

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Інформаційних управляючих систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

«Дослідження методологій і моделей оцінювання ефективності використання SCRUM в управлінні IT-проектами»
(тема)

Виконав:
студент 2 курсу, групи УПГІТм-21-1
Дмитро ЛУТФУЛІН
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)


Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Управління проектами в галузі інформаційних технологій
(повна назва освітньої програми)

Керівник проф. Віктор ЛЕВИКІН
(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри


(підпис)

Костянтин ПЕТРОВ
(власне ім'я, прізвище)

2023 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
Кафедра Інформаційних управляючих систем
Рівень вищої освіти другий (магістерський)
Спеціальність 122 Комп'ютерні науки
(код і повна назва)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Управління проектами в галузі інформаційних технологій
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)
« 03 » квітня 20 23 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Лутфуліну Дмитру Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методологій і моделей оцінювання ефективності використання SCRUM в управлінні ІТ-проектами

затверджена наказом університету від 03 квітня 2023 р. № 319Ст

2. Термін подання студентом роботи до екзаменаційної комісії 19 05 2023 р.

3. Вихідні дані до роботи Науково-технічні публікації, джерела інтернету, що стосуються теми кваліфікаційної роботи.


4. Перелік питань, що потрібно опрацювати в роботі: Вступ; Аналіз існуючих моделей і методологій; Дослідження моделей і методологій; Розробка вдосконаленої моделі ІТ-проекту; Практичне використання; Висновки


:

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз матеріалів з теми роботи	04.04.23 – 10.04.23	виконано
2	Постановка задачі атестаційної роботи	09.04.23 – 12.04.23	виконано
3	Обробка матеріалу	12.04.23 – 15.04.23	виконано
4	Попереднє дослідження галузі завдання	15.04.23 – 18.04.23	виконано
5	Аналіз існуючих методів і моделей задоволення запитів клієнтів	18.04.23 – 25.04.23	виконано
6	Дослідження розробленої моделі задоволення запитів клієнтів	25.04.23 – 28.04.23	виконано
7	Практичне використання вдосконаленої моделі задоволення запитів клієнтів	28.04.23 – 1.05.23	виконано
8	Написання пояснювальної записки	1.05.23 – 6.05.23	виконано
9	Підготовка презентації	6.05.23 – 9.05.23	виконано
10	Перевірка на плагіат	15.05.23	виконано
11	Нормконтроль	9.04.23 – 15.04.23	виконано
12	Захист кваліфікаційної роботи в ЕК	19.05.23	виконано

Дата видачі завдання 03.04.2023 р.

Студент _____  _____
(підпис)

Керівник роботи _____  _____ професор каф. ІУС Віктор ЛЕВИКІН
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до магістерської кваліфікаційної роботи містить: 67 с., 12 рис., 8 табл., 22 джерел, 1 додаток.

АНАЛІЗ ПРАКТИЧНОГО ВИКОРИСТАННЯ SCRUM, МЕТОДОЛОГІЯ SCRUM, ОСНОВНІ МОДЕЛІ ОЦІНЮВАННЯ.

Об'єктом дослідження є методології і моделі оцінювання ефективності використання SCRUM у керуванні IT-проектами.

Аналіз IT-проекту, де використовується SCRUM, з метою визначення ефективності його використання та оцінки результатів застосування даної методології.

Оцінка ефективності використання SCRUM в управлінні IT-проектами шляхом аналізу методологій та моделей оцінювання.

Виявлення чинників, що впливають на успішність впровадження SCRUM в IT-проектах.

Висновки про переваги та обмеження SCRUM в контексті управління IT-проектами, що можуть бути використані для подальших досліджень або впровадження в практику.

Покращення ефективності впровадження SCRUM шляхом ідентифікації факторів успіху та недоліків методології. Посилення розуміння переваг та обмежень SCRUM у контексті IT-проектів і сприяння вдосконаленню практик управління проектами.

ABSTRACT

The Explanatory note to the master's qualification work contains: 67 pages, 12 figures, 8 tables, 22 sources, 1 appendix.

ANALYSIS OF THE PRACTICAL USE OF SCRUM, BASIC EVALUATION MODELS SCRUM METHODOLOGY.

The object of the study is methodologies and models for evaluating the effectiveness of using SCRUM in managing IT projects. Analysis of an IT project, where SCRUM is used, in order to determine the effectiveness of its use and evaluate the results of the application of this methodology. Evaluating the effectiveness of using SCRUM in IT project management by analyzing evaluation methodologies and models. Identification of factors affecting the success of SCRUM implementation in IT projects.

Conclusions about the advantages and limitations of SCRUM in the context of IT project management, which can be used for further research or implementation in practice. Improving the efficiency of SCRUM implementation by identifying the success factors and shortcomings of the methodology. Increasing understanding of the benefits and limitations of SCRUM in the context of IT projects and contributing to the improvement of project management practices.

ЗМІСТ

Скорочення та умовні позначки.....	5
Вступ.....	8
1 Аналіз предметної області та постановка задачі	10
1.1 Аналіз методології SCRUM.....	10
1.2 Аналіз і порівняння SCRUM з іншими методологіями управління проектами.....	12
1.3 Аналіз ефективності методології SCRUM в управлінні ІТ-проектами.....	24
1.4 Постановка задачі.....	26
2 Методології та моделі оцінювання ефективності SCRUM.....	27
2.1 Опис формули “Швидкість доставки продукту до замовників” ...	27
2.2 Дослідження методології та методів Agile.....	30
2.3 Дослідження методології та методів Waterfall.....	31
2.4 Порівняльний аналіз методологій управління проектами.....	38
3 Проектування інформаційної системи за допомогою SCRUM.....	43
3.1 Планування проекту.....	43
3.2 Результати вирішення задачі оптимізації змісту проекту.....	48
Висновки.....	53
Перелік джерел посилання.....	54
Додаток А Графічний матеріал.....	57

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

IT – інформаційні технології;

IS – інформаційна система;

IAC – інформаційно-аналітична система;

ХНУРЕ – Харківський національний університет радіоелектроніки;

PO – product manager

SM – scrum master

ВСТУП

В сучасному світі, де ІТ-галузь постійно зростає і розвивається, ефективне управління проектами стає все більш важливим для компаній, що працюють у цій сфері. SCRUM - це методологія управління проектами, яка стала досить популярною серед розробників програмного забезпечення та ІТ-фахівців.

Однак, важливо досліджувати ефективність використання SCRUM в управлінні ІТ-проектами, оскільки існують різні моделі та методи оцінювання, які можуть бути використані для оцінки успішності проекту. Дослідження цих моделей та методів може допомогти компаніям у зробленні кращих рішень у відношенні вибору методології управління проектами та забезпеченні успіху їхніх проектів.

Ця кваліфікаційна робота зосереджена на дослідженні моделей та методів оцінювання ефективності використання SCRUM в управлінні ІТ-проектами. Дослідження включає в себе огляд літератури, дослідження даних та порівняння різних моделей та методів оцінювання.

Метою цього дослідження є виявлення та оцінка ефективності різних моделей та методів оцінювання використання SCRUM в управлінні ІТ-проектами. Дослідження може бути корисним для розробників програмного забезпечення, менеджерів проектів та усіх, хто цікавиться ефективним управлінням ІТ-проектами.

Отримані результати можуть допомогти компаніям у зробленні кращих рішень при виборі методології управління проектами та забезпеченні успіху їхніх проектів. Дослідження також може допомогти у покращенні процесів управління проектами, забезпеченні зменшення ризиків та збільшення ефективності роботи команди проекту.

Узагалі, дане дослідження є важливим з точки зору практичної застосовності, оскільки надає певні рекомендації щодо вибору та

використання різних моделей та методів оцінювання SCRUM в управлінні IT-проектами.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз методології SCRUM

Зважаючи на широке застосування SCRUM у сфері IT-проектів, аналіз методології є важливим етапом дослідження її ефективності у управлінні проектами. У даному розділі роботи будуть детально проаналізовані ключові аспекти методології SCRUM, такі як її основні принципи, ролі та відповідності, етапи та процеси, засоби та техніки.

Основні принципи SCRUM. У цьому пункті буде розглянуто основні принципи, на яких базується SCRUM, зокрема, ітераційність, інкрементальність, самоорганізація та інші. Також будуть проаналізовані переваги та недоліки застосування цих принципів.

Ролі та відповідності. У цьому пункті буде розглянуто ролі, які відіграють учасники процесу SCRUM, такі як Product Owner, Scrum Master та Development Team, а також їхні відповідності. Також будуть проаналізовані переваги та недоліки такого розподілу ролей.

Етапи та процеси. Є етапи та процеси, які виконуються в рамках SCRUM, зокрема, Sprint Planning, Daily Scrum, Sprint Review та Sprint Retrospective. Також будуть проаналізовані переваги та недоліки кожного з цих етапів та процесів.

Засоби та техніки. У цьому пункті будуть розглянуті засоби та техніки, які використовуються у рамках SCRUM, зокрема, Product Backlog, Sprint Backlog, Burn-down Chart та інші. Також будуть проаналізовані переваги та недоліки кожного з цих засобів та технік.

Відмінності від інших методологій управління проектами. У цьому пункті будуть проаналізовані відмінності SCRUM від інших методологій управління проектами, таких як Waterfall, Agile та інші. Будуть проаналізовані переваги та недоліки застосування SCRUM порівняно з іншими методологіями.

Проблеми застосування SCRUM. Є проблеми, з якими можуть стикатись команди при застосуванні SCRUM, такі як відсутність досвіду у використанні методології, неефективна комунікація та інші. Будуть проаналізовані шляхи розв'язання цих проблем.

Вимоги до команди для успішного застосування SCRUM. У цьому пункті будуть розглянуті вимоги до команди для успішного застосування SCRUM, такі як знання методології, відповідальність та самоорганізація, комунікаційні навички та інші. Будуть проаналізовані переваги та недоліки вимог до команди.

Застосування SCRUM у різних типах проектів. У цьому пункті будуть проаналізовані різні типи проектів, для яких може бути застосована методологія SCRUM, зокрема, проекти зі складними вимогами, проекти з динамічною зміною вимог та інші. Будуть проаналізовані переваги та недоліки застосування SCRUM у різних типах проектів.

Інструменти підтримки SCRUM. У цьому пункті будуть проаналізовані інструменти підтримки SCRUM, такі як Jira, Trello та інші. Будуть проаналізовані переваги та недоліки використання цих інструментів для успішного застосування SCRUM.

Аналіз методології SCRUM є важливим етапом дослідження ефективності використання SCRUM в управлінні IT-проектами. Результати цього аналізу можуть бути використані для подальшого дослідження ефективності SCRUM, а також для вдосконалення його застосування у практиці управління IT-проектами.

