

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА

Удосконалення методу роботи з метрикою середнього часу між відмовами для
забезпечення ефективного оцінювання результатів тестування програмного
забезпечення

Рівень вищої освіти Другий (магістерський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма Освітньо-професійна програма «Інженерія програмного
забезпечення»

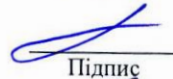
Шифр КвРІПЗ.180115.01.07.ПЗ

Виконав студент 2 курсу, група ІПЗм-22-1


Підпис

Артем СВЕРБА
Ім'я, ПРІЗВИЩЕ

Керівник д-р фіз.-мат. наук, професор
Науковий ступінь, звання


Підпис

Леонід БЕДРАТЮК
Ім'я, ПРІЗВИЩЕ

Нормоконтролер канд. техн. наук, доцент


Підпис

Галина РАДЕЛЬЧУК
Ім'я, ПРІЗВИЩЕ

До захисту допускаю:

Завідувач кафедри
інженерії програмного забезпечення


Підпис

Леонід БЕДРАТЮК
Ім'я, ПРІЗВИЩЕ

7 грудня 2023 р.

Хмельницький 2023

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Інженерії програмного забезпечення

Рівень вищої освіти Другий (магістерський)

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри ІПЗ

Л. П. Бедратюк

01.09.2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Свербі Артему Андрійовичу

Прізвище, ім'я, по батькові здобувача

1. Тема роботи Удосконалення методу роботи з метрикою середнього часу між відмовами для забезпечення ефективного оцінювання результатів тестування програмного забезпечення

Керівник роботи Бедратюк Леонід Петрович, д-р фіз.-мат. наук, професор

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 15.08.2023 р. № 30

2. Строк подання студентом роботи на кафедру 01.12.2023 р.

3. Вихідні дані до роботи Матеріали науково-дослідної практики

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

1. Дослідження предметної області та постановка задачі

2. Аналіз сучасних підходів до удосконалення методу роботи з метрикою МТВФ


3. Розробка удосконаленої методики роботи з метрикою МТВФ для ефективного оцінювання результатів тестування ПЗ

4. Результати застосування удосконаленої методики роботи з метрикою МТВФ та шляхи подальшого покращення

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Презентаційні матеріали (слайди)

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Антиплагиат	Форкун Ю. В., доцент	 23.11.2023р.	 01.12.2023р.
Нормоконтроль	Радельчук Г. І., доцент	 23.11.2023р.	 01.12.2023р.

7. Дата видачі завдання «01» вересня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1 Вивчення предметної області; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження; формування логістичної структури кваліфікаційної роботи	01.09-09.09.2023	
2 Робота над розділом 1 кваліфікаційної роботи – вивчення літературних джерел; аналіз відомих моделей, методів та засобів за темою роботи; визначення методологічних підходів до вирішення задачі; висновки до розділу та постановка задач дослідження	10.09-24.09.2023	
3 Робота над розділом 2 кваліфікаційної роботи – аналіз та порівняння сучасних підходів до удосконалення методу роботи з метрикою, пропозиції щодо їх покращення	25.09-11.10.2023	
4 Робота над науковими статтями	12.10-30.10.2023	
5 Робота над розділом 3 кваліфікаційної роботи – формування вимог та критеріїв оцінювання застосування методики, аналіз та вибір засобів реалізації методики, розробка удосконаленої методики роботи з метрикою MTBF	11.10-29.10.2023	
6 Робота над розділом 4 кваліфікаційної роботи – представлення та аналіз результатів, оцінка ефективності застосування удосконаленої методики, рекомендації щодо подальшого використання та розвитку методики	30.10-17.11.2023	
7 Попередній захист кваліфікаційної роботи	Листопад (згідно графіка)	
8 Узгодження постановки задачі, отриманих результатів та висновків; написання вступу, загальних висновків, оформлення джерел посилання та додатків; оформлення пояснювальної записки та графічних матеріалів згідно вимог чинних стандартів	18.11-30.11.2023	
9 Перевірка роботи на наявність плагіату; нормоконтроль; брошурування пояснювальної записки; підготовка супровідних документів	01.12-04.12.2023	
10 Підготовка до захисту кваліфікаційної роботи	з 01.12.2023 р.	

Студент


 Підпис


 Ім'я, ПРІЗВИЩЕ

Керівник роботи


 Підпис


 Ім'я, ПРІЗВИЩЕ

РЕФЕРАТ

Тема кваліфікаційної роботи: «Удосконалення методу роботи з метрикою середнього часу між відмовами для забезпечення ефективного оцінювання результатів тестування програмного забезпечення».

Автор роботи: Свєрба Артем Андрійович.

Керівник роботи: Бедратюк Леонід Петрович.

Пояснювальна записка: 130 с., 6 рис., 2 дод., 27 джерел.

МЕТРИКА, МТВФ, ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, НАДІЙНІСТЬ, МОДЕЛЮВАННЯ, ІНТЕГРАЦІЯ МЕТРИК, АВТОМАТИЗАЦІЯ, МОНІТОРИНГ В РЕАЛЬНОМУ ЧАСІ, УДОСКОНАЛЕНА МЕТОДИКА.

Об'єктом дослідження є процес тестування програмного забезпечення.

Предметом дослідження є методика роботи з метрикою середнього часу між відмовами (МТВФ).

Мета дослідження – розроблення удосконаленої методики роботи з метрикою середнього часу між відмовами для підвищення ефективності оцінювання результатів тестування програмних продуктів та забезпечення високої надійності їх функціонування.

В процесі виконання кваліфікаційної роботи було виконано дослідження предметної області з описом тестування програмного забезпечення та метрики середнього часу між відмовами. Враховуючи основні аспекти метрики, було представлено особливості її застосування, позитивні сторони та обмеження. Також було здійснено представлення наявних способів покращення роботи з метрикою безвідмовної роботи програмного забезпечення (ПЗ). На основі відповідних підходів було проведено їх порівняльний аналіз та подано рекомендації щодо подальшого їх покращення.

На подальших кроках виконання кваліфікаційної роботи розроблена удосконалена методика роботи з метрикою МТВФ. Для цього спочатку були висунуті вимоги до відповідної методики та визначено критерії оцінки її застосування. Після цього була виконана безпосередня реалізація удосконаленої

методики. На подальших етапах виконано вибір проекту для валідації методики та описано процес тестування відповідного програмного проекту із застосуванням розробленого підходу.

Для реалізації удосконаленої методики роботи з метрикою MTBF обрано високорівневу мову програмування Python, відповідні їй бібліотеки Pandas, NumPy, Matplotlib та середовище розробки Jupyter Notebook.

