Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____Комп'ютерних наук_____
(повна назва)

Кафедра ______Інформаційних управляючих систем_____
(повна назва)

# КВАЛІФІКАЦІЙНА РОБОТА
## Пояснювальна записка

рівень вищої освіти _____другий (магістерський)_____

Дослідження моделей і методів процесу верифікації ІТ-продукту, які є інваріантними для різних методологій управління ІТ-проєктами

Виконав:
студент _2_ курсу, групи _УПГІТм-22-1____
Таранченко Софія Ігорівна_____
(прізвище, ім'я, по батькові)
Спеціальність ____122 Комп'ютерні_____
_____науки_____
(код і повна назва спеціальності)
Тип програми ____освітньо-наукова_____
(освітньо-професійна або освітньо-наукова)
Освітня програма __Управління проектами в галузі інформаційних технологій_____
( повна назва освітньої програми)

Керівник проф.каф. ІУС Максим ЄВЛАНОВ__
(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри _____ ___Костянтин ПЕТРОВ__
(підпис) (власне ім'я, прізвище)

2024 р.

Харківський національний університет радіоелектроніки

Факультет _____Комп'ютерних наук_____

Кафедра _____Інформаційних управляючих систем_____

Рівень вищої освіти _____другий (магістерський)_____

Спеціальність _____122 Комп'ютерні науки_____
(код і повна назва)

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

Освітня програма_Управління проектами в галузі інформаційних технологій
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«_01__»__квітня__ 20 24_ р.

# ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _Таранченко Софії Ігорівні_____
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження моделей і методів процесу верифікації ІТ-продукту, які є інваріантними для різних методологій управління ІТ-проєктами_____
затверджена наказом університету від _01 квітня_2024_ р. № _258 Cm_
2. Термін подання студентом роботи до екзаменаційної комісії 01 ___06_____ 2024 р.
3. Вихідні дані до роботи _PMBOK 7, ISO 9001:2015_____

4. Перелік питань, що потрібно опрацювати в роботі аналіз предметної галузі та постановка задачі, дослідження універсальних методів. стандартизація процесу тестування, розробка інфологічної моделі процесу тестування, апробація отриманих результатів

# КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів роботи | Терміни виконання етапів роботи | Примітка |
|---|---|---|---|
| 1 | Вивчення складнощів предметної галузі | 01.04.2024 – 06.04.2024 | Виконано |
| 2 | Дослідження існуючих підходів до тестування | 07.04.2024 – 12.04.2024 | Виконано |
| 3 | Формування обов’язкових етапів процесу | 13.04.2024 – 17.04.2024 | Виконано |
| 4 | Проведення декомпозиції етапів тестування | 18.04.2024 – 20.04.2024 | Виконано |
| 5 | Оцінка залученості людських ресурсів та витрати часового ресурсу для процесу | 21.04.2024 – 25.04.2024 | Виконано |
| 6 | Розробка табличної частини інфологічної моделі | 26.04.2024 – 29.04.2024 | Виконано |
| 7 | Розробка графічної частини інфологічної моделі | 30.05.2024 – 02.05.2024 | Виконано |
| 8 | Планування за допомогою методу критичного | 03.05.2024 – 07.05.2024 | Виконано |
| 9 | Планування за допомогою методу критичного | 07.05.2024 – 09.05.2024 | Виконано |
| 10 | Інтеграція інфологічної можеді в методи критичного шляху та критичного ланцюга | 10.05.2024 – 14.05.2024 | Виконано |
| 11 | Оформлення кваліфікаційної роботи | 15.05.2024 – 20.05.2024 | Виконано |
| 12 | Попередній захист кваліфікаційної роботи | 01.06.2024 | Виконано |
| 13 | Захист кваліфікаційної роботи | 04.06.2024 | |

Дата видачі завдання __01__ ___квітня__2024_ р.

Студент _____
(підпис)

Керівник роботи _____ проф.каф. ІУС Максим ЄВЛАНОВ
(підпис) (посада, власне ім’я, прізвище)

# РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 96 сторінок, 47 рисунків, 22 таблиці, 2 додатки, 7 джерел.

ВЕРИФІКАЦІЯ, ІНФОЛОГІЧНА МОДЕЛЬ, ІНФОРМАЦІЙНА СИСТЕМА, ПАТЕРН, УПРАВЛІННЯ ПРОЦЕСОМ ТЕСТУВАННЯ.

Об’єктом дослідження кваліфікаційної роботи є процес управління процесом тестування.

Предметом дослідження даної кваліфікаційної роботи є аналіз та порівняльний огляд моделей (і методів) процесу верифікації ІТ-продукту, що зберігають свою актуальність та ефективність незалежно від обраної методології управління ІТ-проектами.

Метою кваліфікаційної роботи є створення інфологічної моделі процесу тестування.

Методи дослідження – аналіз існуючих методів управління процесом тестування та актуальних відповідних досліджень.

Галузь застосування – проекти з «Визначеним» рівнем зрілості.

Результатом кваліфікаційної роботи є інфологічна модель для управління процесом тестування.

# ABSTRACT

Attestation work: 96 pages, 47 pictures, 22 tables, 7 sources, 2 applications.

VERIFICATION, INFORMATION SYSTEM, INFORMATION MODEL, PATTERN, TESTING PROCESS MANAGEMENT.

The object of the research of the qualification work is the process of managing the testing process.

The subject of this qualification work is the analysis and comparative review of models (and methods) of the IT product verification process, which retain their relevance and effectiveness regardless of the selected IT project management methodology.

The purpose of the qualification work is to create an informational model of the testing process.

Research methods – analysis of existing methods of testing process management and current relevant research.

Field of application – projects with "Defined" level of maturity.

The result of the qualification work is an infographic model for managing the testing process.

# ЗМІСТ

Вступ…….9

1 Аналіз предметної галузі та постановка задачі…….10

1.1 Тестування в процесі управління проектами……...10

1.2 Особливості керування процесом тестування. Переваги та недоліки існуючих підходів……....18

1.3 Опис результатів аналізу……...22

1.4 Постановка задачі……..26

2 Дослідження універсальних методів. Стандартизація процесу тестування…...28

2.1 Дослідження обов’язкових етапів процесу тестування……..28

2.1.1 Створення тестових сценаріїв……..28

2.1.2 Підготовка тестового середовища……...31

2.2 Інтеграція універсального підходу до тестування в рамках третього рівня зрілості проекту……33

2.3 Висновки до другого розділу……45

3 Розробка інфологічної моделі процесу тестування……..47

3.1 Розробка табличної частини інфологічної моделі процесу тестування…….47

3.2 Розробка графічної частини інфологічної моделі процесу тестування…….57

3.3 Висновки до третього розділу……...59

4 Апробація отриманих результатів……60

4.1 Загальні відомості, щодо проекту……60

4.2 Планування процесу тестування стандартними методиками…….63

4.2.1 Планування за допомогою методу критичного шляху…….63

4.2.2 Планування за допомогою методу критичного ланцюга…….65

4.3 Планування процесу тестування за допомогою розробленої інфологічної моделі……67

4.4 Висновки до четвертого розділу……...70

Висновки…….72

Перелік джерел посилання……74

Додаток А. Графічний матеріал кваліфікаційної роботи……..75

Додаток Б. Графічне представлення інфологічної моделі процесу тестування..87

# СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

TMap – Test management approach – підхід до управління тестами;

QA engineer – Quality Assurance engineer – фахівець із забезпечення якості (або тестувальник).

# ВСТУП

Інформаційні технології є ключовим фактором для розвитку різних галузей. Зі зростанням важливості ІТ-продуктів зростає і складність їх розроблення та управління. Питання якості ІТ-продуктів та ефективності їх верифікації стає все більш важливим. Існують різні підходи до управління процесом тестування, які підходять для одних методологій, але не для інших. Відсутність єдиного патерну для управління тестуванням ускладнює процес, вимагаючи додаткового часу для побудови плану.

Для побудови універсального підходу до управління процесом тестування потрібна інфологічна модель для управління процесом тестування.

Об’єктом дослідження кваліфікаційної роботи є процес управління процесом тестування.

Предметом дослідження даної кваліфікаційної роботи є аналіз та порівняльний огляд моделей (і методів) процесу верифікації ІТ-продукту, що зберігають свою актуальність та ефективність незалежно від обраної методології управління ІТ-проектами.

Метою кваліфікаційної роботи є інфологічної моделі для управління тестуванням.

Завданнями кваліфікаційної роботи є:

– аналіз існуючих підходів до управління тестуванням;

– аналіз універсальних методів та стандартизація процесу тестування;

– розробка інфологічної моделі процесу тестування;

– апробація отримани результатів.

На основі результатів побудови інфологічної моделі буде створення патерну для процесу управління процесом тестування.

Пояснювальна записка до кваліфікаційної роботи виконано згідно з основними вимогами методичних вказівок та стандарту ДСТУ 3008:2015 [1].

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

## 1.1 Тестування в процесі управління проектами

Управління проектом є невід'ємною частиною розробки будь-якого продукту. В свою чергу, якісне управління проектом забезпечує ефективну координацію, комунікацію та керування всіма етапами розробки продукту. Таким чином, управління проектом є складним процесом, оскільки потребує комплексного підходу та підготовки.

Життєвий цикл системи складається з наступних чотирьох груп процесів:

– процес узгодження. Даний процес визначає механізми для узгодження вимог, обміну інформацією та управління ризиками між різними сторонами, що мають інтереси або відповідальність у визначенні, розробці, поставці, експлуатації або підтримці системи. Дана група процесів включає в себе встановлення контрактних зобов'язань, управління ризиками, обмін інформацією та звітністю між сторонами, а також управління змінами та конфігурацією;

– процес організаційного управління. Даний процес описує діяльність з управління проектом та ресурсами, які спрямовані на підтримку виконання проекту з розробки системи. Ці процеси включають в себе планування проекту та ресурсів, управління комунікаціями та звітністю, управління стейкхолдерами, а також управління якістю та відстеженням метрик проекту;

– процес технічного управління. Даний процес орієнтований на управління технічними аспектами проекту, такими як технічне планування, визначення вимог, архітектура, розробка, валідація та верифікація. Ці процеси включають у себе розроблення технічних стратегій та планування, управління технічними ресурсами, керування конфігурацією та змінами, а також забезпечення дотримання стандартів та регуляторних вимог;

– технічні процеси. Даний процес описує конкретні дії та завдання, пов'язані з інженерією систем, а саме аналіз вимог, проектування системи, реалізацію, інтеграцію, тестування та підтримку. Ці процеси визначають технічні аспекти життєвого циклу системи, забезпечуючи ефективну розробку та функціонування системи згідно з встановленими вимогами та стандартами.

Розглянемо процес верифікації, який є обов’язковою складовою життєвого циклу системи, і даний процес взаємодіє з такими групами процесів:

– група технічного управління;

– група технічних процесів.

Розглянемо групу технічного управління, яка складається з наступних процесів:

– процес планування проекту (6.3.1);

– процес управління та оцінювання проекту (6.3.2);

– процес управління прийняття рішень (6.3.3);

– процес управління ризиками (6.3.4);

– процес управляння конфігурацією (6.3.5);

– процес управління інформацією (6.3.6);

– процес вимірювання (6.3.7);

– процес забезпечення якості (6.3.8).

Група технічного управління не містить окремо виділеного процесу верифікації (6.4.9), хоча верифікаціє взаємодіє з процесами групи технічного управління. Розглянемо яку діяльність та мету містить процес верифікації (6.4.9):

а) підготовка до верифікації;

1) визначення обсягу верифікації та відповідних дій для верифікації;

2) визначення обмежень, які потенційно обмежують можливості дій із верифікації;

3) визначення відповідних методів верифікації або методи та відповідні критерії для кожної дії верифікації;

4) визначення стратегії верифікації;

4) визначення обмежень системи щодо стратегії верифікації, які потрібно вмістити у вимоги до системи, архітектури чи проектування;

5) визначення і планування необхідних систем забезпечення або послуг, необхідних для підтримання верифікації;

6) отримання або придбання доступу до систем забезпечення чи послуг, використовуваних для підтримки верифікації.

б) виконання верифікації;

1) визначення процедури верифікації, кожна з яких підтримує одну чи набір дій з верифікації;

2) виконання процедури верифікації.

в) управління результатами верифікації;

1) запис результатів верифікації та будь яких аномалій, що виникли;

2) фіксування оперативних інцидентів та проблем й відстежування їхнього вирішення;

3) отримання згоди зацікавлених сторін на те, що елемент системи або система відповідає заданим вимогам;

4) підтримання простежуваності верифікованих елементів системи;

5) забезпечення ключових інформаційних елементів, які вибрані як початкові умови.

Визначимо яким чином взаємодіє процес верифікації з процесами з групи технічного управління, результат представлено в таблиці 1.1.

