

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Дослідження моделей і методів управління ІТ проєктом з метою
підвищення ефективності планування часу та управління задачами

Виконала: студентка VI курсу, групи СНмд-61
спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)

	(підпис)	Зайченко М.І. (прізвище та ініціали)
Керівник	(підпис)	Гарматій Н.М. (прізвище та ініціали)
Нормоконтроль	(підпис)	Дуда О.М. (прізвище та ініціали)
Завідувач кафедри	(підпис)	Боднарчук І.О. (прізвище та ініціали)
Рецензент	(підпис)	Осухівська Г.М. (прізвище та ініціали)

Тернопіль
2023

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

«27» грудня 2023 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Зайченко Марина Ігорівна
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження моделей і методів управління ІТ проєктом з метою підвищення ефективності планування часу та управління задачами

Керівник роботи к.е.н., доц. Гарматій Н. М.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «24» листопада 2023 року № 4/7-1098

2. Термін подання студентом завершеної роботи грудня 2023р.

3. Вихідні дані до роботи Літературні джерела з тематики роботи

4. Зміст роботи (перелік питань, які потрібно розробити)

ВСТУП. 1 ТЕОРЕТИЧНІ АСПЕКТИ УПРАВЛІННЯ ІТ-ПРОЄКТОМ ТА МЕТОДИ

ПЛАНУВАННЯ ЧАСУ; 1.1 Управління ІТ-проєктом: огляд та концепції; 1.2 Методи

планування часу; 1.3 Інструменти та методології управління в ІТ-проєктах;

2 ДОСЛІДЖЕННЯ, ВИБІР ОПТИМАЛЬНИХ МЕТОДІВ УПРАВЛІННЯ ІТ-ПРОЄКТОМ ТА

РОЗРОБКА МОДЕЛІ УПРАВЛІННЯ ІТ-ПРОЄКТОМ; 2.1 Постановка завдання

дослідження; 2.2 Аналіз ефективності існуючих методів; 2.3 Вибір оптимальних методів для

конкретного ІТ-проєкту; 2.4 Опис розробленої моделі на основі фреймворку Nexus;

2.5 Особливості нової моделі 3. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТА АПРОБАЦІЯ

МОДЕЛІ 3.1 Методологія експерименту; 3.2 Оцінка результатів експерименту; 3.3 Висновки;

4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ; 4.1 Питання щодо

охорони праці; 4.2 Питання щодо безпеки в надзвичайних ситуаціях;

ВИСНОВКИ; СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ; ДОДАТКИ

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Сенчишин В. С., доц.		
Безпека в надзвичайних ситуаціях	Клепчик В.М., ст. викладач		

7. Дата видачі завдання _____ 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи		Виконано
2.	Підбір наукових джерел по темі роботи		Виконано
3.	Переклад та опрацювання наукових джерел по темі роботи		Виконано
4.	Виконання дослідження згідно мети кваліфікаційної роботи		Виконано
5.	Оформлення розділу «Теоретичні аспекти управління ІТ-проєктом та методи планування часу»		Виконано
6.	Оформлення розділу «Дослідження, вибір оптимальних методів управління ІТ-проєктом та розробка моделі управління ІТ-проєктом»		Виконано
7.	Оформлення розділу «Експериментальне дослідження та апробація моделі»		Виконано
8.	Виконання завдання до підрозділу «Охорона праці»		Виконано
9.	Виконання завдання до підрозділу «Безпека в надзвичайних ситуаціях»		Виконано
10.	Оформлення кваліфікаційної роботи		Виконано
11.	Нормоконтроль		Виконано
12.	Перевірка на плагіат		Виконано
13.	Попередній захист кваліфікаційної роботи		Виконано
14.	Захист кваліфікаційної роботи		

Студент

(підпис)

Зайченко М.І.

(прізвище та ініціали)

Керівник роботи

(підпис)

Гарматій Н.М.

(прізвище та ініціали)

АНОТАЦІЯ

Дослідження моделей і методів управління ІТ проєктом з метою підвищення ефективності планування часу та управління задачами// Кваліфікаційна робота освітнього рівня «Магістр» // Зайченко Марина Ігорівна // Тернопільський національний технічний університет імені Івана Пулюя, центр перепідготовки та післядипломної освіти, кафедра комп'ютерних наук, група СНмд-61 // Тернопіль, 2023 // с. – 67 ., рис. – 5 , табл. – 6, , додат. – 1, бібліогр. – 50.

Ключові слова: Scrum, Agile Methodology, Nexus Framework, управління задачами.

В дипломній роботі проведено комплексне дослідження сучасних моделей та методів управління ІТ-проєктами з акцентом на підвищення ефективності планування часу та управління задачами. Робота спрямована на аналіз та оцінку різних підходів у сфері Agile і Scrum, з особливою увагою до фреймворку Nexus, який розглядається як інструмент для оптимізації роботи великих та складних ІТ-проєктів. Оцінюється як методології впливають на продуктивність команди, якість кінцевого продукту, задоволеність клієнтів, а також на вирішення загальних викликів у проєктному менеджменті. Робота включає детальний аналіз впровадження Nexus у реальних умовах, висвітлюючи як переваги, так і потенційні труднощі його застосування. Результати дослідження надають важливий внесок у розробку стратегій для підвищення ефективності управління ІТ-проєктами, а також можуть бути використані для розробки рекомендацій з удосконалення процесів планування і координації у великомасштабних проєктах.

ANNOTATION

Research of IT project management models and methods to increase time planning and task management efficiency // The educational level "Master" qualification work // Zaichenko Maryna Ihorivna // Ternopil Ivan Pulyuy National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, SNmd-61 group // Ternopil, 2023 // p. – 67, fig. – 5, tables – 6 , annexes – 1, ref. – 50.

Key words: Scrum, Agile Methodology, Nexus Framework, task management.

In the thesis, a comprehensive study of modern models and methods for managing IT projects with a focus on improving time planning and task management efficiency has been conducted. The work is aimed at analyzing and evaluating various approaches in the field of Agile and Scrum, with particular attention to the Nexus framework, which is considered as a tool for optimizing the work of large and complex IT projects. The impact of these methodologies on team productivity, the quality of the final product, customer satisfaction, as well as addressing common challenges in project management is assessed. The thesis includes a detailed analysis of the implementation of Nexus in real-world conditions, highlighting both its advantages and potential difficulties in its application. The research results make a significant contribution to the development of strategies for improving IT project management efficiency and can also be used to develop recommendations for enhancing planning and coordination processes in large-scale projects.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

IT – Information Technology

CRM – Customer Relationship Management

ПМ – проєктний менеджер

ПК – персональний комп'ютер

GTD – Getting Things Done

CI/CD – Continuous Integration/Continuous Delivery

DevOps – Development and Operations

ЗМІСТ

ВСТУП.....	8
1 ТЕОРЕТИЧНІ АСПЕКТИ УПРАВЛІННЯ ІТ-ПРОЄКТОМ ТА МЕТОДИ ПЛАНУВАННЯ ЧАСУ.....	11
1.1 ОГЛЯД ТА КОНЦЕПЦІЇ.....	11
1.1.1 Ключові поняття та терміни в управлінні ІТ-проєктами	12
1.1.2 Життєвий цикл ІТ-проєкту	14
1.1.3 Роль та важливість управління проєктом у сфері ІТ.....	15
1.2 МЕТОДИ ПЛАНУВАННЯ ЧАСУ.....	16
1.3 ІНСТРУМЕНТИ ТА МЕТОДОЛОГІЇ УПРАВЛІННЯ ІТ ПРОЄКТАМИ....	22
2 ДОСЛІДЖЕННЯ, ВИБІР ОПТИМАЛЬНИХ МЕТОДІВ УПРАВЛІННЯ ІТ-ПРОЄКТОМ ТА РОЗРОБКА МОДЕЛІ УПРАВЛІННЯ ІТ-ПРОЄКТОМ.....	26
2.1 ПОСТАНОВКА ЗАВДАННЯ ДОСЛІДЖЕННЯ.....	27
2.2 АНАЛІЗ ЕФЕКТИВНОСТІ ІСНУЮЧИХ МЕТОДІВ.....	28
2.3 ВИБІР ОПТИМАЛЬНИХ МЕТОДІВ ДЛЯ КОНКРЕТНОГО ІТ-ПРОЄКТУ	31
2.4 ОПИС РОЗРОБЛЕНОЇ МОДЕЛІ НА ОСНОВІ ФРЕЙМВОРКУ NEXUS....	34
2.5 ОСОБЛИВОСТІ НОВОЇ МОДЕЛІ.....	38
2.5.1 Щоденні зустрічі	39
2.5.2 Product backlog refinement	39
2.5.3 Проведення Sprint Review	40
2.5.4 Проведення Sprint Retrospective	41
2.5.5 Пріоритезація в backlog	42
3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТА АПРОБАЦІЯ МОДЕЛІ.....	44
3.1 МЕТОДОЛОГІЯ ЕКСПЕРИМЕНТУ	44
3.2 ОЦІНКА РЕЗУЛЬТАТІВ ЕКСПЕРИМЕНТУ.....	47
3.2.1 Оцінка якості комунікації.....	47
3.2.2 Оцінка продуктивності	48
3.2.3 Оцінка відсотка завершених спринтів	49
3.2.4 Оцінка рівня задоволеності замовників.....	50
3.2.5 Оцінка якості кінцевого продукту	50

3.3 ВИСНОВКИ.....	52
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	54
4.1 ПИТАННЯ ЩОДО ОХОРОНИ ПРАЦІ.....	54
4.1.1 Ергономіка	54
4.1.2. Освітлення	57
4.1.3. Параметри мікроклімату	58
4.1.4 Електромагнітне і іонізуюче випромінювання.. ..	58
4.1.5 Емоційна психогігієна	59
4.2 ПИТАННЯ ЩОДО БЕЗПЕКИ В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	61
ВИСНОВКИ.....	63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	64
ДОДАТКИ.....	68

ВСТУП

У контексті стрімкого розвитку інформаційних технологій, ІТ-проекти відіграють вирішальну роль у реалізації стратегічних бізнес-ініціатив та інноваційних концепцій. Зростаюча складність цих проектів, в тандемі з посиленням динаміки їх розвитку, накладає високі вимоги до їх управління. Ефективне управління ІТ-проектами передбачає застосування комплексних методів та моделей, що базуються на передових принципах проектного менеджменту.

Ця робота фокусується на вивченні сучасних підходів до управління ІТ-проектами, з особливою увагою до аспектів планування, координації ресурсів, оцінки ризиків та контролю якості. В умовах, де часовий фактор і швидкість реагування на зміни у ринковому середовищі є ключовими, адекватність та гнучкість управлінських рішень може стати детермінантом успіху організації. Також, враховуючи постійні зміни в ІТ-галузі, важливо оцінювати та прогнозувати тенденції в цій галузі, щоб забезпечити тривалу конкурентну перевагу.

Актуальність теми. У зв'язку із складністю проектів, яка постійно зростає та значенням інформаційних технологій у сучасному світі, тема магістерської роботи є актуальною. Швидкий розвиток технологій та збільшення масштабів інформаційних систем вимагають нових підходів до управління ІТ-проектами. Професійне управління цими проектами не лише впливає на ефективність та успіх окремих проектів, але й має широкий вплив на загальну продуктивність організацій. Оптимізація процесів планування та управління задачами може значно знизити ризики, пов'язані з виконанням проектів, та підвищити їхню вартість та якість. В цьому контексті, дослідження ефективних моделей і методів управління ІТ-проектами стає надзвичайно важливим, забезпечуючи актуальність та важливість даної роботи.

Метою даної магістерської роботи є дослідження та аналіз існуючих моделей і методів управління ІТ-проектом, а також розробка нової моделі, спрямованої на підвищення ефективності планування часу та управління задачами в ІТ-проектах.

Для досягнення цієї мети були поставлені наступні завдання:

- провести огляд сучасних теоретичних підходів до управління проєктами, зокрема в галузі ІТ;
- проаналізувати існуючі методи та інструменти планування часу та управління задачами;
- визначити проблеми та недоліки існуючих підходів до управління ІТ-проєктами;
- розробити нову модель управління ІТ-проєктом, яка базується на кращих практиках та сучасних технологіях;
- провести експериментальне дослідження ефективності запропонованої моделі та порівняти її з методами, які застосовувалися до неї.

Об'єкт дослідження – це методи планування часу в ІТ проєктах. В дипломній роботі досліджено і проаналізовано процеси, методи та моделі, які використовуються для управління ІТ проєктами з метою підвищення ефективності планування часу та управління задачами.

Предметом дослідження є моделі і методи управління ІТ проєктом з метою підвищення ефективності планування часу та управління задачами.

Наукова новизна одержаних результатів кваліфікаційної роботи полягає у тому, що була розроблена нова модель управління ІТ проєктами на основі фреймворка Nexus, яка допомагає покращити ефективність планування часу та управління задачами.

Практичне значення одержаних результатів. Магістерська робота, присвячена розробці нової моделі управління ІТ-проєктом, має велику практичну цінність у сучасному світі інформаційних технологій. Нова модель забезпечує більш ефективний розподіл завдань серед великих команд, що допомагає пришвидшити процес розробки та покращити його якість. Застосування моделі в реальних проєктних умовах дає можливість отримати зворотний зв'язок від реальних користувачів, що підвищує її практичну цінність. Ефективне управління проєктом зменшує ризики затримок, зайвого збільшення бюджету та недотримання вимог якості.

Апробація результатів магістерської роботи. Розроблену модель було впроваджено для оптимізації роботи команди, яка складається з 50 працівників і займається розробкою веб-додатку. На початку для учасників команди був проведений тренінг, де було розглянуто основи нової моделі, її принципи та відмінності від традиційного Scrum. Наступним етапом було здійснення реорганізацію роботи команди згідно нової моделі: розподіл завдань, встановлення нових ролей, зміна системи комунікації тощо. Зібрані дані з моніторингу та зворотного зв'язку були проаналізовані з метою визначення ефективності нової моделі, її переваг та недоліків порівняно з традиційними методами управління.

Результати кваліфікаційної роботи опубліковані у тезах студентських наукових конференцій «Інформаційні моделі системи та технології – 2023» та «Актуальні задачі сучасних технологій – 2023», які проводились у ТНТУ.

Структура й обсяг кваліфікаційної роботи. Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку літератури з 50 найменувань та 1 додатків. Загальний обсяг кваліфікаційної роботи складає 67 с.

1 ТЕОРЕТИЧНІ АСПЕКТИ УПРАВЛІННЯ ІТ-ПРОЄКТОМ ТА МЕТОДИ ПЛАНУВАННЯ ЧАСУ

Теоретичні аспекти управління ІТ-проєктами та методологія планування часу є невід'ємними складовими успіху проєктів у швидкоплинному технологічному середовищі. Комплексне управління ІТ-проєктами охоплює широкий спектр завдань, що вимагають інтеграції технічних знань та управлінських навичок. Основною метою є забезпечення відповідності проєкту визначеним цілям, умовам, часовим обмеженням та бюджетним рамкам, при цьому враховуючи потенційні ризики та непередбачувані обставини.

Стратегічне планування часу відіграє ключову роль у процесі управління проєктом. Це передбачає глибоке розуміння обсягу роботи, необхідних ресурсів та точної оцінки часу, потрібного для кожного етапу проєкту. Ефективне управління часом дозволяє менеджерам проєктів антиципувати потенційні виклики та своєчасно вживати заходів для їх вирішення, знижуючи таким чином ризики затримок і перевитрат [1].

Науковий підхід до управління ІТ-проєктами включає використання різноманітних методологій та інструментів, які сприяють організації, плануванню, моніторингу та контролю проєктних процесів. Серед них можна виділити такі методології як Agile, Scrum, Waterfall та інші, кожна з яких має свої унікальні характеристики та підходить для певних типів проєктів, залежно від їх специфіки та вимог.

