Яворський Назар, Лукащук Богдан*
[1] НЛТУ України, Львів, Україна, ORCID: 0009-0007-1426-9413,
n.yavorskyy@nltu.lviv.ua
[2] НЛТУ України, Львів, Україна. ORCID: 0009-0002-2365-4923,
bohdan.lukashchuk@nltu.edu.ua

Розроблення модуля інтелектуального аналізу та оптимізації навантаження на команду для системи управління стартап-командою

Анотація. *У роботі представлено підхід до розроблення модуля аналізу та оптимізації навантаження команди для підвищення ефективності управління IT-проєктами. Система автоматично збирає, обробляє та аналізує дані про продуктивність учасників, виявляє перевантаження й формує рекомендації щодо оптимального розподілу завдань. Архітектура побудована з використанням.NET (C#), Angular, RabbitMQ, Nginx і Docker, а математичні моделі забезпечують кількісну оцінку навантаження та ефективності команди. Отримані результати можуть бути використані для створення інтелектуальних систем підтримки управлінських рішень.*

Ключові слова *– оптимізація навантаження, аналіз даних, командна продуктивність,.NET, Angular, RabbitMQ, Docker, управління проєктами.*

Вступ. У сучасних умовах цифрової економіки ефективність командної роботи є ключовим чинником успішної реалізації IT-проєктів. Зростання кількості завдань, складність їх взаємозв’язків та необхідність раціонального використання людських ресурсів зумовлюють потребу в системах, здатних автоматично аналізувати навантаження учасників команди та пропонувати рекомендації щодо його оптимізації.

Актуальність теми «Розроблення модуля інтелектуального аналізу та оптимізації навантаження на команду для системи управління стартап-командою» полягає у підвищенні продуктивності праці та якості управлінських рішень шляхом використання сучасних підходів до аналітики даних. Автоматизація процесів оцінки завантаженості, планування завдань і контролю ефективності дозволяє уникнути перевантажень окремих співробітників, своєчасно виявляти

* науковий керівник

дисбаланс у роботі команди та забезпечувати стабільне виконання проєктів у межах визначених термінів.

Математична модель. Для реалізації аналітичної складової модуля оптимізації навантаження команди використано систему математичних моделей, що забезпечують кількісну оцінку рівня завантаженості кожного учасника, визначення ступеня дисбалансу у розподілі завдань та формування рекомендацій для його усунення. Основна концепція полягає у створенні функції ефективності команди, яка враховує індивідуальні показники продуктивності, кількість поточних завдань та їхню складність.

Для узгодження параметрів, що мають різну природу (кількість задач, витрачений час, пріоритетність тощо), застосовується z-нормалізація, яка дозволяє привести всі дані до єдиної шкали вимірювання. Це забезпечує коректне порівняння між показниками різних учасників.

$$Z_i = \frac{x_i - \mu}{\sigma} \tag{1}$$

де:

- x_i — фактичне значення параметра,
- μ — середнє значення показника для всієї команди,
- σ — стандартне відхилення.

Рівень завантаженості кожного члена команди визначається як зважена сума нормалізованих параметрів, що відображає вплив кожного з них на загальний показник ефективності.

$$L_i = \sum_{j=1}^{n} w_j \cdot p_{ij} \tag{2}$$

де:

- L_i — загальне навантаження учасника i,
- w_j — ваговий коефіцієнт параметра j,
- p_{ij} — нормалізоване значення параметра j для учасника i.

Для оцінки рівномірності розподілу завдань у команді використовується індекс балансу команди (B).

* науковий керівник

$$B = 1 - \frac{\sigma_L}{\mu_L} \tag{3}$$

де:

- σ_L — стандартне відхилення навантаження між членами команди,
- μ_L — середнє навантаження команди.

Якщо значення BBB наближається до 1, це свідчить про рівномірне навантаження між учасниками. Якщо ж показник падає нижче 0,6 – спостерігається істотний дисбаланс, який потребує коригування.

