

РОЗДІЛ 10. МАТЕМАТИЧНІ МЕТОДИ, МОДЕЛІ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ЕКОНОМІЦІ

ЕВОЛЮЦІЯ МЕТОДОЛОГІЇ УПРАВЛІННЯ ІТ-ПРОЕКТАМИ В СУЧАСНИХ ЕКОНОМІЧНИХ УМОВАХ

EVOLUTION OF METHODOLOGY OF MANAGEMENT OF IT PROJECTS IN THE MODERN ECONOMIC CONDITIONS

УДК 005.8:004 (045)

Буткевич В.В.

студентка

Національний технічний університет
України

«Київський політехнічний інститут
імені Ігоря Сікорського»

Польська Т.Д.

доцент кафедри теорії та практики
управління

Національний технічний університет
України

«Київський політехнічний інститут
імені Ігоря Сікорського»

У статті досліджено еволюцію методології управління проектами як інноваційного підходу в управлінні, наведено різні методи управління проектами, проаналізовані можливості адаптації методик до нових економічних умов. Особлива увага приділена в статті гнучким (Agile) та жорстким (Каскадній) типам методології. За результатами аналізу встановлено, що використання гнучкої методології Scrum доцільне в нестабільній економічній ситуації, що актуалізується в сучасних умовах розвитку української економіки.

Ключові слова: інформаційне суспільство, управління проектами, гнучка методологія (Agile), жорстка методологія, Scrum методологія.

В статье исследована эволюция методологии управления проектами как инновационного подхода в управлении, рассмотрены различные методики управления проектами, проанализированы возможности адаптации методик к новым экономическим условиям. Особое внимание уделено в статье гибким (Agile) и жестким (каскадные) типам методологий. По

результатам анализа установлено, что использование гибкой методологии Scrum целесообразно в нестабильной экономической ситуации, которая актуализируется в современных условиях развития украинской экономики.

Ключевые слова: информационное общество, управление проектами, каскадная модель, гибкая методология, гибкая методология, Scrum методология.

This article considers the evolution of project management methodologies as an innovative approach to management, are given the different project management techniques, analyzed the possibility of methods to adapt to the new economic environment. Particular attention is given in the article to flexible (Agile) and hard (cascading) the types of methodologies. The analysis found that the use of agile methods Scrum advisable in an unstable economic situation, which is actualized in the modern conditions of development of the Ukrainian economy.

Key words: informational society, project management, flexible methodologies (Agile), a rigid methodologies, Scrum methodology.

Постановка проблеми. Процеси, що відбуваються в сучасному інформаційному суспільстві, де на передній план виходить виробництво знання та інформації, все з більшою очевидністю демонструють взаємозв'язок і взаємозалежність глобалізації і зростання ролі інформаційно-комунікаційних технологій.

Організації на теренах сучасності можуть існувати та бути конкурентоспроможними лише за умови безперервного розвитку та значних ресурсів адаптації до мінливих умов ведення бізнесу, впровадження ІТ-технологій. Проектно-технологічний тип організаційної культури, впровадження в економіці нових форм управління надає конкурентні переваги і в цьому контексті в останні десятиліття надзвичайно актуалізується проблематика управління проектами.

Саме проектна діяльність вирішує проблеми вчасної адаптації до зовнішніх умов, що швидкоплинно змінюються. З іншого боку, невпинний розвиток інформаційних технологій призводить до збільшення впливу інформаційних технологій на саму проектну діяльність. Є різні види методик управління проектами, використання яких по-різному впливатиме на результат та процес

виконання, та не всі можливо адаптувати до економічних умов в країні, оскільки кожна вищезгадана методика повинна враховувати витрачений час та матеріальні ресурси в досягненні поставлених цілей проекту.

