

DOI: <https://doi.org/10.32782/2524-0072/2025-73-57>

УДК 004.413

УПРАВЛІННЯ НЕФУНКЦІОНАЛЬНИМИ ВИМОГАМИ ДО ПРОГРАМНИХ ПРОДУКТІВ У СФЕРІ ФІНАНСОВИХ ТЕХНОЛОГІЙ

MANAGEMENT OF NON-FUNCTIONAL REQUIREMENTS FOR SOFTWARE PRODUCTS IN THE FIELD OF FINANCIAL TECHNOLOGIES

Омельченко Дмитро Вікторович

аспірант,

Приазовський державний технічний університет

ORCID: <https://orcid.org/0009-0005-0420-1808>

Omelchenko Dmytro

Pryazovskyi State Technical University

Підходи до управління розробкою програмних продуктів залежать від багатьох аспектів. Важливе значення має сфера, для якої розробляються програмні продукти. У процесі розробки програмних продуктів важливим етапом є визначення вимог. В рамках статті розглянуто аспекти визначення нефункціональних вимог до програмних продуктів у сфері фінансових технологій. Відзначено, що програмні продукти у цій сфері є системами зі складною архітектурою і логікою. Відзначено, що окрім програмного коду продукту важливе значення має інфраструктура, необхідна для коректної роботи продукту. Визначено перелік категорій нефункціональних вимог до програмних продуктів у сфері фінансових технологій. Визначено ключові аспекти і вплив перелічених вимог на етапи розробки, розгортання та експлуатації програмних продуктів. Зроблено загальний висновок, що нефункціональні вимоги мають не менш важливе значення ніж функціональні вимоги.

Ключові слова: програмний продукт, фінансові технології, функціональні вимоги, нефункціональні вимоги, розробка, розгортання, експлуатація, документація.

Approaches to software development management depend on many aspects. The area for which software products are developed is of great importance. In the process of developing software products, an important stage is defining requirements. The article considers aspects of defining non-functional requirements for software products in the field of financial technologies. The methods of classification, generalization and comparison were used in the study. It is noted that software products in the field of financial technologies are systems with a complex architecture and logic. A list of categories of non-functional requirements for software products in the field of financial technologies is determined. It is noted that when developing software products in this area, it is important to take into account requirements related to performance, fault tolerance, data security, compliance with legal requirements, documentation, as well as an approach to the deployment of software products. The influence of the listed requirements on the stages of development, deployment and operation of software products is determined. It is noted that when using flexible iterative approaches to the development of software products, it is important to monitor the performance indicators of the system after each iteration in order to promptly identify signs of its degradation. It has been established that in addition to the product's program code, the infrastructure required for the correct operation of the product is of great importance. International regulatory documents governing data and payment security issues are provided, and their key aspects that should be taken into account when developing software products in the field of financial technologies are highlighted. It is noted that when documenting software products, the description of the API (Application Programming Interface), which is used for integration with external systems, is of great importance. It has been established that in order to accelerate the development of API, integration with API and preparation of API documentation, it is advisable to use tools that allow the automation of these actions. Comparisons of the main approaches to the deployment of software products are performed and criteria for their selection are defined. A general conclusion is made that non-functional requirements are no less important than functional requirements.

Keywords: software product, financial technologies, functional requirements, non-functional requirements, development, deployment, operation, documentation.

Постановка проблеми. Визначення та оформлення вимог є невід’ємним етапом розробки будь-якого програмного продукту. Ігнорування або неякісне виконання цього етапу може призвести до невідповідності розробленого продукту очікуванням кінцевих користувачів, а також збільшення часових і фінансових витрат, необхідних для розробки продукту, оскільки відсутність або неузгодженість вимог до продукту може призвести до необхідності повторної розробки частини продукту або продукту в цілому. Згідно результатів опитування серед керівників бізнесу і технічних команд [1] відсутність ясності або узгодженості щодо бізнес-результатів є однією з найбільш критичних причин невдач проектів розробки програмного забезпечення.