Важливою складовою успішного застосування SCRUM є розуміння командою її основних принципів та вимог до команди. Вимоги до команди включають знання методології, відповідальність та самоорганізацію, комунікаційні навички та інші. Також важливо мати належний інструментарій для успішного застосування SCRUM.

У висновках до аналізу методології SCRUM будуть зроблені висновки щодо ефективності застосування SCRUM у управлінні IT-проектами. Будуть

проаналізовані ключові висновки з кожного з пунктів та їх взаємозв'язок, а також надані рекомендації щодо подальшого використання методології SCRUM у управлінні IT-проектами.

Аналіз методології SCRUM є важливим етапом дослідження ефективності використання SCRUM в управлінні IT-проектами.

Для досягнення успішного застосування SCRUM необхідно знати його основні принципи та елементи. Основні принципи SCRUM включають ітеративну та інкрементальну розробку, самоорганізацію команди, залучення клієнта та інші. Елементами SCRUM є такі поняття, як продуктовий беклог, спринт, скрам мітинги та інші.

У пункті "Ітеративна та інкрементальна розробка беклога" буде розглянуто, що ці принципи означають та як вони допомагають управляти проектом. Важливим аспектом ітеративної та інкрементальної розробки є постійний зворотний зв'язок між командою та клієнтом, що дозволяє вчасно реагувати на зміни в вимогах до проекту та покращувати якість продукту який потрібен клієнту.

У пункті "Самоорганізація команди" буде розглянуто, як вона допомагає команді ефективно виконувати завдання та розв'язувати проблеми. Важливим аспектом самоорганізації команди є збільшення мотивації та відповідальності кожного члена команди за успішне виконання проекту.

У пункті "Залучення клієнта" буде розглянуто, як взаємодія з клієнтом впливає на ефективність проекту та як цю взаємодію можна покращити(((. Важливим аспектом залучення клієнта є можливість вчасної валідації вимог та результатів роботи команди.

У пункті "Продуктовий беклог" буде розглянуто, що це за інструмент та як його використовувати для керування проектом. Важливим аспектом продуктового беклогу є можливість пріоритезації завдань та визначено (привертизація).

Основні принципи SCRUM - це набір цінностей до клієнта та практик, які дозволяють ефективно управляти проектами розробки програмного

забезпечення. Розроблені в рамках Agile-підходу до управління проектами, вони засновані на ідеї постійного зворотного зв'язку та самоорганізації команди. Основні принципи SCRUM включають:

Ітераційність - розробка програмного забезпечення здійснюється через серію коротких ітерацій (спринтів), кожен з яких має визначену мету та термін виконання. Після кожного спринту команда отримує зворотний зв'язок від замовника та може на його основі покращувати результати своєї роботи.

Інкрементальність - розробка програмного забезпечення здійснюється поетапно, з кожним спринтом додаються нові функції та можливості до програмного продукту.

Самоорганізація - команда самостійно вирішує, які завдання потрібно виконати та як краще це зробити, враховуючи вимоги замовника та обмеження проекту.

Залучення клієнта - замовник є активним учасником процесу розробки, він бере участь у мітингах та приймає рішення про пріоритети та вимоги до продукту.

Постійний зворотний зв'язок - команда отримує зворотний зв'язок від замовника після кожного спринту та на його основі вносить корективи до розробки.

Застосування цих принципів дозволяє зменшити ризики, пов'язані з розробкою програмного забезпечення, забезпечує швидкий розвиток продукту та підвищує його якість.

У методології SCRUM існує троє головних учасників: Product Owner, Scrum Master та Development Team. Кожен з них має свої відповідності та ролі, які спрямовані на досягнення мети проекту.

Product Owner - це особа, яка відповідає за розробку продукту, зокрема, за його вимоги та пріоритети. Product Owner забезпечує зв'язок між замовником та розробниками. Він відповідає за розробку Product Backlog - переліку функцій, які мають бути реалізовані в продукті, та їх пріоритетів.

Scrum Master - це особа, яка відповідає за виконання процесу SCRUM. Scrum Master забезпечує правильне виконання всіх етапів SCRUM, контролює взаємодію між учасниками, а також допомагає уникнути перешкод для виконання завдань.

Development Team - це команда розробників, яка відповідає за реалізацію функцій, які визначені в Product Backlog. Кожен член команди відповідає за свої завдання та забезпечує їх виконання у відповідному терміні.

Такий розподіл ролей дозволяє досягати ефективності в процесі розробки продукту. Наприклад, Product Owner забезпечує відповідність продукту вимогам замовника, Scrum Master - правильне виконання процесу SCRUM, а Development Team - реалізацію функцій. Однак, необхідно враховувати те, що такий розподіл ролей може мати й недоліки, зокрема, неефективність комунікації між учасниками, можливість виникнення конфліктів між різними ролями. Тому важливо забезпечити відкритість та співпрацю між учасниками процесу SCRUM для досягнення максимального успіху.

SCRUM містить декілька етапів та процесів, кожен з яких має свої особливості та мету. У цьому пункті ми розглянемо кожен з них детальніше.

Sprint Planning

Перший етап у SCRUM - Sprint Planning. Його головною метою є планування наступної ітерації проекту, тобто спринта. На цьому етапі Product Owner описує беклог продукту, а Scrum Team визначає завдання, які вони можуть виконати протягом спринту. На завершення Sprint Planning формується Sprint Backlog, який містить завдання, які потрібно виконати під час спринту.

Daily Scrum

Цей процес проводиться щодня протягом спринту, його мета полягає в тому, щоб підвищити комунікацію в команді та забезпечити, щоб усі учасники були на одній хвилі. Кожен учасник команди відповідає на три запитання: що я робив вчора, що я планую робити сьогодні та які перешкоди мені

заваджають. Це допомагає всім учасникам усвідомити, які завдання вони мають виконати, і як вони можуть покращити свою роботу.

Sprint Review

Цей етап проводиться в кінці кожного спринту і має на меті перевірити, чи були досягнуті цілі спринту. Учасники команди демонструють виконані завдання та обговорюють, які питання виникли під час роботи. Також на цьому етапі може проводитися оцінка роботи команди, щоб визначити, які аспекти можна покращити наступного разу.

Sprint Retrospective є засіданням команди, на якому відбувається аналіз роботи команди за попередній спринт. Команда дискутує про те, що було зроблено добре, що можна покращити та які проблеми були виявлені. На основі цих даних команда приймає рішення про те, які зміни потрібно внести в процес розробки для досягнення кращих результатів. чого саме переваги.

Переваги:

- підвищення ефективності процесу розробки шляхом внесення
- змін на основі виявлених проблем та недоліків;
- підвищення мотивації команди шляхом врахування її думки та внесення змін на основі її рекомендацій;
- покращення комунікації в команді, через те, що кожен член команди має можливість висловити свої думки та ідеї.

Недоліки:

- неможливість забезпечити виконання всіх запланованих завдань, якщо виявляться проблеми, які потребують додаткового часу та ресурсів для їх вирішення;
- необхідність проведення додаткових засідань, що може вимагати додаткових зусиль та часу від учасників команди;
- загалом, Sprint Retrospective є важливим етапом в процесі SCRUM, який дозволяє команді аналізувати свою роботу та вносити зміни для поліпшення результатів. Недоліки цього етапу пов'язані з можливими труднощами у виконанні запланованих завдань, а також з необхідністю

проведення додаткових засідань, але вони зазвичай виправдовуються підвищенням ефективності та мотивації команди.

Засоби та техніки є важливими елементами SCRUM, які допомагають команді виконувати процес розробки і відстежувати прогрес. Основні засоби та техніки SCRUM включають:

Product Backlog - це список усіх функціональних вимог, які повинен мати продукт. Product Owner відповідає за створення та підтримку цього списку. Кожен елемент списку містить опис вимоги та її пріоритет.

Product Backlog дозволяє команді відстежувати прогрес проекту та додавати нові задачі в список по ходу роботи.

Список може бути використаний для планування спринту та визначення пріоритетів у виконанні робіт.

Product Backlog може допомогти команді розуміти загальну картину роботи над проектом.

Sprint Backlog - це список завдань, які мають бути виконані командою під час спринту. Кожен елемент містить опис завдання та оцінку часу, необхідного для його виконання.

Burn-down Chart - це графік, який відображає прогрес команди під час спринту. Він показує, скільки часу залишилося до кінця спринту та скільки роботи ще потрібно зробити. Це дозволяє команді відстежувати свій прогрес та прогнозувати, чи буде виконано всю роботу вчасно. На рис. 1.1 Burn-down Chart

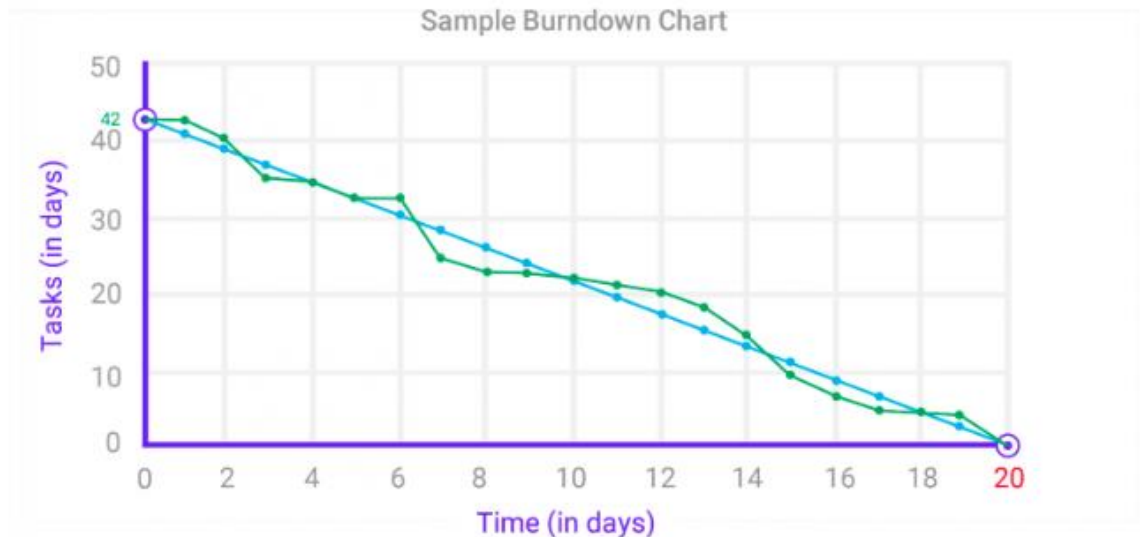


Рисунок 1.1 – Burn-down Chart

Переваги:

- burn-down Chart дозволяє візуалізувати прогрес роботи команди та визначити, наскільки ефективно вона працює;
- графік може бути використаний для попередження проблем з терміновими завданнями та підвищення ефективності команди у виконанні робіт;
- цей інструмент може допомогти команді реалізувати кращу оцінку робіт, встановивши реалістичні цілі та прогнозуючи відповідний прогрес.

