

# РОЛЬ ЯКОСТІ АНАЛІЗУ ВИМОГ У ЗАБЕЗПЕЧЕННІ УСПІШНОСТІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Вінницький національний технічний університет

## Анотація

*У роботі досліджується роль якості аналізу вимог у процесі розробки програмного забезпечення. Розглянуто основні підходи до збору, формалізації та перевірки вимог, а також їх вплив на ефективність розробки та якість кінцевого продукту. Проаналізовано типові проблеми, що виникають у процесі аналізу вимог, та визначено їх вплив на ризики розробки. Встановлено, що якісний аналіз вимог є ключовим фактором успішності проекту та зниження витрат ресурсів.*

**Ключові слова:** вимоги до програмного забезпечення, аналіз вимог, якість вимог, програмне забезпечення, розробка ПЗ, управління вимогами.

## Abstract

*The paper examines the role of requirements analysis quality in the software development process. The main approaches to requirements elicitation, formalization, and validation are considered, as well as their impact on development efficiency and product quality. Typical problems in requirements analysis are analyzed and their influence on development risks is identified. The study shows that high-quality requirements analysis is a key factor in project success and resource optimization.*

**Keywords:** software requirements, requirements analysis, requirements quality, software development, requirements management.

## Вступ

У сучасному середовищі розробки програмного забезпечення якість кінцевого продукту значною мірою залежить від правильності формування вимог. Аналіз вимог є одним із перших і найважливіших етапів життєвого циклу програмного забезпечення, оскільки саме на цьому етапі визначаються функціональні та нефункціональні характеристики системи.

Недостатньо якісний аналіз вимог може призвести до помилок у проектуванні, перевитрат ресурсів, затримок у розробці та невідповідності продукту очікуванням користувачів. У той же час чітко сформульовані та узгоджені вимоги забезпечують ефективну організацію процесу розробки та зменшують ризики.

Метою даної роботи аналіз значення якості формування вимог та її впливу на результати розробки програмного забезпечення.

## Результати дослідження

Якість аналізу вимог безпосередньо визначає подальший хід розробки програмного забезпечення, оскільки саме на цьому етапі формується розуміння того, якою має бути майбутня система [1]. Якщо вимоги сформульовані неточно або поверхнево, це призводить до накопичення помилок, які проявляються вже на етапах реалізації та тестування. Як наслідок, виникає потреба у повторному виконанні робіт, що значно збільшує витрати часу та ресурсів.

Практика розробки програмного забезпечення показує, що найбільш критичними є проблеми, пов'язані з нечіткістю та неоднозначністю вимог. Наприклад, одна й та сама вимога може бути по-різному інтерпретована аналітиком і розробником, що призводить до створення функціоналу, який не відповідає очікуванням замовника. У таких випадках навіть правильно реалізований код не вирішує поставлену задачу, що негативно впливає на загальний результат проекту [2].

Не менш важливою є повнота вимог. У процесі збору інформації часто залишаються неврахованими окремі сценарії використання системи або нефункціональні характеристики, такі як продуктивність, безпека чи зручність використання. Це призводить до того, що вже після впровадження системи виникає необхідність її доопрацювання. Такі зміни є значно дорожчими, ніж їх врахування на початковому етапі.

Суттєвий вплив на якість вимог має організація процесу їх збору. Використання лише одного методу, наприклад інтерв'ю, не дозволяє отримати повну картину потреб користувачів. Ефективнішим є комбінування різних підходів: аналіз існуючих процесів, спостереження за роботою користувачів, проведення опитувань та обговорень. Це дає змогу виявити як явні, так і приховані вимоги, які часто залишаються поза увагою.

Окремої уваги заслуговує питання взаємодії між учасниками проекту. У багатьох випадках проблеми виникають не через відсутність інформації, а через її неправильну передачу. Використання складних або неоднозначних формулювань, відсутність єдиної термінології та недостатнє обговорення вимог призводять до помилок, які складно виявити на ранніх етапах [3]. Тому важливо забезпечити регулярну комунікацію та узгодження рішень між усіма учасниками процесу.

Для підвищення якості вимог широко застосовується їх формалізація. На практиці це означає використання структурованих описів, моделей і діаграм, які дозволяють чітко представити функціональність системи. Наприклад, use-case діаграми допомагають описати взаємодію користувача із системою, а UML-діаграми — відобразити її структуру. Такий підхід значно спрощує розуміння вимог і зменшує ймовірність помилок.

Важливим інструментом є також перевірка вимог ще до початку розробки. Це може здійснюватися у формі рецензування, обговорень або тестових сценаріїв. Така перевірка дозволяє виявити неточності, пропущені елементи або суперечності. В результаті зменшується кількість змін у процесі реалізації, що позитивно впливає на строки виконання проекту.

Сучасні підходи до розробки програмного забезпечення передбачають постійну роботу з вимогами. Зокрема, у гнучких методологіях вони уточнюються поступово, у процесі виконання проекту [4]. Це дозволяє враховувати зміни у потребах користувачів та адаптувати систему до нових умов. Водночас такий підхід вимагає чіткої організації процесу, оскільки відсутність контролю може призвести до хаотичних змін і втрати цілісності системи.

