

Практична цінність поєднання Scrum техніки розробки програмного забезпечення з методами Six Sigma

Карун О. В.

*аспірант кафедри управлінських технологій,
ВНЗ «Університет економіки та права «КРОК», м. Київ, Україна,
e-mail: olegkarun@krok.edu.ua*

Протягом останніх років використання гібридних методологій стало пріоритетним для компаній так команд з розробки програмних продуктів. Згідно з дослідженнями Project Management Institute однією з основних переваг стає використання гібридних методологій управління проєктами [1]. Завдяки своїй природі, гнучкі методології мають потенціал для інтеграції з іншими методологіями або конкретними управлінськими підходами. Огляд літератури показує, що поєднання методологій розробки програмного забезпечення у більшості випадків мають за ціль досягти конкретних цілей [1]. Разом з тим, на практиці команди поєднують переваги окремих методологій, щоб використовувати свої сильні сторони та отримувати переваги різних методів. Поєднання сталих методологій з новими стандартами та ідеями щодо продуктивності та управління, можуть призвести до підвищення ефективності команди, рівня задоволення клієнтів та покращення процесу розробки програмного забезпечення [2]. Гнучкі методології мають низку переваг, але відсутність формалізації процесів, структурованого збору та обробки інформації, часто, призводять до великої кількості дефектів у програмному коді, та, як наслідок, вимагають повторну ітерацію для усунення фрагментів коду з критично низькою якістю [3]. У дослідженні пропонується розглянути Six Sigma для зменшення дефектів програмного продукту, удосконалення процесів та мінімізації циклів розробки, що слугують переробленню неякісних фрагментів. Для цього будуть запропоновані та проаналізовані ключові інтеграції Six Sigma в підхід управління проєктами Scrum, як найпоширенішого представника гнучкої методології Agile. Фреймворки Scrum і Six Sigma були об'єднані, щоб дослідити ефективність використання мислення та стратегії Six Sigma під час виконання програмних проєктів.

Фреймворк Scrum побудований на ітераційній розробці продукту протягом короткочасних інтервалів, які називаються спринтами. У цій техніці під час розробки програмного продукту розглядаються три ключові ролі: майстер Scrum, власник продукту та члени команди розробників. Функції в команді розробників програмного забезпечення розподіляються на ці ролі [2]. Завдяки належній гнучкості методології та її головній ролі в управлінні проєктом, багато аспектів розробки продукту не визначені і команди можуть вільно визначати їх на основі своїх потреб і можливостей. Тому цей підхід має високий потенціал для поєднання з іншими методологіями, стандартами та підходами до управління [3].

Процес Scrum умовно можна розділити на три етапи, де перший етап-підготовка середовища реалізації проєкту, збір вимог і опис початкової архітектури продукту. Друга фаза присвячена ітераційному процесу розробки продукту. У кожній ітерації частина продукту готується і може бути доставлена. Третій крок наголошує на інтеграції розроблених частин і тестуванні кінцевого продукту та доставлення замовнику [3].

Six Sigma-це дисциплінована методологія управління якістю, керована даними, яка використовується для усунення та зменшення дефектів у виконавчих процесах

та є однією з успішних стратегій в галузі, яка приділяє особливу увагу досягненню конкретних цілей, таких як продуктивність процесу, задоволеність споживачів, зниження витрат і підвищення якості [4]. Six Sigma розглядається як стратегія організаційної трансформації, що веде до розширення та розвитку методів управління в організації. У цій стратегії вимірювання та оцінка рівня покращення є необхідністю, щоб допомогти менеджерам приймати рішення щодо вдосконалення процесу. Кінцевою метою Six Sigma є реалізація стратегії, заснованої на вимірюванні, яка зосереджена на вдосконаленні процесу та зменшенні якісних відхилень шляхом впровадження механізмів відновлення Six Sigma [5].

Цілі та цінності, на основі яких була визначена та розроблена Six Sigma, узгоджуються з цінностями та цілями, які переслідуються в процесах гнучкої методології Scrum. Поєднання цих двох підходів може мати позитивний ефект вирішенні поставлених завдань [6]. Негативним аспектом впровадження запропонованих інструкції із Six Sigma може бути додаткове навантаження на гнучкі процеси розробки програмного забезпечення та віддалити їх від ідеалу.

Оскільки основою Six Sigma є вимірювання та вдосконалення на основі вимірювань, цю вимогу запропоновано реалізувати відповідно до етапів техніки Scrum.