Недоліки:

- burn-down Chart не дозволяє відрізнити якість виконання робіт та не враховує фактори, які можуть впливати на прогрес виконання проекту;
- якщо розмір задачі, яку необхідно виконати, змінюється протягом спринту, графік може бути неправильним та дати неточну картину прогресу роботи команди.

1.2 Аналіз і порівняння SCRUM з іншими методологіями управління проектами

У цьому розділі магістерської роботи проводиться огляд та порівняння SCRUM з іншими методологіями управління проектами. Для цього розглядаються такі методології та моделі як Waterfall, Kanban та Lean. Для кожної з методологій розглядається її основні принципи та характеристики, що дозволяє зрозуміти, які етапи та процеси включені в процес розробки проекту. Також розглядаються переваги та недоліки використання кожної методології.

Waterfall це модель управління проектами, в якій кожен етап розробки проекту виконується послідовно, від одного етапу до іншого. У цій методології детально планується весь проект на початку розробки і наступні етапи реалізуються тільки після завершення попередніх. Перевагою цієї методології є те, що вона добре підходить для проектів з точно визначеними вимогами та дедалі менш використовується в управлінні проектами через свою неспроможність адаптуватися до змін під час розробки проекту.

На рис. 1.2 це можна побачити

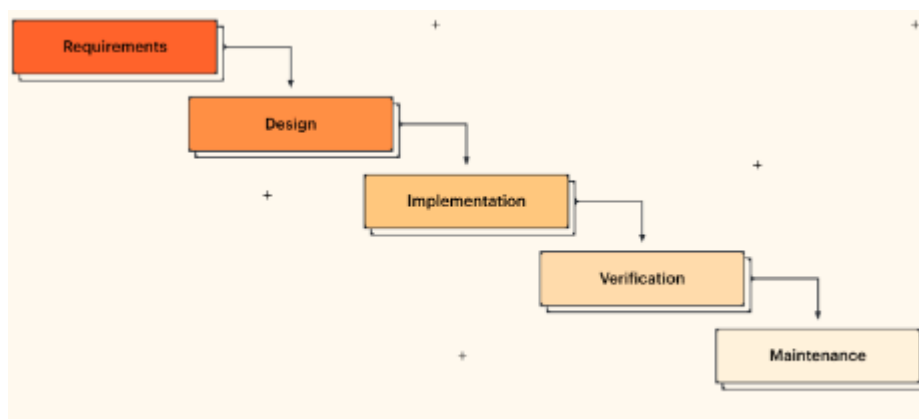


Рисунок 1.2 – Waterfall

Канбан це гнучка методологія управління проектами, яка зосереджується на оптимізації потоків роботи та мінімізації витрат. У цій

методології використовуються візуальні дошки, на яких зображені завдання, що потрібно виконати, і які переміщуються від початкового стану до завершеного стану. На рис. 1.3 це можна побачити

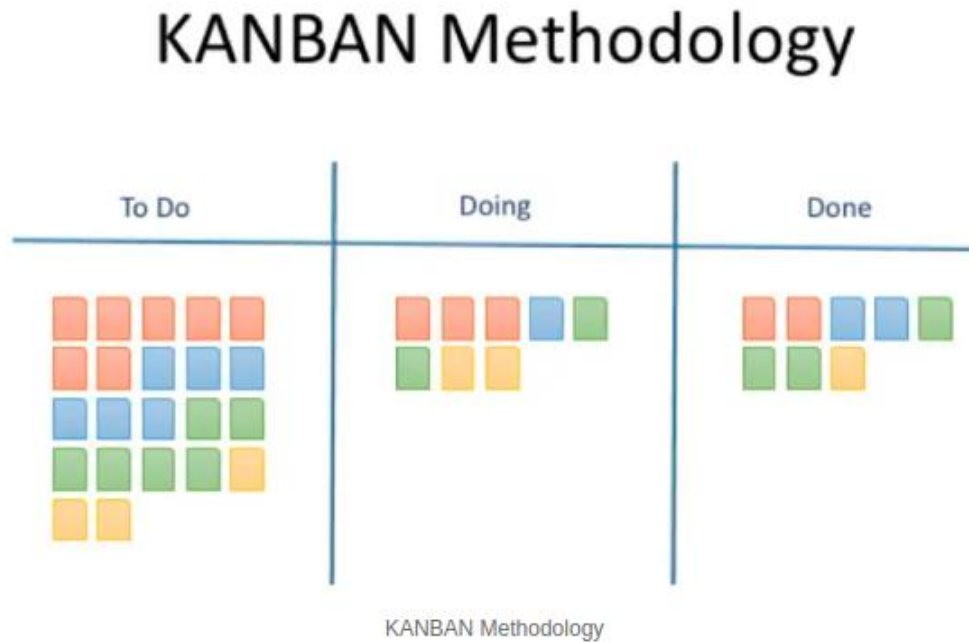


Рисунок 1.3 – Методологія Канбан

Переваги:

- Забезпечується більш ефективне використання ресурсів, тому що завдання виконуються за потребою, а не в заздалегідь визначеному порядку;
- Забезпечується найвища якість продукту, оскільки візуальний контроль над кожним етапом дозволяє виявляти та виправляти помилки швидше;
- Забезпечується більш гнучкий підхід до управління проектом, оскільки потік завдань може бути легко змінений за потребою.

Недоліки:

- Не підходить для проектів з багатьма взаємозалежними завданнями;
- Не забезпечується достатньо формалізований процес управління проектом;

- Можливість виникнення затримок, якщо не вистачає збалансованості між завданнями.

Lean гнучка методологія управління проектами, яка зосереджується на ефективності та мінімізації втрат. Головні принципи Lean включають у себе:

- Зосередженість на потребах клієнта: клієнт завжди повинен бути у центрі уваги, та його потреби та вимоги мають бути у фокусі всієї діяльності;
- Елімінація витрат: необхідно ефективно використовувати ресурси та процеси, щоб мінімізувати витрати;
- Контроль якості: необхідно забезпечувати високу якість продукту, що відповідає потребам клієнта;
- Постійне покращення: процеси та продукти повинні постійно покращуватися, з метою підвищення ефективності та задоволення потреб клієнта.

Переваги Lean на рисунку 1.4 містять зниження витрат, покращення якості продукту та ефективності процесів, забезпечення задоволення клієнта та прискорення випуску продукту на ринок. Однак, недоліком Lean є те, що вона може бути менш гнучкою та менш адаптивною до змін, що може бути недоцільним в деяких ІТ-проектах, де вимагається швидкий розвиток та зміна вимог.



Рисунок 1.4 – Lean методологія

Для порівняння SCRUM з іншими методологіями управління проектами можна використовувати різні критерії, такі як швидкість доставки продукту, якість результуючого продукту, мінімізація ризиків та забезпечення співпраці між різними групами проекту.

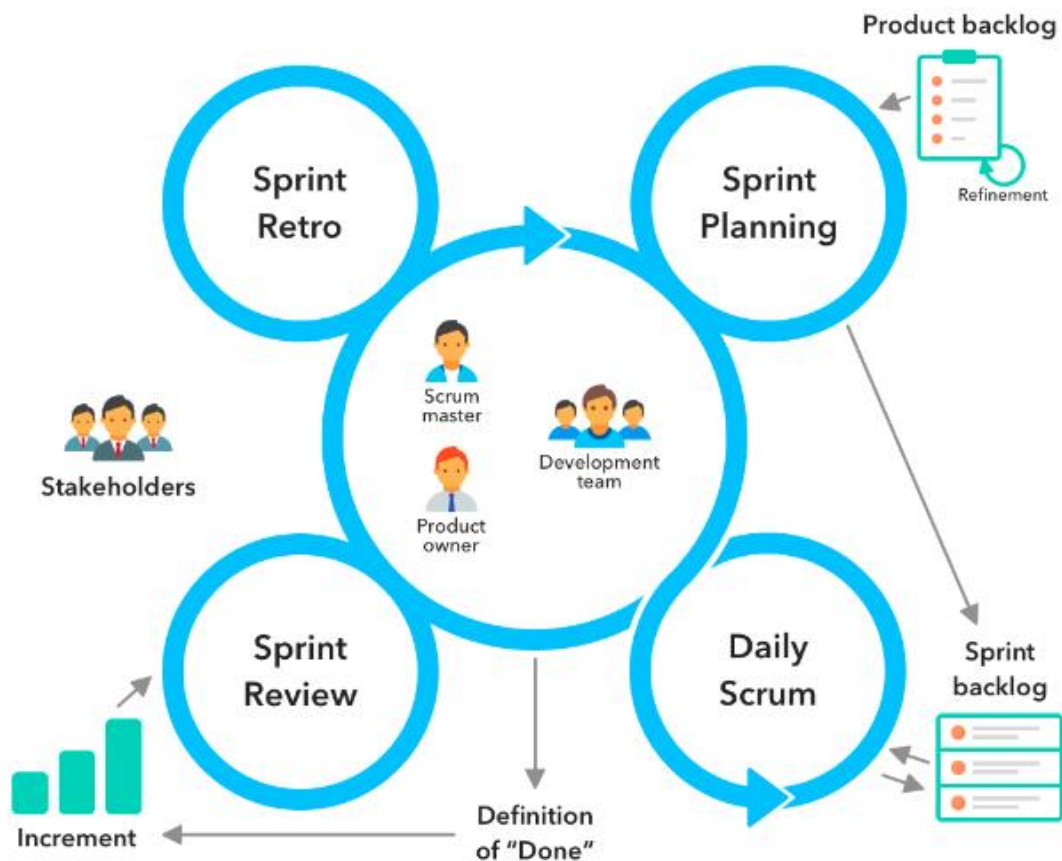


Рисунок 1.5 – SCRUM методологія

На рис. 1.5 можна побачити очевидні відмінності між SCRUM та іншими методологіями є спрямованість на Agile-підхід у SCRUM, що дозволяє команді змінювати напрямок розробки продукту відповідно до зміни умов ринку та потреб користувачів. У порівнянні з традиційним Waterfall-підходом, у якому послідовність етапів розробки фіксована, SCRUM надає більшу гнучкість та адаптивність в процесі розробки продукту.

Щодо швидкості доставлення продукту, SCRUM, як правило, використовує короткі спринти для розробки та тестування продукту від 1 до 4 тижнів, тоді як у Waterfall можуть бути значно більші терміни від 6 до 18 місяців. У порівнянні з Kanban та Lean, SCRUM також може бути більш ефективним шляхом своєї структурованої та організованої методології.

Щодо якості продукту, SCRUM зосереджується на тестуванні продукту на кожному етапі, що дозволяє виявляти та виправляти помилки на ранніх

етапах розробки. У порівнянні з Waterfall, де тестування зазвичай відбувається тільки після завершення розробки продукту, SCRUM може забезпечити вищу якість продукту та зменшення ризиків, пов'язаних з його випуском.