Таблиця 1.1 – Взаємодія процесу верифікації з процесами технічного управління

| Процес з групи процесів технічного управління | Діяльність процесу верифікації | Мета | Результат |
|---|---|---|---|
| Процес планування проекту | Визначення стратегії верифікації | Аналіз потреб і обмежень для визначення ефективної стратегії верифікації. | Розробка стратегії верифікації, яка включає в себе планування методів та дій для верифікації процесів технічного управління. |
| | Визначення і планування необхідних систем забезпечення або послуг для підтримки верифікації | Виявлення необхідних систем та послуг для підтримки верифікації. | Визначення та розробка плану для забезпечення необхідних інфраструктурних систем та послуг для верифікації. |
| Процес управління та оцінювання проекту | Запис результатів верифікації та будь-яких аномалій, що виникли | Фіксація результатів верифікації та ідентифікація виниклих аномалій. | Запис результатів, що стосуються стану проекту та виявлених аномалій, які впливають на подальші кроки в управлінні проектом. |
| | Фіксація оперативних інцидентів та проблем і відстеження їхнього вирішення | Виявлення та реєстрація інцидентів та проблем, що виникли під час верифікації. | Фіксація та вирішення виявлених проблем з метою поліпшення проектного процесу. |
| Процес управління прийняття рішень | Отримують згоду зацікавлених сторін на те, що елемент системи або система відповідає заданим вимогам | Представлення результатів верифікації для забезпечення згоди зацікавлених сторін. | Згода стейкхолдерів на те, що система або її елементи відповідають вимогам, що впливає на процес прийняття рішень. |

Продовження таблиці 1.1

| Процес з групи процесів технічного управління | Діяльність процесу верифікації | Мета | Результат |
|---|---|---|---|
| Процес управління ризиками | Визначення обсягу верифікації та відповідних дій іх верифікації | Включення ризиків, що виникають з невідповідності вимог або дефектів, ідентифікованих під час верифікації, у план управління ризиками. | Оновлений план управління ризиками, який враховує ризики, що виникають під час верифікації, і визначає відповідні дії для їх управління. |
| Процес управління прийняття рішень | Отримують згоду зацікавлених сторін на те, що елемент системи або система відповідає заданим вимогам | Представлення результатів верифікації для забезпечення згоди зацікавлених сторін. | Згода стейкхолдерів на те, що система або її елементи відповідають вимогам, що впливає на процес прийняття рішень. |
| Процес управління ризиками | Визначення обсягу верифікації та відповідних дій іх верифікації | Включення ризиків, що виникають з невідповідності вимог або дефектів, ідентифікованих під час верифікації, у план управління ризиками. | Оновлений план управління ризиками, який враховує ризики, що виникають під час верифікації, і визначає відповідні дії для їх управління. |
| Процес управління конфігацією | Забезпечення ключових інформаційних елементів, які вибрані як початкові умови | Використання результатів верифікації для перевірки відповідності ключових елементів системи встановленим стандартам управління конфігурацією. | Підтвердження відповідності ключових елементів системи встановленим стандартам ідентифікації та управління конфігурацією. |

Продовження таблиці 1.1

| Процес з групи процесів технічного управління | Діяльність процесу верифікації | Мета | Результат |
|---|---|---|---|
| Процес управління інформацією | Запис результатів верифікації та будь-яких аномалій, що виникли | Запис результатів верифікації та ідентифікація аномалій для подальшого використання. | Документовані результати верифікації та виявлені аномалії, які зберігаються та розповсюджуються для подальшого використання в управлінні інформацією. |
| Процес вимірювання | Запис результатів верифікації та будь-яких аномалій, що виникли | Використання результатів верифікації як частини метрик вимірювання продуктивності та якості процесів управління та розробки. | Запис результатів верифікації та аномалій для використання у вимірюванні продуктивності та якості процесів. |
| | Забезпечення ключових інформаційних елементів, які вибрані як початкові умови | Використання результатів верифікації для визначення вихідних даних для процесу вимірювання, таких як визначення початкових станів елементів системи для подальшого порівняння. | Визначення початкових станів елементів системи для використання у процесі вимірювання |
| Процес забезпечення якості | Отримують згоду зацікавлених сторін на те, що елемент системи або система відповідає заданим вимогам | Визначення початкових станів елементів системи для використання у процесі вимірювання | Отримання згоди зацікавлених сторін на відповідність системи або її елементів встановленим вимогам. |

Кінець таблиці 1.1

| Процес з групи процесів технічного управління | Діяльність процесу верифікації | Мета | Результат |
|---|---|---|---|
| | Запис результатів верифікації та будь-яких аномалій, що виникли | Використання результатів верифікації, разом з будь-якими виявленими аномаліями, для аналізу якості продукту та процесів управління і розробки для подальшого вдосконалення. | Документування результатів верифікації та аномалій для аналізу якості та вдосконалення процесів. |

Розглянемо технічні процеси, дана група процесів містить окремо виділений процес верифікації, розглянутий процес взаємодії представлений в таблиці 1.2.

Таблиця 1.2 – Взаємодія процесу верифікації з технічними процесами

| Технічні процеси | Взаємодія процесу верифікації з процесами технічного управління | Результатом взаємодії є |
|---|---|---|
| Процеси аналізу бізнесу або місії (6.4.1) | Верифікація визначає, чи відповідають встановлені вимоги і потреби тим функціям і характеристикам, які були зазначені на початку процесу аналізу та визначення вимог. | Результатом взаємодії процесу верифікації з цими процесами є підтвердження того, що вимоги і потреби, визначені на ранніх етапах, правильно і повністю відображені у процесі верифікації, що допомагає уникнути ризику невідповідності системи очікуванням у майбутньому. |
| Визначення потреб і вимог зацікавлених сторін (6.4.2) | | |
| Визначення вимог до системи (6.4.3) | | |

Продовження таблиці 1.2

| Технічні процеси | Взаємодія процесу верифікації з процесами технічного управління | Результатом взаємодії є |
|---|---|---|
| Процес визначення архітектури (6.4.4) | Верифікація перевіряє, чи відповідає розроблена архітектура та проект системи вимогам і специфікаціям. | Результатом взаємодії процесу верифікації з цими процесами підтвердження того, що розроблена архітектура та проект системи відповідають встановленим вимогам і специфікаціям, забезпечуючи відповідність між дизайном та очікуваним функціоналом. |
| Процес проектування (6.4.5) | | |
| Процес системного аналізу (6.4.6) | Верифікація перевіряє правильність і повноту аналізу та висновків, щоб переконатися, що вони відповідають вимогам і цілям системи. | Результатом взаємодії процесу верифікації з цим процесом є перевірка правильності аналізу та висновків, щоб гарантувати, що вони відповідають вимогам і цілям системи, що допомагає уникнути неправильних стратегічних виборів у подальшому проектуванні та розробці. |
| Процес імплементування (6.4.7) | Верифікація перевіряє, чи виконані реалізація та інтеграція системи відповідно до специфікацій і вимог. | Результатом взаємодії процесу верифікації з цими процесами є перевірка, що реалізація та інтеграція системи відповідають специфікаціям і вимогам, забезпечуючи коректну роботу окремих модулів та їх спільне функціонування в рамках системи. |
| Процес інтеграції (6.4.8) | | |
| Процес передавання (6.4.10) | Верифікація може включати перевірку, чи готова система до передачі, чи відповідає вимогам користувачів, чи ефективно експлуатується, чи правильно виконується технічне обслуговування і чи відповідає стандартам для списання | Результатом взаємодії процесу верифікації з цими процесами є перевірка відповідності готової системи вимогам користувачів, її ефективної експлуатації, правильної підтримки та оновлення, а також відповідності стандартам для подальшого списання або відновлення. |
| Процес валідації (6.4.11) | | |
| Процес експлуатації (6.4.12) | | |
| Процес технічного обслуговування (6.4.13) | | |

Кінець таблиці 1.2

| Технічні процеси | Взаємодія процесу верифікації з процесами технічного управління | Результатом взаємодії є |
|---|---|---|
| Процес списання (6.4.14) | | |

## 1.2 Особливості керування процесом тестування. Переваги та недоліки існуючих підходів

Тестування системи відіграє ключову роль у забезпеченні якості та надійності програмного продукту. В залежності від обраної методології управління проектом з розробки цієї системи, обирається підхід до управління тестуванням. При цьому варто враховувати, що існують певні універсальні підходи до управління тестуванням, які можна застосовувати незалежно від конкретної методології управління проектом.

Ці універсальні підходи базуються на основних принципах ефективного та систематичного тестування, а також на визнаних стандартах та кращих практиках у сфері тестування програмного забезпечення. Вони включають в себе ретельне планування тестування, визначення стратегії та областей покриття тестування, а також ефективне використання автоматизованих і ручних тестів. Крім того, ці підходи передбачають систематичний аналіз результатів тестування та вчасне виявлення та виправлення дефектів.

Такий універсальний підхід дозволяє забезпечити якість програмного продукту незалежно від конкретної методології розробки, що використовується. Це особливо важливо в умовах постійних змін у розробці програмного забезпечення та різноманітних вимог клієнтів.

Розглянемо основні універсальні підходи до управління процесом тестування, дані підходи підійдуть до більшості або принаймні до декількох методологій управління проектом:

– TMap. Дана методологія управління тестуванням надає детальний підхід до всіх аспектів тестування. Підхід TMap може бути застосований як для традиційних, так і для Agile-проектів. Він надає методологію для керування процесом тестування, яка може бути адаптована до потреб конкретного проекту, незалежно від обраної методології розробки;

– Agile Testing. Даний підхід до тестування, який акцентується на швидкому та гнучкому тестуванні на кожному етапі розробки, що відповідає принципам Agile-методологій. Цей підхід найбільш підходить для Agile-проектів, де тестування інтегроване в кожному ітераційному циклі розробки. Він може бути успішно використаний в будь-якій Agile-методології, такій як Scrum чи Kanban;

– DevOps Testing. Даний підхід до тестування включає тестування на всіх етапах життєвого циклу розробки та впровадження програмного забезпечення, зокрема використанням автоматизованих тестів. Цей підхід може бути успішно застосований у середовищі будь-якої методології розробки, включаючи Agile, Waterfall, Spiral, та інші, оскільки він акцентується на автоматизації, швидкому виявленні та виправленні помилок;

– Risk-Based Testing. Даний підхід до тестування, при якому тестування відбувається відповідно до рівня ризику для продукту. Цей підхід може бути успішно застосованим у будь-якій методології розробки, де визначення та управління ризиками є важливим аспектом. Він може бути корисним як для традиційних, так і для Agile-проектів.

Кожен підхід до управління тестуванням має обов’язкові етапи, які виконуються в незалежності від загальної методології управління проектом, розглянемо ці етапи в таблиці 1.3.

Таблиця 1.3 – Опис підходу до управління тестуванням

| Підхід до управління тестуванням | Методології в рамках яких можна застосувати даний підхід до управління тестуванням | Обов’язкові етапи в управлінні тестуванням |
|---|---|---|
| TMap (Test Management Approach) | – водоспадна модель;<br>– v-модель; | а) визначення тестових вимог та стратегії тестування;<br>б) розробка тест-плану та планування тестування;<br>в) виконання тестування;<br>г) аналіз результатів та звітність;<br>д) управління ризиками тестування. |
| Agile Testing | – scrum;<br>– kanban;<br>– v-модель;<br>– прототипування | а) визначення вимог та планування тестування на кожній ітерації з врахуванням тестових вимог ;<br>б) виконання тестування (у вигляді автоматизованих тестів та ручного тестування);<br>в) підготовка звітів та ретроспектив для аналізу результатів;<br>г) постійна адаптація стратегії тестування на основі змін у вимогах або у власній команді. |
| DevOps Testing | – scrum;<br>– kanban;<br>– v-модель;<br>– прототипування | а) інтеграція тестування в кожен етап життєвого циклу розробки (Continuous Testing);<br>б) автоматизація тестування та впровадження Continuous Integration/Continuous Deployment (CI/CD) практик;<br>в) моніторинг інфраструктури та додатків для виявлення помилок у реальному часі. |

Кінець таблиці 1.3

| Підхід до управління тестуванням | Методології в рамках яких можна застосувати даний підхід до управління тестуванням | Обов’язкові етапи в управлінні тестуванням |
|---|---|---|
| Risk-Based Testing | – водоспадна модель;<br>– scrum;<br>– kanban;<br>– спіральна модель | а) оцінка ризиків продукту або проекту;<br>б) планування тестування, зосередженого на областях з найвищим ризиком;<br>в) виконання тестів у відповідності до рівня прийнятого ризику;<br>г) постійна оцінка та оновлення стратегії тестування на основі нових ризиків або змін у вимогах. |

Незважаючи на те, що характеристики та обов’язкові процеси при управління тестування в цих п’яти методологія управління тестування різні, можна виділити деякі процеси, які є повторюваними хоча б у 2 і більше підходах до управління тестуванням:

– визначення тестових вимог. Цей етап перший і один з найважливіших у процесі управління тестуванням. Даний етап є загальним для всіх методологій, оскільки від визначення тестових вимог залежить успішність та ефективність подальших тестових процесів. Під час цього етапу визначаються очікувані результати тестування, функціональні вимоги до продукту, вимоги до продуктивності та безпеки, а також критерії прийняття;

– планування тестування. Цей етап є важливим у кожному підході до управління тестуванням. Під час цього етапу розробляється стратегія тестування, визначаються методики, інструменти та ресурси, які будуть використані для проведення тестів. Також складається графік тестування, розподіляються завдання між учасниками команди тестування та іншими стейкхолдерами;

– виконання тестування. Цей етап включає запуск тестів, збір результатів, аналіз отриманих даних та реєстрацію виявлених дефектів. Цей етап

відбувається в усіх підходах до управління тестуванням і є вирішальним для перевірки роботи програмного продукту на відповідність вимогам та стандартам якості;

– аналіз результатів та підготовка звітності. Цей етап є необхідним у кожній методології управління тестуванням. Після завершення тестування необхідно аналізувати отримані результати, виявлені дефекти та відхилення від визначених критеріїв. На основі цих даних готується звітність, яка може містити інформацію про стан якості продукту, рекомендації щодо виправлення дефектів та покращення процесу тестування.

## 1.3 Опис результатів аналізу

Питанням побудови універсального підходу до тестування в рамках різних методологій управління проектами, а саме Waterfall, V-model, Agile і т.д., було розглянуто в рамках дослідження «A Test Process for all Lifecycles» авторства Ian Londesbrough [4], в результаті чого було визначено 6 універсальних етапів процесу тестування, а саме:

– етап ідеї. На цьому етапі тестування та забезпечення якості інтегровані в саму концепцію проекту. Ідея проекту піддається «статичному тестуванню» для визначення її на можливість тестування та потенційних слабких місць, що дозволяє згодом досягти реалізації та втілення ідеї в життя;

– визначення вимог. На цьому етапі аналізуються сценарії, які можуть бути протестовані протягом життєвого циклу проекту. Важливо, щоб вимоги були чітко визначені та могли бути протестованим, щоб забезпечити можливість їх реалізації;

– специфікація та дизайн. На цьому етапі «статичне тестування» використовується для забезпечення того, щоб дизайн був зрозумілим, повним та

максимально простим. Це дозволяє підтримати бізнес-сценарії, визначені на попередніх етапах;

– подубова. Раннє залучення команди тестування на початкових етапах з метою «статичного тестування» ідей, вимог та дизайну, а також планування та проектування тестування, гарантує, що «динамічне тестування» може розпочатися якнайшвидше;

– впровадження. Система або процес, готові до розгортання, проходять через кілька середовищ, включаючи системне тестування, приймальне тестування та пред-продакшен середовище, де і відбувається «динамічне тестування»;

– реалізація цінності. На цьому фінальному етапі визначається момент зупинки тестування, коли система або процес мають достатній рівень якості для досягнення обіцяних переваг, визначених у бізнес-кейсі. Рішення про завершення тестування базується на ідентифікації та усуненні дефектів, демонстрації відповідності вимогам та оцінці якості на основі встановлених вимог.

