

Лі МінАспірант кафедри інформаційних систем і технологій, orcid.org/0000-0002-9396-2852

Київський національний університет імені Тараса Шевченка, Київ

**ЗАСТОСУВАННЯ ГНУЧКОЇ (AGILE) МЕТОДОЛОГІЇ
ДЛЯ ВИКОНАННЯ МІЖНАРОДНИХ СПІЛЬНИХ НАУКОВИХ ПРОЕКТІВ**

Анотація. Активні методи управління проектами стали одним з найбільш швидко зростаючих і найбільш популярних аспектів управління IT-проектами. Використання гнучких методів у розробці програмного забезпечення може зробити різницю між проектом, який має низькі шанси на завершення, і тим, який дасть результати дуже швидко і продовжуватиме досягати результатів з часом. Проте гнучке мислення ніколи не було призначене для обмеження лише розробкою програмного забезпечення. Застосування цієї концепції управління проектами до процесів та інших типів проектів передбачалося з самого початку. Гнучкі (Agile) методології використовують ітеративний підхід до виконання проекту. На відміну від прямолінійної каскадної моделі, гнучкі проекти складаються з декількох менших циклів – спринту. Кожен з них є проектом в мініатюрі: він має відставання і складається з етапів проектування, впровадження, тестування і розгортання в межах заздалегідь визначеного обсягу робіт. Для виконання міжнародних спільних проектів широко використовується механізм Agile – ітеративний та поступовий розвиток. Цей підхід має дуже велике значення в умовах міжнаціональних розбіжностей та знаходження виконавців проекту в різних країнах, зокрема при виконанні спільних наукових досліджень для підприємств-замовників.

Ключові слова: Agile-методології; канбан-дошка; інтеграція Agile в наукові та освітні проекти

Аналіз літературних джерел

Гнучкі методології використовують ітеративний підхід до розробки програмного забезпечення (рис. 1). На відміну від прямолінійної каскадної моделі, гнучкі проекти складаються з декількох менших циклів – спринту. Кожен з них є проектом в мініатюрі: він має відставання і складається з етапів проектування, впровадження, тестування і розгортання в межах заздалегідь визначеного обсягу робіт.

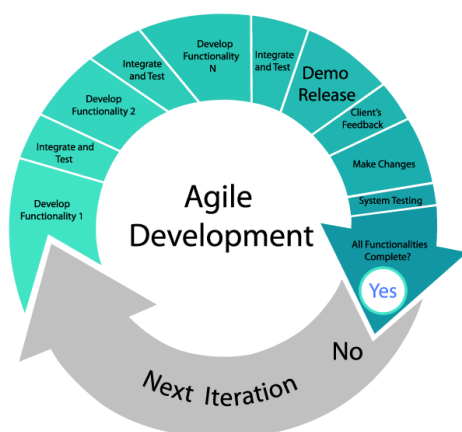


Рисунок 1 – Гнучка методологія управління проектами

Наприкінці кожного спринту доставляється потенційно збільшений приріст продукту. Таким чином, з кожною ітерацією до продукту додаються нові функції, що приводить до поступового

зростання проекту. Оскільки можливості перевіряються на ранніх стадіях в процесі розробки, шанси на постачання потенційно невдалого продукту знижуються.

Враховуючи пріоритетність доступності та швидкого повороту, гнучкий підхід передбачає наступні вигоди, згідно з останніми дослідженнями:

- можливість керувати мінливими пріоритетами (87%);
- підвищену продуктивність команди завдяки щоденному розподілу завдань (84%);
- кращу видимість проекту за рахунок простої системи планування (82%) [1].

Agile Frameworks

Agile – це парасольковий термін для великої різноманітності методологій і методів, які поділяють описані вище принципи і цінності. Кожна з них має свої області використання і відмінні риси.

1. Scrum: ролі, спринти та артефакти

Scrum є домінуючою гнучкою методологією. Виключно її використовують 42% організацій, а ще 54% компаній поєднують її з іншими методами [2]. Вперше описаний у 1986 році Хіротакую Такеучі та Ікуджіро Нонака в грі «Розвиток нових продуктів», цей підхід базується на систематичних взаємодіях між трьома основними ролями: Scrum Master, Product Owner та Team.

Майстер Scrum є центральною фігурою в проекті. Його головна відповідальність полягає в усуненні всіх перешкод, які можуть завадити роботі команди.

