

DOI: <https://doi.org/10.36910/6775-2524-0560-2024-55-37>

УДК 004.4`2

Дейнеко Лев Миколайович, магістр

<https://orcid.org/0009-0000-7203-1883>

Кунанець Наталія Едуардівна, д.н.соц.комун., професор

<https://orcid.org/0000-0003-3007-2462>

Національний університет «Львівська політехніка», м. Львів, Україна

## РОЛЬ UML-ДІАГРАМ У ПЛАНУВАННІ ПРОЄКТУ ІНФОРМАЦІЙНОЇ СИСТЕМИ НА ПРИКЛАДІ СИСТЕМИ ПІШИХ ТУРИСТИЧНИХ МАРШРУТІВ

Дейнеко Л.М., Кунанець Н.Е. Роль UML-діаграм у плануванні проєкту інформаційної системи на прикладі системи піших туристичних маршрутів. Будь-який процес вимагає попереднього аналізу та планування. Сучасні інформаційні системи стають шоразу більш складними, а, відповідно, етап планування потребує додаткового часу та уваги. Одним з інструментів, що активно використовується фахівцями з розроблення IT-продукції, є UML - мова моделювання, що дозволяє описати та відобразити логіку системи. У цій статті розглянуто використання UML-діаграм на етапі планування та вплив даного інструменту на загальний результат розробленого рішення. Дане дослідження базується на прикладі системи піших туристичних маршрутів та розробці окремої реальної системи. Обґрунтовано важливість використання такого підходу в інженерії вимог, що дозволить прослідкувати залежність між наявністю моделей та якістю системи. У даній роботі проаналізовано не тільки переваги моделювання, а й виклики та складнощі, що можуть виникати під час аналізу нефункціональних вимог до системи. Це дослідження дозволило підсумувати роль UML-діаграм та окреслити основоположні принципи, що значною мірою покращать досвід використання такого підходу на різних етапах життєвого циклу інформаційних систем. Дана робота закладає підвалини для подальших досліджень та дозволить масштабувати вже здобуті результати для виявлення додаткових особливостей і деталей застосування моделювання. Варто підкреслити, що такі дослідження можуть стосуватися різних областей та аспектів UML, наприклад, у якості інструменту для пошуку недоліків чи покращення комунікації в команді спеціалістів загалом. Отримані результати можна застосувати у сфері тестування, бізнес-аналізу та під час розробки архітектури програмних рішень.

**Ключові слова:** UML; моделювання; діаграма; інформаційна система; дефект; планування.

Deineko L., Kunanets N. The role of UML diagrams in planning an information system project on the example of a hiking tourist routes system. Any process requires preliminary analysis and planning. Modern information systems are getting more and more complex, and, accordingly, the planning stage requires additional time and attention. One of the tools actively used by specialists in the development of IT products is UML – a modeling language that allows you to describe and display the logic of the system. This article outlines the use of UML diagrams at the planning stage and its impact on the overall result of the developed solution. This study is based on the example of a hiking tourist routes system and the development of a separate real solution. The importance of using such an approach in requirements engineering is substantiated, which will enable tracking the dependence between models' availability and the quality of the system. This paper analyzes not only the advantages of modeling but also the challenges and difficulties that may arise during the analysis of non-functional requirements for the system. This research made it possible to summarize the role of UML diagrams and outline the basic principles that will significantly improve the experience of using this approach at various stages of the information systems life cycle. This work lays the foundation for further research and will enable scaling the already obtained results to reveal additional features and details of the modeling application. It is worth emphasizing that such studies can pertain to different areas and aspects of UML, such as a tool for identifying the flaws or improving communication in a team of engineers in general. The obtained results can be applied in the field of testing, business analysis, and during the development of the architecture of software solutions.

**Keywords:** UML; modeling; diagram; information system; defect; planning.

**Постановка проблеми.** Інформаційні системи (ІС) стали невід'ємною складовою сучасного життя, проникаючи у різні сфери діяльності. Вони значно полегшують та видозмінюють велику кількість процесів – від електронної комерції до управління великими корпораціями, від медичних засобів та устаткування до автомобільних технологій [6]. Зростання складності ІС та потреби у якісному управлінні проєктами приводять до пошуку ефективних інструментів планування, аналізу та розробки ІС.

Серед таких інструментів можна виокремити «уніфіковану мову моделювання» (UML), що зарекомендувала себе як один з найефективніших та найпоширеніших способів моделювання різного роду систем. Даний засіб не просто став стандартом у світі розробки програмного забезпечення, а також є невід'ємною частиною інженерії вимог, їх аналізу та проєктування ІС [2].