Порівняємо етапи універсального підходу до тестування з даного дослідження з визначеними етапами, які найчастіше повторюються, дані етапи можуть мати різні назви в які закладених однаковий сенс. Результат повіряння представлено в таблиці 1.4.

Таблиця 1.4 – Порівняння обов’язкових етапів процесу тестування з дослідження та визначених в підрозділі 1.2

| Універсальні етапи процесу тестування з дослідження «A Test Process for all Lifecycles» авторства Ian Londesbrough | Визначені універсальні етапи процесу тестування в розділі 1.2 |
|---|---|
| Етап ідеї | – |
| Визначення вимог | Визначення тестових вимог |
| Специфікація та дизайн | – |

Кінець таблиці 1.4

| | |
|---|---|
| Побудова | Планування тестування;<br>Виконання тестування |
| Впровадження | – |
| Реалізація цінності | – |
| – | Аналіз результатів та підготовка звітності |

Можна побачити, що із 4 визначених універсальних етапів в підрозділі 1.2, 3 етапи мають альтернативу в дослідженні «A Test Process for all Lifecycles» авторства Ian Londesbrough, а 1 етап не має відповідної альтернативи.

Також слід розглянути підхід до формування життєвого циклу тестування програмного забезпечення (STLC) [3], який містить 6 основних етапів в процесі тестування, а саме:

– аналіз вимог. На цій стадії аналітики тестувальники занурюються в деталі вимог проекту, щоб визначити, які з них можна тестувати. Це включає спілкування з різними учасниками проекту для глибшого розуміння задач. Вимоги розглядаються як функціональні, так і нефункціональні;

– стратегія тестування. На цьому етапі відповідальний за якість проекту розробляє комплексний план тестування, оцінюючи необхідні зусилля, вартість та визначаючи ресурси. Визначаються також обмеження, середовище для тестування та розклад;

– створення тестових сценаріїв. Цей етап включає дизайн, перевірку та коригування тестових сценаріїв. Після розробки плану тестування команда визначає тестові дані та розробляє тестові сценарії, готуючись до безпосереднього тестування;

– підготовка тестового середовища. На цій стадії налаштовуються умови для тестування, які включають програмне та апаратне забезпечення. Це важливий аспект, який може відбуватися одночасно з розробкою тестових сценаріїв;

– виконання тестів. Під час цього етапу відбувається безпосереднє тестування згідно з розробленими планами та сценаріями. Команда здійснює тестування, документує результати та фіксує помилки для подальшого виправлення розробниками;

– завершення тестування. Остання стадія включає аналіз здійсненого тестування, збір тестових метрик та результатів. Обговорюються здобуті дані для визначення подальших кроків удосконалення процесу та підготовки до майбутніх тестових циклів.

Розглянемо порівняння даного підходу з визначеними етапами процесу тестування в підрозділі 1.2, результат порівняння представлено в таблиці 1.5.

Таблиця 1.5 – Порівняння обов’язкових етапів процесу тестування з STLC та визначених в підрозділі 1.2

| Універсальні етапи процесу тестування STLC | Визначені універсальні етапи процесу тестування в підрозділі 1.2 |
|---|---|
| Аналіз вимог | Визначення тестових вимог |
| Стратегія тестування | Планування тестування |
| Створення тестових сценаріїв | – |
| Підготовка тестового середовища | – |
| Виконання тестів | Виконання тестування |
| Завершення тестування | Аналіз результатів та підготовка звітності |

Даний підхід побудови життєвого циклу містить всі 4 етапи визначені в підрозділі 1.2, але не всі етапи життєвого циклу тестування програмного забезпечення (STLC) мають альтернативні етапи знайдені в підрозділі 1.2.

Обидва підходи, представлені в дослідженні «A Test Process for all Lifecycles» та в підході до формування життєвого циклу тестування програмного забезпечення (STLC), мають перетини у визначенні ключових етапів процесу тестування в підрозділі 1.2. Це демонструє наявність спільних принципів та

практик у тестуванні програмного забезпечення, що застосовуються у різних методологіях управління проектами.

## 1.4 Постановка задачі

Аналіз дослідження «A Test Process for all Lifecycles» [4] і підходу до формування життєвого циклу тестування програмного забезпечення (STLC) [3] виявив існування значних перетинів у визначенні ключових етапів процесу тестування, що були розглянуті в підрозділі 1.2.

Основна цінність такої універсальності полягає в можливості застосування цих принципів тестування в широкому спектрі проектів розробки програмного забезпечення, незалежно від обраних методологій управління. Це забезпечує фундамент для створення ефективних стратегій тестування, які можуть бути адаптовані та оптимізовані для задоволення унікальних вимог та цілей кожного проекту.

Водночас, хоча перетини між обома підходами свідчать про спільні базові елементи, важливо враховувати, що повного співпадіння між усіма етапами немає. Це відкриває можливість для додаткової індивідуалізації та доповнення процесу тестування, що дозволяє компаніям та командам розробників вносити власний досвід та знання для підвищення ефективності та якості кінцевого продукту.

Таке додавання вимагає спеціального представлення знань, процесів та етапів робіт процесу управління тестуванням, одним із способів такого представлення є подання здобутих знань для багаторазового використання в вигляді найпростіших інфологічних моделей, які описують досліджуваний процес управління тестуванням у вигляді сукупності таблиць, графіків, текстів і т.д. Важливість інваріантних моделей верифікації для різних методологій

управління IT-проєктами. Гнучкість та адаптивність таких моделей забезпечують стабільність та ефективність верифікаційного процесу на різних етапах розробки. Автоматизація та універсальність інструментів тестування виявляються ключовими для успішної інтеграції змін в вимогах та стратегіях управління проектами. Ці інфологічні моделі є критичним елементом для досягнення високої якості IT-продукту в змінному та конкурентному середовищі [5].

Предметом дослідження виступають універсальні етапи та методології процесу тестування та верифікації IT-продуктів, що можуть бути застосованими у різних підходах управління IT-проектами.

Метою дослідження є побудова інваріативного підходу до тестування для різних методологій управління IT-проєктами.

Для досягнення поставленої мети пропонується вирішити такі задачі дослідження:

– аналіз предметної галузі та постановка задачі;

– дослідження універсальних методів. Стандартизація процесу тестування;

– застосування стандартизованого підхожу до тестування;

– перевірка результатів використання стандартизованого підходу до тестування.

# 2 ДОСЛІДЖЕННЯ УНІВЕРСАЛЬНИХ МЕТОДІВ. СТАНДАРТИЗАЦІЯ ПРОЦЕСУ ТЕСТУВАННЯ

## 2.1 Дослідження обов’язкових етапів процесу тестування

У розділі 1 було проведено аналіз п'яти ключових підходів до тестування, що дозволило визначити чотири основні етапи процесу тестування. Ці етапи були порівняні з результатами дослідження "A Test Process for all Lifecycles" авторства Іана Лондесбро. Даний аналіз показав, що один із чотирьох визначених етапів відсутній у дослідженні "A Test Process for all Lifecycles". Крім того, порівнюючи ці етапи з процесом тестування STLC, можна відзначити, що всі чотири визначені етапи були відображені у процесі STLC, однак два етапи STLC не мали відповідних альтернатив у визначених етапах. Відсутність визначених альтернативних підходів у процесі тестування за стандартом STLC ставить під сумнів необхідність цих етапів у стандартизованому підході до тестування, детальніше розглянемо кожен з цих 2 етапів в підрозділі 2.1.1 та 2.1.2.

### 2.1.1 Створення тестових сценаріїв

Як було зазначено в підрозділі 1.3 етап «Створення тестових сценаріїв» включає в себе дизайн, перевірку та корегування тестових сценаріїв. Даний етап забезпечує:

– якість та структурованість процесу тестування. Створення тестових сценаріїв дозволяє чітко визначити послідовність дій та очікувані результати для тестування певної функціональності програмного забезпечення;

– підвищення якості тестування. Тестові сценарії дозволяють провести повноцінну оцінку функціональності програмного забезпечення та виявити якомога більше проблем;

– підготовка до процесу автоматизації тестування. Тестові сценарії можуть слугувати основою для розробки автоматизованих тестів, що сприятиме підвищенню ефективності тестування та економії часу для його проведення;

– основа для регресійного тестування. Тестові сценарії можуть бути використані повторно для перевірки того, чи не виникли нові помилки в програмному забезпеченні в результаті внесених змін.

Зважаючи на важливість етапу "Створення тестових сценаріїв" і потенційну значну користь від цього етапу, цей процес можна інтегрувати в до двох основних етапів, визначених в підрозділі 1.2, а саме:

– визначення тестових вимог;

– планування тестування.

Для інтеграція етап «Створення тестових випадків» розділимо на наступні підетапи:

а) аналіз бізнес-вимог та функціональних специфікацій. Даний підетап включатиме;

1) детальний аналіз вимог щодо функціональності продукту;

2) визначення обсягу тестування;

3) визначення основних функціональних областей продукту.

б) визначення тестових сценаріїв. Даний підетап включатиме розробку тестових сценаріїв основуючись на функціональних вимогах та областях тестування;

в) включення тестових сценаріїв у стратегію тестування. Даний підетап включатиме;

1) пріоритизація та визначення послідовності виконання тестових сценаріїв;

2) розподіл ресурсів та визачення відповідальних за створення та виконання тестових сценаріїв.

Підетапи етапу «Створення тестових випадків» розподілені між двома етапами представлено в таблиці 2.1

Таблиця 2.1 – Розподіл підетапів етапу «Створення тестових випадків» між етапами «Визначення тестових вимог» та «Планування тестування»

| Обов'язковий етап визначений в підрозділі 1.2 | Підетап етапу «Створення тестових випадків» |
|---|---|
| Визначення тестових вимог | Аналіз бізнес-вимог та функціональних специфікацій |
| | Визначення тестових сценаріїв |
| Планування тестування | Включення тестових сценаріїв у стратегію тестування |

Таким чином оновлений перелік дій етапів «Визначення тестових вимог» та «Планування тестування» представлений в таблиці 2.2.

Таблиця 2.2 – Оновлений перелік дій етапів «Визначення тестових вимог» та «Планування тестування»

| Назва етапу | Оновлений перелік дій |
|---|---|
| Визначення тестових вимог | – визначення очікуваних результатів тестування;<br>– визначення функціональних вимог до продукту;<br>– визначення вимог до продуктивності та безпеки;<br>– визначення критеріїв прийняття;<br>– аналіз бізнес-вимог та функціональних специфікації;<br>– визначення тестових сценраїв. |
| Планування тестування | – розробка стратегії тестування;<br>– визначення методик, інструментів та ресурсів, які будуть використаними для проведення тестування;<br>– формування графіку тестування;<br>– розподіл завдання між учасниками команди тестування та іншими стейкхолдерами;<br>– включення тестових сценаріїв у стратегію тестування |

Інтеграція етапу «Створення тестових сценаріїв» як частина визначених обов'язкових етапів з підрозділу 1.2 забезпечить більш систематичний та комплексний підхід до створення тестових сценаріїв, що сприятиме знаходженню та виявленню проблем на більш ранніх стадіям розробки. Також

результатом даної інтеграції буде забезпечення більшої уваги до деталей та відповідність між тестовими сценаріями та стратегією тестування.

### 2.1.2 Підготовка тестового середовища

Етап «Підготовка тестового середовища» включає в себе налаштування програмного та апаратного забезпечення. Даний етап забезпечує:

– ефективність тестування. Готове тестове середовище дозволяє уникнути втрати часу на вирішення технічних проблем пов’язаних із налаштування оточення і тим самим дозволяючи зосередитись на процесі тестування;

– готовність та стабільність тестування. Правильно підготовлене тестове середовище гарантує, що всі необхідні складові системи будуть доступні вчасно для виконання тестів і уникає непередбачених проблем, таких як конфлікти між різними компонентами або неправильне налаштування.

Зважаючи на важливість етапу «Підготовка тестового середовища» і вплив на процес тестування в цілому, цей процес слід інтегрувати до таки основних етапів, визначених в підрозділі 1.2:

– планування тестування;

– виконання тестування.

Розділимо етап «Підготовка тестового середовища» на підетапи, а саме:

а) аналіз вимог до тестового середовища. Даний підетап включатиме визначення потреб та вимог для тестового середовища;

б) налаштування програмного забезпечення. Даний підетап включатиме встановлення та налаштування необхідного програмного забезпечення для виконання тестів;

в) налаштування апаратного забезпечення. Даний підетап включатиме підготовку та налаштування апаратного забезпечення для тестування;

г) перевірка тестового середовища. Даний підетап включатиме перевірку налаштувать тестового середовища для забезпечення впевненості правильності його конфігурації та готовності для виконання тестових сценаріїв.

Підетапи етапу «Налаштування тестового середовища» розподілені між двома етапами представлено в таблиці 2.2.

Таблиця 2.3 – Розподіл підетапів етапу «Налаштування тестового середовища» між етапами «Планування тестування» та «Виконання тестування»

| Планування тестування | Виконання тестування |
|---|---|
| Аналіз вимог до тестового середовища | Налаштування програмного забезпечення |
| Перевірка тестового середовища | Налаштування апаратного забезпечення |

Таким чином оновлений перелік дій етапів «Планування тестування» та «Виконання тестування» представлений в таблиці 2.2.

