**DOI: https://doi.org/10.15276/hait.09.2026.29**
**UDC 378.147:004.6:331.5**

# Multi-criteria selection of the specialist for task execution in the software development process

**Radim V. Chorba[1)]**
ORCID: https://orcid.org/0009-0005-9879-4375; radim.chorba@gmail.com. Scopus Author ID: 60241199800
**Oleksii B. Kungurtsev[1)]**
ORCID: https://orcid.org/0000-0002-3207-7315; akungurtsev19@gmail.com. Scopus Author ID: 57188743440
[1)] Odesa National Polytechnic University, 1, Shevchenko Ave. Odesa, 65044, Ukraine

## ABSTRACT

**The object** of this study is a process of selecting executors for tasks in software development among the available specialists. **The problem** being solved is the insufficient accuracy and explainability of existing approaches to selecting a specialist caused by the incompleteness of the characteristics of specialists, in particular soft skills. This can lead to delays in delivery, insufficient quality of delivery, and inefficient use of labor resources in the software development process. The article **proposes** an information technology for selecting an executor, which is based on a specialist model and a **method** for selecting an executor for a task. The specialist model takes into account the historical dynamics of possession of technical and social-communicative skills, the success of performing previous tasks, domain expertise, roles in projects, and the context of their execution. **The key feature** of the developed technology is the formalized representation of the specialist's profile in the form of a structured tuple with classifiers of competence and skill levels, which allows for a quantitative assessment of compliance with task requirements. The model includes seven main components: personal data, education, current role, role history, technical skills, domain knowledge, and soft skills. For each characteristic category, separate classifiers have been developed, which are converted into numerical values for calculations. Based on the created model, a method for evaluating the compliance of a labor resource to a task has been built, which takes into account the weight coefficients of requirements, normalizes the values of characteristics, and provides an explainable compliance metric. **The method** implements a four-step process: determining the characteristics necessary for task execution; representing specialists in terms of their compliance with task requirements; computing the compliance function with weight coefficients; searching for the best candidates among available specialists. The technology allows automating the process of selecting specialists, taking into account both the current state and the development of competencies over time. Experimental evaluation demonstrated a 41% reduction in decision-making time compared to the traditional approach. **The results** are explained by the use of a clear structure of characteristics, mathematical formalization of the compliance function, and validation under real operating conditions. The technology can be effectively used in organizations provided there are historical data on the activities of specialists that are guaranteed to be reliable, the dynamics of competence and skill development, and sufficient detail of requirements for the tasks being performed. The technology limitations are the dependence on the reliability of the input data and its inexpediency for use for tasks with an expected effort of less than one man-week.

**Keywords:** Specialist model; specialist-to-task compliance measure; competency proficiency classifiers; executor selection information technology; multi-criteria selection

***For citation:*** **Chorba R. V., Kungurtsev O. B. "Multi-criteria selection of the specialist for task execution in the software development process".** ***Herald of Advanced Information Technolog.*** **2026; Vol.9 No.3: 443–455. DOI: https://doi.org/10.15276/hait.09.2026.29**

## 1. INTRODUCTION

In the context of intensive information technology implementation across various domains an important aspect is the software development processes acceleration while maintaining product quality. Crucial criteria for completing a task on time and with appropriate quality are the correct executor's selection. Incorrect executor selection for software development tasks significantly harms productivity and delays the new technologies introduction into production [1]. Tasks related to software development were previously examined by the authors in [2], [3]. However, those studies did not pay enough attention to investigating specialists and their characteristics required for task execution.

To a large extent, the task execution results depend on the executors' technical qualifications [4]. Most modern software development technologies involve collaborative work, the productivity of which largely depends on team relationships. Therefore, it is necessary to identify and take into account the social and communicative (soft) skills when choosing a task executor [5].

Existing researches propose various approaches to analyzing executor efficiency; thus, it is important to investigate factors that enable better performance in software development projects. In this context, models that consider both professional and soft skills can significantly improve the efficiency of executor selection. The authors understand an effective process of executor selection as a process that allows to select executor for a software development task who best meet the needs for competencies and skills needed for the task completion, while the time spent searching for such executor is less than when using existing approaches in software development organizations.

Therefore, there is an executor determining problem for a software development task that considers both professional and soft skills.

The paper is structured as follows.

Section 2 provides a review of related approaches, Section 3 defines the objectives of the study, Section 4 introduces the proposed models and supporting technology, Section 5 presents the experimental results, and Section 6 offers concluding remarks and summarizes the outcomes.

## 2. RELATED WORKS

According to the findings of the study [1], an improper executor selection for a task negatively affects the outcome of its implementation. Since the executor selection process is based on data, which is critically important both for the process itself and for final task execution result.

Several approaches exist to modeling specialists within systems, such as cognitive models, behavioral models, and data-based profiling. In most cases, these models address narrow and specific aspects. For example, one proposed model represents the dynamics of specialists' behavior, organizational behavior, and factors influencing group dynamics [6]. Study [7] focuses on modeling the physical and ergonomic environment of specialists to improve their performance. There is a model that describes the dynamics of changing specialists' qualification levels [8]. Authors in study [9] proposed the specialists' modeks focusing on the sustainability and capacity development.

Study [10] proposes the production resources functioning model that emphasizes data integration. The model is designed for the selection and development of specialists' competencies. However, this study provides only a conceptual framework without algorithmic analysis of compatibility or skill history. Therefore, all implementation details for a particular organization must be determined through experimental and research methods.

The professional competence matrix use was proposed in [11]. The authors suggest employing the matrix for assigning tasks to specialists, taking into account execution time and resource availability. The algorithm considers the correspondence between task requirements and technical competencies. However, the authors do not consider the soft skills levels and do not include historical data. A potential solution to this limitation could be the extension of the model with social and communicative skills, which is described in [12]. The proposed approach still does not consider skill development and historical data on task execution results. Moreover, the system's output is not transparent, even though a mathematical approach is used; thus, the results are not explainable.

The task of modeling specialists' skills is considered in [13], skill levels are tracked relative to each individual competency. The study implements a mechanism for learning/forgetting competencies depending on trainings and practical application of skills. Although the work provides a simple and organic method for replenishing the knowledge base regarding specialists' skills, the method does not pay attention to the specialist's roles in the project and the success of task execution. Also, the work does not propose structuring of social-communicative skills and does not propose a classification of skill proficiency levels.

The effect of competency aging in analyzing the specialists' match to a task is discussed in [14]. The authors propose the model for selecting an executor in such a way that the specialist's qualification level does not degrade significantly over time. As a result, the method achieves a balance between the specialist's productivity during task execution and the ability to maintain competencies at the sufficient level for performing relevant tasks. However, the method does not consider soft skills.

