

М. О. Чубенко¹, аспірант, проектний менеджер,
e-mail: chubenko.nikolay.mail@gmail.com

В. І. Семиног², аспірант, розробник програмного забезпечення «UAPAY»,
e-mail: vlad.seminog@gmail.com

О. С. Іларіонов³, к.т.н., доцент
e-mail: oilarionov@gmail.com

¹ Маркетингове агентство «OSD GROUP», м. Київ; тел.: (096) 539-33-66;

² м. Київ; тел.: (063) 596-90-13

³ Київський національний університет імені Тараса Шевченка, м. Київ; тел.: (097) 818-83-07

ВИКОРИСТАННЯ ГНУЧКИХ МЕТОДОЛОГІЙ SCRUM І KANBAN В ІТ-ПРОЕКТАХ

Розглянуті інноваційні методології управління ІТ-проектами SCRUM, KANBAN, виділені їх ключові особливості, характерні риси, окреслена зона ефективного застосування. Проведено аналіз основних завдань і виробничих вимог до мультифункціональних команд, визначений розподіл ролей її членів, та наведені механізми підвищення ключових показників ефективності, показано перетворення членів майбутньої команди у групу спеціалістів, що діють в рамках певної методології та здатні розв'язувати поставлені завдання.

Ключові слова: методологія гнучкої розробки, управління ІТ проектами, розробка програмного забезпечення, ітерація, спринт.

Постановка проблеми. Люди, які займаються управлінням проектами, добре знають, наскільки складно організувати злагоджену роботу в рамках певного проекту або портфеля проектів. Відсутність розуміння між членами команди призводить до порушення плану, відставання від графіку, бюджет проекту роздувається, фінансові інвестиції та час використовуються нерационально. Також варто зауважити, що деякі задачі всередині процесів дублюються, команди використовують свої ресурси неефективно і часто не співпрацюють, а навпаки конфліктують, хоча, на перший погляд, переслідують однакові цілі. Крім цього, продукт після випуску часто може не задовольняти замовника.

Аналіз останніх досліджень і публікацій. Такі проблеми породжувались традиційними або каскадними підходами до реалізації проектів, які пропонували поетапний рух до цілі. Такий підхід має масу недоліків в сучасних умовах, крім того, весь процес розробки проходить дуже повільно, часто з'являються непередбачувані труднощі і, більше того, часто відбувається так, що створюється продукт, який абсолютно не підходить замовнику.

Каскадні моделі передбачають використання графіків, на яких відображаються етапи робіт і час на їх виконання. Всі кроки детально описані і розмічені на шкалі. Такі графіки

називаються діаграмою Ганта, і передбачається, що кожна фаза проекту закінчується і плавно переходить в наступну, що і являє собою принцип каскаду, або водоспаду [1,3].

Плани, звісно, важливі в своїй суті, але не можливо слідувати їм повністю – частіше за все при впровадженні їх в реальне життя, графіки та плани просто “розсипатимуться”.

З удосконаленням функціоналу, який інтегрується в програмне рішення, з'явилася необхідність оперативних змін в структурі проекту. Це виявилось вигідно як розробникам, так і замовнику. Перші отримали можливість оптимізувати свій продукт, виробляти його швидше і якісніше, нехай і невеликими частинами, а другі – адаптувати під свої потреби, бачити швидкі результати і можливість контролювати бюджет проекту, шляхом оплати за вже готові блоки.

Тому, опція додавати в роботу можливість змін, відкриттів та нових ідей, що і відбувається в гнучких методологіях управління проектами Scrum і Kanban, має досить суттєве значення.

Метою даної статті є використовуючи сучасні гнучкі підходи та методики, показати можливість їх застосування на ранніх етапах проекту, що дасть змогу вчасно реагувати на побажання замовника, і виключає створення не потрібного йому продукту або функціоналу.

Виклад основного матеріалу. Підвищення складності проектів, використання нових технологій вимагають створення комплексних команд. Вони складаються з різних спеціалістів, які працюють поруч один з одним.

Команда проекту – це певна чисельність людей, які працюють разом для досягнення спільної мети, і підпорядковані керівникові (менеджеру) проекту. Команда проекту створюється на період реалізації проекту, а після його завершення – розпускається.

Робота у команді може поєднати людей таким чином, що вони підвищують продуктивність своєї праці, не втрачаючи своєї індивідуальності (наприклад оркестр). Командна робота має синергічний ефект, коли опрацьовуються різні пропозиції, надається конструктивна допомога одним членам команди з боку інших, що сприяє досягненню більш високих результатів.

