

<https://doi.org/10.31891/2307-5732-2026-363-12>

УДК 004.4:004.45:65.011.56

### САМЧУК ЛЮДМИЛА

Луцький національний технічний університет, Україна

<https://orcid.org/0000-0003-2516-045X>

e-mail: samchuk204@gmail.com

### ПОВСТЯНА ЮЛІЯ

Луцький національний технічний університет, Україна

<https://orcid.org/0000-0001-5426-4157>

e-mail: yuliapovstiana@ukr.net

## ВІЗУАЛІЗАЦІЯ ОРГАНІЗАЦІЙНОЇ СТРУКТУРИ ІТ-КОМПАНІЇ ЗАСОБАМИ UML

У даній роботі розглянуто процес моделювання організаційної структури ІТ-компанії на основі уніфікованої мови моделювання UML. Основна мета дослідження полягає у створенні моделей, що відображають взаємодію ролей, структуру підприємства та бізнес-процеси в межах інформаційної системи організації. У межах дослідження побудовано ключові UML-діаграми: діаграму варіантів використання, діаграму класів, діаграму послідовності та діаграму діяльності. Вибір саме цього набору діаграм обумовлений необхідністю покрити основні архітектурні перспективи, як це вимагає визнана в інженерії програмного забезпечення «4+1 View Model».

Кожна з діаграм висвітлює окремий аспект функціонування ІТ-компанії. Зокрема, діаграма варіантів використання моделює взаємодію між основними акторами системи: аналітиком, системним адміністратором, проектним менеджером, HR-менеджером, розробником та керівником відділу. Діаграма класів описує логічну структуру системи, включаючи сутності «Працівник», «Відділ», «Команда», «Проект», «Завдання» та інші. Діаграма послідовності ілюструє динаміку створення та погодження технічного завдання в системі документообігу. Діаграма діяльності демонструє процес обробки технічного звернення клієнта службою підтримки.

Запропоноване моделювання дозволяє отримати комплексне уявлення про внутрішні процеси ІТ-компанії, підвищити ефективність взаємодії між підрозділами, а також створити надійну основу для подальшого проектування програмної інфраструктури. Результати дослідження можуть бути використані в навчальних цілях, а також як аналітичний інструмент для вдосконалення управлінських та організаційних процесів у сфері ІТ.

**Ключові слова:** UML-діаграми, організаційна структура, інформаційна система, ІТ-компанія, моделювання, бізнес-процеси.

SAMCHUK LYUDMILA, POVSTIANA YULIYA

Lutsk National Technical University

## VISUALIZATION OF THE ORGANIZATIONAL STRUCTURE OF AN IT COMPANY USING UML.

This research paper examines the process of modeling the organizational structure of an IT company based on the Unified Modeling Language (UML). The main objective of the study is to create models that reflect the interaction of roles, the enterprise structure, and the business processes within the organization's information system. Choosing this particular set of diagrams based on need opens up key architectural perspectives, as recognized in software engineering's «4+1 View Model».

Within the study, the key UML diagrams were developed: the use case diagram, class diagram, sequence diagram, and activity diagram. Each of these diagrams highlights a specific aspect of the IT company's operation. In particular, the use case diagram models the interaction among the main system actors: analyst, system administrator, project manager, HR manager, developer, and department head. The class diagram describes the logical structure of the system, including entities such as Employee, Department, Team, Project, and Task. The sequence diagram illustrates the dynamics of creating and approving a technical specification within the document management system, while the activity diagram demonstrates the process of handling a client's technical request by the support service.

The proposed modeling approach provides a comprehensive understanding of the internal processes of an IT company, enhances the efficiency of interdepartmental communication, and establishes a solid foundation for further software infrastructure design. The research results can be used for educational purposes as well as an analytical tool for improving management and organizational processes in the IT domain.

**Keywords:** UML diagrams, organizational structure, information system, IT company, modeling, business processes.

Стаття надійшла до редакції / Received 18.01.2026

Прийнята до друку / Accepted 11.02.2026

Опубліковано / Published 26.03.2026



This is an Open Access article distributed under the terms of the [Creative Commons CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)

© Самчук Людмила, Повстяна Юлія

### Постановка наукової проблеми

У сучасних ІТ-компаніях організаційні структури часто містять численні ролі, підрозділи та бізнес-процеси, що ускладнює їхню візуалізацію й аналіз. Хоча Unified Modeling Language (UML) є загальноприйнятим засобом моделювання об'єктно-орієнтованих систем і забезпечує потужний механізм для візуалізації дизайну, його застосування саме для формалізації внутрішньої організаційної архітектури залишається фрагментарним і неузгодженим [1].

У контексті Enterprise Modelling визнається необхідність комплексного зображення структури, поведінки та ресурсів підприємства, проте UML-нотації зазвичай використовуються в обмежених профілях або для окремих аспектів (як-от компонентна або класова модель), без уніфікації підходу до ієрархії підрозділів і ролей [2]. Незважаючи на те, що діаграми випадків використання здатні наочно відобразити взаємодію ключових акторів із системою, сучасні практики рідко пропонують цілісну методологію, яка б поєднувала їх із класовими, послідовнісними та діяльними діаграмами для повного опису організаційної структури ІТ-компанії.

Діаграми класів дозволяють формалізувати статичну ієрархію сутностей, проте в більшості досліджень вони використовуються переважно для опису даних і зв'язків між ними, а не для моделювання ролей чи взаємодій. Діаграми діяльності з підтримкою swimlanes справді забезпечують можливість візуалізувати бізнес-процеси та відповідальність між командами, але їх застосування в аналізі внутрішньої організаційної структури IT-організацій досі обмежене, що створює прогалини в розумінні потоків інформації та зон відповідальності.