The reviewed models concentrate on specific and limited sets of specialist characteristics, without considering other aspects of their professional activity such as soft skills, the importance of which is indicated in [5] for software development tasks.

With the development of artificial intelligence (AI) in recent years, numerous attempts have been made to use it in solving some aspects of the issues investigated in this work. In particular, in the work [15], numerous works on the use of artificial intelligence in personnel management and competency analysis are systematized; the work [16] investigates the possibility of using artificial intelligence in selecting competencies of specialists that are appropriate to develop. Study [17] proposes to apply a neural network to the specialist selection problem. This approach requires careful data preparation and system validation before deployment. Its disadvantage lies in the results' opacity, as neural network outputs are inherently non-explainable. Furthermore, the study does not provide skills classification or their contextual use.

It should be noted that all approaches using neural networks or artificial intelligence in the process of executor selection have significant ethical shortcomings, such as discrimination, algorithmic bias, and dehumanization according to [18]. The spectrum of risks associated with the unexplainability of artificial intelligence algorithms received additional consideration in [19]. The authors also reached the similar conclusions about the risks of using unexplainable approaches in [20]. These drawbacks arise from the lack of explainability in AI-based algorithms.

For all approaches that lack explainability, it should be emphasized that their practical applicability is limited, since their decisions cannot be adequately justified. Study [21] proposes a solution to the problem of the inexplainability of AI decisions at the conceptual and experimental levels, but the study does not currently offer practical application.

From the presented analysis, it follows that existing approaches to selecting a specialist do not take into account all important aspects of the specialist's functioning when performing software development tasks, and approaches based on artificial intelligence additionally have ethical shortcomings. As a consequence, existing methodologies do not provide sufficiently accurate formalized approaches for selecting a specialist, which in turn can lead to time losses in the process of selecting an executor and when performing tasks in software development.

## 3. RESEARCH OBJECTIVE AND AIMS

The research goal is to reduce the time required to determine an executor for a software development task.

For achieving this goal, the following objectives were defined:

– to refine the task for the software devenopment model to reflect the skill requirements;

– to create the specialist model that contains data necessary for performing specific tasks;

– to develop the method for selecting an executor for a task;

– to propose the technology for implementing the method within an softare development organization;

– to conduct the experimental validation of the proposed method and evaluate the time savings in the executor selection process.

## 4. PROPOSED METHODOLOGY

### Refined software development task model

To solve the specialist selection problem for the specific task, it is necessary to formulate the requirements for the executor of that task.

As a base, it is proposed to use the task model presented in [3], extended with information on the required professional specialties and roles for task execution:

$$Task = \langle GenInfo, SpInfo \rangle, \quad (1)$$

where *GenInfo* – represents the generic task details:

$$GenInfo = \langle Title, Det, P, Dur, DS, DE \rangle, \quad (2)$$

where *Title* is task title; *Det* is task details; *P* is task priority; *Dur* is number of man-days required for task completion; *DS* is expected execution start date; *DF* is expected completion date.

*SpInfo* represents the skills requirements for the task executors:

$$SpInfo = \langle sR, sHSkN, sVrN, sSN, sSSkN \rangle, \quad (3)$$

where *sR* is multiset of task executor roles; *sHSkN* is set of necessary technical qualifications of executors [22]; *sVrN* is multiple requirements for proficiency in subject areas [23], [24]; *sSN* is set of titles of possible educational specialties; *sSSkN* is set of necessary soft skills of executors [25], [26].

If the execution of the task *Task* is planned for one executor, then the multiset *sR* consists of one element, the executor must perform all the jobs *Det* and for this must meet all the requirements of the task.

In this case, the requirements for the executor will take the form:

$$Req1 = \langle sHSkN, sVrN, sSN, sSSkN \rangle. \quad (4)$$

If the task execution involves more than one executor, it is necessary to form separate requirements for each role within *sR*:

$$Req_j = \langle ssHSkN_j, ssVrN_j, ssSN_j, ssSSkN_j \rangle, \quad (5)$$

where $ssHSkN_j$ – subset of the requirements for proficiency of the subject areas of the role; $ssVrN_j$ – subset of the requirements for proficiency in the subject areas of the role; $ssSN_j$ – a subset of titles of possible educational specialties for the role; $ssSSkN_j$ – subset of necessary soft skills for the role.

**Specialist model**

To determine the necessary components of the specialist model, the authors discoveried the processes of selecting a specialist to execute a task in software development companies and conducted a survey of experts regarding the necessary characteristics of a specialist. Since the specified process is interconnected with the specialist's development process, competencies aging aging prevention, determining career growth, etc., when making a decision, it is necessary to take into account a wider range of criteria than the needs of a separate task.

It was revealed that classifiers are needed for the certain components that are not possible to measure numerically and difficult to compare (in particular, for skill levels). Classifiers were created for such model components that allow for granular and clear categorization of the relevant characteristics. The implementation of classifiers creates the possibility of an unambiguous and transparent assessment of the compliance of the qualification under consideration with the stated requirements. When implementing the methodology in an organization, the classifiers can be changed by increasing or decreasing the level of detail, clarifying the criteria, etc. in order for them to meet the needs of the organization to the maximum extent.

The classifier values are converted to numeric values for calculation in way that the classifier value A (the highest degree of proficiency) corresponds to the value 1, the last classifier value (the minimum degree of proficiency) corresponds to the value 0. The remaining values are evenly distributed on a digital scale in descending order from A. When collecting information for such model components, the values obtained from experts are averaged.

Based on the needs mentioned, a specialist model that incorporates all necessary characteristics was developed:

$$Sp = \langle PDt, sEd, CR, sPrR, sHSk, sVr, sSSk \rangle, \quad (6)$$

where *PDt* is information identifying the specialist; *sEd* is education data; *CR* is current role; *sPrR* is records on roles previously held by the specialist; *sHardSkill* is specialist's technical qualifications; *sVr* is records on domain (subject area) expertise; *sSSk* is records on soft skills.

Education data is defined as a tuple

$$Ed = \langle Inst, EndD, Spec, KA \rangle, \quad (7)$$

where *Inst* is higher education institution; *EndD* is graduation date; *Spec* is educational specialtiy; *KA* is academic performance data.

Data on previous roles is defined as

$$PrR = \langle Cmp, R, Resp, FrDate, ToD \rangle, \quad (8)$$

where *Cmp* – identifies the organization; *R* is role in the organization or project; *Resp* is the specialist's areas of responsibility description; *FrD* and *ToD* is employment period in the given role.