1.1 Огляд та концепції

У сучасному світі, де інформаційні технології займають вирішальну роль в багатьох аспектах життя, управління ІТ-проєктами стає все важливіше. Завдяки розробці та реалізації ІТ-проєктів можлива інноваційна трансформація бізнесу, державного управління та соціальних процесів. Центральним елементом успіху таких проєктів є ефективне планування та управління ресурсами, серед яких час є одним з найцінніших.

Важливість грамотного планування часу в ІТ-проектах важко переоцінити. Невірне чи неефективне розподілення часових ресурсів може призвести до затримки в реалізації проекту, збільшення вартості та, як результат, до втрати конкурентних переваг [2, 3].

1.1.1 Ключові поняття та терміни в управлінні ІТ-проектами

У цьому розділі розглядаються ключові поняття та визначення, які є фундаментальними для розуміння та використання теоретичних аспектів управління ІТ-проектом. Детальніше розглянемо кожне з цих понять:

1. ІТ-проект (інформаційно-технологічний проект) - це проект, який створений для розробки, впровадження, підтримки або покращення інформаційних технологій чи програмного забезпечення. Такий проект спрямований на досягнення конкретних цілей в галузі інформаційних технологій, таких як створення нового програмного продукту, модернізація існуючого програмного забезпечення, впровадження нових технологій, створення або покращення інформаційних систем, а також розв'язання інших завдань, пов'язаних з інформаційними технологіями.

Основні характеристики ІТ-проекту:

- обмеженість в часі і ресурсах;
- унікальність;
- цільове спрямування;
- командна робота.

Прикладами ІТ-проектів можуть бути розробка нового веб-додатку, міграція на покращену версію операційної системи, впровадження системи управління відносинами з клієнтами (CRM) у компанії, створення кібербезпечної інфраструктури та інші проекти, пов'язані з інформаційними технологіями.

2. Управління ІТ-проектом – це систематичний підхід до планування, виконання та контролю ІТ-проекта з метою досягнення його цілей та завдань. Воно включає в себе організацію ресурсів, здійснення комунікації, ризик-

менеджмент, контроль процесів і відстеження прогресу для забезпечення успішного завершення проєкту.

3. Замовник проєкту є особою або організацією, яка ініціює та фінансує ІТ-проєкт. Вони визначають основні вимоги, мету та очікувані результати проєкту. Замовник взаємодіє з проєктним менеджером (ПМ) і командою проєкту для забезпечення відповідності проєкту їхнім потребам.

4. Проєктний менеджер – він є відповідальним за керування усіма аспектами ІТ-проєкта. ПМ може розробляти план проєкту, розподіляти ресурси, керувати командою проєкту та відстежувати прогрес досягнення цілей проєкту. ПМ грає ключову роль у забезпеченні вчасного та успішного виконання проєкту.

5. Етапи проєкту зазвичай поділяються на етапи або фази щоб краще організувати та контролювати виконання завдань. Кожен етап може мати свої унікальні цілі та завдання, і їх послідовне виконання сприяє досягненню загальних цілей проєкту [4, 5].

6. Складність проєкту визначається різними факторами, включаючи технічну складність інформаційних технологій, обсяг робіт, ризики, залежності між завданнями і багато інших аспектів, які впливають на проєкт. Розуміння складності є ключовим для ефективного управління проєктом.

7. Інструменти управління проєктом включають програмне забезпечення та апаратне забезпечення, які використовуються для планування, моніторингу та керування ІТ-проєктом. Ці інструменти допомагають ПМ та команді проєкту краще виконувати їхні обов'язки та досягати успішних результатів.

Успішне управління ІТ-проєктами включає в себе не лише технічні аспекти, такі як розробка програмного забезпечення та тестування або інфраструктурні оновлення, а й стратегічне планування, адекватний ризик-менеджмент, ефективне спілкування та розуміння потреб замовника.

Важливо використовувати сучасні інструменти управління проєктами, які спрощують процес моніторингу, координації роботи команди та забезпечують вчасне і якісне виконання завдань. Глибоке розуміння та ефективне застосування цих понять і термінів є ключовими для досягнення успіху в управлінні ІТ-проєктами.

1.1.2 Життєвий цикл ІТ-проєкту

Життєвий цикл ІТ-проєкту – це послідовність етапів та процесів, які ІТ-проєкт проходить від початку до завершення. Цей цикл розглядається як загальна схема для планування, виконання, контролю та оцінки проєкту, і включає такі основні етапи:

1. Ініціація. Етап ініціації є одним із перших та найважливіших етапів у життєвому циклі ІТ-проєкту. На цьому етапі визначаються основні параметри та рамки проєкту і приймається рішення про його реалізацію або відхилення. Нижче наведено основні етапи та завдання етапу ініціації ІТ-проєкту:

- визначення бізнес-потреб;
- аналіз потреб та ресурсів;
- визначення обсягу проєкту;
- оцінка фінансової доцільності;
- розробка проєктної документації;
- прийняття рішення та створення проєкту.

2. Планування. Розробка детального плану проєкту та обсягу робіт, включаючи вибір методології управління, складання дорожньої карти, розкладу та цілей. На цьому етапі критично важливим є встановлення чітких завдань, розподіл ресурсів та визначення механізмів контролю.

3. Виконання. Реалізація плану проєкту через виконання запланованих завдань. Включає виділення ресурсів, проведення регулярних нарад зі стану проєкту та активну роботу ключових робочих груп. Ефективне спілкування та координація між учасниками проєкту є важливими для успішного виконання цього етапу.

4. Відстеження та керування. Постійний нагляд за виконанням робіт, включаючи моніторинг за допомогою інструментів як діаграма згорання завдань, порівняння показників виконання з ключовими цілями проєкту. Важливим є також вчасне виявлення та усунення можливих проблем і відхилень від плану.

5. Завершення. Аналіз та оцінка ефективності виконання проєкту та роботи команд, що брали в ньому участь. Завершення проєкту включає збір

зворотного зв'язку, формування висновків та рекомендацій для оптимізації майбутніх проєктів, а також документування отриманого досвіду.

Кожен етап має свої власні завдання, вимоги та результати, і вони пов'язані з попередніми та наступними етапами для забезпечення ефективного управління ІТ-проєктом від початку до кінця [6, 7].

1.1.3 Роль та важливість управління проєктом у сфері ІТ

У сучасних реаліях, коли технології проникають у кожен аспект нашого життя, ІТ-проєкти стають дедалі складнішими та багатогранними. Управління такими проєктами вимагає особливого підходу, який міг би вберегти від непередбачуваних ризиків та гарантував якість кінцевого продукту.

Управління проєктами в ІТ-сфері є критично важливим через ряд факторів. Перш за все, кожен ІТ-проєкт має свої обмеження у ресурсах, включаючи фінанси, час та людські ресурси. Ефективне управління проєктами дозволяє правильно розподілити завдання, забезпечуючи оптимальний ритм роботи для команди, що сприяє підвищенню ефективності проєкту. Додатково, сучасні ІТ-проєкти часто включають комплексні та багатокomпонентні завдання, які можуть включати розробку програмного забезпечення, інтеграцію з існуючими системами, розгортання інфраструктури та інше. Управління проєктами в такому контексті дозволяє систематизувати ці складні компоненти, забезпечуючи їх ефективну взаємодію.

Крім того, ІТ-сфера характеризується високою динамікою, де вимоги можуть швидко змінюватися у відповідь на нові технологічні тренди чи зміну комерційних потреб замовника. Гнучке управління проєктами в таких умовах надає команді можливість швидко адаптуватися до цих змін, що допомагає мінімізувати потенційні затримки та зайві витрати. Отже, управління проєктами в ІТ є ключовим для забезпечення успіху та відповідності проєкту змінюваним потребам і обмеженням.

Основні компоненти управління проєктами в ІТ

1. Планування. Детальне планування - це підготовчий етап, на якому визначаються завдання, встановлюються пріоритети, виділяються ресурси та

розробляється загальний план дій. Це може включати розробку різних схем, визначення критичного шляху та розподіл ролей в команді.

2. Моніторинг та контроль. Слідкування за виконанням плану та коригування його відповідно до змінних обставин. Це включає регулярний звіт про стан проєкту, виявлення можливих проблем та їх своєчасне вирішення.

3. Комунікація. Інформаційний обмін між учасниками проєкту, замовниками, стейкхолдерами. Ефективна комунікація включає регулярні зустрічі команди, відгуки, зворотній зв'язок та забезпечення відкритості комунікаційних каналів.

4. Управління ризиками. Проактивне виявлення, аналіз та мінімізація можливих ризиків, що можуть виникнути під час реалізації проєкту. Це також включає розробку запасних планів та стратегій для реагування на непередбачувані обставини.

У світі, де технології постійно змінюються, управління ІТ-проєктами не є просто рекомендацією – це необхідність. Воно стає мостом, який з'єднує технічну виконавчу команду з бізнес-цілями компанії, гарантуючи, що проєкти виконуються вчасно, з максимальною якістю та в рамках бюджету.

1.2 Методи планування часу

Планування часу є важливим компонентом успішного управління проєктами, а також ефективної особистої роботи. Існує безліч методів планування часу, далі будуть розглянуті деякі найпоширеніші з них.

Метод матриці Ейзенхауера. Матриця Ейзенхауера – це гарний інструмент для короткотермінового планування, що охоплює день або кілька днів. Ця методика є відмінним засобом систематизації завдань, які людина намагається виконати протягом обраного часового періоду. Зазвичай створюється велика кількість завдань, яка часто перевищує можливість їх виконання.

Така ситуація призводить до накопичення завдань, які залишаються без уваги. Це може стати перешкодою у досягненні основних цілей. Також існує ризик витрати великої частини енергії на завдання, які насправді не є пріоритетними.

Матриця Ейзенхауера допомагає виявити головні пріоритети, а також відфільтрувати завдання, які не потребують негайного виконання. Вона містить чотири сектори - кожен відображає певну категорію завдань. Основні критерії для їх розподілу - це важливість та терміновість: чи є завдання важливим, чи є воно терміновим. На рисунку 1.1 схематично показано таку матрицю.



Рисунок 1.1 – Матриця Ейзенхауєра

Метод Помодоро був розроблений Франческо Чірілло в кінці 1980-х років як спосіб покращення ефективності роботи і фокусування. Назва «Pomodoro» походить від італійського слова «помідор», оскільки Франческо використовував таймер у формі помідора для відліку часу.

Метод Помодоро передбачає поділ часу на інтервали, які зазвичай тривають 25 хвилин, і відомі як "помідори". Після кожного помідора передбачено коротку перерву, тривалість якої становить 5 хвилин. Після чотирьох помідорів слід зробити тривалу перерву, яка зазвичай триває 15-30 хвилин. Під час кожного помідора основна мета – повністю концентруватися на

виконанні завдання, уникати відволікань та перерв, а також зробити коротку паузу після кожного помідора для відновлення ефективності.

Метод Помодоро має наступні переваги:

- збільшення продуктивності – під час кожного помідора виконується інтенсивна робота без відволікань, що допомагає підвищити продуктивність;
- зниження стресу – регулярні перерви дозволяють відпочити;
- покращення концентрації – фокус на виконанні завдання під час кожного помідора допомагає підвищити концентрацію та уникнути розпилення уваги;
- планування і контроль – метод допомагає планувати робочий час і контролювати прогрес у виконанні завдань.

Важливо враховувати, що метод Помодоро може бути адаптованим до індивідуальних потреб та робочого графіку. Ця стратегія дозволяє більш ефективно розподілити свій час та досягати кращих результатів у виконанні завдань та досліджень.

Метод GTD (Getting Things Done). Основні принципи методу GTD, розробленого Девідом Алленом, є фундаментальними у побудові ефективної системи управління особистою продуктивністю. Метод GTD заснований на ідеї, що розум людини є кращим для генерування ідей, але не для їх зберігання. Тому основною метою GTD є звільнення розуму від навантаження зберігання інформації та перетворення цього процесу в систематизовану структуру [8].

Першим принципом GTD є фіксація усіх завдань та ідей поза межами голови, що дозволяє розуму зосередитися на поточній задачі, замість того, щоб перейматися забутими обов'язками. Другий принцип полягає у регулярному перегляді зібраної інформації, що забезпечує актуальність задач та планів. Третім важливим елементом методу є розподіл завдань за категоріями, що спрощує організацію роботи та прийняття рішень щодо пріоритетів. Четвертий принцип - це систематичне визначення наступних кроків для кожної задачі, що сприяє конкретизації дій та уникненню застою. Нарешті, п'ятий принцип GTD закликає до регулярного самоаналізу та адаптації системи управління задачами,

що забезпечує її ефективність та відповідність особистим потребам та обставинам.

В цілому, метод GTD є комплексним підходом до управління завданнями та часом, що спрямований на підвищення продуктивності та ефективності в робочому процесі. Його застосування може бути корисним у різних сферах діяльності та професійних сферах, включаючи IT, бізнес, освіту та інші [9].

Часові блоки. Метод часових блоків, відомий також як блокування часу, є ефективним інструментом управління часом, який широко використовується в різних професійних контекстах, зокрема у сфері комп'ютерних наук, для оптимізації продуктивності та ефективності робочого процесу. Цей метод полягає у розподілі робочого дня на фіксовані інтервали часу, кожен з яких присвячується конкретному завданню або набору завдань.

Основна ідея методу часових блоків полягає в призначенні специфічного часового проміжку для кожного завдання, що дозволяє зосередитися на одній справі, мінімізуючи перерви та відволікання. Це сприяє глибокій зосередженості та ефективності роботи, оскільки мозок краще фокусується, коли знає, що час для виконання завдання обмежений.

Метод також передбачає стратегічне планування дня. Наприклад, завдання, що вимагають високого рівня концентрації та креативного мислення, краще розміщувати в першій половині дня, коли рівень енергії та зосередженості зазвичай вищий. Водночас, більш рутинні завдання можуть бути заплановані на другу половину дня. Окрім того, метод часових блоків допомагає уникнути перенавантаження завданнями. Плануючи часові блоки, можна реалістично оцінити, скільки часу буде потрібно на виконання кожного завдання, що забезпечує більшу продуктивність і менше стресу.

Важливим аспектом методу часових блоків є гнучкість у його застосуванні. Залежно від специфіки завдань та індивідуальних уподобань, часові блоки можуть бути коротшими чи довгими, дозволяючи адаптувати метод під особливості кожного конкретного робочого процесу. Цей метод є особливо корисним в сфері, де висока концентрація та ефективне управління часом є ключовими факторами успіху.

Метод 2 хвилини. Метод "2 хвилини", який є ключовою складовою системи GTD, розробленої Девідом Алленом, відіграє важливу роль у ефективному управлінні часом та завданнями. Цей метод полягає у визначенні та негайному виконанні завдань, які можуть бути завершені протягом двох хвилин або менше. Застосування цього підходу значно підвищує ефективність робочого процесу, зменшуючи накопичення незначних завдань та сприяючи кращому використанню часу.

Основна ідея методу полягає у тому, що якщо виконання завдання займає менше часу, ніж його планування або делегування (умовно до 2 хвилин), то воно повинно бути виконане негайно. Це дозволяє зменшити обсяги незавершених завдань та звільняє час для більш складних та часовитратних задач.

Метод "2 хвилини" використовується як частина більш широкої стратегії управління завданнями, що передбачає оцінку всіх завдань та визначення найбільш ефективного способу їх виконання. Це сприяє оптимізації робочого процесу та підвищенню продуктивності. Крім того, метод "2 хвилини" допомагає розвивати здатність швидко приймати рішення та діяти, що є важливою компетенцією у сучасному динамічному робочому середовищі.

Список завдань «To-Do». Метод є однією з найпопулярніших систем управління завданнями та часом. Він полягає у створенні детального переліку завдань, які потрібно виконати протягом дня, тижня або навіть місяця. Основна ідея цього методу полягає в тому, щоб вивільнити свій розум від потреби постійно пам'ятати про всі завдання, передаючи цю роль на папір або електронний пристрій. Коли завдання записане, можна фокусуватися на його виконанні, а не на його пригадуванні.