Власник продукту, як правило, клієнт або інша зацікавлена сторона, бере активну участь у всьому проекті, передаючи глобальне бачення продукту та надаючи своєчасну інформацію про роботу після кожного спринту.

Команда Scrum – це група людей, що відповідає за реалізацію продукту. Вона повинна складатися не більше ніж з 7 членів команди для того, щоб залишатися гнучкою і продуктивною.

Основна одиниця роботи в Scrum – спринт (рис. 2). Це короткий цикл розробки, необхідний для того, щоб забезпечити приріст продукту, що розробляється. Спринт зазвичай тривалістю від 1 до 4 тижнів: більш тривалі ітерації не мають передбачуваності та доступності, що є фундаментальними основами Scrum. Не маючи стандартної тривалості (до тих пір, поки вона менше чотирьох тижнів, всі спринти в рамках проекту повинні мати зазначену довжину. Це полегшує планування та відстеження прогресу.

Scrum спирається на три основні артефакти, які використовуються для керування вимогами та відстеження прогресу – відставання продукту, відставання у спринті, графік розгортання спринту. Процес формалізується через низку повторюваних зустрічей, таких як щоденний Scrum (Standup), спринтове планування, огляд та ретроспективні зустрічі.

Встановлюючи потреби клієнтів і своєчасну бюджетну доставку як найвищий пріоритет, Scrum має загальний рівень успішності проекту в 62% [2]. Список компаній, що використовують цей підхід, вражає. Відома публічна таблиця з такими організаціями, включно з Microsoft, IBM, Yahoo і Google.

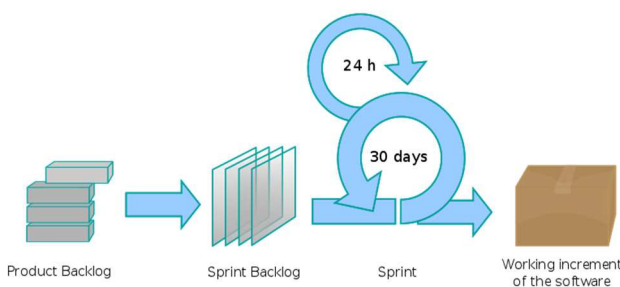


Рисунок 2 – Scrum методологія управління проектами

2. Канбан: комплексне рішення щодо незавершеної роботи

Інша найбільш поширена структура управління проектами – Канбан. Як відомо, в деяких випадках вона використовується 43% організацій, які обрали гнучке моделювання [2]. Виходячи з візуальної системи карт, що використовуються у виробництві Toyota, метод управління виробництвом Канбан є дуже простим, але з потужним підходом до розроблення програмних продуктів.

Канбан надає пріоритет незавершеним роботам

(WIP), обмежуючи його обсяг, щоб ефективно відповідати можливостям команди. Як тільки завдання буде завершено, команда може взяти наступний пункт. Отже, процес розвитку має більшу пріоритетність у плануванні, більш швидкому обігу, чітких цілях та прозорості.

У Канбан не потрібні стандартні процедури, так само як і фіксовані ітерації, на відміну від Scrum. Розроблення проекту базується на роботі команди. Візуалізація відбувається через дошку Канбан (рис. 3). Дошка, як правило, представлена липкими нотатками на розмаркірованій поверхні або онлайн-інструментами, такими як Trello.

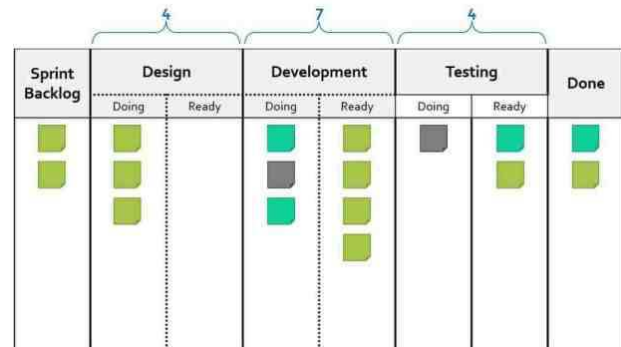


Рисунок 3 – Канбан-дошка для управління проектами

Такі компанії, як Spotify і Wooga (провідна компанія з розвитку мобільних ігор) успішно використовують цей підхід протягом багатьох років. Проте 43% організацій поєднують Scrum з Канбанськими методами, використовуючи скоріше Scrumban, ніж оригінальну методологію [2].