### Аналіз досліджень

Ефективна візуалізація організаційної структури та бізнес-процесів є критично важливою передумовою для успішного проектування та впровадження інформаційних систем у сучасних IT-компаніях. Ця необхідність підтверджується дослідженнями в галузі архітектури підприємства (EA), де організаційна архітектура визначена як фундаментальний шар, що забезпечує контекст для подальших IT-рішень. Зокрема, наукові праці, що використовують такі фреймворки, як Zachman Framework, підкреслюють, що планування архітектури IT-систем починається з глибокого аналізу бізнес-процесів і ролей [3]. Уніфікована мова моделювання (UML) виступає як ключовий інструмент, оскільки вона забезпечує формалізований апарат для опису не лише програмного забезпечення, але й бізнес-моделей, що є важливим для узгодження бізнесу та технологій [4]. При цьому концептуальну основу побудови таких моделей доцільно розглядати у контексті підходу «4+1 View Model», запропонованого Ф. Круштеном [5], який визначає п'ять взаємопов'язаних перспектив архітектури системи: логічну, процесну, розгортання, реалізації та сценарії використання. Саме цей підхід забезпечує структурно-поведінкову цілісність моделі та дає змогу поєднати різні аспекти системи – від статичних структур до динаміки взаємодій. Таким чином, застосування UML-діаграм у поєднанні з принципами «4+1 View Model» дозволяє досягти глибшого рівня узгодженості між бізнес-вимогами, архітектурними рішеннями та процесами функціонування IT-компанії.

Актуальність використання UML підтверджується також його роллю у формуванні архітектурних рішень. Дослідження [6] фокусується на значущості різних UML-діаграм, зокрема у середніх IT-проектах, підкреслюючи, що комбінація діаграм класів, діяльності та варіантів використання є найбільш ефективною для охоплення статичних, динамічних та функціональних аспектів системи. Це узгоджується з необхідністю інтеграції бізнес-процесів у загальну корпоративну інформаційну систему. Робота [7] безпосередньо розглядає інтеграцію бізнес-процесів в ІС підприємства за допомогою UML, демонструючи, як діаграми діяльності та послідовності можуть бути використані для усунення розриву між бізнес-вимогами та технічною реалізацією. Таким чином, наукова спільнота визнає, що багатоаспектне моделювання UML є необхідною умовою для цілісного проектування.

Водночас, вибір UML як основної нотації вимагає обґрунтування на тлі існування спеціалізованих мов моделювання архітектури підприємства, таких як ArchiMate або Unified Architecture Framework (UAF). Порівняльний аналіз, проведений у роботі [8], показує, що хоча спеціалізовані фреймворки можуть бути більш комплексними, UML зберігає свою позицію як універсальна та гнучка мова, що здатна охоплювати логічні, процесні, та функціональні аспекти. UML обирається як базис для даного дослідження саме завдяки його стандартизації та балансу між деталізацією, необхідною для технічного проектування, і широкою впізнаваністю в інженерному середовищі.

Виявлення наукової прогалини. Аналіз останніх публікацій чітко засвідчує наукову потребу поперше: в моделюванні організаційної архітектури (ролі, структура, процеси) як основи для IT-систем [3], по друге використанні багатоаспектного набору UML-діаграм для забезпечення цієї інтеграції [6, 7], та по третє обґрунтуванні вибору UML як універсального інструменту [4, 8]. Однак, наявна література недостатньо повно висвітлює комплексний, цілісний підхід до візуалізації всіх цих аспектів – статичної організаційної структури, динаміки процесів та взаємодії ролей – у межах однієї моделі IT-компанії. Отже, метою даної роботи є заповнення цієї прогалини шляхом розробки цілісної UML-моделі організаційної структури IT-компанії, яка слугуватиме основою для проектування ефективної інформаційної системи.

У контексті архітектури підприємства (EA), вибір оптимальної мови моделювання є предметом наукової дискусії, оскільки різні нотації, такі як ArchiMate та BPMN, пропонують специфічні інструменти для окремих шарів архітектури. Тоді як ArchiMate спеціалізується на моделюванні цілісної архітектури підприємства (від бізнес-стратегії до технологічної інфраструктури), а BPMN (Business Process Model and Notation) орієнтована виключно на процеси, UML (Unified Modeling Language) зберігає позицію універсального та гнучкого інструменту [8]. У порівняльних дослідженнях методів моделювання EA було виявлено, що, попри комплексність спеціалізованих фреймворків, UML залишається ключовою універсальною мовою, здатною охоплювати різні аспекти: від логічного до процесного та варіантів використання [8]. Виходячи з цього, UML обирається як базис для даного дослідження завдяки його здатності створювати точні, технічно реалізовані моделі, які є зрозумілими як для інженерного персоналу (розробників), так і для бізнес-аналітиків.

Таким чином, хоча окремі дослідження та практичні посібники демонструють ефективність окремих UML-нотацій, відсутня єдина методологія, яка б поєднувала всі ключові діаграми (use case, class, sequence, activity) для всебічного моделювання організаційної структури IT-компанії.

### Мета роботи

Сформулювати моделі організаційної структури IT-компанії за допомогою засобів уніфікованої мови моделювання (UML), що дозволяють відобразити ролі співробітників, взаємозв'язки між підрозділами, а також

ключові бізнес-процеси, пов'язані з управлінням кадрами, документацією та технічною підтримкою, з метою створення формалізованої основи для проектування інформаційної системи підприємства.

#### Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.