Після реалізації покращеної методики роботи з метрикою середнього часу між відмовами було виконано представлення та аналіз результатів її застосування, а також виконано оцінку її ефективності. Удосконалена методика позитивно впливає на процес тестування ПЗ та дозволяє приймати важливі рішення щодо підвищення рівня надійності ПЗ. Методика надає можливість виявляти критичні місця роботи ПЗ, допомагає знижувати кількість збоїв та збільшувати середній час між ними. На останніх кроках виконання кваліфікаційної роботи було описано рекомендації щодо застосування методики для різних видів ПЗ та представлено перспективи подальшого її удосконалення та розвитку.


Підпис

06.12.2023
Дата

ABSTRACT

Master's thesis: «Enhancing the method of working with the Mean Time Between Failures metric to ensure effective evaluation of software testing results».

Author: Sverba Artem.

Head of research: Bedratyuk Leonid.

Master's thesis consists of: 130 p., 6 pc., 2 add., 27 src.

METRIC, MTBF, SOFTWARE TESTING, RELIABILITY, SIMULATION, METRICS INTEGRATION, AUTOMATION, REAL-TIME MONITORING, ADVANCED METHODOLOGY.

The object of research is the software testing process.

The subject of the study is the method of working with the metric of mean time between failures (MTBF).

The aim of the research is to develop an improved methodology for working with the metric of mean time between failures to increase the effectiveness of evaluating the results of software testing and ensuring high reliability of its operation.

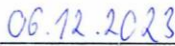
In the course of the thesis, a study of the subject area was performed with a description of the software testing and the metric of the average time between failures. Taking into account the main aspects of the metric, the features of its application, positive sides and limitations were presented. The available ways to improve work with the metric of software uptime were also presented. Based on the respective approaches, their comparative analysis was conducted and recommendations for their further improvement were presented.

In the further steps of the qualification work, an improved methodology for working with the MTBF metric was developed. For this purpose, the requirements for the appropriate methodology were first put forward and the criteria for evaluating its application were determined. After that, the improved methodology was directly implemented. In the subsequent stages, the selection of a project for the validation of the methodology was carried out and the process of testing the corresponding software project using the developed approach was described.

The high-level Python programming language, the corresponding Pandas, NumPy, Matplotlib libraries and the Jupyter Notebook development environment were chosen to implement the improved methodology for working with the MTBF metric.

After the implementation of the improved method of working with the metric of average time between failures, the presentation and analysis of the results of its application was performed, as well as an assessment of its effectiveness. The improved technique has a positive effect on the software testing process and allows to make important decisions about increasing the level of software reliability. The methodology provides an opportunity to identify critical areas of software operation, helps to reduce the number of failures and increase the average time between them. At the last steps of the diploma work, recommendations on the application of the methodology for various types of software were described and prospects for its further development and improvement were presented.


Signature


Date

ЗМІСТ

Вступ.....	9
1 Дослідження предметної області та постановка задачі.....	12
1.1 Аналіз предметної області	12
1.2 Основні поняття та визначення методу МТВФ.....	18
1.3 Методи та техніки розрахунку МТВФ	21
1.4 Використання МТВФ у тестуванні програмного забезпечення	25
1.5 Обмеження та недоліки методу МТВФ у тестуванні ПЗ.....	28
1.6 Постановка задачі	31
2 Аналіз сучасних підходів до удосконалення методу роботи з метрикою МТВФ	33
2.1 Аналіз сучасних підходів до покращення методу МТВФ.....	33
2.2 Інтеграція МТВФ з іншими метриками тестування.....	40
2.3 Автоматизація процесу розрахунку МТВФ.....	47
2.4 Порівняльний аналіз сучасних підходів до удосконалення методу МТВФ...	52
2.5 Загальні пропозиції щодо удосконалення методу МТВФ на основі проведеного аналізу.....	56
3 Розробка удосконаленої методики роботи з метрикою МТВФ для ефективного оцінювання результатів тестування ПЗ	60
3.1 Визначення вимог до удосконаленої методики роботи з МТВФ	60
3.2 Опис критеріїв оцінки ефективності удосконаленої методики МТВФ	65
3.3 Засоби та технології для розробки удосконаленого методу роботи з МТВФ	69
3.4 Реалізація удосконаленого методу роботи з МТВФ	72
3.5 Рекомендації щодо провадження удосконаленої методики МТВФ в процес тестування програмного забезпечення	78
3.6 Вибір конкретного проекту для валідації роботи удосконаленої методики МТВФ.....	82
3.7 Опис процесу тестування обраного проекту із застосуванням удосконаленої методики роботи з МТВФ.....	86
4 Результати застосування удосконаленої методики роботи з метрикою МТВФ та шляхи подальшого покращення	91
4.1 Представлення результатів застосування удосконаленої методики	91

4.2 Аналіз результатів застосування удосконаленої методики.....	94
4.3 Оцінка ефективності удосконаленої методики МТВФ в реальних умовах....	95
4.4 Рекомендації щодо використання удосконаленої методики МТВФ для різних видів програмного забезпечення.....	97
4.5 Перспективи подальшого розвитку та удосконалення методики роботи з метрикою МТВФ.....	100
Висновки	104
Перелік джерел посилання	106
Додаток А Копії наукових публікацій	108
Додаток Б Презентаційні матеріали	122

ВСТУП

Програмне забезпечення в сучасному житті відіграє ключову роль у формуванні, розвитку та оптимізації різноманітних аспектів нашого повсякденного і професійного існування. Зростання важливості програмного забезпечення відбувається на всіх рівнях – від особистих пристроїв та комунікаційних технологій до складних корпоративних систем та індустріальних рішень. Відповідний зростаючий вплив відображає взаємозв'язок програмного забезпечення з сучасним способом життя, технологічним прогресом і стрімкими змінами в суспільстві.

У сучасному світі, що перебуває на перетині цифрової революції, програмне забезпечення є не лише інструментом, але й виступає рушійною силою для здійснення різних трансформацій. Воно визначає ефективність виробництва, забезпечує гнучкість в бізнес-процесах, стимулює інновації та розширює можливості взаємодії між людьми, корпораціями та суспільством в цілому.

Зі стрімким розвитком програмних продуктів та значним їх впливом на повсякденне життя людини важливість тестування програмного забезпечення набуває визначального значення. Тестування виступає як один з найбільш ключових етапів у життєвому циклі розробки програм, основним завдання якого є забезпечення їхньої високої якості, надійності та безпеки. Відповідний етап розробки ПЗ є не просто технічним кроком, але й стратегічною ініціативою, оскільки він визначає коректність та ефективність роботи програм, а також їхню здатність задовольняти потреби користувачів. Висока якість програми визначається не лише відсутністю помилок, але й здатністю відповідати функціональним та нефункціональним вимогам, забезпечуючи стабільність та відмінну продуктивність.

