

DOI: <https://doi.org/10.32782/2524-0072/2024-63-120>

УДК 330.322

ОПТИМІЗАЦІЯ AGILE: СИНЕРГІЯ SCRUM І KANBAN У РОЗРОБЦІ ТА ПІДТРИМЦІ ПРОГРАМНОГО ПРОДУКТУ

OPTIMIZING AGILE: THE SYNERGY OF SCRUM AND KANBAN IN SOFTWARE DEVELOPMENT AND MAINTENANCE

Равлюк Максим Орестович

магістр,

Національний університет «Львівська політехніка»

ORCID: <https://orcid.org/0009-0006-8726-7235>

Ситник Йосиф Степанович

доктор економічних наук, доцент,

Національний університет «Львівська політехніка»

ORCID: <https://orcid.org/0000-0001-8718-5738>

Ravliuk Maksym, Sytnyk Yosyf

Lviv Polytechnic National University

Стаття присвячена актуальним питанням пошуку обґрунтованих рішень для оптимізації розробки та підтримки програмного продукту. Розглянуто різні змістові наповнення поняття синергії між Agile методологіями Scrum і Kanban. Проаналізовано властиві ознаки цих методологій, які виокремлюють їх серед інших стратегій розробки та підтримки програмних продуктів. Досліджено й узагальнено основні принципи їхнього використання в сучасних ринкових умовах для реального проєкту. Обґрунтовано взаємозв'язок між типом проєкту і складністю впровадження Agile підходів. Виявлено різну чутливість Agile методологій Scrum і Kanban до змін у внутрішньому та зовнішньому середовищі підприємства. Визначено особливо чутливі параметри, які впливають на успішність їхньої інтеграції в процесі розробки та підтримки програмного продукту.

Ключові слова: стратегія, управління, оптимізація, Scrum, Kanban.

The article addresses current issues in finding well-grounded solutions for optimizing the development and maintenance of software products. It explores the various conceptual meanings of synergy between Agile methodologies, Scrum, and Kanban. The distinctive features of these methodologies that set them apart from other software development and maintenance strategies are analyzed. The main principles of their application in modern market conditions for a real project are researched and summarized. The relationship between the project type and the complexity of implementing Agile approaches is substantiated. The varying sensitivity of Agile methodologies Scrum and Kanban to changes in the internal and external environment of the enterprise is identified. Particularly sensitive parameters affecting the successful integration of these methodologies in the process of software development and maintenance are determined. Special attention is paid to identifying the advantages and disadvantages of using Kanban and Scrum depending on the specific conditions and requirements of the project. It is shown that the Scrum methodology allows for clear planning and control of the development process through short iterations (sprints) and regular team meetings (stand-ups). The Kanban methodology, in turn, is distinguished by its ability to maintain a constant flow of tasks and effectively allocate team resources. It provides the ability to instantly see the status of tasks, allowing for quick identification and elimination of "bottlenecks" in the development process. This is especially important for teams working on the maintenance of existing products, where it is crucial to quickly and effectively respond to user requests and bug fixes. The article also examines practical aspects of combining Kanban and Scrum to achieve optimal productivity of the development team. Real cases are described where the combination of these methodologies significantly increased the team's efficiency, as well as the quality of the product and company profits.

Key words: strategy, management, optimization, Scrum, Kanban.

Постановка проблеми. В сучасному ринковому середовищі пошук оптимальних стратегій для розробки та підтримки програмного

продукту є ключовим завданням для бізнесу, що працює в інтернеті. Постійна адаптивність до змін та необхідність безперервного розви-



тку продуктів є критичними чинниками успіху. Висока конкуренція та швидкий темп змін змушують підприємства застосовувати гнучкі підходи, які дозволяють оперативно реагувати на нові виклики, забезпечуючи стабільне зростання та підтримку якості програмного забезпечення. У цьому контексті особливе значення набуває інтеграція методологій Agile, зокрема Scrum і Kanban, які сприяють підвищенню ефективності роботи команд розробників. Ці підходи не лише допомагають зберегти високий рівень якості та інноваційності програмних продуктів, від них також залежить здатність компаній до конкурентоспроможності на ринку.