Аналіз останніх досліджень і публікацій. Проблемам управління проектами присвячені роботи багатьох зарубіжних і вітчизняних дослідників. Так, в дослідженні В.Н. Буркова та Д.А. Новикова [1] представлено цілісне уявлення про весь комплекс механізмів, які використовуються на різних етапах проектів, починаючи з визначення цілей проекту і закінчуючи на етапі оперативного реагування. Б. Трейсі [7] в своїй роботі розглядає конкретні проблеми оптимізації, продуктивності та ефективності досягнення результату проекту. С.В. Пятенко [6] вважає, що ключовим інгредієнтом успіху проекту є управління проблемами. Проблемам управління будівельними проектами присвячені роботи В. В'язова [5]. Д. Сазерленд [3] в своїй книзі розкриває філософію Scrum-революційного методу управління проектами, вважаючи, що успіх проекту залежить від методології його управління. В.М. Кожухар [2] звертає увагу на фактор ринкової економіки, зазначаючи,

що управління ризиками є частиною діяльності управління проектами [2]. І.Г. Чиркова та К.Ч. Акбаров [4] вважають, що проект потребує техніко-економічної оцінки. А.А. Трусь [8] акцентує на партнерській взаємодії та мотивації співробітників як невід'ємних факторів продуктивної роботи.

Виклад основного матеріалу дослідження.

Управління проектами в сфері інформаційних технологій повинні відповідати основним критеріям, виконання яких забезпечує якість проекту. Оскільки розробка програмного забезпечення, як і будь-яка інша технічна дисципліна, має справу з такими основними проблемами як якість, вартість і надійність, то в виборі методики слід орієнтуватися як на кінцеву мету, продукт проекту, так і на фінансові та часові показники.

Розглядаючи різні методологічні підходи до управління ІТ проектами в різних галузях економіки, проаналізуємо каскадну модель, яка була року в використанні до 2005 року.

Каскадна модель відноситься до моделей класичного життєвого циклу.

Етапи розробки класичної каскадної моделі виглядають наступним чином:

- Аналіз вимог проекту;
- Проектування продукту;
- Реалізація ПО;
- Тестування продукції;
- Інтеграція системи;
- Підтримка ПО.

Перехід на кожний наступний етап в даній моделі можливий лише після успішного завершення попереднього етапу. Така жорстка послідовність дозволяє формалізувати процес розробки, що робить його надзвичайно прозорим, при цьому реальна протяжність етапів часто не відповідає проміжкам часу, які визначені на графіках та в документації. При необхідності внесення правок в документацію розробка продукту зупиняється аж до моменту повторного погодження документів. Отже, при недостатньому рівні опрацювання вимог існує ризик збільшити терміни розробки до абсолютно неприйнятних величин, змінюючи об'єми витрат.

Каскадна модель передбачає строго послідовне (у часі) і одноразове виконання всіх фаз проекту з жорстким (детальним) попереднім плануванням в контексті визначених одноразово і цілком певних вимог до програмної системи. Така складна система погребує великої кількості часу, яке буде відведене на складання графіків та документації перед початком розробки проекту.

Етапи проекту відповідно до каскадної моделі: Формування вимог; проектування; реалізація; тестування; впровадження; експлуатація та супровід.

Водоспадна схема включає кілька важливих операцій, які можна застосувати до всіх проектів: складання плану дій з розробки системи; плану-

вання робіт, пов'язаних з кожною дією; застосування операції відстеження ходу виконання дій з контрольними етапами.

У зв'язку з тим, що згадані завдання є невід'ємним елементом всіх добре керованих процесів, практично не існує причин, що перешкоджають утвердженню повнофункціональних, класичних методів керівництва проектом, таких як аналіз критичного шляху і проміжні контрольні етапи.

Всі визначення документуються в специфікації аналізу. Тут же завершується рішення задачі планування проекту.

Проектування програмного забезпечення полягає в створенні уявлень:

архітектури програмного забезпечення; модульної структури програмного забезпечення; алгоритмічної структури програмного забезпечення; структури даних; вхідного і вихідного інтерфейсу (вхідних і вихідних форм даних).

