

Концевий Владислав Вячеславович

Аспірант кафедри управління проєктами,

<https://orcid.org/0000-0002-6636-5489>

Київський національний університет будівництва і архітектури, Київ

Войтенко Олександр Степанович

Доцент кафедри управління проєктами,

<https://orcid.org/0000-0002-7414-7836>

Київський національний університет будівництва і архітектури, Київ

**ОЦІНКА ВПЛИВУ УЧАСНИКІВ ПРОЄКТНИХ КОМАНД
НА СИСТЕМУ КОМУНІКАЦІЙ**

Анотація. У статті представлено комплексний підхід до оцінки впливу учасників проєктних команд на систему комунікацій у цифровому середовищі. Актуальність дослідження зумовлена зростанням ролі віртуальних і гібридних команд в управлінні проєктами, де ефективність комунікацій значною мірою залежить не лише від технічних засобів, а й від поведінкових характеристик їх учасників. Визначено, що окремі особи можуть виступати як деструктивні чинники, що впливають на ефективність командної взаємодії та інформаційні потоки. Для виявлення та кількісної оцінки такого впливу використано модель FIRO-B, яка дозволяє визначити міжособистісну орієнтацію кожного члена команди за трьома ключовими вимірами: включення, контроль і прихильність. Аналіз за вимірами було поєднано з оцінкою за вісьмома сферами деструктивного впливу: технологічні бар'єри, опір змінам, негатив, мікроменеджмент, дезінформація, відсутність співпраці, домінування у комунікаціях та часові порушення. Розроблено формули для обчислення індивідуального показника поточного впливу та відносного впливу на систему комунікацій, з урахуванням зворотної залежності від рівня лідерських навичок. Практичне втілення реалізовано у вигляді веб-додатку на основі фреймворку Flask. Розкрито архітектуру програмного рішення, яка включає серверну логіку, інтерфейс користувача, базу даних, модуль авторизації, а також систему візуалізації результатів. Додаток дозволяє вводити індивідуальні показники членів команди, обчислювати рівень впливу на комунікаційне середовище та відображати результати у вигляді таблиць і графіків. Для обробки та аналізу даних реалізовано інтеграцію з базами даних SQLite/PostgreSQL, а також передбачено можливість експорту результатів і автоматизації збору вхідної інформації за допомогою Microsoft Forms та Power Automate. Розроблений інструмент дозволяє керівникам проєктів своєчасно виявляти ризики щодо поведінки членів команди, оцінювати ефективність комунікаційної структури та приймати обґрунтовані управлінські рішення щодо складу команди. Практичне значення полягає в інтеграції поведінкової аналітики в систему управління проєктами, що відкриває перспективи для подальшого розвитку через використання технологій нейронних мереж та машинного навчання.

Ключові слова: віртуальні команди; IT організація; метод FIRO-B; SQL; Python

Постановка проблеми

Управління проєктами в умовах цифрової трансформації вимагає нових підходів до оцінки якості командної взаємодії, зокрема – ефективності комунікацій. Наявність у команді учасників, які деструктивно впливають на інформаційні потоки, здатна значно знижувати продуктивність навіть за наявності належної технічної організації. У зв'язку з цим постає завдання розробки інструментів, які дозволять своєчасно ідентифікувати вплив окремих учасників на систему комунікацій, оцінюючи як міжособистісні характеристики, так і фактори

поведінкового впливу. Одним із ключових напрямів автоматизації в управлінні проєктами є розробка цифрових рішень для аналізу командної взаємодії. У даному дослідженні розглянуто підхід до створення веб-додатку на мові Python, призначеного для обчислення поточного впливу учасників команди на систему комунікацій на основі моделі FIRO-B та узагальнених сфер деструктивного впливу.

Мета дослідження полягає в розробці концепції та створенні прототипу веб-додатку на базі Python, який дозволяє кількісно оцінювати поточний вплив учасників проєктної команди на систему комунікацій. Інструмент базується на моделі FIRO-B

та узагальнених сферах деструктивного впливу, що дає змогу своєчасно ідентифікувати ризики порушення командної взаємодії та підтримувати ефективність комунікацій в умовах цифрової трансформації управління проектами.

Аналіз досліджень і публікацій

Дослідження [1] спрямоване на розробку системи управління базами даних коледжу, яку можна використовувати в навчальному закладі. Для реалізації цієї програми використовується фреймворк Python Flask, доступ до якої можливий по всьому закладу або з будь-якого конкретного відділу. Система пропонує різні функції для студентів та співробітників, включаючи відвідуваність та оцінки студентів, до яких можуть отримати доступ як студенти, так і співробітники, але які можуть оновлювати лише співробітники певного відділу. Для цього користувачі повинні бути зареєстрованими в системі, після чого вони зможуть отримати доступ до своїх облікових записів, а також змінювати дані відповідно до наданих їм прав. Python Flask був використаний для даної розробки. В якості фронтенду використовувався HTML, CSS та JavaScript, для бекенду – Python Flask та MySQL.