Таблиця 2.4 – Оновлений перелік дій етапів «Визначення тестових вимог» та «Планування тестування»

| Назва етапу | Оновлений перелік дій |
|---|---|
| Планування тестування | – розробка стратегії тестування;<br>– визначення методик, інструментів та ресурсів, які будуть використаними для проведення тестування;<br>– формування графіку тестування;<br>– розподіл завдання між учасниками команди тестування та іншими стейкхолдерами;<br>– включення тестових сценаріїв у стратегію тестування;<br>– аналіз вимог до тестового середовища;<br>– перевірка тестового середовища. |
| Виконання тестування | – запуск тестів;<br>– збір та аналіз результатів;<br>– реєстрація багів;<br>– налаштування тестового середовища;<br>– налаштування апаратного середовища. |

Інтеграція етапу «Створення тестових сценаріїв» як частина визначених обов’язкових етапів з підрозділу 1.2 підвищить ефективність тестування за рахунок уникання непередбачуваних проблем із програмним або апаратним

середовищем забезпечуючи стабільність тестового середовища. Дані інтеграція сприятимуть ефективному запуску тестів та надійним результатам.

## 2.2 Інтеграція універсального підходу до тестування в рамках третього рівня зрілості проекту

Під час створення та впровадження програмного продукту існує необхідність у визначенні рівня зрілості проекту для оцінки його стану. На даний момент існують 5 рівнів зрілості проекту в саме:

а) початковий. Даний рівень характеризується неструктурованістю та хаотичність, де основний акцент робиться на індивідуальних зусиллях, а успіх проекту залежить від ініціативи окремих працівників. На цьому етапі відсутні стандарти та процедури, і проекти часто виконуються без чіткого планування;

б) повторюваний. Даний рівень управління проектами стає більш організованим, завдяки базовим процесам управління, встановленим для відстеження витрат, графіка виконання та функціональності, процес розробки стає більш стабільним. Повторюються успіхи попередніх проектів, і встановлюються процедури, які допомагають досягти цілей;

в) визначений. Даний рівень управління проектами здійснює основний акцент на стандартизацію та документування процесів. Процеси розробки програмного забезпечення детально описані, стандартизовані та інтегровані у всі аспекти діяльності організації. Встановлені ключові процеси забезпечують збалансований підхід до розробки програмного забезпечення;

г) керований. На даному рівні управління проектами збираються детальні дані про якість процесу та продукту. Керування проектом та якістю продукту стає більш систематичним і прозорим завдяки квантовому аналізу та контролю. Управління якістю стає центральною метою на цьому етапі;

д) оптимізація. Даний рівень характеризується постійним вдосконаленням. На цьому рівні фокус зосереджується на інноваціях та постійному розвитку. Впроваджуються нові ідеї та технології для покращення процесу розробки програмного забезпечення.

Як можна побачити «Початковий» та «повторюваний» рівні не зважаючи на певні відмінності все одно характеризуються певною хаотичністю процесів, в той час як рівні «Визначений», «Керований» та «Оптимізація» характеризуються стандартизацією тих чи інших процесів та побудовою універсальних підходів до різних етапів.

Подальшу побудову та інтеграцію стандартизованого підходу до тестування здійснимо для третього рівня зрілості проекту – «Визначений».

Загальний процес тестування є множиною, дана множина розбивається на підмножини визначені в підрозділі 1.2 було, а саме:

– визначення тестових вимог;

– планування тестування;

– виконання тестування;

– аналіз результатів та підготовка звітності.

Проведемо процес декомпозиції підмножин для знаходження найменших неподільних процесів для встановлення ієрархії відношень належностей, результат представлено в таблиці 2.5.

Таблиця 2.5 – Декомпозиція процесів тестування до найменших неподільних процесів

| Загальний етап | Декомпозиція загальний етапів | Декомпозиція підетапів | Додаткова декомплозиція |
|---|---|---|---|
| МТ1 Визначення тестових вимог | МТ1.1 Аналіз вимог | МТ1.1.1 Збір вимог | МТ1.1.1.1 Організація зустрічей з представниками бізнесу для обговорення вимог |
| | | | МТ1.1.1.2 Проведення аналізу документів, що містять вимоги |

Продовження таблиці 2.5

| Загальний етап | Декомпозиція загальний етапів | Декомпозиція підетапів | Додаткова декомплозиція |
|---|---|---|---|
| | | | МТ1.1.1.3 Проведення інтерв'ю зі зацікавленими сторонами для збору додаткової інформації |
| | | | МТ1.1.1.4 Формулювання списку вимог та їхнє документування |
| | | МТ1.1.2 Аналіз вимог на визначення тестових сценаріїв | МТ1.1.2.1 Ідентифікація ключових функцій та можливих сценаріїв використання |
| | | | МТ1.1.2.2 Визначення пріоритетів та важливості кожного сценарію для тестування |
| | | | МТ1.1.2.3 Розроблення стратегії тестування на основі вимог та визначених сценаріїв |
| | МТ1.2 Визначення критеріїв прийняття | МТ1.2.1 Формулюванн я критеріїв прийняття | – |
| | | МТ1.2.1 Перевірка та узгодження критеріїв прийняття зі зацікавленими сторонами | – |
| | МТ1.3 Створення тестових сценаріїв | МТ1.3.1 Визначення сценаріїв для покриття вимог | – |
| | | МТ1.3.2 Розробка тестових сценаріїв | – |
| | | МТ1.3.3 Перевірка тестових сценаріїв | – |
| | МТ1.4 Узгодження із зацікавленими сторонами | МТ1.4.1 Проведення консультацій та обговорення зауважень із зацікавленими сторонами | – |

Продовження таблиці 2.5

| Загальний етап | Декомпозиція загальний етапів | Декомпозиція підетапів | Додаткова декомплозиція |
|---|---|---|---|
| | | МТ1.4.2 Внесення змін в тестові вимоги за результатами узгодження | – |
| | МТ1.5 Створення тестових даних | МТ1.5.1 Визначення необхідних тестових даних | – |
| | | МТ1.5.2 Завантаження та підготовка тестових даних | – |
| МТ2. Плану вання тестування | МТ2.1 Розробка стратегії тестування | МТ2.1.1 Визначення цілей тестування та стратегії їх досягнення | МТ2.1.1.1 Визначення основних цілей, які потрібно досягти в процесі тестування |
| | | | МТ2.1.1.2 Формулювання конкретних завдань та вимог, які входять у визначені цілі |
| | | МТ2.1.2 Вибір методів та підходів до тестування | – |
| | | МТ2.1.3 Встановлення пріоритетів у проведенні тестів | – |
| | МТ2.2 Визначення інструментів та ресурсів | МТ2.2.1 Підбір необхідних інструментів для проведення тестування | – |
| | | МТ2.2.2 Оцінка доступності та необхідності ресурсів для виконання тестів | – |
| | МТ2.3 Формування графіку тестування | МТ2.3.1 Визначення термінів та пріоритетів проведення окремих етапів тестування | – |
| | | МТ2.3.1 Створення графіка роботи команди тестування | – |

Продовження таблиці 2.5

| Загальний етап | Декомпозиція загальний етапів | Декомпозиція підетапів | Додаткова декомплозиція |
|---|---|---|---|
| | МТ2.4 Розподіл завдань | МТ2.4.1 Визначення обсягу робіт та розподіл завдань між учасниками команди тестування та іншими зацікавленими сторонами | – |
| | | МТ2.4.2 Встановлення відповідальності за виконання кожного завдання | – |
| | МТ2.5 Аналіз вимог до тестового середовища | МТ2.5.1 Визначення вимог до конфігурації тестового середовища | – |
| | | МТ2.5.2 Перевірка наявності необхідних ресурсів та програмного забезпечення | – |
| | МТ2.6 Перевірка тестового середовища | МТ2.6.1 Проведення тестів для перевірки функціональності тестового середовища | – |
| | | МТ2.6.1 Вирішення будь-яких виявлених проблем або недоліків | – |
| МТ3 Виконання тестування | МТ3.1 Запуск тестів | МТ3.1.1 Підготовка до запуску тестових сценаріїв | – |
| | | МТ3.1.2 Запуск автоматизованих та ручних тестів | МТ3.1.2.1 Запуск автоматизований тестів |
| | | | МТ3.1.2.2 Проходження ручних тестів |
| | МТ3.2 Збір та аналіз результатів | МТ3.2.1 Збір результатів виконання тестів | – |
| | | МТ3.2.2 Аналіз отриманих результатів на предмет виявлення дефектів | – |
| | МТ3.3 Реєстрація багів | МТ3.3.1 Фіксація виявлених дефектів або проблем в системі | – |

Продовження таблиці 2.5

| Загальний етап | Декомпозиція загальний етапів | Декомпозиція підетапів | Додаткова декомплозиція |
|---|---|---|---|
| | | МТ3.3.2 Описання деталей та відображення результатів тестування в системі управління багами | – |
| | МТ3.4 Налаштування тестового середовища | МТ3.4.1 Оновлення програмного забезпечення тестового середовища | – |
| | | МТ3.4.2 Конфігурація середовища для подальших тестів | – |
| | МТ3.5 Налаштування апаратного середовища | МТ3.5.1 Підготовка апаратного забезпечення для виконання тестів | – |
| | | МТ3.5.1 Перевірка та налаштування робочих параметрів апаратури перед тестуванням | – |
| МТ4 Аналіз результатів та підготовка звітності | МТ4.1 Визначення моменту зупинки тестування | МТ4.1.1 Оцінка метрик якості | МТ4.1.1.1 Аналіз метрик відповідності до визначених критеріїв. |
| | | | МТ4.1.1.2 Порівняння результатів з попередніми вимогами |
| | | МТ4.1.2 Порівняння отриманих результатів з попередніми критеріями завершення | МТ4.1.2.1 Визначення критеріїв завершення тестування |
| | | | МТ4.1.2.2 Порівняння фактичних результатів зі встановленими критеріями |
| | | | МТ4.1.2.3 Прийняття рішення про продовження або завершення тестування |
| | | МТ4.1.3 Вирішення, чи досягнутий достатній рівень якості для завершення тестування | – |

Кінець таблиці 2.5

| Загальний етап | Декомпозиція загальний етапів | Декомпозиція підетапів | Додаткова декомплозиція |
|---|---|---|---|
| | МТ4.2 Ідентифікація та усунення дефектів | МТ4.2.1 Аналіз результатів тестування для виявлення дефектів | – |
| | | МТ4.2.2 Класифікація та призначення пріоритетів дефектам | МТ4.2.2.1 Визначення важливості та впливу дефектів на функціональність продукту |
| | | | МТ4.2.2.2 Налаштування пріоритетів для подальшого усунення |
| | | МТ4.2.3 Усунення виявлених дефектів розробниками | МТ4.2.3.1 Передача дефектів розробникам для виправлення |
| | | | МТ4.2.3.2 Перевірка виправлених дефектів |
| | МТ4.3 Демонстрація відповідності вимогам та оцінка якості на основі встановлених вимог | МТ4.3.1 Порівняння результатів тестування з вимогами до продукту | МТ4.3.1.1 Перевірка відповідності функціональності продукту встановленим вимогам |
| | | | МТ4.3.1.2 Виявлення потенційних неспівпадінь |
| | | МТ4.3.2 Виявлення невідповідностей та їх вирішення | – |
| | | МТ4.3.3 Оцінка загальної якості продукту | – |
| | | МТ4.3.4 Підготовка звітності та презентація результатів зацікавленим сторонам | МТ4.3.4.1 Створення звіту з результатами тестування |
| | | | МТ4.3.4.2 Підготовка до презентації |

Проаналізувавши декомпозовані етапи було визначено ті етапи, які потенційно можуть зайняти найменше часу, а саме:

– МТ1.1.1.1 Організація зустрічей з представниками бізнесу для обговорення вимог;

– МТ1.1.1.3 Проведення інтерв'ю зі зацікавленими сторонами для збору додаткової інформації;

– MT1.1.2.2 Визначення пріоритетів та важливості кожного сценарію для тестування;

– MT1.4.2 Внесення змін в тестові вимоги за результатами узгодження.

Найбільш ймовірно етап «MT1.1.1.1 Організація зустрічей з представниками бізнесу для обговорення вимог» займе найменше часу, адже цей етап не потребує додаткового аналізу або детального вивчення вимог, оскільки його ціль лише організація зустрічей із стейкхолдерами для обговорення вимог. Отже етап «MT1.1.1.1 Організація зустрічей з представниками бізнесу для обговорення вимог» буде виступати за 1 відносного часового ресурсу.