3. Lean: усунення відходів у розробці програмного забезпечення

Lean – третій найбільш широко застосовуваний гнучкий підхід, прийнятий 21% організацій [2]. Маючи таке ж походження, як Канбан, підхід почався як технологічний процес, що застосовувався до фізичного виробництва. Започаткування цього підходу приписується компанії Toyota. Виробнича система, як управлінський підхід, спрямована на те, щоб “доставити транспортні засоби, замовлені клієнтами, найшвидшим та найбільш ефективним способом, якомога швидше” [3].

Застосування принципів Lean до розроблення програмного забезпечення було спочатку запроваджено Мері та Томом Поппендік у своїй книзі Lean Software Development: Agile Toolkit. Вона включає в себе сім основних принципів:

- усунути відходи;
- покращити навчання і створити нові знання;
- приймати рішення якомога пізніше;
- доставляти якомога швидше;
- посилювати команду;
- підвищувати цілісність / якість;
- бачити повну картину [4].

Ці основи прекрасно описують Lean-філософію: її метою є досягнення більшої вартості через менші

витрати, інвестиції та час. З точки зору проекту термін «відходи» означає все, що не додає значення до проекту, а отже, його слід усунути. У інженерії програмного забезпечення це може бути вільний час, непотрібні функції або дефекти.

Розроблення програмного забезпечення Lean – це ітераційна та інкрементальна методологія. Оскільки, як і в будь-якому іншому підході Agile, приріст робочого продукту здійснюється на ранніх етапах процесу розроблення, подальший прогрес залежить переважно від зворотного зв'язку власника продукту.

Що стосується підходу Lean (рис. 4), це те, що команда не обмежується використанням будь-яких формальних процесів, таких як повторні зустрічі або пріоритетність завдань. Однак його принципи часто поєднуються з іншими методами Agile: 21% команд поєднують Scrum з Lean [2].

Ефективно прийнятий великою кількістю виробничих компаній, таких як Nike, Ford і Intel, принципи Lean широко використовуються в інших галузях. Стартапи та успішні компанії, наприклад Corbis, Patient Keeper і Xerox застосовують технології Lean для розробки програмного забезпечення в своїх процесах.



Рисунок 4 – Lean-методологія управління проектами

Країні практики Agile Software Engineering: Extreme Programming

Хоча вищенаведені методика переважно зосереджені на процесі управління проектами та його формальних аспектах, відома велика кількість гнучких методів, які є виключно технічними. Екстремальне програмування (XP) поєднує в собі найважливіше, забезпечуючи гнучкі команди з рядом інструментів для оптимізації технологічного процесу.

Екстремальне програмування – це набір певних практик, що застосовуються до інженерії програмного забезпечення з метою поліпшення його якості та здатності до адаптації до мінливих вимог. Кент Бек один з перших, хто підписав маніфест Agile. Він був першим, хто задокументував цю практику у своїй книзі *Extreme Programming Explained: Embrace Change*.

Найчастіше застосовуються практики XP:

- розробка з використанням тестів (TDD);
- рефакторинг;
- безперервна інтеграція;
- програмування пари.

Тест-Driven Development є передовою інженерною технікою, яка використовує автоматизовані блок-тести для просування процесу проектування програмного забезпечення. На відміну від звичайного циклу розроблення, де тести написані після коду (або взагалі не написані), TDD має перший підхід. Це означає, що модульні тести записуються до самого коду.

Згідно з цим підходом, тест не може проходити, поки немає коду для виконання функції. Після цього інженери пишуть код, зосереджуючись на функціональності, щоб зробити прохід тесту. Як тільки це буде зроблено, необхідно поліпшити вихідний код, щоб пройти всі тести. Ці три кроки часто називають циклом червоно-зеленого рефактора [5].

TDD виявив такі переваги:

1. Тести використовуються для виявлення будь-яких дефектів або помилок в коді, забезпечуючи постійний відгук про стан кожного програмного компонента. Таким чином, якість кінцевого продукту стає все більш високою.

2. Блокові тести можуть використовуватися як сучасна проектна документація, що змінюється в міру розвитку проекту.

3. Будучи глибоко залученою до розроблення продукту, команда повинна вміти критично аналізувати його і передбачати заплановані результати, щоб правильно перевірити його. Це допомагає мотивувати та залучати команду, сприяючи якості продукції.

4. При ретельному початковому тестуванні час налагодження мінімальний.

