

УДК: 004.413

JEL Classification: M15

[https://doi.org/10.31498/2225-6725.1\(39\).2024.314686](https://doi.org/10.31498/2225-6725.1(39).2024.314686)Камишнікова Е.В.¹, Омельченко Д.В.²**АНАЛІЗ ПІДХОДІВ ДО УПРАВЛІННЯ РОЗРОБКОЮ ПРОГРАМНИХ ПРОДУКТІВ У СФЕРІ ФІНАНСОВИХ ТЕХНОЛОГІЙ**

Kamyshnykova E., Omelchenko D. Analysis of approaches to managing the development of software products in the field of financial technologies.

The article analyzes the existing approaches to managing the development of software products in order to determine the main criteria for their selection in the field of financial technologies. It provides an analysis of the main models and methodologies for the development of software products in order to determine their main characteristics, advantages and disadvantages. The article recognizes methodologies based on Agile principles as the most popular. The main requirements for software products in the field of financial technologies were determined, as well as the key points that must be paid attention to during the development of software products in this field. It was established that during the development of software products in the field of financial technologies, special attention should be paid to the stages of testing, implementation, integration with external systems, as well as development and support of documentation. The main criteria for the selection of approaches to the development of software products in this area were determined. It was established that flexible methodologies based on an iterative model are suitable for the development of software products in this field.

Keywords: *Software products; financial technologies; models; principles; methodologies; iteration, testing; implementation; integration; documentation.*

Камишнікова Е. В., Омельченко Д. В. Аналіз підходів до управління розробкою програмних продуктів у сфері фінансових технологій.

У статті виконано аналіз існуючих підходів до управління розробкою програмних продуктів з метою визначення основних критеріїв їх вибору у сфері фінансових технологій. Проведено аналіз основних моделей і методологій розробки програмних продуктів з метою визначення їх основних характеристик, переваг і недоліків. Найбільш популярними визнано методології, які базуються на принципах Agile. Визначено основні вимоги до програмних продуктів у сфері фінансових технологій, а також ключові моменти, на які необхідно звернути увагу під час розробки програмних продуктів у даній сфері. Встановлено, що під час розробки програмних продуктів у сфері фінансових технологій особливу увагу необхідно звернути на етапи тестування, впровадження, інтеграції з зовнішніми системами, а також розробки і підтримки документації. Визначено основні критерії вибору підходів до розробки

¹ <https://orcid.org/0000-0003-1835-9786>

Камишнікова Евеліна Вікторівна, професор кафедри економіки підприємств д.е.н., професор, ДВНЗ «Прийзовський державний технічний університет», м. Дніпро, Україна, kamyshnykova_e_v@pstu.edu
Evelina Kamyshnykova, Professor of Department of Economics of Enterprise, D.Sc. (Economics), Professor, SHEI «Pryazovskyi State Technical University», Dnipro, Ukraine

² <https://orcid.org/0009-0005-0420-1808>

Омельченко Дмитро Вікторович; аспірант, ДВНЗ «Прийзовський державний технічний університет», м. Дніпро, Україна, omelchenko_d_v@aspirants.pstu.edu
Dmytro Omelchenko; postgraduate student, SHEI «Pryazovskyi State Technical University», Dnipro, Ukraine

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted reuse, distribution and reproduction in any medium provided the original work is properly cited.

програмних продуктів у даній сфері. Встановлено, що для розробки програмних продуктів у даній сфері підходять гнучкі методології, які базуються на ітераційній моделі.

***Ключові слова:** програмні продукти; фінансові технології; моделі; принципи; методології; ітерація, тестування; впровадження; інтеграція, документація.*

Постановка проблеми та її зв'язки з науковими чи практичними завданнями. Розробка програмних продуктів — це складний процес, який у більшості випадків є роботою, яка виконується командою фахівців, тому управління розробкою програмних продуктів є важливим процесом для організації ефективної роботи і взаємодії всіх учасників команди. Від ефективності управління буде залежати здатність організації-розробника утримувати позиції в умовах конкуренції, а також здатність реалізованих продуктів задовольняти потреби кінцевих користувачів. Згідно статистики невдач проектів розробки програмних продуктів [1] 32% невдач пов'язані з поганим управлінням вимогами, 47% невдач пов'язані з поганим управлінням проектами, лише 52% проектів проходять після релізу перевірку якості, більше 80% менеджерів проектів визнають, що свідомо випустили дефектний продукт.