Існує чимало розбіжностей щодо практичного застосування цієї мови та відмінностей, що накладаються іншими мовами графічного моделювання. Для спрощення ситуації можна виокремити 3 найпопулярніші сценарії застосування UML: підготовка ескізу, робота над проєктом та, власне, програмування. У першому випадку UML використовується для загальної візуалізації між деякими

компонентами системи. У другому – під час планової або зворотної розробки, тобто підготовка UML-діаграм перед початком написання коду, або ж навпаки – побудова діаграми на основі вже існуючого коду (системи) для кращого його розуміння [2].

Для більш ефективної відповіді на поточні виклики, ця робота базуватиметься на дослідженні та аналізі ролі UML-діаграм під час життєвого циклу програмного забезпечення на прикладі проєкту інформаційної системи піших туристичних маршрутів. Огляд практичного застосування, оцінка позитивних аспектів та виявлення можливих ризиків допоможуть відповісти на різноманітні питання, наприклад: «Як зрозуміти необхідний обсяг робіт та як вони вплинуть на імплементацію?», або «Які ризики можуть виникнути під час роботи над конкретною частиною застосунку?».

**Аналіз останніх досліджень та публікацій.** Моделювання за допомогою UML не є новий підходом у сфері розробки програмного забезпечення. Існує чимало досліджень, що детальніше розглядають застосування UML, особливості такого процесу та його переваги.

У своїй роботі С. Денисенко [8] вивчає проблеми, що виникають під час формалізації бізнес-процесу. Дослідження спирається на техніки IDEF та UML, де UML здобуває перевагу на етапах опрацювання складної та деталізованої інформації. Український професор та доктор технічних наук Є. Буров [7] запропонував інтелектуальну систему проєктування та керування бізнес-процесами. Це дослідження охоплює різні способи моделювання включно з UML та аналізує процес адаптації систем задля зменшення витрат на супровід. У науковій публікації В. Яковина та Ю. Парфенюк [9] проаналізували UML як засіб прогнозування надійності програмного забезпечення, а також розробили підхід для визначення надійності на етапі описання архітектури. Запропонований метод дозволяє оцінити ймовірність сценаріїв використання програмної системи.

На основі проведеного аналізу можна зробити висновок, що сучасні дослідження підкреслюють різні аспекти моделювання та його застосування у багатьох сферах. Однак, досі є велика потреба у ґрунтовних дослідженнях, які б прослідкували роль моделювання за допомогою UML на етапі планування проєкту та порівняли отримані показники з кінцевими результатами впровадження.

**Обґрунтування важливості UML-діаграм у плануванні проєктів.** «Планування» є першим етапом розробки у розрізі життєвого циклу продукту. Відповідно, планування визначає якість виконаного продукту і є фундаментальним етапом у розробленні ПЗ. Ефективно виконане планування дозволяє виявити й усунути можливі ризики та проблеми на ранніх етапах розроблення, сприяючи успішному завершенню проєкту в обумовлений термін та бюджет. Недоліки у плануванні можуть призвести до затримок, перевищення бюджету (через залучення додаткових ресурсів) та низької якості продукту.

Таке твердження можна аргументувати дослідженням у роботі Хатчінсона [3], який стверджував, що перші 20% часу на проєкті присвячуються розробці моделей. У такому випадку обсяги моделювання можна відобразити наступним способом (Рис. 1):