Окрім того, що використовується в циклі TDD, рефакторинг коду є звичайною практикою в розробці гнучких програм. Здебільшого це процес постійного вдосконалення коду через його спрощення та роз'яснення. Процес є виключно технічним і не передбачає будь-яких змін у поведінці програмного забезпечення.

Розширюючи вихідні коди з кожною ітерацією, гнучкі команди використовують рефакторинг як спосіб уникнути безладу коду та копіювання. Це допомагає запобігти старінню програмного забезпечення, зберігаючи, підтримуючи та розширюючи код.

Безперервна інтеграція (CI) є ще однією практикою, яка використовує гнучкі команди для керування загальним кодом і тестуванням програмного забезпечення. Такі інструменти, як CruiseControl або Jenkins, використовуються для перевірки якості програмного забезпечення та автоматизації його розгортання. На додаток до цього, CI допомагає підтримувати спільний код, усуваючи питання інтеграції. Таким чином, магістраль

продукту є надійною, чистою і може бути швидко розгорнута за необхідності.

Програмування пари, або "сполучення", вважається дуже суперечливою гнучкою практикою. Ця методика передбачає спільну роботу двох інженерів. Хоча один з них фактично пише код, інший активно бере участь як спостерігач, робить пропозиції та керує процесом.

Орієнтована як на кодові, так і на більш абстрактні технічні завдання, ця команда з двох має бути ефективнішою, створюючи кращі розробки програмного забезпечення та роблячи менше помилок. Ще одним плюсом цього підходу є поширення знань про проект серед членів команди.

Однак цю практику часто звинувачують у негативному впливі на короткострокову продуктивність команди. Дослідження показують, що кожне завдання зазвичай вимагає понад 15 – 60% часу [6], що є головним недоліком підходу. Однак існує думка, що додатковий час легко компенсується в довгостроковій перспективі завдяки більш високій якості програмного забезпечення [7].

Гнучкий підхід часто помилково вважається єдиною методологією. Проте відомі десятки методологій і певних практик, які не були порушені в цьому дослідженні.

Незалежно від точних методологій і методів, які вони використовують, Agile-команди довели, що збільшують прибуток на 37% швидше і генерують на 30% більше доходів, ніж не гнучкі компанії [8]. Вища швидкість, гнучкість і продуктивність досягнуті завдяки такому підходу, є ключовими факторами, які стимулюють все більше і більше організацій до переходу на Agile.

Програмна інженерія, будучи надзвичайно швидкою галуззю, закликає до доступності та оперативності у всіх аспектах розроблення проекту. Гнучкі методології допомагають забезпечувати найсучасніші продукти та розвивати інноваційний досвід, зберігаючи при цьому продукт у синхронізації з ринковими тенденціями та вимогами користувачів.

Однак завжди є місце для різноманітності. Залежно від вимог та цілей у бізнесі, все одно можна скористатися водоспадною моделлю або комбінацією цих двох моделей.

Застосування гнучкої методології для проектів, що не належать до програмного забезпечення

Активні методи управління проектами стали одним з найбільш швидкозростаючих і найбільш популярних аспектів управління IT-проектами. Використання гнучких методів у розробці програмного забезпечення може зробити різницю між проектом, який має низькі шанси на завершення, і тим, який дасть результати дуже швидко і продовжуватиме досягати результатів з часом. Проте гнучке мислення ніколи не було призначене для обмеження лише розробкою програмного забезпечення. Застосування цієї концепції

управління проектами до процесів та інших типів проектів передбачалося з самого початку [9].

Однією з ключових проблем у проекті підприємства є глибина змін, яку вона може викликати в організації. Загальною помилкою є створення всеосяжного плану, а потім поштовх до розгортання всього плану відразу. Застосування гнучкого мислення до проектів управління змінами є відмінним.

Agile-управління проектами змушує нас думати про проект спочатку з точки зору великих цілей на стратегічному рівні, а потім на тактичному рівні ми думаємо з точки зору надання готових результатів виробництва.

Проект підприємства є таким, що впливає на операції на підприємстві. Проекти підприємства можуть бути системними проектами. Заміна фінансової системи організації, безумовно, була б прийнятною. Фінансова система не впливає лише на бухгалтерію. Це може вплинути на те, як ми купуємо, як продаємо, як відстежуємо наших клієнтів, як ми продаємо і як підтримуємо наші запаси або поставляємо нашу продукцію. Але проект підприємства не потребує програмного забезпечення. Перехід штаб-квартири організації з однієї будівлі в іншу можна розглядати як такий проект. Корпоративне злиття/придбання теж підходить під цю класифікацію. Створення офісу з управління проектами (PMO) майже завжди підходить під проект підприємства.