Організаційна структура ІТ-компанії охоплює широкий спектр ролей і взаємодій, які визначають ефективність управління проектами, кадровими ресурсами та технічною інфраструктурою. Для побудови сучасної інформаційної системи, що підтримує ці процеси, важливо чітко формалізувати вимоги та структуру взаємодій між учасниками. У цьому контексті уніфікована мова моделювання UML виступає як потужний інструмент для візуалізації складних бізнес-процесів і створення цілісної архітектури системи.

Організаційну структуру ІТ-компанії подамо у вигляді формалізованої трійки:

$$M=(R, P, S) \quad (1)$$

де  $R=\{r_1, r_2, \dots, r_n\}$  – множина ролей (аналітик, менеджер, адміністратор, розробник тощо),

$P=\{p_1, p_2, \dots, p_m\}$  – множина бізнес-процесів (керування проектами, підтримка, HR-процеси, документообіг),

$S$  – структура взаємодій між ролями та процесами, задана у вигляді відображення:

$$S:R \times P \rightarrow \{0, 1\} \quad (2)$$

де  $S(r_i, p_j)=1$  означає участь ролі  $r_i$  у процесі  $p_j$ .

Для оцінювання ефективності запропонованої UML-моделі пропонується використовувати такі критерії:

- середній час узгодження технічного завдання ( $T_{avg}$ );
- кількість конфліктів або неузгодженостей між ролями ( $C$ );
- кількість повторних звернень до служби підтримки ( $N_{sup}$ ).

Нехай  $n$  – кількість технічних завдань, що пройшли процедуру погодження протягом аналізованого періоду.

Середній час узгодження технічного завдання визначається як:

$$T_{avg} = \frac{1}{n} \sum_{i=1}^n (t_{approve}^{(i)} - t_{create}^{(i)}) \quad (3)$$

де  $t_{create}^{(i)}$  – момент створення  $i$ -го технічного завдання,

$t_{approve}^{(i)}$  – момент його остаточного погодження.

Кількість конфліктів або неузгодженостей між ролями визначається як загальна кількість випадків повернення технічного завдання на доопрацювання внаслідок виявлених суперечностей між учасниками процесу:

$$C = \sum_{i=1}^n c_i \quad (4)$$

де  $c_i$  – кількість зафіксованих конфліктів або неузгодженостей, пов'язаних з  $i$ -м технічним завданням.

Кількість повторних звернень до служби підтримки визначається як загальна кількість звернень, що повторюються щодо однієї й тієї ж проблеми (інциденту).

Нехай  $G$  – множина всіх інцидентів, а  $k_g$  – кількість звернень, пов'язаних з інцидентом  $g$ , тоді:

$$N_{sup} = \sum_{g \in G} \max(0, k_g - 1) \quad (5)$$

Ефективність моделі може вважатися підвищеною, якщо після її впровадження спостерігається зменшення значень  $T_{avg}$ ,  $C$  та  $N_{sup}$ , що свідчить про покращення координації процесів і зниження кількості помилок у взаємодії між ролями.

У роботі використано ключові типи UML-діаграм: діаграма варіантів використання дозволила визначити функціональні можливості системи та ролі основних акторів (аналітик, менеджер, адміністратор, розробник, HR-фахівець тощо); діаграма класів – структуру системи, що включає сутності, атрибути та зв'язки між ними; діаграма послідовності – сценарій взаємодії компонентів під час обробки технічного завдання; діаграма діяльності – логіку вирішення запитів клієнтів службою підтримки.

Таким чином, застосування UML-моделювання дозволяє не лише створити чітке уявлення про внутрішню організацію ІТ-компанії, а й забезпечити надійну основу для подальшого створення або оптимізації інформаційної системи, що підтримує її повсякденну діяльність.

Запропонована модель була піддана експертному оцінюванню з боку фахівців у сфері бізнес-аналізу та управління ІТ-проектами. За результатами оцінювання встановлено, що модель є зрозумілою, логічно узгодженою та придатною для використання як основа проектування інформаційної системи, а також для аналізу й оптимізації внутрішніх процесів ІТ-компанії.

На діаграмі випадків використання (рис. 1), що моделює організаційну структуру ІТ-компанії, представлено функціональні можливості системи та взаємодію між її ключовими учасниками. У межах даної моделі визначено шість основних акторів: Аналітик, Системний адміністратор, Проектний менеджер, HR-менеджер, Розробник та Керівник відділу. Кожен із них виконує конкретну роль у загальній структурі бізнес-процесів організації.

Головною фігурою у стратегічному аналізі та проектуванні рішень виступає Аналітик. Йому належать такі функції, як: аналіз даних, побудова діаграм і схем, формування технічного завдання, презентація результатів та перевірка відповідності реалізації вимогам. Особливої уваги заслуговує взаємозв'язок між формуванням технічного завдання та збором вимог – останнє включено до складу першого як обов'язковий етап (зв'язок типу <<include>>).