Забезпечення високої якості та надійності програмного забезпечення через тестування має важливий вплив на різні сфери, починаючи від особистих комп'ютерів і закінчуючи критично важливими системами, такими як медичні технології чи фінансові платформи. Нестабільність чи відмова програмного

забезпечення може призвести до серйозних наслідків, включаючи втрату даних, порушення безпеки і навіть загрози життю в окремих випадках.

Необхідність забезпечення високої якості програмних систем породжує важливість постійного удосконалення інструментів та метрик тестування. Одним з головних показників, який визначає стійкість та надійність програм, виступає метрика середнього часу між відмовами (MTBF). Удосконалення методів вимірювання MTBF стає стратегічним завданням, оскільки дана метрика визначає період, протягом якого система або програма може працювати без збоїв. Збільшення MTBF свідчить про збільшення надійності та стійкості програми до різноманітних зовнішніх та внутрішніх ризиків. Саме тому існує потреба пошуку та розробки удосконаленої методики роботи з метрикою середнього часу між відмовами, що є загальною задачею даної кваліфікаційної роботи.

Актуальність теми роботи зумовлена постійним розвитком технологій та зростанням вимог до надійності програмного забезпечення та полягає у пошуку методів для підвищення якості програмних продуктів та зменшення ризиків, пов'язаних із виникненням відмов.

Об'єктом дослідження є процес тестування програмного забезпечення, предметом дослідження є методика роботи з метрикою середнього часу між відмовами (MTBF).

Мета дослідження – розроблення удосконаленої методики роботи з метрикою середнього часу між відмовами для підвищення ефективності оцінювання результатів тестування програмних продуктів та забезпечення високої надійності їх функціонування.

Для досягнення мети роботи потрібно виконати наступні завдання:

- виконати детальний аналіз предметної області;
- проаналізувати наявні методи роботи з метрикою середнього часу між відмовами та їх застосування у процесі тестування ПЗ;
- визначити особливості наявних методів та сформулювати пропозиції для їх покращення;

- визначити вимоги та розробити удосконалену методику роботи з метрикою середнього часу між відмовами;
- провести експериментальне дослідження та виконати оцінку ефективності застосування удосконаленої методики;
- описати рекомендації щодо використання удосконаленої методики МТВФ для різних видів програмного забезпечення;
- оцінити перспективи подальшого розвитку та удосконалення методики роботи з метрикою МТВФ.

Наукова новизна отриманих результатів є такою:

- розроблено комплексний підхід до удосконалення методу роботи з метрикою МТВФ, який враховує різну специфіку тестування програмного забезпечення, що дозволяє підвищити точність прогнозування надійності ПЗ та забезпечити більш ефективне оцінювання результатів відповідного етапу;
- отримав подальший розвиток метод врахування контексту ПЗ при роботі з метрикою МТВФ, що передбачає збір, обробку та аналіз додаткових даних про програмне забезпечення, його архітектуру, мову програмування, зовнішні залежності та інші характеристики, які можуть впливати на надійність та результати тестування.

Практичне значення отриманих результатів полягає у наступному:

- удосконалена методика роботи з метрикою МТВФ може бути впроваджена в процес тестування програмних продуктів різних видів для забезпечення високої їх надійності;
- результати дослідження можуть бути використані при розробці тестових планів, стратегій та різних сценаріїв, оскільки вони дають змогу оптимізувати процес тестування та забезпечити більш ефективне виявлення та усунення дефектів програмного забезпечення;
- рекомендації щодо використання удосконаленої методики МТВФ можуть бути використані при навчанні спеціалістів з тестування ПЗ.

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної області

Життєвий цикл програмного забезпечення описує всі етапи, через які проходить програма від моменту зародження ідеї про її створення до повного виведення з експлуатації. Кожен з етапів є дуже важливим, оскільки успішність виконання одного безпосередньо впливає на досягнення мети виконання всіх інших етапів. Важливими критеріями оцінки роботи ПЗ для кінцевого користувача є якість та надійність [2]. Саме тому одним з етапів життєвого циклу виступає етап тестування розробленої програмної системи для забезпечення коректної роботи програми в подальшому.

Тестування програмного забезпечення – це процес, що спрямований на перевірку та оцінку якості програмного продукту з метою виявлення помилок, недоліків та відхилень від вимог [3]. Тестування включає в себе створення тестових сценаріїв, їх виконання, аналіз результатів та виправлення виявлених помилок. Тестування допомагає забезпечити надійність, функціональність та безпеку програмного продукту перед його впровадженням для повноцінного задоволення потреб та вимог користувачів.

Існують різні типи тестування програмного забезпечення [4]. Зокрема, одним з найважливіших видів є функціональне тестування. Головною метою функціонального тестування є визначення того, чи працює програма відповідно до специфікацій та функціональних вимог. Воно включає в себе перевірку функцій, операцій, дій та сценаріїв, які програма повинна виконувати. Функціональне тестування передбачає створення тестових сценаріїв, які описують, як програма повинна вести себе в певних умовах. Тестувальник виконує ці сценарії, подаючи різні вхідні дані та перевіряючи, чи програма надає очікуваний результат. В загальному під час даного типу тестування виконуються різноманітні види перевірок на предмет виявлення функціональних помилок та

недоліків з метою їх виправлення, забезпечуючи відповідність програмного забезпечення до вимог користувачів та бізнес-потреб.

Іншим видом перевірки програмного забезпечення є Unit-тестування, яке є одним із рівнів тестування програмного забезпечення та спрямоване на перевірку окремих компонентів або "одиниць" програмного коду. Основним завданням тестування на рівні одиниць є перевірка правильності та коректності роботи ізольованих частин програми, таких як функції, методи, класи або модулі. Дане тестування допомагає визначити, чи програмна одиниця виконує визначені функції і логіку без помилок. Тестування на рівні одиниць включає створення тестових випробувань, які перевіряють різні стани і варіанти використання одиниці. Тести можуть бути розроблені вручну або автоматизовані з використанням різних засобів та інструментів для тестування. У процесі перевірки на рівні одиниць інші частини програми, які взаємодіють з тестованою одиницею, імітуються або замінюються, щоб уникнути впливу зовнішніх факторів на результати тестування. В цілому даний вид тестування є ефективним способом перевірити правильність окремих частин програмного коду та забезпечити відповідність їх функціональним вимогам, покращити якість програми та знизити ризики помилок на ранніх етапах розробки.