Також варто порівняти методології за їхньою здатністю адаптуватися до змін у вимогах та умовах проекту, а також за ефективністю комунікації між учасниками проекту. Крім того, слід проаналізувати, як кожна з методологій розподіляє відповідальності між учасниками проекту та як вони співпрацюють між собою.

Наприклад, Waterfall методологія має визначену послідовність етапів розробки продукту, що може бути корисно для проектів з чітко визначеними вимогами та обмеженим бюджетом. Однак, ця методологія не дозволяє змінювати вимоги під час розробки продукту, що може призвести до затримок у виконанні проекту та погіршення якості продукту.

Agile це сімейство методології, такі як Scrum та Kanban, зосереджуються на швидкій доставці продукту та забезпеченні змінної вимог. Вони забезпечують ефективну комунікацію між учасниками проекту та покращують співпрацю між різними групами. Однак, ці методології можуть бути менш ефективними для проектів зі складними вимогами або з обмеженими ресурсами.

Lean методологія, зокрема, зосереджується на мінімізації витрат та зменшенні витрат на запаси. Ця методологія може бути корисною для проектів з обмеженим бюджетом або для організацій з високим рівнем конкуренції. Однак, вона може бути менш ефективною для проектів зі складними технічними вимогами або для проектів зі значною кількістю залежностей між завданнями.

Для порівняння якості результуючого продукту, слід розглянути методики контролю якості та тестування, які використовуються в різних методологіях. SCRUM зазвичай покладає відповідальність за контроль якості на всю команду, зокрема на QA-інженерів, тоді як в Waterfall цю роль виконує

окремий QA-відділ. У Kanban тестування може здійснюватися в режимі реального часу під час розробки продукту.

Щодо зменшення ризиків, Agile методології, включаючи SCRUM, передбачають часті зустрічі та звіти з командою, що дозволяє швидко виявляти та виправляти проблеми. Waterfall, з іншого боку, виконується за лінійною послідовністю, що може призвести до затримок та невдач у випадку несправності в одному з етапів.

Нарешті, для порівняння забезпечення співпраці між різними групами проекту, можна розглянути використання методів комунікації та співпраці в різних методологіях. SCRUM передбачає регулярні зустрічі, які допомагають забезпечити співпрацю між різними групами проекту, тоді як в Waterfall комунікація може бути менш інтенсивною та менш ефективною.

Оцінка відповідності SCRUM вимогам сучасного ринку є одним з важливих пунктів дослідження моделей та методів оцінювання ефективності використання SCRUM в управлінні IT-проектами. Для оцінки відповідності SCRUM вимогам сучасного ринку необхідно провести аналіз сучасних тенденцій ринку та потреб користувачів.

Основні вимоги до управління проектами на сучасному ринку полягають у забезпеченні швидкості доставки продукту, високій якості результуючого продукту, зменшенні ризиків та забезпеченні співпраці між різними групами проекту. SCRUM дозволяє ефективно використовувати доступні ресурси та забезпечує доставку продукту в строк, що дозволяє йому відповідати основним вимогам сучасного ринку.

Для більш точної оцінки відповідності SCRUM вимогам сучасного ринку необхідно провести порівняння з іншими методологіями управління проектами та оцінити, наскільки вони відповідають вимогам ринку. Також важливо провести опитування користувачів та професіоналів, що займаються управлінням проектами, щоб з'ясувати їхні погляди на відповідність SCRUM вимогам сучасного ринку та іншим методологіям управління проектами.

Крім того, SCRUM дозволяє бути більш гнучким та адаптивним до змін вимог та потреб користувачів. Це особливо важливо на сучасному ринку, де вимоги змінюються дуже швидко, а компанії повинні бути готовими до таких змін, щоб забезпечити конкурентні переваги.

Оцінка відповідності SCRUM вимогам сучасного ринку та вимогам може бути проведена шляхом порівняння з іншими методологіями управління проектами, які також відповідають вимогам сучасного ринку. Також можна провести аналіз відгуків користувачів та фахівців з проектного менеджменту про те, наскільки SCRUM допомагає їм досягати мети та реалізувати проекти на сучасному ринку.

Отже, оцінка відповідності SCRUM вимогам сучасного ринку є важливим етапом у дослідженні ефективності використання цієї методології в управлінні ІТ-проектами.

1.3 Аналіз ефективності методології SCRUM в управлінні ІТ-проектами

Методологія SCRUM є однією з найбільш популярних та ефективних методологій управління проектами в галузі ІТ. У даному розділі буде проведено аналіз ефективності методології SCRUM в управлінні ІТ-проектами.

Почнемо з переваг методології SCRUM. Перш за все, вона спрямована на підвищення продуктивності команди розробників шляхом зменшення часу, що витрачається на планування та виконання проекту. SCRUM дозволяє команді розробників зосередитися на головних задачах проекту та визначити пріоритетність робіт. Крім того, методологія SCRUM передбачає неперервну комунікацію між усіма учасниками проекту, що сприяє швидкому виявленню та розв'язання проблем.

Далі, SCRUM є досить гнучкою методологією, яка дозволяє команді розробників змінювати пріоритети та плани проекту в залежності від зміни

вимог клієнта або зовнішніх факторів. Це дає змогу зберігати стабільність та ефективність проекту навіть при змінах.

Крім цього, SCRUM дозволяє зменшити ризики, пов'язані з розробкою проекту, шляхом чіткого планування та регулярної перевірки прогресу. Це дозволяє команді розробників виявляти проблеми та ризики на ранніх етапах проекту та швидко реагувати на них.

Проте, методологія SCRUM має й деякі недоліки. Перш за все, вона може бути складною для впровадження та використання, особливо для команд, які не мають достатнього досвіду в управлінні проектами. Крім того, SCRUM вимагає від учасників проекту високої самодисципліни та відповідальності. Це може бути складно для команд, які не мають достатнього досвіду в роботі в умовах SCRUM.

Додатково, SCRUM не підходить для всіх типів проектів. Наприклад, вона може бути менш ефективною для проектів з високим рівнем необхідності в попередньому проектуванні та детальному плануванні, оскільки SCRUM передбачає гнучкість та швидкість в управлінні проектом.

Нарешті, важливо відзначити, що ефективність використання методології SCRUM в управлінні IT-проектами залежить від багатьох факторів, таких як досвід команди розробників, складність проекту, вимоги клієнта та зовнішні фактори.

Отже, можна стверджувати, що методологія SCRUM є ефективним інструментом управління IT-проектами, здатним підвищити продуктивність команди розробників та зменшити ризики, пов'язані з розробкою проекту. Проте, для ефективного використання SCRUM в управлінні IT-проектами важливо мати достатній рівень досвіду та відповідальності учасників проекту.

1.4 Постановка задачі

Постановка задачі на кваліфікаційну дипломну роботу "Дослідження моделей і методів оцінювання ефективності використання SCRUM в управлінні IT-проектами" полягає в дослідженні ефективності методології SCRUM в управлінні IT-проектами та за допомогою методів оцінювання ефективності використання SCRUM.

Для того, щоб досягнути поставленої мети необхідно вирішити наступні завдання:

Дослідити з теоретичними засадами методології SCRUM

Ознайомитись з чинними або існуючими дослідженнями та публікаціями, які присвячені аналізу ефективності використання методології SCRUM в управлінні IT-проектами, та визначити їхні переваги та недоліки.

Визначити критерії ефективності використання SCRUM в управлінні IT-проектами, які відображають результативність роботи команди розробників та загальну якість проекту.

2. МЕТОДОЛОГІЇ ТА МОДЕЛІ ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ

SCRUM

2.1 Опис формули “Швидкість доставки продукту до замовників”

Для порівняння ефективності використання Waterfall та Agile можна використовувати різні формули та моделі.

Однією з найбільш поширених моделей є "Швидкість доставки продукту до замовників" (Product Delivery Rate), яка оцінює, скільки продукту команда розробників може розробити за одиницю часу. Для розрахунку швидкості доставки продукту до замовників можна використовувати наступну формулу:

$$\text{Product Delivery Rate} = (\text{Total Number of Features Delivered} / \text{Total Time})$$

Щоб порівняти ефективність використання методологій Waterfall і Agile, можна скористатися формулою Product Delivery Rate, яка розраховує швидкість доставки продукту клієнтам. Ця формула представляє кількість функцій, які були успішно доставлені протягом певного періоду часу.

У методології розробки програмного забезпечення Waterfall проект виконується в лінійному порядку, причому кожна фаза завершується по черзі і не починається, доки не завершиться попередня фаза. Це означає, що весь проект має бути повністю розроблений і визначений заздалегідь до початку розробки. Такий підхід може забезпечити високу якість продукту, але може затягнути процес виконання проекту.

У Agile-методології розробка виконується ітераціями. Кожна ітерація — це повний цикл розробки, який включає планування, розробку, тестування та випуск нових функцій. Це дозволяє команді зосередитися на важливих функціях і забезпечити їх своєчасне виконання, підвищуючи гнучкість і здатність адаптуватися до змін.

Застосуємо формулу швидкості доставки продукту, щоб порівняти методології Waterfall і Agile. Припустимо, проект мав 100 функцій, і вони були успішно доставлені протягом 10 місяців за допомогою методології Waterfall, а

80 функцій були успішно доставлені за допомогою методології Agile за 10 місяців.

За допомогою формули можна порівняти ефективність методології Waterfall і Agile. Наприклад, якщо в проекті Waterfall було розроблено 100 функцій, а загальний час, витрачений на проект, становив 12 місяців, ефективність методології можна розрахувати за формулою:

Швидкість доставки товару = $(100 / 12) = 8,33$ функції на місяць

У методології Agile, якщо 10 ітерацій було проведено протягом 12 місяців, і кожна ітерація містила 10 функцій, то ефективність методології можна розрахувати за формулою:

Швидкість доставки продукту = $(10 \text{ ітерацій} * 10 \text{ функцій}) / 12 \text{ місяців} = 8,33$ функції на місяць

Отже, за допомогою цієї формули можна зробити висновок, що з точки зору ефективності використання методології Agile не дає істотних переваг перед методологією Waterfall. Однак важливо зазначити, що обидві методики мають свої переваги та недоліки

2.2 Дослідження методології та методів Agile

Таблиця 2.1 - "Чотири цінності Agile-управління проектами"

Цінності	Опис
Індивідуали та взаємодія	Люди і взаємодія важливіші процесів та інструментів
Робочий продукт	Робочий продукт важливіший вичерпної документації
Співпраця замість контракту	Співпраця з замовником важливіша узгодження умов контракту
Реагування на зміни	Готовність до змін важливіша проходження попереднім планом

Існують "12 принципів Agile-управління проектами" які представлені в таблиці 2.2

Таблиця 2.2 - "12 принципів Agile-управління проектами"

Номер принципу	Опис
1	Задоволення потреб користувача, шляхом ранньої та неперервної постачі цінності
2	Вітання змін у вимогах, навіть на пізній стадії розробки
3	регулярне постачання продукту, з періодичністю від кількох тижнів до кількох місяців
4	Співпраця між бізнесом та розробниками, протягом проекту
5	Забезпечення швидкої та ефективної комунікації між розробниками
6	Використання простої інформаційної структури
7	Співпраця замість конфронтації між розробниками та замовниками
8	Реагування на змін вимог
9	Регулярне вдосконалення та оптимізація процесу розробки
10	Простота, як основа розробки
11	Самоорганізація команди
12	Регулярний ретроспективний аналіз процесу розробки

Як і інші популярні методології розробки та управління проектами, Agile з'явився порівняно недавно в США. На відміну від СРМ і ССРМ, за появу гнучкої методології розробки відповідальна відразу ціла група людей — 17 американських ІТ-фахівців зі штату Юта. Разом з «Маніфестом гнучкої розробки ПО», в якому вперше прозвучав термін «Agile» вони прописали 12 принципів Agile-розробки.