- Переваги класичного життєвого циклу:
 - дає план і часовий графік по всіх етапах проекту;
 - впорядковує хід конструювання.
- Недоліки класичного життєвого циклу:
 - реальні проекти часто вимагають відхилення від стандартної послідовності кроків;
 - цикл заснований на точному формулюванні вихідних вимог до програмного забезпечення (реально на початку проекту вимоги замовника визначені лише частково);
 - результати проекту доступні замовнику тільки в кінці роботи.

Робота за цією методикою повинна бути покроковою, поступовою. Попередньо складався етапний план, представлений на діаграмах Г. Ганта, що давало ілюзію контролю над часом і всіма процесами (рис. 1). При цьому впродовж проекту могли з'являтися обставини, які вимагали порушувати графік й виходити за рамки бюджету. Поширення методики сталося з 1980-х років з моменту появи персональних комп'ютерів, що надавало більше можливостей для детального представлення будь-якого процесу.

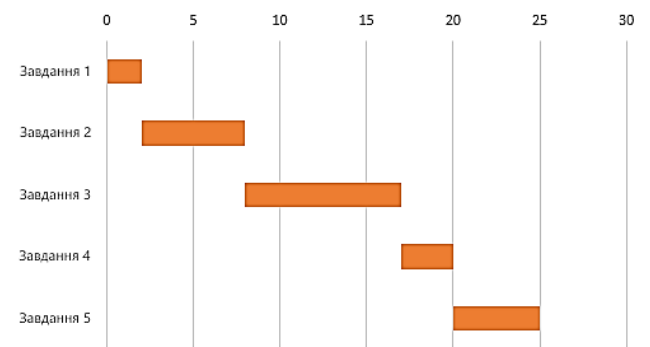


Рис. 1. Графік Ганта

Названо діаграми в честь самого автора Генрі Ганта, який створив їх в 1910-ому році. Першим

звернувся до графіків генерал Вільям Крузер, який на той час був начальником артилерійсько-технічної служби збройних сил США в Першу світову війну [2, с. 17].

Критикуючи подібні графіки і в цілому каскадну методологію, автор методології Скрам писав: «Діаграми Ганта справляють враження. З ними є тільки одна проблема: вони завжди і без виключення неправильні» [2, с. 17]. Як приклад доцільності тієї чи іншої методики, Джефф Сазерленд в своїй роботі «Scrum. Революційний метод управління проектами» говорить про важливий проект Федеральної служби безпеки під назвою «Страж», багатообіцяючий великомасштабний план з модернізації управління інформацією, який допоміг би в майбутньому запобігти подіям, подібним до 11 вересня.

Минав 2010-ий рік, а більшість документації зберігалась в паперовому вигляді, програма, по якій працювала Федеральна служба безпеки, не змінювалась з далеких вісімдесятих. Більшість працівників не бажали нею користуватися, так як швидкість небезпек, з якими потрібно було впоратися, випереджувала швидкість роботи з програмою, тобто програма не відповідала роботі у 21-ому сторіччі.

Ось як описує цю процедуру Джонсон, який працював над проектом «Страж»: «Ви склали в текстовому редакторі докладну записку і роздрукували її в трьох примірниках. Одну копію послали на затвердження, і вона проходила весь ланцюжок до самого верху. Другу відправляли до місцевого архіву на випадок, якщо перша загубиться. Ну, а з третьою ви сідали, брали червону ручку – так-так, я не жартую, червону ручку – і обводили ключові слова для занесення в базу даних. Ви індексували власний звіт «. Таким чином, «на Бюро покладали провину, що його підрозділи не зуміли зв'язати всі відомості воедино і виявити численних активістів» Аль-Каїди «, що в'їхали в країну по місяці і навіть лічені тижні до 11 вересня» [2, с. 13].