Таблиця 2.6 – Визначення людського та часового ресурсу для декомпозованних етапів процесу тестування

| Етап | Підетап | Декомпозиція підетапу | Людський ресурс (QA engineer(s)) | Часовий ресурс (на 1 людський ресурс) |
|---|---|---|---|---|
| MT1 Визначення тестових вимог | | | ≥1 | 57 |
| MT1 Визначення тестових вимог | MT1.1 Аналіз вимог | | ≥1 | 17 |
| | MT1.1.1 Збір вимог | | ≥1 | 14 |
| | | MT1.1.1.1 Організація зустрічей з представниками бізнесу для обговорення вимог | 1 | 1 |
| | | MT1.1.1.2 Проведення аналізу документів, що містять вимоги | ≥1 | 5 |
| | | MT1.1.1.3 Проведення інтерв'ю зі зацікавленими сторонами для збору додаткової інформації | 1 | 5 |
| | | MT1.1.1.4 Формулювання списку вимог та їхнє документування | ≥1 | 3 |

Продовження таблиці 2.6

| Етап | Підетап | Декомпозиція підетапу | Людський ресурс (QA engineer(s)) | Часовий ресурс (на 1 людський ресурс) |
|---|---|---|---|---|
| | МТ1.1.2 Аналіз вимог на визначення тестових сценаріїв | | ≥1 | 13 |
| | | МТ1.1.2.1 Ідентифікація ключових функцій та можливих сценаріїв використання | ≥1 | 5 |
| | | МТ1.1.2.2 Визначення пріоритетів та важливості кожного сценарію для тестування | ≥1 | 3 |
| | | МТ1.1.2.3 Розроблення стратегії тестування на основі вимог та визначених сценаріїв | 1 | 5 |
| МТ1 Визначення тестових вимог | МТ1.2 Визначення критеріїв прийняття | | 1 | 8 |
| | МТ1.2.1 Формулювання критеріїв прийняття | | 1 | 5 |
| | МТ1.2.1 Перевірка та узгодження критеріїв прийняття зі зацікавленими сторонами | | 1 | 3 |
| МТ1 Визначення тестових вимог | МТ1.3 Створення тестових сценаріїв | | ≥1 | 11 |
| | МТ1.3.1 Визначення сценаріїв для покриття вимог | | ≥1 | 3 |
| | МТ1.3.2 Розробка тестових сценаріїв | | ≥1 | 5 |
| | МТ1.3.3 Перевірка тестових сценаріїв | | 1 | 3 |
| МТ1 Визначення тестових вимог | МТ1.4 Узгодження із зацікавленими сторонами | | 1 | 5 |
| | МТ1.4.1 Проведення консультацій та обговорення зауважень із зацікавленими сторонами | | 1 | 3 |
| | МТ1.4.2 Внесення змін в тестові вимоги за результатами узгодження | | 1 | 2 |
| МТ1 Визначення тестових вимог | МТ1.5 Розробка тестових даних | | 1 | 6 |

Продовження таблиці 2.6

| Етап | Підетап | | Декомпозиція підетапу | Людський ресурс (QA engineer(s)) |
|---|---|---|---|---|
| | МТ1.5.1 Визначення необхідних тестових даних | | 1 | 3 |
| | МТ1.5.2 Завантаження та підготовка тестових даних | | 1 | 3 |
| МТ2. Планування тестування | | | ≥1 | 49 |
| МТ2. Планування тестування | МТ2.1 Розробка стратегії тестування | | ≥1 | 16 |
| | МТ2.1.1 Визначення цілей тестування та стратегії їх досягнення | | 1 | 10 |
| | | МТ2.1.1.1 Визначення основних цілей, які потрібно досягти в процесі тестування | 1 | 5 |
| | | МТ2.1.1.2 Формулювання конкретних завдань та вимог, які входять у визначені цілі | 1 | 5 |
| | МТ2.1.2 Вибір методів та підходів до тестування | | 1 | 3 |
| | МТ2.1.3 Встановлення пріоритетів у проведенні тестів | | ≥1 | 3 |
| МТ2. Планування тестування | МТ2.2 Визначення інструментів та ресурсів | | ≥1 | 6 |
| | МТ2.2.1 Підбір необхідних інструментів для проведення тестування | | ≥1 | 3 |
| | МТ2.2.2 Оцінка доступності та необхідності ресурсів для виконання тестів | | ≥1 | 3 |
| МТ2. Планування тестування | МТ2.3 Формування графіку тестування | | ≥1 | 10 |
| | МТ2.3.1 Визначення термінів та пріоритетів проведення окремих етапів тестування | | ≥1 | 5 |
| | МТ2.3.1 Створення графіка роботи команди тестування | | 1 | 5 |
| МТ2. Планування тестування | МТ2.4 Розподіл завдань | | 1 | 8 |
| | МТ2.4.1 Визначення обсягу робіт та розподіл завдань між учасниками команди тестування та іншими зацікавленими сторонами | | 1 | 5 |
| | МТ2.4.2 Встановлення відповідальності за виконання кожного завдання | | 1 | 3 |
| МТ2. Планування тестування | МТ2.5 Аналіз вимог до тестового середовища | | 1 | 5 |

Продовження таблиці 2.6

| Етап | Підетап | | Декомпозиція підетапу | Людський ресурс (QA engineer(s)) |
|---|---|---|---|---|
| | МТ2.5.1 Визначення вимог до конфігурації тестового середовища | | 1 | 3 |
| | МТ2.5.2 Перевірка наявності необхідних ресурсів та програмного забезпечення | | 1 | 2 |
| МТ2. Плануванння тестування | МТ2.6 Перевірка тестового середовища | | 1 | 4 |
| | МТ2.6.1 Проведення тестів для перевірки функціональності тестового середовища | | 1 | 2 |
| | МТ2.6.2 Вирішення будь-яких виявлених проблем або недоліків | | 1 | 2 |
| МТ3 Виконання тестування | | | 1+ | 31 |
| МТ3 Виконання тестування | МТ3.1 Запуск тестів | | | 11 |
| | МТ3.1.1 Підготовка до запуску тестових сценаріїв | | 1 | 3 |
| | МТ3.1.2 Запуск автоматизованих та ручних тестів | | 1 | 8 |
| | | МТ3.1.2.1 Запуск автоматизований тестів | 1 | 3 |
| | | МТ3.1.2.2 Проходження ручних тестів | 1 | 5 |
| МТ3 Виконання тестування | МТ3.2 Збір та аналіз результатів | | ≥1 | 5 |
| | | МТ3.2.1 Збір результатів виконання тестів | ≥1 | 3 |
| | | МТ3.2.2 Аналіз отриманих результатів на предмет виявлення дефектів | ≥1 | 2 |
| МТ3 Виконання тестування | МТ3.3 Реєстрація багів | | 1 | 7 |
| | МТ3.3.1 Фіксація виявлених дефектів або проблем в системі | | 1 | 5 |
| | МТ3.3.2 Описання деталей та відображення результатів тестування в системі управління багами | | 1 | 2 |
| МТ3 Виконання тестування | МТ3.4 Налаштування тестового середовища | | 1 | 4 |
| | МТ3.4.1 Оновлення програмного забезпечення тестового середовища | | 1 | 2 |
| | МТ3.4.2 Конфігурація середовища для подальших тестів | | 1 | 2 |
| МТ3 Виконання тестування | МТ3.5 Налаштування апаратного середовища | | 1 | 4 |

Продовження таблиці 2.6

| Етап | Підетап | | Декомпозиція підетапу | Людський ресурс (QA engineer(s)) |
|---|---|---|---|---|
| | МТ3.5.1 Підготовка апаратного забезпечення для виконання тестів | | 1 | 2 |
| | МТ3.5.1 Перевірка та налаштування робочих параметрів апаратури перед тестуванням | | 1 | 2 |
| МТ4 Аналіз результатів та підготовка звітності | | | ≥1 | 49 |
| МТ4 Аналіз результатів та підготовка звітності | МТ4.1 Визначення моменту зупинки тестування | | | 16 |
| | МТ4.1.1 Оцінка метрик якості | | 1 | 5 |
| | | МТ4.1.1.1 Аналіз метрик відповідності до визначених критеріїв. | 1 | 3 |
| | | МТ4.1.1.2 Порівняння результатів з попередніми вимогами | 1 | 2 |
| | МТ4.1.2 Порівняння отриманих результатів з попередніми критеріями завершення | | ≥1 | 8 |
| | | МТ4.1.2.1 Визначення критеріїв завершення тестування | 1 | 3 |
| | | МТ4.1.2.2 Порівняння фактичних результатів зі встановленими критеріями | 1 | 3 |
| | | МТ4.1.2.3 Прийняття рішення про продовження або завершення тестування | ≥1 | 2 |
| | МТ4.1.3 Вирішення, чи досягнутий достатній рівень якості для завершення тестування | | ≥1 | 3 |
| МТ4 Аналіз результатів та підготовка звітності | МТ4.2 Ідентифікація та усунення дефектів | | ≥1 | 16 |
| | МТ4.2.1 Аналіз результатів тестування для виявлення дефектів | | 1 | 5 |
| | МТ4.2.2 Класифікація та призначення пріоритетів дефектам | | ≥1 | 6 |
| | | МТ4.2.2.1 Визначення важливості та впливу дефектів на функціональність продукту | ≥1 | 3 |
| | | МТ4.2.2.2 Налаштування пріоритетів для подальшого усунення | 1 | 3 |

Кінець таблиці 2.6

| Етап | Підетап | | Декомпозиція підетапу | Людський ресурс (QA engineer(s)) |
|---|---|---|---|---|
| | МТ4.2.3 Усунення виявлених дефектів розробниками | | 1 | 5 |
| | | МТ4.2.3.1 Передача дефектів розробникам для виправлення | 1 | 2 |
| | | МТ4.2.3.2 Перевірка виправлених дефектів | 1 | 3 |
| МТ4 Аналіз результатів та підготовка звітності | МТ4.3 Демонстрація відповідності вимогам та оцінка якості на основі встановлених вимог | | ≥1 | 16 |
| | МТ4.3.1 Порівняння результатів тестування з вимогами до продукту | | ≥1 | 6 |
| | | МТ4.3.1.1 Перевірка відповідності функціональності продукту встановленим вимогам | 1 | 3 |
| | | МТ4.3.1.2 Виявлення потенційних неспівпадінь | ≥1 | 3 |
| | МТ4.3.2 Виявлення невідповідностей та їх вирішення | | 1 | 3 |
| | МТ4.3.3 Оцінка загальної якості продукту | | 1 | 3 |
| | МТ4.3.4 Підготовка звітності та презентація результатів зацікавленим сторонам | | 1 | 5 |
| | | МТ4.3.4.1 Створення звіту з результатами тестування | 1 | 3 |
| | | МТ4.3.4.2 Підготовка до презентації. Проведення демонстрації | 1 | 2 |

## 2.3 Висновки до другого розділу

Отже, 2 підетапи "Створення тестових випадків" та "Підготовка тестового середовища" було декомпозовано та інтегровано в чотири основних етапи процесу управління тестуванням та як результат оновлено перелік дій цих етапів.

За результатами оцінювання декомпозованних етапів процесу тестування відносно підетапу «МТ1.1.1.1 Організація зустрічей з представниками бізнесу для обговорення вимог» який було визначено за 1 часового ресурсу, кожен із чотирьох основних етапів тестування в загальному оцінюється наступним чином:

– МТ1 Визначення тестових вимог – 57 відносних часових одиниць;

– МТ2. Планування тестування – 49 відносних часових одиниць;

– МТ3 Виконання тестування – 31 відносних часових одиниць;

– МТ4 Аналіз результатів та підготовка звітності – 49 відносних часових одиниць.

Слід враховувати, що розрахунок здійснювався на 1 людський ресурс, тобто для 1 тестувальника. Отриманий результат слугуватими патерном для подального використання та може бути адаптованим від відповідний проект.

Результати отримані в цьому розділі опубліковані у [6].

# 3 РОЗРОБКА ІНФОЛОГІЧНОЇ МОДЕЛІ ПРОЦЕСУ ТЕСТУВАННЯ

## 3.1 Розробка табличної частини інфологічної моделі процесу тестування

У підрозділі 2.4 було розроблено підхід до процесу тестування в якому було визначено чотири основних етапи тестування, які в свою чергу було декомпозовано до найменших підетапів. Після проведення декомпозиції та визначення найменшого етапу, а саме «MT1.1.1.1 Організація зустрічей з представниками бізнесу для обговорення вимог» даний підетап було визначено за 1 часового ресурсу і на основі цього проведено подальшу оцінку витрат часового та людського ресурсу для виконання всіх інших підетапів.

Отриманий підхід може слугувати шаблоном для організації процесу тестування, проте варто зазначити, що немає абсолютно однакових проектів, тож при плануванні процесу тестування слід враховувати особливості проекту, а саме обрану методологію управління проектом, графік виконання проекту в цілому та кожної окремо задачі, кількість ресурсів і т.д., відповідно до цих особливостей шаблон організації процесу тестування слід адаптувати до кожного проекту. Таким чином деякі етапи підходу можуть бути модифіковані або виключені за необхідності. Дане рішення дозволяє забезпечити максимальну адаптивність тестового процесу до конкретних потреб кожного проекту, забезпечуючи ефективне виконання завдань тестування відповідно до вимог проекту і методології

Розглянемо залежності між підетапами для етапу «MT1 Визначення тестових вимог» складається з наступних підетапів», які представлені в таблиці 3.1.