Аналіз останніх досліджень і публікацій. Актуальність теми управління вимогами до програмних продуктів, а також теми розвитку і впровадження фінансових технологій підтверджується наявністю існуючих досліджень. Тему управління вимогами до програмних продуктів досліджували Коваленко О. Є., Смолій В. М. і Лі Л. [2], Корнієнко О. О. і Крупа Н. О. [3], Ольховська О.О., Парфьонова Т. О. і Бражніченко А. О. [4], Храпкін О. М., Кіндрат О. В. і Чопей Р. С. [5] та інші. Тему розвитку і впровадження фінансових технологій досліджували Микитюк П. і Микитюк В. [6], Перебийніс Д. С. [7] та інші.

Виділення невирішених раніше частин загальної проблеми. Підхід до визначення та оформлення вимог до програмних продуктів залежить від багатьох аспектів. Важливе значення має сфера, для якої виконується розробка програмних продуктів. Особливу увагу варто звернути на сферу фінансових технологій, оскільки дана сфера активно розвивається і передбачає значний вплив на розвиток фінансових послуг [6-7]. Важливим аспектом розробки програмних продуктів у даній сфері є реалізація не лише функціональних, але й нефункціональних вимог, оскільки від їхньої реалізації будуть залежати експлуатаційні характеристики продукту, відповідність продукту очікуванням користувачів і, як наслідок, успіх продукту в цілому, тому є доцільними подальші дослідження у даному напрямку.

Метою статті є визначення переліку категорій нефункціональних вимог до програмних продуктів у сфері фінансових технологій, а також їхнього впливу на процес розробки, розгортання та експлуатації програмних продуктів.

Виклад основного матеріалу дослідження. Вимоги до будь-якого програмного продукту поділяються на функціональні і нефункціональні. Функціональні вимоги – це перелік функцій, які повинні надаватися програмним продуктом. Нефункціональні вимоги – це перелік умов, у яких повинен використовуватися програмний продукт.

Програмні продукти у сфері фінансових технологій – це системи зі складною архітектурою і логікою. Архітектура визначає перелік компонентів системи і зв'язки між ними. Логіка визначає послідовність дій, які повинні виконуватися системою для досягнення очікуваного результату. Сучасні програмні продукти – це не лише код, написаний на певній мові програмування. Для розробки і розгортання програмних продуктів можуть знадобитися додаткові компоненти, такі як сервери додатків, сервери баз даних, черги повідомлень, сервіси логування, моніторингу і т. д. Ці компоненти формують інфраструктуру, необхідну для коректної роботи програмних продуктів.

Вимоги до програмних продуктів можуть передбачати взаємодію з зовнішніми системами, такими як платіжні системи, банки, реєстри, довідники і т. д. Взаємодія з зовнішніми системами передбачає необхідність налаштування відповідних доступів, захищеного обміну даними, а також забезпечення зворотної сумісності.

Отже, можна зробити висновок, що важливе значення мають вимоги до розгортання програмних продуктів. Ігнорування або недостатня увага до визначення та оформлення цих вимог може призвести до ускладнення процесу розгортання та експлуатації реалізованих програмних продуктів навіть за умови правильної реалізації функціональних вимог.

До основних категорій нефункціональних вимог до програмних продуктів у сфері фінансових технологій можна віднести наступні:

- Вимоги до швидкодії.
- Вимоги до відмовостійкості.
- Вимоги до безпеки даних.
- Вимоги до відповідності вимогам законодавства.
- Вимоги до документації.
- Вимоги до розгортання.