Professional (hard) skills are represented as a tuple

$$HSk = \langle Tech, D, RQ, IW, Pj, R, PS, FrD, ToD \rangle, \quad (9)$$

where *Tech* is programming language or technology; *D* is technology proficiency level; *RQ* is result quality assessment; *IW* is autonomy in work assessment; *Pj, R* is project and role in the project; *PS* is project task execution success metric; *FrD* and To*D* is participation period in the specified project and role.

Technology proficiency level *D* takes values from the classifier:

A – technology was fully applied in practice;

B – technology was fully applied in practice according to project requirements;

C – technology was partially applied in practice according to project requirements;

D – technology was studied theoretically and used in non-commercial practice;

E – theoretical understanding of the technology is present;

F – technology was not studied, perceived as simple;

G – technology was not studied, perceived as complex.

Result quality assessment *RQ* takes values from the classifier:

A – work completed on time without revisions;

B – work completed on time with minor revisions;

C – work completed after several iterations and/or minor delays;

D – work completed with major revisions or significant delays;

E – solution rejected due to quality or timing issues.

Autonomy in work assessment *IW* takes values from the classifier:

A – architecturally correct solution proposed and completed independently;

B – work completed independently based on proposed architectural solution, including pair programming (without supervisor involvement);


ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

C – completed with minor supervisor or peer assistance;

D – completed with periodic assistance;

E – completed with constant supervision.

Project execution success metric *PS* takes values from the classifier:

A – project completed successfully;

B – project completed with minor deficiencies or schedule deviations;

C – project completed with significant deficiencies or delays;

D – project rejected by the client.

A skill proficiency level *SkillLevel* is introduced, taking the values:

A – full proficiency,

B – predominant proficiency,

C – partial proficiency,

D – no proficiency.

In order to conduct more granular analysis of the specialist's performance development, granulation of data collection regarding the specialist's work results is possible. To achieve this, indicators are gathered and recorded based on the results of each development iteration, in this case project execution success metric *PS* will have the meaning of success of a separate iteration. This approach requires additional efforts for data collection and analysis, but the value it provides is that at early stages of project execution and/or by a newly formed team, the possibility is created for analyzing the trend of adaptation of the team as a whole and individual engineer to the project environment.

Domain expertise is represented as

$$Vr = \langle VerticalName, VerticalQualif \rangle, \qquad (10)$$

where *VerticalName* is domain area; *VerticalQualif* is domain proficiency level according to the classifier:

A – there are confirmed successful projects in the subject area;

B – there are partially successful or unsuccessful projects in the subject area;

C – theoretical understanding of the subject area exists and successful projects in related areas;

D – theoretical understanding of the subject area or successful projects in related areas exists;

E – theoretical understanding of the subject area is absent.

Soft skills are represented as

$$SoftSkill = \langle SSkN, L, Dt, Cn, SDt, Rv \rangle, \qquad (11)$$

where *SSkN = enum(CriticalThinking, Leadership, EmotionalIntellect, Teamwork, SelfOrganization, ProblemSolving, Creativity,ConflictSolving)* is skill from the list of critical thinking, leadership abilities, emotional intelligence, team work skills, self-organization, problem-solving, creativity, conflict management [27]. The list can be expanded to ensure maximum compliance with the needs of the organization implementing the methodology; *L* is skill proficiency level, *Dt* is observation date; *Cn* is skill application circumstances; *SDt* is additional notes; *Rv* is person who documented or assessed the skill manifestation.

Here skill proficiency level *L* takes values from the classifier:

A – possesses the skill and capable of developing the skill in colleagues;

B – possesses and uses the skill systematically;

C – sometimes demonstrates possession of the skill;

D – weak possession of the skill.

**Method for Selecting an Executor for a Task**

The proposed method is based on the specialist model and consists of four main steps.

*Step 1.* Determining the characteristics (requirements) necessary for task execution

According to (3), the full requirements for all specialists are presented in the form:

$$SpInfo = \langle sR, sSt, sHSkN, sSSkN, sVrN, sSN \rangle. \qquad (12)$$

Using the SpecMatch software [3], requirements are encoded according to (12) in an automated mode.

*Step 2.* Determining the requirements for an individual specialist

Since in the general case the task is planned for several executors, their number $q$ is determined from the multiset of roles $sR$: $q=|sR|$. The determination of the task executors will be completed after the executors are determined for all roles.

According to (5), for $q=1$, the full requirements for the first role and the corresponding specialist from the point of view of compliance to the task requirements are presented in the form:

$$Req_1 = \langle ssHSkN_1, ssVrN_1, ssSN_1, ssSSkN_1 \rangle. \qquad (13)$$

The responsible person who manages the task execution must present a list (set) of candidates for the role of task executors – *sDeveloper*. List of potential lask executors will be gradually refined by the correspondence of specialists from the list of roles in the task.

For the first specialist, according to requirements of $Req_1$, based on (6), the set of characteristics are determined:

$$sDevCap_1 = \langle sPrR_1, sHSk_1, sVr_1, Ed_1, sSSk_1 \rangle. \quad (14)$$

*Step 3.* Determining the specialist's compliance with the task requirements

Since the solution explainability is actual, it was decided to use approaches that can be explained relatively trivially. This criterion is met by the family of MCDM approaches, the development of which was analyzed in detail in [28]. In [29], the authors note the popularity of the TOPSIS method for solving similar problems. The authors of the study [30] emphasize the comparability of the results when using the SAW and TOPSIS methods. The simplicity of solution explainability and ease of setup are considered important by the authors, therefore, the authors believe that the development of the SAW method for solving the research problem is more appropriate for the problem being solved.

To assess a specialist's ability to perform a task, their characteristic values are compared to the task's requirements.

The solution can be written as

$$Solution = F\left(sReq, sDevCap\right), \quad (15)$$

where $F$ is a function that determines the compliance level between the specialist and the task.

The function and its arguments must satisfy the following conditions:

– the function result must be normalized, allowing correct results comparison under different factor values;

– the function must return a value within the range [0, 1], where 0 indicates total non-compliance and 1 indicates complete compliance;

– factor values must be normalized to allow heterogeneous factors to be used within a single evaluation; their values must lie in the interval (0, 1], where 0 corresponds to the minimum and 1 to the maximum;

– the function must consider the averaged factor values.

The function F can be represented as

$$F = \{ f_1, f_2, \dots f_i, \dots f_m \}, \quad (16)$$

where $f_i$ is a sub-function determining the correspondence between one task requirement and the specialist's characteristic, and $m = |sTaskReq|$ is the number of task characteristics.