Їх суть зводиться до таких ключових моментів, що визначає характер гнучкого методу розробки:

- Люди і взаємодія важливіші процеси та інструментів;
- Робочий продукт важливіший вичерпної документації;
- Співпраця з замовником важливіше узгодження умов контракту;

- Готовність до змін важливіше проходження попереднім планом.

Agile став основою для цілого ряду гнучких методик, серед яких найбільш відомі Scrum, Lean та екстремальне програмування.

Scrum — методологія гнучкої розробки на основі Agile, в основі якого лежить «спринт» — відрізок від 1 до 4 тижнів, після закінчення якого повинна бути отримана робоча версія продукту. Lean — метод, який виріс на основі системи управління виробництвом Toyota Production System. В його основі — філософія постійного вдосконалення на всіх рівнях організації, де одне з ключових понять — цінність (то, за що готовий платити замовник). Екстремальне програмування (XP) — одна з Agile — методик, де важлива роль відводиться періодичній грі в планування із залученням замовника. Вона дозволяє визначити недоліки попередньої ітерації, пріоритетність задач, бажану функціональність продукту з урахуванням побажань замовника.

Переваги та недоліки методу Agile:

- короткі і зрозумілі ітерації: цикли розробки тривають від 2 тижнів до 2 місяців, по закінченню яких замовник отримує робочу версію продуктів;
- високий ступінь залучення виконавців, організаторів і замовників проекту;
- головним стоїть робочий продукт як основний показник прогресу — це можна розглядати як плюс, так і мінус, адже в такому випадку до команди проекту висуваються високі вимоги по самоорганізації;
- мінімізація ризиків завдяки гнучкій системі внесення змін;
- Стимулювання постійних змін проекту: гнучкість розробки продукту може призвести до того, що він ніколи не дійде до фінальної версії;
- Підвищені вимоги до кваліфікації і досвіду команди: крім безпосередньо створення продукту команда повинна аналізувати можливі способи поліпшення ефективності власної роботи, безперервно обмінюватися інформацією по проекту, бути вмотивованою і самоорганізованою. Далеко не завжди ресурси проекту дозволяють залучити таких фахівців:

- Філософський характер методології: Agile — це не чітка інструкція до дії, а ціла філософська концепція. Команда не може механічно застосувати механіку «гнучкої» розробки, потрібно прийняти ключові принципи системи;
- Складність підрахунку підсумкової суми робіт: стимуляція змін і вдосконалення кінцевого продукту призводить до плаваючого значення вартості проекту.

2.3 Дослідження методології та методів Waterfall

Методика Waterfall (водоспадна система розробки) — дітище Вінстона Уолкера Ройса, директора Lockheed Software Technology Center в Остіні (штат Техас, США), піонера в області розробки програмного забезпечення.

Водоспадна модель розробки передбачає послідовне проходження процесу, розбитого на стадії. Перехід до нового етапу можливий тільки після завершення попереднього.

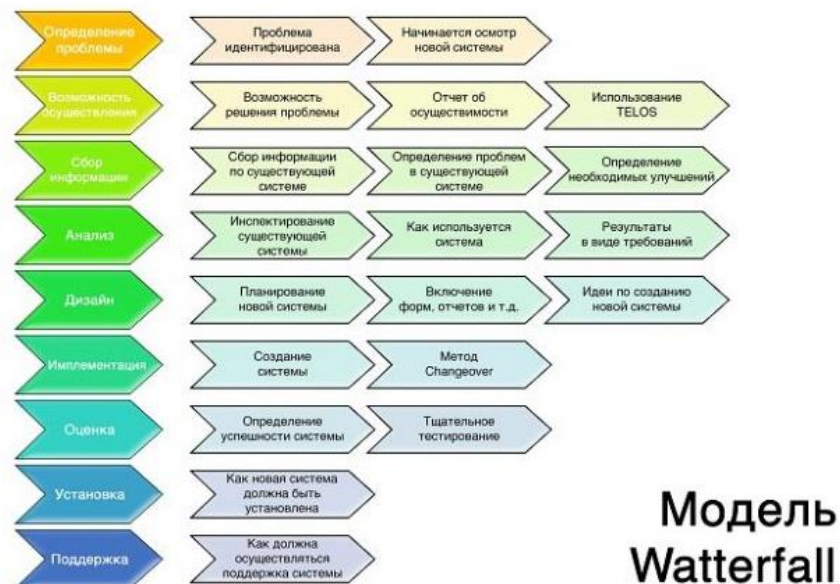


Рисунок 2.2 – Модель Waterfall

В оригінальній роботі Уолкера "Managing the development of large software systems«описані 6 стадій розробки продукту, які в 1985 році Департамент захисту США закріпив в стандартах роботи з розробниками програмного забезпечення:

- Системні і програмні вимоги : закріплюються в PRD (документі вимог до продукту).
- Аналіз : втілюється в моделях, схемах і бізнес-правилах.
- Дизайн : розробляється внутрішня архітектура програмного забезпечення, способи реалізації вимог. Це не тільки про інтерфейс і зовнішній вигляд ПО, а й про його внутрішню структурну логіку.
- Кодінг : безпосередньо пишеться код програми, йде інтеграція програмного забезпечення .
- Тестування : баг-тестери (тестувальники) перевіряють фінальний продукт, заносючи в трекери відомості про дефекти коду програми або функціоналу. У разі помилок і наявності часу/фінансів відбувається виправлення багів.
- Операції : продукт адаптується під різні операційні системи, регулярно оновлюється для виправлення виявлених користувачами багів і додавання функціоналу. В рамках стадії також здійснюється технічна підтримка клієнтів.

Переваги та недоліки Waterfall

В число найбільших переваг методики Waterfall увійшли:

- зрозуміла і проста структура процесу розробки: це знижує поріг входження для команд
- зручна звітність: можна легко відстежити ресурси, ризики, витрачений час і фінанси завдяки суворій етапності процесу розробки та детальної документації проекту

- стабільність задач: задачі, які стоять перед продуктом, ясні команді з самого початку розробки, і залишаються незмінними протягом усього процесу
- оцінка вартості та термінів задачі проекту: терміни випуску готового продукту, як і його підсумкова вартість можуть бути прораховані до моменту запуску розробки.

Серед недоліків водоспадного методу можна виділити:

- позбавлений гнучкості процес: так, якщо проект вимагає більше тимчасових і фінансових ресурсів, чим можливо, то під ніж піде фаза тестування. Згідно з дослідженнями консалт-групи Rothman, вартість виправлення багів після випуску продукту вища в середньому в 20 разів, ніж під час повноцінного багатоетапного тестування в процесі розробки;
- «стійкість» до змін: жорсткий каркас з етапів розробки і умова надання тільки готового продукту визначають неможливість вносити зміни під час розробки;
- інерційність: на перших стадіях прогноз тимчасових і фінансових витрат може змінитися в бік збільшення, але змінити проект в сторону оптимізації витрат, зміни функціоналу або концепції до випуску готового продукту неможливо;
- підвищений ризик: класична система тестування має на увазі окреме тестування кожного з компонентів проекту, в тому числі, у взаємодії з іншими. При використанні Waterfall відбувається тестування готового продукту.

Частково недоліки водоспадної моделі розробки виправлені в модифікаціях Waterfall: Сашими, Waterfall з субпроектів та водоспадна модель розробки зі зниженням ризику.

Сашимі або водоспадна модель з фазами, що нашаровуються — найвідоміша серед них. У ній етапи, як і в оригінальній методиці, йдуть один за одним, але при цьому перекриваються одна іншою в часі.

Waterfall з субпроектів — модель, в якій ви працюєте з трьома великими блоками: концептуалізацію, проектуванням вимог і архітектурною структурою продукту. Потім для кожного з них ви проходитье стадії (субпроекти) детального проектування, кодування і тестування. В кінці проводиться інтеграція всіх компонентів на етапі тестування системи.

Таблиця 2.3 Порівняння Agile та Waterfall

	Agile	Waterfall
Суть	Гнучка модель розробки, заснована на ітеративних принципах	Каскадна система розробки, заснована на жорсткій послідовності процесу розробки
Дата створення	2001 р	1956 року, 1961 р 1970 г.
Розробники	Група ІТ-фахівців (США)	Г. Беннінгтон, Хозьєр, В. Уолкер Ройс
Принципи застосування	найвищий пріоритет в задоволенні потреб замовника протягом всього проекту команда і замовник щодня взаємодіють між собою і один з одним робочий продукт — головний показник прогресу роботу можна довірити тільки самоорганізованій, мотивованій команді оптимальні терміни випуску робочого продукту	<ul style="list-style-type: none"> – жорстка послідовність етапів розробки – перехід до нового етапу — тільки після успішного завершення попереднього – фіксована вартість продукту – замовник не залучається до безпосереднього процесу розробки – зміни можуть бути внесені тільки після завершення всього процесу розробки.