Для аналізу впливу рівня завантаженості на продуктивність роботи використовується лінійна регресійна модель, що дозволяє оцінити залежність між трудовим навантаженням і результативністю виконання завдань.

$$\hat{y}_i = \beta_0 + \beta_1 L_i + \varepsilon_i \tag{4}$$

де:

- $\hat{y}_i$ — прогнозований рівень продуктивності,
- L_i — поточне навантаження,
- β_0, β_1 — параметри моделі,
- ε_i — випадкова похибка.

Основна оптимізаційна задача полягає у мінімізації дисперсії навантаження між усіма учасниками команди, тобто у зменшенні відхилення індивідуальних показників від середнього значення. Формально це дозволяє забезпечити найбільш збалансований розподіл ресурсів у межах команди.

$$\min \sum_{i=1}^{m} (L_i - \mu_L)^2 \tag{5}$$

де m — кількість учасників команди.

Для оцінки якості та точності роботи аналітичної системи застосовуються такі метрики:

- MAPE (Mean Absolute Percentage Error) – середня відносна похибка прогнозу;

* науковий керівник

- R^2 (коефіцієнт детермінації) – показує ступінь відповідності моделі фактичним даним;
- Індекс балансу B – характеризує рівномірність розподілу навантаження між учасниками.

Використання зазначених математичних моделей дозволяє системі не лише аналізувати поточний стан навантаження команди, а й прогнозувати можливі перевантаження, виявляти неефективне використання ресурсів і формувати обґрунтовані рекомендації для підвищення ефективності командної роботи.

Програмна реалізація. Розроблення модуля здійснювалося у середовищі Microsoft Visual Studio, яке забезпечує зручну роботу з C#, інтеграцію з Git, підтримку Docker і фреймворку.NET 8 / ASP.NET Core [1]. Це дало змогу швидко реалізувати серверну логіку, провести налагодження та тестування API.

Архітектура системи побудована за мікросервісним принципом. Серверна частина реалізована у вигляді ASP.NET Core Web API, що взаємодіє з клієнтською частиною через REST API [2]. Основні компоненти:

- Controllers – обробка запитів;
- Services – бізнес-логіка аналізу;
- Data Layer (EF Core) – робота з базою даних;
- RabbitMQ – обмін повідомленнями між сервісами.

Для балансування використано Nginx, а клієнтський інтерфейс створено з використанням Angular, який забезпечує інтерактивну візуалізацію аналітичних даних у реальному часі. Основні модулі:

- Dashboard – стан навантаження команди;
- Task Management – управління завданнями;
- Analytics – показники ефективності;
- Optimization – рекомендації системи.

Комунікація між клієнтом і сервером реалізована через HTTP та WebSocket, що забезпечує оновлення даних без перезавантаження сторінки.

Система розгортається у контейнерах Docker, описаних у *docker-compose.yml*. Запуск здійснюється командою *docker-compose up -d*

* науковий керівник

RabbitMQ [3] забезпечує асинхронну обробку повідомлень, а логування реалізовано за допомогою Serilog з подальшим аналізом через ELK Stack. Для тестування API застосовувались Postman та Swagger UI.

Модуль автоматично збирає дані про виконання завдань, розраховує індекс навантаження, визначає дисбаланс і формує рекомендації для оптимізації роботи команди (рис.1).