У статті [2] наведено порівняльний аналіз веб-фреймворків Flask та Django на основі мови програмування Python. Зазначено, що через високий попит на веб-додатки розробникам необхідно створювати рішення, які є економічно ефективними, безпечними та якісно реалізованими. Особливу увагу приділено фреймворку Django, який працює на мові Python і використовує архітектуру MVC. Django підходить для проектів із обмеженим часом і невеликим бюджетом. Авторами даного дослідження було додано FastAPI до аналізу, проведеного в [2], за тими ж критеріями. Отже, порівняльний час

розробки Flask, Django та FastAPI наведені в таблиці. Якщо необхідно створити додаток з базовою маршрутизацією URL-адрес та шаблонами, в [2] рекомендовано використовувати фреймворк Flask. Водночас підкреслюється, що в результатах роботи Flask-додатків HTML-файли містять більше додаткових тегів та скриптів, а також зберігають більше інформації про поля форм, що призводить до збільшення розміру файлів порівняно з Django.

Серед переваг статті можна відзначити чітку структуру подачі матеріалу, акцент на практичній корисності для розробників та наявність конкретних рекомендацій щодо вибору між Django та Flask. До недоліків можна віднести те, що стаття більше зосереджується лише на двох фреймворках, залишаючи поза увагою інші сучасні альтернативи, а також не надає детального порівняння за додатковими критеріями, такими як масштабованість, продуктивність або підтримка спільноти.

У статті [3] проведено аналіз впливу різних серверів баз даних на основі реляційної моделі – SQL Server, MySQL та PostgreSQL на продуктивність веб-застосунків, створених за допомогою фреймворку Laravel на мові програмування PHP. Дослідження зосереджується на вимірюванні часу виконання запитів різної складності, включаючи прості запити та запити з конкатенацією стовпців і таблиць. Виявлено, що продуктивність обробки даних змінюється залежно від характеру операцій, навіть при однаковій структурі даних. Особливу увагу звернено на ефективність роботи з невеликими обсягами даних, де кількість записів не перевищує тисячу, а сервер має мінімальні або середні технічні характеристики. У такому випадку MySQL демонструє кращі результати, забезпечуючи швидкий час обробки запитів і високу стабільність.

Таблиця – Час, витрачений на розробку програми

Дія	Написання (хв)			Виправлення (хв)			Загалом (хв)		
	Flask	Django	FastAPI	Flask	Django	FastAPI	Flask	Django	FastAPI
Вхід	80	50	70	0	30	10	80	80	80
Реєстрація	105	40	85	0	0	5	105	40	90
Активність	60	30	50	30	0	10	90	30	60
Додавання записів	25	30	20	5	0	0	30	30	20
Видалення записів	25	30	20	5	0	0	30	30	20
Оновлення записів	25	30	20	5	0	0	30	30	20
Профіль користувача	60	60	50	0	0	15	60	60	55
Видалення профілю	10	5	8	0	0	0	10	5	8
Оновлення профілю	10	5	8	0	0	0	10	5	8
Додавання файлів	25	30	20	5	0	0	30	30	20
Видалення файлів	25	30	20	5	0	0	30	30	20
Оновлення файлів	25	30	20	5	0	0	30	30	20
Перевірка даних	60	20	50	10	0	5	70	20	55
Панель адміністратора	80	50	65	0	0	5	80	50	70
Відновлення паролю	10	10	8	0	0	0	10	10	8
Разом	630	375	564	80	60	50	710	435	614

Джерело: [2] та доповнені дані щодо FastAPI

Недоліком дослідження є обмеженість розгляду лише на невеликих обсягах даних, без тестування продуктивності баз даних при значно більших навантаженнях або в умовах масштабованих систем. Крім того, вплив інших аспектів роботи баз даних, таких як індексація або паралельна обробка запитів, не було детально проаналізовано.

Предметом дослідження [4] є розробка веб-фреймворку з вбудованими функціями безпеки, які запобігатимуть поширеним вразливостям безпеки в електронних медичних записах. Були розглянуті та порівняні функції безпеки трьох PHP-фреймворків: Laravel, CodeIgniter та Symfony. На основі результатів Laravel оснащений функціями безпеки, які наразі потрібні для електронних медичних записів. Також, наведений опис запропонованої концептуальної структури, яку можна адаптувати для впровадження безпечних систем електронних медичних записів.

У статтях [5 – 6] досліджується порівняльна продуктивність двох JavaScript-фреймворків – Angular та Vue.js – у сфері розробки ігор. Основними критеріями оцінювання обрано: час обміну даними з сервером; швидкість рендерингу компонентів програми; обсяг споживаної пам'яті при оновленні ігрової інформації та при відновленні користувача в грі; ступінь завантаження браузера, а також розмір кінцевих файлів застосунку. На основі проведеного тестування встановлено, що Vue.js показує вищу ефективність за більшістю параметрів.

Серед переваг цієї роботи можна виділити чітке визначення критеріїв порівняння, що забезпечує об'єктивність дослідження, а також практичну спрямованість тестування саме на розробку ігор, що є актуальним напрямком у сучасній веб-розробці. Крім того, результати надають розробникам чіткі орієнтири у виборі фреймворку залежно від потреб проекту. Недоліком є дослідження лише двох фреймворків без залучення інших популярних технологій, що могли б розширити рамки аналізу. Також тестування могло б бути більш репрезентативним, якби проводилося на різних типах ігрових застосунків з різною складністю логіки та графіки.

