УДК 004.41+005.2

МЕТОД АВТОМАТИЗОВАНОГО ВИБОРУ ВИКОНАВЦЯ НА ОСНОВІ МОДЕЛІ ЗАВДАННЯ З РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

METHOD FOR AUTOMATED PERFORMER SELECTION BASED ON A SOFTWARE DEVELOPMENT TASK MODEL

[1]Kungurtsev O.B, [2]Chorba R.V.
[1]Кунгурцев О.Б., [2]Чорба Р.В.
[1,2] Odessa Polytechnic National University, Odessa, Ukraine
ORCID: [1]http://orcid.org/0000-0002-3207-7315, [2]https://orcid.org/0009-0005-9879-4375;
Email: [1]akungurtsev19@gmail.com, [2]radim.chorba@gmail.com

DOI: 10.15673/atbp.v17i3.3262

Анотація. *Робота спрямована на вирішення проблеми вибору фахівця, що найбільш підходить для виконання конкретного завдання щодо створення програмного забезпечення (ПЗ). Проаналізовано існуючі способи подання завдань в області розробки програмних продуктів. Встановлено, що не існує класифікація та формального подання завдань, які дозволяють сформулювати чіткі вимоги до вибору його виконавця. На основі аналізу технологій та практики розробки програмного забезпечення запропоновано дворівневу класифікацію завдань, яка враховує не тільки професійні, але й комунікаційні вимоги до виконавця. Для різних типів завдань відповідно до класифікації побудовано загальну модель завдання і моделі для підкласів з конкретними вимогами до компетенцій виконавця завдання і надані рекомендації щодо рівня його навичок. Розроблено метод вибору найкращого із наявних спеціалістів для виконання конкретного завдання, який враховує відповідність завданню його професійних компетенцій, комунікаційних навичок, а також ступень зайнятості іншими роботами. Для апробації запропонованого методу було створено ПЗ SpecMatch і перевірено ефективність методу шляхом його застосування при виборі виконавців. Експерименти показали, що застосування запропонованого методу дозволило скоротити витрати часу на прийняття рішення щодо вибору виконавця на 31%.*

Abstract. *The work is aimed at solving the problem of selecting the most suitable specialist for a specific software (SW) creation task. The existing methods for presenting tasks in the field of software product development are analyzed. It is established that there is no existing classification and formal representation of tasks that allows for formulating clear requirements for selecting their performer (or executor). Based on the analysis of software development technologies and practices, a two-level task classification is proposed, which takes into account not only professional but also communication requirements for the performer.*

For different task types, in accordance with the classification, a general task model and models for subclasses with specific requirements for the performer's competencies are constructed, and recommendations regarding their skill level are provided. A method for selecting the best available specialist for a specific task has been developed, which considers the match between the task and their professional competencies, communication skills, and degree of commitment to other work (or current workload).

To test the proposed method, the SpecMatch software was created, and the method's effectiveness was verified by applying it in performer selection. Experiments showed that the application of the proposed method allowed for a 31% reduction in the time spent on making the decision regarding performer selection.

Keywords: task model, specialist selection information technology, decision support, multicriteria selection (or multi-criteria selection)

Ключові слова: модель завдання, інформаційна технологія вибору спеціаліста, підтримка прийняття рішень, багатокрітеріальний вибір

1. ВСТУП

Створення програмного забезпечення (ПЗ) відноситься до кваліфікованої праці. Розвиток комп'ютерних технологій спричинив значне підвищення складності систем і до появи великої кількості спеціалізацій в індустрії створення ПЗ. Успішність виконання кожного окремого завдання в створенні ПЗ залежить, в тому числі, і від якості вибору виконавця [1]. У практиці розробки програмного забезпечення існує набір підходів до оцінювання компетенцій фахівців [2]. Ці підходи часто базуються на емпіричних даних, таких як резюме, інтерв'ю чи ретроспективний аналіз виконаних завдань, і застосовуються в різних організаційних контекстах - від великих корпорацій до стартапів. Відсутні підходи, що дозволили б отримати однозначну оцінку відповідності компетенцій спеціаліста до конкретного завдання. Процеси підбору та розподілу ресурсів для виконання завдань залишаються значною мірою суб'єктивними і позбавленими чіткої структури. Це призводить до неефективного використання людських ресурсів як під час підбору спеціаліста, так і під час виконання завдань, а також до потенційних затримок у реалізації проєктів [3].

Наявна проблема полягає в відсутності класифікації завдань за типами, яка б враховувала специфіку вимог до технічних і комунікаційних навичок виконавців, а також формального представлення моделей компетенцій, що дозволило б однозначно визначати співвідношення необхідних навичок для конкретної задачі. Крім того, бракує структурованого методу підбору спеціалістів, який би інтегрував ці елементи для об'єктивного прийняття рішень, що посилює суб'єктивність і збільшує витрати часу на розподіл ресурсів.

Таким чином, існує проблема представлення завдання на розробку ПЗ у вигляді, який дозволяє обрати для його виконання спеціаліста, кваліфікації і навички якого в найбільшій мірі відповідають вимогам завдання.

2. КРИТИЧНИЙ ОГЛЯД НАЯВНИХ РІШЕНЬ

Дослідження [4] надає гранулярну деталізацію навичок та компетенцій розробників ПЗ, зазначаючи важливість як технічних компетенцій так і комунікаційних навичок для успішної роботі в проєктах. Проте, автори не пропонують підходів для визначення відповідності навичок до конкретних завдань, що виконуються.

Автори дослідження [5] досліджували компетенції і навички розробників ПЗ в динаміці і створили фундацію для класифікації типів завдань. В роботі досліджується вплив навичок на розвиток кар'єри, запропоновані можливі сценарії розвитку. Проте, дослідження фокусується на академічному контексті і не заглиблюється в питання відповідності навичок спеціаліста вимогам конкретного завдання.