Ключовою характеристикою проекту підприємства є зміна культури. Очікується, що проект приведе до зміни поведінки людей в організації.

Проблеми проекту підприємства

Є кілька викликів, які настільки поширені для корпоративних проектів, що мають бути розглянутими.

Проекти підприємства майже завжди недооцінюються.

Незалежно від того, наскільки детально розроблено такий проект, складність проекту підприємства майже завжди недооцінюється. Сама природа проекту робить це майже неминучим. Коли проект призведе до змін у поведінці, зрозуміло, що з розвитком проекту, ті, хто в ньому беруть участь, побачать його з перспективи змін. Проекти підприємств не можуть бути повністю оцінені, доки вони не пройдуть.

Як наслідок, оцінки, які здаються цілком зрозумілими до стартової точки, змінюються внаслідок змін потреби системи. Це може мати серйозні наслідки в кінці проекту, оскільки графік і бюджет перебувають під тиском.

Переміщення офісу – це означає, що нам потрібно перемістити всі ноутбуки. Хіба не було б чудово тоді скористатися цією можливістю, щоб змінити всі ноутбуки?

Ні, це не так.

Зміна фінансової системи – це означатиме, що відділ доставки повинен змінити свої процедури. Чи не було б прекрасним часом замінити всі наші вантажні автомобілі?

Ні, це не так.

Проекти підприємств можуть забрати багато часу і великий відсоток корпоративного бюджету. Це робить його очевидною метою для тих, хто намагається завершити свої проекти.

Якою б не була проблема, її можна вирішити за допомогою технології.

Нині ми дуже залежимо від технологій і можна подумати, що має бути технологічна відповідь на те, що є завданням для підприємства. Гірше, якщо технологія бере участь у проекті підприємства, тоді проект може швидко стати технологією, а не рішенням, розробленим проектом.

Таким чином, зміна фінансової системи стає проектом Oracle або SAP. Перехід до централізованого офісу з управління проектами стає проектом Microsoft Project Server або Primavera або Clarity [9].

Загальноприйнятим для виконавчого спонсора є недооцінка впливу проекту підприємства. Як наслідок, керівництво може не зрозуміти, що ви будете змінювати корпоративну культуру і що це може призвести до розладу. Не треба думати, що керівництво має повне розуміння того, скільки вони будуть потрібні у ключові моменти проекту, щоб допомогти втримати персонал на робочих місцях. Це може бути найскладнішим в культурі, де управління консенсусом є правилом. Консенсус однієї великої щасливої сім'ї, що радує проект підприємства, оскільки він розвивається, не є звичайним явищем.

Чим більше на організацію впливає підприємство, тим більш ризикованим є цей шлях. Кожен день проект є неповним, він виривається на горизонті, як шторм, що наближається до тих, хто постраждав. Це ставить проект під величезний ризик. Щодня він прогресує, з ним щось може статися. Спонсор може змінитися. Компанія може бути продана. Компанія може придбати іншу компанію. Промисловість може змінитися. Економіка може змінитися. Протягом усього цього часу вигоди від проекту є лише теоретичними.

І навіть якщо чудовий день прийде, коли проект буде виконано, то будьте впевнені, що в цей день хтось скаже: "О, але це не те, чого ми дійсно хотіли".

Досвід застосування гнучкої методології на підприємствах

Багато IT-організацій використовують гнучкі методології як основний метод управління проектами розвитку. У більшості організацій гібридне середовище більш традиційного планування та управління проектами співпрацює з гнучкими методами, які більш ймовірно застосовуватимуться на тактичному рівні.

Найбільш помітною перевагою для організацій, що використовують цей підхід, є ітеративний випуск

корисної функціональності. Клієнт починає бачити прибутки від розробки, оскільки кожна частина завершується, а отже проект прогресує, глибина і коштовність продукту зростає.

Менш помітним, але набагато більш важливим аспектом цих середовищ є те, що клієнти, природно, стають невід'ємною частиною процесу, працюючи з командою розробників все більше і більше в міру просування проекту, і вони можуть бачити і коментувати те, що вони отримують.

Ми можемо скористатися цим практично в будь-якому проекті, якщо застосовувати однакоове мислення.

Навіть в офісах з управління проектами лише для програмного забезпечення, звичайні методи управління проектами майже завжди співіснують з гнучкими методами.