Список завдань «To-Do» також сприяє розподіленню завдань за пріоритетами. Виконуючи завдання по черзі зі списку, можна зосередитися на найважливіших і першочергових задачах. Це допомагає ефективно використовувати час і ресурси. Однак важливо підходити до створення списку завдань розумно. Довгий і перевантажений список може викликати відчуття пригніченості і стресу. Тому важливо періодично переглядати та коригувати список, вилучаючи непотрібні завдання та додаючи нові [10].

Метод 1-3-5. Це система планування завдань, що допомагає розподілити свій день на пріоритетні задачі так, щоб не перевантажувати себе і залишити можливість для гнучкості. Суть методу полягає у наступному: кожного дня потрібно обрати одне найважливіше завдання, три завдання середньої важливості та п'ять менших завдань. Це допомагає конкретизувати та структурувати цілі, дозволяючи зосередитися на тому, що дійсно важливо, і не розпилювати увагу на численних менших задачах. Використовуючи метод 1-3-5, можна бути впевненим, що щодня робота йде над найбільш важливими завданнями, при цьому не забуваючи про менші задачі. Він забезпечує баланс між потребою виконувати ключові завдання та можливістю адаптації до змінних обставин.

Зокрема, метод 1-3-5 є дуже корисним для людей, які часто відчують відчуття пригніченості від великої кількості завдань. Він допомагає створити чіткий план дій на день, забезпечуючи почуття досягнення і продуктивності на закінчення дня.

Метод «Eat the Frog». Він базується на концепції, що найкращий спосіб подолати відстрочення і зробити свій день продуктивним – це почати з найважчого завдання. Ідея цього методу полягає у тому, що, починаючи свій день з виконання найбільш неприємного або складного завдання, людина відчуває велике полегшення і мотивацію до виконання інших завдань.

Назва методу взята з вислову Марка Твена, який говорив, що якщо перш за все вранці ви з'їсте жабу, то це буде найгірше, що вас може трапитися протягом цього дня. Тобто жаба символізує важливе і найменш бажане завдання. Використовуючи метод «Eat the Frog», людина зосереджується на головному пріоритеті, не дозволяючи іншим факторам, які можуть відволікати, завадити їй. Це також сприяє формуванню дисципліни, оскільки змушує індивіда регулярно виконувати завдання, які він може відкладати.

Загалом, метод «Eat the Frog» є ефективним засобом подолання прокрастинації, збільшення продуктивності та досягнення бажаних результатів шляхом виконання найважливіших завдань на початку дня.

Ці методи можна адаптувати та комбінувати залежно від особистих потреб та особливостей роботи. Важливо пам'ятати, що головна мета планування часу -

це не лише ефективне використання часу, але й збільшення якості виконуваної роботи та загального комфорту життя. Управління задачами є ключовим компонентом успішного управління ІТ-проектами. Для забезпечення ефективності процесів та досягнення поставлених цілей використовуються різноманітні інструменти та методи.

1.3 Інструменти та методології управління ІТ проектами

Управління задачами в ІТ-проектах вимагає використання специфічних інструментів та методів, які дозволяють ефективно організувати робочий процес, відстежувати прогрес та сприяти співпраці всередині команди. До них належать системи управління проектами, інструменти для спілкування, ведення документації, версіонування коду тощо.

До систем управління проектами належать такі програми, як JIRA, Trello, Asana, та Microsoft Project, які допомагають відстежувати прогрес проекту, визначати відхилення, та спілкуватися з командою.

Серед інструментів для спілкування найпопулярніші такі месенджери, як Slack, Microsoft Teams, та Discord. Вони забезпечують миттєвий обмін повідомленнями між учасниками команди, що забезпечує швидке рішення питань. Серед інструментів для документування можна виділити Confluence, Google Docs, та Notion, вони дозволяють створювати, зберігати та спільно користуватися документацією проекту. Найбільш популярна система керування версіями – це Git, але окрім неї також інколи використовується Mercurial та інші системи контролю версій. Вони сприяють управлінню кодом, слідкуванню за змінами та співпраці програмістів [11].

Вибір методології проекту є критично важливим кроком у керуванні проектами, оскільки він значною мірою впливає на успіх проекту. Методологія задає рамки, в яких буде здійснюватися планування, виконання, моніторинг та контроль над проектом, і тим самим визначає як ресурси будуть виділятися та використовуватися, як будуть управлятися ризики, а також як буде вимірюватися прогрес.

Ефективний вибір методології може поліпшити комунікацію між учасниками проєкту, забезпечити більшу прозорість у процесах та допомогти управляти очікуваннями зацікавлених сторін [12]. Методології управління проєктами можна класифікувати за різними характеристиками, але головними є традиційні та гнучкі методології.

Методологія Waterfall (каскадна модель) – це традиційна модель, де етапи проєкту сліднують один за одним у строгому порядку. Схематичне зображення цієї методології наведено на рисунку 1.2.

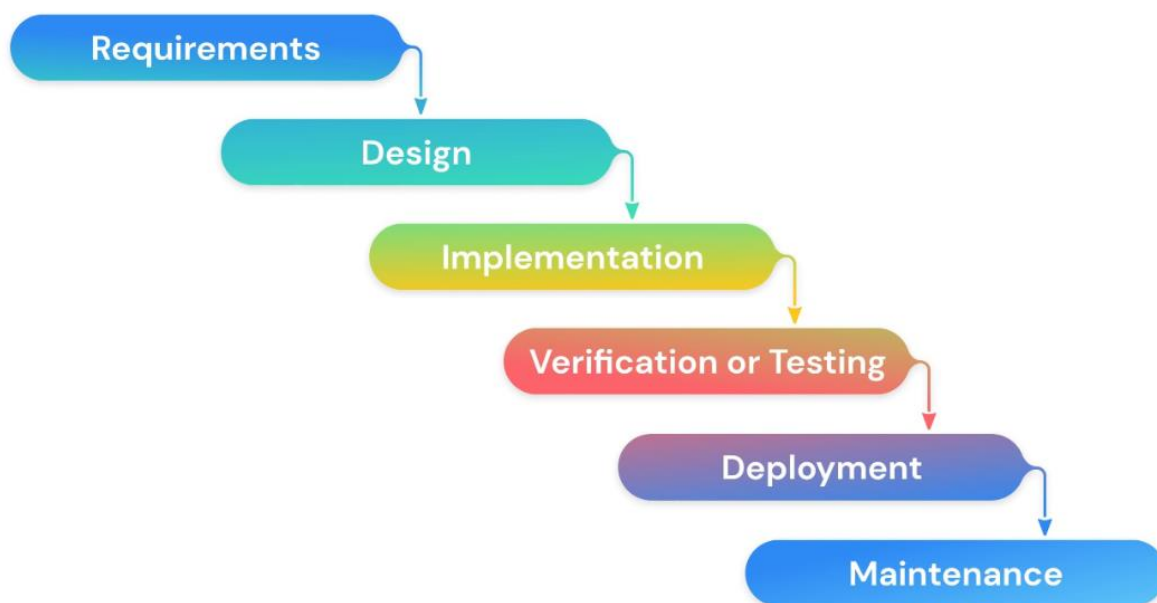


Рисунок 1.2 – Схематичне зображення методології Waterfall

Методологія Waterfall, що також відома як каскадна модель, являє собою одну з найбільш ранніх парадигм управління проєктами в області розробки програмного забезпечення, що була сформульована та широко поширена в комп'ютерних науках в середині XX століття. Ця методологія характеризується строго секвенційним процесом розвитку, де кожен етап проєкту починається лише після повного завершення попереднього, формуючи лінійно-послідовну структуру.

Waterfall відзначається своєю передбачуваністю та структурованістю, що робить її популярною в середовищах, де зміни вимог є мінімальними чи недопустимими, та де важливим є ретельне планування та документування на

ранніх стадіях розробки. Такий підхід забезпечує чітке розуміння процесу та кінцевого продукту ще на початкових етапах, а також полегшує процес контролю та управління проектом.

Проте, незважаючи на свою структурованість, методологія Waterfall має ряд обмежень, особливо в контексті сучасних динамічних та адаптивних умов розвитку ІТ-проектів. Найважливішим обмеженням є негнучкість моделі у відповіді на зміни вимог чи обставин, що може призводити до значних витрат часу та ресурсів у випадку необхідності внесення змін на пізніших етапах розробки.

Незважаючи на це, досвід використання Waterfall продовжує бути важливим для розуміння основ процесів управління проектами в комп'ютерних науках, а також надає стабільний фундамент для розробки та вдосконалення більш гнучких та адаптивних методологій, таких як Agile. Схематично різницю підходів цих методологій зображено на рисунку 1.3

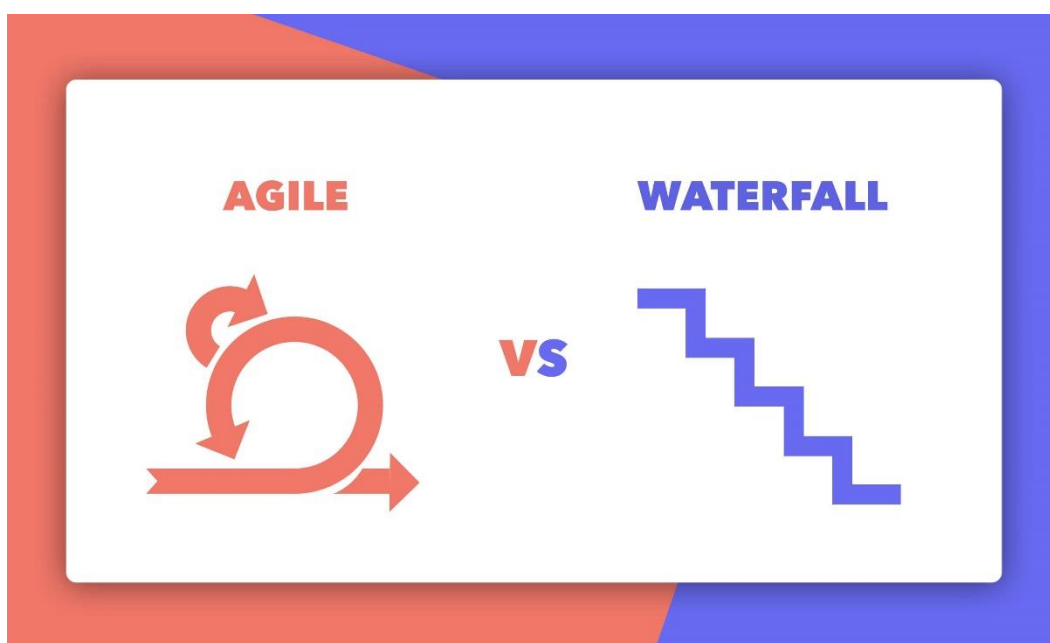


Рисунок 1.3 – Схематична різниця підходів методологій

Agile методологія, що виникла на початку 21-го століття у відповідь на обмеження традиційних підходів, таких як Waterfall, представляє собою гнучкий підхід до управління розробкою програмного забезпечення [13]. Вона базується на ітеративному, інкрементному процесі розвитку, де проект розвивається через

короткі цикли, відомі як спринти. Цей підхід акцентує увагу на ранній та безперервній доставці функціонального програмного забезпечення, адаптивності до змінних вимог, та активному включенню замовника у процес розвитку.

Основною перевагою Agile є його здатність швидко адаптуватися до змін у проєкті та середовищі, що дуже важливо у сучасному швидкоплинному світі технологій. Цей підхід дозволяє проєктним командам зосереджуватись на неперервному поліпшенні продукту та процесу, що, у свою чергу, сприяє збільшенню продуктивності та ефективності.

Scrum являє собою одну з найбільш популярних методологій управління проєктами в сфері розробки програмного забезпечення, що базується на принципах Agile. Ця методологія орієнтована на впровадження гнучкості, адаптивності та колаборативності в процес розробки.

В основі Scrum лежить ідея самоорганізації та взаємодії в команді, де кожен член вносить свій вклад у спільне рішення завдань. Процес розвитку поділяється на фіксовані ітерації, відомі як спринти, кожен з яких триває від одного до чотирьох тижнів. Це дозволяє команді регулярно аналізувати результати роботи, вчасно ідентифікувати проблеми та вносити необхідні корективи.

Управління задачами в IT-проєктах вимагає комбінації відповідних інструментів та методів, які найкращим чином відповідають конкретним потребам проєкту та команди. Важливо регулярно аналізувати результативність використовуваних підходів та адаптуватися до змін, щоб оптимізувати процеси та досягати поставлених цілей.

2 ДОСЛІДЖЕННЯ, ВИБІР ОПТИМАЛЬНИХ МЕТОДІВ УПРАВЛІННЯ ІТ-ПРОЄКТОМ ТА РОЗРОБКА МОДЕЛІ УПРАВЛІННЯ ІТ-ПРОЄКТОМ

Дослідження, вибір оптимальних методів управління ІТ-проєктами та розробка моделі управління ІТ-проєктом представляють собою складну й багатогранну сферу, яка є вирішальною для успішності проєктів у галузі інформаційних технологій. В цьому контексті, ключове значення має розуміння різних методологій управління проєктами та їх придатності до конкретних умов ІТ-проєктів.

Управління ІТ-проєктами охоплює комплексні процеси, такі як планування, організація, мотивація та контроль ресурсів для досягнення специфічних цілей проєкту. Важливою складовою є вибір методології управління, яка повинна відповідати унікальним особливостям кожного проєкту, включаючи його обсяг, складність, командний склад та специфіку виконуваних задач [14].

Методології, такі як Agile, Scrum та Waterfall, пропонують різні підходи до управління проєктами. Agile-методологія, наприклад, акцентує на гнучкості, ітеративному розвитку та здатності швидко адаптуватися до змін. Scrum, будучи однією з рамок Agile, орієнтований на спрощення управління складними проєктами через регулярні зустрічі команди та ітеративну доставку продукту. З іншого боку, Waterfall-модель представляє собою традиційний підхід, заснований на послідовному виконанні етапів проєкту з чітко визначеними вимогами та результатами.

Розробка моделі управління ІТ-проєктом вимагає не лише вибору підходящої методології, але й її інтеграції з конкретними потребами проєкту та організаційним контекстом. Це включає адаптацію процесів, інструментів, технік та стратегій управління для забезпечення максимальної ефективності та відповідності до особливостей проєкту. Таким чином, успіх ІТ-проєкту значною мірою залежить від глибокого розуміння й вмілого застосування принципів управління проєктами [15].

2.1 Постановка завдання дослідження

В контексті стрімкого розвитку інформаційних технологій, управління IT-проєктами вимагає особливого підходу, який би враховував специфіку та динаміку цієї сфери. Сьогодні, коли ринок і технології розвиваються з безпрецедентною швидкістю, актуальним стає питання вибору оптимальних методів управління IT-проєктами, здатних гарантувати ефективність, гнучкість та відповідність сучасним бізнес-вимогам.

Основною метою даного дослідження є глибокий аналіз впровадження нової моделі управління проєктами на основі Nexus. Необхідно визначити ключові тези та розробити стратегічний план дій, який спрямований на оптимізацію робочих процесів, покращення якості продукту та ефективності командної роботи.

Завдання дослідження:

- аналіз існуючих підходів;
- формулювання гіпотез щодо покращення робочих процесів в команді;
- вибір методів для проєкту;
- оцінка впровадження нової моделі на основі Nexus.

В результаті проведеного дослідження передбачається отримання аналізу ефективності моделі Nexus та її впливу на управління проєктами. Дослідження має на меті визначити ключові переваги та потенційні виклики, пов'язані з впровадженням цієї моделі.

Очікується, що результати дослідження нададуть цінні висновки та теоретичні узагальнення, які сприятимуть поглибленому розумінню впливу моделі на основі Nexus на процеси управління проєктами. Ці знання можуть бути використані для підвищення загальної ефективності в галузі проєктного менеджменту, допомагаючи у вдосконаленні стратегій та методологій управління.

2.2 Аналіз ефективності існуючих методів

У рамках сучасного управління IT-проектами, методологія Waterfall, відома своєю лінійною та послідовною структурою, заслуговує детального розгляду. Цей традиційний підхід в управлінні проектами, який передбачає суворе дотримання попередньо визначених етапів, включає послідовне планування, дизайн, реалізацію, тестування та впровадження.