Інтеграційні тести є ще одним видом перевірки працездатності програмної системи [5]. Тестування на рівні інтеграції – це тестування, яке спрямоване на перевірку взаємодії між різними компонентами або модулями програми. Даний вид перевірки має на меті виявлення та вирішення проблем, які можуть виникнути внаслідок взаємодії між різними елементами ПЗ. У процесі проведення інтеграційних тестів здійснюється перевірка того, що дані та функціональність передаються між компонентами належним чином. Сам процес тестування може бути побудований на основі реальних або потенційно можливих ситуацій в роботі програмного забезпечення. Також важливим аспектом в межах даного типу тестування є перевірка внутрішньої та зовнішньої інтеграції ПЗ, перша з яких включає в себе взаємодію компонентів всередині програми, друга представляє зв'язки з іншими програмними системами чи службами. Тестування на рівні

інтеграції є важливим кроком у забезпеченні належної взаємодії різних компонентів і систем, що допомагає суттєво покращити якість та надійність їх загальної роботи.

Тестування продуктивності є ще одним видом перевірки роботи програми. Тестування продуктивності є процесом, що спрямований на оцінку та вимірювання швидкодії, ефективності та інших аспектів продуктивності програмного забезпечення під навантаженням. Даний вид тестування важливий для забезпечення того, що програма працює задовільно при реальних умовах використання. В межах даного тестування можливе виділення ще декількох підвидів тестів, зокрема, тестування пропускну здатності, відгуку, стійкості, завантаження та масштабованості. Кожен з даних підвидів описує процес перевірки певного аспекту ПЗ з використанням різних особливостей конкретної програми. Тестування продуктивності включає створення навантаження, яке може бути реалістичним для використання програми. Таке навантаження може включати в себе введення різних вхідних даних, запитів та сценаріїв користувача. Тести запускаються зі збільшенням навантажувального ресурсу, спостерігаючи, як програма реагує на різні рівні навантаження. Результати тестування продуктивності аналізуються для виявлення вразливих місць ПЗ та покращення ефективності його роботи.

Системне тестування є ще одним із ключових етапів в процесі тестування програмного забезпечення. Даний вид перевірки представляє оцінку всієї системи як єдиного цілого після того, як було виконано інші види тестування, зокрема, тестування на рівні одиниць та інтеграційне тестування. Метою даного тестування є визначення відповідності цілого програмного продукту функціональним вимогам, специфікаціям та очікуванням користувачів, що включає в себе перевірку поведінки програми у реальних умовах використання. Системне тестування дає загальну оцінку про роботу програми. Для відтворення процесу тестування розробляються тестові сценарії, які відображають різні ситуації та дії користувачів, включаючи типові та нестандартні сценарії. Саме ці

сценарії допомагають визначити, чи програма працює належним чином та чи відтворює очікуваний функціонал в цілому.

Окрім вже вказаних типів тестувань, існують також деякі інші види випробувань. Зокрема, тестування на рівні приймальних тестів перевіряє відповідність програми вимогам користувача та приймання її клієнтом, тестування безпеки здійснює оцінку програми на предмет вразливості та виявлення потенційних загроз для її роботи, тестування сумісності перевіряє можливість роботи програмного забезпечення на інших пристроях, операційних системах, платформах тощо.

Процес тестування будь-якого типу відбувається у строгій послідовності та з дотриманням необхідних правил для отримання належної оцінки про стан програмної системи [6]. Забезпечення успішного результату тестування є першочерговим завданням даного етапу життєвого циклу ПЗ. Оцінка результатів тестування програмного забезпечення є важливою в процесі забезпечення якості продукту і прийняття рішення щодо його подальшого випуску кінцевим користувачам. Оцінка полягає в аналізі результатів тестів, ідентифікації помилок і визначенні, чи відповідає програмне забезпечення вимогам і очікуванням користувачів. Для контролю та оцінки результуючих даних процесу тестування можуть бути використані різні засоби, інструменти та числові показники. Зокрема, одним з важливих критеріїв для оцінки перебігу етапу тестування та його результатів виступають метрики тестування.

Метрика тестування – це вимірювальний показник або стандартизований спосіб вимірювання певного аспекту якості тестування програмного забезпечення. Метрики тестування використовуються для оцінки та контролю різних аспектів тестового процесу, що можуть включати загальні компоненти та окремі одиниці програми [7]. Вони допомагають розробникам визначити, наскільки ефективно проводиться тестування та як можна поліпшити якість програмного забезпечення на загальному рівні.

Оскільки метрик тестування можуть стосуватися різних аспектів ПЗ, то існують різні види даних числових показників. Одним з перших типів є метрики

покриття коду. Метрики покриття коду - це інструменти для вимірювання обсягу програмного коду, який був перевірений тестами. Вони надають інформацію про те, наскільки ефективно тестування охоплює код та які частини програми залишилися без тестування. Дані метрики допомагають ідентифікувати недоліки в тестуванні та фокусувати увагу на критичних частинах коду для поліпшення якості тестів і виявлення помилок.

Наступним видом серед числових критеріїв оцінки тестування є метрики якості. Метрики якості визначають рівень якості та ефективність самого процесу тестування. Вони оцінюють різні аспекти, включаючи кількість та серйозність виявлених помилок, час їх виявлення та виправлення, а також загальну стійкість програми під час тестування. Метрики якості допомагають розробникам визначити, наскільки якісно програмне забезпечення пройшло тестування та допомагають удосконалювати його якість в подальшому.

Метрики продуктивності в тестуванні програмного забезпечення є ще одним видом метрик. Вони вимірюють ефективність тестового процесу та ресурсів, які витрачаються на тестування. Також дані метрики оцінюють, скільки тестів виконується протягом певного часу, тривалість виконання тестів, споживання ресурсів (час, обладнання, працівники) та інші аспекти продуктивності. Метрики продуктивності метрики допомагають визначити ефективність тестового процесу та можливості для оптимізації, забезпечуючи швидший та більш продуктивний цикл тестування.

Ще одним загальним типом є метрики дефектів. Метрики дефектів використовуються для оцінки та керування якістю виявлених помилок або дефектів. Вони включають в себе інформацію про кількість, серйозність та стан дефектів, а також про час їх виявлення та виправлення. Метрики допомагають тестувальникам визначити прогрес у виправленні дефектів, вплив дефектів на якість продукту та забезпечують об'єктивну інформацію для прийняття рішень щодо пріоритетів виправлення та оптимізації процесу тестування.

Іншими важливим видом метрик є метрики надійності. Метрики стійкості в процесі тестування ПЗ визначають рівень надійності та стабільності програми під

час тестування. Вони оцінюють частоту виникнення відмов, збоїв та інших неполадок, пов'язаних з програмним забезпеченням, а також тривалість нормального функціонування без помилок. Метрики надійності допомагають виявити проблеми, які можуть впливати на коректну роботу програми та надають інформацію для подальшого вдосконалення її стійкості та надійності під час проведення процесу тестування.