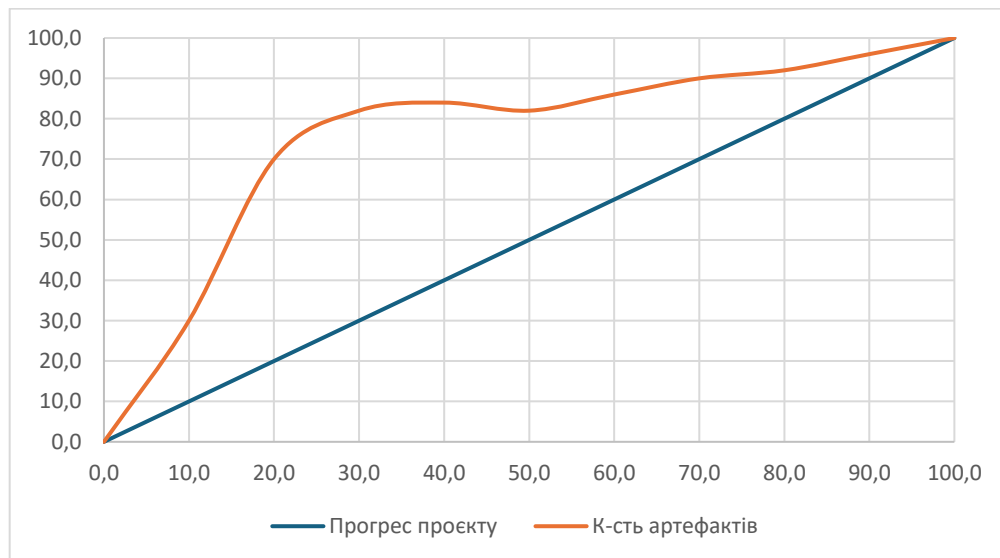


Рис. 1. Обсяги артефактів під час імплементації системи

Граничні значення прогресу та кількості артефактів прийняті за 100%. Оскільки переважна більшість моделювання відбувається на ранніх етапах, то в подальшому змін дуже мало, що свідчить про те, що впродовж цього періоду робота полягає переважно у внесенні правок до існуючих діаграм. Отже, графік можна інтерпретувати наступним чином: зазвичай, проєкт розпочинається з визначення загального бачення продукту, і тільки згодом додаються деталі та незначні зміни.

Для кращого розуміння такого підходу необхідно проаналізувати фактичний стан інформаційної системи, що розроблялася з червня 2023 року до квітня 2024 (Рис. 2). У даній ситуації не було надано належної уваги використанню UML-моделювання, відповідно, ризик недоліків значно більший і це підтверджує діаграма на Рисунку 2. Незважаючи на спроби команди виправити недоліки, загальна кількість неточностей в системі невпинно зростає і обсяг технічного боргу збільшується. Проаналізувавши обидві діаграми (Рис.1 та Рис. 2), цілком доцільним буде припущення, що брак належного моделювання на етапі планування погіршує загальну якість продукту, що, у свою чергу, впливає на подальшу розробку та супутні витрати. Отже, нижча якість розроблених моделей також негативно впливає на якість коду.