Thus, the decision takes the form

$$Solution = \frac{\sum_{i=1}^{m} f\left(i\right)}{m} \; . \quad (17)$$

A more precise solution may be obtained using the principal component method, in this case, by assigning different weights to task factors. Increasing the weight of some factors must not violate normalization.

Let us denote the set of factors for which weights need to be assigned as *sTaskReqW*, where $sTaskReqW \subseteq sTaskReq$. Then

$$sTaskReqW = \left\{ tr_{i,1}, tr_{i,2}, \dots, tr_{q,p} \right\}, \quad (18)$$

where the first index identifies the element within *sTaskReq*, and the second within *sTaskReqW*. The maximum factors sum with assigned weights under the condition $Sw \le m$ is

$$Sw = \sum_{j=1}^{p} tr_j \times w_j , \quad (19)$$

where $tr_j \in sTaskReqW$ and $w_j$ is the $tr_j$ weight; $m$ is the total characteristics number used in the solution.

The normalization coefficient for weights *Kw* must be introduced for factors without predefined weights to maintain normalization. From the equation $Sw + \left(m - p\right) \times Kw = m$,

$$Kw = \frac{m - Sw}{m - p}. \quad (20)$$

To enable the weighting coefficients use, equation (13) is transformed into the form

$$Solution = \frac{\sum_{i=1}^{m} f\left(i\right) \times w0_i \times Kw}{m}, \quad (21)$$

where $w0_i = 1$ is the default weight value.

If necessary, blocking factor values can be introduced as

$$Solution = \begin{cases} 0, \text{ if } f_i \le LowestAcceptable_i \\ \dfrac{\sum_{i=1}^{m} f_i \times w0_i \times Kw}{m} \end{cases}, \quad (22)$$

where $LowestAcceptable_i$ is the threshold value for the factor $f_i$.

*Step 4.* Forming the specialists set for the task execution

The selection of the task executors is performed for all roles from *sR* of the task and within all candidates for the roles of executors from *sDev*. The result is a set *sDevSel* of *n* specialists whose characteristics most closely match the requirements for performing the task.


ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

We define the function *fChoice* for selecting specialists to perform a task as

$$sDevSel = fChoice\left(sDev, sTaskReq, n\right), \quad (23)$$

where *sDevSel* is a set of *n* specialists whose characteristics best match the requirements of the task; *sDev* is the set of *k* specialists from which the executors must be selected (where $n < k$).

The function returns a set of tuples containing the specialist identifier and the value of the *Solution* function (22) for that specialist.

This study does not address the issue of specialists' workload or involvement in other tasks. That aspect was discussed in [2].

**Specialist selection technology**

The proposed technology for specialist selection includes the following six stages.

Stage 1. Updating specialist data

To obtain accurate results, the potential executors' data must be up to date. This requires collecting and accumulating specialists' data in accordance with the proposed model. The technology enables working with incomplete specialist data. Data updates are performed continuously according to the organization's policy.

Stage 2. Task formalization

For formalization, the task is loaded into the updated SpecMatch software, where the task type is selected and an initial set of requirements with recommended proficiency levels is provided [3]. The formalization process is based on the proposed task model (1).

Stage 3. Criteria definition

Using the updated SpecMatch software, the following are set: weight coefficients, blocking factor values for task characteristics (requirements), possible educational specialties and organizational roles.

Stage 4. Preliminary selection of candidates

Specialists whose education or roles meet the specified criteria are pre-selected from the available specialists.

The condition for preliminary selection of $\text{Specialist}_i$ is:

$$\left(Spec_i \in sSN\right) \cup \left(CR_i \in sR\right). \quad (24)$$

The composition of the candidate group can be modified by adding or excluding specialists based on practical experience.

Stage 5. Determination of the specialist's final group

A group of specialists who are recognized as capable of performing the task according to the results of applying the proposed method is determined.

Using the updated SpecMatch [3] software, specialists are selected among those who were identified at the previous stage.

If *sDevSel=0* (no specialists match the criteria), a decision is made either to terminate the process (task rejection due to the development resource unavailability) or to return to Stage 3 to adjust weighting coefficients, educational specialties and roles.

Stage 6. Decision-making on the specialist assignment

Since organizations perform not only software development tasks but also specialist development, competency aging prevention, career growth, and motivation, the final decision is made based on $sPR_i$, $CR_i$, and $sEd_i$ values for each specialist identified at Stage 5 ($DevSel_i$), as well as on their current workload according to the method proposed in [2].

The technology is shown on Fig. 1. The responsible persons who manages the task execution are directly involved in the processes of formalizing tasks, establishing criteria, making a decision on the assignment, and appointing the executor.

## 5. THE PROPOSED METHOD APPROBATION

To implement the proposed technology, the SpecMatch software [3] functionality was extended. To provide additional flexibility, the REST API (Python Flask application) was developed to implement the proposed method logic. Within the SpecMatch software, additional functionality was introduced to set characteristic weights and select candidates through the created API.

A task is provided to SpecMatch, processed, and evaluated through the API, which returns compliance metrics used for decision-making regarding the task executor assignment. To enable integration with the organization's systems, the proposed method can also be used directly via the API without using SpecMatch software. The API accepts weight values for criteria and specialist data as input, and returns the Solution function result for each specialist (data format: JSON, request method: POST).

The API implementation includes three subsystems: Criteria, Candidates, and Matcher. The Criteria module receives the specialist requirements list and their priorities. Priorities are normalized. The Candidates module receives the specialist identifier and data on competency and skill proficiency levels. The Matcher module performs the computation of the Solution function (22) for each provided specialist based on the defined requirements.