Поряд з вказаними видами метрик існують деякі інші метрики, які також використовуються в процесі тестування ПЗ. До таких метрик належать: метрики ризиків, автоматизації та звітності. Метрики ризиків оцінюють ризики, пов'язані з тестуванням, і визначають, як ефективно здійснюється контроль над ними. Метрики автоматизації вимірюють відсоток автоматизованих тестів, час, який вони заощаджують у порівнянні з ручними тестами, та інші показники, що використовуються в процесі автоматизації тестування. Метрики звітності визначають, як ефективно тестові звіти та документація підготовлені та використовуються під час тестування.

Кожен тип метрик тестування має високий рівень значущості для керування та оцінки якості програмних систем. Метрики надають об'єктивну інформацію про стан тестового процесу, ризики та загальний рівень якості продукту. Також вони допомагають ідентифікувати проблеми та можливості для їх вирішення, визначають прогрес у тестуванні та сприяють прийняттю конкретних рішень в розробці програмного забезпечення.

В загальному етап тестування програмного забезпечення є дуже важливим при розробці будь-яких програмних продуктів, оскільки він визначає їхню якість, відповідність заявленим вимогам та потребам користувачів. Різні види тестувань виконують послідовний процес перевірки програми відповідно до попередньо встановлених норм та правил. Оцінка результатів тестування за допомогою різних засобів та критеріїв, включаючи метрики, представляє загальну картину про готовність та можливість подальшого введення розроблюваного програмного забезпечення в експлуатацію.

1.2 Основні поняття та визначення методу MTBF

В контексті даної кваліфікаційної роботи важливою є метрика середнього часу між відмовами (Mean Time Between Failures). Дана метрика відноситься до типу метрик надійності, що вимірюють показники безвідмовної роботи ПЗ.

Метрика середнього часу між відмовами (MTBF) – це метрика, яка визначає середню тривалість часу між відмовами або збоями в програмному забезпеченні, системі або в будь-якому її компоненті [8]. MTBF використовується для оцінки надійності та стійкості об'єкта та вказує на середній інтервал часу, протягом якого очікується безвідмовна робота до настання наступної відмови.

Для використання методу MTBF важливим є розуміння основних термінів та понять, які стосуються даної метрики. Серед них можна виділити такі: Відмова (Failure), Надійність (Reliability), Експлуатаційний період (Operational Period), Ремонт (Repair), Середній час на ремонт (MTTR, Mean Time To Repair), Неперервна робота (Availability) [9].

Відмова – це небажана подія, яка виникає, коли продукт, система або компонент не може виконувати свої функції згідно з вимогами. У випадку програмного забезпечення, відмова може бути результатом помилок в коді, неправильних налаштувань чи впливу зовнішніх чинників, таких як апаратні неполадки або проблеми з мережею.

Надійність – це характеристика, яка визначає, наскільки система здатна безвідмовно функціонувати протягом визначеного періоду часу або при виконанні певної кількості операцій. Метрика MTBF є основним показником, що характеризує надійність системи.

Експлуатаційний період – це час від початку використання системи до моменту виведення з експлуатації або до проведення запланованого технічного обслуговування. Розрахунок MTBF визначається відношенням експлуатаційного періоду до загальної кількості відмов, що виникли протягом цього періоду.

Ремонт – це процедура відновлення працездатності системи після виникнення відмови. Даний процес може включати в себе виправлення помилок у

програмному коді, заміну апаратних компонентів або налаштування конфігурації. Залежно від ситуації, ремонт може проводитися негайно після виявлення відмови або під час запланованого технічного обслуговування.

Середній час на ремонт (MTTR) – це статистичний показник, який вказує на середній час, який необхідний для відновлення роботи системи після виникнення відмови. MTTR використовується разом із метрикою середнього часу між відмовами (MTBF) для оцінки як надійності системи, так і для планування обслуговування та виправлення проблем.

Неперервна робота – це властивість системи, яка визначає значення часу, протягом якого система готова та доступна для виконання своїх функцій. Для обчислення неперервної роботи використовується відношення MTBF до суми MTBF та MTTR.

Метрика середнього часу між відмовами допомагає інженерам та розробникам визначити ступінь надійності системи та передбачити, скільки часу може пройти між відмовами. Високе значення MTBF вказує на стійкість і надійність системи, тоді як низьке значення може вказувати на проблеми, які потребують виправлення. MTBF особливо важлива для критичних систем, де надійність грає важливу роль.

Ключовими подіями для вимірювання даної метрики виступають відмови в роботі програми. Основний показник метрики – це значення часу, що вказує на тривалість безпомилкової роботи програмного забезпечення. Схематичне позначення метрики середнього часу між відмовами подане на рисунку 1.1.

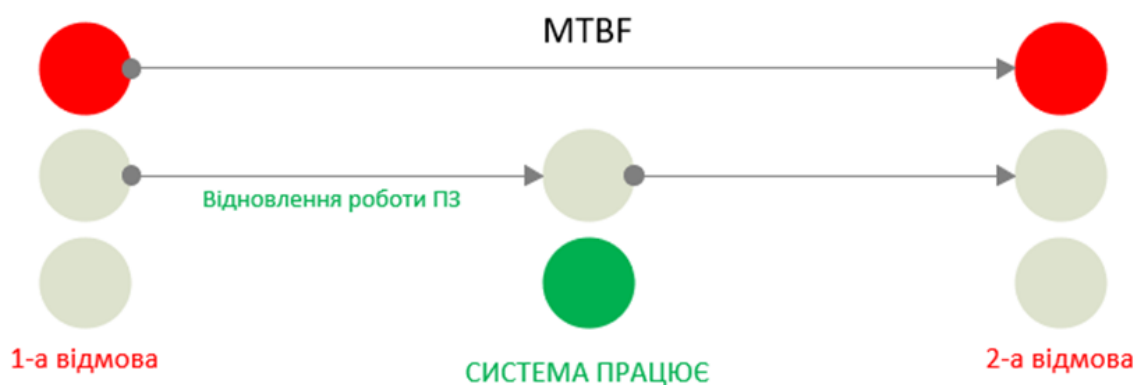


Рисунок 1.1 – Метрика середнього часу між відмовами

Метрика середнього часу між відмовами виконує різні завдання, які так чи інакше позитивно впливають на загальний процес тестування програмного забезпечення [10]. Серед основних систематизованих цілей застосування даної метрики можна виділити наступні.

