

автоматизувати більш складні процеси та забезпечать гнучкість і масштабованість в мінливих умовах ринку.

Список літератури

1. Александер Д. Фінансове планування і аналіз та управління ефективністю. Нью Джерсі: Wiley & Sons, Inc. 2018. 612 с.
2. Шклярук С.Г. Проектне фінансування: навчальний посібник: за наук. ред. проф., д-ра екон. наук. Г. К. Ялового. Київ: ДП «Видавничий дім «Персонал». 2009. 472 с.
3. Поллітт К. Інтеграція фінансового менеджменту та управління ефективністю. OECD 2001. URL: <https://www.oecd.org/governance/budgeting/43515173.pdf> (дата звернення 03.04.23).

УДК 005.8 : 004.4

Борисов О.В.¹, Данченко О.Б.¹, Мисник Б.В.²

¹Черкаський державний технологічний університет

²Черкаський національний університет ім. Б. Хмельницького

ОСОБЛИВОСТІ РЕСУРСНОГО УПРАВЛІННЯ ПРОДУКТИВНИМИ ІТ-ПРОЄКТАМИ

Управління проектами орієнтованими на розробку і довготривалу підтримку продукту не використовує підхід який передбачає планування на весь життєвий цикл проекту [1]. Тому загальноприйнятим підходом у таких випадках є ітеративне планування що наслідуює підхід Agile. В основі цього підходу лежить розділення всього терміну проекту на ітерації, кожна з яких планується окремо по завершенню попередньої ітерації. Часовий інтервал

ітерації залежить від конкретного продукту та може обиратися у межах 5-9 тижнів. Кожна з ітерацій проєкту планується подібно до самостійного проєкту, за виключенням команди, стейкхолдерів, ризиків та мети проєкту, що зберігається, з можливістю модифікацій, протягом усього проєкту [2].

Ініціація такого проєкту відбувається подібно до класичних проєктів.

Визначається мета проєкту, потреба чи проблема, яку він має вирішити. За проєктом призначається керівник та створюється статут. Формується команда, та початковий список вимог до проєкту. Після успішної ініціації проєкту, він переходить до ітеративної роботи.

Стандартна ітерація включає в себе чотири групи процесів що складають її життєвий цикл. Ці процеси подібні до класичних процесів управління обмеженими у часі проєктами, тим не менш вони мають і ряд відмінностей.

Спершу відбувається етап планування — керівник проєкту та команда проєкту працюють разом, щоб спланувати всі кроки, визначити набір вимог що мають бути втілені протягом етапу, спланувати розподілити задачі між учасниками, визначити бюджет ітерації.

По завершенню етапу планування, ітерація переходить до етапу розробки. Коли план ітерації створено, команда проєкту приступає до виконання плану ітерації. За потреби, повернення до планування ітерації можливе протягом усього її періоду.

Протягом всієї ітерації відбувається моніторинг і контроль — оскільки ітерація виконується командою проєкту, керівник проєкту відстежує та контролює роботу щодо часу, вартості, обсягу, якості та інших факторів ітерації. Моніторинг і контроль є безперервними процесами, щоб переконатися, що ітерації відбувається згідно власним, короткостроковим, цілям і довгостроковим цілям проєкту.

Останнім етапом ітерації є закриття. Робота завершена та схвалена, результат ітерації успішно передано у загальне користування.

Ітеративне управління проектами має на меті надати робочий продукт протягом кількох тижнів. Кожен цикл розробки закінчується можливістю зібрати відгуки від критично важливих зацікавлених сторін або всередині команди, що дозволяє команді реагувати на будь-який внесок або зміни в наступній ітерації.

Оскільки ітерації будуються одна на одній, команда може легко відстежувати та вимірювати прогрес між ними. Цей метод також сприяє тісній співпраці між членами команди та іншими зацікавленими сторонами протягом усього проєкту, зменшуючи ризики відокремлених команд [3].

При переході від ітерації до ітерації, продуктова ІТ-компанія може потребувати розширення штату, крім того, деякі з прийнятих раніше підходів потребують модифікації для збереження ефективності роботи співробітників. Одним з таких підходів є онбординг нових співробітників на проєкти.