Аналіз останніх досліджень і публікацій. Актуальність теми управління розробкою програмних продуктів підтверджується великою кількістю існуючих досліджень. Дану тему досліджували Сеспедес Гарсія Н. В. і Сеспедес Гарсія П. Д. [2], Овсак Б. О., Тульчинська С. О. і Ліскович Н. Ю. [3], Кордунова Ю. С., Смотров О. О., Кокотко І. Я. і Малець Р. Б. [4], Близнюкова І. О., Семко І. Б. і Кійко С. Г. [5], Грицюк П. Ю., Іванишин А. В. і Грицюк Ю. І. [6], Кравчук О. [7] та інші. У статтях [2-5] були проаналізовані існуючі підходи до управління розробкою програмних продуктів з метою визначення критеріїв їх вибору. У статті [6] був зроблений акцент на підходах до забезпеченні якості програмних продуктів. У статті [7] був зроблений акцент на підходах до впровадження програмних продуктів.

Виділення невирішених частин загальної проблеми, котрим присвячується стаття. Незважаючи на існування переліку розроблених

підходів до управління розробкою програмних продуктів, у зв'язку з динамічним розвитком фінансових технологій [8], а також необхідністю управління ризиками, які можуть виникнути під час розробки програмних продуктів, застосування даних підходів у сфері фінансів потребує подальшого дослідження.

Формування цілей статті. Мета дослідження — на основі аналізу існуючих підходів до управління розробкою програмних продуктів визначити основні критерії їх вибору у сфері фінансових технологій.

Результати дослідження. Підходи до розробки програмних продуктів базуються на моделях, принципах і методологіях.

Основні моделі розробки програмних продуктів - це водоспадна та ітераційна.

Основні характеристики водоспадної моделі [9]: послідовне виконання всіх етапів розробки; перехід до наступного етапу можливий лише після повного завершення попереднього етапу. Різновидом водоспадної моделі є V-подібна модель, яка передбачає наявність етапу тестування для кожного основного етапу розробки.

Основні характеристики ітераційної моделі [9]: вимоги до продукту розділяються на декілька частин; для кожної частини виконується повний цикл розробки згідно водоспадної моделі. Різновидом ітераційної моделі є спіральна модель, яка передбачає наявність етапів аналізу ризиків перед початком кожної ітерації. Переваги і недоліки вказаних вище моделей вказані у таблиці 1.

Згідно статистики [10] найбільш популярними є методології, які базуються на принципах Agile, серед яких найбільш популярними є методології Scrum і Kanban. Також варто розглянути методології Scrumban та XP (Extreme Programming).

Основні принципи Agile [11]: акцент на активній взаємодії між учасниками команди; акцент на активній взаємодії з замовником; акцент на швидкій

адаптації до зміни вимог; акцент на працездатному продукті, а не повноцінній документації.

Таблиця 1. Переваги і недоліки моделей розробки програмних продуктів

Модель	Переваги	Недоліки
Водоспадна	<ul style="list-style-type: none"> - Можливість створення якісної документації для кожного етапу розробки. - Можливість точного планування термінів виконання і витрат. 	<ul style="list-style-type: none"> - Необхідність збору і затвердження всіх вимог перед початком розробки. - Необхідність починати процес розробки спочатку у випадку зміни вимог або виявлення проблем на більш пізніх етапах розробки.
Ітераційна	<ul style="list-style-type: none"> - Можливість прискореного впровадження продукту шляхом розробки початкової версії продукту з базовим функціоналом. - Можливість внесення змін і додавання нового функціоналу в рамках наступних ітерацій. - Можливість паралельної розробки незалежних частин продукту. 	<ul style="list-style-type: none"> - Складність планування термінів виконання і витрат через невизначеність подальших вимог. - Дефекти, які не були вчасно виявлені в рамках попередніх ітерацій, потраплять у наступні ітерації.