1. Оцінка надійності. Метрика MTBF допомагає оцінити рівень надійності системи в цілому або окремого її компоненту. Якщо значення показника MTBF є високим, то надійність системи перебуває на високому рівні і навпаки.

2. Прогнозування відмов. MTBF дозволяє прогнозувати, наскільки довго система може функціонувати без серйозних відмов або неполадок. Дане завдання є важливим для планування обслуговування та профілактичних заходів програмних систем.

3. Виявлення проблем. Низьке значення MTBF може свідчити про проблеми в якості програмного забезпечення, апаратного забезпечення або в процесі розробки. Згідно цього метрика допомагає виявити та виправити дефекти та помилки в роботі ПЗ та апаратної частини.

4. Планування ресурсів. Метрика MTBF допомагає визначити, як часто потрібно виділяти ресурси на обслуговування певних компонентів програмного забезпечення, що дозволяє раціонально використовувати ресурси та уникати несподіваних відмов.

5. Покращення продукту. MTBF надає інформацію для удосконалення програм, зокрема, для розробки більш надійних та стійких систем.

6. Моніторинг в реальному часі. Під час роботи системи MTBF може використовуватися для моніторингу та виявлення можливих відмов або погіршення надійності в реальному часі.

Виконання цілей метрики MTBF веде за собою до покращення надійності та стійкості до помилок в програмному забезпеченні [11]. Виходячи з цього, дана метрика має позитивні сторони її використання, які є такими:

– MTBF дозволяє оцінити, наскільки стійкою є система або компонент по відношенню до відмов або можливих збоїв;

- значення МТBF вказує, як часто потрібно проводити технічне обслуговування системи;
- МТBF дозволяє порівнювати різні програмні системи чи компоненти щодо їхньої надійності;
- визначення МТBF допомагає ефективно використовувати час та ресурси шляхом спрямування їх туди, де вони найбільше потрібні;
- збільшення МТBF сприяє покращенню досвіду користувачів завдяки більш надійному та стабільному продукту.

В загальному метрика середнього часу між відмовами виступає необхідною складовою процесу тестування. Результат її розрахунку безпосередньо вказує на рівень надійності та загальної якості розробленого програмного забезпечення. Використання даної метрики дає можливість підвищити час безперебійної роботи програмної системи та зменшити кількість потенційно можливих її помилок.

1.3 Методи та техніки розрахунку МТBF

Метрика середнього часу між відмовами є основним показником надійності системи та стійкості до виникнення помилок в її роботі. Для отримання результатів роботи метрики використовується декілька методів та технік її розрахунку [12]. Зокрема, основними методами обчислення метрики середнього часу між відмовами є такі:

- історичний метод;
- метод на основі статистичних розподілів;
- метод на основі тестування;
- метод на основі аналізу надійності;
- метод експертних оцінок.

Історичний метод розрахунку середнього часу між відмовами (МТBF) базується на аналізі даних про відмови та ремонт, які вже сталися в роботі програмного забезпечення протягом певного періоду часу. Даний метод дуже

простий та вимагає наявності достатньої кількості історичних даних про відмови для його застосування.

Для розрахунку показника МТBF за історичним методом спочатку необхідно зібрати історичні дані про помилкові збої в роботі ПЗ та інформацію про їх усунення. Відповідні дані мають включати час виникнення кожної відмови та час, який був витрачений на її виправлення. На наступному кроці потрібно впевнитись, що дані є унікальними та правильно відображають час виникнення відмов та час ремонту. Після цього відбувається розрахунок показника середнього часу між відмовами. Загальна формула розрахунку є такою:

$$MTBF = \frac{\text{Загальна кількість робочих годин}}{\text{Загальна кількість відмов}} \quad (1)$$

Основними компонентами формули (1) є показники загальної кількості робочих годин та кількості відмов. Перший числовий показник вказує на кількість годин, коли робота програмного забезпечення була успішною та безпомилковою. При визначенні фактичного значення даного показника не враховується час виконання відладки ПЗ та приведення його до робочого стану повторно. Значення загальної кількості відмов вказує на кількість відмов в роботі програмного забезпечення за певний проміжок часу. Результуючий показник МТBF представляє кінцеве значення проведення вимірювання метрикою середнього часу між відмовами за історичним методом.

Наступний метод розрахунку МТBF, метод на основі статистичних розподілів, використовується для точнішої оцінки середнього часу між відмовами системи шляхом застосування певного статистичного розподілу, який відображає характер відмов. Даний підхід дозволяє більш точно моделювати час між відмовами та прогнозувати МТBF на основі характеристик та властивостей певного типу розподілу.

Для початку розрахунку метрики МТBF за методом розподілів спочатку необхідно визначити відповідний статистичний розподіл, який найкраще відповідає природі відмов системи. Його вибір може базуватися на припущеннях

про природу відмов системи або аналізі даних попередньо вказаного історичного методу. Після цього потрібно провести оцінку параметрів вибраного статистичного розподілу на основі наявних даних про відмови. Оцінка параметрів дозволяє налаштувати вибраний розподіл таким чином, щоб він якнайкраще відповідав спостережуваним даним. На наступному етапі роботи методу розраховується саме значення метрики МТВФ. Воно може бути розраховане за допомогою інструментів обраного статистичного розподілу або за іншими формулами, зокрема формулою історичного методу. Останнім кроком роботи методу є перевірка того, наскільки добре розподіл відповідає спостережуваним даним про відмови. Результат перевірки буде вказувати на адекватність моделі, яка може бути використана для моделювання та прогнозування часу між відмовами в майбутньому.

Інший підхід для визначення показника МТВФ – метод, що заснований на процесі тестування – використовується для оцінки надійності системи шляхом проведення спеціальних тестів з метою виявлення відмов. Основна ідея полягає у випробуванні системи в реальних умовах та вимірюванні при цьому часу між збоями в роботі ПЗ.

Перший крок використання цього методу – визначення конкретних тестових сценаріїв, які відображають реальне використання системи. Відповідні сценарії повинні включати навантаження та умови, які максимально відповідають роботі системи в реальному робочому середовищі. Після цього виконується процес тестування, в ході якого фіксується час виникнення відмови під час кожного тесту. Далі застосовується формула, яку використовують в історичному методі, для розрахунку загального показника МТВФ. Останній етап передбачає аналіз результатів тестування, враховуючи МТВФ та інші показники надійності, що допомагає виявити можливі слабкі місця системи та визначити шляхи для поліпшення її надійності.

Ще один спосіб розрахунку метрики середнього часу між відмовами заснований на надійності програми та її компонентів. Даний метод передбачає

збір даних про відмови елементів системи та використання різних аналітичних технік для оцінки МТBF.