Однією з ключових переваг методології Waterfall є її здатність до ретельного планування на ранніх етапах проекту. Це планування забезпечує високий рівень передбачуваності, що є важливим для проектів з чітко визначеними цілями та вимогами. Наприклад, у випадку розробки стандартного програмного забезпечення для банківської системи, де вимоги та функціональність заздалегідь відомі і малоймовірно зміняться, Waterfall може бути надзвичайно ефективним.

Однак, головним недоліком Waterfall є обмежена гнучкість у внесенні змін після початку проекту. Якщо виникає потреба адаптувати або модифікувати проект, це може призвести до значних затримок і додаткових витрат. Це можна проілюструвати на прикладі розробки програмного забезпечення для стартапу, де вимоги можуть швидко змінюватися: у таких умовах Waterfall може бути неефективним через свою негнучкість.

Waterfall відрізняється високою зосередженістю на документації. Це забезпечує чітке розуміння проектних цілей та вимог на кожному етапі, але водночас може призвести до збільшення часу та витрат на підготовку та підтримку документів. Для великих корпоративних проектів з жорсткими регуляторними вимогами це може бути перевагою, але для менших або більш гнучких проектів - недоліком.

В сукупності, методологія Waterfall є ефективною для проектів з чіткими, незмінними вимогами, а також там, де важлива детальна документація. Однак, в умовах, де вимоги швидко змінюються, і адаптивність є ключовою, Waterfall може виявитися менш вдалим вибором.

Agile, як гнучка методологія управління проектами, займає важливе місце в сучасному управлінні IT-проектами. Характеризуючись своєю ітеративністю

та гнучкістю, Agile дозволяє командам швидко адаптуватися до змін у вимогах проєкту та зовнішньому середовищі. Основною перевагою Agile є його висока адаптивність. Методологія дозволяє команді ефективно реагувати на зміни, що робить її ідеальною для проєктів у швидко змінюваних умовах, таких як розробка нових продуктів або стартапів. Наприклад, у випадку стартапу, що розробляє мобільний додаток, вимоги можуть швидко змінюватися в залежності від зворотного зв'язку користувачів, і Agile дозволяє швидко адаптувати функціональність додатка без необхідності перегляду всього проєкту.

Agile також підкреслює важливість колаборації та комунікації в команді, що сприяє більш ефективному вирішенню проблем та інноваційному мисленню. Через регулярні наради, такі як щоденні стендапи та спринт-ретроспективи, команди Agile здатні оперативно вирішувати проблеми та покращувати робочі процеси. Однак, Agile може стикатися з викликами у масштабних проєктах або в умовах, де потрібна висока ступінь передбачуваності та структурованості. Наприклад, великі корпоративні ІТ-системи з численними інтеграціями та залежностями можуть потребувати більш структурованого підходу, ніж той, що забезпечує Agile.

Agile також зосереджений на швидкому наданні цінності клієнту та покращенні продуктивності команди. Це досягається за рахунок ітеративного випуску продукту з регулярними оновленнями та поліпшеннями, що забезпечує постійний потік зворотного зв'язку та можливість оперативно реагувати на нього. Agile ефективно працює в динамічних та невизначених умовах, де швидкість, гнучкість та адаптивність мають вирішальне значення. Водночас, у проєктах, що вимагають жорсткої структуризації, високого рівня документації та передбачуваності, можуть виникнути складнощі з використанням Agile. Таким чином, ефективність Agile значною мірою залежить від специфіки проєкту та гнучкості організаційного середовища.

Kanban є методологією управління проєктами, що використовується для ефективного управління робочими процесами. Цей метод вперше був розроблений в Японії у Toyota Motor Corporation у 1940-х роках як система своєчасного управління запасами в автомобільній промисловості. Ключовою

ідеєю Kanban є оптимізація потоку робіт та зменшення часу, витраченого на виконання завдань [14].

Основна ідея Kanban полягає у візуалізації робочих процесів за допомогою спеціальної дошки, де завдання представлені картками, які переміщуються між різними стадіями виконання. Це дозволяє командам чітко бачити статус роботи та ефективно управляти навантаженням. Kanban найкраще працює в умовах стабільних, повторюваних виробничих процесів. Якщо завдання генерують багато підзавдань або потребують тривалого утримання, Kanban може виявитися неефективним [15, 16].

Методологія Scrum, є однією з найвідоміших та широко використовуваних реалізацій Agile-підходу в управлінні ІТ-проєктами. Ця методологія заснована на принципах гнучкості, колаборативності та ітеративного розвитку. Вона орієнтована на створення функціонального продукту через короткі цикли розвитку, відомі як спринти. Основною перевагою Scrum є його ітеративний підхід до розвитку продукту, що дозволяє командам швидко адаптуватися до змін у проєкті та ринкових умовах. Наприклад, у процесі розробки нового програмного забезпечення, Scrum дозволяє команді вносити зміни та покращення на основі зворотного зв'язку від кінцевих користувачів після кожного спринту.

У Scrum визначаються чіткі ролі: Scrum Master, Product Owner та команда розробників. Scrum Master забезпечує дотримання принципів Scrum, допомагаючи команді бути продуктивною. Product Owner відповідає за визначення вимог до продукту та пріоритетів. Ця структура сприяє чіткій координації роботи та ефективній комунікації всередині команди. Однак, Scrum може стикатися з труднощами при масштабуванні на великі проєкти або організації. В таких випадках, управління залежностями між різними командами та координація роботи на вищому рівні може стати складнішою.

Scrum ставить акцент на самоорганізацію команди та її продуктивність. Регулярні зустрічі, такі як щоденні стендапи, спринт-планування, ретроспективи та огляди спринтів, сприяють високому рівню взаєморозуміння в команді та постійному поліпшенню процесів. Scrum демонструє високу ефективність у середовищах, де необхідна швидка адаптація до змін, колаборативний підхід та

швидке впровадження продукту. Однак, для складних проєктів з численними залежностями або великих організацій, можуть знадобитися додаткові зусилля для координації та управління на вищому рівні. Таким чином, Scrum найбільш ефективний в умовах, де гнучкість, швидкість та співпраця є ключовими для успіху проєкту [17, 18].

2.3 Вибір оптимальних методів для конкретного ІТ-проєкту

При виборі оптимальних методів управління для конкретного ІТ-проєкту, важливо враховувати ряд ключових факторів, що впливають на успіх проєкту. Ці фактори включають специфіку проєкту, його масштаб, складність, динаміку вимог, а також ресурси та культуру організації.

Кожен ІТ-проєкт унікальний і має свої особливості, які слід враховувати при виборі методології. Наприклад, проєкт, що передбачає розробку заздалегідь визначеного програмного продукту з чіткими вимогами, може ефективно керуватися за допомогою традиційної методології Waterfall. Натомість, проєкт, що розвивається в умовах невизначеності, з частими змінами та потребою в швидкій адаптації, буде краще керований за допомогою Agile-методологій, таких як Scrum [19].

Масштаб та складність проєкту також відіграють важливу роль у виборі методології. Для великих проєктів з численними залежностями та інтеграціями може знадобитися більш структурований підхід з чіткими процесами управління. Тут можуть бути корисні гібридні методології, які поєднують елементи Waterfall та Agile. Важливим аспектом є також спроможність методології адаптуватися до змінних вимог. У динамічному ІТ-середовищі, де технології та потреби клієнтів швидко змінюються, гнучкість стає ключовою для успіху проєкту. Agile-методології, які надають високу гнучкість та можливість швидкого внесення змін, часто є гарним вибором для таких ситуацій.

Останнім, але не менш важливим фактором, є культура організації та готовність команди до прийняття та впровадження обраної методології. Важливо, щоб методологія була сумісна з корпоративними цінностями та стилем

роботи команди, а також щоб команда мала необхідні знання та навички для її ефективного застосування [20].

Вибір оптимальної методології для конкретного ІТ-проєкту вимагає комплексного підходу та глибокого розуміння унікальних аспектів проєкту. Врахування специфіки проєкту, масштабу, складності, динаміки вимог, а також культури організації є ключовими для забезпечення успішного управління проєктом [21].

Проєкт, для якого була розроблена модель управління, яка описується в цій магістерській роботі, зосереджується на розробці та підтримці веб-додатку, що характеризується високим рівнем динамічних змін та адаптації до змінних вимог ринку. Ключовою особливістю проєкту є його реалізація командою з 50 співробітників, які працюють на аутсорсинговій основі за довгостроковим контрактом.

Команда складається з професіоналів, які покривають всі аспекти розробки та підтримки веб-додатку, включаючи фронтенд та бекенд розробку, дизайн UX/UI, тестування, обслуговування інфраструктури та підтримку клієнтів. Робота організована таким чином, щоб забезпечити ефективну координацію між різними підкомандами та високу адаптивність до змінних вимог проєкту.

Веб-додаток розробляється з використанням сучасних технологій та фреймворків, що гарантує високий рівень масштабованості, безпеки та користувацького досвіду. Акцент робиться на створенні адаптивного дизайну та інтеграції з різноманітними зовнішніми сервісами та API. З огляду на динаміку змін та необхідність швидкої адаптації до нових вимог, для управління проєктом на початку роботи було обрано Agile-методологію, зокрема Scrum. Це дозволяло команді оперативно реагувати на зміни, ефективно управляти пріоритетами та постійно підтримувати високу якість продукту. Однак з часом виникла потреба в масштабуванні Scrum, оскільки зі збільшенням розміру проєкту та кількості задіяних команд зросла необхідність в координації та управлінні залежностями.

Проєкт передбачає тісну взаємодію з клієнтом для забезпечення відповідності веб-додатку бізнес-вимогам та очікуванням користувачів. Регулярні зустрічі та зворотний зв'язок від клієнта є важливою частиною процесу розвитку. Використання аутсорсингової моделі співпраці в рамках

довгострокового контракту дозволяє забезпечити гнучкість у масштабуванні ресурсів проєкту та доступ до висококваліфікованих фахівців.

Фактори, які призвели до необхідності розробки нової моделі управління проєктом:

1. Зросла складність проєкту. У міру розширення проєктів та збільшення кількості задіяних команд, виникає потреба в більш ефективному управлінні складними залежностями і координації між різними частинами проєкту.
2. Потреба в стандартизації та єдності процесів. Великі проєкти часто вимагають уніфікації та стандартизації процесів управління для забезпечення консистентності та ефективності в роботі різних команд.
3. Висока динаміка та потреба у швидкій адаптації. У середовищах, де проєктні вимоги та умови швидко змінюються, зростає потреба в методології, яка дозволяє гнучко адаптуватися до нових викликів та умов.
4. Складності інтеграції та доставки продукту. У великих проєктах критично важливою стає здатність інтегрувати роботу різних команд для забезпечення високої якості кінцевого продукту.
5. Підвищена потреба у прозорості та контролі. Масштабні проєкти вимагають збільшеної прозорості та кращого контролю за ходом виконання робіт для ефективного управління проєктом.
6. Виклики міжкомандної взаємодії. Збільшення кількості команд у проєкті вимагає більш ретельної координації та співпраці між ними для забезпечення ефективної роботи.

У сфері управління ІТ-проєктами, особливо на великих масштабах, існує ряд факторів, які обумовлюють необхідність розробки нових моделей управління. Перш за все, це пов'язано зі зростанням комплексності та масштабу проєктів, що вимагає більш витончених механізмів координації та управління взаємозалежностями між різними частинами проєкту. У такому контексті, стандартизація процесів стає ключовою для забезпечення консистентності та ефективності роботи великої кількості команд [22].

В даному випадку, у сукупності всі ці фактори зумовили необхідність розробки нової моделі управління ІТ-проектом, яка б змогла вирішувати виклики сучасного динамічного та комплексного проектного середовища. Нова модель повинна бути гнучкою, адаптивною, забезпечувати високу ступінь інтеграції та координації між командами, а також дозволяти ефективно управляти проектом з урахуванням всіх залежностей та змінних умов.

2.4 Опис розробленої моделі на основі фреймворку Nexus

У рамках дослідження нових підходів до управління проектами, вибір було зроблено на користь одного з фреймворків Scrum – Nexus, який виступає як фундаментальна основа. Фреймворк Nexus, який був спеціально розроблений для масштабування методології Scrum у контекстах, що вимагають високої ступені координації робіт між численними Scrum-командами, забезпечує вирішення типових проблем, які виникають при адаптації Scrum в умовах великомасштабних та складних проектів [23]. Nexus дозволяє інтегрувати від 3 до 9 Scrum-команд, що спільно працюють над одним продуктом або проектом, створюючи умови для ефективної синергії та взаємодії. Схематичне зображення фреймворку показано на рисунку 2.1

NEXUS™ FRAMEWORK

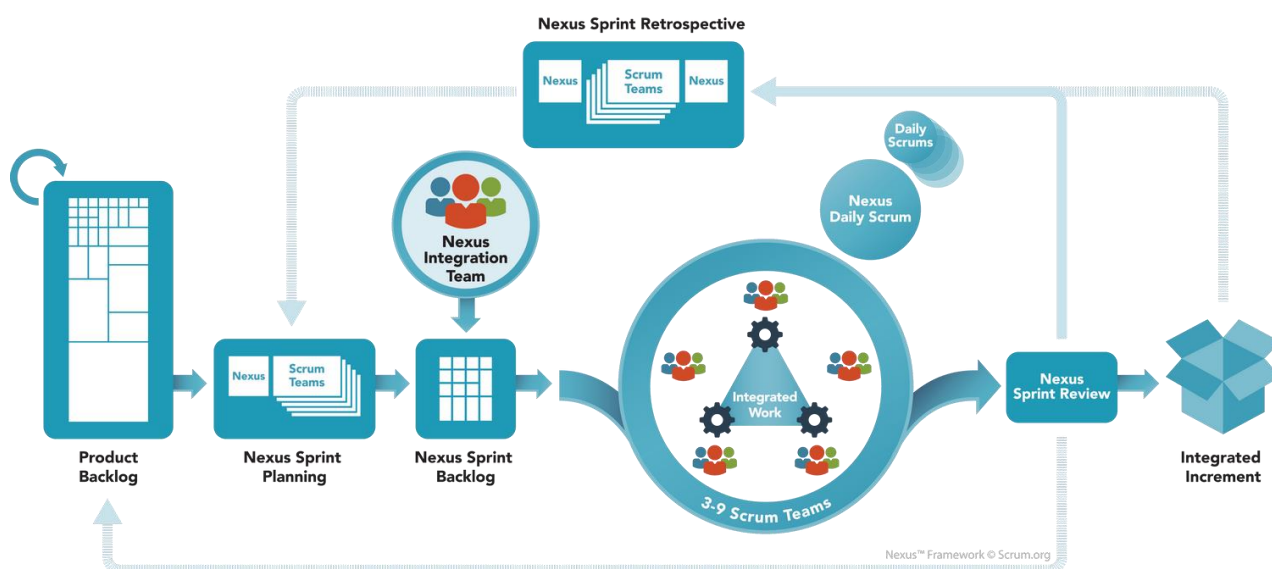


Рисунок 2.1 – Схематичне зображення фреймворку Nexus

Такий допомагає синхронізувати роботу та забезпечити послідовний напрямок розвитку. Фреймворк зосереджений на поліпшенні взаємодії та співпраці між командами. Це включає щоденні зустрічі, планування спринтів, ретроспективи та огляди, які відбуваються на мультикомандному рівні. Nexus підкреслює важливість прозорості у всіх аспектах проєкту. Це включає спільні артефакти та інструменти управління, які дозволяють усім учасникам проєкту бути обізнаними про актуальний стан виконання робіт та можливі виклики і ризики.

Відповідно до нової моделі, одна загальна велика команда, яка існувала на проєкті, була структурно розподілена на нові команди, що проілюстровано у таблиці 2.1.

Feature teams – це команди, які зосереджені на розробці нових функціональних можливостей та їх інтеграції в системи, що вже імплементовані. Це включає аналіз вимог, проєктування, написання коду та впровадження, тестування та впровадження. До складу кожної команди входять фронтенд-розробники, бекенд-розробники, тестувальники та дизайнер.