У статті [7] наведена характеристика багатосторінкових та односторінкових додатків, пояснюється різниця між веб-застосунками MPA (Multi Page Application) та SPA (Single Page Application). Показано переваги та недоліки MPA порівняно з веб-застосунками SPA. Визначено необхідні функції фреймворку, оптимізованого для створення веб-застосунків MPA та SPA.

Виклад основного матеріалу

Дана робота є продовженням дослідження, присвяченого пошуку та класифікації елементів що впливають на систему комунікації [8]. Для розробки

додатку обґрунтовано використання моделі FIRO-B, яка дозволяє оцінити міжособистісну орієнтацію учасника за трьома вимірами: включення, контроль і прихильність [9 – 10]. Ці дані, у поєднанні з сьомома сферами деструктивного впливу (технологічні бар'єри, опір змінам, негатив, мікроменеджмент, дезінформація, відсутність співпраці, домінування в комунікації, часові порушення) дозволяють розрахувати показник кумулятивного поточного впливу члена команди на комунікації за формулою:

$$PI_j = (1 - L_j) \cdot \sum_{i=1}^8 D_i \quad (1)$$

де $j \in [1; m]$ – член команди проекту, m – кількість членів команди проекту; PI_j – поточний кумулятивний вплив члена команди (за сферами впливу); D_1 – технологічні бар'єри; D_2 – опір змінам; D_3 – негатив; D_4 – мікроменеджмент; D_5 – дезінформація; D_6 – відсутність співпраці; D_7 – домінування в комунікації; D_8 – часові порушення; L_j – показник лідерських навичок члена команди.

При обчисленні поточного кумулятивного впливу члена команди враховується обернена пропорційність лідерських навичок, адже чим менші показники лідерства, тим сильніший негативний вплив особа робить на домен комунікації.

Таким чином, вплив члена команди проекту на систему комунікацій загалом можна визначити шляхом розрахунку співвідношення показників поточного впливу за всіма сферами комунікацій до показників системи комунікацій (цілісність системи комунікацій проекту) за його відсутності:

$$I_j = \frac{PI_j \cdot 100}{WD} \quad (2)$$

де I_j (Influence) – показник впливу члена команди на систему комунікацій, виражений у відсотках; WD – цілісність системи комунікацій проекту за відсутності дисраптора.

Основу функціональності додатку становлять математичні формули обчислення поточного кумулятивного впливу члена команди та показника впливу члена команди на систему комунікацій, які дозволяють визначити рівень поточного впливу кожного учасника з урахуванням його лідерських характеристик та множини факторів, що порушують комунікації. Реалізація обчислень здійснюється у вигляді окремого логічного модуля, який імплементується у веб-середовище через обраний фреймворк.

Для розробки додатку обраний Python-фреймворк Flask, що забезпечує легкість розгортання, гнучкість структури та зручність обробки HTTP-запитів. За потреби у масштабуванні або асинхронній обробці даних можливе використання FastAPI. Інтерфейс користувача реалізується через HTML-шаблони з підтримкою Bootstrap, а логіка обробки даних – через модулі calculator.py та routes.py.

Основною перевагою Flask є його мікрофреймворкова архітектура, яка надає розробнику можливість вибору інструментів, бібліотек і шаблонів. Це корисно для створення додатку, у якому необхідно інтегрувати кілька окремих компонентів – модулі обчислення показників, інтерфейси для введення даних, обробку запитів користувача, збереження результатів та візуалізацію. Flask дозволяє організувати компоненти у вигляді незалежних модулів або, так званих, Blueprints, що полегшує супровід і масштабування додатку.

Flask використовує концепцію маршрутів, де кожен URL відповідає окремій функції (view). Такий підхід забезпечує контроль поведінки сторінок додатку. Наприклад, головна сторінка може відображати форму введення даних, інша – обробляти форму та запускати алгоритм розрахунку PI_j та I_j , інша – відображати результати у вигляді таблиць і графіків. Обробка запитів здійснюється через методи POST і GET, що дозволяє налаштувати логіку збереження, обробки та виведення даних.

Для реалізації інтерфейсу Flask використовує руські шаблони Jinja2, який дозволяє створювати HTML-сторінки з динамічним вмістом. Це дає можливість будувати гнучкі форми, автоматично заповнювати таблиці, відображати повідомлення про помилки або підтвердження, а також інтегрувати інтерактивні елементи – графіки, випадальні списки, фільтри тощо. Стилзація інтерфейсу реалізується шляхом підключення CSS-фреймворків, таких як Bootstrap, що дозволяє створити зручний і адаптивний для користувача веб інтерфейс без значних витрат часу.

Обробка даних у Flask може бути пов'язана з базою даних через ORM (Object Relational Mapping), наприклад, SQLAlchemy. Це забезпечує зручний механізм взаємодії з таблицями користувачів, результатів обчислень, історії сеансів або інших структурованих даних. Якщо проєкт не передбачає складної структури даних, на початковому етапі використовується легкий варіант на основі SQLite, який інтегрується з Flask.