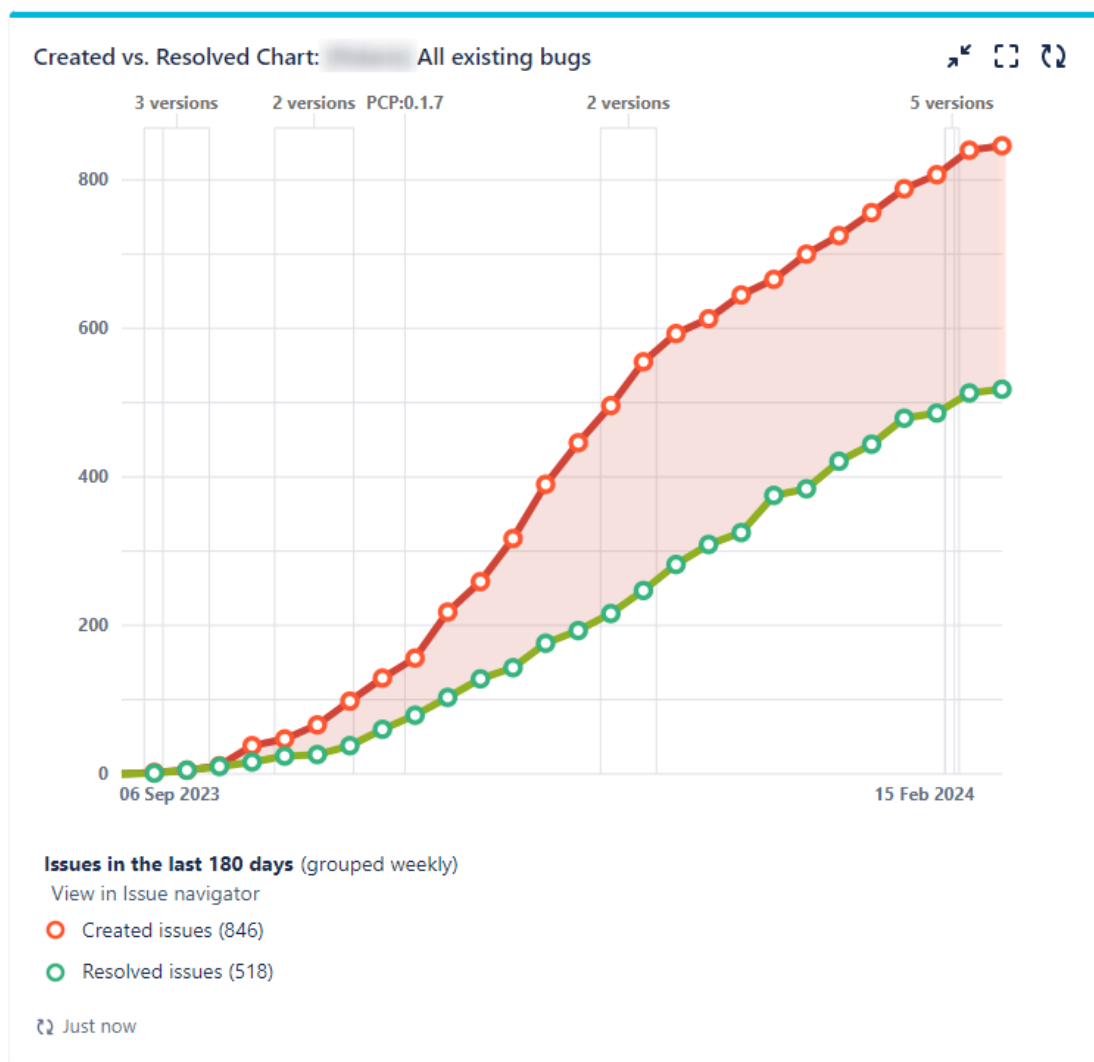


Рис. 2. Співвідношення загальної кількості дефектів та вирішених недоліків

Враховуючи вищесказане, розглянемо приклад діаграми використання інформаційної системи піших туристичних маршрутів (Рис. 3).

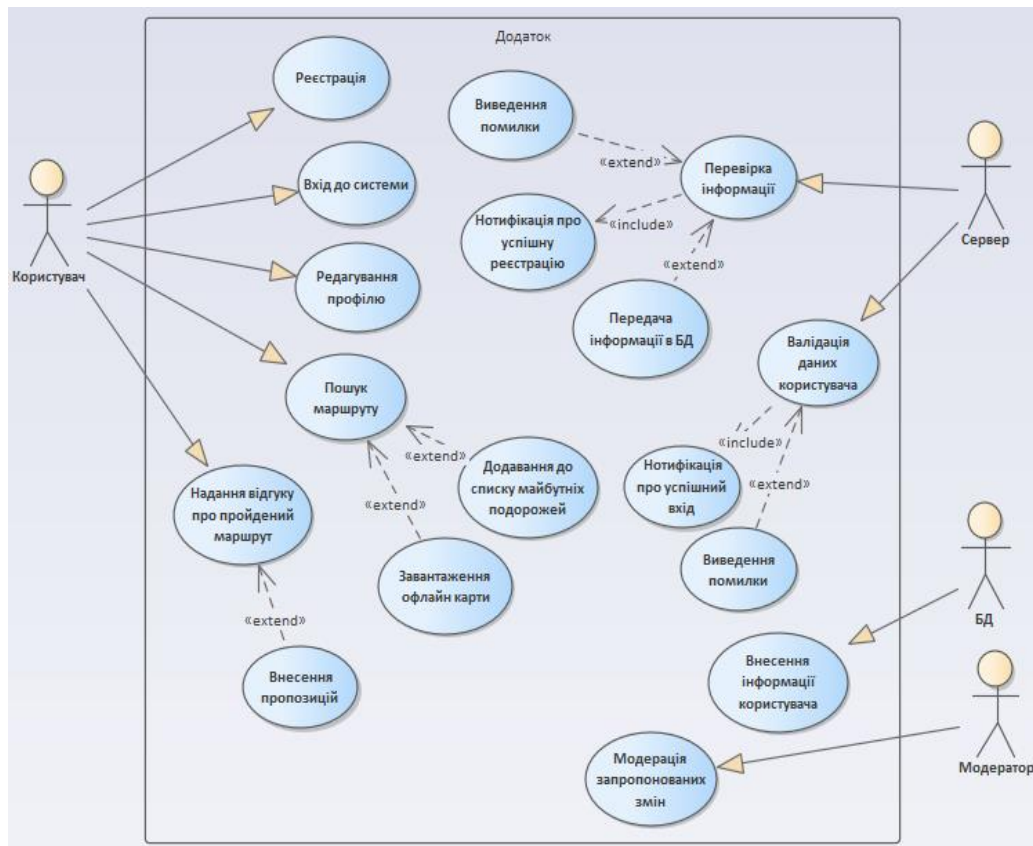


Рис. 3. Діаграма використання інформаційної системи піших туристичних маршрутів

Діаграма використання ІС піших туристичних маршрутів є одним з ключових інструментів для подальшого розроблення: вона ідентифікує основних акторів, відображає їх головні взаємодії через варіанти використання (use cases). Тобто, якщо було отримано даний артефакт на початку проєкту – це дозволить використати діаграму для ідентифікації функціональних вимог, визначити основні сценарії, що стане базою для подальшого аналізу, планування розробки та потенційного зменшення неточностей.

Наступні кроки повинні враховувати дуже важливий аспект, визначення на якому етапі розроблених моделей буде достатньо. Це вимагатиме певного компромісу щодо зусиль, які замовник готовий вкласти у високоякісні і деталізовані артефакти та додаткового ресурсу, що необхідний для виправлення неточностей у моделях. Вартість змін продемонстровано на графіку (Рис. 4). Відтак, можна зробити висновок, що економічно вигідно виправляти будь-які недоліки якомога швидше. Типовими видами недоліків у моделях UML є невідповідності, неповнота та навіть дублювання [4]. Якщо враховувати, що однією з головних цілей UML-діаграм є донесення спільного та однозначного контексту до усіх членів команди – це доволі серйозний показник того, що однозначність та завершеність моделей підлягає постійному контролю.