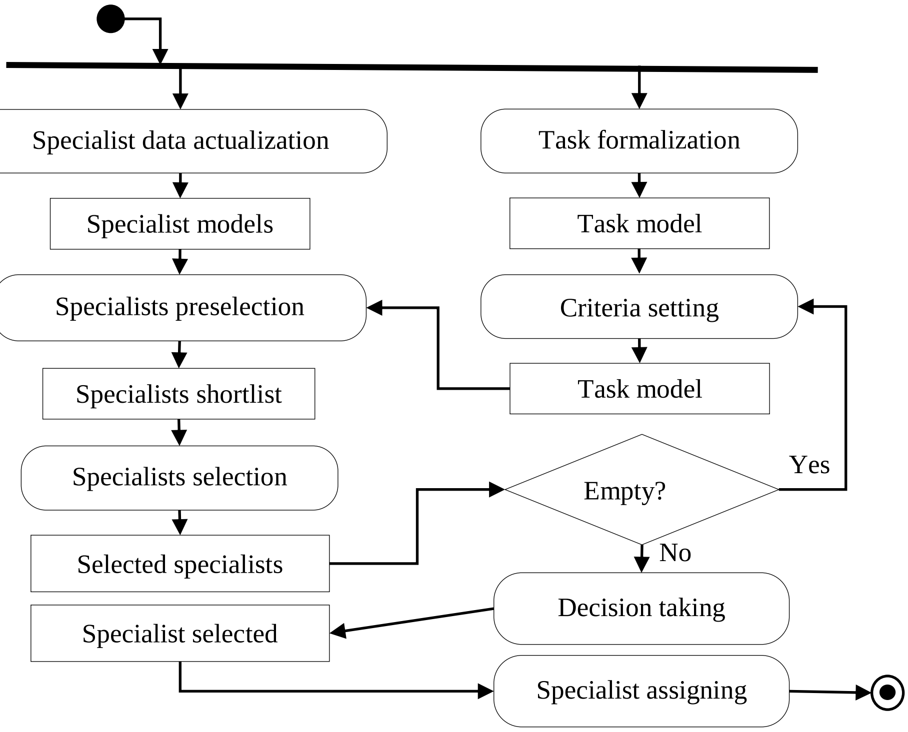


*Fig. 1.* **Specialist selection technology**
***Source*: compiled by the authors**

For the experimental evaluation, the authors of the study involved experts from 19 organizations specializing in software development. The organizations' term of operation on the market is at least 7 years, the delivery predictability indicator is from 71 % to 87 %, which makes it possible to consider the processes in the organizations as sufficiently stabilized to use the data in the experiment.

Specialists from organizations who make decisions on the selection of executors for tasks from the selected organizations were taken as experts. They were selected in way that their work experience in the relevant position is in range from 3 to 5 years; the share of *ResultQuality* level *D* for project delivery is not more than 20 %. This makes it possible to consider the experts as sufficiently qualified and the data of their assessments relevant.

With the help of experts from completed projects that correspond to levels A, B of the proposed classification of the Project execution success metric *PS*, 57 tasks of varying complexity with a completion period of 4 to 17 days were selected. An expert survey was conducted on the time spent on selecting an executor for these tasks. Time costs included the following stages: formalization of the task with required skills selection (T1), preselection of candidates (T2), candidates data refinement (T3), data analysis and decision-making on the best executors (T4).

The number of candidates was determined by the results of the selection of potential candidates and depended on the complexity of the task, the required technologies and established processes in the organization. Tasks with more than 10 candidates were excluded from the analysis.

The average time spent on the specified process are given in Table. 1 and are shown in Fig. 2, time consumption is provided in man-hours.

*Table 1.* **Average time spent on operations without using the proposed technology**

| Candi-dates | Tasks analyzed | T1 | T2 | T3 | T4 |
|---|---|---|---|---|---|
| 1 | 1 | 0.33 | 0.17 | 0.08 | 0.00 |
| 2 | 2 | 0.50 | 0.21 | 0.25 | 0.25 |
| 3 | 6 | 0.81 | 0.31 | 0.49 | 0.63 |
| 4 | 10 | 1.18 | 0.32 | 0.68 | 1.08 |
| 5 | 13 | 1.44 | 0.39 | 0.88 | 1.35 |
| 6 | 11 | 1.52 | 0.41 | 0.95 | 1.52 |
| 7 | 7 | 1.54 | 0.43 | 1.11 | 1.61 |
| 8 | 3 | 1.58 | 0.45 | 1.25 | 1.67 |
| 9 | 2 | 1.63 | 0.50 | 1.25 | 1.75 |
| 10 | 2 | 1.63 | 0.50 | 1.38 | 1.88 |

***Source*: compiled by the authors**


ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

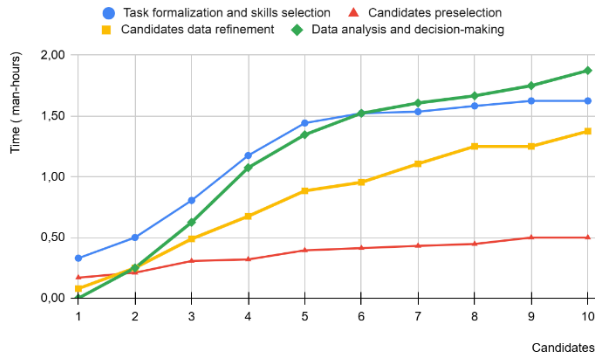


*Fig. 2.* **Task preparation and executor selection time without the proposed technology**
***Source*: compiled by the authors**

In order to test the proposed technology, the authors introduced the experts to the proposed method of selecting executors. The time spent on the new technology introduction was on average 0.95 hours.

The 57 new tasks were selected from current projects. These new tasks were similar to the previously analyzed tasks in terms of subject areas, nature of work and expected execution timeframe.

The experts selected the specialists to be the part of the experiment. In accordance with the proposed specialist model, the authors, with the help of experts, structured the input data on the selected specialists of the organizations. The average time spent on structuring the specialist information was 0.18 hours.

The experts, with the help of the authors, used the proposed method in the process of selecting executors for new tasks. Time costs included the following stages: formalization of the task with required skills selection (T1*), preselection of candidates (T2*), data analysis and decision-making on the best executors (T4*).

The average time spent on the specified process using the proposed technology is provided in Table 2 and shown in Fig. 3, time consumption is provided in man-hours.

The discrepancy in the number of candidates for tasks (see Table 1 and Table 2) is caused by the inability to determine the number of candidates at the stage of task selection and does not introduce a significant error into the experimental results.

The reduction in time for task formalization is a consequence of using the automated task definition process proposed by the authors in [3].

A slight reduction in time for candidates preselection can be explained by the formalized representation of the list of skills in the proposed method.