Сумісність Flask із зовнішніми бібліотеками Python дозволяє підключати обчислювальні модулі, наприклад, модулі з розрахунками PI_j та I_j , написані окремо у вигляді функцій або класів, без необхідності інтегрувати їх у основну логіку вручну. Крім того, Flask інтегрується з бібліотеками візуалізації, такими як Plotly або Matplotlib, що дає змогу відображати аналітичні графіки безпосередньо у веб інтерфейсі.

У разі потреби розширення функціональності Flask дозволяє реалізувати RESTful API, що слугує для інтеграції з іншими системами, такими як

корпоративні платформи, мобільні додатки або зовнішні аналітичні модулі. Через Flask API можна забезпечувати обмін даними з Microsoft Forms або Power Automate для автоматизації збору вхідної інформації.

Розгортання Flask-додатку є гнучким: він може працювати локально на сервері розробника, бути розміщеним у контейнері Docker або може бути розміщений на хмарних платформах типу Heroku, Railway, Render або PythonAnywhere. Така універсальність дозволяє адаптувати рішення до наявної IT-інфраструктури будь-якої організації.

Таким чином, Flask є раціональним вибором для розробки додатку, що поєднує психологічну аналітику, поведінкові обчислення та управління командними процесами у проєктному середовищі. Його гнучкість у застосуванні, розширюваність та широка екосистема суміжних бібліотек забезпечують ефективну реалізацію як додатків з мінімальним ключовим функціоналом, так і повноцінних програм для цифрового управління командною динамікою.

Додаток автоматично імпортує дані щодо лідерських показників та рівнів деструкції по восьми сферах (доменах) для кожного члена команди. Після обробки даних результати виводяться у вигляді таблиці з автоматичним розрахунком показників PI_j та I_j . Додатково реалізується візуалізація результатів за допомогою бібліотек Plotly або Chart.js, що підвищує зручність інтерпретації.

Додаток має бути доповнений модулями авторизації користувачів, збереженням історії даних у базі SQLite або PostgreSQL [11–12], а також підтримкою експорту результатів у форматах CSV або Excel. Окремим напрямом розвитку є інтеграція з зовнішніми сервісами, зокрема, отримання вхідних даних із Microsoft Forms через API або Power Automate, що дозволить повністю автоматизувати процес обробки анкет та аналітики.

Модуль авторизації користувачів є важливою складовою веб-додатку, орієнтованого на управління та аналітику поведінкових показників у проєктних командах. Його реалізація дозволяє забезпечити контроль доступу до даних, зберігати індивідуальні результати користувачів, обмежувати права редагування і перегляду, а також організувати роботу з декількома командами або проєктами в межах однієї системи.

На етапі ініціалізації проєкту розробляється система облікових записів, яка включає форму реєстрації та форму входу. Форма реєстрації дозволяє створити новий акаунт, де користувач вводить базову інформацію – ім'я, електронну адресу, пароль та, за потреби, роль у системі (наприклад, адміністратор, менеджер, учасник команди). Під час обробки цієї інформації пароль шифрується (наприклад, алгоритм bcrypt або

PBKDF2), після чого облікові дані зберігаються у захищеній базі даних. Такий підхід дозволяє уникнути прямого зберігання чутливої інформації у відкритому вигляді, що є важливим з міркувань безпеки.

Форма входу до системи забезпечує автентифікацію користувача на основі введених облікових даних. Після входу створюється сеанс користувача, в межах якого він має доступ до особистих даних, результатів попередніх обчислень, інтерфейсу введення нових даних, а також можливості переглядати результати візуалізацій. У межах сесії користувача реалізується система "маркерів доступу", за допомогою Flask-Login, що дозволяє зберігати стан авторизованого користувача між запитами.

Для забезпечення функціональності модуль авторизації може бути доповнений механізмами розмежування прав доступу. Наприклад, адміністратор має право створювати нові проекти, призначати учасників, редагувати вихідні дані та переглядати результати всіх користувачів у рамках проекту. Менеджер команди, може мати доступ до перегляду лише своєї групи, а звичайний користувач лише до власних результатів. Це дозволяє забезпечити конфіденційність у роботі як з персоналізованими психологічними показниками, так і загальними показниками впливу на організацію.

Крім того, можливе впровадження багаторівневої перевірки автентичності (наприклад, через одноразовий пароль або підтвердження через електронну пошту), що особливо актуально у випадках, коли платформа використовується в корпоративному середовищі або для обробки конфіденційних HR-даних.

Модуль авторизації може бути інтегрований з зовнішніми системами керування обліковими

записами (наприклад, Azure Active Directory або Google Workspace), що дозволить організаціям централізовано контролювати доступ до аналітичного інструменту. Також реалізація функціональності «забути пароль», «змінити облікові дані» та системи повідомлень підвищує зручність використання додатку та адаптивність до реальних потреб користувачів.