* сформовано на основі аналізу джерела [9]

Основні характеристики методології Scrum [12]: розробка виконується в рамках спринтів — обмежених періодів часу тривалістю від одного до декількох тижнів; перед початком спринта виконується планування задач, які необхідно виконати в рамках спринта; протягом спринта проводяться регулярні зустрічі учасників команди з метою оцінки прогресу виконання задач; після завершення спринта проводиться ретроспектива з метою оцінки результатів спринта, а також прийняття рішень щодо наступних спринтів.

Основні характеристики методології Kanban [13]: візуалізація прогресу виконання задач за допомогою дошки з колонками, кожна з яких відповідає статусу задачі (наприклад, Ready for Development, In progress, Code Review, Ready for Testing і т. д.); у кожній колонці розміщуються задачі з відповідним статусом; метою кожного учасника команди є переміщення його задач з початкової колонки у кінцеву; відсутність явно визначених ітерацій.

Основні характеристики методології Scrumban [14]: комбінація методологій Scrum і Kanban; від методології Scrum запозичена організаційна структура (спринти, планування, щоденні зустрічі, ретроспективи); від методології Kanban запозичений підхід до візуалізації прогресу виконання задач.

Основні характеристики методології XP (Extreme Programming) [15]: розробка виконується в рамках коротких ітерацій; передбачається використання перевірених практик для написання якісного коду, який можна легко розширювати, підтримувати і використовувати повторно; передбачається написання автоматичних тестів для вчасного виявлення дефектів.

Переваги і недоліки вказаних вище методологій вказані у таблиці 2.

Фінансові технології — це сукупність технологій, метою яких є покращення якості та ефективності фінансових процесів і послуг. Для реалізації фінансових технологій необхідна розробка відповідних програмних продуктів. Програмні продукти у сфері фінансових технологій повинні відповідати наступним вимогам: забезпечення косистентності даних під час виконання фінансових операцій; забезпечення стабільної та ефективної роботи в умовах високого навантаження (одночасне виконання великої кількості операцій, обробка великої кількості даних); забезпечення відмовостійкості у випадку виникнення аварійних ситуацій; забезпечення захисту персональних і конфіденційних даних від несанкціонованого доступу; забезпечення відповідності чинному законодавству країн, у яких планується використання програмного продукту.

Таблиця 2. Переваги і недоліки методологій розробки програмних продуктів

Методологія	Переваги	Недоліки
Scrum	<ul style="list-style-type: none"> - Структурованість процесу розробки за рахунок наявності явно визначених ітерацій. - Регулярні випуски оновлених версій продукту. 	<ul style="list-style-type: none"> - Складність застосування для великих команд через ускладнення взаємодії між учасниками команди.
Kanban	<ul style="list-style-type: none"> - Наглядність процесу розробки за рахунок його візуалізації. - Гнучкість реагування на зміни пріоритетів задач. 	<ul style="list-style-type: none"> - Відсутність явно визначених ітерацій і можливість неочікуваних змін пріоритетів задач можуть призвести до зменшення контрольованості процесу розробки.
Scrumban	<ul style="list-style-type: none"> - Збільшення гнучкості управління процесом розробки за рахунок використання переваг методологій Scrum і Kanban. 	<ul style="list-style-type: none"> - Збільшення гнучкості може призвести до зменшення структурованості і контрольованості процесу розробки.
XP (Extreme Programming)	<ul style="list-style-type: none"> - Висока якість реалізованих продуктів. - Регулярні випуски оновлених версій продукту. 	<ul style="list-style-type: none"> - Складність застосування через необхідність наявності у розробників високої культури проектування, програмування і тестування. - Всі розробники повинні однаково добре знати деталі системи і нести спільну відповідальність за зміни, які вони вносять.