Модель розвитку компетенцій запропонована в роботі [6]. Автори дослідили динаміку навичок і компетенцій на різних шаблях кар'єрного шляху в галузі розробки ПЗ для електронної комерції. Автори пропонують оригінальний підхід до структурування кваліфікацій і навичок, проте не заглиблюються в різницю вимог завдань що виконуються і не надають відповідних рекомендацій. Це може бути пов'язано з вузьким фокусом на дослідження в галузі електронної комерції. Таким чином, результати роботи не можуть бути застосовані в процесі визначення відповідності навичок виконавця завданню.

В роботі [7] автори підійшли до питання збору інформації щодо можливості залучення спеціаліста до роботи над завданням шляхом ретроспективного аналізу історичних даних в середовищах розробки ПЗ з відкритим кодом. Запропонований підхід включає аналіз завдань за допомогою обробки натуральної мови (NLP) і аналізу результатів виконання завдань. Проте, в дослідженні не приділяється уваги класифікації завдань за типом. Також питання виявлення комунікаційних навичок при цьому підході лишається відкритим, що робить сумнівним практичне застосування такого підходу.

В роботі [8] використано технологію блокчейн для побудови децентралізованої системи призначення задач в розподіленому середовищу розробки ПЗ. Автори підійшли до класифікації завдань з точки зору складності з використанням кластеризації. Підхід дозволяє досягти високого ступеню автоматизованості, проте він не має характеристики пояснюваності і не враховує комунікаційні навички спеціалістів, що ставить під сумнів доцільність його запровадження в організацях.

В роботі [9] досліджується вплив індивідуальних кваліфікацій і комунікативних навичок на успішність роботи самоорганізованих команд. Автори дослідили 84 команди і запропонували модель впливу індивідуальних характеристик на групову результативність команди. При тому що підходи, запропоновані в дослідженні, вкрай цінні для процесу формування команд, дослідження не надає підходів для визначення виконавця для конкретного завдання.

Дослідники запропонували структуру для комунікаційних навичок в [10] з великим ступенем деталізації, підкреслюючи їх важливість в процесі створення ПЗ. Детальне структурування навичок створює цінність цієї роботи, проте необхідно зазначити що аналіз навичок з таким ступенем деталізації вимагає високих трудозатрат. Дослідники не надають інформації щодо вимог володіння навичками для виконання завдань, що обмежує цінність роботи в практичному застосуванні керівниками команд з створення ПЗ.

З поданого аналізу випливає, що є актуальною задача формалізації вимог до виконавця завдання з створення ПЗ, що враховували б як кваліфікації так і комунікаційні навички, і методу підбору виконавця що відповідає формалізованим вимогам до компетенцій для виконання завдання з створення ПЗ, які б забезпечили більш повний аналіз критеріїв відповідності при виборі виконавця для завдання при зменшені часу на прийняття рішення за рахунок формалізації вимог і використання структурованого підходу до прийняття рішення.

3. МЕТА ТА ЗАДАЧІ ДОСЛІДЖЕННЯ

Метою даного дослідження є зменшення витрат часу на процесу вибору найкращого із наявних виконавців для виконання завдання в створенні ПЗ при забезпеченні більшої точності у визначенні вимог завдання, що сприятиме покращенню процесів створення і підвищенню якості програмних продуктів за рахунок збільшення відповідності виконавців вимогам проектних завдань.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. Розробити класифікацію завдань
2. Створити модель завдань з вимогами до компетенцій виконавця завдання
3. Розробити метод визначення спеціаліста для завдання, що базується на моделях завдання.

4. КЛАСИФІКАЦІЯ ЗАВДАНЬ

Для виконання завдання з створення ПЗ необхідне чітке розуміння набору затребуваних технічних кваліфікацій спеціаліста. Більш того, в сучасному проєктному середовищі при виконанні завдань виникають потреби в комунікаціях з стейкхолдерами проекту, спеціалістами що виконують суміжні завдання та ін. Таким чином, завдання накладає на виконавця вимоги як технічного [11], так і нетехнічного характеру [12].

Якщо вимоги до технічних кваліфікацій формуються на підставі документації на розробку завдання, то вимоги до рівня володіння комунікаційними навичками грунтуються на організації процесу створення ПЗ.

На підставі аналізу літератури, експертного опитування і власного практичного досвіду запропоновано класифікацію завдань, яка з одного боку перекриває основні процеси створення ПЗ, а з іншого – дозволяє сформулювати специфічні вимоги до виконавця. Перший рівень класифікації завдань відображено на рис. 1.

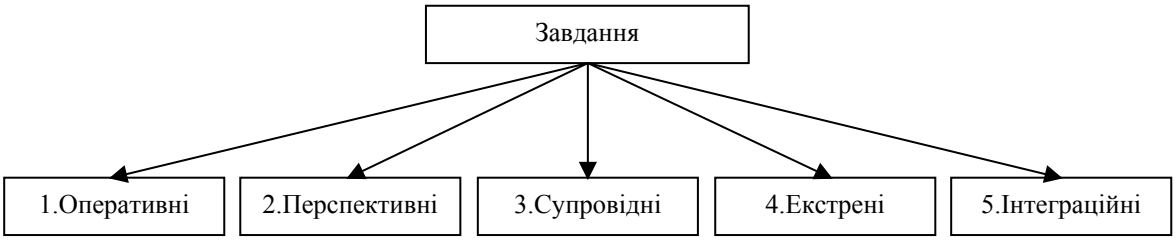

Рис. 1. Класифікація завдань, перший рівень
Fig. 1. Task Classification, First Level

Перший тип - оперативні завдання. Особливістю цього типу завдань є чітке визначення цілей та вимог. Завдання вимагають від спеціаліста професійні якості: впевненого володіння мовами програмування, фреймворками та інструментами, уміння застосовувати стандарти кодування і засоби забезпечення якості, розуміння предметної області. Завдання вимагають від спеціаліста наявності гнучких навичок: самоорганізації, дисципліни та здатності ефективно комунікувати з командою для уточнення вимог чи погодження змін [13]. Поділяються на 4 підтипи (рис. 2).