Виділення невирішених раніше частин загальної проблеми. Відсутність прагматичних верифікацій щодо ефективності синергії між Agile методологіями Scrum і Kanban в умовах конкретних проєктів, а також нестача досліджень щодо їхнього впливу на довгострокову стабільність, якість і розвиток програмного продукту. Зазначене є критично важливим для будь-якої компанії, що надає послуги ПЗ, оскільки це дозволяє оцінити, наскільки комбіноване використання цих методологій сприяє досягненню реальних показників ефективності та конкурентоспроможності на ринку.

Формулювання цілей статті (постановка завдання). Метою статті є дослідження та систематизація теоретико-методичних засад і прикладних рекомендацій щодо оптимізації розробки та підтримки програмного продукту з використанням синергії Agile методологій Scrum і Kanban.

Виклад основного матеріалу дослідження. Поточний стан розвитку торгівлі в інтернеті характеризується динамічністю економічних, соціальних і технологічних процесів, що природно впливають на діяльність онлайн-бізнесів. Такі тенденції змушують підприємницьке середовище зважено обирати напрями розвитку та всебічно обґрунтовувати управлінські рішення, пов'язані з оптимізацією розробки та підтримки програмного продукту. Висока конкуренція та швидкі зміни на ринку вимагають впровадження гнучких методологій, що дозволяють оперативно реагувати на нові виклики, забезпечуючи стабільне зростання та підтримку якості програмного забезпечення.

Якість програмного забезпечення є одним із основних чинників конкурентоспроможності на ринку, економічного зростання та масштабування діяльності підприємств. Останнім

часом ключовим аспектом у процесі розвитку та підтримки програмного продукту є розуміння та впровадження методологій Agile.

Agile – це підхід, набір принципів до управління проєктами, особливо в сфері розробки та підтримки програмного забезпечення, який базується на ітеративній та інкрементній розробці, де вимоги та рішення еволюціонують завдяки співпраці самоорганізованих крос-функціональних команд. Agile методології заохочують адаптивне планування, еволюційний розвиток, ранню доставку та постійне вдосконалення, а також сприяють швидкій і гнучкій реакції на зміни, що дуже важливо для ведення успішного бізнесу [3].

Сьогодні одними з провідних і широкозастосовуваних методологій Agile сьогодні є Scrum і Kanban.

Scrum – це рамкова робоча методологія для керування проєктами, що базується на ітеративній моделі розробки та використовує регулярні короткострокові цикли розробки, відомі як "спринти" ("sprints"), для керування процесами та максимізації командної продуктивності [4].

Kanban – це потокова робоча методологія для керування проєктами, що базується на візуалізації процесів роботи та обмеженні одночасної кількості завдань, які перебувають у процесі виконання. Вона сприяє підвищенню продуктивності й ефективності команд, забезпечуючи стабільний рух завдань від початку до завершення проєкту [5].

Обидві методології є самодостатніми, але іноді в процесі роботи виникає потреба у пошуку синергії – гібридного підходу спрямованого на поєднання їхніх переваг. Це може статися через бажання відповісти на специфічні вимоги проєкту, наприклад, оптимізації використання ресурсів заради досягнення певних цілей.

Зокрема, подібна потреба виникла на реальному проєкті, стартапі, який надає програмні послуги AI-комерції для численних інтернет-магазинів (як локальних, так і світових брендів). З одного боку, забезпечуючи постійну й оперативну підтримку ПЗ, керівництво стартапу зацікавлене в покращенні співпраці з уже наявними клієнтами (інтернет-магазинами). З іншого боку, розширюючи функціонал наявного ПЗ, керівництво стартапу сподівається привернути нових клієнтів і розширити коло співпраці.

Розгляньмо цей кейс детальніше на прикладі робочих процесів команди front-end веб-розробників. Для забезпечення постійну

й оперативної підтримки ПЗ найкраще зарекомендувала себе методологія Kanban, а для розширення функціоналу наявного ПЗ – методологія Scrum.