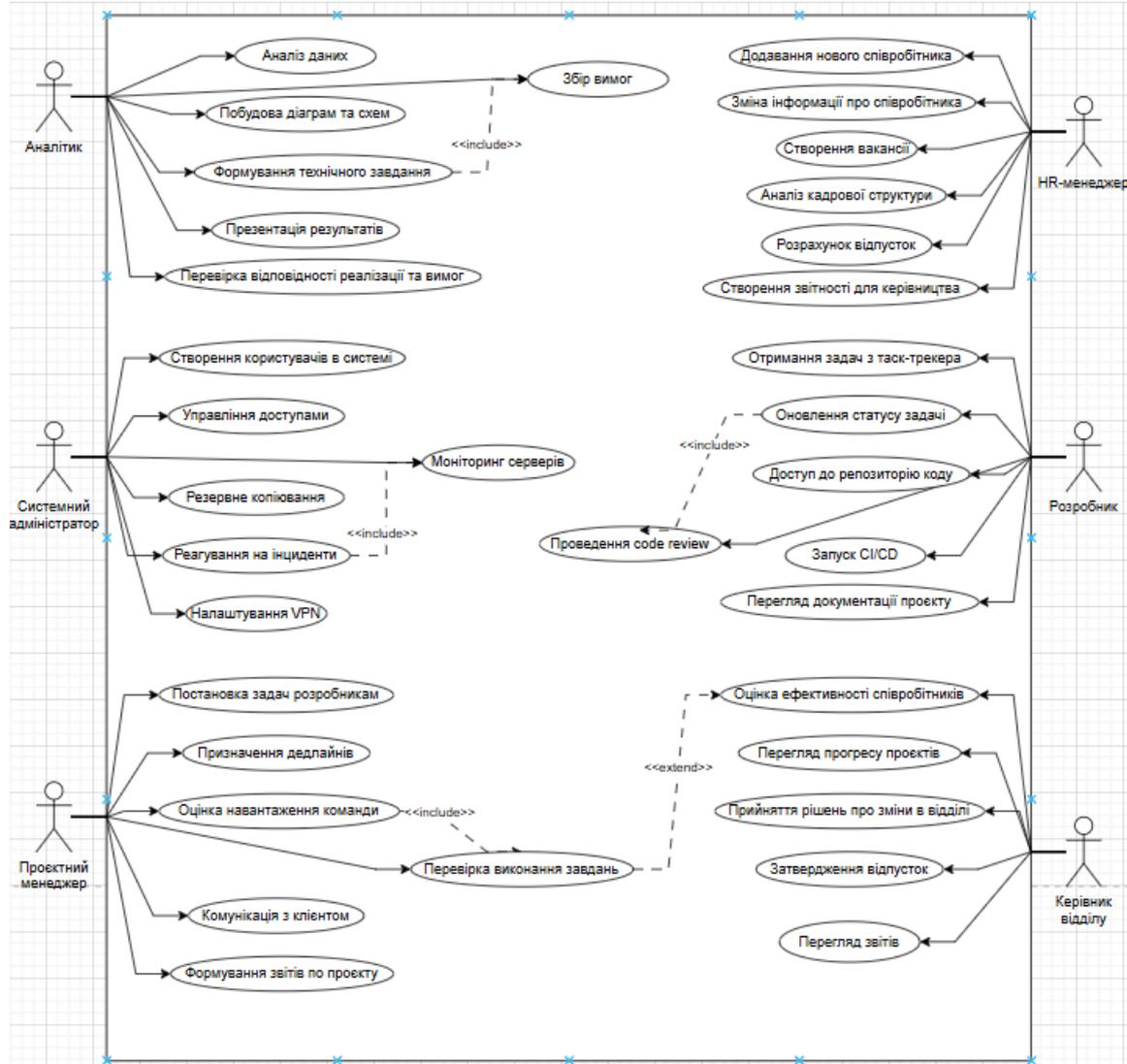


Рис. 1. Діаграма випадків використання

Системний адміністратор відповідає за технічну підтримку та забезпечення стабільної роботи внутрішніх систем компанії. До його обов'язків належать: створення облікових записів користувачів, управління доступами, моніторинг серверів, налаштування VPN-з'єднань, резервне копіювання та реагування на інциденти. У разі виявлення проблем під час моніторингу активується розширений прецедент реагування на інциденти, що є результатом умовного сценарію (зв'язок <<extend>>).

Проектний менеджер координує роботу команди та слідкує за виконанням проектних завдань. Його роль полягає у постановці задач розробникам, призначенні дедлайнів, оцінці завантаженості команди, комунікації з клієнтами та формуванні звітів щодо проектного прогресу. Важливим є зв'язок оцінка навантаження команди з перевіркою виконання завдань, яка включається до загального аналізу поточного статусу команди.

HR-менеджер реалізує кадрову політику компанії, зокрема, виконує дії з додавання та оновлення даних співробітників, створення вакансій, аналізу кадрової структури, розрахунку відпусток та формування звітності для керівництва. Ці процеси спрямовані на підтримку стабільної HR-екосистеми компанії.

Розробник бере участь у технічному втіленні рішень: він отримує задачі з task-трекера, оновлює їхній статус, працює з репозиторієм коду, запускає CI/CD-процеси, переглядає проектну документацію та виконує code review. Важливо, що оновлення статусу задачі включає проведення code review як частину перевірки результатів виконання.

Керівник відділу виконує управлінські функції високого рівня: оцінює ефективність співробітників, переглядає прогрес проектів, приймає рішення щодо змін у відділі, затверджує відпустки та аналізує звітну документацію. У ряді випадків дії керівника розширюють або завершують ініціативи, розпочаті іншими учасниками команди (наприклад, через <<extend>>-зв'язок до оцінки ефективності на основі перевірки виконання завдань).

Отже, діаграма дозволяє візуально представити структуру функціональної взаємодії між учасниками ІТ-компанії, деталізує їхні ролі та сфери відповідальності. Такий підхід до моделювання є надзвичайно

корисним як для аналізу існуючих процесів, так і для проектування нових інформаційних систем або оптимізації внутрішньої організаційної структури.

Далі йде побудова діаграми класів, що моделює логічну структуру організаційної системи ІТ-компанії. Вона відображає сутності, які становлять кадрову, управлінську та операційну архітектуру організації, їх атрибути, методи та типи взаємозв'язків. Діаграма класів (рис. 2) наочно демонструє внутрішню взаємодію між співробітниками, командами, проектами, задачами, активами, активами та супровідною документацією.