«Ми у своєму розпорядженні мали інформацію, яка могла б запобігти подіям 11-го вересня. А вони там сиділи, і ніхто не вжив жодних заходів... Я до сих пір не бачу, що вони усувають проблему... Поки ми дійдемо до технології XXI століття, вже настане ХХІІ століття» [4, с. 5]. Таким було зізнання Патріка Лехі, сенатора-демократа від штату Вермонт, яке він оприлюднив на сторінках Washington Post.

Аналізуючи процес управління невдалим проектом «Страж» і іншими, можна зробити висновки, що проблема полягає в способі роботи та методі, яким користуються для управління цими проектами. Зазвичай це робили таким чином: вивчали потреби замовника; за потребами йшло їх рішення; тривале планування, що саме потрібно зробити; тривале обмірковування, як це потрібно здійснити, робота над графіками, «де були позначені і всі

подробниці, які потрібно виконати, і час, який буде потрібно на кожну задачу. Потім за рахунок точного підбору кольорів, вони показували, як кожна фаза проекту послідовно переходить в наступну – все це нагадувало водний каскад» [2, с. 16]. Робота по каскадній моделі потребує також додаткового робочого місця, в обов'язки працівника входить своєчасне поповнення і оновлення графіків.

Джефф Сазерленд в книзі «Scrum» порівнює цей метод з організацією роботи в ЦК КПРС, коли влада Радянського Союзу отримувала звіти перед розпадом СРСР та істинно вірила написаним діаграмам. «Сьогодні, як і в ті роки, звіти продовжують бути важливіше дійсності – але ж вони, судячи з усього, покликані її описувати, – але, якщо раптом сплинуть невідповідності, то винним називають реальність, а не діаграму [2, с. 18].

Маючи досвід того, що саме Scrum зміг завершити систему «Страж», можна робити висновки, що це єдина правильна методика, яка спроможна реалізувати такі проекти.» У ФБР стверджують, що для завершення проекту «Страж» вони вдавалися до «гнучкої методології розробки» [2, с. 25], що означає використання методології Scrum.

Перша проблема, з якою зіштовхнулася команда по роботі завершення проекту «Страж», була документація, розбір якої зайняв велику кількість ресурсів. «Мені доводилося бачити паперові стовпи висотою більше метра. З проекту в проект я спостерігаю одне і те ж: як копіюються стандартні формулювання і вставляються в нескінченні документи, але ніхто толком їх не читає», – згадує Джонсон. Але при всіх цих обставинах, «була створена система, що змушує людей схвалювати порожні ілюзії» [2, с. 24].

Модель WaterFall відноситься до жорсткого типу моделей і є найбільш поширеною моделлю серед моделей життєвого циклу розробки програмного забезпечення. Вона дуже проста для розуміння та використання. В цій моделі кожна фаза повинна бути завершена до наступного етапу. В кінці кожного етапу проводиться огляд, який допомагає визначити, чи перебуває проект на правильному шляху та чи буде цей проект продовжено [8].

Модель WaterFall може бути реалізована для будь-якого розміру проекту. Кожен етап повинен здійснюватися окремо в потрібний час, поступово переходячи від попереднього до наступного етапу, повністю закінчивши попередній. Документація, яка заповнюється на кожному етапі моделі WaterFall дозволяє людям зрозуміти, що було зроблено.

Методика WaterFall складається з послідовних, неперетинних фаз, де одна фаза не може початися, поки попередня фаза ще не завершена. В кінці кожного етапу є контрольна фаза, де приймається рішення стосовно того, чи дозволити проекту рухатися вперед чи ні (етап Gate). Основні зміни дозволені тільки якщо ССВ (Board

Control Change) затверджує їх. Продукт завершено тільки в кінці останньої фази. Після того як проект буде зроблено, продукт / послуга входить у фазу обслуговування.