Таблиця 3.1 – Встановлення залежностей для «МТ1 Визначення тестових вимог» складається з наступних підетапів»

| 1 підетап | 2 підетап | 3 підетап | Залежності | Математичне представлення |
|---|---|---|---|---|
| МТ1 Визначення тестових вимог | | | | |
| МТ1.1 Аналіз вимог | | | | |
| | МТ1.1.1 Збір вимог | | – | – |
| | | МТ1.1.1.1 Організація зустрічей з представникам и бізнесу для обговорення вимог | – | – |
| | | МТ1.1.1.2 Проведення аналізу документів, що містять вимоги | – | – |
| | | МТ1.1.1.3 Проведення інтерв'ю зі зацікавленими сторонами для збору додаткової інформації | – МТ1.1.1.1 Організація зустрічей з представникам и бізнесу для обговорення вимог;<br>– МТ1.1.1.2 Проведення аналізу документів, що містять вимоги | МТ1.1.1.1 → МТ 1.1.1.3<br>&<br>МТ1.1.1.2 → МТ 1.1.1.3 |
| | | МТ1.1.1.4 Формулювання списку вимог та їхнє документування | МТ1.1.1.3 Проведення інтерв'ю зі зацікавленими сторонами для збору додаткової інформації | МТ1.1.1.3 → МТ 1.1.1.4 |
| | МТ1.1.2 Аналіз вимог на визначення тестових сценаріїв | | МТ1.1.1 Збір вимог | МТ1.1.1 → МТ1. 1.2 |

Продовження таблиці 3.1

| 1 підетап | 2 підетап | 3 підетап | Залежності | Математичне представлення |
|---|---|---|---|---|
| | | МТ1.1.2.1 Ідентифікація ключових функцій та можливих сценаріїв використання | МТ1.1.1 Збір вимог | МТ1.1.1 → МТ1.1.2.1 |
| | | МТ1.1.2.2 Визначення пріоритетів та важливості кожного сценарію для тестування | МТ1.1.2.1 Ідентифікація ключових функцій та можливих сценаріїв використання | МТ1.1.2.1 → МТ1.1.2.2 |
| | | МТ1.1.2.3 Розроблення стратегії тестування на основі вимог та визначених сценаріїв | – МТ1.1.2.1 Ідентифікація ключових функцій та можливих сценаріїв використання;<br>– МТ1.1.2.1 Ідентифікація ключових функцій та можливих сценаріїв використання; | МТ1.1.2.1 → МТ1.1.2.3<br>&<br>МТ1.1.2.1 → МТ1.1.2.3 |
| МТ1.2 Визначення критеріїв прийняття | | | МТ1.1 Аналіз вимог | МТ1.1 → МТ1.2 |
| | МТ1.2.1 Формулювання критеріїв прийняття | | МТ1.1 Аналіз вимог | МТ1.1 → МТ1.2.1 |
| | МТ1.2.2 Перевірка та узгодження критеріїв прийняття зі зацікавленими сторонами | | МТ1.2.1 Формулювання критеріїв прийняття | МТ1.2.1 → МТ1.2.2 |
| МТ1.3 Створення тестових сценаріїв | | | МТ1.2 Визначення критеріїв прийняття | МТ1.2 → МТ1.3 |
| | МТ1.3.1 Визначення сценаріїв для покриття вимог | | МТ1.2 Визначення критеріїв прийняття | МТ1.2 → МТ1.3.1 |

Кінець таблиці 3.1

| 1 підетап | 2 підетап | 3 підетап | Залежності | Математичне представлення |
|---|---|---|---|---|
| | МТ1.3.2 Розробка тестових сценаріїв | | МТ1.3.1 Визначення сценаріїв для покриття вимог | МТ1.3.1 → МТ1.3.2 |
| | МТ1.3.3 Перевірка тестових сценаріїв | | МТ1.3.2 Розробка тестових сценаріїв | МТ1.3.2 → МТ1.3.3 |
| МТ1.4 Узгодження із зацікавленими сторонами | | | МТ1.3 Створення тестових сценаріїв | МТ1.3 → МТ1.4 |
| | МТ1.4.1 Проведення консультацій та обговорення зауважень із зацікавленими сторонами | | МТ1.3 Створення тестових сценаріїв | МТ1.3 → МТ1.4.1 |
| | МТ1.4.2 Внесення змін в тестові вимоги за результатами узгодження | | МТ1.4.1 Проведення консультацій та обговорення зауважень із зацікавленими сторонами | МТ1.4.1 → МТ1.4.2 |
| МТ1.5 Створення тестових даних | | | МТ1.4 Узгодження із зацікавленими сторонами | МТ1.4 → МТ1.5 |
| | МТ1.5.1 Визначення необхідних тестових даних | | МТ1.4 Узгодження із зацікавленими сторонами | МТ1.4 → МТ1.5 |
| | МТ1.5.2 Завантаження та підготовка тестових даних | | МТ1.5.1 Визначення необхідних тестових даних | МТ1.5.1 → МТ1.5.2 |

Розглянемо залежності між підетапами для етапу «МТ2. Планування тестування» складається з наступних підетапів», які представлені в таблиці 3.2.

Таблиця 3.2 – Встановлення залежностей для «МТ2. Планування тестування»

| 1 підетап | 2 підетап | 3 підетап | Залежності | Математичне представлення |
|---|---|---|---|---|
| МТ2. Планування тестування | | | МТ1 Визначення тестових вимог | МТ1 → МТ2 |
| МТ2.1 Розробка стратегії тестування | | | | |
| | МТ2.1.1 Визначення цілей тестування та стратегії їх досягнення | | МТ1.2 Визначення критеріїв прийняття | МТ1.2 → МТ2.1.1 |
| | | МТ2.1.1.1 Визначення основних цілей, які потрібно досягти в процесі тестування | МТ1.2 Визначення критеріїв прийняття | МТ1.2 → МТ2.1.1 |
| | | МТ2.1.1.2 Формулювання конкретних завдань та вимог, які входять у визначені цілі | МТ2.1.1.1 Визначення основних цілей, які потрібно досягти в процесі тестування | МТ2.1.1.1 → МТ2.1.1.2 |
| | МТ2.1.2 Вибір методів та підходів до тестування | | МТ2.1.1 Визначення цілей тестування та стратегії їх досягнення | МТ2.1.1 → МТ2.1.2 |
| | МТ2.1.3 Встановлення пріоритетів у проведенні тестів | | МТ2.1.1 Визначення цілей тестування та стратегії їх досягнення | МТ2.1.1 → МТ2.1.3 |
| МТ2.2 Визначення інструментів та ресурсів | | | МТ2.1 Розробка стратегії тестування | МТ2.1 → МТ2.2 |
| | МТ2.2.1 Підбір необхідних інструментів для проведення тестування | | МТ2.1 Розробка стратегії тестування | МТ2.1 → МТ2.2.1 |
| | МТ2.2.2 Оцінка доступності та необхідності ресурсів для виконання тестів | | МТ2.2.1 Підбір необхідних інструментів для проведення тестування | МТ2.2.1 → МТ2.2.2 |
| МТ2.3 Формування графіку тестування | | | МТ2.2 Визначення інструментів та ресурсів | МТ2.2 → МТ2.3 |

Кінець таблиці 3.2

| 1 підетап | 2 підетап | 3 підетап | Залежності | Математичне представлення |
|---|---|---|---|---|
| | МТ2.3.1 Визначення термінів та пріоритетів проведення окремих етапів тестування | | МТ2.2 Визначення інструментів та ресурсів | МТ2.2 → МТ2.3.1 |
| | МТ2.3.2 Створення графіка роботи команди тестування | | МТ2.3.1 Визначення термінів та пріоритетів проведення окремих етапів тестування | МТ2.3.1 → МТ2.3.2 |
| МТ2.4 Розподіл завдань | | | МТ2.3 Формування графіку тестування | МТ2.3 → МТ2.4 |
| | МТ2.4.1 Визначення обсягу робіт та розподіл завдань між учасниками команди тестування та іншими зацікавленими сторонами | | МТ2.3 Формування графіку тестування | МТ2.3 → МТ2.4 |
| | МТ2.4.2 Встановлення відповідальності за виконання кожного завдання | | МТ2.4.1 Визначення обсягу робіт та розподіл завдань між учасниками команди тестування та іншими зацікавленими сторонами | МТ2.4.1 → МТ2.4.2 |
| МТ2.5 Аналіз вимог до тестового середовища | | | МТ2.4 Розподіл завдань | МТ2.4 → МТ2.5 |
| | МТ2.5.1 Визначення вимог до конфігурації тестового середовища | | МТ2.4 Розподіл завдань | МТ2.4 → МТ2.5.1 |
| | МТ2.5.2 Перевірка наявності необхідних ресурсів та програмного забезпечення | | МТ2.5.1 Визначення вимог до конфігурації тестового середовища | МТ2.5.1 → МТ2.5.2 |
| МТ2.6 Перевірка тестового середовища | | | МТ2.5 Аналіз вимог до тестового середовища | МТ2.5 → МТ2.6 |
| | МТ2.6.1 Проведення тестів для перевірки функціональності тестового середовища | | МТ2.5 Аналіз вимог до тестового середовища | МТ2.5 → МТ2.6 |
| | МТ2.6.2 Вирішення будь-яких виявлених проблем або недоліків | | МТ2.6.1 Проведення тестів для перевірки функціональності тестового середовища | МТ2.6.1 → МТ2.6.2 |

Розглянемо залежності між підетапами для етапу «МТ3 Виконання тестування», які представлені в таблиці 3.3.

Таблиця 3.3 – Встановлення залежностей для «МТ3 Виконання тестування»

| 1 підетап | 2 підетап | 3 підетап | Залежності | Математичне представлення |
|---|---|---|---|---|
| МТ3 Виконання тестування | | | МТ2.Плануванння тестування | МТ2 → МТ3 |
| МТ3.1 Запуск тестів | | | МТ2. Плануванння тестування | МТ2 → МТ3.1 |
| | МТ3.1.1 Підготовка до запуску тестових сценаріїв | | МТ2. Плануванння тестування | МТ2 → МТ3.1.1 |
| | МТ3.1.2 Запуск автоматизованих та ручних тестів | | МТ3.1.1 Підготовка до запуску тестових сценаріїв | МТ3.1.1 → МТ3.1.2 |
| | | МТ3.1.2.1 Запуск автоматизований тестів | МТ3.1.1 Підготовка до запуску тестових сценаріїв | МТ3.1.1 → МТ3.1.2.1 |
| | | МТ3.1.2.2 Проходження ручних тестів | МТ3.1.2.1 Запуск автоматизований тестів | МТ3.1.2.1 → МТ3.1.2.2 |
| МТ3.2 Збір та аналіз результатів | | | МТ3.1 Запуск тестів | МТ3.1 → МТ3.2 |
| | МТ3.2.1 Збір результатів виконання тестів | | МТ3.1 Запуск тестів | МТ3.1 → МТ3.2.1 |
| | МТ3.2.2 Аналіз отриманих результатів на предмет виявлення дефектів | | МТ3.2.1 Збір результатів виконання тестів | МТ3.2.1 → МТ3.2.2. |
| МТ3.3 Реєстрація багів | | | МТ3.2 Збір та аналіз результатів | МТ3.2 → МТ3.3 |
| | МТ3.3.1 Фіксація виявлених дефектів або проблем в системі | | МТ3.2 Збір та аналіз результатів | МТ3.2 → МТ3.3.1 |
| | МТ3.3.2 Описання деталей та відображення результатів тестування в системі управління багами | | МТ3.3.1 Фіксація виявлених дефектів або проблем в системі | МТ3.3.1 → МТ3.3.2 |
| МТ3.4 Налаштування тестового середовища | | | МТ3.3 Реєстрація багів | МТ3.2 → МТ3.4 |

Кінець таблиці 3.3

| 1 підетап | 2 підетап | 3 підетап | Залежності | Математичне представлення |
|---|---|---|---|---|
| | МТ3.4.1 Оновлення програмного забезпечення тестового середовища | | МТ3.3 Реєстрація багів | МТ2 → МТ3.4.1 |
| | МТ3.4.2 Конфігурація середовища для подальших тестів | | МТ3.4.1 Оновлення програмного забезпечення тестового середовища | МТ3.4.1 → МТ3.4.2 |
| МТ3.5 Налаштування апаратного середовища | | | МТ3.4 Налаштування тестового середовища | МТ3.4 → МТ3.5 |
| | МТ3.5.1 Підготовка апаратного забезпечення для виконання тестів | | МТ3.4 Налаштування тестового середовища | МТ3.4 → МТ3.5.1 |
| | МТ3.5.2 Перевірка та налаштування робочих параметрів апаратури перед тестуванням | | МТ3.5.1 Підготовка апаратного забезпечення для виконання тестів | МТ3.5.1 → МТ3.5.2 |

Розглянемо залежності між підетапами для етапу «МТ4 Аналіз результатів та підготовка звітності», які представлені в таблиці 3.4.

Таблиця 3.4 – Встановлення залежностей для «МТ4 Аналіз результатів та підготовка звітності»

| 1 підетап | 2 підетап | 3 підетап | Залежності | Математичне представлення |
|---|---|---|---|---|
| МТ4 Аналіз результатів та підготовка звітності | | | МТ3Виконання тестування | МТ3 → МТ4 |
| МТ4.1 Визначення моменту зупинки тестування | | | МТ3 Виконання тестування | МТ3 → МТ4.1 |
| | МТ4.1.1 Оцінка метрик якості | | МТ3 Виконання тестування | МТ3 → МТ4.1.1 |
| | | МТ4.1.1.1 Аналіз метрик відповідності до визначених критеріїв. | МТ3 Виконання тестування | МТ3 → МТ4.1.1.1 |

Продовження таблиці 3.4

| 1 підетап | 2 підетап | 3 підетап | Залежності | Математичне представлення |
|---|---|---|---|---|
| | | МТ4.1.1.2 Порівняння результатів з попередніми вимогами | МТ4.1.1.1 Аналіз метрик відповідності до визначених критеріїв. | МТ4.1.1.1 → МТ4.1.1.2 |
| | МТ4.1.2 Порівняння отриманих результатів з попередніми критеріями завершення | | МТ4.1.1 Оцінка метрик якості | МТ4.1.1 → МТ4.1.2 |
| | | МТ4.1.2.1 Визначення критеріїв завершення тестування | МТ4.1.1 Оцінка метрик якості | МТ4.1.1 → МТ4.1.2.1 |
| | | МТ4.1.2.2 Порівняння фактичних результатів зі встановленими критеріями | МТ4.1.2.1 Визначення критеріїв завершення тестування | МТ4.1.2.1 → МТ4.1.2.2 |
| | | МТ4.1.2.3 Прийняття рішення про продовження або завершення тестування | – МТ4.1.2.1 Визначення критеріїв завершення тестування;<br>– МТ4.1.2.2 Порівняння фактичних результатів зі встановленими критеріями | МТ4.1.2.1 → МТ4.1.2.3<br>&<br>МТ4.1.2.2 → МТ4.1.2.3 |
| | МТ4.1.3 Вирішення, чи досягнутий достатній рівень якості для завершення тестування | | МТ4.1.2 Порівняння отриманих результатів з попередніми критеріями завершення | МТ4.1.2 → МТ4.1.3 |
| МТ4.2 Ідентифікація та усунення дефектів | | | МТ4.1 Визначення моменту зупинки тестування | МТ4.1 → МТ4.2 |
| | МТ4.2.1 Аналіз результатів тестування для виявлення дефектів | | МТ4.1 Визначення моменту зупинки тестування | МТ4.1 → МТ4.2.1 |
| | МТ4.2.2 Класифікація та призначення пріоритетів дефектам | | МТ4.2.1 Аналіз результатів тестування для виявлення дефектів | МТ4.2.1 → МТ4.2.2 |