Першим кроком Six Sigma є визначення початкових вимог до реалізації проєкту, цілях проєкту, оцінці фінансових ефектів та продуктивності членів команди та легко інтегрується у підготовку середовища реалізації проєкту процесу Scrum.

Другий крок-вимірювання дозволяє використати діаграму SIPOC для визначення відповідних елементів проєкту, які потребують вдосконалення [2]. Причинно-наслідкову діаграму та аналіз рекомендується використовувати для визначення проблем, що виникають під час спринту. Такі дані можна отримати із практик Scrum: щоденної, оглядової та ретроспективної зустрічі [3]. Рекомендується використовувати відповідні матриці для аналізу досягнень та затрат, а також пріоритетів, які визначає замовник. Нарешті, для забезпечення точності процесу збору даних необхідно визначити систему вимірювання та перевірки, що може базуватися на визначених завданнях.

Третій крок Six Sigma-фаза аналізу. Аналіз «дерева відмов» можна використати для визначення першопричин проблеми, щоб знайти першопричину дефекту. Крім того, регресійний аналіз дозволяє моделювати зв'язки між вхідними та вихідними змінними процесу та їх залежністю. Це може бути використано для аналізу позитивних чи негативних подій, що відбулися в процесі розробки, і дозволяє виявити недоліки та усунути їх [5].

На етапі покращення Six Sigma створюється виконавчий план, у якому визначено заходи, які необхідно вжити під час щоденної, оглядової та ретроспективної зустрічі. Оновлення карти SIPOC і початкової карти процесу з урахуванням реалізації виконавчого плану дозволить покращити процеси та знизити кількість дефектів, разом з тим зменшити необхідність повторних ітерацій розробки. Також на цьому етапі має бути оцінена та проаналізована ефективність вжитих заходів з покращення [6].

На етапі контролю рекомендується створити план контролю на основі аналізу критичних вхідних даних. Такий план може бути використаний як інструмент контролю та моніторингу ймовірних покращень, що відбулися в процесах та в роботі окремих членів команди.

Для інтеграції Six Sigma в процес розробки пропонується використовувати такі

інструкції:

- впровадження DMAIC в рамках спринта;
- використання карт процесу для створення беклогу;
- визначення причин дефектів або помилок під час ревію та ретроспективних зустрічей;

- пріоритезація беклогу продукту з огляду на очікувані замовником покращення.

Виходячи зі сказаного вище, можемо зробити висновок, що використання гібридних методологій у розробці програмного забезпечення є одним із рішень досягнення більшої конкурентоспроможності та вищої якості програмних продуктів. Ця комбінація може включати методології, підходи, стандарти якості та управління. Scrum є однією з найпопулярніших технік методології Agile, та в силу високої гнучкості має ряд недоліків, одна з яких висока кількість дефектів та повторних ітерацій розробки. Методологія Six Sigma є одним з управлінських підходів, які допомагають компаніям досягти вищої якості та більшої задоволеності споживачів шляхом удосконалення процесів виробництва та обслуговування. У цьому дослідженні було запропоновано та проаналізовано поєднання технік Scrum і Six Sigma. Наведені приклади гібридизації можуть призвести до підвищення продуктивності команди та рівня задоволення замовника. Однак кількість виявлених дефектів, доробок і тривалість реалізації проєкту можуть збільшитися з огляду на ресурси, необхідні для впровадження запропонованих інструментів. Кількість дефектів та доробок будуть рости на початку проєкту, проте, аналіз і робота над процесами мають поступово знижувати їх кількість. Такі недоліки будуть виправданими у досягненні критеріїв якості та забезпеченні високого рівня задоволення замовника. Загалом використання методів Six Sigma в процесі Scrum є органічними та злагодженими, а синергія цих двох підходів веде до кращого досягнення цілей цих технік.

Список використаних джерел

1. *Project Management Institute. Industrial management. The Future of Work: Leading the Way with PMTQ. 2019, P. 4-5.*
2. *Abrahamsson P., Salo O., Ronkainen J., Warsta J., Agile software development methods: Review and analysis. 2017, P. 145-178*
3. *Kuhrmann M., Hybrid Software Development Approaches in Practice: A European Perspective. 2019. Vol. 36, No 4, P. 20-38*
4. *T. Pyzdek and P. Keller. Estados Unidos: Mcgraw-hill. The Six Sigma. 2003. P 480-794*
5. *Grant D., Mergen A. E., Total Quality Management. Towards the use of Six Sigma in software development. 2009. Vol. 20, No 7, P. 705-712*
6. *A. M. E. Hamid. 2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEE). Application of Six Sigma Methodology in Software Development». 2018. P. 1-5.*