Для роботи даного методу спочатку потрібно зібрати дані про відмови програмного забезпечення. Така інформація може бути зібрана будь-яким з доступних способів. Головними даними виступають час безвідмовної роботи та час відмов програмного забезпечення. Далі проводиться аналіз надійності системи та її окремих компонентів на основі зібраних даних. Наступним кроком є розрахунок показника МТBF шляхом комбінування надійності компонентів, використовуючи формули або аналітичні моделі. На останньому кроці відбувається перевірка отриманих результатів, порівнюючи їх з фактичними даними про відмови системи. Перевірка може включати порівняння прогнозованого МТBF з фактичними спостереженнями або порівняння прогнозу надійності компонентів з реальним значенням даного показника.

Метод експертних оцінок є останнім серед наявних способів розрахунку показника МТBF. Даний метод використовується для оцінки середнього часу між відмовами системи на основі експертних знань та досвіду фахівців. Особливістю даного методу є те, що він застосовується тоді, коли недостатньо даних про відмови або коли система є новою і немає історичних даних. Принцип його роботи заснований на суб'єктивному сприйнятті роботи програмного забезпечення в цілому та окремих його компонентів.

На початку використання даного методу для розрахунку показника МТBF відбувається формування експертної групи, що володіє необхідною інформацією про роботу програмного забезпечення. Далі сформована група розглядає різні аспекти системи, які можуть впливати на її надійність, та визначає параметри, які можуть бути використані для оцінки відповідної характеристики системи. На наступному кроці група експертів здійснює оцінювання середнього часу між відмовами програмної системи, що може бути представлено числовими значеннями або діапазонами. На останніх кроках роботи методу відбувається аналіз встановлених оцінок часу та здійснюється їх перевірка за допомогою

додаткових методів, таких як порівняння з існуючими даними або використання інших незалежних експертних оцінок.

Кожен з методів розрахунку показника середнього часу між відмовами є ефективним та по-своєму формулює підхід до знаходження потрібного значення. Важливо враховувати те, що різні методи можуть бути використані в різних ситуаціях в залежності від особливостей роботи програмного забезпечення, природи виникнення відмов та доступності даних для проведення розрахунку. Також варто враховувати те, що результати роботи вказаних методів можуть різнитися, тому важливо розуміти можливі їх обмеження та застосування для коректної інтерпретації МТВФ, а також використання цієї метрики для покращення якості програмного забезпечення.

1.4 Використання МТВФ у тестуванні програмного забезпечення

Метрика середнього часу між відмовами (МТВФ) дає загальне представлення про рівень стабільності та стійкості до помилок в роботі програмного забезпечення. Використання показника МТВФ можливе на всіх етапах розробки ПЗ для виявлення вразливих місць програми, визначення загальних пріоритетів розробки та планування ресурсів [13]. Метрика може бути корисною в багатьох аспектах розробки та тестування програмних систем.

Одним з перших аспектів використання показника МТВФ є його важлива роль при плануванні тестування. Використання середнього часу між відмовами (МТВФ) для планування тестування програмного забезпечення важливо для забезпечення надійності та якості продукту.

МТВФ допомагає визначити, як часто необхідно проводити тестові цикли для різних частин програми. Для компонентів з високим МТВФ, можна обирати рідше тестування, тоді як для менш надійних компонентів – частіше. Функції або компоненти з низьким МТВФ можуть стати пріоритетними для тестування. Така діяльність допоможе ідентифікувати та усунути можливі проблеми, що

призводять до відмов. Також дана метрика здатна визначити, як багато тестів потрібно виконати для кожного компонента або функції, щоб впевнитися в їх надійності. План тестування може бути скорегований відповідно до MTBF: надійні компоненти можуть потребувати менше тестових сценаріїв, тоді як менш стабільні можуть вимагати більшої кількості тестових кейсів. Показник метрики може використовуватися для оцінки ризиків відмов та їх впливу на користувачів. Важливо зосередити увагу на компонентах з низьким MTBF та вжити додаткові заходи для їх тестування.

Використання MTBF у плануванні тестування допомагає оптимізувати ресурси, забезпечити високу надійність програмного забезпечення та задовольнити вимоги користувачів.

Наступним аспектом для використання метрики середнього часу між відмовами є її значимий вплив на безпосереднє виконання тестів. Застосування показника при виконанні тестів може бути корисним для визначення значень показників надійності та планування тестових активностей.

При проведенні тестів MTBF може допомогти визначити, які компоненти або функції системи потребують найбільшої уваги під час даного етапу. В процесі перевірки компонентів можуть бути створені тестові сценарії, які відображають реальні умови експлуатації продукту на основі MTBF. Також показник може слугувати для визначення того, як часто необхідно проводити тести над окремими компонентами або системою в цілому. Метрика допомагає виявити та протестувати елементи з низьким показником надійності, що дозволяє ідентифікувати проблемні частини системи та виправити їх до випуску продукту. Порівнюючи результати тестування з MTBF, можна визначити, чи співпадають вони з очікуваними значеннями. Якщо тестові відмови відбуваються надто часто або надто рідко в порівнянні з MTBF, це може сигналізувати про проблеми або надмірну консервативність у розрахунках. Використання показника також може дати чітке розуміння того, наскільки ефективним був тестовий процес у виявленні відмов. Порівняння фактичних відмов з MTBF може допомогти зрозуміти, чи були тести достатньо точними та вимогливими.

Інший аспект використання метрики MTBF є її важливість під час здійснення оцінки результатів тестування. Показник дає можливість порівняти фактичні результати тестування з очікуваним. Якщо MTBF вказує на те, що система має витримувати певний період без відмови, результати тестів повинні вказувати на подібну надійність. Порівняння може виявити розходження між очікуваними та фактичними показниками. Коли відмови виникають під час тестування, важливо аналізувати їх причини та докладати зусиль для виправлення проблем, оскільки це важливо для підвищення MTBF та загальної надійності продукту. Використовуючи результати тестування та інформацію про MTBF, можлива розробка плану подальших дій щодо виправлення виявлених проблем та підвищення часу безвідмовної роботи програмного забезпечення. План може включати у себе виправлення помилок, підвищення надійності коду та проведення додаткових тестів. Результати тестів та аналіз MTBF можуть служити як важлива інформація для запобігання відмовам в майбутньому. Вони можуть допомогти ідентифікувати вразливі та слабкі місця програмної системи, щоб в подальшому вдосконалити її надійність та підвищити час безвідмовної роботи.