* сформовано на основі аналізу джерел [12-15]

Дефекти, пов'язані з некоректним виконанням фінансових операцій, можуть призвести до негативних наслідків для кінцевих користувачів програмного продукту і, як наслідок, негативно вплинути на репутацію

організації-розробника. виправлення наслідків таких дефектів може виявитися значно складнішою задачею ніж виправлення самих дефектів, що може негативно вплинути на планування подальшої розробки продукту. Отже, можна зробити висновок, що етап тестування продукту є критично важливим. Окрім функціональних проблем можуть виникнути проблеми зі швидкістю продукту, тому може виявитися доцільним виконання навантажувального тестування, яке передбачає моделювання умов використання продукту, наближених до реальних.

Для впровадження програмного продукту необхідна підготовка відповідної інфраструктури (розгортання серверів додатків і баз даних, налаштування мережі, доступів і т. д.). Етап впровадження є критично важливим для організації випуску оновлених версій продукту, забезпечення стабільної та ефективної роботи продукту в умовах великого навантаження, відмовостійкості продукту у випадку виникнення аварійних ситуацій, а також забезпечення захисту персональних і конфіденційних даних від несанкціонованого доступу.

Програмний продукт може взаємодіяти з зовнішніми системами (банками, платіжними системами і т. д.) у режимі клієнта або сервера. Для інтеграції з зовнішніми системами необхідна наявність API (Application Programming Interface) — це контракт між клієнтом і сервером, який описує операції сервера без розкриття їх реалізації. Важливими аспектами інтеграції є автентифікація та авторизація клієнта, а також захист даних, які передаються між клієнтом і сервером. У режимі клієнта програмний продукт стає залежним від зовнішніх систем, що може призвести до збільшення термінів його розробки і впровадження, тому може виявитися доцільним використання програмних продуктів для імітації роботи API, що дозволить виконувати розробку і тестування продукту без необхідності доступу до зовнішніх систем.

Для централізованого зберігання інформації про можливості і деталі реалізації програмного продукту необхідна розробка і підтримка документації. Відсутність або низька якість документації може призвести до відсутності або

недостатнього розуміння архітектури і можливостей продукту всіма учасниками команди, неправильного використання продукту кінцевими користувачами, а також проблем під час інтеграції продукту з зовнішніми системами.

Враховуючи проаналізовані вище підходи до управління розробкою програмних продуктів і ключові моменти у процесах розробки програмних продуктів у сфері фінансових технологій, можна визначити наступні критерії вибору підходів до розробки програмних продуктів у даній сфері:

1. Якщо неможливо передбачити всі вимоги до програмного продукту перед початком розробки, тоді може виявитися доцільним використання гнучких методологій розробки, які пропонують кращу адаптацію до зміни вимог, на відміну від водоспадної моделі.

2. Для забезпечення максимальної відповідності програмного продукту вимогам замовника необхідне активне залучення замовника у процес розробки з метою вчасного отримання відгуку про роботу продукту, а також уточнення вимог. Такий підхід передбачає використання гнучких методологій розробки.

3. Якість і надійність програмних продуктів у сфері фінансів має критичне значення, тому особливу увагу необхідно звернути на етап тестування. В ітераційній моделі розробки етап тестування буде менш трудозатратний, оскільки в рамках ітерації розробляється лише частина функціоналу, на відміну від водоспадної моделі. З іншого боку, в рамках кожної ітерації необхідно виконувати перевірку функціоналу, реалізованого в рамках попередніх ітерацій (регресійне тестування), оскільки під час реалізації нового функціоналу може змінюватися існуючий код продукту. З метою зменшення часу, необхідного для виконання регресійного тестування, а також зменшення впливу людського фактору, може виявитися доцільною автоматизація тестування. Також може виявитися доцільним виконання навантажувального тестування.

4. Впровадження програмного продукту є невід'ємним етапом процесу розробки. В ітераційній моделі розробки впровадження оновлених версій продукту виконується в рамках кожної ітерації, на відміну від водоспадної

моделі. З метою зменшення часу, необхідного для випуску оновлених версій продукту, може виявитися доцільною автоматизація впровадження, що особливо критично у випадку виправлення дефектів, виявлених під час використання продукту кінцевими користувачами.

