the effectiveness of the developed recommendation system, the Average Precision metric was calculated. It allows evaluating the ranking order of the items in the recommendation list. The assessment of the quality of the RS work according to the indicators proposed by the authors is calculated. Numerical studies have shown the possibility of using the proposed method in real conditions in IT-companies to improve the efficiency of management decisions.

## 6. Acknowledgements

This paper and the research behind it would not have been possible without the support of Vitalii Hlushko – Senior Solution Architect, GlobalLogic, Kharkiv, Ukraine. We thank Vitalii Hlushko for assistance with development of proposed method and for comments that greatly improved the paper.

## 7. References

- [1] K. V. Melnyk, N. V. Borysova, Rozrobka informatsiinoi systemy dlia vyrishennia zadachi avtomatyzatsii protsesu pidboru personalu, in: V. Vychuzhanin (Ed.), Aktualni problemy informatsiinykh system i tekhnolohii, NU "OMA", Odesa, 2020, pp. 129-142.
- [2] The Top 12 Best Applicant Tracking Systems (ATS) – February 2021, 2021. URL: <https://www.selectsoftwarereviews.com/buyer-guide/applicant-tracking-systems>.
- [3] A. Stepanov, 10 Sistem dlya Upravleniya Proektami: Moshhno, Nadezhno, Effektivno, 2021. URL: <https://blog.ganttpro.com/ru/sistemy-instrumenty-servisy-upravlenie-proektami/>.
- [4] The Best Project Management Software for 2021, 2021. URL: <https://www.pcmag.com/picks/the-best-project-management-software>.
- [5] R. Hogan, Personality and the Fate of Organizations, 1st. ed., Routledge, London, 2006.
- [6] B. Tuckman, The Forming, Storming, Norming & Performing Team Development Model, 2021. URL: <https://www.coachingcultureatwork.com/bruce-tuckman-team-development-model/>.
- [7] D. Keirse, Personology, Prometheus Nemesis Book Company, Del Mar, 2010.
- [8] L. M. Spencer, S. M. Spencer, Competence at Work: Models for Superior Performance, Wiley India Pvt. Limited, Delhi, 2008.
- [9] Ch. Belhekar, Recommendation Systems – A walk through, 2020. URL: <https://medium.com/@chaitanyarb619/recommendation-systems-a-walk-through-33587fecc195>.
- [10] F. O. Isinkaye, Y. O. Folajimi, B. A. Ojokoh, Recommendation systems: Principles, methods and evaluation, Egyptian Informatics Journal, 16.3 (2015) 261–273. doi:10.1016/j.eij.2015.06.005.
- [11] S. Prabhakaran, Mahalanobis Distance – Understanding the math with examples (python), 2019. URL: <https://www.machinelearningplus.com/statistics/mahalanobis-distance/>.
- [12] B. S. Everitt, S. Landau, M. Leese, D. Stahl, Cluster Analysis, 5th. ed., Wiley, Chichester, 2011.
- [13] K. Melnyk, N. Borysova, Integrated Technology for Personnel Assessment Based on the Competencies Model, in: Proceedings of the 9th International Conference "Information Control Systems & Technologies", ICTS-2020, Odessa, 2020, pp. 343-357. doi:10.13140/RG.2.2.26024.60169.
- [14] N. V. Borysova, K. V. Melnyk, I. V. Olifenko, Tekhnolohiia pidboru komandy it-fakhivtsiv dlia vykonannia proektu, in: Proceedings of the V International Scientific-Practical Conference "Information Technologies in Education, Science and Technology", ITEST-2020, ChSTU, Cherkasy, 2020, pp. 40–41.
- [15] K. V. Melnyk, N. V. Borysova, V. I. Melnyk, Information system development for teaching staff performance evaluation, in: S.V. Kotlik (Ed.), Informatsiini tekhnolohii ta avtomatyzatsiia, Astroprynt, Odesa, 2020, pp. 46-54.
- [16] S. Sawtelle, Mean Average Precision (MAP) For Recommender Systems, 2016. URL: <http://sdsawtelle.github.io/blog/output/mean-average-precision-MAP-for-recommender-systems.html>.
- [17] What is process quality?, 2021. URL: <http://www.taskmanagementguide.com/glossary/what-is-process-quality.php>.