Швидкодія – це характеристика системи, яку можна описати як кількість операцій або даних, оброблених протягом одиниці часу, наприклад кількість оброблених запитів/транзакцій/рядків файлу за секунду. Ця характеристика має важливе значення для систем,

які повинні стабільно працювати в умовах високого навантаження. У випадку використання гнучких підходів до розробки програмних продуктів, які передбачають розробку продуктів поступово в рамках ітерацій, важливим аспектом процесу розробки є моніторинг впливу змін, виконаних в рамках ітерації, на показники швидкодії системи. Такий моніторинг повинен виконуватися на тестових і production-оточеннях з метою оперативного виявлення ознак деградації швидкодії і виправлення її причини. Це рутинна робота, яка передбачає збір, візуалізацію та аналіз статистики, тому може виявитися доцільним використання інструментів для її автоматизації, наприклад Kibana або New Relic.

Відмовостійкість – це характеристика системи, яку можна описати як здатність системи продовжувати роботу у випадку виникнення аварійних ситуацій. Ця характеристика тісно пов'язана зі швидкодією, оскільки в умовах високого навантаження можуть виникнути аварійні ситуації, пов'язані з недостатністю ресурсів, необхідних для стабільної роботи системи.

Для досягнення визначеного рівня швидкодії і відмовостійкості необхідно передбачити можливість масштабування інфраструктури, необхідної для коректної роботи програмних продуктів. Основним підходом до масштабування є кластеризація, яка передбачає розгортання і синхронізацію декількох екземплярів одного компоненту інфраструктури, наприклад розгортання декількох серверів додатків, які зможуть паралельно обробляти необхідні операції, а також забезпечувати працездатність системи у випадку виходу з ладу одного або декількох учасників кластера. Варто відзначити, що правильне налаштування інфраструктури має не менш важливе значення ніж правильна реалізація функціональних вимог, оскільки той самий програмний код може по-різному працювати на різних оточеннях з точки зору швидкодії і відмовостійкості. У першу чергу це важливо під час тестування програмного продукту на тестовому оточенні, оскільки через різницю у налаштуваннях тестового і production-оточення можуть виникнути ситуації, коли деякі проблеми не будуть проявлятися на тестовому оточенні, але можуть проявитися на production-оточенні. Наприклад, тестове оточення може бути налаштоване без використання кластеризації, а production-оточення – з використанням кластеризації. Як наслідок, на тестовому оточенні не буде можливості виявити потенційні проблеми,

пов'язані з синхронізацією учасників кластера, а також забезпеченням працездатності системи у випадку виходу з ладу одного або декількох учасників кластера. Отже, можна зробити висновок, що для якісного тестування програмних продуктів налаштування тестового оточення повинні максимально відповідати налаштуванням production-оточення. Також потрібно брати до уваги кількість даних та одночасних операцій, які обробляються на тестовому оточенні, оскільки деякі проблеми можуть проявитися лише в умовах високого навантаження. Наприклад, може виникнути нестача ресурсів (процесора, оперативної пам'яті, дискової пам'яті) або обробка операцій/даних може виконуватися повільно, що може бути пов'язано з неправильною або неефективною реалізацією програмного коду, а також неправильними налаштуваннями інфраструктури.

Безпека даних має критичне значення для програмних продуктів у сфері фінансових технологій. Налаштування програмних продуктів та інфраструктури, необхідної для коректної роботи програмного продукту, повинні забезпечувати надання необхідних доступів, надійний захист даних, які зберігаються у базах даних, а також надійний захист каналів обміну даними. У першу чергу це важливо під час розробки і розгортання компонентів системи, які повинні взаємодіяти з зовнішніми системами. До таких компонентів можна віднести API (Application Programming Interface), який визначає перелік операцій, необхідних для інтеграції з зовнішніми системами. Основними аспектами під час розробки і розгортання API є автентифікація, авторизація і захист каналу для обміну даними між клієнтом і сервером.