Scrum. Слово Scrum (сутичка) запозичене з гри в регбі. Воно «позначає» метод командної гри, що дозволяє заволодіти м'ячем, і вести його далі по полю, а для цього потрібні злагодженість, єдність намірів і чітке розуміння мети. "Сутичка" являє собою ідеальну модель повного взаємодії гравців. І це саме те, що потрібно для успішної командної роботи. Переважна більшість практиків, які застосовують гнучкі методології, формують спільну думку, що SCRUM це сукупність методів розробки і управління, гармонічно поєднаних між собою, в основі яких лежать принципи виробничої ітеративності, що дозволяють замовнику частіше отримувати певний робочий функціонал в рамках розробки проекту. Також необхідно зазначити «простоту» і інформативність такого підходу, високий рівень комунікації всіх учасників, та відсутність обтяжуючої проектної документації. Ітеративна розробка, в кінці якої створюється готовий до використання продукт, застосовується в усіх гнучких методологіях. Але в Scrum добре реалізований процес збору функцій і їх розподіл по ітераціях. Спірний момент – відсутність жорстко заданого розподілу ролей і обов'язків в команді. Принцип «всі відповідальні за все» працює далеко не завжди, навпаки, чіткий розподіл ролей дозволяє концентруватися співробітникам лише на тих роботах, де вони можуть принести максимум користі. Як і багато інших гнучких методологій, Scrum вимагає наявності представника

замовника або представника ринку. У обох випадках все не так добре, як описується в методології. Замовник не завжди може визначити, що саме йому потрібно, а часта зміна вимог лише уповільнює роботу [7].

Також можна стверджувати, що Scrum - це набір принципів, на яких будується процес розробки, який дозволяє в короткі проміжки часу (спринти), надавати кінцевому користувачеві робоче програмне забезпечення з додатними можливостями, для яких був визначений найбільший пріоритет. Потрбінний функціонал для реалізації в черговому спринті визначається до його початку на етапі планування і не може змінюватися на протязі всього спринта. При цьому жорстко фіксований не великий термін спринта додає процесу розробки гнучкості (рис. 1).

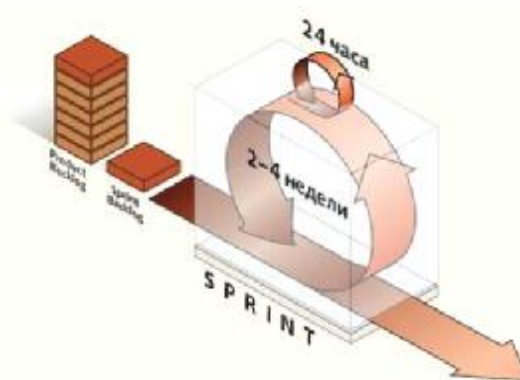


Рис. 1. Схема процесу Scrum

Scrum являється найбільш поширеним наслідувачем гнучкої методології розробки програмного забезпечення.

В даній методології на кожен спринт виділяється наступний склад команди:

1. Скрам Майстер (Scrum Master)
2. Замовник або Власник продукту (Product owner).
3. Команда (Team).

Скрам Майстер (Scrum Master) - одна з найважливіших ролей в методології. Він являється з'єднуючою ланкою між менеджментом та командою. Як правило цю роль отримує менеджер проекту.

Основними обов'язками Скрам Майстра є: створити атмосферу довіри, прибирати перепони, відповідати за дотримання практик і процесу в команді. Також Скрам Майстер проводить зібрання кожного дня (Daily Scrum meeting) і слідкує за прогресом команди використовуючи список задач спринта (Sprint Backlog).

Замовник або власник продукту (Product Owner) – це людина яка відповідає за розробку продукту. Як правило, це менеджер продукту або представник замовника, якщо це аутсорс продукт.

Замовник – це єдина точка прийняття кінцевих рішень для команди в проекті, тому важливо щоб це була людина, а не група чи комітет. Замовник повинен відповідати за формування і бачення продукту, керувати рентабельністю, керувати очікуваннями замовників і всіх зацікавлених осіб, координувати і розставляти за пріоритетами потік задач, взаємодіяти з командою і замовниками. Замовник ставить задачі команді, але він не повинен ставити задачі конкретному члену команди [4].

Команда (Team). В методології Scrum команда є самоорганізованою та самокеруючою.