*Table 2.* **Average time spent on operations using the proposed technology**

| Candidates | Tasks analyzed | T1* | T2* | T4* |
|---|---|---|---|---|
| 1 | 1 | 0.33 | 0.17 | 0.33 |
| 2 | 3 | 0.50 | 0.22 | 0.50 |
| 3 | 6 | 0.71 | 0.28 | 0.55 |
| 4 | 8 | 0.93 | 0.31 | 0.70 |
| 5 | 12 | 1.10 | 0.36 | 0.76 |
| 6 | 12 | 1.23 | 0.37 | 0.80 |
| 7 | 7 | 1.25 | 0.42 | 0.84 |
| 8 | 4 | 1.31 | 0.44 | 0.90 |
| 9 | 2 | 1.38 | 0.46 | 0.92 |
| 10 | 2 | 1.38 | 0.46 | 0.92 |

***Source*: compiled by the authors**

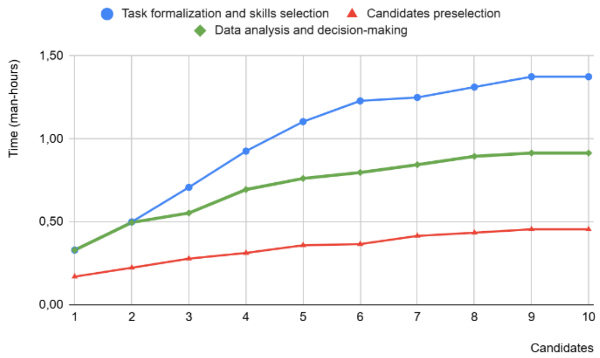


*Fig. 3.* **Task preparation and executor selection time using the proposed technology**
***Source*: compiled by the authors**

The data refinement operation (old technology, T3) is not required since the data on specialists are actual at the beginning of the process.

The reduction in time for the final selection of executors was achieved due to the automated execution of the function of searching for the best specialists (22).

Fig. 4 shows the comparison bettween the total time spent on the process starting from task preparation and ending at specialist's selection using old and new technologies.

The experiment demonstrated that the use of the proposed technology allowed to reduce the time spent on the process of selecting an executor by 1.7 times. The time savings amounted to an average of 1.5 hours. The payback of the time spent on the new technology introductuion and the initial structuring of specialists data occurs on the 3rd iteration of selecting an executor, if the number of specialists processed does not exceed 19 and provided that the accumulation of new specialist data occurs permanently, according to stage 1 of the proposed technology.

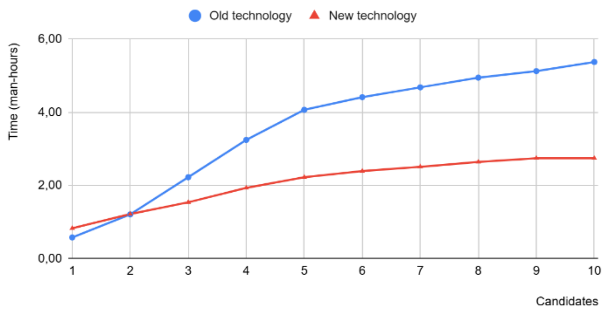


*Fig. 4*. **Average time spent on the process from preparing the task to selecting the executors**
***Source*: compiled by the authors**

Fig. 5. shows the result of a call to the API that implements the proposed method of selecting an executor.

**Executor Selection Method Approbation Results Discussion**

The reduction in time spent on selecting a specialist to perform a task (Fig. 4) can be explained by the use of the mathematical compliance function (22) and the structured system of specialist characteristics with classifiers.

The proposed method's advantages include the compliance function explainability, which enables quantitative assessment of how well the specialist matches the task requirements. The method is transparent enabling the avoidance of ethical risks.

The proposed model and method effectively address the problems identified in Section 2 by integrating technical qualifications, soft skills, historical data, and mathematical transparency, ensuring explainability.

The proposed method and technology possess the following key advantages compared to existing approaches. First, the method provides explainability through its mathematical transparency, addressing the opacity issues noted in [15]. Second, it incorporates soft skills assessment alongside technical competencies, extending beyond the scope of [13]. Third, the approach offers implementation simplicity through API automation, contrasting with the conceptual framework presented in [10].

The study limitation is the dependency on the reliability of input data regarding both candidates and the tasks to be performed. Another limitation is that the proposed technology is not recommended for tasks with an expected workload of less than 40 man-hours. Yet another study limitation is the psychological factors (e.g., burnout) analysis lack that may affect specialist's performance. The scope of this research did not allow detailed investigation of that aspect.

Future work may involve extending the task model for use in specialist recruitment and selection processes, as well as developing approaches to formalizing task descriptions, career management, and specialist development.

## 6. CONCLUSIONS

The task execution requirements for candidates' professional qualifications and soft skills were analyzed. The task model was refined according to the identified requirements.

The specialist model that includes a set of characteristics represented by structured data groups was developed. It allows specialists' capability assessment to perform specific tasks.

The method that provides explainable, normalized results for determining a specialist's readiness to execute a particular task was created. It takes into account the weighted priority of individual requirements.

The technology that ensures the determination of the most suitable executor from the candidate set was proposed. The technology considers the specialist's temporal development of competencies.

Research results experimental validation was conducted. The findings demonstrated a 2.7-fold reduction in time required to identify the best candidate for a given task.

In future research, the specialist model is planned to be applied for solving problems related to recruitment, career management, and specialists professional development.

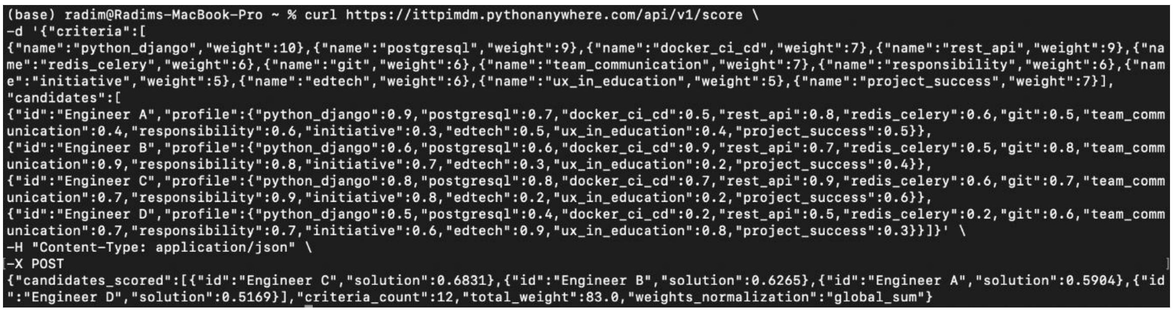


*Fig. 5*. **Executor selection with API**
***Source*: compiled by the authors**


ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

## REFERENCES

1. Huang, J., Balezentis, T. Shen, S. & Streimikiene, D. "Human capital mismatch and innovation performance in high-technology enterprises: An analysis based on the micro-level perspective". *J. Innov. Knowl.*, 2023; 8 (4): 100452, https://www.scopus.com/pages/publications/85178181696. DOI: https://doi.org/10.1016/j.jik.2023.100452.

2. Kungurtsev, O. & Chorba, R. "Task execution flow management in the software development process under the minor change event". *Herald of Advanced Information Technology*. 2023; 6 (4): 297–307. DOI: https://doi.org/10.15276/hait.06.2023.19.

