

Список використаних джерел

1. Оптимізація управління запасами в багаторівневому ланцюгу постачань. URL: <https://abmcloud.com/uk/optimizatsiya-upravlinnya-zapasami-v-bagatorivnevomu-lantsyugu/> (дата звернення: 30.11.2023).
2. Nahmias S. Analysis of production and operations, 6^a edn. McGraw W-Hill / InterAmerican Editors, S. A. de C. V., Mexico. 2014. P. 205.

УДК 004.01:005

*Куцмай В. Я., здобувач 3 курсу спеціальності 122 Комп'ютерні науки,
Волонтир Л. О., канд. техн. наук, доцент, доцент кафедри інформаційних
технологій*

ОПТИМІЗАЦІЯ ПРОЦЕСІВ УПРАВЛІННЯ ПРОЄКТАМИ В ІТ-СФЕРІ

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасному світі, коли технології стрімко розвиваються, ІТ-проєкти стають ключовим елементом успішного функціонування підприємств. Швидкі темпи змін і висока конкуренція вимагають від компаній ефективних методів управління проєктами для забезпечення високої якості продукції та своєчасного впровадження інновацій. Оптимізація процесів управління проєктами в ІТ-сфері стає ключовим елементом успіху, сприяючи підвищенню продуктивності, зниженню ризиків та вдосконаленню комунікацій.

Розглянемо основні методи оптимізації запасів, які знаходять своє використання в роздрібній торгівлі та мають значимий вплив на бізнес.

1. Використання методів Agile в управлінні проєктами

Використання методів Agile в управлінні проєктами в ІТ-сфері є однією з найбільш перспективних і ефективних стратегій для досягнення успішності та гнучкості в реалізації проєктів. Agile – це філософія та набір практик, які спрямовані на реалізацію ітераційних та інкрементальних змін у процесах розробки, зокрема в ІТ-проєктах. Використання методів Agile в управлінні проєктами в ІТ-сфері має декілька ключових аспектів, які сприяють оптимізації процесів.

Методи Agile сприяють гнучкому та адаптивному підходу до управління проєктами, що особливо важливо в умовах швидкозмінюваного ІТ-середовища. Класичні методи управління проєктами, які базуються на жорсткому плануванні та фіксації вимог, можуть виявитися неефективними у змінних умовах ринку та технологічного прогресу. Agile дає змогу командам більш ефективно реагувати на зміни та швидко адаптуватися до нових вимог.

Аналізувати сезонність та патерни попиту є критично важливо. Наприклад, у сфері роздрібного продажу одягу сезонні колекції можуть вимагати більших запасів перед початком сезону.

2. Впровадження інструментів управління завданнями та звітністю

Інструменти управління завданнями надають можливість точно розподіляти завдання між членами команди. Наприклад, платформи, як-от Jira чи Asana, дають змогу визначити конкретні завдання, призначити їх відповідальним співробітникам та встановити терміни виконання. Це не лише забезпечує чіткість щодо обов'язків, але й дає змогу ефективно відслідковувати прогрес кожного завдання.

Інструменти управління завданнями також відіграють ключову роль у створенні прозорості в процесах проєкту. Вони дають змогу генерувати різноманітні звіти щодо прогресу, витрат, термінів виконання та інших параметрів. Це дає змогу керівництву та всім учасникам проєкту отримувати необхідну інформацію в реальному часі, що сприяє швидкому виявленню можливих проблем та прийняттю своєчасних рішень.

Вони створюють сприятливе середовище для покращення комунікації між членами команди, допомагають обмінюватися інформацією, коментувати завдання, вирішувати питання та обговорювати стратегії в межах конкретних завдань. Це сприяє збільшенню взаєморозуміння та покращенню співпраці, що є критичним для успішності IT-проєктів.

3. Вдосконалення процесів тестування та якості програмного забезпечення

Використання автоматизованих тестів допомагає прискорити процес тестування та забезпечити його повторюваність. Автоматизовані тести можуть бути запущені автоматично під час кожної зміни коду, що дає змогу виявляти помилки раніше та зменшує час, потрібний для тестування. Це особливо важливо в умовах Agile-розробки, коли вимоги можуть змінюватися на ранніх етапах проєкту.

Впровадження CI / CD-процесів допомагає автоматизувати процеси інтеграції коду та впровадження змін у продукт. Це сприяє швидкому виявленню конфліктів та помилок, а також забезпечує можливість швидкої доставки нових функцій або виправлень користувачам.

Використання інструментів моніторингу та профілювання допомагає виявляти проблеми з продуктивністю та оптимізувати шляхи в коді. Це допомагає забезпечити ефективність програмного забезпечення та уникнути проблем, які можуть виникнути в продукційному середовищі [1].

4. Ефективне управління змінами та ризиками

Перший крок в ефективному управлінні ризиками – це ідентифікація та аналіз можливих небезпек. Команда проєкту повинна активно займатися пошу-

ком потенційних загроз та визначенням їх впливу на проєкт. Це може бути здійснено шляхом проведення SWOT-аналізу (Strengths, Weaknesses, Opportunities, Threats) та використання методів, як-от аналіз Fishbone (Ishikawa), або дерево визначення ризиків. Чітка ідентифікація допомагає побачити зміни та ризики заздалегідь, щоб вони не впливали негативно на виконання проєкту.

На основі ідентифікованих ризиків необхідно розробити план управління ризиками. Цей план містить стратегії зниження, прийняття, перенесення чи уникнення ризиків. Для кожного ризику треба визначити відповідальних за управління та контроль ризиками, а також визначити етапи реагування в разі виникнення негативних сценаріїв. Чітко сформований план допомагає команді ефективно реагувати на ризики та впроваджувати відповідні заходи ще до їх виникнення [2].

Зміни в процесі роботи нерідко необхідні для вдосконалення проєкту чи адаптації до нових вимог, однак важливо ефективно управляти цими змінами. Використання стандартів управління змінами, як-от ITIL (Information Technology Infrastructure Library) або CMDB (Configuration Management Database), допомагає контролювати та документувати всі зміни, визначати їх вплив на проєкт і забезпечувати комунікацію між усіма сторонами.

Ефективне управління змінами та ризиками потребує чіткої комунікації. Важливо відкрито обговорювати будь-які потенційні ризики з учасниками проєкту та стейкхолдерами. Застосування методів, як-от регулярні збори, статус-звіти та комунікація через спільні платформи, сприяє розумінню всіх сторін щодо поточного стану проєкту та змін, що можуть виникнути.

Список використаних джерел

1. Sutherland J. Scrum. Learn to do twice as much in less time. Crown Business, New York City, U.S. 2014. P. 280.
2. Anderson D. J. Kanban: Successful Evolutionary Change for Your Technology Business. Blue Hole Press, 2010. P. 261.