Під час визначення вимог до програмних продуктів у сфері фінансових технологій особливу увагу потрібно приділяти вимогам, реалізація яких підпадає під дію законодавства, оскільки невідповідність програмних продуктів вимогам законодавства може стати перешкодою для можливості їхнього використання у тій чи іншій країні. Окрім законодавства конкретної країни важливо враховувати вимоги міжнародних нормативно-правових актів, директив і стандартів. Особливу увагу потрібно звернути на GDPR (General Data Protection Regulation), PSD2 (revised Payment Services Directive) і PCI DSS (Payment Card Industry Data Security Standard). GDPR – це нормативно-правовий акт Європейського союзу, у якому визначені положення щодо

захисту персональних даних в межах Європейського союзу, а також експорту персональних даних за межі Європейського союзу [8]. PSD2 – це директива Європейського союзу, у якій визначені положення щодо регулювання платіжних послуг у Європейському союзі [9]. PCI DSS – це стандарт безпеки у сфері захисту даних платіжних карт [10]. Ключові аспекти GDPR, PSD2 і PCI DSS наведені у таблиці 1.

На основі перелічених ключових аспектів можна зробити висновок, що відповідність програмних продуктів вимогам законодавства тісно пов'язана з безпекою даних.

Документація є важливим компонентом програмного продукту, оскільки вона надає інформацію про деталі його реалізації та особливості його використання. У випадку використання гнучких підходів до розробки програмних продуктів, які передбачають реалізацію продуктів поступово в рамках ітерацій, підтримка документації в актуальному стані є неперервним процесом. На відміну від водоспадного підходу, в ітеративних підходах підготовка повноцінної документації є значно складнішим процесом, оскільки у більшості випадків на початковому етапі розробки відсутні всі вимоги до продукту. Також існує ризик зміни існуючих вимог в рамках наступних ітерацій. До основних категорій інформації, яка повинна бути включена у документацію, можна віднести інструкції користувача, схеми з описом архітектури, інфраструктури, структур даних, бізнес-процесів, а також опис API. Особливу увагу варто звернути на опис API, оскільки він необхідний

для інтеграції програмних продуктів з зовнішніми системами. У випадку, коли API передбачає наявність великої кількості операцій, а також складну структуру запитів/відповідей, може виявитися доцільним опис API в уніфікованому форматі, який дозволить автоматично згенерувати програмний код API для сервера і клієнта, а також документацію API. Зручним інструментом для уніфікованого опису API є OpenAPI, який дозволяє виконувати опис API у вигляді YAML файлу з визначеною структурою. На основі цього файлу можна згенерувати програмний код API для сервера і клієнта, а також документацію API з детальним описом всіх операцій у вигляді веб-сторінки.

Для розгортання програмних продуктів є необхідним придбання або оренда відповідного апаратного забезпечення. Основними підходами до розгортання програмних продуктів є On-Premises і Cloud. Порівняння цих підходів наведено у таблиці 2.

На основі виконаного порівняння можна зробити висновок, що основними критеріями для вибору підходу до розгортання програмних продуктів є вартість і відповідність законодавству. Важливе значення має вибір правильного підходу до розрахунку кількості ресурсів, необхідних для забезпечення визначеного рівня швидкодії і відмовостійкості, оскільки результати цього розрахунку матимуть безпосередній вплив на вартість розгортання програмних продуктів.

Висновки. Нефункціональні вимоги до програмних продуктів мають не менш важливе значення ніж функціональні вимоги, оскільки

Таблиця 1

Ключові аспекти GDPR, PSD2 і PCI DSS

Документ	Ключові аспекти
GDPR	<ul style="list-style-type: none"> – Принципи обробки персональних даних – Права суб'єктів даних – Обов'язки контролерів і процесорів – Передача персональних даних до третіх країн або міжнародних організацій
PSD2	<ul style="list-style-type: none"> – Запровадження нових типів постачальників платіжних послуг – Посилення вимог до безпеки і запровадження суворої автентифікації клієнтів (Strong Customer Authentication – SCA) – Покращення прав споживачів і прозорості платіжних послуг – Регулювання доступу до платіжних рахунків
PCI DSS	<ul style="list-style-type: none"> – Побудова і підтримка безпечної мережі – Захист даних власників карток – Підтримка програми управління вразливістю – Впровадження суворих заходів контролю доступу – Регулярний моніторинг і тестування мереж – Підтримка політики інформаційної безпеки