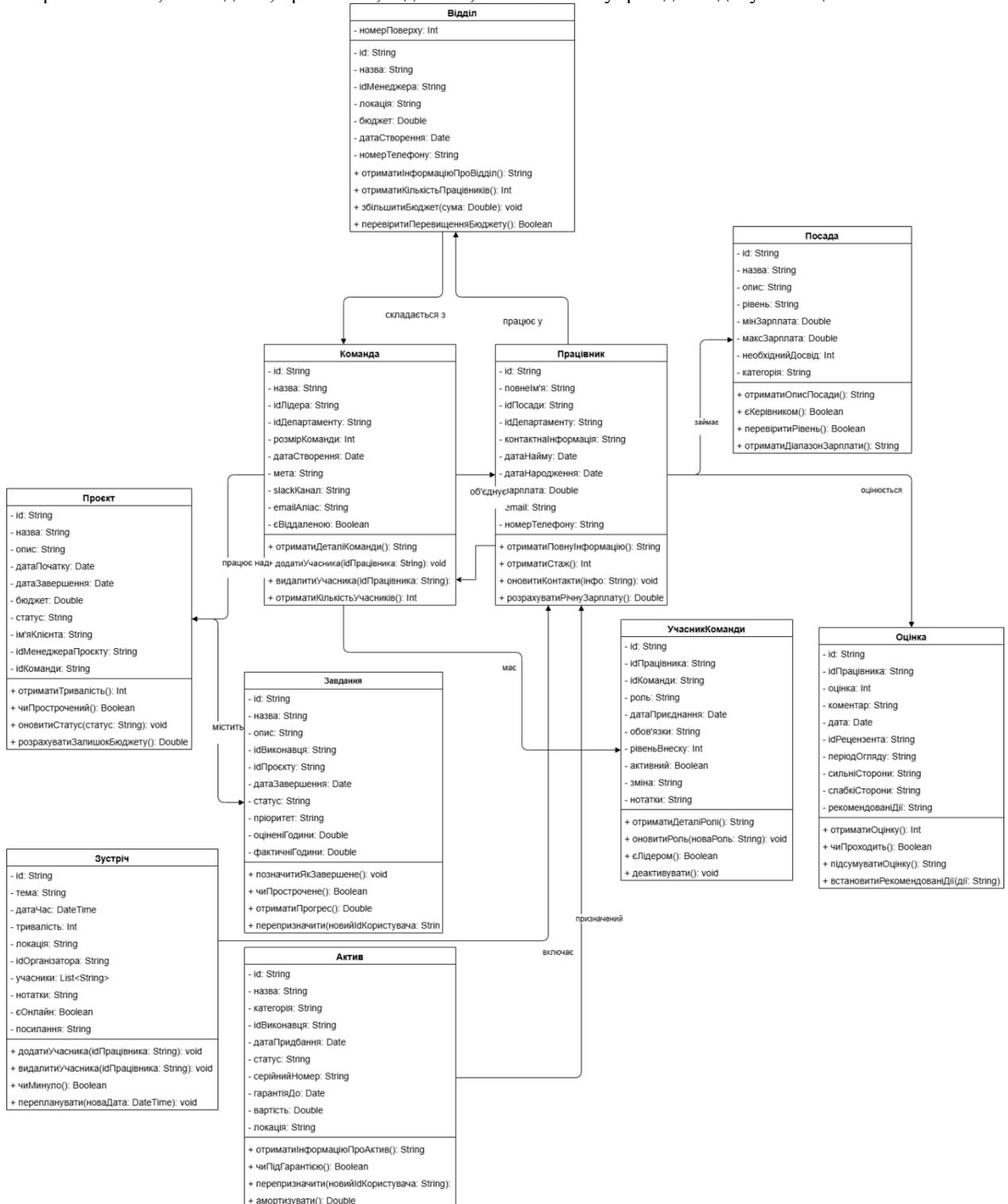


Рис. 2. Діаграма класів

Клас «Працівник» є центральною сутністю діаграми, яка представляє працівника ІТ-компанії. Містить унікальний ідентифікатор, особисті дані, контактну інформацію, посаду, зарплатні дані, а також дату народження та прийняття на роботу. Основні методи дозволяють отримати повну інформацію про співробітника, розрахувати трудовий стаж та річну зарплату. Клас асоційований з багатьма іншими, включаючи: Посада, Відділ, Член команди, Проект, Завдання, Оцінювання, Актив та Зустріч, що ілюструє його ключову роль у структурі.

Клас «Посада» визначає службовий рівень співробітника: його роль, межі зарплати, вимоги до досвіду та рівень управлінської відповідальності. Посада прив'язана до одного або декількох співробітників та дозволяє аналізувати кадрову ієрархію компанії. Включає методи перевірки рівня, категоризації та отримання зарплатного діапазону.

Клас «Відділ» відображає структурну одиницю організації. Містить поля для назви, бюджету, локації та телефону. Кожен відділ має менеджера (зв'язок з Працівник) та включає кілька Команд. Також підтримується метод обчислення кількості працівників, перевірка бюджету та його зміна.

Клас «Команда» моделює групу співробітників, об'єднаних для реалізації проєктів. Команда має назву, лідера, дату створення, Slack-канал, мету та зв'язки з Відділ, Проєкт та Член команди. Структура дозволяє керувати кількістю учасників, визначати приналежність до департаменту та додавати/видаляти працівників.

Клас «Член команди» є проміжною структурою між Працівник та Команда. Містить роль, дату приєднання, активність, зміну, нотатки та рівень внеску. Зв'язок реалізовано як «багато до багатьох», що дозволяє одному співробітнику бути частиною декількох команд.