Через тимчасові обмеження в фінансових ресурсах, компанія не має змоги найняти додаткових розробників для утворення двох паралельних front-end команд, отже рішення довелося шукати шляхом оптимізації роботи наявних працівників, пошуком "золотої середини" між підтримкою проєкту та його розвитком, пошуком синергії Kanban і Scrum.

Методологія Kanban має кілька ключових переваг, які можуть бути особливо корисними для команди front-end веб-розробників, яка забезпечує підтримку програмного забезпечення. Перелік цих переваг подано в таблиці 1.

Отже, коли процеси підтримки проєкту вже поставлені, завдяки перевагам у стабільності потоку роботи та гнучкості в управлінні задачами, Kanban дозволяє команді розробників фокусуватися на якісній і швидкій доставці: підтримувати наявних клієнтів, миттєво реагувати та швидко вносити запитовані зміни, виправляти помилки. Це дуже добре впливає на репутацію компанії, на її стосунки з клієнтами й приносить їй миттєву фінансову вигоду. Однак, це дещо гальмує розвиток

проєкту, оскільки не дає змоги сконцентруватися на розширенні послуг, які компанія буде готова запропонувати і наявним, і новим клієнтам. У "беклозі" Kanban практично щоденно додається якийсь новий терміновий "тикет" для підтримки наявних клієнтів й, беручи його до роботи, доводиться жертвувати часом, концентрацією та пріоритизацією "тикетів" для розробки нового функціоналу. За задумом керівництва компанії, ці нюанси повинні вирішити деякі практики методології Scrum (таблиця 2).

Експерименти зі синергією Kanban і Scrum вилилися для команди та компанії в такі позитивні результати:

1) Більшу структурованість роботи: завдяки новій методології зник хаос у роботі. Розробники відчують себе більш впевнено, оскільки робота стала більш планованою та прогнозованою. Прохання екстра терміново відкласти виконуване завдання та зайнятися іншим тепер практично виключене.

2) Планування "вікна" для термінових завдань: усі термінові завдання з підтримки мають змогу скористатися спеціальним "вікном", тоді коли решта не таких термінових завдань можуть бути спокійно відкладені до наступного спринта. Це дозволяє ефектив-

Таблиця 1

Переваги методології Kanban для підтримки ПЗ

Підтримка стабільного потоку роботи	Рівномірний розподіл навантаження: Kanban допомагає збалансувати навантаження на команду, оскільки завдання переходять від одного етапу до іншого на основі їхньої готовності, а не за заздалегідь визначеним графіком. Зменшення часу очікування: Завдяки видимості та управлінню WIP, команди можуть швидше виявляти і усувати блокуючі фактори, що знижує час простою і прискорює доставку цінності користувачам.
Гнучкість в управлінні задачами	Адаптивність до змін: Kanban дозволяє швидко реагувати на зміну пріоритетів і умов, тож розробники можуть взяти на себе нові завдання, щойно завершують поточні. Без жорстких спринтів: У Kanban немає потреби планувати роботу на фіксовані інтервали часу ("спринти"), як у Scrum. Це означає, що команда може більш гнучко реагувати на зміну вимог і непередбачені завдання.
Прозорість і видимість	Чітка візуалізація робочого процесу, що дозволяє легко відстежувати прогрес і виявляти проблеми. Поліпшення комунікації: Завдяки візуалізації процесу всі учасники команди, а також зацікавлені сторони, завжди в курсі поточного стану справ, що зменшує потребу в частих зустрічах і оновленнях.
Підвищення ефективності	Оптимізація робочого процесу: Kanban спрямований на мінімізацію часу очікування завдань, що дозволяє команді ефективно витратити час і ресурси на те, що є важливим в даному контексті. Безперервне покращення: Kanban заохочує команду до постійного аналізу свого процесу і пошуку шляхів його покращення, що сприяє підвищенню продуктивності та якості роботи.