Джерело: сформовано на основі [8–10]

Таблиця 2

Порівняння підходів до розгортання програмних продуктів

Критерій	On-Premises	Cloud
Місце розгортання	Використовуються ресурси замовника. Замовник відповідає за всі процеси, пов'язані з налаштуванням і підтримкою ресурсів.	Використовуються ресурси постачальника cloud-рішення. Постачальник відповідає за налаштування і підтримку ресурсів.
Контроль	Більше контролю над налаштуваннями ресурсів і доступом до даних.	Менше контролю над налаштуваннями і доступом до даних.
Безпека	Безпека залежить від політики замовника.	Безпека залежить від політики постачальника cloud-рішення.
Відповідність законодавству	З точки зору безпеки локальне зберігання даних може виявитися більш надійним рішенням за рахунок більшого контролю над налаштуваннями ресурсів і доступом до даних.	Постачальник cloud-рішень повинен дотримуватися законодавства щодо роботи з даними, однак збереження даних на сторонніх серверах може виявитися більш ризикованим, особливо у випадку зберігання персональних і конфіденційних даних.
Вартість	Більша вартість розгортання, масштабованості і підтримки.	Менша вартість розгортання, масштабованості і підтримки, оскільки оплата здійснюється за фактично використані ресурси. Кількість необхідних ресурсів можна змінювати залежно від потреб у той чи інший момент.

Джерело: сформовано на основі [11]

від їхньої реалізації будуть залежати експлуатаційні характеристики продукту. В рамках дослідження були визначені основні категорії нефункціональних вимог, які мають важливе значення для програмних продуктів у сфері фінансових технологій. Була обґрунтована важливість кожної категорії вимог, визначений вплив цих вимог на процес розробки, розгортання та експлуатації програмних продуктів. Зроблено висновок, що визначення і реалізація нефункціональних вимог тісно пов'язані з етапом розгортання та експлуатації програмних продуктів, оскільки для забезпечення швидкодії, відмовостійкості, безпеки даних і

відповідності законодавству є важливою не лише коректна реалізація програмного коду, але й коректні налаштування інфраструктури, необхідної для роботи програмного продукту. Зроблено висновок, що розробка документації є не менш важливим процесом ніж розробка самого програмного продукту, оскільки від неї буде залежати правильність використання і зручність підтримки продукту. Загалом можна зробити висновок, що визначення нефункціональних вимог є актуальною темою, тому є доцільними подальші дослідження з метою визначення ефективних підходів до реалізації цих вимог.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ:

1. Software Projects Don't Have to Be Late, Costly, and Irrelevant. URL: <https://www.bcg.com/publications/2024/software-projects-dont-have-to-be-late-costly-and-irrelevant> (дата звернення: 22.04.2025)
2. Коваленко О. Є., Смолій В. М., Лі Л. Інженерія вимог систем інтернету речей. *Математичні машини і системи*. 2025. № 1. С. 64–75.
3. Корнієнко О. О., Крупа Н. О. Комплексний підхід до управління бізнес-вимогами в системах електронної комерції. *Телекомунікаційні та інформаційні технології*. 2024. № 4 (85). С. 73–82.
4. Ольховська О. О., Парфьонова Т. О., Бражніченко А. О. Бізнес-аналіз та проектний менеджмент в ІТ: ефективні стратегії та практики. *Збірник наукових праць НУК*. 2024. № 3. С. 94–98.
5. Храпкін О. М., Кіндрат О. В., Чопей Р. С. Управління проектами в ІТ-галузі: методики, інструменти та керування ризиками. *Економіка та суспільство*. 2023. № 55.