Реалізація гнучкого методу означає, що ви не відмовитесь від наявних процесів управління проектами, але характер виконання проектів може в кінцевому рахунку різко змінитися [9].

Приклад того, які аспекти методу Agile можна застосовувати в інших контекстах

Нижче наведено деякі з найбільш популярних прикладних практик, які можна застосувати до корпоративних проектів.

Беклоги

Беклоги – це функції та характеристики, які стануть частиною остаточного проекту. Їх можна розглядати як велику колекцію предметів, які були описані з точки зору того, що вони означатимуть для кінцевих користувачів. Ці елементи призначені для ресурсів у невеликій колекції робіт за дуже короткий час, який називається спринтом.

Спринт

Спринт – це короткий міні-проект всього за кілька днів. Очікується, що всі завдання (предмети беклогів), введені в спринт, будуть завершені протягом тривалості спринту. Робота жорстко управляється всередині самого спринту, команда відчуває, що вони мають велику свободу. Крім того, існує структурна напруга в практиці, щоб завершити всю роботу. Люди, як правило, наполегливо працюють у цих середовищах, щоб не бути єдиною з команд, чия робота не була виконана до закінчення спринту. І, як правило, краще керувати більш короткими завданнями.

Міжфункціональна команда

Концепція міжфункціональних команд в методологіях гнучкості легко переноситься на інші типи проектів. У деяких проектах команди працюватимуть на відстані і виходять працювати один з одним лише у визначені моменти. Проблема, яку розробники програмного забезпечення визначили, полягає в тому, що в ці визначені моменти команди змушені вважати, що їхня робота знаходиться в перехресних цілях з мисленням інших команд, або що деякі роботи були надмірно відтворені іншими командами. Міжфункціональна

команда працює з підвищеною ефективністю і має більш своєрідний фокус.

Легко уявити собі таку структуру, яка добре працює в проекті реорганізації підприємства або в корпоративному проекті злиття підприємств або в проекті переїзду офісу підприємства.

Безперервна інтеграція

Елементи проекту з різних груп повинні бути об'єднані на постійній основі, щоб жоден елемент проекту не став силосом. Візьмемо, наприклад, проект підприємства для переміщення офісу на 1000 осіб. Одна група могла б працювати над інтер'єрними офісними меблями, а інша група працює над офісними комп'ютерними мережами. У традиційному мисленні з управління проектами легко уявити ситуацію, коли ці групи зіставляють свої напрацювання лише після завершення, що призводить до конфліктів. Залучення такого роду спільної роботи під час розвитку проекту, як правило, є набагато ефективнішим.

Ітеративний та поступовий розвиток

Це один з фундаментальних аспектів моделей гнучкості, який є найбільш корисним для всіх типів підприємницьких проектів. Проект підприємства часто характеризується тим, що його важко точно передбачити заздалегідь. Деякі сфери діяльності в проекті практично неможливо оцінити до того, як учасники проекту побачили рішення та проекти, що зроблені раніше. Як наслідок, спроби бути прогностичними у великих масштабах часто є високим ризиком для таких проектів. Натомість адаптивний підхід до розробки раних етапів і повернення, щоб знову і знову вдосконалювати проект, може бути набагато ефективнішим. Повернення до концепції планування рухомих хвиль, як це описано в Керівництві з управління знаннями (PMBOK® Guide) [10], має потенціал для величезної користі.

Засідання Scrum

На засіданнях Scrum проводяться зустрічі, в яких перехресні функціональні команди зустрічаються з фасилітатором (який називається ScrumMaster). Група оновлює хід виконання останнього спринту завдань і перегрупується для наступного спринту. Фокус робиться на беклогах найкращих характеристик, завданнях і питаннях, які було прийнято членами команди, які беруть на себе ті чи інші завдання на наступний короткий період наявного або майбутнього спринта.

Відмінна подія на зборах Scrum – це спосіб, у який проводяться зустрічі. По-перше, вони майже завжди швидкі, оскільки проект був розбитий на згуртований спринт, щоби було легше зосередитися. І, нарешті, стандарт для фасилітатора – не бути учасником. Це означає, що спостерігачі, які не беруть участь у проекті, можуть просувати проект. Навіть ті зобов'язання, які люди беруть на себе, стосуються невеликої кількості пунктів протягом найближчих кількох коротких днів. Кінець спринту завжди знаходиться в полі зору, навіть у перший день, і це

призводить до збереження швидкості проекту на дуже високому рівні.