Схема даної моделі ділиться на етапи чи фази. Перший етап, який вміщує в собі збір вимог та аналіз. Це перша фаза моделі WaterFall, яка включає в себе зустріч з клієнтом, щоб зрозуміти його вимоги. Це найбільш важливий етап, будь-яка неправильна інтерпретація вимог на даному етапі може привести до виникнення проблем з перевіркою та виконанням пізніше. Дуже важливо зрозуміти вимоги та очікування замовника, щоб кінцевий продукт відповідав його вимогам. Основні вимоги до системи повинні бути зрозумілі інженеру-програмісту, або, як його ще називають, аналітику. Всі вимоги якісно документуються і обговорюються детально з замовником.

Друга фаза є детальним проектуванням (дизайн). Вимоги замовника розбиті на логічні модулі для простоти реалізації. Вимоги до апаратного та програмного забезпечення для кожного модуля ідентифікуються і розробляються відповідним чином. Крім того, на даному етапі встановлюється зв'язок між різними логічними модулями, алгоритми і схеми, визначаються масштаби і цілі кожної логічної моделі. Коротше кажучи, ця фаза є фундаментом для фактичного програмування і здійснення роботи. Це проміжний етап між аналізом вимог і кодуванням. Всі процеси та конструкція повинні бути задокументовані для подальшого використання.

Третя фаза кодування являє собою стадію, в якій дизайн перетворюється в форми. Якщо конструкція робиться досить докладно, то кодування може бути зроблено ефективно. На цьому етапі розробляються програми. Все програмне забезпечення ділиться на невеликі модулі, проводиться кодування для цих невеликих модулів.

На четвертому цьому етапі проводиться методичне тестування як окремих компонентів, так і цілого, для переконання, що вони вільні від помилок і в повній мірі відповідають вимогам, викладеним в першому кроці (збір вимог та аналіз).

Технічне обслуговування є останньою фазою моделі WaterFall, в якій завершений програмний продукт вже переданий клієнту після того, як пройшов альфа та бета-тестування. Якщо клієнт пропонує внести зміни або удосконалити програмне забезпечення, потрібно повертатись до першої фази, до аналізу вимог.

Недоліки WaterFall моделі.

1. Аналіз вимог на першому етапі може не відповідати очікуванням, заявленим замовником. Це означає, що команда повинна проходити знову всі етапи з самого початку і до кінця, на що буде витрачений додатковий час та кошти.

2. Клієнт може побачити діючу модель проекту тільки в кінці. Як нам здається, це є особли-

вим мінусом, бо це єдиний момент, коли він може вносити правки, отже помилки, які були допущені раніше, зводять всю розробку проекту нанівець.

3. Ми не можемо повернутися на попередній етап, ця модель цього не дозволяє.

4. Важко простежити послідовність в процесі розробки програмного забезпечення [11].

Головною причиною популярності каскадної моделі програмування слід назвати вищезгадану прозорість процесу розробки – завдяки послідовному переходу від етапу до етапу і високому рівню формалізації процесу, управління масштабними проектами здійснюється набагато простіше, а команда, в свою чергу, працює більш злагоджено.

Крім того, жорстка послідовність дозволяє дати точну оцінку вартості розробки проекту і його термінів, що дозволяє точно спрогнозувати ефект, отриманий від запуску додатка.

Але, зіштовхуючись з реальними обставинами, дуже часто трапляються затримки з запусками проектів та відхилення від вказаного бюджету. Виникає безліч питань про цінність каскадної методики для сучасного суспільства і нових технологічних особливостей сучасного світу, оскільки зусилля, затрачені на створення графіків і їхньої адаптації до процесу, поступаються мінливим обставинам сучасного постіндустріального світу.

Джефф Сазерленд говорить про методики так: «Існує два підходи до роботи: старий «каскадний»- при ньому викидаються на вітер сотні мільйонів доларів, і часто він так ні до чого і не призводить; новий – коли зобов'язання виконуються меншими силами, в короткі терміни і з низькими витратами, а підсумковий продукт відрізняється відмінною якістю і забезпечує високу продуктивність» [2, с. 20].