Таблиця 2.1 – Нові сформовані команди

Назва команди	Кількість спеціалістів у команді	Кількість команд
Features Teams	8	5
Nexus Integration Team (NIT)	2	1
Базова Core Team	4	1
NIT	4	1

До основних завдань кожної Feature Team належать:

1. Розробка функціональності. Команди відповідають за розробку та імплементацию специфічних функціональних можливостей в продукті. Це

включає визначення вимог, проєктування, програмування, тестування та впровадження функціональності.

2. Ітеративний розвиток. Вони працюють у рамках ітеративного циклу розвитку, використовуючи принципи гнучкої методології Agile, що дозволяє швидко адаптуватися до змін у вимогах.

3. Співпраця та комунікація. Команди здійснюють тісну співпрацю з іншими командами та зацікавленими сторонами, включаючи Product Owner, аналітиків, менеджерів продуктів зі сторони клієнта, для забезпечення відповідності розробки загальній стратегії продукту.

4. Технічне вдосконалення та підтримка. Відповідальність за технічне вдосконалення існуючих компонентів, а також за підтримку та вирішення проблем, що виникають після впровадження нової функціональності.

5. Неперервний аналіз та вдосконалення. Команди регулярно аналізують робочі процеси, впроваджують покращення для підвищення ефективності та якості роботи, а також відстежують нові технологічні тенденції та практики.

6. Керування якістю та тестування. Забезпечують високу якість розробленої функціональності шляхом ретельного тестування та перевірки відповідності вимогам, включаючи автоматизоване та ручне тестування.

Основний принцип формування спеціалізованих функціональних команд базується на здатності цих команд до реалізації повноцінних функціональних можливостей (фіч), починаючи з етапу розробки прототипів (мокапів) і завершуючи етапом тестування. Це включає послідовність операцій: концептуалізацію, проєктування, розробку, інтеграцію та перевірку, що забезпечує повний цикл створення функціональності з мінімальною залежністю від зовнішніх ресурсів або команд.

NIT - це спеціально призначена команда, яка відповідає за координацію та інтеграцію роботи всіх Scrum-команд. NIT забезпечує, щоб всі аспекти проєкту були синхронізовані та що інтеграція відбувається послідовно та зрозуміло для всіх учасників інших команд. До її складу на даному проєкті увійшли ПМ та Scrum master.

Базова команда Core Team складається з фахівців у сфері бекенд-розробки, які відповідають за створення та підтримку централізованого бекенд

API, ключового компоненту системи. Даний API слугує як основа для функціональних команд, що забезпечує їх спільну роботу та взаємодію. Крім того, склад команди включає в себе Tech Lead проєкту, який відіграє ключову роль у забезпеченні технічної якості продукту та ефективності робочих процесів

Команда DevOps, у своєму функціональному вимірі, є відповідальною за інтеграцію та безперервну доставку (CI/CD) розроблюваних програмних рішень. Її ключовими завданнями є розробка та підтримка інфраструктури, оптимізація процесів автоматизації, забезпечення безперервності та стабільності робочих процесів, а також моніторинг системного середовища. Команда DevOps фокусується на підтримці взаємодії між розробкою програмного забезпечення та його експлуатацією, що дозволяє досягти ефективної координації між різними стадіями життєвого циклу програмного продукту. Основною метою їх діяльності є максимізація швидкості, надійності та безпеки процесів розгортання та підтримки програмних продуктів.

У кожній з новостворених команд є Team Lead, який несе відповідальність за кінцевий результат роботи своєї команди. Лідери команд виконують роль ключових координаторів, забезпечуючи не тільки внутрішнє управління своєю командою, але й участь у міжкомандних зустрічах на рівні Nexus. Це включає активну участь у стратегічних плануваннях, координаційних засіданнях та інших організаційних заходах, які передбачають міжкомандний обмін інформацією та узгодження загальних цілей і стратегій. Роль Team Lead також передбачає активне сприяння інтеграції зусиль усіх команд для забезпечення загальної координації та ефективності проєкту.

Нова модель управління базується на інтеграції декількох Scrum-команд, які працюють у координації одна з одною для досягнення спільних цілей проєкту. Центральним елементом є Nexus Integration Team (NIT), яка відіграє роль організаційного та технічного координатора між усіма учасниками. NIT забезпечує належну інтеграцію робіт, вирішення проблем, що виникають на міжкомандному рівні, та підтримку єдності процесів у всьому проєкті.

Основою моделі є ітеративний підхід Scrum, що передбачає регулярне планування, виконання, перевірку та коригування роботи [24]. Циклічність спринтів дозволяє командам оперативно реагувати на зміни у проєкті та

адаптувати свою роботу до нових вимог. Синхронізація роботи між командами та регулярні огляди сприяють забезпеченню цілісності та високої якості кінцевого продукту.

Ефективна комунікація є ключовим аспектом у моделі. Встановлення чітких каналів зв'язку між командами, регулярні зустрічі для обговорення прогресу та вирішення проблем забезпечують високу прозорість роботи та сприяють формуванню спільного бачення цілей проєкту. Модель розроблена таким чином, щоб бути максимально адаптивною та гнучкою до змінних умов проєкту. Механізми для швидкої адаптації до змін у вимогах, технологіях та ринковому середовищі є інтегрованими в структуру моделі, дозволяючи проєкту ефективно реагувати на нові виклики.

2.5 Особливості нової моделі

У розробленій моделі управління ІТ-проєктами, що базується на фреймворку Nexus, втілені специфічні особливості та ритуали, які спрямовані на оптимізацію роботи великих команд. Ці особливості охоплюють широкий спектр процесів, від планування до виконання робіт. Спеціально призначена команда NIT відповідає за координацію та інтеграцію роботи всіх Scrum-команд. NIT забезпечує, щоб всі аспекти проєкту були синхронізовані та була змога впевнитись, що інтеграція відбувається плавно (а якщо це не так – буде можливість розробити та імплементувати зміни). Основна задача NIT полягає у забезпеченні, щоб всі команди працювали над спільною метою та що їх індивідуальні внески гармонійно інтегровані в один продукт. Це включає координацію планів, технічних вимог та часових рамок. Для забезпечення успішної інтеграції, NIT впроваджує різноманітні інтеграційні практики, які включають регулярне тестування інтеграції, вирішення проблем, які виникають при злитті різних частин продукту, та забезпечення неперервної інтеграції.

2.5.1 Щоденні зустрічі

Впровадження нової моделі управління, яка базується на принципах Nexus, призводить до вдосконалення комунікаційних процесів у кожній з команд. Одним з ключових елементів є проведення щоденних ранкових зустрічей daily meetings, які обмежуються 15-хвилинним форматом. Така структурована та часово ефективна організація зустрічей сприяє концентрації на нагальних завданнях і виявленні перешкод, тим самим мінімізуючи витрати часу на непродуктивні обговорення. Питання, що потребують детальнішого розгляду та можливої координації з іншими командами, виносяться на загальний стендап мітинг на рівні Nexus. Це забезпечує наявність механізму для ефективного вирішення складних інтеграційних питань, що виходять за межі однієї команди.

Важливою особливістю є те, що загальний стендап на рівні Nexus не обмежується участю лише лідерів команд. Модель передбачає активне залучення різних учасників команди, що сприяє розподілу відповідальності та зниженню ризику вигорання, оскільки ніхто не перевантажений постійною участю у великій кількості зустрічей.

Така модель управління, яка втілює ідеї Nexus, вносить суттєві вдосконалення в процеси комунікації та координації в командах, забезпечуючи ефективність обговорень і водночас підтримуючи високий рівень залученості учасників. Через оптимізацію часу, використаного на зустрічі, та структурований підхід до обговорення завдань, модель сприяє підвищенню продуктивності та зниженню втоми, що є ключовим для успіху великих та складних ІТ-проектів.

2.5.2 Product backlog refinement

У рамках вдосконаленої моделі управління на основі Nexus, важливим елементом є проведення регулярних мітингів з планування роботи над продуктом, відомих як мітинги з очищення (refinement) продуктового беклогу. Ці зустрічі мають стратегічне значення для ефективної підготовки до майбутніх спринтів, оскільки вони дозволяють командам детально аналізувати та планувати майбутні завдання.

На цих зустрічах команди проводять глибокий аналіз задач, які планується виконати у наступних спринтах. Цей процес включає попередню оцінку завдань за складністю та обсягом роботи, а також їх декомпозицію на більш дрібні та зрозумілі частини. Така деталізація задач забезпечує краще розуміння роботи, яка має бути виконана, та сприяє більш точному плануванню. Ці мітинги є важливими для того, щоб кожна команда мала чітке уявлення про свої завдання на наступний спринт. Попередній аналіз та планування дозволяють командам підходити до кожного спринту з чітким планом дій та розумінням очікуваного обсягу роботи.

Проведення мітингів з очищення продуктового беклогу перед спринтами є ключовим для успіху моделі управління на основі Nexus, оскільки це забезпечує не тільки ефективну підготовку та планування завдань, але й сприяє кращій координації та розумінню між командами. Цей підхід підвищує якість планування та допомагає уникнути непорозумінь та зайвих витрат часу під час виконання спринту.

2.5.3 Проведення Sprint Review

Sprint Review є невід'ємною частиною процесу Scrum, яка у моделі управління на основі Nexus набуває додаткових особливостей. Цей процес включає оцінку роботи, виконаної командами протягом спринту, та є важливим для забезпечення прозорості та відкритого діалогу між командами та зацікавленими сторонами.

У рамках Sprint Review, кожна команда представляє результати своєї роботи, включаючи завершені завдання та досягнуті результати. Це дає можливість не тільки демонструвати прогрес, але й отримувати зворотний зв'язок від інших учасників проєкту, зокрема від представників Nexus Integration Team, Product Owners та інших зацікавлених сторін. Важливим аспектом Sprint Review є також обговорення того, як результати впливають на подальший розвиток продукту. Це включає оцінку поточного стану продуктового беклогу, планування майбутніх спринтів та внесення необхідних коригувань у стратегію проєкту.

Sprint Review у моделі Nexus також сприяє зміцненню взаєморозуміння між різними командами та забезпечує координацію спільних зусиль. Це допомагає уникнути дублювання роботи, забезпечує єдине бачення загального напрямку проєкту та допомагає вирішувати міжкомандні проблеми.

Загалом, Sprint Review в моделі Nexus є не тільки механізмом звітності та отримання зворотного зв'язку, але й засобом для підтримки неперервної адаптації та вдосконалення процесів управління проєктом. Вона забезпечує відкритість та прозорість у роботі команд, що є важливим для успішної реалізації складних ІТ-проєктів.

2.5.4 Проведення Sprint Retrospective

Sprint Retrospective є ключовим процесом у Scrum-методології, який в новій моделі управління на основі Nexus виконується у два етапи: спочатку на рівні індивідуальних команд, а потім на загальному рівні Nexus. Цей підхід дозволяє ефективно виявляти та адресувати проблеми, які виникають у проєкті, та забезпечує вдосконалення процесів на всіх рівнях.

На першому етапі кожна команда проводить свій власний Sprint Retrospective, де аналізуються специфічні виклики, з якими команда зіткнулася під час спринту. Це включає обговорення ефективності робочих процесів, комунікації, взаємодії в команді та інших аспектів. Команди розробляють план дій для вирішення внутрішніх проблем, вдосконалення своєї роботи та підвищення ефективності у наступних спринтах.

Після індивідуальних зустрічей відбувається загальна Nexus Retrospective. На цій зустрічі представники кожної команди, разом з Nexus Integration Team, обговорюють проблеми та виклики, які впливають на кілька команд або весь проєкт. Основна увага приділяється вирішенню проблем, які виходять за рамки однієї команди і вимагають міжкомандної взаємодії для їх вирішення. Команди розробляють спільні стратегії та плани дій для покращення інтеграції та ефективності на рівні всього проєкту.

Двоетапний підхід до проведення Sprint Retrospective в моделі Nexus дозволяє не тільки вирішувати внутрішньокомандні проблеми, але й ефективно

адресувати міжкомандні виклики та вдосконалювати загальні процеси проєкту. Така організація зустрічей сприяє побудові більш гармонійної, ефективної та інтегрованої робочої атмосфери, що є критично важливим для успішної реалізації великих та складних ІТ-проєктів.

2.5.5 Пріоритезація в backlog

Пріоритезація беклогу є фундаментальним процесом в моделі управління на основі Nexus, який вимагає злагодженої взаємодії та координації між всіма командами проєкту. Основною метою цього процесу є визначення та ранжування завдань за їх значимістю та терміновістю, щоб забезпечити оптимальне використання ресурсів і часу, а також максимізувати цінність продукту для кінцевого користувача.

У контексті Nexus, процес пріоритезації беклогу включає кілька ключових аспектів. По-перше, він вимагає активної участі лідерів кожної Scrum-команди, які співпрацюють з NIT для забезпечення єдності та послідовності беклогу на всьому проєкті. Це забезпечує, що всі команди працюють над завданнями, які мають найбільше значення для загальних цілей проєкту.

Другий важливий аспект полягає у застосуванні систематичного підходу до оцінки завдань. Завдання оцінюються за критеріями, такими як їх вплив на бізнес-цілі, технічну складність, ресурсні вимоги, та ризики. Це дозволяє об'єктивно визначити пріоритети та вирішити, які завдання слід виконувати першочергово.

Третім елементом є неперервна адаптація беклогу. У динамічному проєктному середовищі, де зміни відбуваються регулярно, беклог не є статичним. Він постійно оновлюється та адаптується відповідно до нових вимог, змін у ринкових умовах, або зворотного зв'язку від користувачів.

Четверте – увага приділяється забезпеченню прозорості та відкритості у процесі пріоритезації. Всі члени команди мають доступ до беклогу, що сприяє загальному розумінню напрямків роботи та забезпечує можливість для внесення пропозицій та обговорення.

Підсумовуючи, пріоритезація беклогу в моделі Nexus є складним, але водночас систематичним процесом, який вимагає активної участі різних учасників проєкту, об'єктивної оцінки завдань та гнучкості у прийнятті рішень. Це забезпечує, що проєкт розвивається у відповідності до змінюваних потреб та умов, максимізуючи вклад кожної команди у загальний успіх.

Основними характеристиками нової моделі є інтеграція команд, ефективне використання щоденних стендапів, проведення Sprint Retrospective та Sprint Review, а також систематична пріоритезація беклогу. Ці елементи сприяють вибудовуванню чіткої структури управління, що дозволяє кожній команді працювати над своїм сегментом проєкту з високою ефективністю та продуктивністю.

Важливою перевагою моделі є її адаптивність та здатність до масштабування, що робить її придатною для проєктів різної складності та розміру. Модель також забезпечує прозорість у всіх аспектах проєкту, від відкритої комунікації до спільної відповідальності за кінцевий результат. У підсумку, нова модель на основі Nexus виявляється ефективним інструментом управління, який сприяє досягненню високих результатів у розробці та виконанні складних ІТ-проєктів. Її впровадження в проєктні процеси дозволяє оптимізувати роботу команд, підвищити якість продукту та забезпечити його вчасну та ефективну доставку.

3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТА АПРОБАЦІЯ МОДЕЛІ

Експеримент присвячений дослідженню та апробації нової моделі управління на основі Nexus, також він спрямований на оцінку практичної ефективності моделі у реальних умовах роботи, виявлення її сильних сторін та потенційних областей для подальшого удосконалення. Основною метою є визначення, чи нова модель забезпечує покращення у продуктивності, якості роботи та комунікації в команді порівняно з традиційними методами управління. В рамках дослідження були використані різні методи збору та аналізу даних, що дозволить провести всебічну оцінку та вивести обґрунтовані висновки про ефективність запропонованої моделі.

3.1 Методологія експерименту

Методологія експерименту зі створення нової моделі управління проєктами на основі Nexus вимагає комплексного підходу, що інтегрує теоретичні знання з практичним застосуванням. Це включає ретельне планування, чітке визначення цілей та гіпотез, а також систематичне збір та аналіз даних. Основною метою є визначення ефективності нової моделі в реальних умовах та її порівняння з традиційними методами управління проєктами.