На відміну від традиційного графіка, де легко заспокоїтися, думаючи, що існує необмежена кількість часу, у спринті ви працюєте ефективніше, тому що проект має лише кілька коротких днів.

Таймбокс

Таймбокс – це термін, який дуже добре знайомий тим, хто цікавиться традиційним управлінням проектами. Він охоплює широкий спектр робіт і вкладає його в графік – бокс часу. Ті, хто знайомі з традиційною методологією критичного шляху для планування розкладу, сприймають цей графік за замовчанням. Ті, хто виховується в більш гнучко-орієнтованому середовищі управління проектами, вважають це винятком, а не правилом. Однак важливо не відмовлятися від традиційних способів управління проектами. Незалежно від того, наскільки далеко ви йдете в прийнятті гнучкої практики для ваших непрограмних проектів, завжди буде місце для того, щоб забезпечити своєчасне і бюджетне використання методів і практик, які стали догматичними в галузі управління проектами.

Випадок використання

Гнучкий випадок використання є дуже часто використовуваним методом опису того, що хтось зробить для виконання функції. Ця методика надзвичайно корисна практично в будь-якому процесному проекті. Керівники проектів часто будуть мати процес, який описується назвою і змістом, але при описі випадку використання кроки, які необхідно виконати, перераховуються один за одним, а кінцевим результатом є завершення процесу. Описуючи процес таким чином, можна усунути багато потенційних пасток у прогнозованій практиці. Легко уявити собі наскільки корисним буде це у корпоративній реорганізації або переїзді в офіс, або зміні будь-якого корпоративного процесу.

Якщо цей вид техніки не застосовується, то виникає великий потенціал для того, щоб процес був реалізований лише з метою, аби дізнатися про процедуру, що стосується чогось фундаментального і критичного для організації. В цьому випадку існує великий ризик для затримки або зриву проекту.

Історія користувача

Якщо сценарій використання описує процес і кроки, що необхідні для завершення цього процесу, то історія користувача є описом бізнес-проблеми. Дуже часто цей тип документації відсутній в неагресивному проекті, але він критично корисний. Історія користувача окреслює мету, з якої саме відбуваються зміни. На жаль, занадто поширеним є бачення проекту, в якому рішення впроваджується без фундаментального розуміння проблеми.

Коли ви створюєте історію користувача, ви описуєте простою мовою, саму проблему і як її можна подолати. Це очевидна вимога, якщо ви бажаєте керувати завершенням проекту. Що може бути більш важливим для керівника проекту?

Викладення основного матеріалу

Багато організацій вже почали застосовувати гнучкі методи в управлінні проектами у сферах, не пов'язаних з ІТ. Цей досвід можна використовувати і в міжнародних освітніх і наукових проєктах. Як і будь-які зміни в корпоративній культурі, більш ефективним є розпізнавання того, які методи прийняти і як швидко їх прийняти.

Не є виключенням і Янченський політехнічний інститут, який інтегрує останні досягнення в

управлінні проєктами та програмами з навчальним та науковим процесами.

Також застосовується структура управління проєктами – Канбан. Як приклад представлена дошка завдань для виконання спільної наукової теми “Розробка газоочисного пристрою для систем вентиляції промислових приміщень”, що здійснюється за замовленням китайського підприємства (таблиця).

Таблиця – Канбан-дошка проєкту з китайським підприємством

Беклог спрінту	Сторони виконавці	Розробка		Розвиток		Тестування		Виконано
		Робиться	Зроблено	Робиться	Зроблено	Робиться	Зроблено	
1	2	3	4	5	6	7	8	9
Підготовка технічного завдання	Китайська сторона		Початкові технічні характеристики	Уточнення технічних характеристик		Узгодження із замовником		
	Українська сторона		Початкова документація	Підготовка супровідної документації		Узгодження із замовником		
Проведення математичного моделювання газодинаміки процесу	Китайська сторона	Інтегрування уточнених технічних характеристик				Узгодження із замовником		
	Українська сторона	Побудова математичної моделі Проведення моделювання газодинаміки потоку		Уточнення метематичної моделі Коригування геометричних характеристик залежно від результатів моделювання		Узгодження із замовником		
Здійснення креслень	Китайська сторона	Робоче проєктування		Робоче проєктування		Узгодження із замовником		
	Українська сторона	Ескізне проєктування		Ескізне проєктування		Узгодження із замовником		
Побудова дослідного зразка	Китайська сторона	Підготовка деталей		Підготовка деталей		Узгодження із замовником		
	Українська сторона	Збірка дослідного зразка		Збірка дослідного зразка		Узгодження із замовником		
Проведення експериментальних досліджень	Китайська сторона	Монтування дослідного зразка		Монтування дослідного зразка		Замовник приймає участь у тестуванні		
	Українська сторона	Проведення експериментів		Проведення експериментів		Замовник приймає участь у тестуванні		