Необхідно також враховувати, що вимоги мають бути перевірюваними. Кожна вимога повинна мати критерії, за якими можна визначити, чи реалізована вона правильно. Це особливо важливо на етапі тестування, коли потрібно оцінити відповідність системи поставленим задачам. Відсутність таких критеріїв ускладнює перевірку і може призвести до суб'єктивної оцінки якості продукту.

Ще одним важливим аспектом є використання спеціалізованих інструментів для управління вимогами. Вони дозволяють зберігати вимоги, відстежувати їх зміни та встановлювати зв'язки між різними елементами системи. Це особливо актуально для великих проєктів, де кількість вимог може бути значною, а їх структура — складною.

Таким чином, проведений аналіз показує, що якість вимог визначає не лише технічні характеристики системи, але й ефективність усього процесу розробки. Чіткі, узгоджені та структуровані вимоги дозволяють уникнути значної частини проблем, зменшити витрати ресурсів і забезпечити створення програмного продукту, який відповідає очікуванням користувачів [5].

Ще одним важливим аспектом є простежуваність вимог (traceability). Вона дозволяє встановити зв'язок між вимогами, елементами системи та результатами їх реалізації. Завдяки цьому можна відстежити, як кожна вимога впливає на окремі компоненти програмного забезпечення, а також швидко визначити, які частини системи потребують змін у разі оновлення вимог. Відсутність простежуваності ускладнює процес управління проєктом і може призвести до втрати важливої інформації.

Крім того, значну роль відіграє пріоритизація вимог. У реальних умовах ресурси проєкту обмежені, тому неможливо одночасно реалізувати всі функції з однаковою увагою. Визначення пріоритетів дозволяє зосередитися на найбільш важливих вимогах, які мають найбільший вплив на користувачів або бізнес-процеси. Це сприяє більш ефективному плануванню розробки та забезпечує швидке отримання цінності від програмного продукту.

Також слід враховувати вплив якості вимог на тестування програмного забезпечення. Чітко сформульовані вимоги значно спрощують розробку тестових сценаріїв і дозволяють об'єктивно оцінити відповідність системи очікуваним результатам. У випадку нечітких або неповних вимог процес тестування ускладнюється, оскільки відсутні конкретні критерії перевірки. Це може призвести до пропуску помилок і зниження загальної якості програмного продукту.

Окремо варто звернути увагу на вплив якості вимог на процес підтримки та розвитку програмного забезпечення після його впровадження. Якщо вимоги були сформульовані чітко і задокументовані належним чином, це значно спрощує подальшу модифікацію системи. Розробники можуть швидше зрозуміти логіку роботи функціоналу та внести необхідні зміни без ризику порушення цілісності системи. У протилежному випадку відсутність якісної документації ускладнює супровід і призводить до накопичення технічного боргу.

Крім того, якісний аналіз вимог сприяє підвищенню загальної передбачуваності процесу розробки. Чітко визначені вимоги дозволяють більш точно оцінити обсяг робіт, строки виконання та необхідні ресурси. Це є важливим для планування проєкту та контролю його виконання. У результаті зменшується кількість непередбачених ситуацій, підвищується стабільність процесу розробки та забезпечується більш ефективно досягнення поставлених цілей.

## Висновок

Таким чином, якість аналізу вимог відіграє визначальну роль у забезпеченні успішності розробки програмного забезпечення. Саме на цьому етапі формується основа майбутньої системи, від якої залежить правильність реалізації функціоналу, ефективність використання ресурсів та відповідність продукту потребам користувачів.

Проведений аналіз показує, що нечіткі, неповні або суперечливі вимоги призводять до значних ускладнень у процесі розробки, зокрема до збільшення витрат часу, необхідності повторного виконання робіт та зниження якості кінцевого продукту. Натомість якісно сформульовані вимоги забезпечують узгодженість дій усіх учасників проєкту, зменшують ризики та сприяють ефективному досягненню поставлених цілей.

Отже, забезпечення високої якості аналізу вимог є необхідною умовою успішної реалізації програмних проєктів. Використання сучасних методів збору, формалізації, перевірки та управління вимогами дозволяє підвищити ефективність процесу розробки, покращити якість програмного забезпечення та забезпечити його відповідність очікуванням користувачів.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Sommerville I. Software Engineering. 10th ed. Boston: Pearson, 2020. 816 p.
2. Pressman R. S., Maxim B. R. Software Engineering: A Practitioner's Approach. 9th ed. New York: McGraw-Hill, 2020. 970 p.
3. Wiegers K., Beatty J. Software Requirements. 3rd ed. Redmond: Microsoft Press, 2021. 672 p.
4. Agile Alliance. Manifesto for Agile Software Development. [Електронний ресурс]. Режим доступу: <https://agilemanifesto.org/> (дата звернення: 19.03.2026).
5. Морзе Н. В., Кочарян А. Б. Інформаційні системи та технології. К.: Університет «КРОК», 2021. 320 с.

**Автор:** *Гусятинська Яна Сергіївна* – студентка групи 5ПІ-22б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: gusatinska.yana@gmail.com.

**Науковий керівник:** *Ліщинська Людмила Броніславівна* – д-р техн. наук, професор, професор кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: llb@vntu.edu.ua

**Author:** *Husiatynska Yana* – student of the 5PI-22b group, faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: gusatinska.yana@gmail.com.

**Scientific Supervisor:** *Lishchynska Lyudmyla Bronislavivna* – Dr. Sc. (Eng.), Full Professor, Professor of Program Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: llb@vntu.edu.ua