Ініціювання експерименту починається з чіткого формулювання цілей, що включає визначення ключових показників ефективності нової моделі. Основні гіпотези, які було сформовано і які мають бути перевірені під час експерименту наведено в таблиці 3.1.

Наступний етап включає вибір методів збору та аналізу даних, а також планування ресурсів і часових рамок.

Методи збору та аналізу даних, які були обрані для використання в цьому експерименті:

1. Кількісні опитування – використання структурованих опитувань для збору кількісних даних про продуктивність, часові показники та інші вимірювані аспекти роботи команд.

Таблиця 3.1 – Гіпотези, сформовані на початку експерименту

Гіпотеза	Короткий опис
Підвищення продуктивності	Нова модель на основі Nexus підвищує загальну продуктивність команд порівняно з традиційними методами управління
Ефективність координації та комунікації	Впровадження Nexus сприяє більш ефективній координації та комунікації між командами, зменшуючи час на вирішення міжкомандних проблем
Зменшення часу на розробку	Модель сприяє скороченню часу, необхідного для розробки та впровадження продукту
Підвищення якості продукту	Застосування моделі Nexus призводить до підвищення якості кінцевого продукту за рахунок кращої інтеграції та координації робіт
Гнучкість у відповіді на зміни	Модель дозволяє швидше та ефективніше реагувати на зміни у вимогах або умовах ринку
Задоволеність команди та зниження рівня вигорання	Впровадження Nexus покращує загальне задоволення команди та знижує рівень вигорання через кращу балансування робочого навантаження та чіткішу організацію
Краще управління ризиками та проблемами	Модель ефективно допомагає управляти ризиками та проблемами проєкту

2. Якісні опитування – проведення індивідуальних або групових інтерв'ю з членами команд, керівниками проєктів, а також зі стейкхолдерами для отримання глибокого розуміння їх досвіду та сприйняття нової моделі.

3. Аналіз документації – вивчення проєктної документації, звітів, планів роботи та інших документів для збору даних про процеси та результати роботи команд.

4. Спостереження – безпосереднє спостереження за робочими процесами, щоб оцінити взаємодію в командах, ефективність зустрічей та інші аспекти роботи.

5. Аналіз продуктивності – використання інструментів для відстеження продуктивності та виконання завдань, таких як системи керування проєктами або трекери задач.

6. Case Study – детальний аналіз конкретних випадків, щоб зрозуміти контекстуальні фактори та отримати глибоке розуміння впливу нової моделі.

При плануванні ресурсів та визначенні часових рамок для експерименту зі створення нової моделі управління на основі Nexus, важливо було враховувати кілька ключових аспектів. Перш за все, було визначено кількість та склад команд, які братимуть участь у експерименті. У рамках експерименту з новою моделлю управління на основі Nexus створено п'ять Feature Teams з восьми спеціалістів кожна, одна базова Core Team з чотирьох учасників, одна команда DevOps також з чотирьох осіб, та Nexus Integration Team з двома учасниками, яка координує та інтегрує роботу всіх команд.

Важливим елементом є також підготовка та навчання учасників, щоб вони могли ефективно застосовувати модель Nexus. Це включає організацію тренінгів та семінарів, а також забезпечення доступу до необхідних технічних засобів та інструментів для співпраці. Бюджет експерименту є також важливим компонентом, оскільки він повинен покривати витрати на навчання, технологічне забезпечення, а також можливі витрати на зовнішні консультації або експертну підтримку.

Загальна тривалість експерименту склала 6 місяців. Часові рамки можна розділити на такі етапи:

- 1-2 місяць – підготовчий період, включаючи формулювання цілей та гіпотез, планування ресурсів, навчання та орієнтацію учасників щодо моделі Nexus, та підготовку технічних засобів.
- 3-4 місяць – активне впровадження моделі у проєкт, збір перших даних та початковий моніторинг ефективності. Проведення регулярних зустрічей для збору зворотного зв'язку від команд.

- 5 місяць – продовження моніторингу та збору даних, аналіз проміжних результатів, внесення коригувань у модель за необхідності.
- 6 місяць - фінальний збір та аналіз даних, підготовка звітів з експерименту, формулювання висновків та рекомендацій щодо ефективності та можливих удосконалень моделі.

Такий розподіл дозволяє на кожному етапі забезпечити ретельну підготовку, ефективне впровадження та аналіз результатів, що є ключовими для оцінки ефективності моделі Nexus у реальних умовах

Завершальним етапом експерименту є підготовка звітів та аналітичних матеріалів, які подаються по завершенню дослідження, щоб оцінити ефективність та вплив нової моделі.

3.2 Оцінка результатів експерименту

Оцінка результатів експерименту з впровадження нової моделі управління на основі Nexus та аналіз зібраних даних дозволяє оцінити ступінь досягнення визначених цілей, а також виявити ключові показники покращення або проблемні зони.

3.2.1 Оцінка якості комунікації

На основі проведеного опитування серед учасників команд, що брали участь у експерименті з впровадження нової моделі управління на основі Nexus, було отримано значні позитивні відгуки стосовно покращення якості комунікації та координації. Систематичне збирання відповідей дозволило виявити, що загальний рівень задоволеності процесами комунікації серед членів команд зріс з середнього показника 5 до 8 на шкалі від 1 до 10.

Це покращення можна пояснити кількома факторами. По-перше, модель Nexus сприяє створенню більш структурованих та ефективних каналів комунікації. Це включає регулярні зустрічі між командами, що дозволяє учасникам обмінюватися важливою інформацією та координувати свої зусилля. По-друге, впровадження чітких ролей та відповідальності у межах Nexus

Integration Team забезпечило більш ефективне управління інтеграцією між різними командами. Це не лише полегшило взаємодію між командами, але й сприяло зменшенню можливості виникнення непорозумінь або конфліктів.

Опитування також показали, що вдосконалені процеси комунікації позитивно впливають на загальне відчуття задоволеності роботою серед членів команд. Збільшення прозорості у процесах рішення та планування, а також покращення доступу до важливої інформації зробили процеси більш зрозумілими та передбачуваними для учасників.

Таким чином, результати опитування свідчать про ефективність нової моделі управління в контексті покращення комунікації та координації між командами, що відображається у зростанні рівня загальної задоволеності комунікаційними процесами.

3.2.2 Оцінка продуктивності

Оцінка продуктивності в рамках експерименту з новою моделлю управління на основі Nexus виявила значні покращення у роботі команд. За даними, отриманими в ході експерименту, спостерігалось зниження часу, необхідного для виконання проектних завдань, а також поліпшення загальної ефективності робочих процесів.

Порівняння показників до та після впровадження моделі Nexus демонструє відчутне скорочення часу на розробку. Наприклад, середній час, необхідний для завершення визначених проектних етапів, зменшився на 25%. Це може бути пояснено кращою координацією між командами, зменшенням переривань у роботі та більш ефективним розподілом завдань. Крім того, було відмічено зниження кількості помилок на етапі тестування продукту. Зменшення кількості помилок на 40% свідчить про підвищення якості роботи та більш глибоке розуміння технічних вимог проекту. Це також відображає кращу синергію між командами та підвищення рівня експертизи учасників.

Зазначені поліпшення у продуктивності суттєво позитивно впливають на загальну ефективність реалізації проекту. Зниження часу на розробку та

кількості помилок не лише знижує витрати, але й сприяє кращому дотриманню графіків роботи та планів проекту [25, 26].

Подібні результати підтверджують, що впровадження нової моделі управління на основі Nexus сприяло підвищенню продуктивності команд, що є ключовим фактором успіху в умовах сучасних вимог до швидкості та якості розробки програмного забезпечення.

3.2.3 Оцінка відсотка завершених спринтів

Оцінка відсотка успішно завершених спринтів в рамках експерименту з впровадженням нової моделі управління на основі Nexus виявила суттєве покращення в здатності команд ефективно досягати поставлених цілей. Зі збіркою та аналізом даних було встановлено, що процент успішно завершених спринтів значно зріс порівняно з періодом до впровадження моделі.

Це поліпшення можна пояснити кількома факторами, притаманними моделі Nexus. По-перше, підвищення рівня організації та чіткість визначення ролей і відповідальності в межах Nexus Integration Team забезпечили більш ефективну координацію роботи команд. Це допомогло уникнути зайвих перепланувань і затримок, сприяючи своєчасному виконанню завдань.

По-друге, застосування принципів Nexus сприяло підвищенню прозорості у процесах планування та виконання робіт. Регулярні зустрічі, щоденні стендапи та постійний обмін інформацією між командами дозволили виявляти та вирішувати потенційні проблеми на ранніх стадіях. Це, у свою чергу, знизило ризики, що могли перешкодити успішному завершенню спринтів.

Внаслідок цих змін, команди стали більш гнучкими та адаптивними до змінних умов та вимог, що також позитивно вплинуло на успішність спринтів. Відсоток успішно завершених спринтів відображає не лише покращення в управлінні та координації, але й загальне підвищення продуктивності та ефективності команд.

3.2.4 Оцінка рівня задоволеності замовників

У рамках експерименту з впровадженням нової моделі управління на основі Nexus, була проведена оцінка задоволеності клієнтів, результати якої виявилися вражаючими. Для збору даних використовувалися опитування та анкетування, що охоплювали різні аспекти співпраці та якості продукту.

Порівняння відповідей клієнтів до та після впровадження моделі Nexus показало значне покращення у кількох ключових областях, які показано в таблиці 3.2.

Ці результати вказують на те, що впровадження моделі Nexus позитивно позначилося на робочих процесах та кінцевих результатах, викликавши покращення у загальній задоволеності клієнтів. Поліпшення в якості, швидкості та ефективності комунікації є ключовими факторами, що сприяли цьому зростанню задоволеності.

3.2.5 Оцінка якості кінцевого продукту

В рамках експерименту з впровадженням нової моделі управління на основі Nexus, було здійснено глибоку оцінку якості розробленого веб-додатку. Ця оцінка базувалася на ретельному аналізі різних аспектів продукту, включаючи функціональність, користувацький інтерфейс, продуктивність та відповідність вимогам клієнтів.

Перевірка функціональності веб-додатку показала, що після впровадження Nexus було досягнуто високого рівня відповідності специфікаціям та вимогам. Кількість функціональних помилок, виявлених під час тестування, зменшилася на 40%, що свідчить про підвищення точності в реалізації вимог.

Оцінка користувацького інтерфейсу веб-додатку виявила значні поліпшення в зручності користування та естетиці. Інтерфейс став більш інтуїтивно зрозумілим, а також були впроваджені сучасні дизайнерські тенденції, що покращило загальне сприйняття продукту кінцевими користувачами.

Таблиця 3.2 – Зворотний зв'язок від клієнтів

Назва показника	Зворотний зв'язок від клієнта
Задоволеність якістю продукту	Клієнти відзначили значне покращення в якості кінцевого продукту. Після впровадження Nexus, відсоток клієнтів, які висловили високу задоволеність якістю, зріс з 60% до 85%.
Відповідність вимогам	Було зазначено, що продукт краще відповідає вимогам та очікуванням клієнтів. Кількість клієнтів, які підтвердили, що продукт повністю відповідає їх вимогам, зросла з 50% до 78%.
Швидкість доставки продукту	Клієнти також відзначили покращення в швидкості доставки продукту. За даними опитувань, час від ідеї до реалізації продукту скоротився в середньому на 30%.
Комунікація та взаємодія	За результатами опитувань, показник задоволеності ефективністю комунікації зросла з 65% до 90%

Аналіз продуктивності веб-додатку показав значне покращення часу відгуку та швидкості обробки даних. Завдяки оптимізації коду та вдосконаленню архітектури, час відгуку скоротився на середньо 30%, що позитивно позначилося на користувацькому досвіді.

Таблиця 3.3 – Порівняльний аналіз показників ефективності проєкту перед та після впровадження Nexus

Показник	Scrum (до впровадження Nexus)	Nexus (після впровадження)
Середній час виконання задач (дні)	5,5	3,5
Рівень задоволеності команди (шкала 1-10)	6,5	8,2
Середня кількість виявлених помилок на етапі тестування	28	17
Процент успішно завершених спринтів	70%	90%
Тривалість спринту	12	10
Відсоток задач, завершених вчасно	64%	86%
Рівень участі команди у плануванні (шкала 1-10)	5	8

Порівняння веб-додатку з первісними вимогами клієнтів виявило, що високий рівень відповідності було досягнуто завдяки покращеній координації та комунікації в рамках Nexus. Зворотний зв'язок від клієнтів підтвердив, що продукт задовольняє їхні потреби та очікування.

3.3 Висновки

В рамках експерименту було зафіксовано, що впровадження моделі Nexus призвело до значного покращення управління проєктами. Це охоплює підвищення рівня управління ризиками, більшу гнучкість у відповідях на зміни, та покращення загальної ефективності проєктних команд.

Було зафіксовано покращення в організаційних процесах, включаючи планування, виконання та контроль проектних робіт. Завдяки більш ефективному розподілу завдань та координації між командами, процеси стали більш виразними та прозорими. Було відмічено суттєві зміни в культурі роботи в командах, включаючи зростання відкритості, співпраці, та спільної відповідальності за результати. Модель Nexus сприяла формуванню більш гнучкої та адаптивної робочої культури, що позитивно вплинуло на загальну атмосферу та динаміку в командах.

На основі аналізу даних було зроблено висновок, що впровадження моделі Nexus сприяло значному покращенню у продуктивності та якості роботи команд. Проте попри позитивні зміни, були виявлені певні виклики, зокрема у сфері інтеграції деяких командних процесів, що вимагають додаткової уваги та оптимізації:

- оцінка та управління потенційними ризиками в проекті потребує більшої уваги, особливо у контексті швидких змін та адаптацій.
- загальна комунікація суттєво покращилася, але деякі аспекти комунікації, особливо щодо обміну критичною інформацією та вирішення проблем у реальному часі, потребують додаткових удосконалень
- визначення та управління залежностями між різними завданнями та етапами проекту виявилось складним та потребує подальшого вдосконалення.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

В умовах прискореного розвитку інформаційних технологій, професії у сфері ІТ займають ключове місце в економічній структурі та соціальному устрої. Відповідно, набуває актуальності забезпечення адекватних умов праці та охорони здоров'я для спеціалістів цієї галузі. Попри загальноприйняту думку про порівняно низький ризик в роботі програмістів порівняно з іншими професіями, ця сфера містить специфічні ризики та особливості, що вимагають індивідуального підходу до питань охорони праці.

Аналіз робочого середовища ІТ-спеціалістів показує, що, незважаючи на зовнішню зручність, існують недоліки з точки зору ергономіки, психологічного навантаження та інших важливих аспектів. У цьому контексті важливим є вивчення основних аспектів охорони праці програмістів, аналіз потенційних ризиків та розробка рекомендацій щодо оптимізації умов праці для фахівців у галузі інформаційних технологій.

4.1 Питання щодо охорони праці

Фактори трудового середовища можуть істотно впливати на здоров'я та працездатність розробника програм. Неправильно організоване робоче місце може викликати проблеми із хребтом, шиєю, зап'ястям та іншими частинами тіла. Довгий час роботи за комп'ютером може призвести до тунельного синдрому зап'ястного каналу.

4.1.1 Ергономіка

У сучасних умовах трудової діяльності значуще місце займає організація робочого місця користувачів персональних комп'ютерів. Законодавство України наголошує на необхідності забезпечення безпечних та комфортних умов праці. Зокрема, відповідно до КЗпП та Закону України «Про охорону праці», роботодавці мають обов'язки стосовно забезпечення працівників належними умовами.

користувача: висота – мінімум 600 мм, ширина – мінімум 500 мм, глибина на рівні колін – 450 мм і на рівні витягнутої ноги – не менше 650 мм.