Команда бере на себе обов'язки по виконанню задач перед Замовником. Основні обов'язки команди:

- відповідати за оцінку задач;
- приймати рішення по імплементації та дизайну;
- розробляти програмне забезпечення;
- відповідати за результат перед замовником;

Розмір команди обмежується розміром групи людей яка може ефективно комунікувати між собою особисто, тому стандартний розмір команд – 7-9 людей.

Також, в команді знаходяться люди з різними навиками, які потрібні для виконання проекту – розробники, тестувальники, аналітики. Не існують заздалегідь визначені і поділені ролі в команді, які б визначали область відповідальності та обов'язків.

Команда являється самоорганізованою одиницею, що дозволяє їй гнучко та швидко реагувати на будь-які можливі задачі. Також для команди важливою є комунікація, тому переважно всіх членів команди розміщують в одному місці, офісі чи кімнат, щоб зменшити або прибрати перешкоди в спілкуванні.

Scrum – це шлях постійного вдосконалення інструментів, тому в кінці кожного спринта учасники збираються на ретроспективу, на якій розповідають про виконану роботу і обговорюють що добре, а що можна покращити. Вони знаходять основні проблеми і обдумують як їх можна вирішити і прибрати в наступному спринті. Це і є постійне вдосконалення.

Так як Scrum відноситься до гнучких методологій, тому досвід його впровадження на практиці надзвичайно різноманітний, це дозволяє виділити основні особливості:

Ідеальні умови для Scrum: гнучкі терміни і бюджет, клієнт який розуміє суть цього підходу і схвалює його, компетентний Product Owner і спрацьована професійна команда.

Скрам чудово підходить коли є розуміння, що продукт може розроблятися поетапно, коли не має чіткого розуміння, що потрібно, але клієнт хоче отримати в кінці те що йому потрібно, а не те що вийшло. Коли необхідний швидкий запуск проекту (спочатку з обмеженим функціоналом), є розуміння які складові і пріоритети в проекті змінюватимуться в процесі розробки.

Скрам передбачає роботу за фреймворком. Це безперервний циклічний та енергійний процес, який перешкоджає застою в команді. Він дозволяє зменшувати невизначеність в вимогах до мінімуму, дає можливість детального прототипування та підвищує внутрішню прозорість процесів.

Зрозуміло, що Scrum – не інструкція по роботі над тим чи іншим конкретним проектом. Це форма, в яку кожна компанія і кожна команда вносить свій зміст, стикаючись з різними підводними каменями методики по ходу її впровадження. Scrum досить простий у вивченні, дозволяє економити час, за рахунок оптимізації активностей. Це дозволяє отримати потенційно робочий продукт в кінці кожного Sprint'a. Scrum робить упор на багатфункціональність команди, яка здатна вирішувати необхідні завдання без бюрократизації та надмірних часових втрат [10]. Це особливо привабливо для малих компаній і стартапів, оскільки позбавляє від необхідності наймати спеціалізований персонал керівників. Але в той же час не потрібно робити із скраму культ і сліпо слідувати всім формальним пунктам та вимогам, це лише різновид гнучких підходів до управління проектами, який передбачає модифікацію під різні вимоги проекту, або навіть взаємодію з іншими підходами, наприклад такими як Kanban.

Kanban. Методологія бере свій початок на виробничих лініях компанії TOYOTA, яка на початку 50-х років XX століття активно почала запроваджувати систему так званого «ощадливого виробництва» за принципами, які показують, що потрібно виробляти, коли і скільки [2].

На той час класична схема мала на увазі виробництво комплектуючих, вузлів і агрегатів, які поставлялися різними підрозділами а іноді й підприємствами не зв'язаними між собою. До прикладу, перед фінальним комплектуванням автомобіля могли пройти тижні або навіть місяці. Ризик полягав у тому, що у разі помилки, допущеною тим чи іншим виробничим підрозділом, збитки від бракованих деталей були колосальні. Рятівною і ключовий тут була ідея створення єдиного потоку, тобто процесу, де немає простою від незавершених або неузгоджених завдань, відсутні паузи, спостерігається поступальний рух вперед. Виходить справжня естафета: завдання, передаються відповідальними сторонами чітко з рук в руки. У цьому випадку помилки попередньої стадії очевидні і виявляють себе миттєво. Це дозволяє уникнути безглузвих витрат, збільшує якість продукту, знижує вартість і скорочує терміни виконання [9].