Для дотримання практики збалансування розподілу зусиль між якістю моделювання та зниженням ризиків необхідно впроваджувати належний рівень тестування, що сприятиме правильній витраті ресурсів. Наприклад, критичні і складніші компоненти потребуватимуть детальнішого тестування, а простіші – переважно поверхневого. Це вимагає правильної класифікації та розрізнення критичних елементів системи, що більше належить до площини тестування і належної комунікації між командою та бізнесом, де часто постає питання правильного виваження знайдених недоліків. Тобто, моделювання напряму впливатиме на хід роботи усього проєкту, а правильний розподіл зусиль допоможе визначити належний рівень кількості артефактів та їх якість.

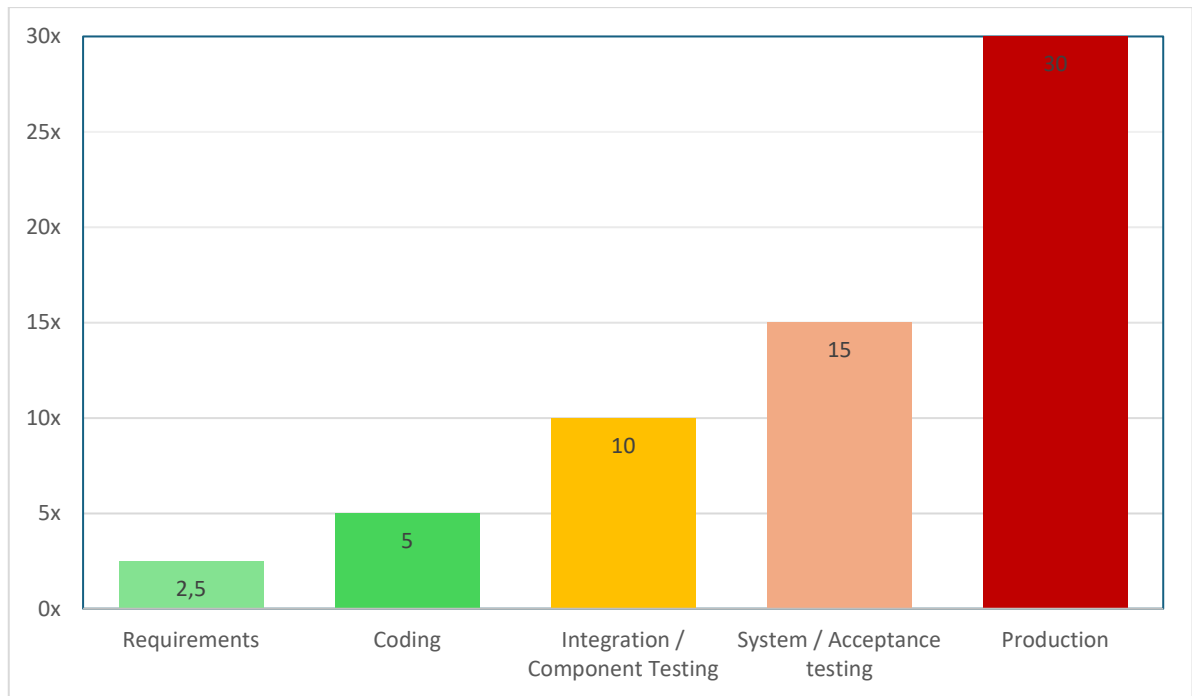


Рис. 4. Ріст вартості змін під час життєвого циклу розробки ПЗ

У книзі С. Макконелла наведено порівняльну характеристику різних засобів, що сприяють виявленню дефектів [5]. З точки зору тестування ПЗ, моделювання можна трактувати як один з можливих та ефективних способів пошуку неточностей під час реалізації життєвого циклу системи (Рис. 5).