Клас «Проєкт» описує об'єкт, над яким працює команда. Має назву, опис, бюджет, статус, дату початку і завершення, а також керівника (Працівник) і команду-виконавця. Кожен проєкт складається з кількох Завдань. Методика дозволяє оцінити тривалість, бюджет та виконання.

Клас «Завдання» моделює окремі операційні одиниці проєкту. Має назву, опис, дедлайн, статус, пріоритет, відповідального, оцінений і фактичний час. Зв'язок з Проєкт є композиційним: при видаленні проєкту пов'язані задачі також зникають. Методи дозволяють управляти прогресом, перевіряти запізнення та переназначати відповідального.

Клас «Оцінювання» відповідає за збереження результатів періодичного оцінювання співробітників. Має оцінку, коментар, рецензента, період оцінювання, сильні та слабкі сторони та рекомендації. Підтримується зв'язок з двома Працівниками: оцінюваним і тим, хто проводить оцінку.

Клас «Зустріч» організаційна подія, в якій беруть участь співробітники. Містить тему, дату, тривалість, місце проведення, посилання (за умови онлайн-формату), список учасників та організатора. Забезпечує додавання, видалення учасників, а також перевірку актуальності зустрічі.

Клас «Актив» містить дані про обладнання чи ресурси, закріплені за працівниками (ноутбуки, телефони, тощо). Містить унікальні дані, такі як серійний номер, місце зберігання, дата купівлі, вартість, гарантія. Зв'язок один-до-одного з Працівник.

Ця діаграма класів забезпечує всебічне представлення внутрішньої структури ІТ-компанії. Вона дає змогу проаналізувати кадрову модель, керування командами, розподіл завдань, використання ресурсів, систему оцінювання працівників та ефективність організаційних процесів.

На діаграмі послідовності (рис. 3) змодельовано поведінку інформаційної системи ІТ-компанії під час створення, збереження та погодження технічного завдання (ТЗ). Діаграма демонструє типову взаємодію між ключовими об'єктами організаційної структури компанії: проєктним менеджером, веб-клієнтом, сервером додатку, базою даних, системою менеджменту документів та менеджером команди розробників.

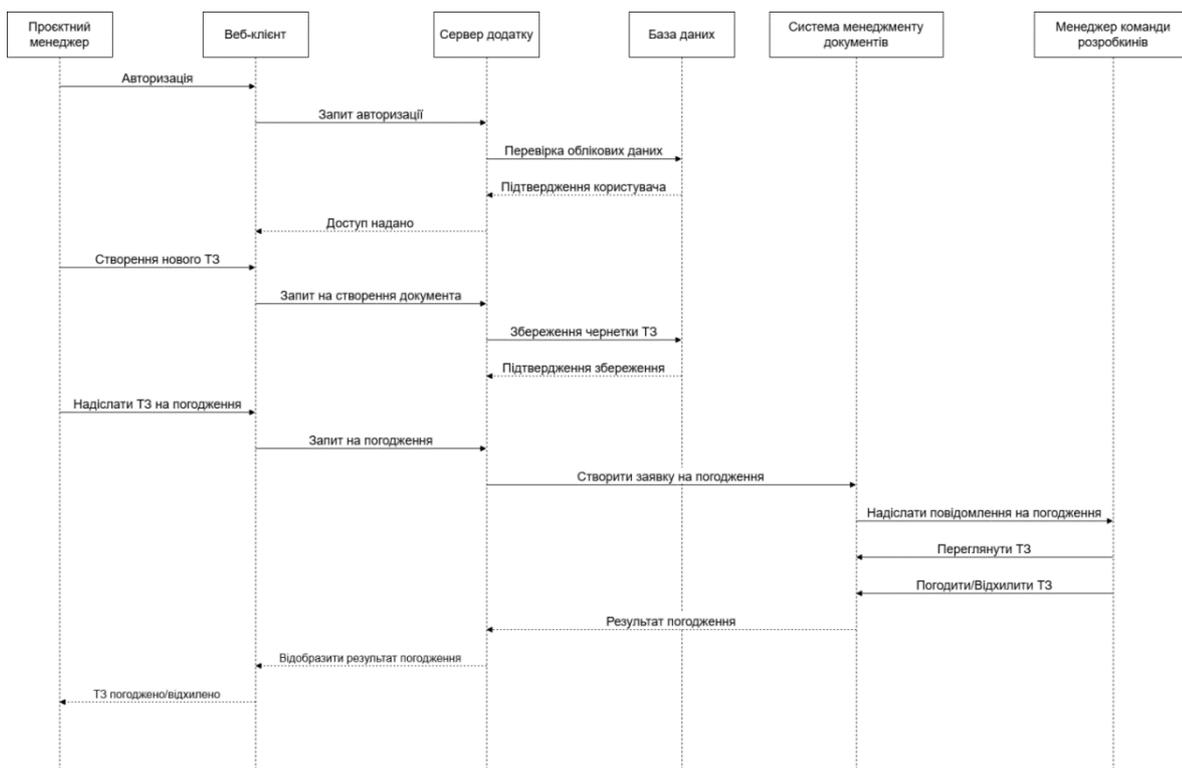


Рис. 3. Діаграма послідовності інформаційної системи

Процес починається з авторизації проектного менеджера у системі через веб-клієнт. Після успішної перевірки облікових даних, система надає доступ до функціоналу. Наступним етапом є створення нового документа ТЗ, що ініціюється менеджером через веб-інтерфейс. Запит обробляється сервером додатку, який зберігає чернетку ТЗ у базі даних та підтверджує успішне збереження.