Проаналізуємо підхід, який підтримує різні методики гнучкої розробки проектів. Agile-методологія має свої принципи, описані в Agile – маніфесті, які дозволяють організувати дисциплінований гнучкий процес управління IT-проектом, проводячи всю роботу ітераціями з проміжними перевітками, створюючи злагоджену самоорганізовану команду розробників і здійснюючи постійну комунікацію з замовником. Найбільш поширеною є методологія розробки Scrum, яку можна вважати набором конкретних практик, що використовуються в процесі розробки програмного забезпечення.

Agile підхід до управління проектами був створений в 1995 році в рамках спільних зусиль між APMG-міжнародних систем, метод розвитку (DSDM) Консорціум [7]. Agile підхід базується на принципі управління взаємодіями людей і заснований на процесі людського співробітництва. Підхід використовується у програмному забезпеченні, веб-сайт-технології, а також у творчій сфері та індустрії маркетингу. У Agile підході проект сприймається як ряд відносно невеликих заходів, заду-

маних і використаних для управління відповідно до адаптивних методів, на відміну від того, що має попередньо спланований процес [5].

Підхід Agile управління проектами має три відмінні риси:

- Часте тестування проекту на стадії розробки;
- Це єдиний підхід, який активно залучає клієнта в процесі управління проектами;
- Як правило, клієнт повинен брати участь в етапах розробки процесу та залучатися до взаємодії.

Scrum-підхід до розробки програмного забезпечення для технологічних галузей. Ця методологія була розроблена Кеном Швабер і Джеффом Сазерлендом в 1993 році. Ця методологія була більш швидкою, надійною і ефективною, як стверджували її автори, в порівнянні з класичними методиками управління проектами, такими як Водоспад, V-модель та іншими.

Згідно з визначенням, Scrum – це каркас розробки, з використанням якого люди можуть вирішувати проблеми, які з'являються, продуктивно і виробляючи продукти найвищої значущості [2].

Командна гра (сутичка), саме так перекладається слово Scrum. «Цей метод дозволяє учасникам групи ефективно взаємодіяти як з замовником, так і один з одним під час всього процесу розробки» [2, с. 5]. Як бачимо, в методології Scrum велике значення як фактору успішності надається командній роботі при повній взаємодії учасників команди і чіткому розумінні її мети. Адепти Scrum сповідують творчий підхід до роботи, тим самим дозволяючи команді добиватися високих результатів, не втрачаючи віри в свої сили.

Ідея методології Scrum: «Коли б не був запущений проект, вам ніщо не заважає регулярно пере-

віряти хід робіт і послідовно з'ясовувати: чи справляєтеся ви з завданням; чи в потрібному напрямку рухаєтеся; чи створюєте саме те, що насправді хоче отримати замовник» [2, с. 22].

Результатом успішного аналізу потреб управління проектами став «Маніфест гнучкої методології розробки програмного забезпечення», розроблений Д. Сазерлендом, який проголошував такі цінності: «люди важливіші за процеси; фактична робота продукту важливіше ніж документація, яка фіксує, що і як продукт повинен робити; співпраця з замовником важливіше обговорення умов договору з ним; реакція на зміни важливіша, ніж дотримання попереднього плану. Scrum – це концепція, створена ..., щоб втілити ці цінності в життя. Не існує ніякого єдиного підходу під назвою «гнучка методологія» [2, с. 17].

Етапи розробки програмного забезпечення за методологією Скрам детально зображено на малюнку, зображеному нижче.

Дана модель управління проектами в середовищі ІТ найпоширеніша, оскільки вона дозволяє адаптувати вимоги власника продукту, який має можливість коригувати свої вимоги відповідно до змін в навколишньому середовищі.

При використанні «гнучких» методологій завдання проекту розбиваються на малі частини (ітерації) з ретельним короткостроковим плануванням і майже незначним довгостроковим плануванням.