5. Інтеграція програмного продукту з зовнішніми системами передбачає необхідність налагодження взаємодії з програмними продуктами, які розробляються іншими організаціями. В ітераційній моделі розробки інтеграцію продукту з зовнішніми системами можна розглядати як розширення функціоналу, яке може виконуватися в рамках послідовних або паралельних ітерацій.

6. Розробка і підтримка документації програмного продукту є важливим етапом процесу розробки. У гнучких методологіях розробки акцент робиться на розробці працездатного продукту, а не створенні повноцінної документації, на відміну від водоспадної моделі, тому важливо визначити основні типи інформації, яка повинна потрапити у документацію. До такої інформації можна віднести опис архітектури, опис інфраструктури, опис API, інструкції для користувачів і т. д.

Отже, можна зробити висновок, що для розробки програмних продуктів у сфері фінансових технологій підходять гнучкі методології, які базуються на ітераційній моделі. До таких методологій можна віднести Scrum, Scrumban, XP. Методологія Kanban не передбачає наявності явно визначених ітерацій, що робить її менш передбачуваною з точки зору термінів випуску оновлених версій продукту.

Висновки даного дослідження і перспективи подальших робіт у цьому напрямку. Програмні продукти у сфері фінансових технологій повинні відповідати ряду вимог, які необхідно враховувати у процесі розробки даних продуктів. Як наслідок, важливе значення має вибір підходів до управління розробкою програмних продуктів. Вибір конкретного підходу залежить від поточного стану проекту (початок, активна розробка, підтримка), вимог до

конкретного продукту, розміру команди, термінів виконання, бюджету і т. д. Отже, є доцільними подальші дослідження з метою визначення підходів до оптимізації процесів управління розробкою програмних продуктів у сфері фінансових технологій в умовах впливу різноманітних факторів (технічних, організаційних, фінансових і т. д.).

Перелік використаних джерел:

1. *Software Project Failure Statistics [Fresh Research]. URL: <https://gitnux.org/software-project-failure-statistics/> (дата звернення: 31.03.2024)*
2. *Сеснедес Гарсія, Н. В., & Сеснедес Гарсія, П. Д. (2023). Моделі життєвого циклу розробки програмного забезпечення. Молодий вчений, 2(114), 17-20. <https://doi.org/10.32839/2304-5809/2023-2-114-4>*
3. *Овсак, Б. О., Тульчинська, С. О., & Ліскович, Н. Ю. (2023). Методологічні аспекти управління проектами в галузі інформаційних технологій. Economic Synergy, (4), 8-22. <https://doi.org/10.53920/ES-2023-4-1>*
4. *Кордунова, Ю. С., Смор, О. О., Кокотко, І. Я., & Малець, Р. Б. (2021). Аналіз традиційного та гнучкого підходів до створення програмного забезпечення в динамічних умовах. Управління розвитком складних систем, (47), 71-77. <https://doi.org/10.32347/2412-9933.2021.47.71-77>*
5. *Близнакова, І. О., Семко, І. Б., & Кійко, С. Г. (2020). Огляд сучасних методологій управління командами IT-проектів. Управління розвитком складних систем, (43), 60-66. <https://doi.org/10.32347/2412-9933.2020.43.60-66>*
6. *Грицюк, П. Ю., Іванишин, А. В., & Грицюк, Ю. І. (2023). Забезпечення якості програмного продукту за стандартом IEEE 730-2014 в межах життєвого циклу реалізації проекту. Науковий вісник НЛТУ України, 33(2), 101-117. <https://doi.org/10.36930/40330214>*
7. *Кравчук, О. (2023). Модель впровадження CI/CD для оптимізації управління IT-проектами. Measuring and computing devices in technological processes, (3), 73-82. <https://doi.org/10.31891/2219-9365-2023-75-8>*
8. *Must-Know Fintech Statistics [Recent Analysis]. URL: <https://gitnux.org/fintech-statistics/> (дата звернення: 31.03.2024)*
9. *Software Development Models: Sliced, Diced and Organized in Charts. URL: <https://www.scnsoft.com/blog/software-development-models> (дата звернення: 31.03.2024)*
10. *State of Agile Report. URL: <https://info.digital.ai/rs/981-LOX-968/images/AR-SA-2022-16th-Annual-State-Of-Agile-Report.pdf> (дата звернення: 31.03.2024)*
11. *Manifesto for Agile Software Development. URL: <https://agilemanifesto.org/iso/en/manifesto.html> (дата звернення: 31.03.2024)*
12. *What is scrum and how to get started. URL: <https://www.atlassian.com/agile/scrum> (дата звернення: 31.03.2024)*
13. *Kanban. URL: <https://www.atlassian.com/agile/kanban> (дата звернення: 31.03.2024)*
14. *Scrumban: Mastering two Agile methodologies. URL: <https://www.atlassian.com/agile/project-management/scrumban> (дата звернення: 31.03.2024)*

