

УДК 004.415.2

## ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ УПРАВЛІННЯ ЗАДАЧАМИ ТА ОЦІНКИ ЕФЕКТИВНОСТІ КОМАНДИ РОЗРОБНИКІВ

Васильченко В.С., студент

*Київський національний університет технологій та дизайну*

Новак Д.С., кандидат технічних наук, доцент

*Київський національний університет технологій та дизайну*

*Ключові слова:* управління задачами, ефективність команди, Agile, KPI, інформаційна система.

В умовах стрімкого розвитку індустрії програмного забезпечення ефективно управління командою розробників є одним із ключових чинників успіху ІТ-компаній. Сучасні проекти відрізняються значною складністю, залученням великої кількості учасників та жорсткими часовими обмеженнями, що зумовлює необхідність застосування спеціалізованих інструментів для координації роботи та відстеження прогресу. Незважаючи на наявність на ринку відомих продуктів для управління проектами – Jira, Trello, Asana – вони здебільшого є універсальними рішеннями, що не враховують специфіки роботи команд розробників. Зокрема, вони не забезпечують глибокого аналізу індивідуальної продуктивності, не інтегруються з системами контролю версій для автоматичного збору метрик та не надають гнучких механізмів оцінювання ефективності. Управління задачами у сфері розробки програмного забезпечення нерозривно пов'язане з методологіями гнучкої розробки – Agile, Scrum, Kanban. Зазначені підходи передбачають ітеративний процес виконання робіт, де кожна задача має чіткий опис, виконавця, пріоритет та терміни виконання. Для оцінки ефективності команди розробників використовуються ключові показники ефективності (KPI – Key Performance Indicators). До основних метрик відносяться: кількість виконаних задач за спринт, час виконання задач відносно планової оцінки, відсоток задач, виконаних у встановлені строки, якість коду (кількість виявлених дефектів), а також активність у системі контролю версій.

Проведений порівняльний аналіз існуючих систем управління задачами показав, що жодна з них не задовольняє повною мірою потреби команд розробників у контексті інтегрованої оцінки ефективності. Це підтверджує актуальність розробки спеціалізованої системи. Розроблена інформаційна система побудована за клієнт-серверною архітектурою з використанням мікросервісного підходу. Серверна частина реалізована на платформі Node.js з фреймворком Express.js, що забезпечує високу продуктивність при обробці запитів. Клієнтська частина розроблена з використанням бібліотеки React.js, яка дозволяє створювати динамічний та відзивчивий інтерфейс користувача. Для зберігання даних використовується реляційна СУБД PostgreSQL у поєднанні з Redis для кешування часто запитуваних даних. Система охоплює такі основні функціональні модулі: модуль управління задачами – створення,

редагування, призначення та відстеження статусу задач, підтримка спринтів та backlog-у, система пріоритетів і тегів; модуль аналітики та звітності – автоматичний розрахунок KPI для кожного члена команди, побудова графіків продуктивності, формування звітів за обраний період; модуль інтеграції з Git – автоматичний збір статистики комітів, pull-реквестів та code review, зв'язування задач з конкретними змінами в коді; модуль сповіщень – автоматичні нагадування про дедлайни та сповіщення про зміну статусу задач з інтеграцією в месенджери. Для комплексної оцінки ефективності кожного розробника запропонована багатокритеріальна модель, що враховує кількісні та якісні показники. Інтегральний показник ефективності  $E$  розраховується за формулою:

$$E = w_1 \cdot K_1 + w_2 \cdot K_2 + w_3 \cdot K_3 + w_4 \cdot K_4,$$

де  $K_1$  – відсоток задач, виконаних вчасно;  $K_2$  – відношення фактичного часу виконання до оціночного;  $K_3$  – показник якості коду (обернено пропорційний кількості дефектів);  $K_4$  – показник активності в системі контролю версій;  $w_1, w_2, w_3, w_4$  – вагові коефіцієнти, що визначаються керівником команди залежно від специфіки проєкту (сума вагових коефіцієнтів дорівнює 1). Вагові коефіцієнти є налаштовуваними параметрами системи, що дозволяє адаптувати модель оцінки до конкретних умов та пріоритетів проєкту. Наприклад, для проєктів із жорсткими строками доцільно збільшити вагу показника  $K_1$ , тоді як для проєктів з підвищеними вимогами до якості коду –  $K_3$ .

В результаті проведеної роботи спроектовано та реалізовано інформаційну систему для управління задачами та оцінки ефективності команди розробників. Система успішно пройшла тестування в умовах реального проєкту.

Розроблена система відрізняється від наявних аналогів завдяки інтеграції з системами контролю версій для автоматичного збору метрик, гнучкій налаштовуваній моделі оцінки ефективності, а також зручному інтерфейсу, орієнтованому на специфіку роботи IT-команд.

Перспективами подальшого розвитку системи є впровадження алгоритмів машинного навчання для прогнозування термінів виконання задач та автоматичне визначення вагових коефіцієнтів моделі оцінки ефективності на основі накопиченої статистики.

#### Список використаних джерел

1. Schwaber K. The Scrum Guide / K. Schwaber, J. Sutherland. – Scrum.org, 2020. – 13 p.
2. Cohn M. Agile Estimating and Planning / M. Cohn. – New Jersey : Prentice Hall, 2005. – 368 p.
3. Fowler M. Patterns of Enterprise Application Architecture / M. Fowler. – Boston : Addison-Wesley, 2002. – 533 p.
4. Kim G. The DevOps Handbook / G. Kim, J. Humble, P. Debois, J. Willis. – Portland : IT Revolution Press, 2016. – 480 p.
5. Документація React.js [Електронний ресурс]. – Режим доступу: <https://react.dev> (дата звернення: 20.04.2026).