Висновок

Для виконання міжнародних спільних проєктів широко використовується механізм Agile – ітеративний та поступовий розвиток. Цей підхід має дуже велике значення в умовах міжнаціональних розбіжностей та знаходження виконавців проєкту в різних країнах, зокрема при виконанні спільних наукових досліджень для підприємств-замовників.

Також застосовується структура управління проєктами – Канбан. Як приклад у статті представлена дошка завдань для виконання спільної наукової теми “Розробка газоочисного пристрою для систем вентиляції промислових приміщень”, що здійснюється за замовленням китайського підприємства. Головними виконавцями виступають дві сторони – китайська та українська. А замовник бере безпосередню участь у виконанні проєкту як сторона, з якою узгоджується кожен результат спринту – беклог.

Список літератури

1. <https://www.versionone.com/pdf/state-of-agile-development-survey-ninth.pdf>
2. <https://www.scrumalliance.org/scrum/media/scrumsalliancemedial/les%20and%20pdfs/state%20of%20scrum/scrumsalliance-state-of-scrum-2015.pdf>
http://www.toyota-global.com/company/vision_philosophy/toyota_production_system/
3. <http://www.poppendieck.com/>
4. [https://msdn.microsoft.com/en-us/library/aa730844\(v=vs.80\).aspx](https://msdn.microsoft.com/en-us/library/aa730844(v=vs.80).aspx)
5. <http://www.cs.utah.edu/~lwilliam/Papers/ieeSoftware.PDF>
6. https://www.ibm.com/devops/method/content/code/practice_pair_programming
7. <http://www.pmi.org/~media/PDF/learning/pulse-of-the-profession-2015.ashx>
8. Vandersluis, C. (2014). *Apply agile methodology to non-software enterprise projects. Paper presented at PMI® Global Congress 2014—North America, Phoenix, AZ. Newtown Square, PA: Project Management Institute.*
9. *PMBOK® Guide – Sixth Edition (2017) / FOUNDATIONAL STANDARDS*, <https://www.pmi.org/pmbok-guide-standards>.

Стаття надійшла до редколегії 20.03.2019

Li Ming

PhD student Department of Information Systems and Technology, orcid.org/0000-0002-9396-2852
Taras Shevchenko National University of Kyiv, Kyiv

APPLICATION OF THE FLEXIBLE (AGILE) METHODOLOGY FOR THE IMPLEMENTATION OF INTERNATIONAL JOINT SCIENTIFIC PROJECTS

Abstract. Active project management techniques have become one of the fastest growing and most popular aspects of IT project management. Using flexible methods in software development can make a difference between a project that has a low chance of completion, and those that will give results very quickly and continue to achieve results in time. However, flexible thinking has never been intended to restrict only the development of software. Applying this concept of project management to processes and other project types is anticipated from the outset. Agile methodologies use an iterative approach to project implementation. In contrast to the rectilinear cascade model, flexible projects consist of a few smaller cycles of sprint. Each of them is a project in miniature: It has lag and consists of stages of design, implementation, testing and deployment within a predetermined amount of work. In order to carry out international joint projects, the Agile mechanism is widely used-iterative and gradual development. This approach is very important in terms of interethnic differences and finding project performers in different countries. In particular, in the implementation of joint scientific researches for the enterprises-customers.

Keywords: Agile methodology; the Kanban board; the integration of Agile into scientific and educational projects

Link to publication

APA Li Ming. (2019). *Application of the flexible (Agile) methodology for the implementation of international joint scientific projects. Management of development of complex systems*, 38, 103 – 110, [dx.doi.org/10.6084/m9.figshare.9788555](https://doi.org/10.6084/m9.figshare.9788555).

ДСТУ Лі Мін. Застосування гнучкої (Agile) методології для виконання міжнародних спільних наукових проєктів [Текст] / Лі Мін // *Управління розвитком складних систем*. – № 38. – 2019. – С. 103 – 110, [dx.doi.org/10.6084/m9.figshare.9788555](https://doi.org/10.6084/m9.figshare.9788555).