Стілець на робочому місці повинен мати підйомно-поворотні характеристики, можливість регулювання по висоті, куту нахилу сидушки і спинки, а також відстані від спинки до зовнішнього краю сидіння. Поверхня сидіння має бути рівною, а її зовнішній край має мати округлу форму. Налаштування по кожному параметру повинно бути індивідуальним, інтуїтивним та надійно фіксуватися.

Інтервал регулювання частин стільця: для лінійних розмірів – 15-20 мм, для кутових 2-5°. Зусилля для регулювання не має перевищувати 20Н. Висота сидіння повинна бути від 400 до 500 мм, а її ширина і глибина – не менше 400 мм. Сидіння має мати можливість нахилу до 15° вперед і до 5° назад. Висота спинки стільця – 300 ± 20 мм, ширина – не менше 380 мм, радіус кривизни горизонтально – 400 мм. Кут нахилу спинки може регулюватися в діапазоні 1-30° від вертикального положення. Відстань від спинки до сидіння – від 260 до 400 мм. Для зменшення напруження в руках рекомендується використовувати підлокітники довжиною від 250 мм, шириною 50-70 мм, що налаштовуються по висоті від 230 до 260 мм та по відстані між ними від 350 до 500 мм. Матеріал сидіння і спинки має бути напівтвердим, антиковзним, що пропускає повітря і легко миється, а також не збирає статичний заряд.

Робоча зона повинна бути обладнана підніжкою шириною мінімум 300 мм, глибиною не менше 400 мм, з можливістю регулювання по висоті до 150 мм і нахилом до 20°. На підніжці має бути рельєфна поверхня і маленький борт по зовнішньому краю висотою 10 мм. Екран комп'ютера слід розміщувати на оптимальній відстані від користувача, що варіюється в межах 500-700 мм, але не ближче ніж 500 мм, з урахуванням легкості читання тексту і зображень. Екран повинен бути розташований так, щоб забезпечити комфорт при спостереженні, кутом у 30° від вертикалі.

4.1.2. Освітлення

Ефективне та грамотне виробниче світло підвищує якість зорової діяльності, зменшує втомленість, стимулює зростання продуктивності, сприяє комфортному робочому середовищу, додаючи спокій та позитив працівникам, а також підсилює безпеку роботи, зменшуючи ризик травм. Недостатнє або надто яскраве освітлення може спричинити зорове перевтомлення і головний біль.

Недолік світла може призводити до перевантаження очей, зниження уваги та передчасної втоми. Занадто яскраве світло викликає сліпоту, незадоволення та відчуття дискомфорту в очах. Неправильне розташування світла може утворювати відблиски, контрастні тіні та дезорієнтувати працівника. Ці фактори можуть спричинити нещасний випадок або професійні захворювання, тому важливо правильно розраховувати інтенсивність світла.

Розрізняють три типи світла - природне, штучне та комбіноване.

Природне світло – це варіант освітлення, при якому денне світло проникає через вікна або інші прозорі елементи приміщення. Інтенсивність природного світла може сильно коливатися в залежності від доби, сезону та інших факторів. Штучне світло використовують у вечірній час або коли природне світло недостатнє. Якщо природне світло доповнюється штучним, таке освітлення називають змішаним.

За своїм призначенням штучне світло може бути робочим, евакуаційним, аварійним чи охоронним. Робоче світло поділяється на загальне та місцеве. При загальному освітленні світильники розташовані рівномірно по приміщенню. У системі комбінованого освітлення до загального додається додаткове місцеве світло.

Згідно норм «ДБН В.2.5-28:2018 Природне і штучне освітлення», в комп'ютерних залах слід застосовувати комбінований тип освітлення. Для виконання робіт високої зорової точності (об'єкт розрізнення 0,3 ... 0,5 мм) інтенсивність природного світла має бути не менше 1,5%. Для робіт середньої точності (об'єкт розрізнення 0,5 ... 1,0 мм) - не менше 1,0%. Як джерела штучного світла часто використовують лампи типу ЛБ або ДРЛ, об'єднані в світильниках над робочими столами.

Вимоги до освітленості для робочих місць з комп'ютерами такі: при високій точності зорової роботи – 300 лк (загальне) та 750 лк (комбіноване); при середній точності – 200 та 300 лк відповідно.

Також важливо, щоб усе поле зору було якісно освітлене. Світлова інтенсивність приміщення та яскравість екрану комп'ютера повинні бути близькими, адже надлишкове світло може збільшувати втомленість очей.

4.1.3. Параметри мікроклімату

Для створення комфортних умов праці в приміщеннях з комп'ютерною технікою важливо дотримуватися певних стандартів мікроклімату. В таблиці 4.1 наведені параметри мікроклімату для таких приміщень

Таблиця 4.1 – Параметри мікроклімату для приміщень, де встановлені комп'ютери

Параметри мікроклімату	Літній період	Зимовий період
Температура повітря, °C	20 – 24	22 – 24
Відносна вологість, %	40 – 60	40 – 60
Швидкість руху повітря, м/с	0,1 – 0,2	0,1 – 0,2
Рівень шуму, дБ	до 50	До 50

4.1.4 Електромагнітне і іонізуюче випромінювання

Більшість науковців переконані, що короткочасна та тривала експозиція випромінюванням від екрана комп'ютера не шкодить здоров'ю працівників, які використовують персональні комп'ютери (ПК). Але повних даних про потенційну шкоду від такого випромінювання для людей, що працюють за комп'ютерами, ще немає, тому дослідження в цій сфері тривають.

Дозволені показники для неіонізуючого електромагнітного випромінювання від екрана ПК наведені в таблиці 4.2.

Таблиця 4.2 – Допустимі значення параметрів неіонізуючого електромагнітного випромінювання

Найменування параметра	Допустимі значення
Напруженість електричної складової електромагнітного поля на відстані 50см від поверхні монітора	10 В / м
Напруженість магнітної складової електромагнітного поля на відстані 50см від поверхні монітора	0,3 А / м
Напруженість електростатичного поля	20кВ / м

Зазвичай максимальна інтенсивність рентгенівського випромінювання на робочому місці користувача комп'ютера не перевищує 10 мкбер/год. Щодо ультрафіолетового та інфрачервоного випромінювання від екрану, їх інтенсивність знаходиться у діапазоні від 10 до 100МВт/м².

4.1.5 Емоційна психогігієна

У сфері ІТ охорона праці включає не лише фізичне, але й психічне здоров'я працівників. Фактори, що впливають на психіку працівників у галузі інформаційних технологій, є різноманітними та мають велике значення у створенні здорового робочого середовища.

До них належать:

- тривалість та інтенсивність робочого дня: довгі години роботи без відпочинку можуть призвести до перевтоми та стресу, впливаючи на психічне здоров'я;

- терміни виконання завдань і робочий тиск: нереалістичні терміни виконання завдань або надмірний робочий тиск можуть викликати стрес та тривогу;
- міжособистісні взаємодії: конфлікти в команді або нездорова робоча атмосфера можуть спричиняти емоційне вигорання;
- робота в умовах ізоляції або віддалена робота: відсутність соціальної взаємодії або підтримки може вплинути на почуття ізоляції та самотності;
- work-life баланс : нездатність знайти баланс між роботою та особистим життям може призвести до стресу та емоційного вигорання;
- значне розумове навантаження: постійна потреба в освоєнні нових технологій та адаптації до змін може бути стресовою;
- відсутність автономії або контролю над роботою: обмежений контроль над робочими процесами та відсутність автономії можуть викликати почуття безпорадності та фрустрації.

Для ефективної профілактики психоемоційних проблем, пов'язаних з роботою в ІТ-галузі, необхідно розробити комплексний підхід, який враховує різноманітність факторів, що впливають на психічне здоров'я працівників. Значущість цього підходу полягає в його спрямованості на зменшення стресорів у робочому середовищі та підвищення ресурсів для психічного відновлення.

Першочергово, акцентується увага на регулюванні тривалості робочого дня та інтенсивності роботи. Це включає розробку ефективних графіків роботи, що передбачають достатні перерви для відновлення, та уникнення перевантаження завданнями. Крім того, розглядається важливість балансу між роботою та особистим життям, який може бути підтриманий через гнучкі робочі години та можливість дистанційної роботи.

Ще одним важливим аспектом є оптимізація робочого середовища з точки зору ергономіки. Покращення умов роботи на робочому місці, включаючи забезпечення якісного обладнання та комфортних умов, сприятиме зниженню фізичного та психологічного дискомфорту.

Ключову роль відіграє також розвиток корпоративної культури, яка підтримує психологічне благополуччя. Це передбачає створення відкритого, підтримуючого співробітництва та комунікації середовища, заохочення взаємодопомоги між колегами, та реалізацію програм з управління стресом.

Для забезпечення довгострокового психічного благополуччя, розглядається імплементація регулярних тренінгів та навчальних програм, спрямованих на розвиток навичок управління стресом та підвищення особистісної стійкості. Також важливим є застосування систематичного моніторингу психічного стану працівників з використанням психометричних інструментів для раннього виявлення потенційних проблем.

Розуміння та належне врахування цих факторів є ключовим для забезпечення психічного благополуччя працівників у галузі ІТ, що, у свою чергу, позитивно впливає на їх продуктивність та загальне задоволення роботою.

4.2 Питання щодо безпеки в надзвичайних ситуаціях

В умовах швидкого розвитку інформаційних технологій та зростання залежності бізнес-процесів від ІТ-систем, питання безпеки в сфері ІТ набуває особливої актуальності. Розглядаючи безпеку праці в ІТ, особливу увагу необхідно приділити розробці та імплементації процедур реагування на надзвичайні ситуації, що включають технічні збої, кібератаки, витік конфіденційної інформації, фізичні загрози обладнанню та перебої в електропостачанні, а також інші аварійні стани, здатні порушити звичний хід робочих процесів.

Відповідальність за реалізацію ефективної системи безпеки лежить на ІТ-спеціалістах, керівництві компаній та інших зацікавлених сторонах. Необхідно забезпечити встановлення чітких правил та інструкцій, які регламентують дії персоналу у випадку виникнення надзвичайних ситуацій. Це включає визначення швидких комунікаційних каналів, розробку планів відновлення роботи систем, а також проведення регулярних тренінгів та інструктажів з охорони праці для всіх працівників.

Плани евакуації та реагування мають бути адаптовані до можливих ІТ-ризиків. Вони повинні включати дії для захисту життя та здоров'я працівників, а також процедури збереження даних та обладнання. Плани мають бути детальними та зрозумілими для кожного члена команди, з чіткими інструкціями щодо дій в кризових ситуаціях.

Регулярні тренінги з охорони праці, які охоплюють надзвичайні ситуації в ІТ, є необхідними для підготовки персоналу до ефективного реагування на інциденти. Навчальні програми повинні включати імітації надзвичайних ситуацій, навчання з використанням спеціалізованого обладнання та вивчення найкращих практик у сфері кібербезпеки. Систематичний аналіз ризиків є важливим для ідентифікації потенційних вразливостей системи та розробки відповідних заходів запобігання. Встановлення систем моніторингу, які можуть виявити ознаки надзвичайних ситуацій на ранніх стадіях, допомагає зменшити потенційні збитки та забезпечити оперативне реагування.

Після кожної надзвичайної ситуації необхідно проводити детальний аналіз її причин, ефективності дій персоналу та наявних процедур реагування. На основі цього аналізу слід вносити корективи у плани надзвичайних дій, поліпшувати процедури безпеки та організовувати додаткове навчання. Ефективне управління надзвичайними ситуаціями в ІТ-сфері вимагає всебічного підходу, який поєднує в собі як технічні, так і організаційні аспекти. Відповідальність за це лежить на всіх рівнях організації, починаючи з ІТ-фахівців і закінчуючи вищим керівництвом. Завдяки встановленню та дотриманню відповідних процедур можна мінімізувати ризики та забезпечити безперебійну роботу в умовах, що постійно змінюються.

ВИСНОВКИ

У ході дослідження було проведено детальний аналіз моделі Nexus, який включав оцінку її структурних компонентів, принципів взаємодії між командами та інтеграції робочих процесів. Виявлено, що модель сприяє підвищенню прозорості у взаємодії команд, що веде до зменшення часу на узгодження завдань та ефективнішого вирішення міжкомандних конфліктів. Також було відзначено поліпшення в управлінні залежностями завдань, оскільки Nexus забезпечує краще розуміння перетину робіт та сприяє своєчасному виявленню та вирішенню потенційних проблем.

Дослідження виявило, що впровадження Nexus має позитивний вплив на продуктивність команд. Було зафіксовано зниження часу на розробку завдань, що корелюється з покращенням у якості продукту та скороченням циклу його випуску. Зокрема, зменшення кількості помилок у коді свідчить про підвищення уваги до деталей та кращу координацію в тестуванні та якості. Проте, не дивлячись на значні переваги, дослідження також виявило ряд викликів, особливо у сфері інтеграції та управління різноманітними командними процесами. Виявлено, що деякі командні процеси, такі як спільне використання ресурсів, узгодження технічних рішень та вирішення конфліктів інтересів між командами, вимагають додаткової уваги та ретельного планування для досягнення оптимальної інтеграції.

В цілому, впровадження фреймворку Nexus виявилось ефективним у багатоаспектному покращенні управління проектами, сприяючи кращій координації та інтеграції між командами. Незважаючи на виявлені виклики, загальні результати дослідження свідчать про значний потенціал моделі для масштабування Agile практик. Подальше дослідження та адаптація моделі можуть призвести до ще більшої оптимізації процесів управління проектами, підвищення якості продукту та задоволеності клієнтів. Таким чином, висновки цієї роботи мають бути використані як основа для розробки стратегічних планів та покращення процесів роботи в проектно-орієнтованих компаніях.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Jeff Sutherland, “Scrum: the art of doing twice the work in half the time”, Random House Business, 2015, 256 p.
2. Esther Derby, “Agile retrospectives: making good teams great”, Pragmatic Bookshelf, 2006, 178 p.
3. Rubin Kenneth S., “Essential Scrum: a practical guide to the most popular agile process”, Addison-Wesley Professional, 2012, 496 p.
4. ISO/IEC TS 33074:2020 Information technology – Process assessment, (Process capability assessment model for service management).
5. ISO/IEC 33001:2015 - Information technology – Process assessment – Concepts and terminology.
6. Carsten Schneider, “Operational IT implementation management. Transition from IT projects to IT operations”, GRIN Verlag, 2015, 58 p.
7. Ginger Levin, “Program Management: A Life Cycle Approach”, Auerbach Publications, 2016, 584 p.
8. David Allen, James Fallows, “Getting Things Done: the art of stress-free productivity”, Penguin Books, 2015, 352 p.
9. David Kadavy, “Mind management, Not Time Management: Productivity When Creativity Matters”, Kadavy, Inc., 2020, 249 p.
10. Damon Zahariades, “To-Do List Formula: A Stress-Free Guide To Creating To-Do Lists That Work!”, CreateSpace Independent Publishing Platform, 2016, 150 p.
11. Richard E. Silverman, “Git Pocket Guide: A Working Introduction”, O'Reilly Media, 2013, 286 p.
12. Mayank Mittal, “The art of building great products: Combine your intuition with the best proven methodologies to build digital products everyone will love”, Mayank Mittal, 2022, 350 p.
13. Dr. Festus Elleh, “Project Management Waterfall-Agile-It-Data Science”, Authorhouse, 2023, 402 p.
14. David J. Anderson, “Kanban: successful evolutionary change for your technology business”, Blue Hole Press, 2010, 278 p.