Продовження таблиці 3.4

| 1 підетап | 2 підетап | 3 підетап | Залежності | Математичне представлення |
|---|---|---|---|---|
| | | МТ4.2.2.1 Визначення важливості та впливу дефектів на функціональність продукту | МТ4.2.1 Аналіз результатів тестування для виявлення дефектів | МТ4.2.1 → МТ4.2.2.1 |
| | | МТ4.2.2.2 Налаштування пріоритетів для подальшого усунення | МТ4.2.2.1 Визначення важливості та впливу дефектів на функціональність продукту | МТ4.2.2.1 → МТ4.2.2.2 |
| | МТ4.2.3 Усунення виявлених дефектів розробниками | | МТ4.2.2 Класифікація та призначення пріоритетів дефектам | МТ4.2.2 → МТ4.2.3 |
| | | МТ4.2.3.1 Передача дефектів розробникам для виправлення | МТ4.2.2 Класифікація та призначення пріоритетів дефектам | МТ4.2.2 → МТ4.2.3.1 |
| | | МТ4.2.3.2 Перевірка виправлених дефектів | МТ4.2.3.1 Передача дефектів розробникам для виправлення | МТ4.2.3.1 → МТ4.2.3.2 |
| МТ4.3 Демонстрація відповідності вимогам та оцінка якості на основі встановлених вимог | | | МТ4.2 Ідентифікація та усунення дефектів | МТ4.2 → МТ4.3 |
| | МТ4.3.1 Порівняння результатів тестування з вимогами до продукту | | МТ4.2 Ідентифікація та усунення дефектів | МТ4.2 → МТ4.3.1 |
| | | МТ4.3.1.1 Перевірка відповідності функціональності продукту встановленим вимогам | МТ4.2 Ідентифікація та усунення дефектів | МТ4.2 → МТ4.3.1.1 |

Кінець таблиці 3.4

| 1 підетап | 2 підетап | 3 підетап | Залежності | Математичне представлення |
|---|---|---|---|---|
| | | МТ4.3.1.2 Виявлення потенційних неспівпадінь | МТ4.3.1.1 Перевірка відповідності функціональності продукту встановленим вимогам | МТ4.3.1.1 → МТ4.3.1.2 |
| | МТ4.3.2 Виявлення невідповідностей та їх вирішення | | МТ4.3.1 Порівняння результатів тестування з вимогами до продукту | МТ4.3.1 → МТ4.3.2 |
| | МТ4.3.3 Оцінка загальної якості продукту | | МТ4.3.2 Виявлення невідповідностей та їх вирішення | МТ4.3.2 → МТ4.3.3 |
| | МТ4.3.4 Підготовка звітності та презентація результатів зацікавленим сторонам | | МТ4.3.3 Оцінка загальної якості продукту | МТ4.3.3 → МТ4.3.4 |
| | | МТ4.3.4.1 Створення звіту з результатами тестування | МТ4.3.3 Оцінка загальної якості продукту | МТ4.3.3 → МТ4.3.4.1 |
| | | МТ4.3.4.2 Підготовка до презентації | МТ4.3.4.1 Створення звіту з результатами тестування | МТ4.3.4.1 → МТ4.3.4.2 |

## 3.2 Розробка графічної частини інфологічної моделі процесу тестування

Результати визначення залежностей між етапами та підетапами (табл. 3.1 – 3.4) є основою для ефективної побудови діаграми Ганта. Ця діаграма є важливим інструментом управління як цілим проектом, так і конкретним процесом тестування, оскільки дозволяє візуалізувати та планувати часові рамки для виконання різних завдань.

Розглянемо графічне представлення для найвищого рівня та декомпозиції 4 основних процесів.

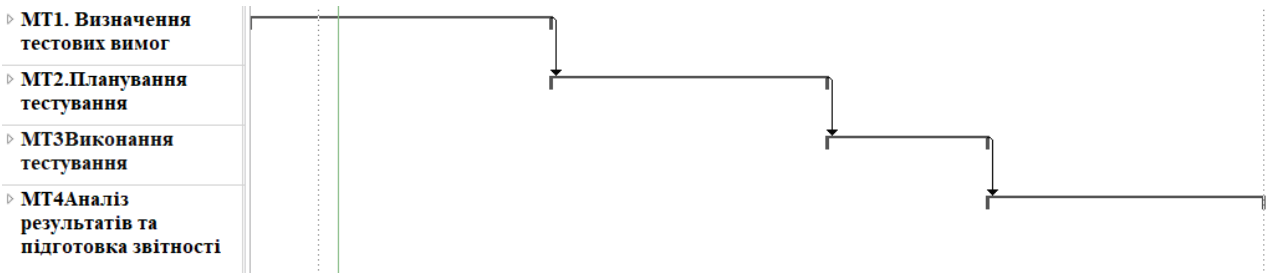


Рисунок 3.1 – Діаграма Ганта для найвищого рівня

Рисунок 3.2 – Діаграма Ганта з декомпозицією пергошо етапу

Рисунок 3.3 – Діаграма Ганта з декомпозицією другого етапу

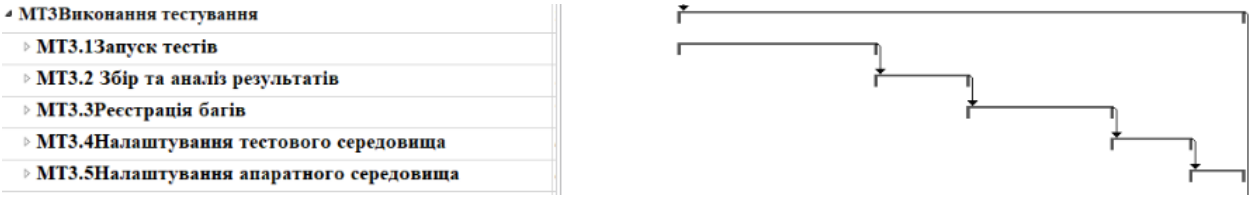


Рисунок 3.4 – Діаграма Ганта з декомпозицією третього етапу

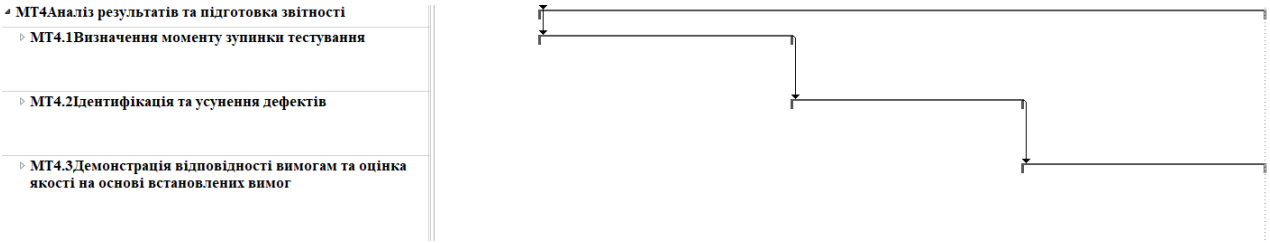


Рисунок 3.5 – Діаграма Ганта з декомпозицією четвертого етапу

Отримані графічні результати представлені на рисунках 3.1 – 3.5 базуються на результатах визначення залежностей між етапами та етапами. Деталізована діаграма Ганта для нижчих рівнів декомпозиції представлено в додатку Б.

## 3.3 Висновки до третього розділу

Визначені залежності між етапами та підетапами в підрозділі 3.1 слугували основою для побудови діаграми Ганта, яка слугуватиме не лише інструментом управління конкретним проектом, але й патерном для подібних проектів з "Визначеним" рівнем зрілості. Це стане можливим завдяки тому, що діаграма Ганта дозволяє систематизувати та візуалізувати послідовність виконання завдань, а також виявляти критичні залежності та ресурси, необхідні для успішного завершення проекту.

# 4 АПРОБАЦІЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

## 4.1 Загальні відомості, щодо проекту

У розділі 3 була розроблена інфологічна модель для управління процесом тестування. Надалі, для оцінки ефективності створеного патерну, буде проведено його порівняння зі стандартними методиками планування.

Перед початком цього порівняння розглянемо мінімально життєздатний продукт (MVP) для нового проекту, який має такі характеристики:

Команда складається з 10 фахівців:

– 3 тестувальника;

– 4 розробника;

– 1 бізнес-аналітик;

– 1 менеджер проекту;

– 1 дизайнер.

Завдання проекту та його концепція представлені у таблицях 4.1 та 4.2 відповідно.

Таблиця 4.1 – Постановка задачі проекту

| | |
|---|---|
| Проблема | створення заходів та реєстрації на них. |
| впливає на | доступність та легкість як створення інформації про захід, так і реєстрації на захід. |
| причина із за якої | складнощі з організацією заходу. |
| вдалим рішенням було би | інформаційна система, що дозволить створювати заходи різних типів (концерти, книжкові заходи, бранч та інші) та спростить реєстрацію на них. |

Таблиця 4.2 – Концепція проекту

| | |
|---|---|
| Для кого | користувачі інтернету, зацікавлені в організації та участі у заходах. |
| APPS | це рішення, що включає в себе адаптивний кросплатформений та кросбраузерний веб-додаток. |
| Це | забезпечить простий та зручний спосіб створення заходів та реєстрації на них. |
| На відміну | від існуючих інформаційних систем для заходів |
| Наш продукт | інформаційна система, що дозволить створювати заходи різних типів (концерти, книжкові заходи, бранч та інші) та спростить реєстрацію на них. |

Бізнес-можливості цієї інформаційної системи полягають у здатності створювати заходи різних типів. Ініціатор заходу зможе самостійно визначати параметри та умови участі, тоді як зацікавлений користувач матиме можливість зареєструватися для участі в заході. Ця інформаційна система забезпечить ініціаторам та учасникам широкий спектр заходів, враховуючи різноманітні смаки та інтереси.

Проблема існуючих додатків із інформацією про заходи полягає в тому, що заходи часто типові, наприклад, концерти, театри чи цирки, і створюються адміністраторами платформи. Такі додатки, як правило, орієнтовані на масштабні події, не враховуючи специфічні чи невеликі заходи, такі як настільні ігри або клуб книголюбів.

Система APPS повинна вирішити ці проблеми, надаючи користувачам широкий каталог заходів у різних містах України. Для MVP робочий процес виглядає наступним чином:

– реєстрація та авторизація. Користувач проходить реєстрацію, авторизацію або відновлює дані для входу;

– створення заходу: Ініціатор заходу заповнює обов'язкові поля щодо заходу, включаючи умови участі;

– реєстрація користувачів. Зацікавлений користувач реєструється на захід, редагує участь або скасовує реєстрацію;

– підтвердження. Ініціатор підтверджує проведення заходу та наявність зареєстрованих учасників;

– участь. Зацікавлений користувач підтверджує свою участь під час проведення заходу;

– перевірка. Система перевіряє наявність реєстрації користувача на кількох заходах одночасно.

Такий підхід дозволяє користувачам та організаторам максимально ефективно взаємодіяти в межах платформи.

Розглянемо функціонал зареєстрованого користувача представленого в таблиці 4.3.

Таблиця 4.3 – Зареєстрований користувач

| | |
|---|---|
| Опис | Користувач з доступом до Інтернету, який бажає організувати або приєднатися до заходу. |
| Тип | Користувач, обізнаний у сучасних технологіях. |
| Критерій успіху | Кількість користувачів, які створили захід або зареєструвалися на захід. |
| Участь | Не визначено |

Поза сфери дії MVP:

– модерація створення заходів для виявлення та видалення провокаційних або небезпечних заходів;

– організація платних заходів;

– впровадження відстеження заходів за обраними параметрами;

– введення платної підписки;

– інтеграція із закладами чи коворкінгами.

## 4.2 Планування процесу тестування стандартними методиками

Розглянемо планування процесу тестування за допомогою двох стандарних методів  для даного проекту, а саме методу критичного шляху та методу критичного ланцюга.

### 4.2.1 Планування за допомогою методу критичного шляху

Метод критичного шляху представляє з себе техніку управління проектами, яка дозволяє визначити критичний шлях проекту, тобто послідовність задач, яка визначає мінімальний час необхідний для завершення проекту. В свою чергу затримка будь-якої задачі на критичному шляху призведе до затримки всього проекту. Даний метод фокусується на ранньому виявлені та управлінні критичними задачами, що в свою чергу дозволяє ефективно розподіляти ресурси.

Розглянемо планування процесу тестування за допомогою методу критичного шляху для проекту описаного в підрозділі 4.1, результат представлено в таблиці 4.4.

Таблиця 4.4 – Управління процесом тестування для проекту методом критичного шляху

| № | Етап | Етапи складаються | Залучені ресурси та їх задачі |
|---|---|---|---|
| 1 | Визначення завдань та оцінка їх тривалості | – ідентифікація завдань;<br>– оцінка часу;<br>– залучення експертів | Проектний менеджер координує визначення завдань та оцінку їх тривалість. |
| | | | Бізнес-аналітик ідентифікує та аналізує бізнес-вимоги, які формують завдання. |
| | | | Головний розробник визначає технічні завдання та їх тривалість. |

Кінець таблиці 4.4

| № | Етап | Етапи складаються | Залучені ресурси та їх задачі |
|---|---|---|---|
| | | | Головний тестувальник визначає завдання з тестування та необхідний час для їх виконання. |
| 2 | Визначення залежностей між етапами | – аналіз зв’язків;<br>– налаштування зв’язків | Проектний менеджер координує розробку мережевої діаграми та ідентифікує залежності. |
| | | | Головний розробник та головний тестувальник визначають технічні залежності між завданнями розробки та тестування. |
| 3 | Побудова мережевої діаграми | | Проектний менеджер керує побудовою мережевої діаграмою та забезпечує її точність. |
| 4 | Розрахунок критичного шляху | – визначення критичного шляху;<br>– оцінка гнучкості проекту | Проектний менеджер аналізує та визначає критичний шлях проекту. |
| 5 | Оптимізація ресурсів | | Проектний менеджер координує розподіл та оптимізацію ресурсів. |
| | | | Головний розробник та головний тестувальник визначають потреби в ресурсах для команд та сприяють ефективному їх використанню. |
| 6 | Моніторинг та контроль | – відстеження прогресу;<br>– моніторинг відхилень;<br>– підготовка звітності | Проектний менеджер моніторить прогрес проекту і коригування планів. |
| | | | Головний тестувальник відстежує прогрес в тестування та надає зворотній зв’язок. |

Для побудови плану управління процесом тестування методом критичного шляху, важливо залучити чотирьох спеціалістів для забезпечення всіх аспектів управління та координації. Втім, використання цього методу супроводжується певними ризиками:

– неточністю оцінки тривалості завдань;

– некоректністю визначення залежностей між завданнями;

– зміни в критичному шляху.