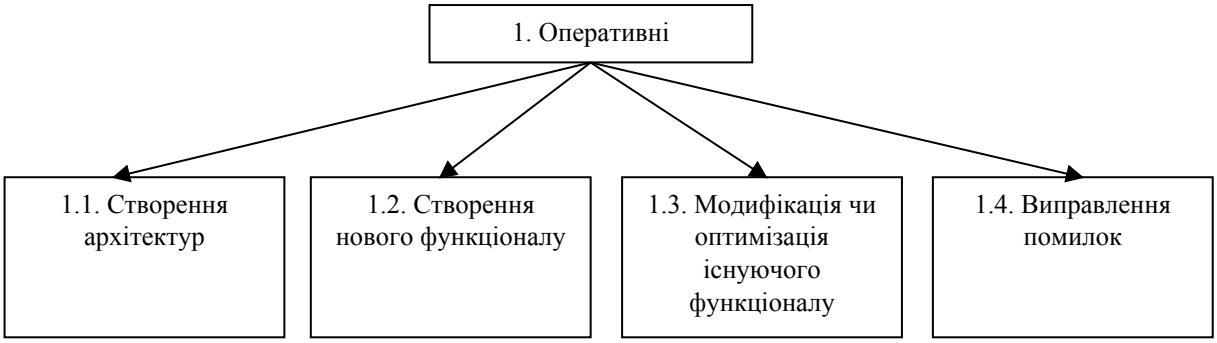

Рис. 2. Класифікація завдань типу *Оперативні*
Fig. 2. Classification of Operational Type Tasks

Для завдань типу *1.1 Створення архітектур* ключовими кваліфікаціями є розуміння предметної області, володіння широким спектром технологій, навичка вирішення проблем. Необхідно звертати увагу на рівень володіння навичками критичного мислення і самоорганізації.

Для завдань типу *1.2 Створення нового функціоналу* ключовим є володіння технологіями, потрібними для завдання, важливими - розуміння предметної області, навички критичного мислення, роботи в команді, самоорганізації та ін.

Для завдань типу *1.3 Модифікація чи оптимізація існуючого функціоналу* ключовими є навички критичного мислення, самоорганізації та вирішення проблем, важливим - володіння технологіями, потрібними для завдання.

Для завдань типу *1.4 Виправлення помилок* ключовою є навичка вирішення проблем, важливими - володіння технологіями, потрібними для завдання, критичне мислення, самоорганізація.

Перспективні завдання мають високий рівень невизначеності. Вимагають від спеціалістів здатності швидко опановувати нові інструменти, критично оцінювати їх сильні та слабкі сторони. Технічні компетенції тісно пов'язані з аналітичними здібностями й умінням працювати з науковими та технічними джерелами. Гнучкі навички включають креативність та критичне мислення. Поділяються на 5 підтипів (рис. 3).

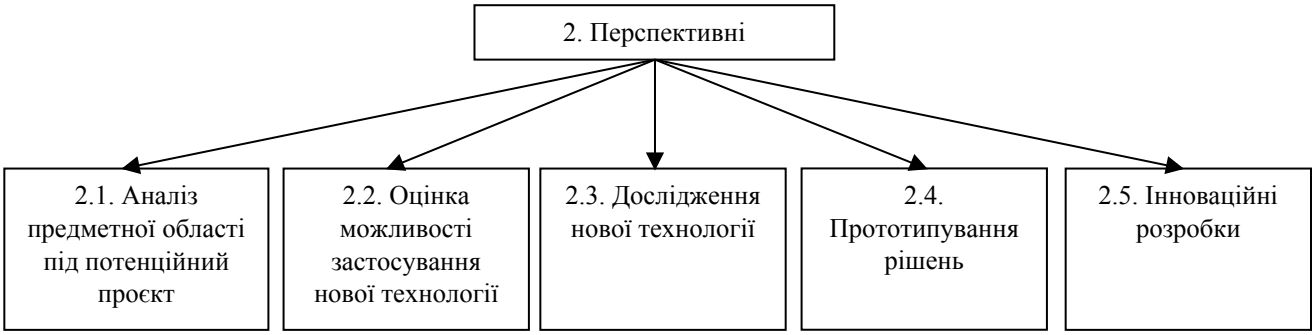

Рис. 3. Класифікація завдань типу *Перспективні*
Fig. 3. Classification of Perspective Type Tasks

Для завдань типу *2.1 Аналіз предметної області під потенційний проєкт* ключовими кваліфікаціями є розуміння предметної області, володіння широким спектром технологій, навички критичного мислення і креативність.

Для завдань типу *2.2 Оцінка можливості застосування нової технології* ключовою навичкою є критичне мислення, важливим є володіння технологіями для завдання.

Для завдань типу *2.3 Дослідження нової технології* ключовою кваліфікацією є володіння широким спектром технологій.

Для завдань типу *2.4 Прототипування рішень* ключовою навичкою є креативність, важливими – критичне мислення, емоційний інтелект і робота в команді.

Для завдань типу *2.5 Інноваційні розробки*, тобто створення принципово нових підходів чи продуктів, ключовою кваліфікацією є володіння широким спектром технологій, а навичкою – креативність, важливими – володіння технологіями для завдання і навички критичного мишлення та лідерства.