15. Robert McCarthy, “Agile and Scrum: unlock the power of Agile project management, Lean Thinking, the Kanban process, and Scrum”, Author's Republic, 2020, 212 p.
16. Alex Campbell, “Kanban: A Complete Step-by-Step Guide to the Basic Concepts in Kanban (Agile Project management with Kanban)”, Author's Republic, 2018, 111 p.
17. Alex Campbell, “Agile project management with Scrum: how to become a great project manager”, Alex Published, 2022, 149 p.
18. Alex Campbell, “Agile and Scrum: Complete Guide. What is Agile and What is Scrum?”, Alex Published, 2020 , 300 p.
19. Dave Todaro, “The Epic Guide to Agile: More Business Value on a Predictable Schedule with Scrum”, R9 Publishing LLC, 2019, 518 p.
20. Murali Chemuturi, Thomas Cagley, “Mastering Software Project Management”, J. Ross Publishing , 2010, 408 p.
21. Chris Sims, Hillary Louise Johnson, “Scrum: a breathtakingly brief and Agile introduction”, Dymaxicon, 2012, 54 p.
22. Hassan Osman, “Hybrid work management: how to manage a hybrid team in the new workplace (A super-short book about how to analyze, plan, manage, and evaluate your team’s hybrid work arrangement)”, 2021, 64 p.
23. Nexus Guide: Посібник із застосування Nexus. – [Електронний ресурс].
– Режим доступу: <https://s3.amazonaws.com/scrumorg-website-prod/drupal/2017-03/NexusGuide-v1.1%20-%20Ukrainian%20nfv3.pdf>
24. Schwaber K., Sutherland J. Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game. Scrumguides.org. – [Електронний ресурс]. –
Режим доступу: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>
25. Andrew Stellman, “Learning Agile: Understanding Scrum, XP, Lean, and Kanban”, O'Reilly Media, 2013, 417 p.
26. Andrew Sammons, “Agile project management with Scrum + Kanban 2 In 1: The Last 2 Approaches You'll Need To Become More Productive And Meet Your Project Goals”, M & M Limitless Online Inc., 2021, 322 p.

27. ДСТУ 8604:2015 Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидючи. Загальні ергономічні вимоги. Чинний від 2017-07-01. Вид. офіц.
28. ДСТУ 7299:2013 Дизайн і ергономіка. Робоче місце оператора. Взаємне розташування елементів робочого місця. Загальні вимоги ергономіки. Чинний від 2014-01-01. Вид. офіц.
29. ДСТУ 8604:2015 Інтерфейс людина-машина. Основні принципи безпеки, маркування та ідентифікація. Принципи включення (ІЕС 60447:2004, IDT). Чинний від 2016-07-01. Вид. офіц.
30. ДБН В.2.5-28-2018 Природне і штучне освітлення. Чинний від 2019-03-01. Вид. офіц.
31. Dr. Sam Swapn Sinha, “Best PMP Exam Prep Guide 2023- 2024: Get PMP Certified in 2 weeks- study 2 hours a day before-after work”, Strategism, 2023, 202 p.
32. James Stockwell, “Agile and Scrum: Streamlining Project Management for Efficient Team Collaboration”, Anthony Pierce, 2023, 123 p.
33. Greg Caldwell, “Agile Project Management: The Complete Guide for Beginners to Scrum, Agile Project Management, and Software Development”, SD Publishing LLC, 2020, 172 p.
34. Darrell Rigby, “Doing Agile Right: Transformation Without Chaos”, Gildan Media, 2020, 265 p.
35. Whitney Soto, “SCRUM: Mastering Agile Project Management for Exceptional Results”, 2023, 54 p.
36. James Shore, Shane Warden, “The art of agile development”, Ascent Audio, 2021, 537 p.
37. Chuck Sherman, “Agile Project Management with Scrum”, 2023, 52 p.
38. Paul Vii, Randal Schaffer, “Scaled Agile Scrum: Nexus: How to Manage Multiple Scrum Teams”, Pashun Consulting Ltd., 2016.
39. Jimmy Mathew, “Scaled Scrum: Practice Questions with Explanations”, 2022.
40. Vinay Kanamarlapudi, “Mastering Agile Scrum”, 2022, 38 p.

41. Z. Lina and S. Dan, "Research on Combining Scrum with CMMI in Small and Medium Organizations," in 2012 International Conference on Computer Science and Electronics Engineering (ICCSEE), 2012, vol. 1, pp. 554–557.
42. Syed Imon Rizvi, "Scrum Guide (2020) - Mind-maps", Agile Supernova Press, 2021, 56 p.
43. Tobias Mayer, "The People's Scrum: Agile Ideas for Revolutionary Transformation", Dymaxicon, 2013, 170 p.
44. Alex Cowan, "Hypothesis-Driven Development", Cooke & McDouglas, 2023, 438 p.
45. Douglas Squirrel, Jeffrey Fredrick, "Agile Conversations: Transform Your Conversations, Transform Your Culture", IT Revolution Press, 2020, 224 p.
46. Gustavo Razzetti, "Remote Not Distant: Design a Company Culture That Will Help You Thrive in a Hybrid Workplace", Liberationist Press, 2022, 328 p.
47. Don McGreal, Ralph Jocham, "The Professional Product Owner: Leveraging Scrum as a Competitive Advantage", Addison-Wesley Professional, 2018, 384 p.
48. Ignacio Paz, "Nexus Scaled Scrum Certification Practice Tests and Questions", 2022 p., 145 p.
49. Jeff Patton, Martin Fowler, Peter Economy, Alan Cooper, "User Story Mapping: Discover the Whole Story, Build the Right Product", O'Reilly Media, 2014, 322 p.
50. Paulo Caroli, Fábio Aguiar, "How to Define, Prioritize and Refine User Stories using the Product Backlog Building Canvas", Editora Caroli, 2023, 164 p.

ДОДАТКИ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет імені Івана Пулюя (Україна)
Університет імені П'єра і Марії Кюрі (Франція)
Маріборський університет (Словенія)
Технічний університет у Кошице (Словаччина)
Вільнюський технічний університет ім. Гедимінаса (Литва)
Міжнародний університет цивільної авіації (Марокко)
Наукове товариство ім. Т.Шевченка

АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ

Збірник
тез доповідей

**XII Міжнародної науково-практичної
конференції молодих учених та студентів**
6-7 грудня 2023 року



УКРАЇНА
ТЕРНОПІЛЬ – 2023

*Матеріали XII Міжнародної науково-практичної конференції молодих учених та студентів
«АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ» – Тернопіль, 6-7 грудня 2023 року*

УДК 004.4

М. І. Зайченко, К.Б. Швирло

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

ДОСЛІДЖЕННЯ МОДЕЛЕЙ І МЕТОДІВ УПРАВЛІННЯ ІТ-ПРОЄКТОМ ДЛЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ПЛАНУВАННЯ ЧАСУ ТА УПРАВЛІННЯ ЗАДАЧАМИ

M. Zaichenko, K. Shvyrlo

RESEARCH OF MODELS AND METHODS OF IT PROJECT MANAGEMENT TO IMPROVE THE EFFICIENCY OF TIME PLANNING AND TASK MANAGEMENT

У світі, де інформаційні технології стрімко розвиваються, виникає неухильна необхідність у вдосконаленні методів управління ІТ-проектами. Це особливо актуально, враховуючи, що управління проектами не просто спрямовує технічні аспекти розробки, але й формує стратегічну основу для досягнення довгострокових бізнес-цілей. Ефективність у цій сфері може мати рішучий вплив на загальний успіх організації, від її конкурентоспроможності на ринку до здатності інноваційно реагувати на зміни у вимогах і технологіях.

З огляду на широке розмаїття методологій, від Agile до Waterfall, а також на популярність гнучких методів управління, як-от Scrum і Kanban, важливо розуміння того, як ці різні підходи можуть бути адаптовані та інтегровані для забезпечення оптимального управління великими проектними командами. Особлива увага в дослідженні приділена фреймворку Nexus, що є одним із варіантів масштабування Scrum, та його потенціалу у підвищенні продуктивності та ефективності управління великими ІТ-проектами. Аналіз базується на глибокому теоретичному дослідженні та практичному застосуванні у реальних умовах, що дозволяє виявити ключові параметри успіху та потенційні труднощі, пов'язані з впровадженням цих методологій у складних проектних середовищах.

Дослідження було здійснене в середовищі команди з 50 спеціалістів, що працювали над масштабним ІТ-проектом з розробки та підтримки веб-додатку. На початковому етапі проекту було застосовано Scrum-методологію, однак зі зростанням масштабу проекту та команди, виникла потреба в більш комплексній системі управління, що призвело до вибору Nexus як інструменту масштабування Scrum. Основною метою було вивчити, як перехід від Scrum до Nexus впливає на ефективність управління проектом, координацію командної роботи, та загальну продуктивність проекту. Приклад схеми взаємодії команди при такому переході відображено на рисунку 1.

Початковий аналіз проводився для оцінки ефективності Scrum-методології у даній команді, що включав оцінку часу на виконання задач, рівень задоволеності команди, та якість комунікації між членами команди. Після встановлення базових показників, було здійснено перехід до Nexus. Цей процес включав розробку інтеграційних механізмів, адаптацію ролей Scrum у контексті Nexus, та структурування багаторівневої системи спринтів. Протягом усього періоду впровадження Nexus здійснювався постійний моніторинг ключових показників ефективності: час виконання задач, рівень вирішення проблем, адаптивність до змін у проекті, та рівень задоволеності команди. Дані збиралися через опитування, інтерв'ю з членами команди, аналіз робочих документів та звітів про спринти. Особлива увага приділялася аналізу змін у координації командної роботи та ефективності виконання проектних завдань.

Матеріали XII Міжнародної науково-практичної конференції молодих учених та студентів
«АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ» – Тернопіль, 6-7 грудня 2023 року

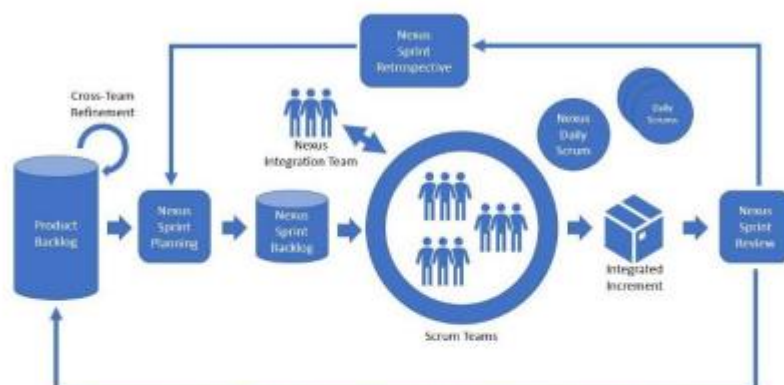


Рисунок 1. Схема роботи та взаємодії великої команди в разі застосуванні NEXUS-фреймворку для Scrum-команд

В таблиці 1 проілюстроване значне покращення в різних аспектах проєктного менеджменту, таких як скорочення часу на виконання задач, підвищення задоволеності команди, зниження кількості помилок, збільшення кількості успішно завершених спринтів, скорочення тривалості спринтів, підвищення відсотка задач, завершених вчасно, підвищення рівня участі команди у плануванні, збільшення кількості ініціатив з покращення процесів.

Таблиця 1. Порівняльний аналіз показників ефективності проєкту перед та після впровадження Nexus

Показник	Scrum (до впровадження Nexus)	Nexus (після впровадження)
Середній час виконання задач (дні)	5,5	3,5
Рівень задоволеності команди (шкала 1-10)	6,5	8,2
Середня кількість виявлених помилок на етапі тестування	28	17
Процент успішно завершених спринтів	70%	90%
Тривалість спринту	12	10
Відсоток задач, завершених вчасно	64%	86%
Рівень участі команди у плануванні (шкала 1-10)	5	8

Результати показали, що впровадження нової розробленої моделі на основі Nexus значно покращило управління великими командами та сприяло підвищенню загальної продуктивності проєкту. Було відзначено покращення у координації роботи між підкомандами, ефективнішу розподіл задач, та швидшу адаптацію до змін у проєкті.

Література

1. Nexus Guide: Посібник із застосування Nexus. – [Електронний ресурс]. – Режим доступу: <https://s3.amazonaws.com/scrumorg-website-prod/drupal/2017-03/NexusGuide-v1.1%20-%20Ukrainian%20nf3.pdf>
2. Schwaber K., Sutherland J. Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game. Scrumguides.org. – [Електронний ресурс]. – Режим доступу: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>

7.	М. І. Зайченко, К.Б. Швирло ДОСЛІДЖЕННЯ МОДЕЛЕЙ І МЕТОДІВ УПРАВЛІННЯ ІТ-ПРОЄКТОМ ДЛЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ПЛАНУВАННЯ ЧАСУ ТА УПРАВЛІННЯ ЗАДАЧАМИ	360
8.	А. І. Маліновський, В. Р. Медвідь РОЗРОБКА КОНТРОЛЕРА АВТОМАТИЗОВАНОЇ СИСТЕМИ КЕРУВАННЯ НА БАЗІ МІКРОПРОЦЕСОРА: ОСОБЛИВОСТІ ТА АСПЕКТИ	362
9.	О. О. Базиль, О. А. Шовкопляс ЕКСПЕРТНІ СИСТЕМИ ДЛЯ АНАЛІЗУ ДАНИХ ІНТЕРНЕТ-ДЖЕРЕЛ	364
10.	А. В. Сербенюк, О. С. Куроп'ятник ФОРМУВАННЯ МЕРЕЖЕВОГО ТРАФІКУ З ВИКОРИСТАННЯМ ФОРМАЛЬНИХ ГРАМАТИК	365
11.	Ю. Б. Апостол, П. І. Довгань, А. Т. Яворський, Р. Б. Трембач АВТОМАТИЗОВАНА СИСТЕМА УПРАВЛІННЯ РЕЗЕРВУАРНИМИ ПАРКАМИ	367
12.	І. І. Бородій ВИБІР ТЕХНОЛОГІЇ ПРОЄКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ ФОРМУВАННЯ АГРЕГОВАНИХ НАДВЕЛИКИХ МАСИВІВ ДАНИХ	368
13.	Л. В. Волинець, Н. А. Гарматюк, В. А. Готович ВЕЛИКІ ЗА ОБСЯГОМ НАБОРИ БІОМЕДИЧНИХ ДАНИХ ТА МАШИННЕ НАВЧАННЯ	370
14.	Р. П. Вархоляк ПІДВИЩЕННЯ ТОЧНОСТІ СИСТЕМ АВТОМАТИЗАЦІЇ ДЛЯ КОНТРОЛЮ ТИСКУ ТА ТЕМПЕРАТУРИ В ПРОМИСЛОВИХ УМОВАХ	372
15.	І. Р. Ралік ВИМОГИ ДО CRM-СИСТЕМИ ДЛЯ АВТОМАТИЗАЦІЇ ТОРГІВЛІ НА ПІДПРИЄМСТВАХ	373
16.	В. В. Никитюк, М. В. Галюк, М. В. Тененський АНАЛІЗ ВИКОРИСТАННЯ ТА ПОРІВНЯННЯ МІКРОСЕРВІСНОЇ І МОНОЛІТНОЇ АРХІТЕКТУР	374
17.	О. Войцьо МЕТОДИ ТА ЗАСОБИ ПОБУДОВИ КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ ВЕДЕННЯ ОБЛІКУ СКЛАДСЬКИХ ПРИМІЩЕНЬ НА ПІДПРИЄМСТВІ SEVN-UA	376
18.	І. Осійчук, М. Фриз ХАРАКТЕРИСТИКИ ПЛАНШЕТУ ДЛЯ ДІАГНОСТУВАННЯ ТРЕМОРУ	378
19.	П. С. Панчишин, М. І. Паламар ОБРОБКА ДАНИХ ПІСЛЯ СКАНУВАННЯ ПОВЕРХНІ LIDAR СИСТЕМОЮ ТА МОЖЛИВОСТІ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ	380
20.	М. А. Беляков, к.т.н., доц. В. Б. Савків, В. І. Гетманюк ІНТЕЛЕКТУАЛЬНА СИСТЕМА ДЛЯ АВТОМАТИЗАЦІЇ ПРОСТЕЖЕННЯ ТА УПРАВЛІННЯ ЖИТТЄВИМ ЦИКЛОМ ПРОДУКЦІЇ	382
21.	В. С. Морохович, М. І. Хом'як ВИКОРИСТАННЯ ДЕРЕВА РІШЕНЬ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧ КЛАСИФІКАЦІЇ НА ПРИКЛАДІ НАБОРУ ДАНИХ ПАСАЖИРІВ «ТИТАНІК»	383
22.	В. І. Козак; В. А. Готович	385