6. Микитюк П., Микитюк В. Фінансові технології в управлінні грошима, інвестиціями та фінансовими послугами. *Вісник економіки*. 2025. № 1. С. 65–78.
7. Перебийніс Д.С. Електронні фінансові технології: зміна підходів до залучення капіталу. *Електронне наукове фахове видання з економічних наук «Modern Economics»*. 2025. № 49. С. 177–187.
8. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016. URL: <https://eur-lex.europa.eu/eli/reg/2016/679> (дата звернення: 22.04.2025)
9. Directive (EU) 2015/2366 of the European Parliament and of the Council of 25 November 2015. URL: <https://eur-lex.europa.eu/eli/dir/2015/2366> (дата звернення: 22.04.2025)
10. Payment Card Industry Security Standards. URL: https://listings.pcisecuritystandards.org/pdfs/pcisssc_overview.pdf (дата звернення: 22.04.2025)
11. On-Premises vs Cloud: Key Differences, Pros & Cons. URL: <https://www.cloudpanel.io/blog/on-premises-vs-cloud-computing/> (дата звернення: 22.04.2025)

REFERENCES:

1. Software Projects Don't Have to Be Late, Costly, and Irrelevant. Available at: <https://www.bcg.com/publications/2024/software-projects-dont-have-to-be-late-costly-and-irrelevant> (accessed April 22, 2025)
2. Kovalenko O. Ye., Smolii V. M., Li L. (2025) Inzheneriia vymoh system internetu rechei [Requirements engineering of Internet of Things systems]. *Matematychni mashyny i systemy*, no. 1, pp. 64–75. (in Ukrainian)
3. Korniienko O. O., Krupa N. O. (2024) Kompleksnyi pidkhod do upravlinnia biznes-vymohamy v systemakh elektronnoi komertsii [A comprehensive approach to managing business requirements in e-commerce systems]. *Telekomunikatsiini ta informatsiini tekhnolohii*, no. 4, vol. 85, pp. 73–82. (in Ukrainian)
4. Olkhovska O. O., Parfyonova T. O., Brazhnichenko A. O. (2024) Biznes-analiz ta proektnyi menedzhment v IT: efektyvni stratehii ta praktyky [Business analysis and project management in IT: effective strategies and practices]. *Zbirnyk naukovykh prats NUK*, no. 3, pp. 94–98. (in Ukrainian)
5. Khrapkin O. M., Kindrat O. V., Chopei R. S. (2023) Upravlinnia proiektamy v IT-haluzi: metodyky, instrumenty ta keruvannia ryzykamy [Project management in the IT industry: methodologies, tools, and risk management]. *Ekonomika ta suspilstvo*, no. 55. (in Ukrainian)
6. Mykytiuk P., Mykytiuk V. (2025) Finansovi tekhnolohii v upravlinni hroshyma, investytsiamy ta finansovymy posluhamy [Financial technologies in money management, investments and financial services]. *Visnyk ekonomiky*, no. 1, pp. 65–78. (in Ukrainian)
7. Perebyinis D. S. (2025) Elektronni finansovi tekhnolohii: zmina pidkhodiv do zaluchennia kapitalu [Electronic financial technologies: changing approaches to raising capital]. *Elektronne nauкове fakhove vydannia z ekonomichnykh nauk «Modern Economics»*, no. 49, pp. 177–187. (in Ukrainian)
8. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016. Available at: <https://eur-lex.europa.eu/eli/reg/2016/679> (accessed April 22, 2025)
9. Directive (EU) 2015/2366 of the European Parliament and of the Council of 25 November 2015. Available at: <https://eur-lex.europa.eu/eli/dir/2015/2366> (accessed April 22, 2025)
10. Payment Card Industry Security Standards. Available at: https://listings.pcisecuritystandards.org/pdfs/pcisssc_overview.pdf (accessed April 22, 2025)
11. On-Premises vs Cloud: Key Differences, Pros & Cons. Available at: <https://www.cloudpanel.io/blog/on-premises-vs-cloud-computing/> (accessed April 22, 2025)