У методології Scrum команда є сомоорганізованою і самокерованою. Команда в Scrum кроссфункціональна. До неї входять люди з різними навичками – розробники, аналітики, тестувальники. Завдяки постійному аналізу виконаної роботи і можливостям здійснювати коригування напряму проекту між ітераціями, (спринт) мето-

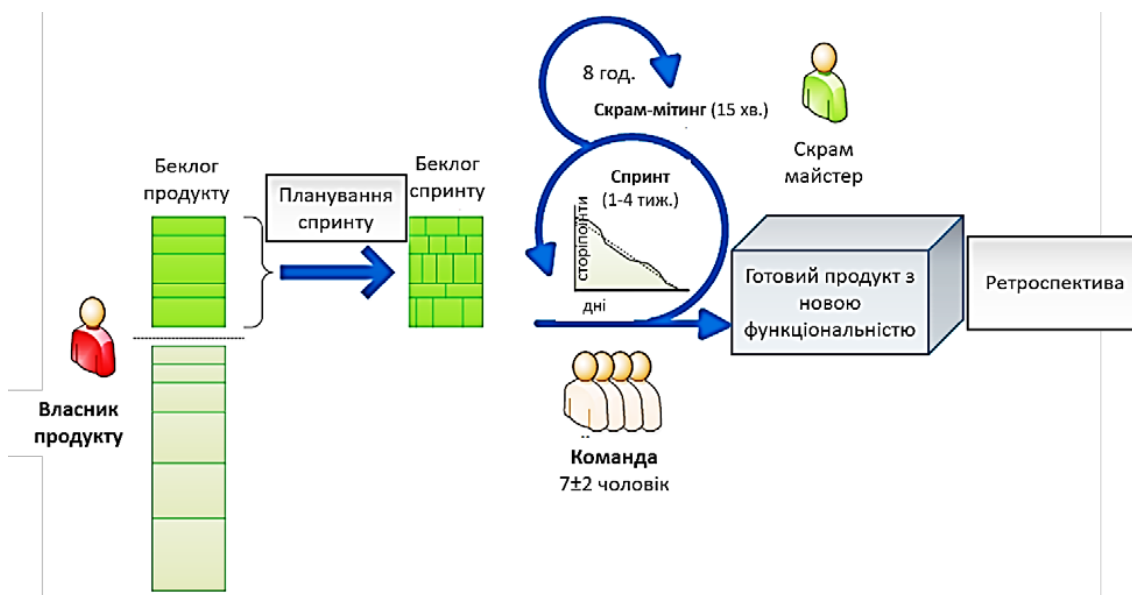


Рис. 2. Схема Скрам підходу

дологія «скрам» дозволяє більш якісно розробити програмне забезпечення і досягти продуктивних результатів.

На сьогоднішній день каскадна модель управління проектами вже практично не використовується в своєму початковому вигляді, що обумовлено малою гнучкістю моделі і її старінням відносно новітніх технологій та інформаційного розвитку суспільства. Одна з перших моделей програмування, яка колись відповідала всім вимогам часу, сьогодні використовується лише в поєднанні з більш сучасними методами, утворюючи гібридні моделі.

Висновки з проведеного дослідження. Вибір методики управління проектами є життєво важливим для реалізації успішного проекту. Вибір конкретного підходу до управління проектами обумовлений цілою низкою чинників, в тому числі в період реалізації проекту, чинниками вартості, складності проекту і т.д., різні методології мають різні функції, які підходять для конкретних вимог проекту.

Каскадне управління є більш актуальною методикою управління проектами при стабільній економіці, оскільки має жорсткий контроль за якістю та реєстрацію всіх процесів на документальному рівні. Це дозволяє контролювати графік виконання та аналізувати роботу над проектом більш пильно, використовуючи додаткові фінанси. Дана документація буде зберігатися до закінчення роботи над проектом та до його кінцевого випуску. При невиконанні якогось етапу розробки будуть потребуватися додаткові кошти для повернення на минулі етапи та їхнє корегування.