Останнім часом, Kanban набирає велику популярність у виробництві програмного забезпечення різного напрямку. Деякі команди вважають цю методологію виключно корисною, деякі використовують за принципом «культу Карго». На основі проведеної аналітики можна стверджувати, що Kanban добре себе показує в командах підтримки, а саме:

- групи підтримки програмного забезпечення, де не важливий «план», але важлива швидкість реагування на зміни;
- групи тестування, що працюють окремо від груп розробки;
- служби підтримки;
- інші приклади «неосновних виробництв».

Окремо необхідно зазначити, що Kanban добре працює на стартапах, тобто там, де не має чіткого плану, але активно працюють над розробкою.

Розглянемо приклад використання KANBAN. Припустимо, перед нами звичайна ІТ-компанія, яка при виробництві застосовує ітеративну методологію, а для відстеження оперативної ситуації всередині використовується KANBAN дошка (рис. 2).

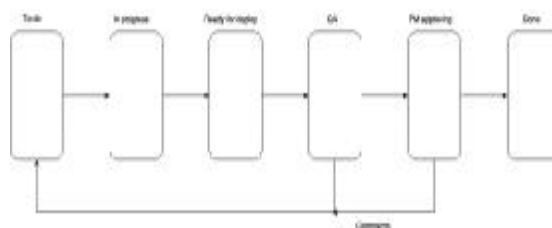


Рис. 2. KANBAN дошка

Колонка «To do» – показує кількість завдань в необхідних для виконання. «In progress» – завдання, які вирішуються в даний момент. «Ready for deploy» – завдання, які вже виконані, але не представлені в тестовому оточенні. «QA» – завдання в процесі тестування. «PM approving» – готові завдання проходять перевірку проєктним менеджером. «Done» – виконані (завершені) завдання поточного спринту.

Процеси в спринті відбуваються наступним чином: взявшись до роботи, програміст перетягує завдання з колонки «To do» в «In progress». В цей момент часу він має тільки одну задачу, концентрується на її ефективному вирішенні, так як паралельне виконання завдань не допускається. Після закінчення своєї частини роботи, він перетягує її в колонку «Ready for deploy» [8].

Починаючи з цього моменту за справу береться відділ тестування. По досягненню ліміту завдань в цій колонці інженери-тестувальники ініціюють складання білда. Також допустимо просто попросити розробника зібрати білд або оновити сервер. При успішних результатах тестування завдання відправляється в колонку «PM approving», де отримує фінальне підтвердження і перетягується в останню колонку «Done». Якщо завдання не проходить тестування, вона знову потрапляє в колонку «To do» з відповідним коментарем.

Для побудови такого процесу необхідно, щоб у всіх членів команди були правильно налаштовані нотифікації. В такому випадку можна уникнути простою при переході завдань між виконавцями. [1,4].

Бувають випадки скупчення завдань у будь-якій з колонок. Це означає, що пропускної здатності на ділянці мало, потрібно знайти причину і вирішити її для запобігання критичного накопичення завдань.

Висновки. В результаті досліджень, отримані результати підтверджують, що переважна більшість практиків, які застосовують гнучкі методології, формують спільну думку, що SCRUM і KANBAN це сукупність методів розробки і управління, гармонічно поєднаних між собою, в основі яких лежать принципи виробничої ітеративності, що дозволяють замовнику частіше отримувати певний робочий функціонал в рамках розробки проєкту. Також необхідно зазначити «простоту» і інформативність такого підходу, високий рівень

комунікації всіх учасників та відсутність обтяжуючої проектної документації [5, 6].

Запровадження гнучких методологій в управлінні ІТ проектами не дає швидкого результату. В першу чергу, вони направлені на зменшення рутинної роботи та проектної документації, маючи на меті збільшення ключових показників ефективності команди за рахунок надійності процесів на всіх етапах проекту, та дотримання встановлених термінів. Члени команди повинні бути готові до змін, і мати чітке розуміння навіщо вони їм. Як показують дослідження, через 5-6 місяців після імплементації гнучкої методології в команду вона починає приносити позитивний результат [11]. Це свідчить про те, що команда відпрацювала для себе комфортні алгоритми роботи, комунікації проходять відповідно до розуміння тих завдань, котрі покладені на команду, а вимоги замовника виконуються без спотворень і згідно обговорених цілей. Поступове впровадження результатів дослідження в деяких компаніях дозволило оптимізувати процеси управління командою ІТ проектів, що відбилося в зменшенні вартості їх реалізації та зниженні рівня збитку від неузгодженості дій всередині проектів, що призвело до їх своєчасного завершення і отримання кінцевих результатів встановленого рівня якості.