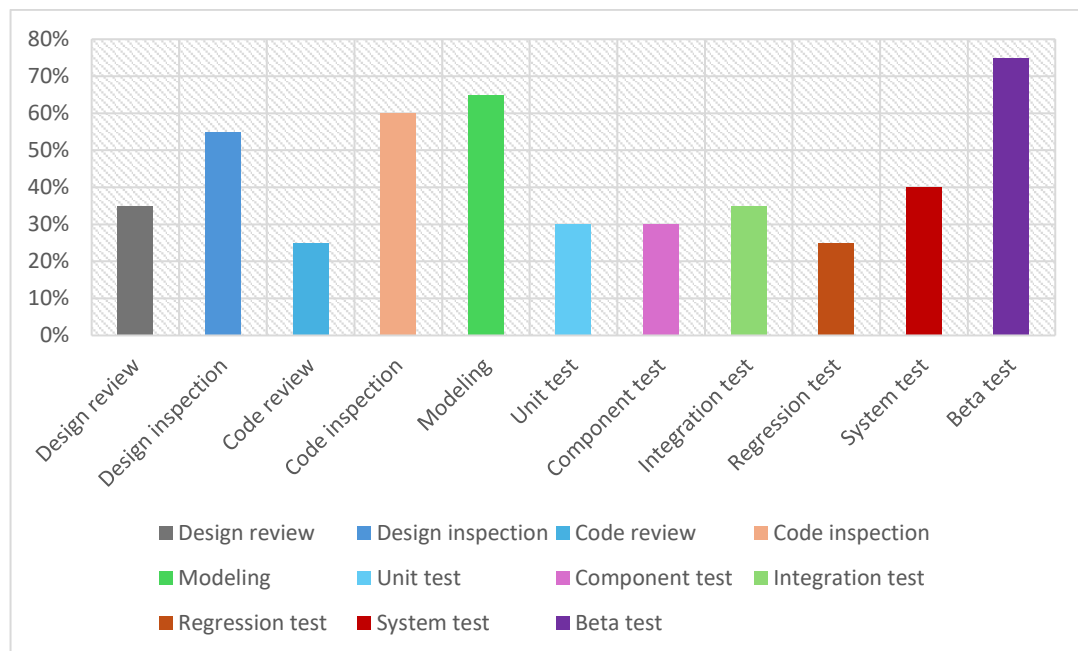


Рис. 5. Ефективність методів пошуку дефектів

Оскільки моделі розробляються переважно перед початком імплементації, їх можна використати для ґрунтовного оцінювання дизайну системи, а не чекати завершення етапу розробки, коли дефекти почнуть проявлятися під час модульного та інтеграційного тестування.

Окрім переваг моделювання, доцільно також висвітлити і деякі його недоліки, що можуть по-різному впливати на процес впровадження та на продукт загалом. Повертаючись до раніше зазначеної діаграми використання інформаційної системи піших туристичних маршрутів (Рис. 3)

можна припустити, що така модель не враховує мотивацію та досвід потенційних кінцевих користувачів. Тобто, розроблений продукт може цілком і повністю відповідати усім вимогам, але не гарантовано, що цей продукт задовольнить кінцевого користувача з погляду валідації. Також варто розглянути діаграму активності даної ІС (Рис. 6). У цьому випадку додаток розроблявся з урахуванням наступних можливостей:

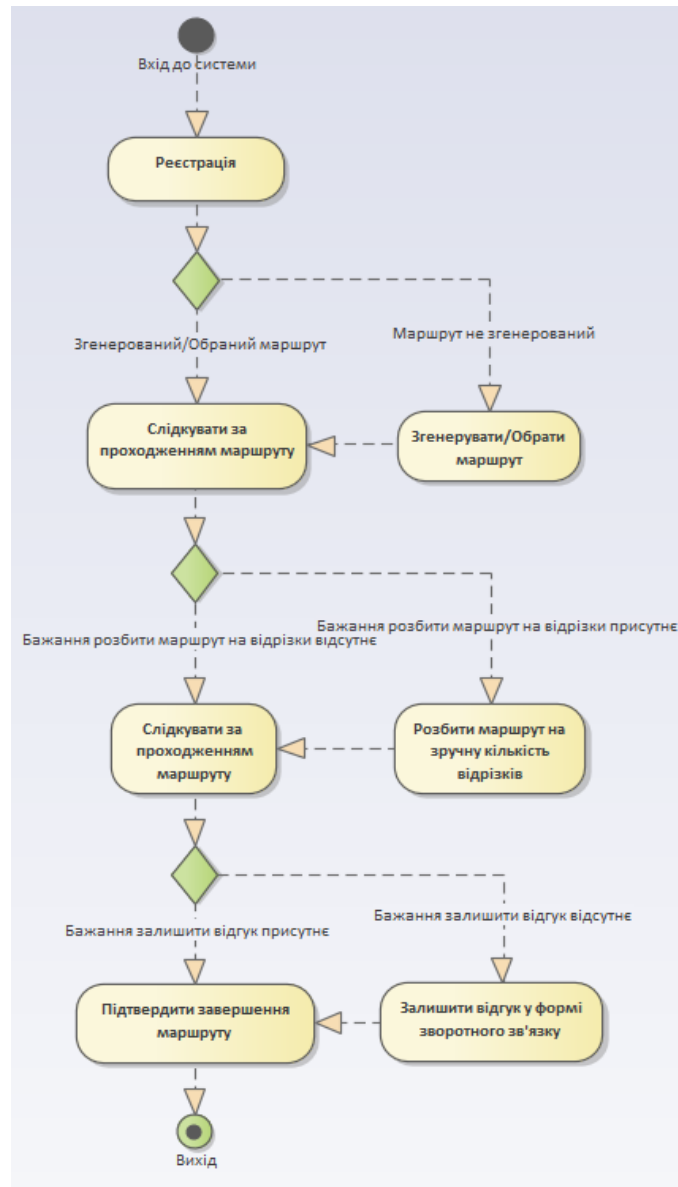


Рис. 6. Діаграма активності системи

Користувач:

- заходить в додаток та реєструється/вводить дані наявного профілю;
- генерує необхідний маршрут;
- у разі потреби розділяє маршрут на декілька етапів/відрізків;
- при бажанні залишає відгук стосовно згенерованого/пройденого маршруту.

Система:

- опрацьовує дані зареєстрованого/наявного користувача;
- відображає згенерований маршрут/можливі альтернативні шляхи;
- відображає стан проходження обраного маршруту;
- приймає відгук користувача через форму зворотного зв'язку.

У певній мірі це дає загальне розуміння обсягу робіт та можливих способів досягнення результату, але на даному етапі розробки немає впевненості, що це остаточна версія, котра не потребуватиме подальших покращень опісля досліджень із користувачами.

До цього списку також входить і зручність використання. Не існує систематизованого та узагальненого підходу щодо покриття нефункціональних вимог у контексті діаграм використання. Діаграми класів переважно використовуються для відображення системи в статистиці. Однак, часом необхідно проаналізувати і динамічні аспекти роботи системи, що, вочевидь, вимагатиме додаткових зусиль. Діаграми послідовності відображають переважно один загальний сценарій, відповідно, у випадках зі складною логікою системи необхідно збирати різні діаграми даного типу разом, щоб відобразити різні можливі варіанти поведінки. Це означає, що інтеграція сценаріїв є однією з критичних проблем у використанні моделювання цього типу. Діаграми послідовності потребують об'єктів у вигляді вхідних даних та станів об'єктів – як вихідних, а це означає, що діаграма класів впливатиме на процес створення такої моделі.

На основі проведеного дослідження можна виокремити декілька важливих принципів у моделюванні. По-перше, «вибір моделі має фундаментальний вплив на формування вирішення задачі» [1]. Тобто, вкрай важливо правильно обрати модель – це підкреслить проблемні місця та рішення, що можна не помітити за інших умов. Ретельне та повторюване тестування моделей дозволить досягнути значно вищого рівня впевненості у тому, що змодельоване рішення поводитиметься максимально близько до очікуваного результату. У такому випадку більшість нестандартних та неочевидних випадків стане помітною для всіх членів команди, що дозволить провести додаткові обговорення та запланувати необхідні дії. Слід відзначити, що спеціальність колег також впливатиме на вихідний результат обраної моделі у розробці ПЗ, адже кожен член команди матиме власне бачення, що базуватиметься на персональному досвіді та кваліфікації. Проте, за умов правильно організованої співпраці це буде радше перевагою, ніж недоліком.

По-друге, «будь-яка діаграма може мати різні рівні трактування» [1]. У будь-якому випадку найкращим вибором буде той, що дозволить застосувати необхідний ступінь деталізації, оскільки аналітики, кінцеві користувачі чи розробники цікавитимуться певними аспектами інформаційної системи у різний час.

По-третє, «найкращим варіантом є той, що найбільш близький до реальності» [1]. Якщо модель тільки частково передає справжній стан речей – користі з неї дуже мало. Найкраще обирати моделі, що максимально передають реальний хід подій, а також шукати конкретні чинники, що негативно впливають на це. Спрощення реальності є хорошим фактором, але тільки за умови, що такі спрощення не приховують жодних важливих деталей. Нездатність подолати такий аспект призводить до того, що з часом фактична реалізація системи все більше відрізнятиметься від першопочаткового задуму.

По-четверте, «жодна діаграма не може вичерпно покрити систему. Краще для будь-якої нетривіальної системи розробляти невеликий набір майже незалежних моделей» [1]. В практичному застосуванні ідея полягає у тому, щоб наявні діаграми залишалися взаємопов'язаними, але водночас дозволяли достатньо детально вивчити конкретну сферу. Певною мірою це альтернативне рішення діаграм декомпозиції, що розбивають складні компоненти на менші та зрозуміліші.