Супровідні завдання характеризуються роботою з великими обсягами спадкового коду, де можлива відсутність повноти документації або спостерігається неоднорідність стилів програмування. Вимагають високої уваги до деталей, здатності до зворотної інженерії та системного мислення для збереження стабільності продукту при внесенні змін. Ключові технічні компетенції - розуміння архітектури системи та володіння застосованими технологіями. Необхідні терплячість, послідовність у роботі та здатність підтримувати комунікацію із замовниками й користувачами, оскільки їхній зворотний зв'язок часто визначає пріоритети оновлень. Поділяються на 5 підтипів (рис. 4).

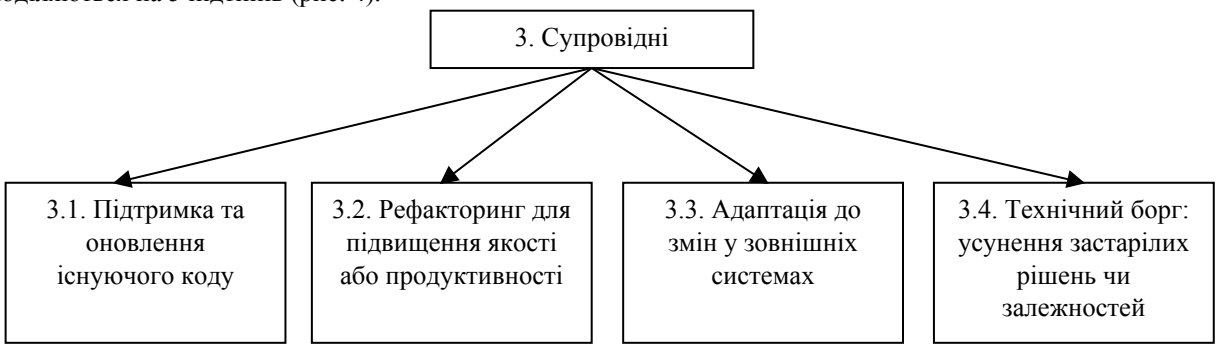

Рис. 4. Класифікація завдань типу *Супровідні*
Fig. 4. Classification of Supporting Type Tasks

Для завдань типу *3.1 Підтримка та оновлення існуючого коду* важливою навичкою є самоорганізованість.

Для завдань типу *3.2 Рефакторинг для підвищення якості або продуктивності* важливими є володіння технологіми для завдання і навичка самоорганізації.

Для завдань типу *3.3 Адаптація до змін у зовнішніх системах* навичка самоорганізації є важливою, а для *3.4 Технічний борг: усунення застарілих рішень чи залежностей* ця навичка є критичною.

Екстрені завдання характеризуються високою невизначеністю і гострим дефіцитом часу, що потребує від спеціаліста максимальної концентрації та вміння приймати рішення в умовах стресу. Технічні компетенції включають глибоке розуміння архітектури системи, інструментів моніторингу та діагностики, методів швидкого пошуку й усунення дефектів. Необхідні стресостійкість, здатність до швидкої комунікації та командної взаємодії

під тиском. Розуміння предметної області важливе при усунені збоїв, оскільки дозволяє спеціалісту правильно оцінити критичність проблеми для бізнесу і визначити пріоритетність дій [14]. Поділяються на 3 підтипи (рис. 5).

Рис. 5. Класифікація завдань типу *Екстрені*
Fig. 5. Classification of Emergency Type Tasks

Для завдань типу *4.1 Усунення критичних збоїв у продуктивному середовищі* критичним є володіння технологіями для завдання, навички вирішення проблем і самоорганізація, важливими – навички роботи в команді, критичного мислення і креативності.

Для завдань типу *4.2 Оперативна ліквідація вразливостей безпеки* та *4.3 Відновлення працездатності після аварій* ключовими є володіння технологіями для завдання, навички вирішення проблем та самоорганізації, важливими – навички критичного мислення, роботи в команді і креативності. При цьому для завданнь типу 4.2 також важливим є розуміння предметної області.

Інтеграційні завдання характеризуються необхідністю враховувати велику кількість взаємозалежностей, що робить їх складними з точки зору системної інженерії. Вимагаються знання архітектурних патернів інтеграції, навички роботи з інструментами оркестрації та глибоке розуміння протоколів обміну даними. Важливі комунікаційні здібності, оскільки інтеграція вимагає координації між різними командами чи зовнішніми постачальниками. Розуміння предметної області дозволяє передбачити можливі ризики на стику технологій та налаштовувати інтеграційні процеси для досягнення стабільності й продуктивності системи [15]. Поділяються на 3 підтипи (рис. 6).

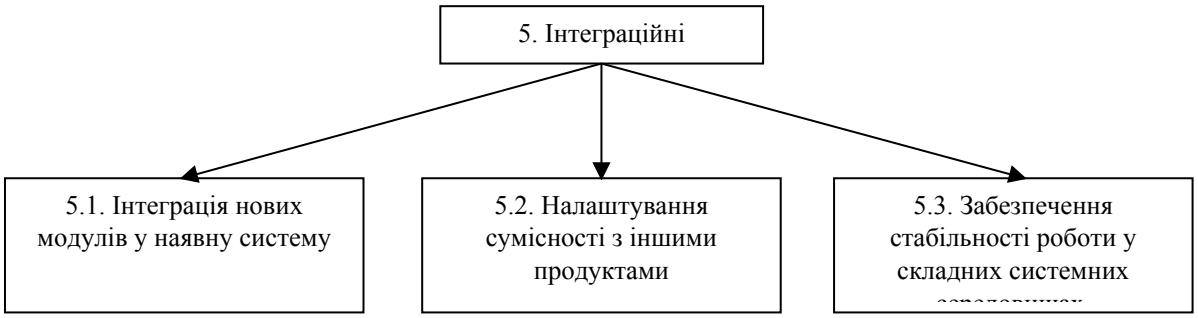

Рис. 6. Класифікація завдань типу *Інтеграційні*
Fig. 6. Classification of Integration Type Tasks

Для завдань типу *5.1 Інтеграція нових модулів у наявну систему* важливими є володіння широким спектром технологій, володіння технологіями для завдання і навичка самоорганізації.