Після завершення редагування, ТЗ надсилається на погодження. Сервер додатку генерує відповідний запит, який передається до системи управління документами. Вона, у свою чергу, надсилає повідомлення менеджеру команди розробників для ознайомлення з документом. Менеджер аналізує ТЗ та приймає рішення про його затвердження або відхилення. Відповідь про результат погодження надходить до системи документообігу, яка передає її назад на сервер додатку.

У фінальній частині процесу проектний менеджер отримує результат погодження ТЗ, який він може переглянути у веб-клієнті. Якщо документ було відхилено, менеджер може внести правки та повторити процес погодження.

Таким чином, ця діаграма демонструє послідовність взаємодії між організаційними ролями та програмними компонентами, що забезпечують життєвий цикл документа в рамках внутрішньої документації ІТ-компанії. Така модель дозволяє візуалізувати комунікаційні шляхи в структурі компанії та відстежити динаміку роботи з технічною документацією.

Діаграма діяльності (рис. 4) моделює типову процедуру обробки звернення до технічної підтримки в межах організаційної структури ІТ-компанії. Вона ілюструє ключові етапи комунікації між учасниками процесу, включаючи клієнта, менеджера підтримки, ІТ-фахівця та DevOps-інженера. Така діаграма дозволяє візуалізувати логіку роботи різних підрозділів, зокрема їхню взаємодію у процесі вирішення технічних проблем, що виникають у клієнтів компанії.

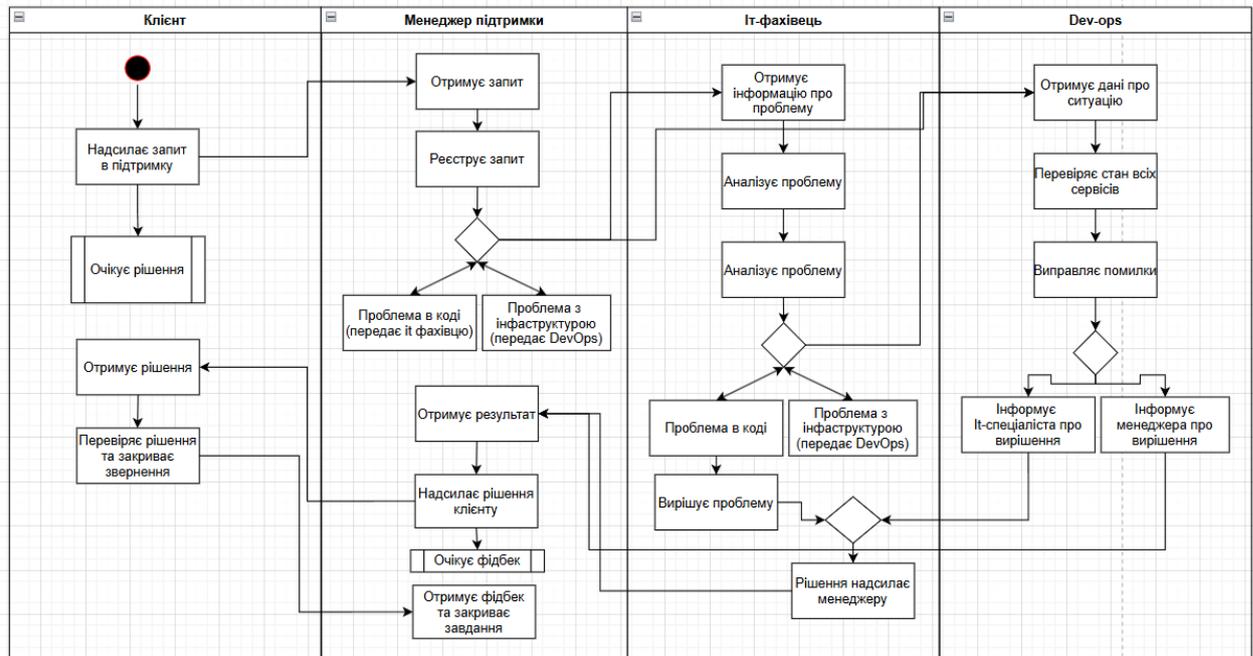


Рис. 4. Діаграма діяльності

На діаграмі використано вертикальні смуги (swimlanes), кожна з яких відповідає окремому учаснику процесу: Клієнт, Менеджер підтримки, ІТ-фахівець та DevOps-інженер. Це дозволяє чітко простежити розподіл обов'язків і відповідальностей між ролями, що входять до структури служби підтримки ІТ-компанії.

Процес розпочинається з ініціації запиту клієнтом. Менеджер підтримки реєструє звернення, визначає його пріоритетність і аналізує суть проблеми. Якщо йдеться про помилку в коді, запит передається ІТ-фахівцю, який виконує відповідні дії для усунення несправності. У випадку проблем з інфраструктурою (наприклад, збій серверів або відмови сервісів), запит передається DevOps-інженеру, який перевіряє стан системи, усуває проблему та надсилає оновлення відповідному співробітнику.

Кожен з учасників процесу після завершення своїх дій передає результат далі – менеджеру, який оцінює інформацію та надає зворотній зв'язок клієнту. Клієнт, у свою чергу, перевіряє отримане рішення й у випадку задоволення – закриває звернення.

Діаграма демонструє важливу особливість організаційної структури ІТ-компанії – чіткий розподіл обов'язків між технічними й адміністративними ролями, а також важливість ефективної комунікації між ними. Така візуалізація дозволяє зрозуміти, як побудована логіка взаємодії в межах компанії, допомагає виявити вузькі місця у процесі обробки звернень та сприяє подальшій оптимізації організаційної структури.

#### Висновки та перспективи подальшого дослідження

У процесі дослідження було досягнуто основної мети, а саме – розроблено цілісну візуалізацію організаційної структури типової ІТ-компанії з використанням уніфікованої мови моделювання UML.

