

## ОГЛЯД ГНУЧКИХ МЕТОДОЛОГІЙ В УПРАВЛІННІ ІТ ПРОЄКТАМИ

Сучасний світ вимагає стандартизації підходів до управління проєк-тами. Зокрема, управління ІТ проєктами відбувається як з застосуванням класичного підходу в управлінні, так і з підходами в управлінні, що націлені на роботу з високим рівнем невизначеності. Так, сьогодні надає ряд методологій гнучкого управління проєктами з широким набором інструментів, що направлені на досягнення цілей проєкту. Так, варто відмітити сімейство гнучких методологій Agile, до якого відносять такі ітеративно-інкрементальні методи управління як Scrum, Lean, Kanban, Crystal, eXtremeProgramming (XP) тощо. Project Management Institute (PMI) та Agile Alliance розробили один з найбільш швидко впроваджуваних стандартів Disciplined Agile [1] та сертифікаційну програму PMI Agile Certified Practitioner (PMI-ACP), що має на меті стандартизувати знання з використання гнучких методологій [2, с. 214].

Глобальна економіка розвивається швидкими темпами насамперед за допомогою інноваційних проєктів. У свою чергу, інноваційні проєк-ти, як правило, мають високий рівень невизначеності. В зв'язку з цим, для реалізації ІТ проєктів широкого застосування набули гнучкі або адаптивні методології управління, використання яких надає змогу швидкої реакції на зміни, та ризики, що виникають під час розробки ІТ проєктів та безпосередньо адаптувати сам проєкт під потреби ринку або середовища, в якому відбувається його розробка.

**Scrum** як гнучкий метод було побудовано на теорії емпіризму та ітеративності з основною ідеєю в побудові продукту ІТ проєкту, починаючи з найважливіших його частин. Функціональність ІТ проєкту відсортовується згідно пріоритету та планується у ітеративні проміжки часу, що в методології мають назву Sprint. До реалізації проєкту залучені такі ключові ролі як: власник продукту (ProductOwner), скрам майс-тер (ScrumMaster), команда розробки (DevelopmentTeam). Опис завдань функціоналу визначено поверхнево у вигляді користувацьких історій (UserStories), що дозволяє команді розробки проводити досить гнучку його реалізацію. Кожна ітерація розробки – спринт (Sprint) завер-шується показом реалізованих частин функціоналу його замовнику та власнику продукту, та безпосередню доставку робочого інкременту продукту. Це надає змогу власнику продукту провести аналіз реалізованих частин проєкту, та адаптувати завдання на наступну ітерацію у відповідності до побажань замовника. Таким чином замовник може безпосередньо відслідковувати реалізацію ІТ проєкту, та впливати на її перебіг шляхом внесення змін у список вимог задля досягнення цілей проєк-ту та максимальної цінності від його реалізації. Скрам майстер відслідковує дотримання виконання процесів, що чітко визначені скрам фреймворком [3; 4].

**Lean** є доволі абстрактною методологією, проте ефективним підходом до управління, що направлена на оптимізацію бізнес-процесів за рахунок орієнтації на потреби замовника, а також врахування мотивації співробітників залучених до реалізації проєкту. Робота розбивається на невеликі частини, та розробка кожної частини відбувається незалежно та окремо. До принципів Lean відносять:

- 1) Визначення цінності. Аналіз цінностей замовника або ринку, та включення цінностей до продукту.
- 2) Створення карти потоку цінностей. Підхід побудови схематичної карти всіх етапів, що приведуть продукт до замовника.
- 3) Забезпечення безперервного потоку. Безперервний потік послуг або продуктів від початку до кінця проєкту на через процеси.
- 4) Витягування продукту. Роботи на одному етапі процесу розпочинаються лише за умов потреби використання продуктів його виходів на вході наступного етапу. Тобто попит тягне продукт або послугу через потік цінностей.

5) Робота над удосконаленням. Максимальне зниження витрат, щоб усі дії приносили цінність клієнту через безперервне покращення або нові відкриття [5].

У Lean відсутні чіткі межі етапів тому є можливість реалізації де-кількох частин проєкту паралельно, при чому кожна з частин може знаходитись на різних етапах. Досить часто Lean комбінують з використанням методу **Kanban**, що реалізує принцип «вчасно» (just in time). На від-міну від Scrum, Kanban не має чіткого визначення залучених ролей, проте також має роль власника продукту, що забезпечує створення карток завдань для реалізації проєкту. Проте Kanban не обмежує час ітерацій (спринтів). Канбан направлений на візуалізацію роботи та організацію безперервного потоку завдань.

Основні ключові практики методу Kanban:

1) Візуалізація. Побудована на ідеї прозорості поточного стану розробки проєкту за допомогою дошки з колонками етапів реалізації, та карток завдань проєкту що переміщуються між етапами. Таким чином, можна бачити завдання та етап, на якому воно знаходиться.

2) Обмеження кількості задач на етапах. Кожен етап має конфігуровані обмеження по мінімальній та максимальній кількості завдань, що можуть знаходитись на даному етапі. Кожний наступний етап забезпечує попит на початок реалізації завдань у попередньому етапі. Таким чином, кожний наступний етап «втягує» старт виконання завдань на попередньому етапі реалізації.

3) Керування потоком. Перехід між станами постійно відслідко-вується, вимірюється та звітується. Згідно даних відбувається оновлення карти реалізації проєкту, та відбувається аналіз змін у системі.

4) Явність потоку. Ефективність системи не буде досягнуто допоки процес не стане явним усім залученим до нього учасникам. Потребує обговорення, явного розуміння та об'єктивного розгляду проблем.

- 5) Цикли зворотного зв'язку. Необхідні для впровадження вдосконалення процесу та організації загалом.
- 6) Вдосконалення через співпрацю та експериментальний розвиток. Еволюція з використанням наукових підходів до здійснення змін [6; 7].

Вибір ефективної методології управління проектом відіграє одну з ключових ролей в реалізації ІТ проекту, адже в залежності від неї буде окреслено інструменти та процеси, що будуть застосовані під час його реалізації. А також визначено спосіб, у який буде відбуватися робота спеціалістів, залучених до реалізації проекту.

#### Список використаних джерел

1. Ambler, S., & Lines, M. *Choose your WoW: A Disciplined Agile Delivery Handbook for Optimizing Your Way of Working*. Project Management Institute, 2020
2. Jovanović M. et al. *The Agile approach in industrial and software engineering project management* //Journal of Applied Engineering Science. – 2015. – Т. 13. – No. 4. – С. 213-216
3. Schwaber K, Beedle M. *Agile software development with Scrum*. Upper Saddle River: Prentice Hall; 2002 Feb.
4. Кон, Майк. *Scrum: гибкая разработка ПО*. Litres, 2019.
5. Ballard G, Tommelein I. *Lean management methods for complex projects*. Engineering Project Organization Journal. 2012 Jun 1;2(1-2):85-96
6. Kniberg H, Skarin M. *Kanban and Scrum – making the most of both*. Lulu. com; 2010.
7. Kanban. Just-In-Time at Toyota. Management Begins at the Workplace Hardcover – December, 1986.