3. Kungurtsev, O. & Chorba, R. "Method for automated executor selection based on a software development task model". *Automation of Technological and Business Processes.* 2025; 17 (3): 115–124. DOI: https://doi.org/10.15673/atbp.v17i3.3262.

4. Nurdin, M. "The effect of work experience and hard skill on employee performance of PT. Multi Kencana Niagatama". *At-Tadbir: J. Ilm. Manaj.* 2023; 7 (1): 41–53. DOI: https://dx.doi.org/10.31602/atd.v7i1.9221.

5. Mirza, F. & Chaudhry, N. I. "Linking employee traits to supervisor-related task performance in post-pandemic era: The mediating/moderation role of career adaptability and career identity". *IIMT J. Manag.* 2025; 2 (1): 7–25. DOI: https://doi.org/10.1108/IIMTJM-11-2023-0049.

6. Jayashankar, R. & Balan, M. "Unveiling the dynamics of employee behavior through Wolfram's cellular automata". *arXiv preprint.* 2024. DOI: https://doi.org/10.48550/arXiv.2407.09581.

7. Haapakangas, A., Sirola, P. & Ruohomäki, V. "Understanding user behaviour in activity-based offices". *Ergonomics*. 2022; 66 (4): 419–431, https://www.scopus.com/pages/publications/85135277867. DOI: https://doi.org/10.1080/00140139.2022.2092654.

8. Alefari, M., Almanei, M. & Salonitis, K. "A system dynamics model of employees' performance". *Sustainability*. 2020; 12 (16): 6511. DOI: https://doi.org/10.3390/su12166511.

9. Dotsenko, N., Chumachenko, I., Nekrasov, I., Akbulut, F. & Husieva, Y. "Conceptual model of human capacity formation and management in a multi-project environment to ensuring sustainable development of recovery programs". *Advanced Information Systems*, 2026; 10 (2): 20–27. DOI: https://doi.org/10.20998/2522-9052.2026.2.02.

10. Asfahani, A. M. "Fusing talent horizons: The transformative role of data integration in modern talent management". *Discover Sustainability*. 2024; 5 (25), https://www.scopus.com/pages/publications/85187105630. DOI: https://doi.org/10.1007/s43621-024-00212-7.

11. Staruch, B. & Staruch, B. "Competence-based assignment of tasks to workers in factories with demand-driven manufacturing". *Central European Journal of Operations Research*. 2021; 29: 553–565, https://www.scopus.com/pages/publications/85101074849. DOI: https://doi.org/10.1007/s10100-021-00739-4.

12. Li, H., Hui, Q. & Y. Wang. "Preemptive software project scheduling considering personality traits". *Systems*. 2024; 12 (10): 407, https://www.scopus.com/pages/publications/85207683293. DOI: https://doi.org/10.3390/systems12100407.

13. Heuser, P., Letmathe, P. & Schinner, M. "Workforce planning in production with flexible or budgeted employee training and volatile demand". *Journal of Business Economics*. 2022; 92 (7): 1093–1124. DOI: https://doi.org/10.1007/s11573-022-01090-z.

14. Szwarc, E., Bocewicz, G., Golińska-Dawson, P. & Banaszak, Z. "Proactive operations management: Staff allocation with competence maintenance constraints". *Sustainability*. 2023; 15 (3): 1949, https://www.scopus.com/pages/publications/85148028686. DOI: https://doi.org/10.3390/su15031949.

15. Kaushal, N., Kaurav, R.P.S., Sivathanu, B., et al. "Artificial intelligence and HRM: identifying future research agenda using systematic literature review and bibliometric analysis". *Management Review Quarterly.* 2023; 73: 455–493. DOI: https://doi.org/10.1007/s11301-021-00249-2.

16. Singh, A. & Chouhan, T. "Artificial intelligence in HRM: Role of emotional–social intelligence and future work skill", Tyagi, P., Chilamkurti, N., Grima, S., Sood, K. and Balusamy, B. (Ed.) *The adoption and effect of artificial intelligence on human resources management, part A (Emerald studies in finance, insurance, and risk management), Emerald Publishing Limited, Leeds*. 2023. p. 175–196. DOI: https://doi.org/10.1108/978-1-80382-027-920231009.

17. Szwarc, E., Golińska-Dawson, P., Bocewicz, G. & Banaszak, Z. "Robust scheduling of multi-skilled workforce allocation: Job rotation approach". *Electronics*. 2024; 13 (2): 392, https://www.scopus.com/pages/publications/85183340335. DOI: https://doi.org/10.3390/electronics13020392.

18. Rodgers, W., Murray, J. M., Stefanidis, A., Degbey W. Y. & Tarba S. Y. "An artificial intelligence algorithmic approach to ethical decision-making in human resource management processes". *Human Resources Management Review*. 2023; 33 (1): 100925, https://www.scopus.com/pages/publications/85133391241. DOI: https://doi.org/10.1016/j.hrmr.2022.100925.

19. Andrieux, P., Johnson, R. D., Sarabadani, J. & Van Slyke, C. "Ethical considerations of generative AI-enabled human resource management". *Organizational Dynamics*. 2024; 53 (1): 101032. DOI: https://doi.org/10.1016/j.orgdyn.2024.101032.

20. Charlwood, A. & Guenole, N. "Can HR adapt to the paradoxes of artificial intelligence?". *Human Resources Management*. 2022; 32 (4): 729–742, https://www.scopus.com/pages/publications/85122494941. DOI: https://doi.org/10.1111/1748-8583.12433.

21. Molnar, C. "Interpretable machine learning: A guide for making black box models explainable". 2025. – Available from: https://christophm.github.io/interpretable-ml-book. – [Accessed: 10 February 2026].

22. Greifenstein, L., Heuer, U. & Fraser, G. "Exploring programming task creation of primary school teachers in training". *In Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1* (ITiCSE). Turku, Finland. DOI: https://doi.org/10.1145/3587102.3588809.

23. Özkan, O., Babur, Ö. & van den Brand, M. "Domain – driven design in software development: A systematic literature review on implementation, challenges, and effectiveness". *Journal of Systems and Software*. 2025. p. 112537, https://www.scopus.com/pages/publications/105009735670. DOI: https://doi.org/10.48550/arXiv.2310.01905.

24. Ghatak, S. K. & Mahanty, B. "Impact of knowledge growth and team composition on the co-located software project performance". *Knowledge Management Research & Practice*. 2023; 21 (2): 397–411, https://www.scopus.com/pages/publications/85108252740.
DOI: https://doi.org/10.1080/14778238.2021.1939173.