Таблиця 2.4 Плюси та мінуси Agile та Watterfall

Плюси	<ul style="list-style-type: none"> – високий рівень взаємодії між членами команди проекту – швидкий результат (робочий код) в результаті «спринтів» – стимулювання змін і поліпшень продукту під час його розробки – безпосереднє залучення замовника до робочого процесу. 	<ul style="list-style-type: none"> – зрозуміла і чітка схема робочого процесу – можливість підрахунку точної кількості витрачених на проект ресурсів – не вимагає витрат по налагодженню комунікацій між усіма членами команди.
Мінуси	<ul style="list-style-type: none"> – ризик нескінченних змін продукту – велика залежність від рівня кваліфікації та досвіду команди – практично неможливо точно підрахувати підсумкову вартість проекту. 	<ul style="list-style-type: none"> – пріоритет формального підходу до послідовності процесу роботи – неможливість внесення змін замовником до закінчення розробки продукту – в разі нестачі ресурсів страждає якість проекту через скорочення етапу тестування.
Компанії-практики	Unilever, ряд банків (Альфа Банк, Home Credit, Райффайзен Банк і т.д.)	Cisco Ericsson AB, Toyota (до 2010)

Таблиця 2.5 Чому підійде Agile та Waterfall

Підійде вам, якщо ...	<ul style="list-style-type: none"> – над проектом працює досвідчена, висококваліфікована команда – ви працюєте над стартапом – потрібно швидко отримати робочу версію продукту – замовник виступає в якості партнера, а не інвестора – продукт розробляється в сфері, схильній до постійних змін. 	<ul style="list-style-type: none"> – більша частина або вся робота над проектом проводиться на аутсорсі – у вас є чітка концепція продукту, який хочете отримати – ви не обмежені в часі і ресурсах створення продукту – створення продукту або бізнесу побудовано на дотриманні суворій послідовності виконання задач.
Чи не підійде , якщо...	<ul style="list-style-type: none"> – ви не готові витратити додаткові ресурси на налагодження щоденної стабільної комунікації між усіма учасниками процесу – продукт повинен бути створений до 	<ul style="list-style-type: none"> – ви хочете створити інноваційний продукт або великий проєкт – ви не впевнені в концепції пропонованого проєкту – фінансові ресурси не є ключовим обмежувачем в вашому проєкті.

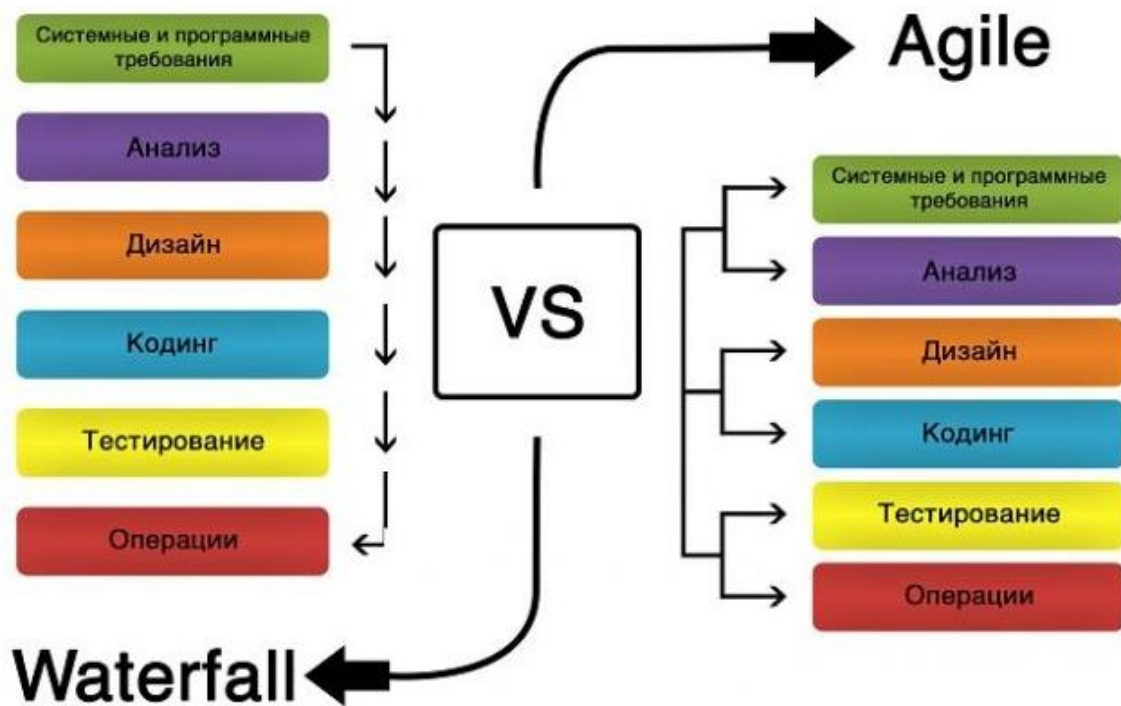


Рисунок 2.3 - Agile та Waterfall

Agile і Waterfall — дві абсолютно різні методики розробки та управління проектами. Кожна з них породила десятки модифікацій і методів.

З урахуванням особливостей кожної з методик і вашого бізнесу, а також на основі критеріїв ризику, часу і залучення зацікавлених осіб, ви зможете можливо самостійно визначити ефективну методологію.

2.4 Порівняльний аналіз методологій управління проектами

У звіті групи Standish Group International, Inc. проектів, які вивчалися з 2013 по 2017 рік, показано загальний розбіг успіхів, проблем і невдач для Agile і Waterfall. За цей період ймовірність успіху в Agile проектах була близько вдвічі вищою, а ймовірність провалу менша на 1/3, як показано на рис. 2.1.

PROJECT SUCCESS RATES AGILE VS WATERFALL



Рисунок 2.4 — Звіт групи Standish Group International, Inc. за 2013-2017 роки

Хоча дані Standish Group за 2020 рік чітко свідчать про успіх Agile проти Waterfall Projects, це не змінює напрямок. Насправді Standish Group чітко виступає за гнучкі проекти замість водоспадних або традиційних проектів.

Standish Group аналізують технологічні проекти понад 25 років. Їх перший Chaos Report вийшов у 1994 році, ще до того, як термін Agile був використаний для підходів до розробки програмного забезпечення. З того часу понад 50 000 технологічних проектів було додано до бази даних Standish Group.

Протягом перших 15 років досліджень Standish Group визначення успіху було таким самим, як і в Інституті управління проектами. Пам'ятайте про час, бюджет, обсяг? Більш відомий як «наскільки добре ми спланували, а потім реалізували цей план?» Agile Manifesto прямо суперечив принципу «сплануй роботу, працюй за планом». Цінності Agile стверджують, що агілісти цінують реагування на зміни.

У 2015 році Standish Group розширила своє визначення успіху до того,

що вони назвали сучасним показником успіху. Нове визначення розширено для включення результатів або впливу проекту. Вони дивилися не тільки на доставку в порівнянні з планом, але й включали отриману цінність і задоволеність клієнтів. Це не просто перевірка нашого процесу планування чи виконання плану.

Отже, «сучасне» визначення успіху Standish Group включає наступні шість окремих атрибутів успіху:

- Вчасно
- Про бюджет
- В ціль (обсяг)
- На меті
- Значення
- Задоволеність користувачів

Дані Standish Group також сегментували успіхи та невдачі відповідно до розміру проекту, Agile та Waterfall Projects. Як можна собі уявити, невеликі проекти досягають успіху частіше, ніж великі. Менші проекти мають менший ризик і коротші цикли зворотного зв'язку. Ці дані показують, наскільки успішнішими є менші проекти. Для гнучких проектів малі проекти мають у 3 рази більше шансів на успіх, ніж великі. А для проектів Waterfall рівень успішності малих проектів у 6 разів вищий, ніж великих.

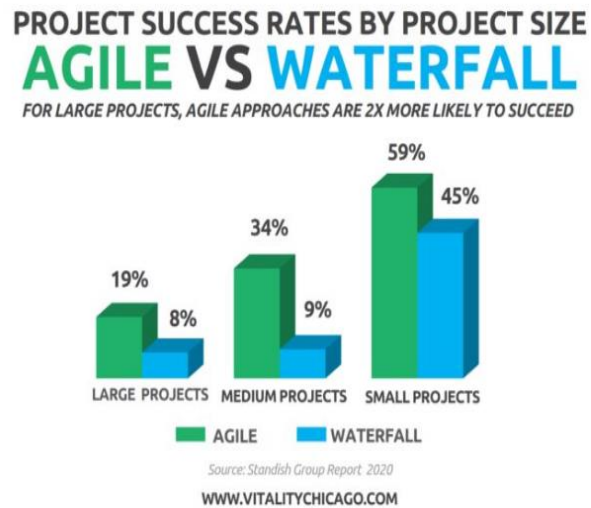


Рисунок 2.5 — Звіт групи Standish Group International, Inc. по розміру проектів

Одним із ключових висновків тут є уникання великих проектів. Розбиваючи великі проекти на менші, ви можете значно зменшити ризик і підвищити шанси на успіх.

Перехід від Waterfall до Agile не є тривіальним, але цілком можливим. Багато організацій намагалися та зазнали невдачі, тоді як інші досягли успіху завдяки підходам Agile. Якщо ви готові дізнатися більше, ось деякі ресурси, які можуть вам допомогти.

3 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ЗА ДОПОМОГОЮ SCRUM

3.1 Проектування інформаційної системи

Розробка інформаційної системи є складним та багатоетапним процесом, який керується за допомогою діаграми Ганта. Ця діаграма дозволяє візуалізувати та контролювати всі етапи проектування інформаційної системи, включаючи завдання, терміни виконання та використані ресурси. Використання методів проектування забезпечує ефективну послідовність етапів розробки, яка може бути успішно реалізована завдяки діаграмі Ганта.

Перед тим, як створити діаграму Ганта, потрібно провести декомпозицію задач проекту. Декомпозиція - це процес розбиття складних задач на менші та більш прості елементи. У контексті діаграми Ганта декомпозиція допомагає поділити проект на менші та керовані етапи, що дозволяє зробити проект більш передбачуваним та керованим. Побудова структури робіт та розбиття їх на менші, більш прості та контрольовані елементи дозволяє краще оцінити час та зусилля, необхідні для виконання кожного елемента. Це є ключовим для планування та керування проектом. Діаграма Ганта - це графічний інструмент, який складається з горизонтальних відрізків, що відображають етапи та стадії проекту, розташованих на осі часу. Початок, кінець та довжина відрізків відповідають початку, завершенню та тривалості кожного етапу. Діаграма Ганта також включає інформацію про

ресурси, використовувані в проекті. Лінії, що з'єднують відрізки, показують послідовність робіт і вказують на те, що наступний етап починається після завершення попереднього.

$$PERT = (Pertopt + Pertpes + 4 * Pertfact) / 6, (3.1)$$

Якщо план проекту складається з великої кількості етапів і фаз робіт, неможливо детально проаналізувати весь план проекту. У таких випадках необхідно проаналізувати ризики завдань, які можуть стати критичними. Для цього використовується метод PERT, який прогнозує песимістичний та оптимістичний час виконання завдань для розрахунку очікуваної тривалості операції. Метод PERT описаний у формулі

$$PERT = (30+35+33*4)/6 = 32.83, (3.2)$$

PERT - очікувана тривалість операцій; PERTopt - оптимістична тривалість операцій; PERTpes - песимістична тривалість операцій; PERTfact - фактична тривалість операцій. Для цього проекту було спрогнозовано песимістичну та оптимістичну тривалість виконання завдань проекту, розрахунки наведено у формулі 3.2. PERT = (30+35+33*4)/6 = 32,83 (3.2) Ці дані приблизно збігаються з часом, необхідним для завершення проекту. Якщо в ході проекту виникають затримки в часі виконання робіт, необхідно вносити зміни до плану проекту.