Скрам – гнучка методологія, яка дозволяє прораховувати ризики за рахунок поетапного виконання частин проекту. Повернення до минулих етапів, використовуючи додаткові кошти, не буде потрібним, оскільки команда скрам виконує проект частинами, в заданий строк, що дозволяє корегувати минулі помилки без великих фінансових затрат.

Скрам команда багатofункціональна та самостійна, кожен член команди не має чітких ролей і відповідальність за проект лежить на ній в цілому, що не передбачає жорсткого контролю за окремими членами команди. Одним з основних принципів Scrum є самоорганізація, багатofункціональність команди. Разом з тим необхідно відмітити, що згідно з дослідженнями соціологів, чисельність здатних на самоорганізацію, особисто мотивованих співробітників не перевищує 15% від працездатного населення [3].

Методологія Скрам може забезпечити зміни в бажаннях клієнта на всіх етапах розробки

проекту. Таке ставлення до управління тягне за собою зміни в оплаті заробітної плати, але гарантує те, що на виході проект буде виконаний в тому варіанті, який буде дійсно потрібний замовнику. Ідеологія Scrum стверджує, що заздалегідь неможливо передбачити всі зміни, таким чином немає сенсу заздалегідь планувати весь проект, обмежившись лише частковим плануванням, що є актуальним в нестабільній економічній ситуації.

БІБЛІОГРАФІЧНИЙ СПИСОК:

1. Бурков В. Н., Новиков Д. А. Как управлять проектами / В. Н. Бурков, Д. А. Новиков. – М: Синтег, 1997. – 190 с.
2. Кожухар В. М. Инновационный менеджмент / В. М. Кожухар. – М: Издательство Ассоциации строительных вузов, 2012. – 292 с.
3. Сазерленд Д. Scrum. Революционный метод управления проектами / Джефф Сазерленд – МИФ. Бизнес, 2016. – 288 с.
4. Чиркова И. Г, Акберов К. Ч. Внутрифирменное планирование проектной деятельности: учебное пособие / И. Г. Чиркова, К. Ч. Акберов. – Н: НГТУ, 2015. – 64 с.
5. Вязовой В. Управление проектами в строительстве / В. Вязовой // [Электронный ресурс] – Режим доступа: <http://www.e-executive.ru/management/practices/338248-upravlenie-proektami-v-stroitelstve>
6. Пятенко С. В. Методы анализа наиболее типичных проблем управления проектом / С. В. Пятенко // [Электронный ресурс] – Режим доступа: https://iteam.ru/publications/project/section_35/article_2808
7. Трейси Б. Управляй своим временем и удвой результаты / Б. Трейси // [Электронный ресурс] – Режим доступа: <http://www.rulit.me/books/upravlyaj-svoim-vremenem-i-udvoj-rezultaty-read-443214-2.html>
8. Трус А. А. Психология управления / А. А. Трус // Вышэйшая школа. – 2014 [Электронный ресурс] – 2014. – Режим доступа: <http://mreadz.com/read-296679>
9. Bhargav R. Waterfall model / Bhargav // [Электронный ресурс] – Режим доступа: http://www.slideshare.net/BHARGAV_VISANI/waterfall-model
10. Pichler R. Agile product management with Scrum: creating products that customers love / Pichler // [Электронный ресурс] – Режим доступа: <http://www.romanpichler.com/romans-books/agile-product-management-with-scrum/>
11. Satalkar B. Waterfall Model vs. V Mode / Satalkar // [Электронный ресурс] – Режим доступа: <http://www.buzzle.com/articles/waterfall-model-vs-v-model.html>
12. Schwalbe K. Information Technology Project Management. / K. Schwalbe // [Электронный ресурс] – Режим доступа: <http://www.twirpx.com/file/1283991/>