Список літератури

1. Хекхаузен Х. Мотивация и деятельность; пер. с нем. С.Пб.: Питер, 2003. 864 с.
2. Laing Samantha and Hryvs Karen. Growing agile: a coach's guide to training Scrum
3. Verheyen Gunther. Scrum – a pocket guide.
4. Книберг Хенрик. Scrum и XP: заметки с передовой.
5. Шегда А. В. Менеджмент: навч. посіб. Київ: Т-во «Знання», КОО, 2002. 583 с.
6. Томпсон А. А. Стратегический менеджмент. Искусство разработки и реализации стратегии: учебник для вузов; пер. с англ., под ред. Л. Г. Зайцева., М. И. Соколовой. Москва: Банки и биржи, ЮНИТИ, 1998. 576 с.
7. Сазерленд Д. Scrum. Революционный метод управления проектами. Москва: Манн, Иванов и Фербер, 2016. 288 с.
8. Кон М. Гибкая разработка ПО. Москва: Вильямс, 2011. С. 576.

9. Трус А. А. Психология. Минск, 2015. 348 с.
10. Демарко Том. Deadline. Роман об управлении проектами. Москва: Вершина, 2006.
11. Чубенко М. О. Основные принципы Agile методологий. *Інформаційні технології та взаємодії (IT & I): III Міжнародна науково-практична конференція.*
12. Чубенко М. О. Особливості управління командами в ІТ проектах. *Українські перспективи у світовому розвитку: щорічна науково-практична конференція молодих учених.*

References

1. Hekhauzen, H. (2003) Motivation and activities. St. Petersburg: Peter, 864 p. [in Russian].
2. Laing, Samantha and Hryvs, Karen. Growing agile: a coach's guide to training Scrum
3. Verheyen, Gunther. Scrum – a pocket guide.
4. Knyberh, Henrik. Scrum and XP: notes from front line [in Russian].
5. Szegda, A. V. (2002) Management. Kyiv: T-vo "Znannya", KOO, 583 p. [in Russian].
6. Thompson A. A. (1998) Strategic management. The art of strategy development and implementation. Moscow: Banki i birzhy, UNITI, 576 p. [in Russian].
7. Sutherland, D. (2016) Scrum. Revolutionary method of project management. Moscow: Mann, Ivanov i Ferber, 288 p. [in Russian].
8. Cohn, M. (2011) Flexible software development. Moscow: Williams, p. 576. [in Russian].
9. Trus, A. A. (2015) Psychology. Minsk, 348 p. [in Russian].
10. DeMarco, Tom (2006) Deadline. A novel about project management. Moscow: Ver-shyna [in Russian].
11. Chubenko, M. O. Basic principles of Agile methodologies. *III International scientific-practical conference "Information technologies and interaction" (IT & I)* [in Ukrainian].
12. Chubenko M.O. "Features of team management in IT projects", *Annual scientific conference of young scientists "Ukrainian prospects in world development"* [in Ukrainian].

M. O. Chubenko¹, *PhD student, project manager,*
e-mail: chubenko.nikolay.mail@gmail.com

V. I. Seminoh², *PhD student, software developer «UAPAY»,*
e-mail: vlad.seminog@gmail.com

O. Ye. Ilarionov³, *PhD, associate professor*
e-mail: oilarionov@gmail.com

¹ Marketing agency «OSD GROUP», Kyiv; tel.: (096) 539-33-66;

²Kyiv; tel.: (063) 596-90-13;

³Kyiv National University named after Taras Shevchenko; tel. (097) 818-83-07

THE USE OF SCRUM AND KANBAN FLEXIBLE METHODOLOGIES IN IT PROJECTS

Considered innovative IT project management methodology SCRUM, KANBAN, highlighted their key features, typical features, delineated area of effective application. In recent years especially noticeable a change vector requirements for the formation of teams in IT projects. Obviously, today, the implementation of complex business projects require a large number of unique knowledge and tools. A project manager, in turn, have to manage a group of different professions, with different levels of qualification and which may present as internal business units and be invited to the project team from the outside. In such circumstances, the question arises organically combine the skills and creative potential production teams to improve their performance efficiency to achieve certain goals. The difficulty lies in the transformation of the future team members in a group of specialists, operating under certain methodologies and are able to solve tasks.

Keywords: *Agile methodology, management of IT projects, software development, iteration, sprint.*

Рецензенти: Данченко О. Б., д.т.н., доцент,
Коломицева О. В., д.е.н., професор.