Створений набір моделей забезпечує формалізований, єдиний погляд на статичні та динамічні аспекти діяльності організації.

Вибір чотирьох ключових діаграм (випадків використання, класів, послідовності та діяльності) дозволить реалізувати підхід, що забезпечує повноцінне охоплення архітектури ІТ-компанії за принципом «4+1 View Model» (структурно-поведінковий підхід). Зокрема, цей набір охоплює статичні та динамічні аспекти: Діаграма класів представляє статичну структурну модель організації, тоді як діаграми діяльності та послідовності моделюють динамічну поведінкову модель, відображаючи бізнес-процеси та взаємодію між структурними елементами. Створені моделі дозволяють побудувати повноцінну цифрову репрезентацію діяльності ІТ-компанії, від структури відділів до деталізованого виконання бізнес-процесів (як-от управління проектами чи комунікація між відділами). Таким чином, розроблені UML-моделі слугують формалізованою основою для наступних етапів: аналізу вузьких місць, реінжинірингу бізнес-процесів та подальшої автоматизації.

Наукова новизна роботи полягає у розробці комплексної UML-моделі організаційної структури ІТ-компанії, що поєднує статичні, динамічні та функціональні аспекти в єдиній формалізованій архітектурі. Запропонований підхід забезпечує структурно-поведінкову цілісність представлення організаційної системи та створює основу для подальшого аналізу, оптимізації та автоматизації бізнес-процесів підприємства.

Обмеженням запропонованого підходу є його орієнтація на компанії малого та середнього розміру, а також залежність від якості початкового опису процесів і ролей. Для великих організацій із високою динамікою процесів модель потребує додаткової адаптації та деталізації.

Отримані результати мають важливе значення для практичного впровадження (слугуючи основою для автоматизації, оптимізації процесів та розробки інформаційних систем) та можуть бути ефективно використані в освітньому процесі для демонстрації студентам-айтівцям реальних корпоративних структур і стандартів документування.

Подальші дослідження будуть зосереджені на розширенні моделі за рахунок інтеграції діаграми станів (для деталізації життєвого циклу об'єктів/документів) та діаграми компонентів (для відображення архітектури програмного забезпечення, що підтримує організаційну структуру).

## Література

1. UML Diagrams in Software Engineering Research: A Systematic Literature Review // Proceedings. 2023. Vol. 74, No. 1, Article 13. MDPI. URL: <https://www.mdpi.com/2504-3900/74/1/13> (дата звернення:12.09.2025).
2. Enterprise modelling: Research review and outlook // ScienceDirect. 2021. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0166361520304991> (дата звернення:15.09.2025).
3. Rohman M. T., Kamisutara M., Mustajib M. Enterprise Architecture Planning at the Health Sciences Faculty of UMSurabaya Using the Zachman Framework. UMSurabaya, 2022. URL: <https://surl.li/ghfxza> (дата звернення:15.09.2025).
4. Borcosi C. A. The Importance of Business Modeling Using the Unified Modeling Language (UML). 2022. URL: <https://surl.li/iifhla> (дата звернення:22.09.2025).
5. Kruchten P. The 4+1 View Model of Architecture // IEEE Software. 1995. Vol. 12, Issue 6. P. 42–50.
6. Beauden J. An Analysis of the Significance of UML Diagrams: Focus on Medium-Sized Projects. 2023. URL: <https://surl.li/aamizy> (дата звернення:22.09.2025).
7. Tufegdžić M., Marić A. Integrating Business Processes to Enterprise Information System Using UML. 2021. URL: <https://surl.li/nxgntv> (дата звернення:24.09.2025).
8. Buriánek J. Comparison of Selected Enterprise Architecture Modeling Techniques from the Perspective of IT Services. CEUR Workshop Proceedings. 2022. Vol. 2991. P. 112–120. URL: <https://ceur-ws.org/Vol-2991/paper15.pdf> (дата звернення:25.09.2025).

## References

1. UML Diagrams in Software Engineering Research: A Systematic Literature Review // Proceedings. 2023. Vol. 74, No. 1, Article 13. MDPI. URL: <https://www.mdpi.com/2504-3900/74/1/13> (Data zvernennia: 12.09.2025).
2. Enterprise modelling: Research review and outlook // ScienceDirect. 2021. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0166361520304991> (Data zvernennia: 15.09.2025).
3. Rohman M. T., Kamisutara M., Mustajib M. Enterprise Architecture Planning at the Health Sciences Faculty of UMSurabaya Using the Zachman Framework. UMSurabaya, 2022. URL: <https://surl.li/ghfxza> (Data zvernennia: 15.09.2025).
4. Borcosi C. A. The Importance of Business Modeling Using the Unified Modeling Language (UML). 2022. URL: <https://surl.li/iifhla> (Data zvernennia: 22.09.2025).
5. Kruchten P. The 4+1 View Model of Architecture // IEEE Software. 1995. Vol. 12, Issue 6. P. 42–50.
6. Beauden J. An Analysis of the Significance of UML Diagrams: Focus on Medium-Sized Projects. 2023. URL: <https://surl.li/aamizy> (Data zvernennia: 22.09.2025).
7. Tufegdžić M., Marić A. Integrating Business Processes to Enterprise Information System Using UML. 2021. URL: <https://surl.li/nxgntv>(Data zvernennia: 24.09.2025).
8. Buriánek J. Comparison of Selected Enterprise Architecture Modeling Techniques from the Perspective of IT Services. CEUR Workshop Proceedings. 2022. Vol. 2991. P. 112–120. URL: <https://ceur-ws.org/Vol-2991/paper15.pdf>(Data zvernennia: 25.09.2025).