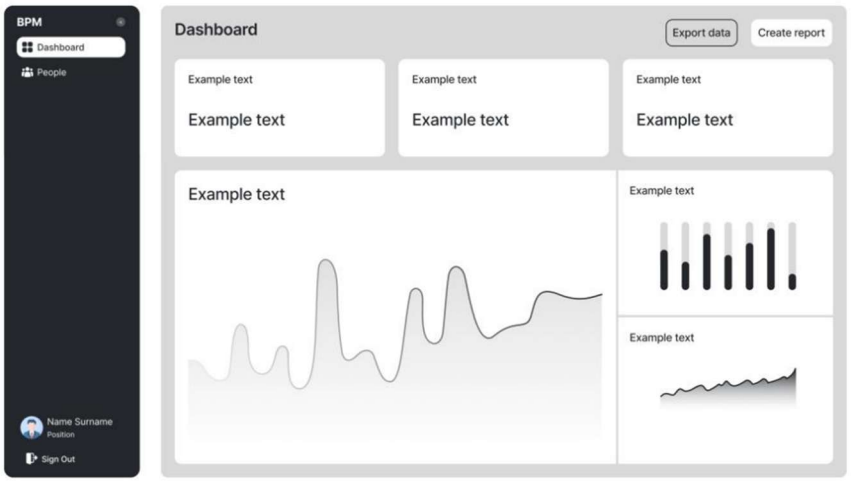

Рисунок 1. Сторінка Dashboard

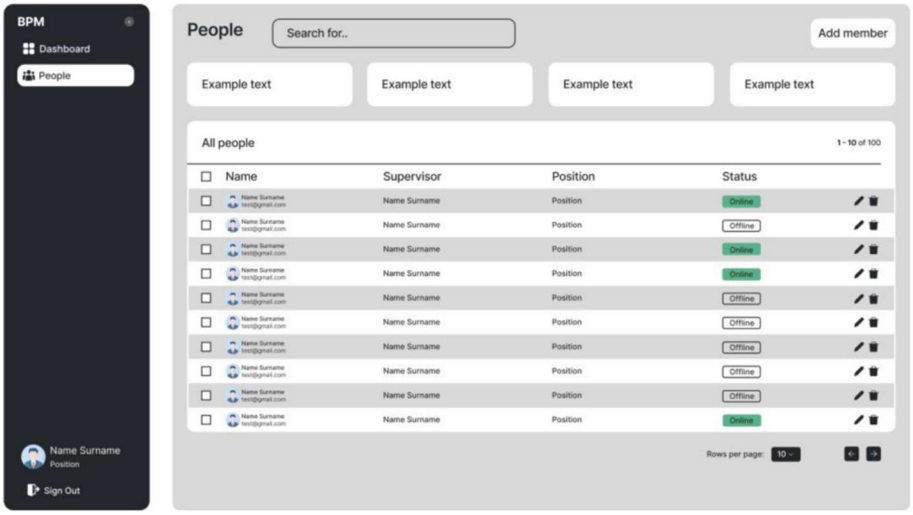

Рисунок 2. Сторінка People

Розроблена система дозволяє в реальному часі аналізувати, прогнозувати та оптимізувати навантаження команди (рис.2), легко інтегрується у корпоративну інфраструктуру й масштабуються для більших колективів.

Висновок. Було розроблено систему для аналізу та оптимізації навантаження команди з використанням сучасних методів аналітики даних і прогнозування продуктивності. Реалізоване рішення автоматично оцінює завантаженість

* науковий керівник

учасників, виявляє дисбаланс у розподілі завдань і формує рекомендації для його усунення. Робота над проєктом дозволила поглибити знання у сфері розробки вебсервісів і здобути досвід створення аналітичних систем підтримки управлінських рішень.

Список використаних літературних джерел

[1] A. Lock, ASP.NET Core in Action, Third Edition, Shelter Island, NY: Manning Publications, Jul. 2023. [Online]. https://www.manning.com/books/asp-net-core-in-action-third-edition

[2] M. Masse, REST API Design Rulebook, Sebastopol, CA: O'Reilly Media, 2011. [Online]. https://pepa.holla.cz/wp-content/uploads/2016/01/REST-API-Design-Rulebook.pdf

[3] G. M. Roy, RabbitMQ in Depth, Shelter Island, NY: Manning Publications, Sep. 2017. [Online]. https://www.manning.com/books/rabbitmq-in-depth

Development of an Intelligent Module for Team Workload Analysis and Optimization in a Startup Management System

Yavorskyi Nazar, Lukashchuk Bohdan

The paper presents an approach to developing a module for analyzing and optimizing team workload to improve the efficiency of IT project management. The system automatically collects, processes, and analyzes data on participant performance, detects overload, and generates recommendations for optimal task distribution. The architecture is built using.NET (C#), Angular, RabbitMQ, Nginx, and Docker, and mathematical models provide a quantitative assessment of team workload and efficiency. The results obtained can be used to create intelligent systems to support management decisions.

* науковий керівник