Перелічені ризики впливатимуть на час виконання проекту за рахунок необхідності внесення корективів в план та повторній залученості спеціалістів.

### 4.2.2 Планування за допомогою методу критичного ланцюга

Метод критичного ланцюгу зосереджується на управлінні ресурсами проекту та враховує доступність ресурсів для виконання задач. Ключовою характеристикою даного методу є встановлення резервів часу, які допомагають поглинати затримки реалізації задач. Даний метод є більш гнучким та дозволяє більш реалістично планувати та виконувати проектні задачі.

Розглянемо планування процесу тестування за допомогою методу критичного ланцюга для проекту описаного в підрозділі 4.1, результат представлено в таблиці 4.5.

Таблиця 4.5 – Управління процесом тестування для проекту методом критичного ланцюга

| № | Етап | Етапи складаються | Залучені ресурси та їх задачі |
|---|---|---|---|
| 1 | Визначення обсягу проекту та основних завдань | – збір вимог;<br>– визначення завдань | Проектний менеджер координує процес збору інформації та визначення всіх необхідних задач. |
| | | | Бізнес-аналітик збирає та аналізує бізнес-вимоги та формує основу для завдань тестування. |
| | | | Головний розробник та головний тестувальник визначають технічні аспекти завдань, які впливають на процес розробки та тестування |
| 2 | Визначення залежностей і критичних ресурсів | – аналіз залежностей;<br>– ідентифікація критичних ресурсів | Бізнес-аналітик та проектний менеджер ідентифікують залежності між бізнес-задачами. |

Кінець таблиці 4.5

| № | Етап | Етапи складаються | Залучені ресурси та їх задачі |
|---|---|---|---|
| | | | Головний розробник та головний тестувальник ідентифікують залежності між технічними задачами та визначають критичні технічні ресурси. |
| 3 | Побудова діаграми критичного ланцюга | – моделювання процесу | Проектний менеджер будує та оптимізує мережеву діаграму. |
| | | | Головний розробник уточнює технічні деталі завдань. |
| 4 | Встановлення буферів | – встановлення буферів проекту;<br>– встановлення буферів ресурсів | Проектний менеджер встановлює буфери часу для критичних завдань та ресурсів. |
| 5 | Оптимізація плану | – аналіз альтернативних шляхів;<br>– побудова гнучкості плану | Головний розробник допомагає з перерозподілом ресурсів та коригуванням завдань. |
| | | | Команда тестування фокусується на оптимальному використанні своїх ресурсів, забезпечуючи якість та швидкість тестування. |
| 6 | Впровадження та моніторинг | – моніторинг виконання | Проектний менеджер моніторить хід виконання проекту. |
| | | | Команда тестування забезпечує звітність про стан виконання тестів. |

Для побудови плану управління процесом тестування методом критичного ланцюга, важливо залучити трьох спеціалістів та команду тестування для ретельного планування. Втім, використання цього методу супроводжується певними ризиками, які слід врахувати під час планування:

– зосередженість на одному критичному ресурсі, а саме даний метод фокусується на критичному ресурсі, дане обмеження може призвести до недостатньої уваги до інших важливих аспектів проекту, які потребують ресурсів та управління;

– складність визначення буферів, а саме існує ризик недооцінки або переоцінки потрібної кількості буфера;

– залежність від точності даних, а саме даний метод залежить від точності вихідних даних по тривалості задач та доступності ресурсів.

## 4.3 Планування процесу тестування за допомогою розробленої інфологічної моделі

Розглянемо яким чином розроблена в розділі 3 інфологічна модель може бути використаною та адаптованою для планування тестування для проекту описаного в підрозділі 4.1. Розглянемо інтеграцію інфологічної моделі при планування управління проектом з використання методу критичного шлях, результат представлено в таблиці 4.6.

Таблиця 4.6 – Оптимізація процесу планування тестування з використанням інфологічної моделі

| № | Етап | Залучені спеціалісти для планування процесу тестування без використання інфологічної моделі | Застосування інфологічної моделі впливатиме | Залучені спеціалісти для планування процесу тестування з використанням інфологічною моделлю |
|---|---|---|---|---|
| 1 | Визначення завдань та оцінка їх тривалості | – проектний менеджер;<br>– бізнес-аналітик;<br>– головний розробник;<br>– головний тестувальник; | Автоматизація процесу ідентифікації та документування задач необхідних для тестування, за рахунок використання готового патерну. | – проектний менеджер;<br>– бізнес-аналітик;<br>– головний тестувальник |

Кінець таблиці 4.6

| № | Етап | Залучені спеціалісти для планування процесу тестування без використання інфологічної моделі | Застосування інфологічної моделі впливатиме | Залучені спеціалісти для планування процесу тестування з використанням інфологічною моделлю |
|---|---|---|---|---|
| 2 | Визначення залежностей між етапами | – проектний менеджер;<br>– головний тестувальний;<br>– головний розробник; | Спрощення побудови залежностей між завданнями, за рахунок патерну, який дозволить більш точно налаштувати зв’язки для проекту. | – проектний менеджер;<br>– головний тестувальник |
| 3 | Побудова мережевої діаграми | – проектний менеджер; | – | – проектний менеджер; |
| 4 | Розрахунок критичного шляху | – головний тестувальник;<br>– проектний менеджер | Пришвидшення визначення критичного шляху проекту та оптимізація його. | – головний тестувальник;<br>– проектний менеджер |
| 5 | Оптимізація ресурсів | – проектний менеджер;<br>– головний тестувальний;<br>– головний розробник; | – | – проектний менеджер;<br>– головний тестувальний;<br>– головний розробник; |
| 6 | Моніторинг та контроль | – проектний менеджер;<br>– головний тестувальний;<br>– головний розробник; | – | – проектний менеджер;<br>– головний тестувальний;<br>– головний розробник; |

У випадку планування процесу тестування за допомогою методу критичного ланцюга з використанням розробленої інфологічної моделі, можна пришвидшити виконання деяких етапів, а саме:

– визначення завдань та оцінка їх тривалості;

– визначення залежностей між етапами;

– розрахунок критичного шляху.

Оптимізація часу для планування відбувається за рахунок того, що інфологічна модель представляє з себе патерн, який можна адаптувати під потреби проекту. Ще одною перевагою використання інфологічної моделі є зменшення використання людський ресурсів. Розглянемо інтеграцію інфологічної моделі при планування управління проектом з використання методу критичного ланцюга, результат представлено в таблиці 4.7.

Таблиця 4.7 – Оптимізація процесу планування тестування з використанням інфологічної моделі

| № | Етап | Залучені спеціалісти для планування процесу тестування без використання інфологічної моделі | Застосування інфологічної моделі впливатиме | Залучені спеціалісти для планування процесу тестування з використанням інфологічною моделлю |
|---|---|---|---|---|
| 1 | Визначення обсягу проекту та основних завдань | – проектний менеджер;<br>– бізнес-аналітик;<br>– головний розробник;<br>– головний тестувальник; | Пришвидшення формування списку завдань та вимог. | – проектний менеджер;<br>– бізнес-аналітик;<br>– головний тестувальник |
| 2 | Визначення залежностей і критичних ресурсів | – проектний менеджер;<br>– головний тестувальний;<br>– головний розробник;<br>– бізнес-аналітик | Пришвидшення визначення критичних ресурсів на основі історичних даних та готових взаємозв’язків. | – проектний менеджер;<br>– бізнес-аналітик; |
| 3 | Побудова діаграми критичного ланцюга | – проектний менеджер;<br>– головний розробник; | – | – проектний менеджер;<br>– головний розробник; |
| 4 | Встановлення буферів | – проектний менеджер; | – | – проектний менеджер; |

Кінець таблиці 4.7

| № | Етап | Залучені спеціалісти для планування процесу тестування без використання інфологічної моделі | Застосування інфологічної моделі впливатиме | Залучені спеціалісти для планування процесу тестування з використанням інфологічною моделлю |
|---|---|---|---|---|
| 5 | Оптимізація плану | – проектний менеджер;<br>– команда тестування; | – | – проектний менеджер;<br>– команда тестування; |
| 6 | Впровадження та моніторинг | – проектний менеджер;<br>– команда тестування; | – | – проектний менеджер;<br>– команда тестування; |

За допомогою інтеграції інфологічної моделі в планування методом критичного ланцюга можна також пришвидшити виконання деяких етапів, а саме:

– визначення обсягу проекту та основних завдань;

– визначення залежностей і критичних ресурсів.

## 4.4 Висновки до четвертого розділу

Інтеграція інфологічної моделі у процес планування за допомогою методів критичного шляху та критичного ланцюга сприятиме оптимізації та підвищенню ефективності. Ця модель може підвищити продуктивність проекту, зменшити потребу в ручному введенні даних, і скоротити час, необхідний для планування та виконання завдань. Інтеграція інфологічної моделі в планування має такі переваги:

– зменшення трудомісткості. Автоматизація багатьох аспектів планування знижує загальну трудомісткість проекту, дозволяючи команді зосередитися на більш складних та критичних завданнях;

– покращення точності. Інфологічна модель зменшує можливість людських помилок за рахунок наявності готового патерну для управління тестуванням, яких можна адаптувати під потреби проекту.

Завдяки цим перевагам, інфологічна модель може стати корисним інструментом для підвищення ефективності управління проектами, особливо в умовах, коли швидкість та точність є вирішальними факторами успіху.

# ВИСНОВКИ

В даній кваліфікаційній роботі було вирішено актуальне питання побудови інфологічної моделі для управління процесом тестування, що дозволить інтегрувати дане рішення для планування процесу тестування для проектів із «Визначеним» рівнем зрілості.

В першому розділі було проведено аналіз різних підходів до управління процесом тестування та актуальних наукових робіт, спрямованих на пошук універсального методу керування тестуванням. В результаті цього аналізу були виділені чотири основних етапи, які є спільними для двох чи більше розглянутих підходів.

У другому розділі було розширено перелік підетапів шляхом декомпозиції та інтеграції двох додаткових етапів. Також здійснено процес декомпозиції до найменших неподільних процесів. Визначивши найменший процес, було проведено оцінку часових витрат для кожного з підетапів на його основі, а також оцінено потребу в людських ресурсах для кожного підетапу.

У третьому розділі було визначено залежності між підетапами та етапами процесу тестування, які були представлені в табличному вигляді для кращого розуміння взаємозв'язків. Спираючись на ці табличні дані, було створено графічне представлення інфологічної моделі за допомогою діаграми Ганта. Це дозволило візуально відобразити послідовність виконання етапів та підетапів, а також встановити їх часові рамки та взаємозалежності, що сприяє більш ефективному плануванню та управлінню процесом тестування

У четвертому розділі було детально розглянуто приклад проекту та його планування з використанням методів критичного шляху та критичного ланцюга. На цьому етапі також було показано, як ці методи дозволяють ідентифікувати ключові завдання, які визначають загальну тривалість проекту, а також визначити ресурси, необхідні для виконання цих завдань.

Після цього було проведено інтеграцію розробленої у розділі 3 інфологічної моделі в процес планування проекту. Ця модель дозволяє значно зекономити час при виконанні деяких задач планування за рахунок автоматизації ряду процесів. Зокрема, інфологічна модель сприяє зменшенню необхідності залучення людських ресурсів для планування, що дозволяє ефективніше розподіляти ресурси та зосередитися на критично важливих аспектах проекту. Інтеграція інфологічної моделі також допомагає покращити точність оцінок і оптимізувати розклад робіт, що в кінцевому підсумку підвищує загальну ефективність управління проектом.

# ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ ISO 3008:2015. Звіти у сфері науки і техніки. Структура та правила оформлювання. Київ, 2016.

2. ДСТУ ISO 9001:2015. Системи управління якістю. Вимоги. Чинний від 14.01.2006. Вид. офіц. Київ: УкрНДНЦ, 2015.

3. Software Testing Life Cycle. URL: https://www.guru99.com/software-testing-life-cycle.html (дата звернення: 17.04.2024).

4. Londesbrough, I. A Test Process for all Lifecycles // IEEE International Conference on Software Testing Verification and Validation Workshop. Lillehammer, 2008. С. 1-4.

5. Таранченко С.І. Особливості інваріантних моделей процесу верифікації ІТ-продукту // IV Всеукраїнська студентська конференція «Науковий простір: аналіз, сучасний стан, тренди та перспективи - Івано-Франківськ, 2023. С. 420-422.

6. Таранченко С.І. Дослідження можливості формування підмножини універсальних моделей та методів тестування ІТ-продуктів // 28-й Міжнародного молодіжного форуму «Радіоелектроніка і молодь в XXIстолітті» зб. матеріали форуму, т.6.2 «Інформаційні системи і технології управління проектною та операційною діяльністю підприємств та організацій» - Харків; ХНУРЕ, 2024;

7. Методичні вказівки щодо розробки та оформлення кваліфікаційної роботи другого (магістерського) рівня вищої освіти за освітньо-науковою програмою «Управління проектами в галузі інформаційних технологій» спеціальності 122 Комп’ютерні науки. Харків : Кафедра Інформаційних управляючих систем, 2024.