Управління проектом складається з низки послідовних дій, а саме

- Визначення та формування вимог до проекту;
- Формування найбільш чітких і зрозумілих цілей;
- Налагодження та реалізація комунікації між сторонами, що беруть участь у проекті;
- Комунікація з командою, врахування її потреб, побажань, очікувань та коригування існуючих планів відповідно до отриманих матеріалів.

Загальні етапи побудови проекту залежать від розміру та складності проекту, а також від специфіки галузі та галузі, в якій він реалізується. Кожен проект може мати свої унікальні кроки та вимоги, але описані вище кроки відображають загальний процес створення проекту.

Ресурси проекту - трудові, технічні та/або матеріальні одиниці, що використовуються при виконанні завдань за проектом та забезпечують реалізацію всіх поставлених цілей проекту.

У проекті вказані трудові ресурси:

- керівник проекту;
- SEO-фахівець;
- верстальник;
- контент менеджер;
- програміст;
- тестер.

Трудові ресурси повинні виконувати такі роботи:

- аналіз методів задоволеності клієнтів. (що дозволить визначити методи, які будуть використовуватися під час реалізації проекту);
- аналіз моделей задоволеності клієнтів. (Що дозволить вам визначитися з моделями, які будуть аналізуватись при реалізації проекту);
- розробка власної моделі задоволеності клієнтів. (Після аналізу існуючих методів та моделей буде розроблено нову модель або покращено існуючу модель);

- реалізація створеної чи покращеної моделі задоволеності клієнтів на сайті інтернет-магазину;
- розробка структури сайту на основі створеної чи покращеної моделі задоволеності клієнтів. (Як сайт виглядатиме для користувача, як має виглядати меню сайту, які категорії мають бути на сайті тощо. Цю роботу виконує seo-фахівець);
- розробка макету сайту. (Розробка самого дизайну сайту, а саме верстки сайту. Ця робота виконується на машинці);
- оптимізація швидкості шаблону веб-сайту. (Після створення шаблону інтернет-магазину, а саме візуальної частини, його потрібно оптимізувати, очистити від зайвого коду, збільшити швидкість завантаження сайту, а саме візуальної частини. Цю роботу виконує дизайнер);
- розробка шаблону сайту (Під час створення макету сайту, а саме його візуальної частини, що складається з Html (мова гіпертекстової розмітки), Css (каскадні таблиці стилів), необхідно пов'язати шаблон із самою платформою Opencart, на якій є сайт, що працює мовою програмування Php (PHP: Hypertext Preprocessor), цю роботу виконують програміст та дизайнер);
- збір ключових слів (На цьому етапі проводиться аналіз ключових слів та запитів, які пишуть люди, щоб знайти той чи інший товар чи послугу, які цікавлять клієнтів. Цю роботу проводить SEO-фахівець);
- оптимізація сайту під пошукову систему. у HTML-кодi сторінки.Цю роботу виконують seo-фахівець та контент-менеджер);
- наповнення сайту товаром (Цей етап включає заповнення опису, характеристик, зображень, ключових слів, що відповідають запитам покупця і товару. Робить контент-менеджер);
- наповнення сайту (Наповнення сторінок з описом компанії, контактами, блогом, цікавими статтями тощо);
- оптимізація кошика. (Оптимізація кошика для зручності покупців);

- тестування – етап перевірки програмного продукту, сайту на наявність помилок, багів. Після завершення цього етапу, за відсутності помилок чи багів, проект вважається завершеним та переходить у стадію «Завершення проекту»;
- завершення проекту - передача замовнику опису проекту, протоколів випробувань, звітів про проведені перевірки, підсумкового звіту щодо проекту та проектної документації, списку відкритих питань та підсумкових робіт, вирішення всіх спірних питань. Документування та аналіз досвіду реалізації проекту.

Проект	33 днів?	Ср 02.09.20	Пт 16.10.20	
▫ Ініціалізація	9 днів	Ср 02.09.20	Пн 14.09.20	Менеджер проекту
Аналіз предметної області	3 днів	Ср 02.09.20	Пт 04.09.20	Менеджер проекту
Постановка задачі	2 днів	Пн 07.09.20	Вт 08.09.20	Менеджер проекту
Написання ТЗ	3 днів	Ср 09.09.20	Пт 11.09.20	Менеджер проекту
Узгодження плану проекту	1 день	Пн 14.09.20	Пн 14.09.20	Менеджер проекту
▫ Розробка	21 днів?	Вт 15.09.20	Вт 13.10.20	
Аналіз методів задоволення клієнтів	1 день	Вт 15.09.20	Вт 15.09.20	Менеджер проекту
Аналіз моделей задоволення клієнтів	2 днів	Ср 16.09.20	Чт 17.09.20	Менеджер проекту
Розробка власної моделі задоволення клієнтів	1 день?	Пт 18.09.20	Пт 18.09.20	Менеджер проекту
Впровадження моделі в сайт інтернет магазин	1 день?	Пн 21.09.20	Пн 21.09.20	Менеджер проекту
Розробка структури сайту	2 днів	Вт 22.09.20	Ср 23.09.20	Сео Спеціаліст
Розробка макету сайту	1 день?	Чт 24.09.20	Чт 24.09.20	Верстальник
Оптимізація шаблону на швидкість завантаження	1 день?	Пт 25.09.20	Пт 25.09.20	Верстальник
Розробка шаблону сайту	2 днів	Пн 28.09.20	Вт 29.09.20	Верстальник, Програміст
Збір ключових слів	2 днів	Ср 30.09.20	Чт 01.10.20	Сео Спеціаліст
Оптимізація сайту под пошукові системи	2 днів	Пт 02.10.20	Пн 05.10.20	Сео Спеціаліст, Контент менеджер
Заповнення сайту товарами	2 днів	Вт 06.10.20	Ср 07.10.20	Контент менеджер
Заповнення контенту сайту	3 днів	Чт 08.10.20	Пн 12.10.20	Контент менеджер
Оптимізація кошику оформлення заказу	1 день?	Вт 13.10.20	Вт 13.10.20	Програміст
▫ Тестування	3 днів	Ср 14.10.20	Пт 16.10.20	Тестувальник
Тестування первинної реалізації	1 день	Ср 14.10.20	Ср 14.10.20	Тестувальник
Тестування кінцевої реалізації	2 днів	Чт 15.10.20	Пт 16.10.20	Тестувальник

Рисунок 3.3 - План проекту

На рисунку 3.1 наведено робочий простір програми MS Project у режимі відображення «Діаграма Ганта». Зображена діаграма Ганта яка відповідає всім роботам які повинні бути виконані за планом проекту.

На Діаграма Ганта, вказано час виконання проекту, а саме час початку виконання роботи, час закінчення виконуваної роботи приведення на рисунку 3.2, попередня завдання яка дотримується перед поточного завдання. З правій частині від кожної задачі яка виконується вказано відповідального за виконання даного завдання. Показаний критичний шлях виконання проекту. Критичний шлях – це шлях моделі, тривалість якої дорівнює критичній.

Критична тривалість – це мінімальна тривалість, протягом якої може бути виконаний весь комплекс робіт проекту. Тобто виходячи з цього, критичний шлях є послідовністю робіт з нульовим резервом часу.

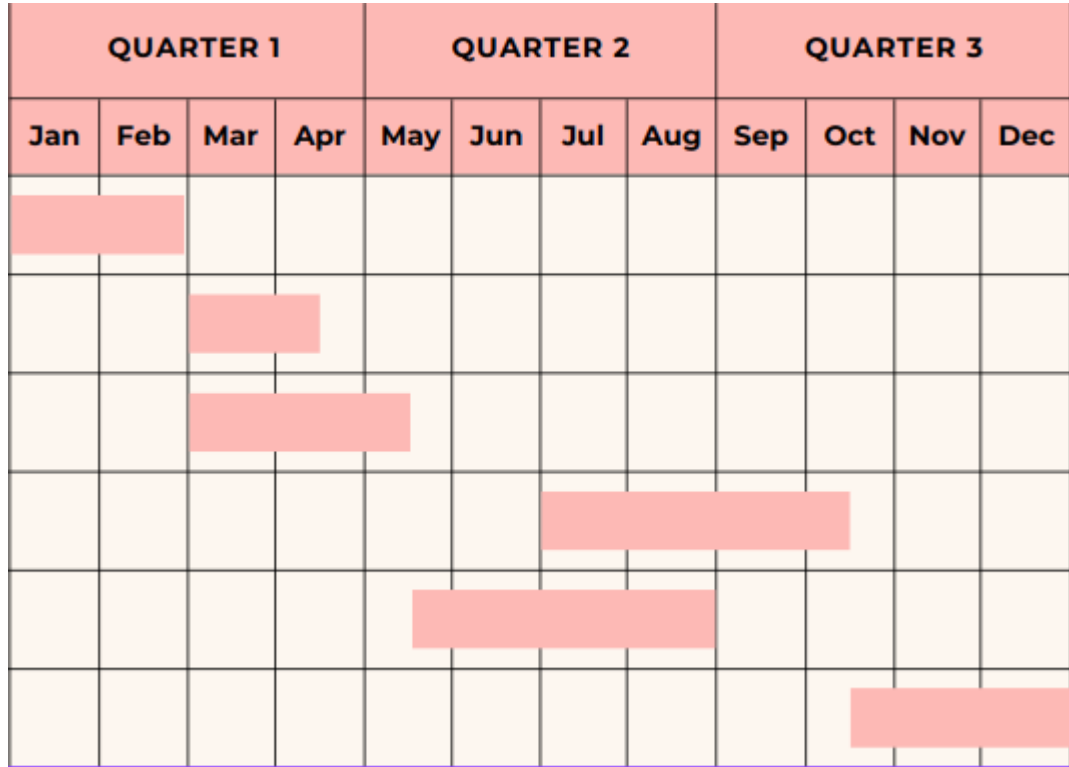


Рисунок 3.4 - Діаграма Ганта час початку і кінця робіт.

PROCESS	QUARTER 1				QUARTER 2				QUARTER 3			
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Planning	■											
Wireframing			■									
Design Process			■									
Front-end development							■					
Back-end development					■							
Deployment										■		

Рисунок 3.5 - Діаграма Ганта проекту, із зазначенням ресурсів

3.2 Результати вирішення задачі оптимізації змісту проекту

У сфері розробки програмного забезпечення існує різноманітність методологій, які можуть представляти собою як нові підходи до старих методів, так і повністю нові підходи. Хоча існують десятки різних підходів до управління проектами, вибір конкретної методології повинен здійснюватися з урахуванням характеру бізнесу та його вимог, оскільки не існує універсальної методології, яка б підходила для будь-якого проекту або організації.