Для завдань типу *5.2 Налаштування сумісності з іншими продуктами* важливими є володіння широким спектром технологій і володіння технологіями для завдання.

Для завдань типу *5.3 Забезпечення стабільності роботи у складних системних середовищах* володіння технологіями для завдання є критичним, навички вирішення проблем і робота в команді – важливими.

Зібрані дані наочно ілюструють різницю в патернах вимог до кваліфікацій і навичок спеціалістів для виконання завдань різних типів.

5. МОДЕЛЬ ЗАВДАННЯ

Запропоновано загальну модель завдання на створення ПЗ:

$$Task = \langle Id, Name, Details, Priority, Duration, DateF, SkillRequirements \rangle, \quad (1)$$

де *Id* – ідентифікатор завдання,
Name – назва завдання,
Details – текст завдання,
Duration – кількість днів що потрібна на виконання завдання,
DateF – очікувана дата закінчення виконання завдання,
SkillRequirements – вимоги до кваліфікацій і навичок до виконавця завдання.

Моделі конкретних завдань відрізняються від загальної моделі складом та значеннями елементів *SkillRequirements*. Кожен елемент є вимогою до кваліфікації або навички, які у свою чергу можуть мати різну «вагу» для різних типів завдань. Тому запропоновано використання рівня кваліфікацій і навичок – *SkillLevel*, який може приймати три значення: A – критично важливо, B – важливо і C – бажано.

Надалі для подання завдання певного типу будемо формувати для нього відповідний набір вимог *SkillRequirements* які повною мірою його ідентифікують. Для завдання типу *1.1 Створення архітектур* в модель слід ввести:

$$SkillRequirements_{1.1} = \left\langle \begin{matrix} VericalRequirement = A, sHardSkillRequirement, \\ ProjectSuccess = A,\ ProblemSolvingSkill = A, \\ CriticalThinkingSkill \geq B,\ SelfOrganizationSkill \geq B, \\ CreativitySkill \geq B \end{matrix} \right\rangle, \quad (2)$$

де *VerticalRequirement* – володіння предметною областю,

HardSkillRequirement – володіння технічними кваліфікаціями, представлене як множина технологій, де $HardSkillRequirement_i \geq SkillLevel_j$,

ProjectSuccess – успішність виконання аналогічних завдань в минулому,

ProblemSolvingSkill – володіння навичкою вирішення проблем,

CriticalThinkingSkill - володіння навичкою критичного мислення,

SelfOrganizationSkill - володіння навичкою самоорганізації,

CreativitySkill - володіння навичкою креативності.

Для завдання типу *1.2 Створення нового функціоналу* в модель слід ввести

$$SkillRequirements_{1.2} = \left\langle \begin{matrix} VericalRequirement \geq B, sHardSkillRequirement, \\ ProjectSuccess \geq B,\ ProblemSolvingSkill \geq B, \\ CriticalThinkingSkill \geq B,\ SelfOrganizationSkill \geq B, \\ CreativitySkill \geq B,\ TeamWorkSkill \geq B \end{matrix} \right\rangle, \quad (6)$$

де *TeamWorkSkill* – рівень володіння навичкою командної роботи, решта аналогічні (5).

Легко бачити, що вимоги до кваліфікацій і компетенцій для різних типів завдань мають суттєві відмінності навіть в рамках однієї групи. Далі в статті наводяться вимоги для деяких типів завдань виключно у якості прикладу.

Для завдань типу *2.2 Оцінка можливості застосування нової технології* в модель слід ввести:

$$SkillRequirements_{2.2} = \left\langle \begin{matrix} HardSkillRequirement, \\ ProjectSuccess = A, SelfOrganizationSkill \geq B, \\ TeamWorkSkill \geq B, CriticalThinkingSkill = A \end{matrix} \right\rangle. \quad (7)$$

Для завдань типу *3.1 Підтримка та оновлення існуючого коду* модель прийме вигляд:

$$SkillRequirements_{3.1} = \left\langle \begin{matrix} VericalRequirement \geq C, HardSkillRequirement, \\ ProjectSuccess \geq B, SelfOrganizationSkill \geq B, \\ TeamWorkSkill \geq C, CriticalThinkingSkill \geq C \end{matrix} \right\rangle. \quad (8)$$

Для завдань типу *4.2 Оперативна ліквідація вразливостей безпеки* в модель слід ввести:

$$SkillRequirements_{4.2} = \left\langle \begin{matrix} VericalRequirement \geq B, HardSkillRequirement, \\ ProjectSuccess = A, SelfOrganizationSkill = A, \\ TeamWorkSkill \geq B, CriticalThinkingSkill = A, \\ CreativitySkill \geq B, ProblemSolvingSkill = A \end{matrix} \right\rangle. \quad (9)$$

Для завдань типу *5.1 Інтеграція нових модулів у наявну систему* в модель слід ввести:

$$SkillRequirements_{5.1} = \left\langle \begin{matrix} HardSkillRequirement, \\ ProjectSuccess \geq B, SelfOrganizationSkill \geq B, \\ TeamWorkSkill \geq C, CriticalThinkingSkill \geq C \end{matrix} \right\rangle. \quad (10)$$

SkillRequirements для всіх типів завдань не наводяться з огляду на обмеження обсягу статті. Наведені вимоги можуть бути модифіковані шляхом розширення або зміни рівнів володіння навичками в рамках організації що імплементує підхід на розсуд відповідальних осіб.

6. МЕТОД ВИБОРУ СПЕЦІАЛІСТА ДЛЯ ВИКОНАННЯ ЗАВДАННЯ

Метод вибору спеціаліста для виконання завдання складається з 5 кроків.