Розробка і підтримка сучасного ПЗ часто потребує від команди розробки знань технологій, специфічних для конкретних типів продуктів. До цього додається специфіка розроблюваного командою проєкту: код технічної та бізнес-логіки, що успадковується від попередніх версій систем, особливості інтеграції з іншими продуктами та внутрішніми системами компанії і, врешті-решт, актуальна кодова база розроблюваного проєкту. Всі ці фактори відчутно впливають на тривалість періоду між прийомом співробітника на проєкт та початком його ефективної діяльності, якість його роботи, та час витрачений більш кваліфікованими спеціалістами на його навчання та перевірку результатів роботи співробітника. Наразі для адаптації нових співробітників, компанії використовують суміш менторства, для навчання співробітника технічним навичкам, та тренінгів, для ознайомлення його з базовими процесами, правилами та корпоративною культурою компанії.

Менторство. Новачку на проєкті призначається ментор — один з діючих розробників проєкту, який буде допомагати йому в процесі адаптації, знайомити з усіма аспектами роботи, допомагати в оволодінні необхідними технічними навичками, а також знайомити з особливостями розробки до якого приєднується новачок. Ментор має підтримувати активний зв'язок з новачком, давати відповіді на його запитання, надавати матеріали для самостійного вивчення. Недоліками такого підходу є ряд проблем при активному залученні нових працівників до проєкту:

- кількість людино-годин що кваліфіковані працівники мають приділяти навчанню новачків значно зростає з кожним новим співробітником, відповідно все менше людино-годин відводиться на розробку і підтримку проєкту що негативно впливає на строки виконання його цілей;

- потреба у наявності вмій та бажання діючих працівників брати на себе відповідальність менторства, що звужує коло потенційних менторів і обмежує як час їх роботи над проєктними задачами, так і темпи найму нових співробітників.

Також помітним чином на перелік та рівень знань новачка впливає роль та кваліфікація його ментора, і оскільки переважна кількість менторів це виключно технічні спеціалісти, фокус їх навчання спрямовується саме на технічний аспект проєкту, приділяючи недостатньо уваги його бізнес-значенню. Наслідком цього є недостатнє розуміння бізнес-сенсу проєктних рішень до розробки яких він підключається [3].

Лекції та підготовчі матеріали. У компаніях що розробляють масштабні ІТ-продукти розповсюдженою практикою є створення підготовчих програм навчання у вигляді курсів, лекцій для ознайомлення як з технічними особливостями проєкту, так і з корпоративними правилами.

У цього підходу також є і недоліки, основний — поверхневість знань отримуваних новачком, текстовий чи лекційний формат подання інформації

статичний і на відміну від взаємодії з ментором не може адаптуватися під особливості сприйняття того хто навчається.

Отож під час масштабування компанії були виявлені проблеми у процесі адаптації нових співробітників: значне зниження ефективності виконання проєктних задач серед менторів, через високе навантаження роботою з новачками; низький рівень розуміння бізнес-сенсу та бізнес-задач проєкту серед новачків, що призводить до зниження самостійності, здатності до дискусії, вмотивованості новачка, а також до додаткових витрат часу з боку досвідченіших співробітників на роз'яснення поставлених новачку задач.

Список літератури

1. Ari-Pekka Hameri, Project management in a long-term and global one-of-a-kind project // International Journal of Project Management. 1997. №15. p. 151
2. Kurt Bittner, Ian Spence, Managing Iterative Software Development Projects. Addison-Wesley Professional, 2006. 214 p.
3. Борисов О.В., Данченко О.Б., Грабіна К.В. Особливості управління віртуальними командами ІТ-проєктів // Project, Program, Portfolio Management. РЗМ: Тези доповідей VI Міжнародної науково-практичної конференції : [у 2т.]. // Відповідальний за випуск П.О. Тесленко — Том 1. — Одеса. : ІШР 2021. – С. 78–81.
4. Борисов О.В., Данченко О.Б., Грабіна К.В. Мультикультурні команди ІТ-проєктів // Міжнародна науково-практична конференція «Інтелектуальні інформаційні системи в управлінні проєктами та економіці в умовах воєнного стану», Коблево, 13-16 вересня 2022 р. Праці – Харків: ХНУРЕ, 2022. – С. 42-46.