Модуль авторизації не лише виконує функцію захисту даних і контролю доступу, але й формує основу для побудови персоналізованої аналітики, зберігання історії обчислень, розмежування проектів та ролей у команді. Це робить його невід'ємним функціональним елементом повноцінної інформаційної системи для оцінювання комунікаційних процесів у проектному середовищі.

Таким чином, розробка веб-додатку на Python для обчислення показників впливу учасників команди є доцільним інструментом для організацій, що прагнуть до цифровізації управління проектами. Гнучкість архітектури, відкритість до масштабування та можливість інтеграції з сучасними сервісами роблять його ефективним рішенням для підвищення рівня прозорості й керованості комунікацій у проектних командах.

Архітектура запропонованого веб-додатку представлена на рис. 1 та побудована за принципами розділення відповідальності між основними функціональними модулями та має визначену структуру взаємодії між ними. Основні компоненти архітектури включають користувацький інтерфейс, серверну частину веб-додатку на базі Flask, модулі обробки даних та обчислень, базу даних для збереження інформації та, за потреби, інтеграцію з зовнішніми джерелами даних.

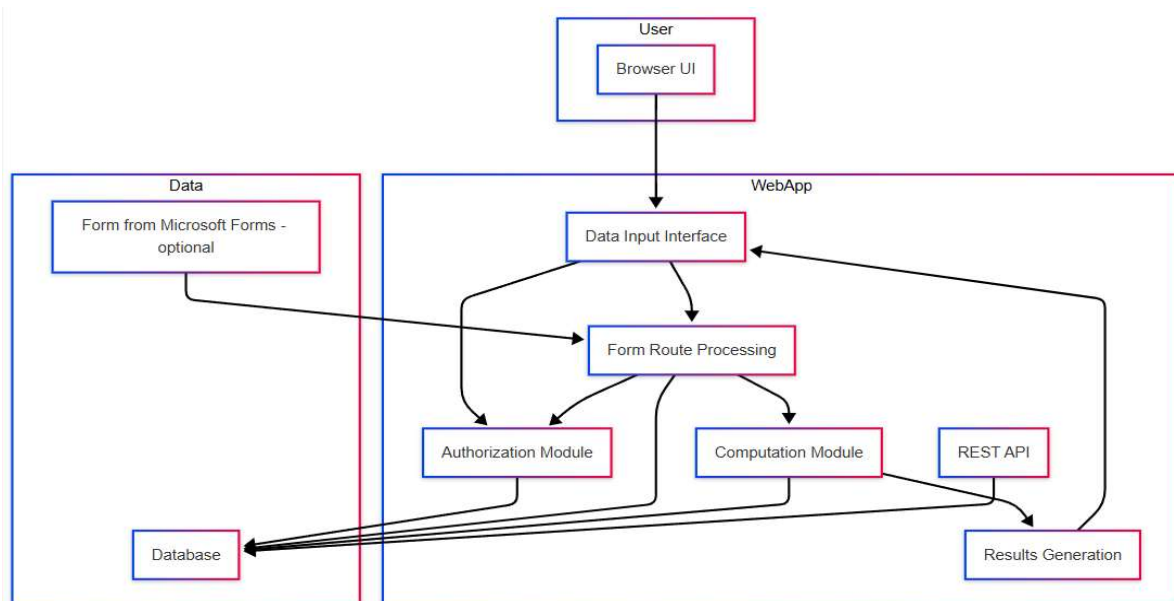


Рисунок 1 – Архітектура запропонованого веб-додатку

Користувач взаємодіє з додатком через браузерний інтерфейс, розроблений з використанням HTML-розмітки та рушія шаблонів Jinja2. Інтерфейс забезпечує введення необхідних даних для розрахунку: показників лідерських навичок (L_j) та рівнів впливу за вісьмома ключовими сферами комунікаційних бар'єрів (D_1-D_8). Цей же інтерфейс відповідає за візуалізацію результатів після проведення обчислень.

Після введення даних форма надсилається на сервер, де дані обробляються відповідно до заданих маршрутів у Flask-додатку. На цьому етапі діє модуль авторизації користувачів, який перевіряє права доступу до системи. Користувачі повинні бути автентифіковані перед тим, як отримати можливість вводити або переглядати дані. Модуль авторизації також забезпечує розмежування прав доступу залежно від ролі користувача (адміністратор, менеджер, звичайний учасник).

База даних, що підтримує роботу веб-додатку для аналізу комунікаційної системи проектних команд, побудована на основі трьох основних сутностей: users, projects і results. Така схема представлена на рис. 2 дозволяє забезпечити розмежування облікових даних користувачів, управління інформацією про проекти та зберігання результатів обчислень показників впливу для подальшого аналізу.

Таблиця users зберігає інформацію про всіх зареєстрованих користувачів системи.

Кожен запис має унікальний ідентифікатор id, а також такі атрибути, як ім'я (name), електронну

пошту (email), хеш пароля (password_hash) та роль користувача (role). Роль визначає рівень доступу користувача до функціональності системи, наприклад, розмежовуючи права адміністратора та звичайного учасника проекту. Поле created_at фіксує дату та час створення облікового запису, що важливо для аудиту та аналізу активності користувачів.

Таблиця projects відповідає за зберігання інформації про окремі проекти, у межах яких проводяться обчислення та оцінка командної взаємодії. Кожен проект має свій унікальний id, назву (name), опис (description) та прив'язку до власника проекту через зовнішній ключ owner_id, який посилається на відповідного користувача в таблиці users. Поле created_at дозволяє фіксувати момент створення проекту для ведення хронології діяльності.

Таблиця results є центральною для зберігання результатів обчислень. Кожен результат пов'язаний із конкретним користувачем (user_id) та проектом (project_id) через зовнішні ключі. Таблиця містить числові поля для збереження індивідуальних значень оцінювання: показник лідерських навичок (L_j), значення для кожної зі сфер впливу (D_1 до D_8), а також результати обчислень – поточний кумулятивний вплив (PI_j) та відсотковий вплив на систему комунікацій (I_j).

Додатково фіксується значення WD , що представляє цілісність комунікаційної системи без впливу конкретного члена команди. Поле submitted_at зберігає дату та час подання розрахунку, що дозволяє відстежувати динаміку змін у часі.

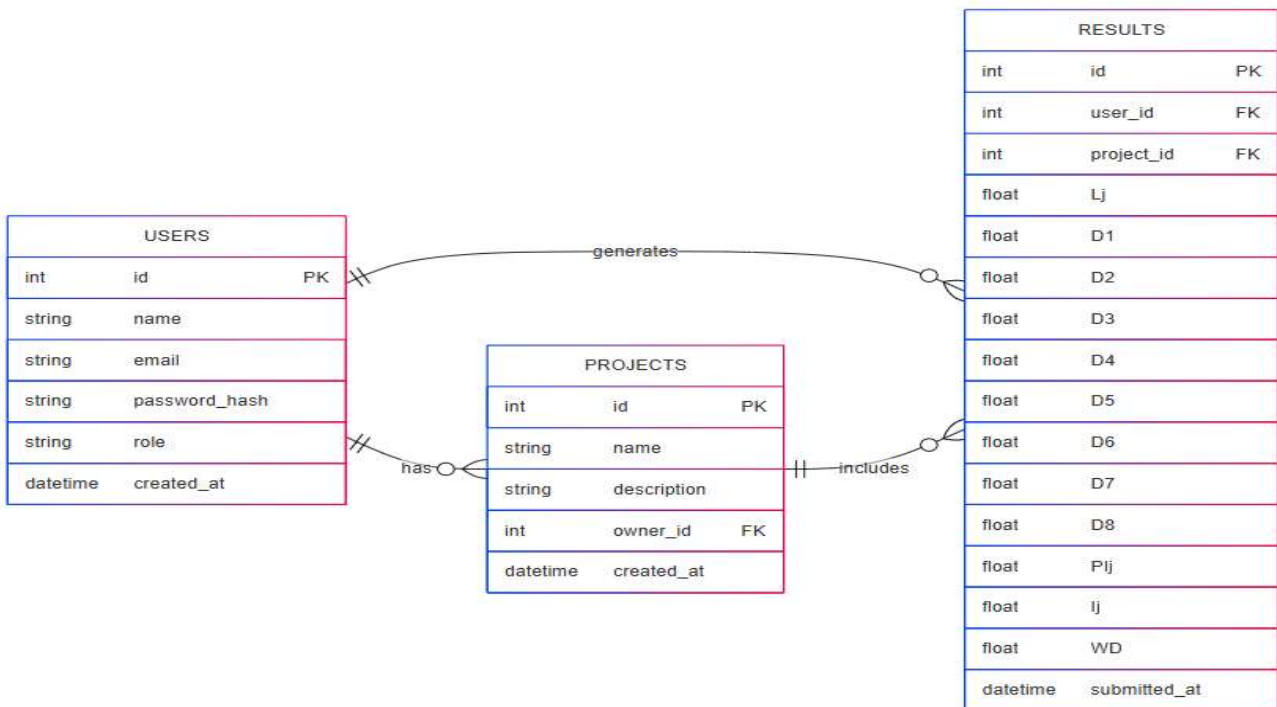


Рисунок 2 – Концептуальна схема бази даних

Така структура бази даних забезпечує гнучкість у роботі системи, дозволяючи ефективно масштабувати її як на рівні користувачів і проектів, так і на рівні обробки великої кількості результатів обчислень. Вона також підтримує необхідний рівень нормалізації даних, мінімізуючи дублювання інформації та забезпечуючи цілісність зв'язків між сутностями.

Обробка форм та маршрути, що реалізовані у Flask через спеціальні функції, спрямовують дані у обчислювальний модуль, де відбувається розрахунок показників PI_j та I_j відповідно до описаних математичних моделей. Модуль обчислень є ізольованим компонентом, що дозволяє легко підтримувати, тестувати та розширювати логіку розрахунків у майбутньому.

Після завершення обчислень результати передаються на модуль візуалізації. За допомогою бібліотек для побудови графіків, таких як Plotly або Chart.js, керівник отримує графічне відображення кумулятивного впливу кожного члена команди (див. рис. 3), що суттєво полегшує аналіз даних і прийняття рішень. Результати відображаються у формі кругових діаграм, де кожен сектор відповідає рівню впливу учасника на конкретну сферу впливу, а також у вигляді стовпчикових графіків, що показують порівняння між учасниками за різними параметрами.

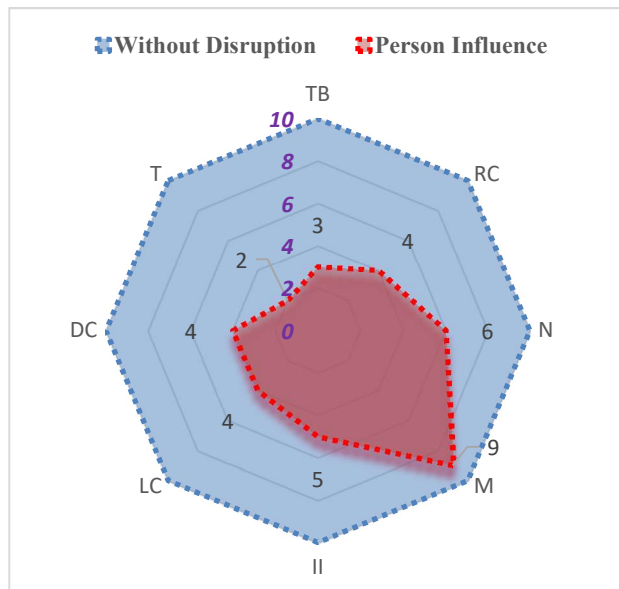


Рисунок 3 – Графічне відображення кумулятивного впливу кожного члена команди

Функціонал передбачає перегляд історії змін, експорт даних у форматі CSV або PDF, а також інтеграцію з зовнішніми системами. Користувач може побачити зміну рівня впливу кожного учасника в динаміці за допомогою лінійних графіків, що

особливо корисно для аналізу ефективності управлінських рішень. Завдяки адаптивному дизайну, додаток відображається як на десктопах, так і на мобільних пристроях, що дозволяє керівнику команди переглядати аналітику з будь-якого місця.

Окрім базового функціоналу, архітектура передбачає можливість створення REST API на базі Flask, що дозволить розширювати систему за рахунок інтеграції з іншими сервісами, мобільними додатками або корпоративними платформами.

Таким чином, запропонована архітектура забезпечує ефективну взаємодію між усіма компонентами системи, дозволяє гнучко обробляти та аналізувати поведінкові дані, забезпечує високий рівень безпеки та конфіденційності, а також створює основу для подальшого масштабування та розвитку цифрового рішення для управління комунікаціями в проектних командах.

Розробка відповідного цифрового інструменту у вигляді веб застосунку або модуля в середовищі управління проектами, дозволяє автоматизувати процес збору даних (через Microsoft Forms або аналогічні засоби), обробку результатів (за допомогою Power Automate або Python-алгоритмів), і виведення аналітики у вигляді таблиць та візуалізацій.

Висновки

У результаті проведеного дослідження було обґрунтовано доцільність інтеграції аналітичного інструменту на основі моделі FIRO-B у процес управління комунікаційною взаємодією в проектних командах. Запропонований підхід дозволяє кількісно оцінювати рівень деструктивного впливу окремих учасників на систему комунікацій та враховувати їхні лідерські характеристики при формуванні та підтримці роботи команди. Важливою складовою підходу є поєднання психологічних аспектів з технічними засобами автоматизації, що забезпечує гнучкість і адаптивність рішення.

Розроблений веб-додаток на базі фреймворку Flask реалізує логіку збору, обробки та візуалізації поведінкових даних команди. Модульна архітектура дозволяє масштабувати функціональність, інтегрувати з зовнішніми сервісами, а також підтримує збереження історії обчислень у захищеній базі даних. Застосування такого інструменту в IT- або проектно-орієнтованих організаціях сприяє підвищенню рівня прозорості, своєчасному виявленню ризиків комунікаційного характеру та прийняттю обґрунтованих управлінських рішень.

Практичне значення даного дослідження полягає у розробці інструменту для керівника

проекту, що дозволяє виявляти поведінкові ризики в команді, проводити ротацию членів команди та ролей, розробляти індивідуальні стратегії розвитку лідерства або вживати профілактичних заходів для збереження комунікаційної стабільності.

Подальші дослідження будуть зосереджені на розширенні системи засобами штучного інтелекту,

зокрема алгоритмами машинного навчання, для прогнозування впливу окремих поведінкових характеристик на динаміку комунікацій у командах. Крім того, перспективним напрямом є розробка системи рекомендацій щодо ролей учасників команди на основі їхніх поведінкових профілів і поточних умов проекту.

Список літератури / References

1. Adamu, J., Hamzah, R., & Rosli, M. M. (2020). Security issues and framework of electronic medical record: A review. *Bulletin of Electrical Engineering and Informatics*, 9 (2), 565–572. <https://doi.org/10.11591/eei.v9i2.1834>.
2. Baida, R., Andriienko, M., & Plechawska-Wójcik, M. (2020). Analiza porównawcza wydajności frameworków Angular oraz Vue.js. *Journal of Computer Sciences Institute*, 14, 59–64.
3. Chauhan, N., Singh, M., Verma, A., Parasher, A., & Budhiraja, G. (2019). Implementation of database using Python Flask framework: College database management system. *International Journal of Engineering and Computer Science*, 8 (12), 24894–24899. <https://doi.org/10.18535/ijecs/v8i12.4390>.
4. Church, A. (2014). Schutz, W. C. (1958). FIRO, A three-dimensional theory of interpersonal behavior. Rinehart & Company, Inc. *The Journal of Symbolic Logic*, 24, 216–217. <https://doi.org/10.2307/2963791>.
5. Kaluža, M., Troskot, K., & Vukelić, B. (2018). Comparison of front-end frameworks for web applications development. *Zbornik Veleučilišta u Rijeci*, 6 (1), 261–282.
6. Kontsevyi, V., & Voitenko, O. (2023). Communications disruptor in project-oriented organisations. *2023 IEEE 18th International Conference on Computer Science and Information Technologies (CSIT)*, 1–4. <https://doi.org/10.1109/CSIT61576.2023.10324097>.
7. Northouse, P. G. (2007). *Leadership theory and practice* (4th ed.). Sage Publications, Inc.
8. Pazdriy, I. (2023). Comparative analysis of software development systems based on frameworks. *Herald of Khmelnytskyi National University. Technical sciences*, 317, 155–161. <https://www.google.com/search?q=https://doi.org/10.31891/2307-5732-2023-317-1-155-161>.
9. Petukhova, E. (2019). *Sitecore JavaScript Services Framework Comparison*. Åbo Akademi University, Faculty of Science and Engineering.
10. Riggs, S., & Ciolli, G. (2015). *PostgreSQL 9 administration cookbook* (2nd ed.). Packt Publishing.
11. Truica, C., & Boicea, A. (2013). Asynchronous replication in Microsoft SQL Server, PostgreSQL and MySQL. *Proceedings of the International Conference on Cyber Science and Engineering*, Guangzhou, China.
12. Wodyk, R., & Skublewska-Paszkowska, M. (2020). Performance comparison of relational databases SQL Server, MySQL and PostgreSQL using a web application and the Laravel framework. *Journal of Computer Sciences Institute*, 17, 358–364. <https://doi.org/10.35784/jcsi.2279>.

Стаття надійшла до редакції 04.05.2025

Kontsevyi Vladyslav

PhD student of project management department,

<https://orcid.org/0000-0002-6636-5489>

Kyiv national university of construction and architecture, Kyiv

Voitenko Oleksandr

Ph.D., Associate Professor Department of Project Management,

<https://orcid.org/0000-0002-7414-7836>

Kyiv National University of Civil Engineering and Architecture, Kyiv

ASSESSMENT OF THE IMPACT OF PROJECT TEAM MEMBERS ON THE COMMUNICATIONS SYSTEM

Abstract. This article presents a comprehensive approach to assessing the impact of project team members on the communication system within a digital environment. The relevance of this study stems from the increasing prevalence of virtual and hybrid teams in contemporary organizations, where communication effectiveness relies significantly not only on technical resources but also on the behavioral attributes of the participants. The study identifies that individuals can act as disruptive factors, affecting the quality of team interaction and information flow. To identify and quantify this influence, the FIRO-B model was employed, which allows for the determination of each team member's interpersonal orientation across three key dimensions:

inclusion, control, and affection. These characteristics were further analyzed in conjunction with an assessment of eight areas of destructive influence: technological barriers, resistance to change, negativity, micromanagement, misinformation, lack of cooperation, dominance in communications, and time disruption. Formulas for calculating the individual indicator of current influence and the relative influence on the communication system have been developed, incorporating an inverse relationship with the level of leadership skills. The practical implementation is realized through a web application built on the Flask framework. The description details the architecture of the software solution, which includes server logic, a user interface, a database, an authorization module, and a system for visualizing results. The application enables the input of individual team member indicators, automatically calculates their level of influence on the communication environment, and displays the results in tables and graphs. For data processing and analysis, integration with SQLite/PostgreSQL databases is implemented, along with the capability to export results and automate the collection of input information using Microsoft Forms and Power Automate. The developed tool allows project managers to promptly identify behavioral risks, evaluate the effectiveness of the communication structure, and make informed management decisions regarding team composition.

Keywords: *virtual teams; IT organization; FIRO-B method; SQL; Python*

Посилання на публікацію

- APA Kontsevyi, V., & Voitenko, O. (2025). Assessment of the impact of project team members on the communications system. *Management of Development of Complex Systems*, 62, 79–87, dx.doi.org/10.32347/2412-9933.2025.62.79-87.
- ДСТУ Концевий В. В., Войтенко О. С. Оцінка впливу учасників проєктних команд на систему комунікацій. *Управління розвитком складних систем*. Київ, 2025. № 62. С. 79 – 87, dx.doi.org/10.32347/2412-9933.2025.62.79-87.