Крок 1. Визначається тип завдання відповідно до запропонованої класифікації. Ідентифікуються необхідні для виконання завдання технології і визначаються рівні володіння ними. Згідно з запропонованою моделлю, відповідно до типу завдання, визначаються необхідні рівні володіння комунікаційними навичками. За потреби, вимоги коригуються. Результатом виконання кроку є структуровані відповідно до моделі вимоги до виконавця завдання.

Крок 2. Актуалізуються дані щодо кваліфікацій і навичок наявних спеціалістів в розрізі кваліфікацій і навичок, ідентифікованих на кроці 1. У випадку наявності актуальних даних цей крок пропускається. Результатом виконання кроку є структуровані дані щодо кваліфікацій і навичок спеціалістів організації відповідно до вимог завдання.

Крок 3. Виконують відбір кандидатів шляхом фільтрування списку спеціалістів відповідно до умов зазначених в моделі, починаючи з найбільш важливих вимог (значення *SkillLevel: A, критично важливо*), продовжуючи фільтрування за спадом важливості вимоги.

Якщо по завершенню відбору список кандидатів є порожнім, виконується переоцінка необхідності кваліфікацій і навичок в сторону раціонального зменшення вимог, після чого крок 3 повторюється.

Крок 4. Згідно з підходами запропонованими у [16] визначається можливість залучення обраних спеціалістів до вирішення поставленого завдання. Для кожного з кандидатів виконується аналіз можливості видачи йому завдання з урахуванням поточної зайнятості, часу потрібного на виконання завдання і очікуваної дати закінчення виконання завдання. Кандидати, при залученні яких необхідна зміна черговості виконання інших завдань, або неможливе виконання в термінах строку *DateF*, відкидаються.

Якщо в результаті отриманий порожній список кандидатів, крок повторюється, але відкидаються тільки кандидати для яких неможливе виконання в термінах строку *DateF,* тобто, зміна черговості виконання інших завдань допускається.

Якщо після повторення ітерації список кандидатів порожній, виконується переоцінка необхідності кваліфікацій і навичок в сторону раціонального зменшення вимог, після чого крок 3 повторюється.

Крок 5. На підставі даних отриманих на кроках 3 і 4, обирається виконавець що за сукупністю зібраних метрик в найбільшій мірі підходить для виконання завдання.

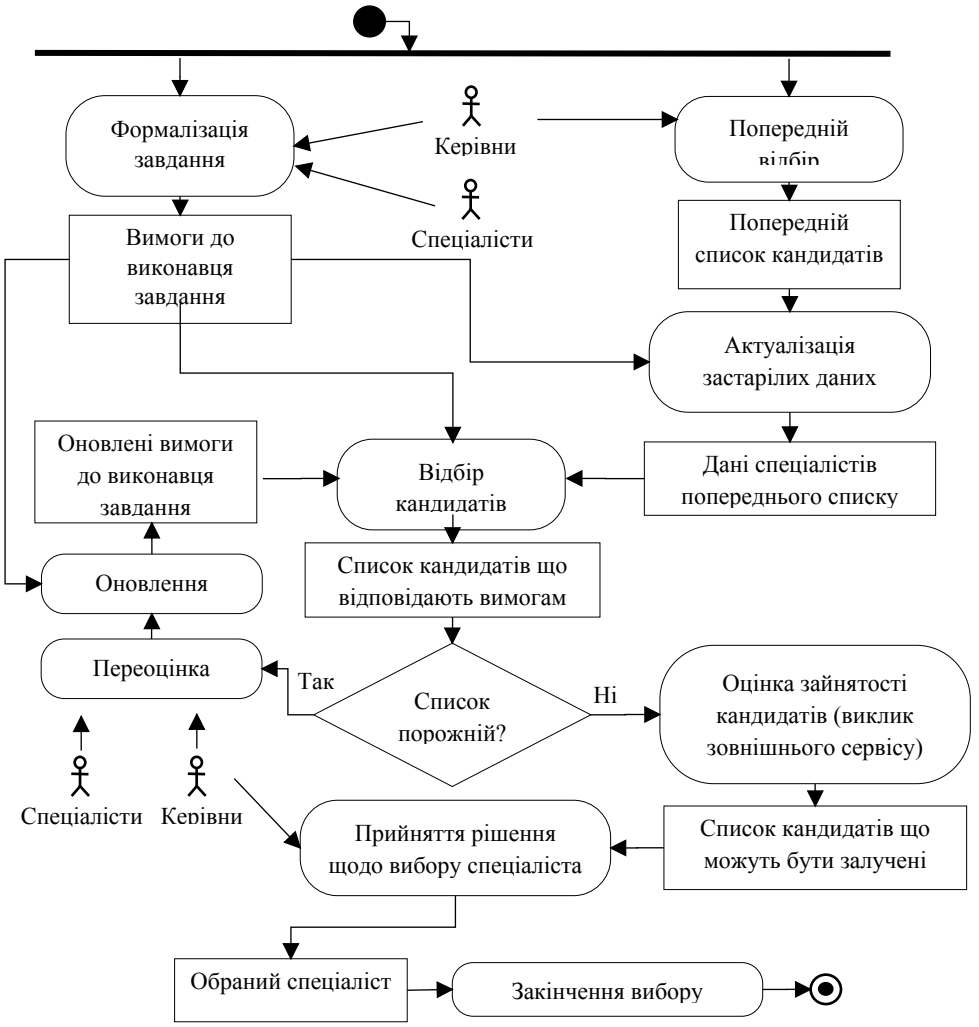

Рис. 7. Послідовність дій запропонованого методу
Fig. 7. Sequence of Actions of the Proposed Method

Послідовність дій методу відображено на рис. 7. Елемент *Попередній вибір спеціалістів* передбачає вибір керівником потенційних виконавців завдання.

7. АПРОБАЦІЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

Для реалізації запропонованого методу створено ПЗ SpecMatch що складається з модулів Task, Specialist і Matcher. Модуль Task приймає ідентифікатор завдання, пропонує обрати його тип згідно з запропонованою класифікацією, обрати вимоги щодо технологій відповідно до завдання і обрати необхідний ступінь володіння ними. Модуль додає до вимог комунікаційні навички і вимоги щодо володіння ними згідно з запропонованою моделлю. Користувач за потреби може змінити необхідні рівні володіння кваліфікаціями та навичками і їх перелік. Модуль Specialist приймає ідентифікатори спеціалістів, їх кваліфікації і навички з рівнем володіння ними. Модуль Matcher виконує співставлення кваліфікацій і навичок Specialist з вимогами Task і пропонує перелік спеціалістів, що відповідають вимогам завдання. ПЗ SpecMatch використовується в якості системи підтримки прийняття рішень. На рис. 8 представлено список кандидатів що відповідають вимогам завдання для подальшого прийняття рішення щодо призначення виконавця.

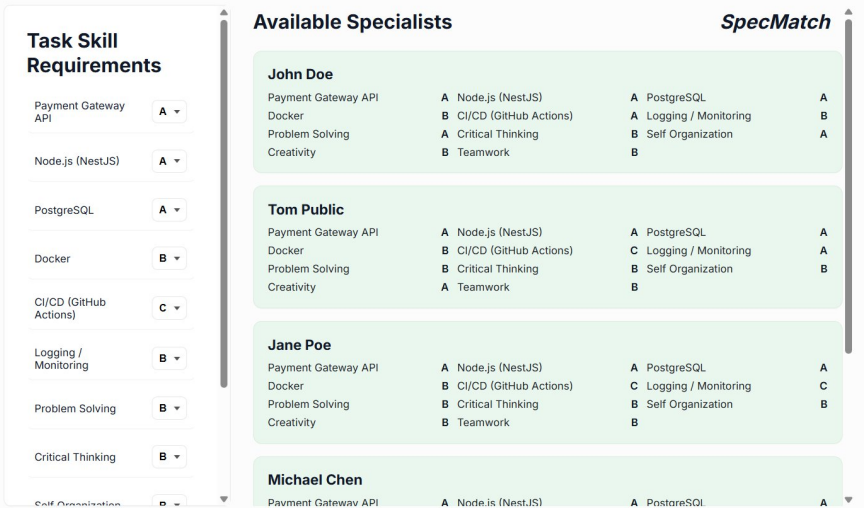

Рис. 8. Список відібраних спеціалістів для виконання завдання
Fig. 8. List of Selected Specialists for Task Execution

Для апробації запропонованого методу було сформовано 4 різних завдання на розробку ПЗ що потребують приблизно 2 тижнів роботи кожне. Завдання належали до різних типів і вимагали використання різних технологій. Для перевірки ефективності методу були залучені експерти що спеціалізуються на розробці ПЗ з різних організацій. Експертам були видані завдання з запитом на вибір спеціалістів для виконання кожного з завдань в доступних їм командах розробки, для 2 завдань – без використання методу, для інших 2 завдань – з використанням запропонованого методу. Експерти мали залучити до обговорення завдання спеціалістів своєї організації на власний вибір. Замірявся час прийняття рішення для вибору виконавця завдання, результати наведено в табл. 1.

Таблиця 1. Витрати часу на вибір виконавця для завдання, годин

Експерт	Традиційним способом					З застосуванням запропон. методом				
	Завд. 1	Завд. 2	Завд. 3	Завд. 4	Сер.	Завд. 1	Завд. 2	Завд. 3	Завд. 4	Сер.
Експ. 1	1.75	1.5			1.63			1.25	1.25	1.25
Експ. 2	1.5			2	1.75		1.25	1.5		1.13
Експ. 3			1.5	1.75	1.63	1	1.25			1.13
Експ. 4		2	2.25		2.13	1.5			1.25	1.38
Експ. 5			1.75	1.75	1.75	1.25	1			1.13
Експ. 6		1.75	1.5		1.63	1			1.25	1.13
Експ. 7	1.75			2	1.88		1	1.25		1.13
Експ. 8	2	2			2			1.5	1.75	1.63
Експ. 9			2	1.75	1.88	1			1.5	1.25
Середнє	1.75	1.81	1.8	1.85	1.8	1.15	1.13	1.25	1.4	1.23

Аналізи результатів експерименту свідчать що використання запропонованої технології призводять до зменшення часу на підготовку завдання і вибір спеціаліста на 31%.

Обмеженість ресурсів під час проведення експерименту не дозволила авторам залучити більше експертів, але відсутність значного відхилення часу на визначення спеціаліста від середнього для кожного з завдань вказує те що це не внесло значної похибки. Грунтуючись на цьому автори статті вважають що для даного експерименту кількість спостережень за часом є достатньою.

Ефективність методу може бути поясненою формалізацією параметрів відбору, забезпеченням порівнюваності характеристик, що дозволяють зменшити суб'єктивність в процесі відбору, і забезпеченням

прозорості процесу прийняття рішень. Сукупність цих елементів призводить до зменшення витрат часу на вибір спеціаліста для виконання завдання.

З обмежень методу необхідно зазначити відсутність аналізу розвитку та втрачання навичок і кваліфікацій з часом. Обмеженість обсягу цього дослідження не дозволила приділити достатньо уваги цим факторам. Також, метод не є пристосованим для вибору кандидатів для завдань, де потрібна комбінація ролей виконавця (наприклад, архітектор + фахівець з безпеки). Для таких ситуацій можливе послідовне застосування методу з відповідними критеріями і вибір виконавця з перетинань запропонованих множин кандидатів.

8. ВИСНОВКИ

Проаналізовано існуючі способи подання завдань в області розробки програмних продуктів. Встановлено, що не існує класифікації та формального подання завдань, які дозволяють сформулювати чіткі вимоги до вибору його виконавця.

Розроблено класифікацію завдань на основі аналізу процесів створення ПЗ. Визначені класи відрізняються характером діяльності, застосуванням технологій, вимогами до володіння технічними компетенціями і комунікаційними навичками потенційних виконавців. Запропонована класифікація є основою для автоматизації визначення виконавців для завдань.

На основі класифікації створено загальну модель, що вміщує елементи спільні для завдань всіх визначених класів. Розроблено моделі які відповідають окремим підкласам класифікації і уточнюють загальну модель.

На основі створених моделей запропоновано метод вибору найкращого із наявних виконавців для виконання конкретного завдання з створення ПЗ. Метод забезпечує врахування як технічних кваліфікацій, так і комунікативних навичок, що підвищує точність вибору виконавця. Збільшення відповідності виконавців вимогам проектних завдань має привести до скорочення часу на створення і підвищення якості програмних продуктів. Метод передбачає визначення типу завдання, вимог до виконавця, уточнення даних виконавців і вибір виконавця на основі визначених критеріїв. Метод завершується прийняттям рішення щодо залучення виконавця.

З метою апробації запропонованого методу було створено ПЗ SpecMatch що функціонує як система підтримки прийняття рішень. Перевірка ефективності методу відбувалась шляхом його застосування при виборі виконавців. Апробація включала в себе порівняння традиційної технології і запропонованого в роботі методу. Застосування запропонованого методу дозволило скоротити витрати часу на підготовку завдання і прийняття рішення щодо вибору виконавця на 31%.

Література

1. Nurdin, M. (2023). The Effect of Work Experience and Hard Skill on Employee Performance of PT. Multi Kencana Niagatama. At-Tadbir: jurnal ilmiah manajemen, 7(1), 41-53. http://dx.doi.org/10.31602/atd.v7i1.9221
2. Li, P.L., Ko, A.J. & Begel, A. What distinguishes great software engineers?. Empir Software Eng 25, 322–352 (2020). https://doi.org/10.1007/s10664-019-09773-y
3. Diniz, W., Valença, M., França, C., Santos, A., & Pincovsky, M. (2024, September). The skill gap in software industry: A mapping study. In Simpósio Brasileiro de Engenharia de Software (SBES) (pp. 192-200). SBC. https://doi.org/10.5753/sbes.2024.3351.
4. Nana Assyne, Hadi Ghanbari, Mirja Pulkkinen. The essential competencies of software professionals: A unified competence framework. Information and Software Technology, Volume 151, 2022, 107020. https://doi.org/10.1016/j.infsof.2022.107020
5. Goth, F., Alves, R., Braun, M., Castro, L. J., Chourdakis, G., Christ, S., ... & Wittke, S. (2024). Foundational Competencies and Responsibilities of a Research Software Engineer. F1000Research, 13, 1429. https://doi.org/10.48550/arXiv.2311.11457
6. Chen, S., & Zhao, Y. (2025). Beyond Hard vs. Soft: A Mixed-Methods Approach to Developing a Competency Model for E-commerce Software Engineers. Information and Software Technology, 107836. https://doi.org/10.1016/j.infsof.2025.107552
7. Fabio Santos. 2023. Supporting the Task-driven Skill Identification in Open Source Project Issue Tracking Systems. SIGSOFT Softw. Eng. Notes 48, 1 (January 2023), 54–58. https://doi.org/10.1145/3573074.3573088
8. Gupta C, Gupta V. A Decentralized Framework for Managing Task Allocation in Distributed Software Engineering. Applied Sciences. 2021; 11(22):10633. https://doi.org/10.3390/app112210633
9. Doblinger, M. (2021). Individual Competencies for Self-Managing Team Performance: A Systematic Literature Review. Small Group Research, 53(1), 128-180. https://doi.org/10.1177/10464964211041114
10. Borges, G. G., & de Souza, R. C. G. (2024). Skills development for software engineers: Systematic literature review. Information and Software Technology, 168, 107395. https://doi.org/10.1016/j.infsof.2023.107395
11. Luisa Greifenstein, Ute Heuer, and Gordon Fraser. 2023. 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 2023), July 8–12, 2023, Turku, Finland. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3587102.3588809.

12. Fagerholm, F, Felderer, M, Fucci, D, Unterkalmsteiner, M, Marculescu, B, Martini, M, Wallgren Tengberg, L G, Feldt, R, Lehtelä, B, Nagyvaradi, B & Khattak, J 2022, 'Cognition in Software Engineering: A Taxonomy and Survey of a Half-Century of Research', ACM Computing Surveys, vol. 54, no. 11, 226, pp. 1-36. https://doi.org/10.1145/3508359
13. Sunakshi Tongia, Dr. Charul Jain, et. al, The Relevance Of Soft Skills In Career Success, Educational Administration: Theory and Practice, 2024; 30(1), 1809-1812. https://doi.org/10.53555/KUEY.V30I1.6642
14. Ghatak, S. K., & Mahanty, B. (2023). Impact of knowledge growth and team composition on the co-located software project performance. Knowledge Management Research & Practice, 21(2), 397-411. https://doi.org/10.1080/14778238.2021.1939173
15. Özkan, O., Babur, Ö., & van den Brand, M. (2025). Domain-Driven Design in software development: A systematic literature review on implementation, challenges, and effectiveness. Journal of Systems and Software, 112537. https://doi.org/10.48550/arXiv.2310.01905
16. Kungurtsev O.B., Chorba R.V. Task execution flow management in the software development process under the minor change event. Herald of Advanced Information Technology. Odesa: Ukraine. 2023; p 6 № 4: 297–307. DOI: https://doi.org/10.15276/hait.06.2023.19

Отримана в редакції 12.06.2025. Прийнята до друку 18.06.2025. Received 12 June 2025. Approved 18 June 2025. Available in Internet 30 June 2025