**Висновки та перспективи подальшого дослідження.** Проведений аналіз засвідчує та обґрунтовує важливість використання UML-діаграм як одного з інструментів моделювання та документування, що посідає чільне місце у роботі над проектами та розробкою рішень. Додаткові приклади базуються на інформаційній системі піших туристичних маршрутів. Результати дослідження окреслюють значення та роль UML в інженерії вимог на різних етапах життєвого циклу. Детальне вивчення результатів впровадження системи без активного використання UML-моделювання засвідчує значну кількість недоліків, що негативно впливатиме на супутні витрати.

Окреслено принципи моделювання, що слугують коротким і водночас зрозумілим посібником з успішного застосування моделювання.

Ця робота дає змогу покращити вже набуті знання та закладає підвалини для нового аналізу сфери моделювання у контексті роботи над інформаційними системами.

Напрямок подальших досліджень є масштабування вже здобутих результатів, розширення вибірки проектів та інформаційних систем, що допоможе виявити додаткові особливості використання моделювання під час розроблення програмних рішень.

**Список бібліографічного опису**

1. Booch, G., Rumbaugh, J.E., & Jacobson, I. (2005). *The Unified Modeling Language User Guide*. Addison-Wesley.
2. Fowler, M. (2004). *UML distilled: A brief guide to the standard object modeling language*. Addison-Wesley.
3. Hutchinson, J., Whittle, J., Rouncefield, M., & Kristoffersen, S. (2011). Empirical assessment of MDE in industry. In *Proceedings of the 33rd International Conference on Software Engineering* (pp. 471-480). New York, NY; ACM.
4. Lange, C., Chaudron, M.R.V., Muskens, J., Somers, L.J., & Dortmans, H.M. (2003). An empirical investigation in quantifying inconsistency and incompleteness of UML designs. In *Proceedings of the Workshop on Consistency Problems in UML-based Software Development* (pp. 26-34). San Francisco, CA; Blekinge Institute of Technology.
5. McConnell, S. (2004). *Code Complete: A practical handbook of software construction* (2nd ed.). Microsoft Press.
6. Rainer, R. K., & Prince, B. (2019). *Introduction to information systems: Supporting and Transforming Business* (8th ed.). John Wiley & Sons, Inc.
7. Буров, С. (2008). Система моделювання інтелектуальної мережі бізнес-процесів. *Вісник Національного університету «Львівська політехніка»*, 610, 34 - 39.
8. Денисенко, С. (2010). Проблемні моменти моделювання бізнес-процесів з метою проведення подальшого реінжинірингу на базі стандартів IDEF та UML-моделювання. *Вісник Національного університету «Львівська політехніка»* 691, 41 - 46.
9. Яковина, В., Парфенюк, Ю. (2013). Використання засобів UML для прогнозування надійності програмного забезпечення на етапі його проектування. *Вісник Національного університету «Львівська політехніка»* 773, 151 - 156.

**References**

1. Booch, G., Rumbaugh, J.E., & Jacobson, I. (2005). *The Unified Modeling Language User Guide*. Addison-Wesley.
2. Fowler, M. (2004). *UML distilled: A brief guide to the standard object modeling language*. Addison-Wesley.
3. Hutchinson, J., Whittle, J., Rouncefield, M., & Kristoffersen, S. (2011). Empirical assessment of MDE in industry. In *Proceedings of the 33rd International Conference on Software Engineering* (pp. 471-480). New York, NY; ACM.
4. Lange, C., Chaudron, M.R.V., Muskens, J., Somers, L.J., & Dortmans, H.M. (2003). An empirical investigation in quantifying inconsistency and incompleteness of UML designs. In *Proceedings of the Workshop on Consistency Problems in UML-based Software Development* (pp. 26-34). San Francisco, CA; Blekinge Institute of Technology.
5. McConnell, S. (2004). *Code Complete: A practical handbook of software construction* (2nd ed.). Microsoft Press.
6. Rainer, R. K., & Prince, B. (2019). *Introduction to information systems: Supporting and Transforming Business* (8th ed.). John Wiley & Sons, Inc.
7. Burov, Ye. (2008). Systema modeliuвання intelektualnoi merezhi biznes-protsesiv. *Visnyk Natsionalnoho universytetu «Lvivska politekhnika»*, 610, 34 - 39.
8. Denysenko, S. (2010). Problemni momenty modeliuвання biznes-protsesiv z metoiu provedennia podalshoho reinzhynirynhu na bazi standartiv IDEF ta UML-modeliuвання. *Visnyk Natsionalnoho universytetu «Lvivska politekhnika»* 691, 41 - 46.
9. Yakovyna, V., Parfeniuk, Yu. (2013). Vykorystannia zasobiv UML dlia prohnouzuvannia nadiinosti prohrannoho zabezpechennia na etapi yoho proektuvannia. *Visnyk Natsionalnoho universytetu «Lvivska politekhnika»* 773, 151 - 156.