

# ОСОБЛИВОСТІ ВІЗУАЛІЗАЦІЇ РОЗРОБКИ ПРОЕКТІВ ЗА SCRUM МЕТОДОЛОГІЄЮ ДЛЯ РОЗПОДІЛЕНИХ КОМАНД

Вінницький національний технічний університет

## *Анотація*

*У даній роботі розглянуто основні способи та особливості візуалізації прогресу виконання завдань по розробці проекту за технологією Scrum для розподілених команд.*

**Ключові слова:** agile, scrum, інформаційні технології, методологія розробки.

## **Abstract**

*This paper describes the main methods and features of visualizing the progress of the tasks development by means of Scrum for distributed teams.*

**Keywords:** agile, scrum, information technology, methodology development.

## **Вступ**

Актуальність теми дослідження зумовлена тим, що існує необхідність в інструментах візуалізації управління проектом з допомогою Agile-методів, зокрема Scrum методології, для розробників ПЗ, оскільки такі інструменти допомагають в написанні якісного програмного продукту.

Гнучка розробка програмного забезпечення (англ. Agile software development, agile-методи) — клас методологій розробки програмного забезпечення, який базується на ітеративній розробці, тобто розробці, в якій вимоги та розв'язки еволюціонують через співпрацю між багатофункціональними командами здатними до автономності та самоорганізації. Гнучка розробка — засіб для підвищення продуктивності розробників програмного забезпечення [1].

Можна сказати, що Scrum у певній мірі є фреймворком, що надає спектр можливостей для продуктивної і творчої розробки продуктів з максимально можливою цінністю і рішенням нетривіальних завдань в процесі роботи [2].

## **Scrum як методологія розробки**

Scrum є своєрідним каркасом процесу, що включає в себе набір попередньо визначених ролей та методів. До головних дійових осіб відносяться:

- Scrum Master – особа, в обов'язки якої входить нагляд за процесами та їх ведення. Scrum Master функціонує у якості керівника проекту.

- Власник Продукту – особа, яка повинна представляти інтереси зацікавлених сторін по розробці продукту.

- Команда – група компетентних осіб, що складається безпосередньо з розробників.

Впродовж кожного спринту – 2-4 тижневого періоду (тривалість визначається кожною командою індивідуально) – працівники створюють функціональний приріст (інкремент) програмного забезпечення.

Набір особливостей, які імплементуються після кожного спринту, приходять з етапу, який носить назву Product backlog (список задокументованих завдань усього проекту). Завдання в Product Backlog, як правило, мають найвищу пріоритетність за рівнем вимог до роботи, яка має бути виконана. Елементи беклогу (backlog items) визначаються протягом наради з планування спринту та переносяться в етап спринту. Народа має на меті донести інформацію від Власника Продукту про завдання, які треба виконати першочергово. Команда оцінює, скільки з заявленого вони можуть встигнути зробити за визначений період, щоб завершити найнеобхідніші роботи до початку наступного спринту. Впродовж спринту команда повинна виконати раніше визначений список завдань, що є фіксованим, тобто ніхто не має права його змінювати до наступного планування.

## Огляд існуючих способів візуалізації

Найрозповсюдженішим інструментом для візуалізації процесу розробки є так звані «дошки завдань». Дошки як підхід до візуалізації існували в багатьох галузях і задовго до ери Agile.

Для Scrum команди створюється дошка, що ділиться на три колонки: «Заплановано», «У роботі» і «Виконано». Беруться паперові картки, на яких пишуться всі історії і завдання. На момент планування ітерації всі картки поміщаються в колонку «Заплановано» відповідно до пріоритетів (зверху вниз). Кожен день, коли хтось говорить «я почав працювати над ...» картку із завданням переносять в колонку «В роботі». На наступному щоденному плануванні (Daily Scrum [3]), кожен, хто каже «я закінчив роботу над ...» переміщує відповідну картку в колонку «Виконано».

Ідея візуалізації на дошці гранично проста, і її можна використовувати, щоб якісно поліпшити наочність розробки всього проекту. Наприклад, Власник Продукту може вести у себе окрему дошку, на якій плануються релізи, або просто історії користувачів згруповані по модулям та ролям. На ту ж дошку завдань команди можна кріпити графіки та інші візуальні елементи. На картках можна робити позначки про пріоритети, модулі, відповідальних за виконання, оцінки і т.д. Можна не обмежувати себе трьома колонками, а додати ще колонки в залежності від етапів, які проходять завдання команди розробки. Допускається варіант колонки під назвою «зроблено розробниками», і тоді «завершено» означає «протестовано».

Все це працює майже ідеально для команди, яка сидить разом. У той же час, більшість команд є розподіленими.

Беклог продукту для розподіленої команди можна зберігати в документі із загальним доступом. Це може бути MS Excel Spreadsheet або Google Docs Spreadsheet, але дані рішення не є ефективними, оскільки вони не передбачають управління беклогом через Drag-and-drop, та не мають діаграми згорання задач на ітерацію.

## Висновки

Таким чином, встановлено, що впровадження методології Scrum для розподілених команд, у разі неспроможності ведення «дошок завдань», має супроводжуватись шляхом використання сторонніх програмних засобів, зокрема інтерактивної системи, яка б імітувала роботу «дошки» у якості веб-інтерфейсу, де був би реалізований весь необхідний функціонал для скрам-команд, зокрема створення беклогу продукту, створення беклогу спринта, можливість створення завдань з маніпуляцією їх статусів, таких як «Заплановано», «У роботі» і «Виконано». Система повинна мати клієнт-серверну архітектуру, оскільки це одна із головних переваг, що ліквідують основний недолік розподілених команд, а саме – роботу на віддаленому доступі.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Гнучка розробка програмного забезпечення [Електронний ресурс]. – Режим доступу до матеріалу: [https://uk.wikipedia.org/wiki/Гнучка\\_розробка\\_програмного\\_забезпечення](https://uk.wikipedia.org/wiki/Гнучка_розробка_програмного_забезпечення)
2. Швабер К. Авторитетний посібник зі Скраму: Правила Гри [Електронний ресурс] / К.Швабер, Дж. Сазерленд. – Режим доступу: <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-UA.pdf>
3. What is a Daily Scrum? [Електронний ресурс]. – Режим доступу до матеріалу: <https://www.scrum.org/resources/what-is-a-daily-scrum>

**Столярик Павло Олегович** – студент групи ІПІ-136, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: [pavelstoliaryk@gmail.com](mailto:pavelstoliaryk@gmail.com).

Науковий керівник: **Романюк Оксана Володимирівна** – к.т.н., доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: [romaniukoksanav@gmail.com](mailto:romaniukoksanav@gmail.com)

**Stoliaryk Pavlo O.** – Faculty of Information Technology and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, email: [pavelstoliaryk@gmail.com](mailto:pavelstoliaryk@gmail.com).

Supervisor: **Romaniuk Oksana Volodymyrivna**, Candidate of Technical Sciences, Docent of Software Department, Vinnytsia National Technical University, Vinnytsia, e-mail: [romaniukoksanav@gmail.com](mailto:romaniukoksanav@gmail.com)