Джерело: складено автором на основі [1; 2; 3; 5]

Таблиця 2

Переваги методології Scrum для розробки (розвитку) ПЗ

Чітке планування	Планування спринтів допомагає визначати цілі та завдання, забезпечуючи структурований підхід до розвитку нового функціоналу.
Регулярні спринти	Фіксовані короткі інтервали роботи дозволяють команді зосереджуватися на конкретних завданнях, не відкладаючи їх на далеку перспективу, забезпечуючи постійний прогрес у розробці нового функціоналу.
Регулярні ретроспективи	Після кожного спринту команда аналізує свої процеси та результати, що допомагає постійно покращувати продуктивність і якість розробки.

Джерело: складено автором на основі [1; 2; 3; 4]

ніше управляти пріоритетами та забезпечувати негайне реагування на критичні потреби.

3) Короткі двотижневі спринти: цей підхід дозволяє не відкладати роботу над новими тикетами з підтримки на занадто довгий час, забезпечуючи і регулярний прогрес у розробці нових фіч, і підтримку наявного функціоналу.

4) Прозорість виконання тасок: усі учасники процесу мають чітке уявлення про статуси виконання завдань, що допомагає передбачити коли дійде черга до їхнього тикета — чи це буде ще в цьому, чи вже в наступному спринті. Це сприяє кращому плануванню та управлінню очікуваннями, виключає тиск, овертайми та даремні обіцянки.

5) Почергова залученість розробників: поки половина команди сфокусована на підтримці наявних клієнтів, інша половина займається розробкою нового функціоналу. Це допомагає уникнути вигорання серед розробників, забезпечуючи їм різнопланові та цікаві завдання.

6) Переваги для компанії: хоч швидкість виконання запитів від існуючих клієнтів незначно впала, значно зросла якість виконання поставлених завдань. Розробники відчувають

психологічну впевненість і відсутність зайвого тиску, що дозволяє їм виконувати свої завдання більш ретельно та без поспіху. Крім того, існуючий функціонал був уже вагомо доповнений, що привернуло нових клієнтів, які потребували саме цих новододаних можливостей.

Отже, пошук синергії між Kanban і Scrum призвів до побудови нової гнучкої методології всередині компанії, що вельми позитивно вплинула як на роботу команди, так і на продукт та прибутки компанії.

Висновки. Оптимізація розробки та підтримки програмного продукту в сучасних ринкових умовах вимагає цілісного підходу, що інтегрує методології Agile, зокрема Scrum і Kanban. Важливим аспектом є розуміння специфічних властивостей кожної методології та розуміння умов їх найкращого використання. Синергія між цими методологіями дозволяє ефективно адаптуватися до змін, стабільно забезпечувати якість і розвиток продукту на високому рівні. Гнучке використання Scrum і Kanban є ефективним інструментом для підтримки й розвитку інноваційних проєктів, що зі свого боку вагомо підвищує конкурентоспроможність компаній на ринку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ:

1. Роберт Мартін Чистий Agile. Харків : Фабула, 2021, 224 с.
2. Девід Дж. Андерсон Канбан. Харків : Фабула, 2021, 288 с.
3. Agile Alliance. URL: <https://www.agilealliance.org> (дата звернення: 06.06.2024)
4. Scrum Alliance. URL: <https://www.scrumalliance.org/why-scrum> (дата звернення: 10.06.2024)
5. Kanban University. URL: <https://kanban.university> (дата звернення: 10.06.2024)

REFERENCES:

1. Martin R. (2021) Chystyj Agile [Clean Agile]. Kharkiv: Fabula. 224 p. (in Ukrainian)
2. Anderson D. J. (2021) Kanban. Kharkiv: Fabula. 288 p. (in Ukrainian)
3. Agile Alliance. Available at: <https://www.agilealliance.org> (accessed June 6, 2024)
4. Scrum Alliance. Available at: <https://www.scrumalliance.org/why-scrum> (accessed June 10, 2024)
5. Kanban University. Available at: <https://kanban.university> (accessed June 10, 2024)