Оптимальний вибір методології управління проектом визначається такими факторами, як характер проекту, розмір, задіяні ресурси та інші.

Для досягнення мети було сформовано вихідні дані для вирішення завдання, яке включало 6 етапів з варіантами виконання робіт на кожному етапі. У таблицях 3.1 та 3.2 подано інформацію про кошти, що виділяються на кожен етап (K_h), кількість альтернативних варіантів виконання робіт на кожному етапі (M_h) та час (t_{hj}), необхідний для реалізації кожного варіанту на кожному етапі.

Таблиця 3.1 — Вихідні дані для розрахунку при використанні методології Scrum

Етап 1	Етап 2	Етап 3	Етап 4	Етап 5	Етап 6
$h=1$	$h=2$	$h=3$	$h=4$	$h=5$	$h=5$
$K1 = 80$ тис. грн	$K2 = 60$ тис. грн	$K3 = 70$ тис. грн	$K4 = 50$ тис. грн	$K5 = 45$ тис. грн	$K6 = 40$ тис. грн
$M1 = 2$	$M2 = 2$	$M3 = 3$	$M4 = 3$	$M5 = 1$	$M6 = 1$
$t11 = 26$ дн.	$t21 = 24$ дн.	$t31 = 22$ дн.	$t41 = 23$ дн.	$t51 = 20$ дн.	$t61 = 14$ дн.
$t12 = 23$	$t22 = 25$	$t32 = 27$	$t42 = 23$		
		$t33 = 26$ дн.	$t43 = 25$ дн.		

Для розрахунку вартості варіанту (Wh_j) призначені трудові та матеріальні ресурси, що необхідні для реалізації певної роботи. Інформація про загальну вартість виконання операцій j -го варіанта на h -му етапі описаного вище проекту представлена для варіантів використання каскадної й гнучкої методологій: Scrum, див. табл. 3.2.

Таблиця 3.2 — Вихідні дані про вартість виконання операцій на h -му етапі для j -го варіанта виконання при використанні методології Scrum

Етап 1	Етап 2	Етап 3	Етап 4	Етап 5	Етап 6
$W11 = 79,5$	$W21 = 58,3$	$W31 = 69,9$	$W41 = 49,6$	$W51 = 44,7$	$W61 = 39,9$
$W12 = 79$	$W22 = 58,3$	$W32 = 67,9$	$W42 = 49,3$		
		$W33 = 69,3$	$W43 = 50$		

Також, отриманні показники якості продукта, див. табл. 3.5 та показники ризиків, див. табл. 3.3, використовуючи Scrum методологію.

Таблиця 3.3 — Значення показників якості продукта при застосуванні методології Scrum

Этап 1	Этап 2	Этап 3	Этап 4	Этап 5	Этап 6
$\Psi_{111} = 0,89$	$\Psi_{211} = 0,89$	$\Psi_{311} = 0,75$	$\Psi_{411} = 0,79$	$\Psi_{511} = 0,89$	$\Psi_{611} = 0,94$
$\Psi_{121} = 0,84$	$\Psi_{221} = 0,86$	$\Psi_{321} = 0,86$	$\Psi_{421} = 0,75$		
		$\Psi_{331} = 0,86$	$\Psi_{431} = 0,91$		

Таблиця 3.4 — Значення показників ризиків при застосуванні методології Scrum

Этап 1	Этап 2	Этап 3	Этап 4	Этап 5	Этап 6
$\Theta_{111} = 0,44$	$\Theta_{211} = 0,44$	$\Theta_{311} = 0,25$	$\Theta_{411} = 0,3$	$\Theta_{511} = 0,44$	$\Theta_{611} = 0,56$
$\Theta_{121} = 0,36$	$\Theta_{221} = 0,39$	$\Theta_{321} = 0,39$	$9 \Theta_{421} = 0,25$		
		$\Theta_{331} = 0,39$	$\Theta_{431} = 0,49$		

Показник ризиків оцінює ризик і-го ризикового події на h-му етапі проекту при реалізації j-го альтернативного варіанту, $i = 1$. Він дорівнює добутку ймовірності появи цієї ризикової події на оцінку негативних наслідків від його настання. Критерієм якості є тип зв'язків між операціями, які містяться в кожній альтернативі. Передбачається, що завдання тільки з послідовним зв'язком забезпечать ідеальну якість, рівне одиниці. В той час, коли будь-яке розгалуження в мережевій моделі веде до втрати якості продукту. Показник якості оцінюється наступним чином:

$$M_w = N_s + (M_w * N_p) / N_s + N_p, (3.3)$$

де ϵ є кількістю послідовних завдань в етапі, ϵ є пенальті за якістю, ϵ є кількістю паралельних завдань в етапі. Покращити якість продукту можна,

якщо операції етапу йдуть послідовно одна за одною. Оскільки, вся увага команди проекту і всі ресурси концентруються на одній операції. Розмір розраховується наступним чином:

$$M_w = N_s / N_s + N_p, (3.4)$$

Розглядається один тип ризиків, а саме, зрив термінів щодо виконання конкретного етапу. Як критерій ризику, в даному випадку, виступає тип зв'язків між операціями. Для даного типу проекту ймовірність зсуву термінів велика, і чим більше завдань розташовано послідовно, тим вище ймовірність того, що збільшення тривалості одного завдання призведе до збільшення тривалості всього етапу. Формули для розрахунку ризиків:

$$O = 1 - N_s + (M_p * N_s) / N_s + N_p, (3.5)$$

Проаналізувавши ринок вебдодатків з грошового обліку, можна спрогнозувати, що отриманий дохід з продажі програми «MoneyCount» складе 750 тис. грн., без урахування витрат на розробку продукту на різних стадіях проекту. Зібрана інформація була застосована для оптимізації змісту проекту, за умови, що для управління ним використовуються методологія Scrum.

ВИСНОВКИ

При розробці кваліфікаційна роботи було проведено аналіз моделей і методів оцінювання ефективності використання SCRUM в управлінні ІТ-проектами. Були визначені основні концепції, принципи та процеси, які використовуються в SCRUM, а також вивчені існуючі дослідження та публікації, що присвячені аналізу ефективності використання методології в даній галузі.

У результаті проведеного дослідження було визначено критерії ефективності використання SCRUM в управлінні ІТ-проектами, такі як терміни виконання проекту, якість продукту, задоволення замовника та команди розробників. Були розроблені методики оцінювання ефективності використання SCRUM в управлінні ІТ-проектами, які дозволяють визначати рівень виконання проекту з використанням SCRUM та здійснювати аналіз ефективності методології в даній галузі.

Для підтвердження ефективності розроблених методик було проведено дослідження на реальному ІТ-проекті. Отримані результати свідчать про те, що використання методології SCRUM в управлінні ІТ-проектами дійсно забезпечує підвищення ефективності проекту, що підтверджується якісними та кількісними показниками.

Отже, можна зробити висновок, що використання методології SCRUM в управлінні ІТ-проектами є ефективним інструментом для забезпечення успішності проекту та підвищення якості продукту. Розроблені методики оцінювання ефективності використання SCRUM в управлінні ІТ-проектів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Про Харківський національний університет радіоелектроніки. URL: <https://nure.ua/universytet/pro-universitet> (дата звернення: 29.04.2022).
2. Статут Харківського національного університету радіоелектроніки (нова редакція), погоджений Конференцією трудового колективу 26 грудня 2017 року. URL: https://nure.ua/wp-content/uploads/Main_Docs_NURE/statut.pdf (дата звернення: 03.05.2022).
3. Кафедра Інформаційних управляючих систем Харківського національного університету радіоелектроніки. URL: <https://nure.ua/ru/department/kafedra-ius> (дата звернення: 02.05.2022).
4. Положення про організацію освітнього процесу в Харківську національну університеті радіоелектроніки (затверджено наказом ХНУРЕ від 27.11.2020 №400). Харків: ХНУРЕ, 2020. URL: https://nure.ua/wp-content/uploads/Main_Docs_NURE/polozhennja-pro-organizaciju-osvitnogo-procesu-v-hnure.pdf (дата звернення: 30.04.2022).
5. Методологія Lean URL: <https://kanbanize.com/lean-management/what-is-lean-management>
6. Методологія Канбан URL: <https://www.diegocalvo.es/en/methodology-kanban-methodology-agil/>
7. Project management methodologies URL: <https://startinfinity.com/project-management-methodologies/scrum>
8. Методолія AGILE URL: <https://stackify.com/agile-methodology/>
9. Best practises AGILE URL: <https://www.altexsoft.com/whitepapers/agile-project-management-best-practices-and-methodologies/>
10. Agile environment URL: <https://www.altexsoft.com/whitepapers/agile-project-management-best-practices-and-methodologies/>
11. Створення графіка Ганта URL: <https://www.canva.com/>

12. Звіт групи Standish Group Chaos URL:
<https://vitalitychicago.com/blog/agile-projects-are-more-successful-traditional-projects/>

13. Алан Д. Опп. Управління проектами. Посібник з ключових процесів, моделей і методів: Підручник – Дніпро: Баланс Бізнес Букс, 2006

14. Який варіант відповідає вашому бізнесу URL:
<https://worksection.com/ua/blog/waterfall-vs-agile.html>

15. Селнес, Ф., 1993. «Дослідження впливу продуктивності продукту на репутацію бренду, задоволеність і лояльність». Європейський журнал маркетингу 27

16. Левикін В.М., Девон В.В. Дослідження задоволеності і лояльності клієнтів в проектах міні-готелів. Науковий журнал «Технологічний аудит та резерви виробництва», 2015, №. 1/5(21). С. 9-21.

17. Методичні вказівки щодо розробки та оформлення магістерської атестаційної роботи за спеціальністю 122 – „Комп'ютерні науки” програма «Управління проектами в галузі інформаційних технологій» освітньокваліфікаційного рівня «магістр» / Упоряд.: Петров К.Е., Левикін В.М., Чалий С.Ф., Євланов М.В., Саєнко В.І., Міхнов Д.К., Міхнова А.В., Чала О.В. – Харків: ХНУРЕ, 2019. – 24 с.

18. Все про методологію SCRUM URL:
<https://www.atlassian.com/ru/agile/scrum>

19. Гібка методологія SCRUM URL:
<https://www.atlassian.com/ru/agile/scrum>

20. SCRUM ефективно управління проектами URL:
<https://brainrain.com.ua/%d1%81%d0%ba%d1%80%d0%b0%d0%bc-%d1%8d%d1%82%d0%be/>

21. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлювання. – Чинний від 22.06.2015. – Київ: ДП «УкрНДНЦ», 2016. – 31

22. ДСТУ 8302:2015. Інформація та документація. Бібліографічні посилання. Загальні положення та правила складання. – Чинний від 04.03.2016. – Київ: ДП «УкрНДНЦ», 2016. – 20 с.