15. The Rules of Extreme Programming. URL: <http://www.extremeprogramming.org/rules.html> (дата звернення: 31.03.2024)

References:

1. Software Project Failure Statistics [Fresh Research]. URL: <https://gitnux.org/software-project-failure-statistics/> (дата звернення: 31.03.2024)
2. Cespedes Garcia, N. V., & Cespedes Garcia, P. D. (2023). Life cycle models of software development. *Young Scientist*, 2(114), 17-20. <https://doi.org/10.32839/2304-5809/2023-2-114-4>
3. Ovsak, B. O., Tulchynska, S. O., & Liskovych, N. Yu. (2023). Methodological aspects of project management in the field of information technologies. *Economic Synergy*, (4), 8-22. <https://doi.org/10.53920/ES-2023-4-1>
4. Kordunova, Yu. S., Smotr, O. O., Kokotko, I. Ya., & Malets, R. B. (2021). Analysis of the traditional and flexible approaches to creating software in dynamic conditions. *Management of Development of Complex Systems*, (47), 71-77. <https://doi.org/10.32347/2412-9933.2021.47.71-77>
5. Blyznyukova, I. O., Semko, I. B., & Kiyko, S. G. (2020). Overview of modern team management methodologies in IT-projects. *Management of Development of Complex Systems*, (43), 60-66. <https://doi.org/10.32347/2412-9933.2020.43.60-66>
6. Hrytsiuk, P. Yu., Ivanyshyn, A. V., & Hrytsiuk, Yu. I. (2023). Quality assurance of software products in accordance with IEEE 730-2014 standard within the project implementation lifecycle. *Scientific Bulletin of UNFU*, 33(2), 101-117. <https://doi.org/10.36930/40330214>
7. Kravchuk, O. (2023). CI/CD implementation model for optimizing IT project management. *Measuring and computing devices in technological processes*, (3), 73-82. <https://doi.org/10.31891/2219-9365-2023-75-8>
8. Must-Know Fintech Statistics [Recent Analysis]. URL: <https://gitnux.org/fintech-statistics/> (дата звернення: 31.03.2024)
9. Software Development Models: Sliced, Diced and Organized in Charts. URL: <https://www.scnsoft.com/blog/software-development-models> (дата звернення: 31.03.2024)
10. State of Agile Report. URL: <https://info.digital.ai/rs/981-LQX-968/images/AR-SA-2022-16th-Annual-State-Of-Agile-Report.pdf> (дата звернення: 31.03.2024)
11. Manifesto for Agile Software Development. URL: <https://agilemanifesto.org/iso/en/manifesto.html> (дата звернення: 31.03.2024)
12. What is scrum and how to get started. URL: <https://www.atlassian.com/agile/scrum> (дата звернення: 31.03.2024)
13. Kanban. URL: <https://www.atlassian.com/agile/kanban> (дата звернення: 31.03.2024)
14. Scrumban: Mastering two Agile methodologies. URL: <https://www.atlassian.com/agile/project-management/scrumban> (дата звернення: 31.03.2024)
15. The Rules of Extreme Programming. URL: <http://www.extremeprogramming.org/rules.html> (дата звернення: 31.03.2024)