25. "PMBOK® Guide". 2021. – Available from: https://www.pmi.org/pmbok-guide-standards/foundational/pmbok. – [Accessed: 25 July 2025].

26. Fagerholm, F., Felderer, M., Fucci, D., Unterkalmsteiner, M., Marculescu, B., Martini, M., Wallgren Tengberg, L.G., Feldt, R., Lehtelä, B., Nagyvaradi, B. & Khattak, J. "Cognition in software engineering: A taxonomy and survey of a half-century of research". *ACM Computing Surveys*. 2022; 54 (11): 226. DOI: https://doi.org/10.1145/3508359.

27. Jooss, S., Collings, D. G., McMackin, J. & Dickmann, M. "A skills matching perspective on talent management: Developing strategic agility". *Human Resource Management*. 2024; 63 (1): 141–157, https://www.scopus.com/pages/publications/85168679772. DOI: https://doi.org/10.1002/hrm.22192.

28. Kumar, R., & Pamucar, D. "A Comprehensive and systematic review of multi-criteria decision-making (MCDM) methods to solve decision-making problems: Two decades from 2004 to 2024". *Spectrum of Decision Making and Applications*. 2025; 2 (1): 177–196. DOI: https://doi.org/10.31181/sdmap21202524.

29. Gottwald, D., Chocholáč, J., Kayacı Çodur, M., Čubranić-Dobrodolac, M. & Yazir, K. "Z-numbers-based MCDM approach for personnel selection at institutions of higher education for transportation". *Mathematics*. 2024; 12: 523. DOI: https://doi.org/10.3390/math12040523.

30. Ciardiello, F. & Genovese, A. "A comparison between TOPSIS and SAW methods". *Ann Oper Res*. 2023: 325: 967–994. DOI: https://doi.org/10.1007/s10479-023-05339-w.

**Conflicts of Interest:** The authors declare that they have no conflict of interest regarding this study, including financial, personal, authorship or other, which could influence the research and its results presented in this article

Received 06.03.2026
Received after revision 22.05.2026
Accepted 09.06.2026

**DOI: https://doi.org/10.15276/hait.09.2026.29**
**УДК 378.147:004.6:331.5**

# Багатокрітеріальний вибір спеціаліста для виконання завдань в процесі створення програмного забезпечення

**Чорба Радім Валерійович**[1)]
ORCID: https://orcid.org/0009-0005-9879-4375; radim.chorba@gmail.com. Scopus Author ID: 60241199800
**Кунгурцев Олексій Борисович**[1)]
ORCID: https://orcid.org/0000-0002-3207-7315; akungurtsev19@gmail.com. Scopus Author ID: 57188743440
[1)] Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

## АНОТАЦІЯ

**Об'єктом** даного дослідження є процес відбору виконавців для завдань у розробці програмного забезпечення. Розв'язуваною **проблемою** є недостатня точність та пояснюваність існуючих підходів до вибору спеціаліста, що може призводити до затримок у постачанні, недостатньої якості виконання та неефективного використання трудових ресурсів. У статті **запропоновано** інформаційну технологію відбору виконавця, яка базується на моделі спеціаліста та методі вибору виконавця для завдання. Модель спеціаліста враховує історичну динаміку володіння технічними та соціально-комунікативними навичками, успішність виконання попередніх завдань, експертизу в предметній області, ролі в проєктах та контекст їх виконання. Ключовою особливістю розробленої технології є формалізоване представлення профілю спеціаліста у вигляді структурованого кортежу з класифікаторами рівнів компетентності та навичок, що дозволяє здійснювати кількісну оцінку відповідності вимогам завдання. Модель включає сім основних компонентів: персональні дані, освіту, поточну роль, історію ролей, технічні навички, знання предметної області та соціально-комунікативні навички. Для кожної категорії характеристик розроблено окремі класифікатори, які перетворюються у числові значення для обчислень. На основі створеної моделі побудовано **метод** оцінювання відповідності трудового ресурсу завданню, який враховує вагові коефіцієнти вимог, нормалізує значення характеристик та забезпечує пояснювану метрику відповідності. **Метод** реалізує чотириетапний процес: визначення характеристик, необхідних для виконання завдання; представлення спеціалістів у термінах їх відповідності вимогам завдання; обчислення функції відповідності з урахуванням вагових коефіцієнтів; пошук найкращих кандидатів серед доступних спеціалістів. Завдяки цим відмінностям технологія дозволяє автоматизувати процес відбору спеціалістів з урахуванням як поточного стану компетенцій, так і розвитку компетенцій у часі. Для апробації методу було розроблено вебсервіс на базі Python Flask, який включає три підсистеми: Criteria (обробка вимог та пріоритетів), Candidates (обробка даних спеціалістів) та Matcher (обчислення функції відповідності). Експериментальна валідація проводилася за участю експертів з організацій-розробників програмного забезпечення. За результатами експерименту досягнуто зменшення часу відбору виконавця на 41% порівняно з традиційним підходом. **Результати** пояснюються використанням чіткої структури характеристик, математичної формалізації функції відповідності та тестуванням у реальних умовах. Технологія може ефективно використовуватися в організаціях за умови наявності історичних даних про діяльність спеціалістів, що гарантовано є достовірними, динаміки розвитку компетенцій і навичок та достатньої деталізації вимог до виконуваних завдань. Обмеженням технології є залежність від достовірності вхідних даних та недоцільність застосування для завдань з очікуваним часом виконаня менше ніж людино-тиждень.

**Ключові слова:** модель спеціаліста; метрика відповідності спеціаліста завданню; класифікатори компетентності; інформаційна технологія вибору виконавця; багатокритеріальний вибір

## ABOUT THE AUTHORS

**Radim V. Chorba** - PhD student, Department of Software Engineering. Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine
ORCID: https://orcid.org/0009-0005-9879-4375; radim.chorba@gmail.com. Scopus Author ID: 60241199800
***Research field**:* Methods and means of increasing the productivity of the development teams

**Чорба Радім Валерійович -** аспірант кафедри Інженерії програмного забезпечення. Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

**Oleksii B. Кунгурцев** - PhD, Professor, Department of Software Engineering. Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine
ORCID: https://orcid.org/0000-0002-3207-7315; akungurtsev19@gmail.com. Scopus Author Id: 57188743440
***Research field*:** Methods and means of increasing the productivity of information systems; communication means with automated systems in natural language

**Кунгурцев Олексій Борисович -** кандидат технічних наук, професор кафедри Інженерії програмного забезпечення. Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна