

АВТОМАТИЗАЦІЯ ТА КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ

УДК 658.512

DOI: 10.30977/BUL.2219-5548.2022.97.0.16

МОДЕЛЬ ВИБОРУ SCRUM-МАЙСТРА ІЗ ЗАСТОСУВАННЯМ НЕЧІТКИХ МНОЖИН

Нефьодов Л. І.¹, Філь Н. Ю.¹¹Харківський національний автомобільно-дорожній університет

Анотація. Для розроблення сучасного програмного забезпечення використовуються гнучкі методології. У роботі проаналізовано гнучкі методології проєктів розробки програмного забезпечення. Найпоширенішою є методологія SCRUM. Розглянуто проблеми розподілу робіт між виконавцями проєкту розробки програмного забезпечення за методологією SCRUM. Розроблено нечітку модель вибору SCRUM-майстра для управління проєктом розробки програмного забезпечення. Наведено приклад розподілу робіт між виконавцями із застосуванням теорії нечітких множин. Розрахунки виконувались у Microsoft Excel. Подальші дослідження будуть спрямовані на практичну реалізацію моделі у вигляді програмно-методичного комплексу.

Ключові слова: програмне забезпечення, гнучка методологія, нечіткі множини, SCRUM-майстер, функція належності.

Вступ

Складність програмного забезпечення (ПЗ) неухильно зростає. Користувача сьогодні цікавить ПЗ, яке інтегровано в усі аспекти життя. Збільшення кількості різноманітних програмних застосунків та їхня складність не можливі без створення великих, добре налагоджених команд для розроблення ПЗ.

Для створення більшості сучасного ПЗ використовуються гнучкі методології. Найпоширеніша – це методологія SCRUM, що дозволяє досягти необхідних результатів у 2,53 рази швидше, ніж з моделлю управління водоспадами [1].

SCRUM-методологія є гнучким фреймворком Agile-системи, на основі якої з'явилося різноманіття гнучких методів, проте кожен зі своїми особливостями та умовами використання. Основна ідея, що характеризує SCRUM – «робота має бути видимою» [1; 2]. SCRUM передбачає поділ проєкту на підзадачі, що виконуються за короткі проміжки часу (від тижня до чотирьох) – спринти, протягом яких група, відповідальна за виконання певної підзадачі, має або повністю виконати її, або показати мінімально працездатний продукт, який задовольняє власника продукту, дати йому змогу визначити, що замовника не задовольняє або що він хотів би побачити чи змінити в готовому продукті. Керує роботою команди SCRUM-майстер.

SCRUM-майстер – найважливіша роль у методології. Він відповідає за успіх SCRUM у проєкті. Власне, SCRUM-майстер є інтерфейсом між власником та командою проєкту.

SCRUM-майстер – це різнобічна роль, він вміє застосовувати свою роль у різних ситуаціях. Це допомагає команді правильно приймати рішення.

SCRUM-майстер може виконувати різні ролі в проєкті.

SCRUM-майстер – насамперед це лідер, чия увага зосереджена на потребах команди та клієнтів для досягнення результатів відповідно до цінностей, принципів та бізнес-цілей організації [2; 3].

SCRUM-майстер – це посередник, який встановлює та підтримує чіткі межі для командної взаємодії.

SCRUM-майстер – це тренер, який допомагає колегам сформувати спосіб мислення.

SCRUM-майстер – це регулятор конфліктів, що вирішує ситуації з неконструктивною поведінкою та складними відносинами.

SCRUM-майстер – це менеджер, який відповідає за усунення перешкод, скорочення втрат, навігацію процесів, здоров'я команди, самоорганізацію та управління культурою.

SCRUM-майстер – це наставник, який передає знання та досвід команді.

SCRUM-майстер – це вчитель, який має переконатися, що принципи SCRUM зрозумілі та прийняті.

Тобто, SCRUM-майстер – це дуже важлива спеціальна роль у методології SCRUM.

Мета SCRUM-майстра полягає в тому, щоб навчити команду працювати за SCRUM, допомогти їй покращувати методи роботи та привести її до самоврядування та високої ефективності. До того ж SCRUM-майстер не має владних повноважень, він не може роздавати завдання, не може приймати рішення за команду або окремих її учасників [3].

Сьогодні не існує програм, які професійно навчали методології SCRUM, що стрімко поширюється в Україні.

Отже, вибір SCRUM-майстра для проекту розробки ПЗ є актуальним науково-прикладним завданням.

Аналіз публікацій

Чимало робіт іноземних учених присвячено SCRUM-методології. Основним принципам цієї методології та умовам її застосування присвячені роботи М. Fowler, J. Highsmith [4], L. Rising, N. Janoff [5], P. Diebold [6].

Х. Кніберг у своєму дослідженні стверджує, що SCRUM-майстер є членом команди, який відповідає за вчасне вирішення поточних проблем членів команди, а SCRUM-команда є запорукою успіху всього проекту [3]. J. Bass у роботі [7] розглядає окремо роль SCRUM-майстра.

У праці [8] розглянуто SCRUM-методологію, а також ті методології, які застосовуються для дослідження моделей командної роботи, питань групової динаміки, лідерства, зокрема лідерства-служіння. У дослідженні зазначено, що SCRUM-методологія зосереджена на робочому процесі й не враховує питання формування колективу та розвитку його членів. Розглянуто модель роботи SCRUM-майстра з розвитку лідерських компетенцій на різних етапах реалізації проекту.

Аналіз літератури показав, що на сьогодні не приділяється достатньої уваги науковому обґрунтуванню вибору SCRUM-майстра для проектів розробки ПЗ з урахуванням індивідуальних характеристик і переваг людини.

Мета та постановка завдання

Метою роботи є підвищення ефективності проектів розробки ПЗ за рахунок розроблення моделі вибору SCRUM-майстра із застосуванням нечітких множин для проектів розробки ПЗ, яка дозволить враховувати індивідуальні характеристики та переваги членів SCRUM-команди проекту.

Загальна постановка задачі вибору SCRUM-майстра для управління проектом розробки ПЗ

Якщо є команда, яка буде виконувати проект, то SCRUM-майстер може бути її членом. Тобто роль SCRUM-майстра є однією із задач проекту. Задача вибору SCRUM-майстра зводиться до раціонального розподілу задач журналу проекту між виконавцями, в якій для виконання кожної задачі проекту розробки ПЗ потрібно один і тільки один виконавець, а кожен виконавець може бути призначений на одну й тільки одну задачу проекту розробки ПЗ. Тобто виконавці не подільні між задачами проекту розробки ПЗ, а задачі не ділені між виконавцями. Отже, задача вибору SCRUM-майстра є окремим випадком транспортної задачі [9–10].

Розглянемо загальну постановку задачі.

Відомо:

– множина задач проекту розробки ПЗ, які необхідно виконати B_i ($i = \overline{1, m}$), де m – кількість задач проекту розробки ПЗ;

– множина виконавців A_j ($j = \overline{1, n}$), де n – кількість виконавців;

– c_{ij} компетентність виконавця A_j для реалізації задачі B_i (можна розглядати якість виконання j -ої задачі i -м виконавцем) [11].

Введемо змінну x_{ij} , яка $x_{ij} = 1$, якщо j -го виконавця призначили на i -ту роботу, інакше $x_{ij} = 0$.

Необхідно знайти $L(X)$ – загальну (сумарну) характеристику якості розподілу виконавців для робіт за проектом [9]:

$$L(X) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \max; \quad (1)$$

за таких умов:

– усі виконавці мають бути призначені на роботу

$$\sum_{j=1}^n x_{ij} = n \quad (i = \overline{1, m}); \quad (2)$$

– усі роботи мають бути виконані

$$\sum_{i=1}^m x_{ij} = m \quad (j = \overline{1, n}). \quad (3)$$

Задача про призначення широко використовується в прикладній діяльності та має множину інтерпретацій.

Існують стандартні алгоритми пошуку оптимального розв'язання задачі про призначення з елементарною лінійною моделлю, що дозволяють отримати точне рішення за поліноміальний час, наприклад, угорський метод. Однак формулювання більшості прикладних задач про призначення не відповідає елементарній лінійній моделі. Часто запропоновані алгоритми рішення узагальнених задач про призначення визначаються громіздкістю використаного математичного апарату, а також не завжди гарантують знаходження оптимального рішення, що може призвести до значних економічних втрат у контексті багатьох прикладних задач [9–10].

Більш того, на сьогодні для вирішення такого завдання можливе використання генетичного алгоритму та його модифікації, тобто виникає досить складна багатокритеріальна задача пошуку оптимальної альтернативи.

Але всі ці математичні методи не відображають нечіткий характер взаємодії різних суб'єктів у задачі.

Постановка задачі SCRUM-майстра із застосуванням нечітких множин

Розглянемо нечітку модель розподілу робіт, які виконуються за проектом розробки ПЗ.

Очевидно, що розподіляючи різні роботи, власник проекту приймає рішення суб'єктивно. Він не має змоги оцінити компетентності кожного виконавця проекту, ступінь його освіченості, знайомства з предметною галуззю, досвід роботи, ті чи інші риси характеру, завантаженість виконавця протягом усього робочого часу тощо [11]. Тому виникає проблема оптимального розподілу задач між виконавцями проекту із врахуванням індивідуальних переваг до тієї або іншої задачі проекту розробки ПЗ.

Відомо

- множина задач $X = \{x_i\}, (i = \overline{1, N})$ за проектом, яку необхідно виконати за визначений період часу;

- множина характеристик кожної задачі $Y = \{y_j\}, (j = \overline{1, P})$;

- множина виконавців $Z = \{z_k\}, (k = \overline{1, M})$, які є членами SCRUM-команди проекту розробки ПЗ. Кожен виконавець відрізняється ступенем переваги щодо тієї чи іншої задачі.

Ці переваги визначаються багатьма факторами.

Необхідно оптимально розподілити всі задачі серед виконавців, тобто для кожного виконавця z_j сформулювати множину задач

$$R_j \{x_i\}, \text{ так щоб виконувалися вимоги } \bigcup_j R_j \{x_i\} = X \text{ і } \bigcap_j R_j \{x_i\} = \emptyset.$$

У загальному випадку для кожного виду задач можна сформулювати свій унікальний набір характеристик, але для простоти обчислень ми обмежимося лише декількома, найбільш загальними.

Кожній конкретній задачі буде притаманна характеристика деякою мірою.

Розглянемо, наприклад, характеристику «рівень кваліфікації». Рівень кваліфікації дозволяє виконавцю проекту розробки ПЗ правильно розподілити робоче навантаження протягом усього робочого дня. Власник може дати експертну оцінку значення функції належності конкретної задачі із всієї множини журналу проекту, які вимагають високої кваліфікації. Очевидно, що всі виконавці мають різний рівень кваліфікації. Наприклад, для виконавця, що має невеликий досвід роботи, вкрай важливим може бути характеристика «простота виконання», оскільки зі складною роботою він може не впоратися [11].

На першому кроці власником проекту проводиться експертне оцінювання, що дозволяє отримати формалізовану постановку задачі.

Нехай $r: X \times Y \rightarrow [0, 1]$ – функція належності нечіткого бінарного відношення R , що задається за допомогою експерта. Ця функція виражає, якою мірою задачі x_i властива характеристика y_j . Значення функції за конкретною задачею x_i запишемо в рядок.

Отже, R – нечітке відношення «задача – характеристика задачі», яке в матричній формі має вигляд:

$$R = \begin{matrix} & y_1 & y_2 & \dots & y_p \\ \begin{matrix} x_1 \\ x_2 \\ \dots \\ x_n \end{matrix} & \begin{bmatrix} \xi(x_1, y_1) & \xi(x_1, y_2) & \dots & \xi(x_1, y_p) \\ \xi(x_2, y_1) & \xi(x_2, y_2) & \dots & \xi(x_2, y_p) \\ \dots & \dots & \dots & \dots \\ \xi(x_n, y_1) & \xi(x_n, y_2) & \dots & \xi(x_n, y_p) \end{bmatrix} \end{matrix}, \quad (4)$$

де $\xi(x_i, y_j)$ – функція належності нечіткого відношення R , що відображає ступінь належ-

ності характеристики y_j задачі x_i . Елементи кожного рядка матриці R показують, якою мірою ця задача потребує наявності необхідних характеристик.

Нехай $s: Y \times Z \rightarrow [0, 1]$ – функція належності нечіткого бінарного відношення S . Для всіх $y \in Y$ і всіх $z \in Z$ $s(y, z)$ дорівнює ступеню важливості характеристики y_j для виконавця z_j . У матричній формі це відношення має вигляд:

$$S = \begin{matrix} & z_1 & z_2 & \dots & z_m \\ y_1 & \left[\begin{array}{cccc} \psi(y_1, z_1) & \psi(y_1, z_2) & \dots & \psi(y_1, z_m) \end{array} \right. \\ y_2 & \left[\begin{array}{cccc} \psi(y_2, z_1) & \psi(y_2, z_2) & \dots & \psi(y_2, z_m) \end{array} \right. \\ \dots & \left[\begin{array}{cccc} \dots & \dots & \dots & \dots \end{array} \right. \\ y_p & \left[\begin{array}{cccc} \psi(y_p, z_1) & \psi(y_p, z_2) & \dots & \psi(y_p, z_m) \end{array} \right. \end{matrix}$$

де $\psi(y_j, z_k)$ – функція належності нечіткого відношення S , що відображає ступінь важливості характеристики y_j для виконавця z_k .

Інакше, матриця S відображає відносний ступінь важливості характеристик робіт y_1, y_2, \dots, y_p для виконавців z_1, z_2, \dots, z_m у процесі виконання робіт журналу X .

Композиція $T = R \otimes S$ бінарних нечітких відношень R і S дозволяє встановити результуючий зв'язок між задачами та виконавцями у формі нечіткого відношення «задача – виконавець», що відображає ступінь переваги (можливості виконати) різні задачі (залежно від їхніх характеристик) з боку членів SCRUM-команди [11].

$$T = R \otimes S =$$

$$= \begin{matrix} & z_1 & z_2 & \dots & z_m \\ x_1 & \left[\begin{array}{cccc} \mu(x_1, z_1) & \mu(x_1, z_2) & \dots & \mu(x_1, z_m) \end{array} \right. \\ x_2 & \left[\begin{array}{cccc} \mu(x_2, z_1) & \mu(x_2, z_2) & \dots & \mu(x_2, z_m) \end{array} \right. \\ \dots & \left[\begin{array}{cccc} \dots & \dots & \dots & \dots \end{array} \right. \\ x_n & \left[\begin{array}{cccc} \mu(x_n, z_1) & \mu(x_n, z_2) & \dots & \mu(x_n, z_m) \end{array} \right. \end{matrix} \quad (5)$$

де $\mu(x_n, z_1)$ – функція належності композиції нечітких відношень, що відображає ступінь переваги виконавцем z_k розв'язати задачу x_i , визначається за формулою [11]:

$$\mu(x_i, z_k) = \frac{\sum_{j=1}^P \xi(x_i y_j) \otimes \psi(y_j, z_k)}{\sum_{j=1}^P \xi(x_i y_j)}, \quad (i = \overline{1, N}; k = \overline{1, M}). \quad (6)$$

Сума $\sum_{j=1}^P \xi(x_i y_j)$ дорівнює ступеню нечіт-

кої підмножини, що вказує на число найважливіших характеристик, які власник проекту або SCRUM-майстер використовує для оцінювання задачі x_i . Отже, можна інтерпретувати як зважений ступінь прийнятності задачі x_i для z_k члена SCRUM-команди.

Далі визначається матриця попарних переваг W , що показує ступінь прийнятності кожного виду задач одночасно для кожної комбінації з двох виконавців.

Матриця W дозволяє знайти поріг розподілу задач журналу, який визначається як ступінь прийнятності всієї множини робіт за журналом проекту одночасно для всіх пар виконавців.

Обчислення порога розподілу задач передбачає три кроки.

Крок 1. Визначаються максимальні значення за стовпцями матриці W .

Крок 2. З них вибирається мінімальна величина ω .

Крок 3. У матриці T знаходиться найбільший елемент строго менший, ніж ω , який і є порогом поділу задач λ .

Після того як поріг обраний, можна за стовпцями матриці T для кожного виконавця z_k визначити рівневу множину L , тобто множину задач, ступінь прийнятності якої для цієї групи виконавців перевищує або дорівнює порогу розподілу задач λ .

Вага кожної групи виконавців, членів SCRUM-команди, визначається за підсумками попередньої діяльності, задається ваговими функціями $\rho(z_k)$, $(k = \overline{1, M})$. Вагові функції можуть вимірюватися у відсотках або в частках від загального обсягу задач спринту.

Розподіл задач між виконавцями описується зваженим об'єднанням рівневих множин L_k :

$$L = \bigcup_{k=1}^p \rho(z_k) L_k. \quad (7)$$

Отже, використання моделі розподілу задач між членами SCRUM-команди, основаної на теорії нечітких відношень, допомагає власнику проекту оптимально розподілити задачі між виконавцями відповідно до принципу поєднання можливості та бажаності, комплексно врахувати їхні знання та досвід, а також зменшити завантаженість. Етапи розподілу задач між виконавцями в SCRUM-

команді проекту розробки ПЗ зображені на рис. 1.

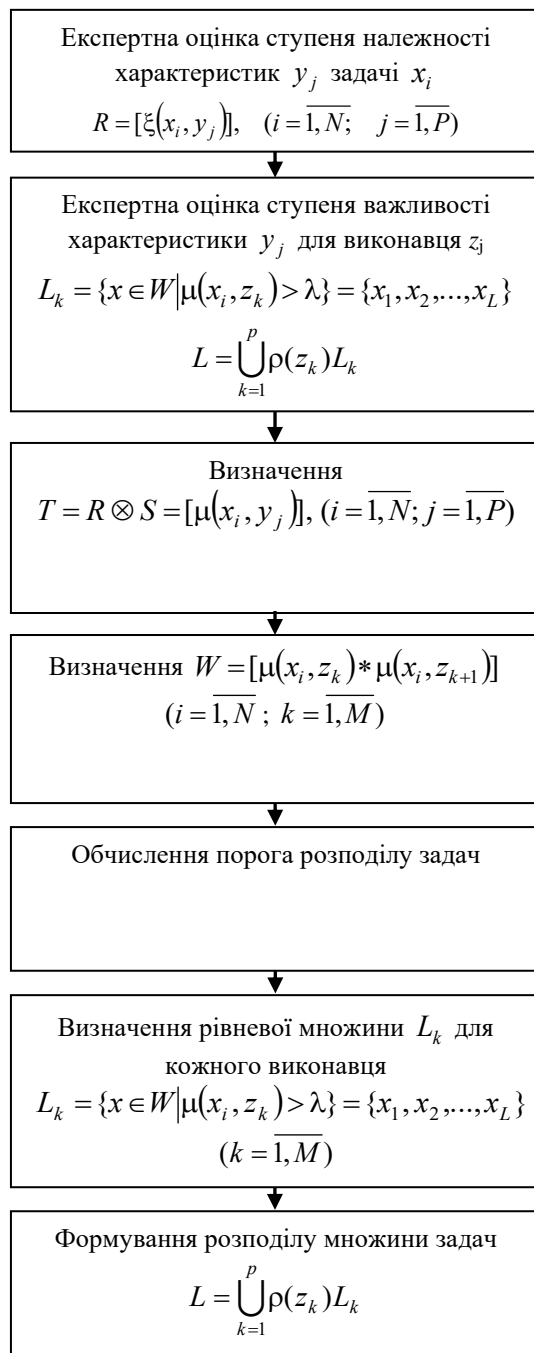


Рис. 1. Етапи формування розподілу задач між виконавцями за SCRUM-методологією

Приклад розподілу задач між виконавцями, членами SCRUM-команди, з використанням теорії нечітких множин

SCRUM-команда може налічувати від 4 до 9 осіб, з урахуванням SCRUM-майстра. Для ілюстрації розв'язання поставленого завдання розглядається перелік задач, які є в журналі.

- x_1 – SCRUM-майстер;
- x_2 – виконавець з малим досвідом роботи (менше ніж рік);
- x_3 – виконавець із середнім досвідом роботи (досвід роботи від 1 року до 3 років);
- x_4 – виконавець із середнім досвідом роботи (досвід роботи від 3 до 5 років);
- x_5 – виконавець із великим досвідом роботи (досвід роботи понад 5 років).

Задачі можуть виконувати множина виконавців, членів SCRUM-команди, як $Z = \{z_1, z_2, \dots, z_M\}$:

- z_1 – Петров;
- z_2 – Іванов;
- z_3 – Сидоров;
- z_4 – Швецов;
- z_5 – Клусович.

Позначимо через $Y = \{y_1, y_2, \dots, y_P\}$ множину характеристик кожної задачі:

- y_1 – швидкість і гнучкість мислення;
- y_2 – вміння швидко приймати рішення;
- y_3 – стійкість і концентрація уваги;
- y_4 – зорова пам'ять;
- y_5 – швидкість реакції;
- y_6 – вміння конструктивного спілкування;
- y_7 – вміння синхронізувати роботи;
- y_8 – вміння організувати роботу команди;
- y_9 – емоційно-вольова стійкість;
- y_{10} – відповідальність.

На рис. 2 наведено функції належності нечітких множин для відповідних задач (посад).

	швидкість і гнучкість мислення	вміння швидко приймати рішення	стійкість і концентрація уваги	зорова пам'ять	швидкість реакції	вміння конструктивного спілкування	вміння синхронізувати роботи	вміння організувати роботу команди	емоційно-вольова стійкість	відповідальність	Сума
SCRUM-майстер	0.9	0.9	0.8	0.3	0.4	0.9	0.8	0.9	0.9	0.9	7.7
виконавець з малим досвідом роботи	0.3	0.3	0.6	0.5	0.3	0.3	0.2	0.2	0.5	0.6	3.8
виконавець з середнім досвідом роботи	0.4	0.4	0.4	0.5	0.6	0.3	0.6	0.6	0.6	0.5	4.9
виконавець з середнім досвідом роботи	0.4	0.4	0.4	0.5	0.6	0.3	0.6	0.6	0.6	0.5	4.9
виконавець з великим досвідом роботи	0.5	0.5	0.6	0.6	0.7	0.5	0.9	0.8	0.7	0.9	6.7

Рис. 2. Функції належності нечітких множин

Представимо в матричному вигляді початкові дані функцій належності нечіткого відношення:

$$R = \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} \begin{bmatrix} y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 & y_9 & y_{10} \\ 0,9 & 0,9 & 0,8 & 0,3 & 0,4 & 0,9 & 0,8 & 0,9 & 0,9 & 0,9 \\ 0,3 & 0,3 & 0,6 & 0,5 & 0,3 & 0,3 & 0,2 & 0,2 & 0,5 & 0,6 \\ 0,4 & 0,4 & 0,4 & 0,5 & 0,6 & 0,3 & 0,6 & 0,6 & 0,6 & 0,5 \\ 0,4 & 0,4 & 0,4 & 0,5 & 0,6 & 0,3 & 0,6 & 0,6 & 0,6 & 0,5 \\ 0,5 & 0,5 & 0,6 & 0,6 & 0,7 & 0,5 & 0,9 & 0,8 & 0,7 & 0,9 \end{bmatrix}$$

Матриця S відображає відносний ступінь важливості психофізіологічних характеристик y_1, y_2, \dots, y_p у виконавців z_1, z_2, \dots, z_m під час розподілу множини задач X .

$$S = \begin{matrix} & z_1 & z_2 & z_3 & z_4 & z_5 \\ y_1 & [0,4 & 0,2 & 0,3 & 0,6 & 0,5] \\ y_2 & [0,2 & 0,4 & 0,6 & 0,5 & 0,5] \\ y_3 & [0,5 & 0,5 & 0,3 & 0,7 & 0,5] \\ y_4 & [0,5 & 0,6 & 0,4 & 0,9 & 0,4] \\ y_5 & [0,4 & 0,3 & 0,5 & 0,8 & 0,6] \\ y_6 & [0,8 & 0,4 & 0,5 & 0,6 & 0,4] \\ y_7 & [0,3 & 0,7 & 0,5 & 0,6 & 0,6] \\ y_8 & [0,5 & 0,3 & 0,7 & 0,4 & 0,6] \\ y_9 & [0,6 & 0,4 & 0,4 & 0,8 & 0,6] \\ y_{10} & [0,3 & 0,5 & 0,6 & 0,9 & 0,8] \end{matrix}.$$

Результат обчислення матриці T за формулою (5) наведено на рис. 3.

Петров	Іванов	Сидоров	Швецов	Клусович
3.05	3.08	3.48	4.77	4.1
1.91	1.63	1.63	2.7	2
2.24	2.17	2.54	3.49	2.88
1.83	1.81	1.84	2.76	2.18
2.03	1.97	2.28	3.06	2.54

Рис. 3. Розрахунок матриці T

Фактично, у формулі (6) в чисельнику стоїть число, яке вийшло б у процесі знаходження додатку матриць R та S , а в знаменнику – сума елементів відповідного рядка матриці R .

Далі визначається матриця попарних переваг W за формулою (рис. 4)

$$W = \begin{bmatrix} \min\{t(x_1, z_1), t(x_1, z_2)\} & \dots & \min\{t(x_1, z_m), t(x_n, z_1)\} \\ \dots & \dots & \dots \\ \min\{t(x_n, z_1), t(x_n, z_2)\} & \dots & \min\{t(x_n, z_m), t(x_n, z_1)\} \end{bmatrix}.$$

Петров	Іванов	Сидоров	Швецов	Клусович
0.3961	0.4	0.4519	0.6195	0.5325
0.5026	0.4289	0.4289	0.7105	0.5263
0.4571	0.4429	0.5184	0.7122	0.5878
0.3735	0.3694	0.3755	0.5633	0.4449
0.303	0.294	0.3403	0.4567	0.3791

Рис. 4. Матриця попарних переваг W

$$W = \begin{bmatrix} 0.396 & 0.400 & 0.452 & 0.532 & 0.396 \\ 0.429 & 0.429 & 0.429 & 0.526 & 0.503 \\ 0.443 & 0.443 & 0.518 & 0.588 & 0.457 \\ 0.356 & 0.369 & 0.376 & 0.445 & 0.373 \\ 0.294 & 0.294 & 0.340 & 0.340 & 0.303 \end{bmatrix}.$$

У кожному стовпці матриці, отриманої на попередньому кроці, знаходимо максимальний елемент

$$w_{\max} = (0.443 \quad 0.443 \quad 0.518 \quad 0.588 \quad 0.503).$$

З них вибирається мінімальна величина $\omega = 0.443$.

У матриці T знаходиться найбільший елемент строго менший, ніж ω , який і є порогом розподілу задач $\lambda = 0,443$.

Далі для кожного виконавця z_k визначаємо рівневу множину, тобто множину задач, ступінь прийнятності яких для виконавців перевищує або дорівнює порогу розподілу робіт λ (рис. 5):

Петров	Іванов	Сидоров	Швецов	Клусович
0.3961	0.4	0.4519	0.6195	0.5325
0.5026	0.4289	0.4289	0.7105	0.5263
0.4571	0.4429	0.5184	0.7122	0.5878
0.3735	0.3694	0.3755	0.5633	0.4449
0.303	0.294	0.3403	0.4567	0.3791

Рис. 5. Рекомендації щодо розподілу виконавців на задачі

$$Z_1 = \{x_2, x_3\}, Z_2 = \{x_3\}, Z_3 = \{x_1, x_3\}, Z_4 = \{x_1, x_2, x_3, x_4, x_5\}, Z_5 = \{x_1, x_2, x_3, x_4\}.$$

На першому кроці обираємо виконавця, який може виконувати тільки одну задачу, це Іванов. А задачу x_3 викреслюємо в усіх виконавців. Далі обираємо виконавця, який може реалізувати тільки одну задачу. Таких виконавців двоє: Петров призначається на другу задачу, а Сидоров – на першу. Задачі один і два викреслюються. Знову обирається виконавець, що може реалізувати одну задачу. Це Клусович – четверта задача. Наприкінці залишився один виконавець і одна задача. Виконавець Швецов призначається на п'яту задачу. За розрахунками на посаду SCRUM-майстра рекомендується виконавець Сидоров.

Розглянуто задачу розподілу робіт для декількох виконавців за допомогою використання нечітких бінарних відношень.

Розв'язання задачі допоможе власнику оптимально розподілити їх між виконавцями відповідно до принципу поєднання можливості та бажаності, комплексно врахувати їхні знання та досвід, психофізіологічне профілювання, а також знизити їхню завантаженість.

Висновки

У роботі проаналізовано гнучкі методології управління проектами розробки програмного забезпечення. Сьогодні існує множина варіантів моделей проектів розробки програмного забезпечення. Вибір того чи іншого варіанта залежить від особливостей і вимог проекту. Розглянуто склад SCRUM-команди й основні вимоги до SCRUM-майстра.

Розроблена нечітка модель вибору SCRUM-майстра для управління проектом розробки програмного забезпечення. Наведено приклад розподілу робіт між виконавцями із застосуванням теорії нечітких множин. Розрахунки виконувались у Microsoft Excel.

Подальші дослідження будуть спрямовані на практичну реалізацію моделі у вигляді програмно-методичного комплексу.

Література

1. Ken Schwaber, Jeff Sutherland. The Scrum Guide. The definitive Guide to Scrum: The Rules of the Game. URL: <http://www.scrumguides.org/docs/scrumguide/v1/scrumguideus.pdf> (дата звернення: 10.06.2022).
2. Jeff Sutherland, Nafis Ahmad. How a Traditional Project Manager Transforms to Scrum: PMBOK vs. Scrum. URL: <https://www.scruminc.com/wpcontent/uploads/2014/05/PMBOKvs.ScrumAgile2011.pdf>.

3. Henrik Kniberg Scrum and XP from the Trenches - 2nd Edition. Lulu.com, 2015. p. 184.
4. Fowler M., Highsmith J. The Agile Manifesto. *Software Development*. 2001. Vol. 9. No. 8. P. 28–32.
5. Rising L., Janoff N. S. The Scrum Software Development Process for Small Teams. *IEEE Software*. 2000. Vol. 17. No. 4. P. 26–32.
6. Diebold P., Ostberg J.-P., Wagner S., Zandler U. What Do Practitioners Vary in Using Scrum? // International Conference on Agile Software Development. Springer, Cham, 2015. P. 40–51.
7. Bass J. M. Scrum Master Activities: Process Tailoring in Large Enterprise Projects // Global Software Engineering (ICGSE). IEEE. 9th International Conference. 2014. P. 6–15.
8. Чорний А. В. Роль Скрам-майстра в розвитку лідерських компетентностей персоналу ІТ-підприємств // Бізнес Інформ. 2019. № 1. С. 383–395.
9. Серая О. В. Минимаксная проблема назначения // Восточно-Европейский журнал передовых технологий. 2009. Т. 3. № 3(39). С. 8–11.
10. Математические методы решения многокритериальной задачи о назначениях / О. Я. Никонов, О. А. Подоляка, А. Н. Подоляка, Е. В. Скакалина // Вісник Харківського національного автомобільно-дорожнього університету. 2011. № 5. С. 103–112.
11. Філь Н. Ю., Кудирко О. М. Нечітка модель розподілу робіт між працівниками на складі підприємства // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. 2021. Вип. 44. С. 66–75. DOI: <https://doi.org/10.36910/6775-2524-0560-2021-44-11>.

References

1. Ken Schwaber, Jeff Sutherland. The Scrum Guide. The definitive Guide to Scrum: The Rules of the Game. URL: <http://www.scrumguides.org/docs/scrumguide/v1/scrumguideus.pdf> (дата звернення: 10.06.2022).
2. Jeff Sutherland, Nafis Ahmad. How a Traditional Project Manager Transforms to Scrum: PMBOK vs. Scrum. URL: <https://www.scruminc.com/wpcontent/uploads/2014/05/PMBOKvs.ScrumAgile2011.pdf>.
3. Henrik Kniberg Scrum and XP from the Trenches - 2nd Edition. Lulu.com, 2015. p. 184.
4. Fowler M., Highsmith J. The Agile Manifesto. *Software Development*. 2001. Vol. 9. No. 8. P. 28–32.
5. Rising L., Janoff N. S. The Scrum Software Development Process for Small Teams. *IEEE Software*. 2000. Vol. 17. No. 4. P. 26–32.
6. Diebold P., Ostberg J.-P., Wagner S., Zandler U. What Do Practitioners Vary in Using Scrum? // International Conference on Agile Software Development. Springer, Cham, 2015. P. 40–51.
7. Bass J. M. Scrum Master Activities: Process Tailoring in Large Enterprise Projects // Global

- Software Engineering (ICGSE). IEEE. 9th International Conference. 2014. P. 6–15.
8. Chornyi, Anton V. (2019). The Role of Scrum Master in the Development of Leader Competencies of the Staff of IT-Enterprises. *Business Inform* 1:383–395.
 9. Sira O.V. Minimaksnaia problema naznacheniiia [Minimax assignment problem]. *Eastern-European Journal of Enterprise Technologies*, vol. 3, no. 3(39), June 2009, pp. 8–11, doi: 10.15587/1729-4061.2009.20274.
 10. Nikonov, O. Podolyaka, O., Podolyaka, A., Skakalina, E. [Mathematical methods of solving multicriterion assignment problem]. *Visnyk Harkivs'kogo nacional'nogo avtomobil'no-dorozhn'ogo universytetu: zb. nauk. pr.* 2011, no. 5, pp. 103–112.
 11. Fil, N. & Kudyrko, O. (2021). Nechitka model rozpodilu robit mizh pratsivnykamy na skladi pidpriemstva. [A fuzzy model for distributing work between workers in the warehouse of an enterprise]. *Computer-integrated technologies: education, science, production.* 66–75. doi: 10.36910/6775-2524-0560-2021-44-11.

Нефьодов Леонід Іванович¹, д.т.н., проф., завідувач кафедри автоматизації та комп'ютерно-інтегрованих технологій, nefedovli@i.ua.

Філь Наталія Юрїївна¹, к.т.н., доц. каф. автоматизації та комп'ютерно-інтегрованих технологій, fil_nu@i.ua.

¹Харківський національний автомобільно-дорожній університет, 61002, Україна, м. Харків, вул. Ярослава Мудрого, 25.

The model of SCRUM master selection using fuzzy sets

Abstract. Problem. *The complexity of software is constantly growing. Today a user is interested in software that is integrated into all aspects of their life. Creating a large number of diverse software applications is impossible without the formation of a software development team. Agile methodologies are used to develop most modern programs. The most common is the SCRUM methodology. Team working*

is managed by a SCRUM master, which is the interface between the owner and the project team. Therefore, choosing an effective SCRUM master is an urgent task. Aim The work aims to increase the efficiency of software development projects by developing the SCRUM master selection model using fuzzy sets, which will consider the individual characteristics and advantages of the SCRUM project team members. **Methodology.** The methods and principles of fuzzy mathematics are used. Decision-making methods in the conditions of fuzzy input data were used to develop a model for choosing a SCRUM master for a software development project. **Results.** Today, there are many variants of models for software development projects. The choice of one or another option depends on the features and requirements of the project. A fuzzy model for choosing a SCRUM master for managing a software development project has been developed. An example of work distribution between performers using the theory of fuzzy sets is given. Calculations were made in Microsoft Excel. **Originality.** The developed model allows choosing a SCRUM master in the conditions of fuzzy information about the individual preferences of the areas of the software development project team. **Practical value.** Using the proposed SCRUM master selection model for a software development project will improve project team communication and reduce the development time. In addition, the proposed model can be used to select a SCRUM master for other projects

Key words: software, agile methodology, fuzzy sets, SCRUM master, membership function.

Nefedov Leonid¹, professor, Doct. of Science, Department of Automation and Computer-Integrated Technologies, nefedovli@i.ua, tel. +38 095-885-33-89,

Fil Nataliya¹, Ph.D., Assoc. Prof. Department of Automation and Computer-Integrated Technologies, fil_nu@i.ua,

¹Kharkov National Automobile and Highway University, 25, Yaroslava Mudrogo str., Kharkiv, 61002, Ukraine.