Метрика середнього часу між відмовами має позитивний вплив на постійне поліпшення якості програмного забезпечення, що є ще одним аспектом її застосування. Шляхом аналізу MTBF можна виявляти тенденції та зміни в стабільності програмного забезпечення протягом різних версій або ітерацій розробки. На основі даних MTBF команда розробників може сфокусуватися на вдосконаленні та оптимізації вразливих компонентів програми. Впровадження нових технік розробки, тестування та управління може допомогти збільшити MTBF, а, отже, надійність ПЗ. Крім того, аналіз MTBF дозволяє виявляти зміни у якості програмного забезпечення під час розробки та після виходу нових версій. За результатами аналізу можна приймати рішення про внесення корекцій та покращень для підвищення рівня якості програмного забезпечення.

Загалом, застосування MTBF під час тестування програмного забезпечення дозволяє не лише визначати середній час між відмовами в його роботі, а також може сприяти покращенню якості продукту, виявленню проблем, пов'язаних із

надійністю та стабільністю його функціонування, а також пошуку стратегій вирішення відповідних проблемних моментів. Використання всіх аспектів метрики на практиці здатне призвести до підвищення задоволеності користувачів, зменшення витрат на підтримку продукту та сприяти більш ефективному використанню ресурсів команди розробки.

1.5 Обмеження та недоліки методу МТВФ у тестуванні ПЗ

Метод МТВФ надає можливість здійснення вимірювання, прогнозу та управління якістю програмного забезпечення. В нормальних умовах оцінка показника дає можливість впливати на стан ПЗ, збільшуючи при цьому його час його успішної та безвідмовної роботи. Проте інколи виникають ситуації, за яких розрахунок показника метрики може нести хибний характер [14]. Відповідно до цього існують певні обмеження та недоліки метрики середнього часу між відмовами програмної системи.

Одним із суттєвих недоліків методу є неможливість його повноцінного використання при недостатності історичних даних про ПЗ. Недостатність або недоступність даних може бути спричинена різними чинниками. Одним з таких випадків може бути ситуація, коли програмне забезпечення або продукт є абсолютно новим і не має історії використання. Деякі проекти мають обмежені ресурси або час для спостережень та збору даних про відмови. У деяких випадках користувачі можуть не повідомляти про відмови або інциденти, що стосуються програмного забезпечення, або ця інформація може бути недоступною. Якщо ж програмне забезпечення має дуже низьку частоту відмов, то для отримання значущих даних може знадобитися більше часу, ніж в звичайних умовах.

Вказані обставини можуть ускладнити використання МТВФ в тестуванні програмного забезпечення і вимагати додаткових заходів для оцінки надійності системи. У таких випадках може бути корисним розглядати інші метрики та

методи оцінки якості ПЗ, а також застосовувати різні заходи для збору даних і покращення процесу вимірювання надійності програмного забезпечення.

Іншим обмеженням застосування метрики середнього часу між відмовами є непостійність часу між відмовами. Важливо розуміти, що MTBF є середнім значенням, і воно не враховує різні коливання у часі між відмовами. В деяких випадках відмови можуть відбуватися з нерегулярними інтервалами, що ускладнює аналіз надійності програмного забезпечення на основі MTBF. Відповідна непостійність у часі між збоями в роботі ПЗ може бути спричинена різними факторами, такими як зміни в умовах експлуатації, внесення змін у програмний код або зміни в вимогах користувачів. В таких випадках користування лише MTBF для оцінки надійності може призвести до неточних результатів та не врахувати суттєві відхилення у часі між відмовами. Тому під час використання MTBF важливо бути обережним та враховувати інші аспекти, які впливають на стабільність та надійність системи. Додаткові метрики та методи можуть допомогти більш точно оцінити надійність програм у випадках, коли MTBF не є інформативним.

За відсутності інформації про причини відмов в роботі програми метод MTBF також може працювати некоректно, що є ще одним обмеженням його застосування. Метрика надає уявлення про те, як часто відбуваються відмови та середній час між ними, але вона не розкриває причини появи відповідних збоїв. Відповідне обмеження стає особливо важливим у випадках, коли для забезпечення якості програмного забезпечення необхідно знати природу відмов для здійснення відповідних виправлень. Без інформації про причини відмов розробникам та тестувальникам може бути важко визначити, які саме аспекти програмного забезпечення потребують вдосконалення. Завдяки аналізу причин відмов можна ідентифікувати конкретні проблеми в програмному коді, а також визначити, які складові системи є найбільш вразливими. Такий аналіз допомагає розробникам вдосконалювати програмне забезпечення та запроваджувати покращення для забезпечення більшої стабільності та надійності. Виходячи з цього варто усвідомлювати, що поряд з методом MTBF потрібно використовувати

деякі інші методи аналізу відмов, які надають інформацію про їх причини, що дозволить покращити динаміку удосконалення якості ПЗ.

Ще одним обмеженням використання метрики МТВФ є її непридатність до деяких видів програмного забезпечення. Показник МТВФ може бути менш придатним для оцінки надійності програмного забезпечення, що характеризується високою складністю, динамічною поведінкою або взаємодією зі значною кількістю зовнішніх компонентів. МТВФ фокусується на середньому часі між відмовами, але не завжди може зрозуміло враховувати всі можливі сценарії та взаємодії, які відбуваються в складних програмах. Динамічна поведінка програми, яка змінюється в залежності від ситуації, може також створювати виклики для МТВФ. Оскільки МТВФ базується на середніх значеннях, воно може не враховувати коливання та зміни у надійності, які можуть бути типовими для динамічних систем. Коли програмне забезпечення взаємодіє з багатьма зовнішніми компонентами або іншими системами, визначення МТВФ може бути ускладненим. В таких випадках важливо враховувати, як ці компоненти взаємодіють між собою та як це впливає на надійність всього продукту.

Обмеження використання показника середнього часу між відмовами може також полягати у недостатності врахування відмов, що не впливають на роботу системи. МТВФ спрямований на аналіз відмов, які призводять до припинення роботи системи. Однак існують випадки, коли деякі відмови мають обмежений вплив на функціонування системи або автоматично відновлюються без втрат продуктивності. У цих ситуаціях МТВФ може надавати недостатньо повну картину щодо надійності ПЗ, оскільки він не враховує такі випадки відмов.

Беручи до уваги перелічені обмеження та недоліки методу МТВФ, важливо враховувати його можливості та здійснювати обережне використання при оцінці результатів тестування програмного забезпечення. Потрібно також розглядати можливість використання альтернативних методів та підходів для оцінки надійності програмного забезпечення, які можуть краще враховувати специфічні властивості ПЗ і процесу його тестування.

1.6 Постановка задачі

В сучасному світі роль програмного забезпечення продовжує швидко зростати, супроводжуючись високими вимогами до якості та надійності програмних продуктів. Відповідно до цього актуальність вирішення проблеми оцінювання результатів тестування програмного забезпечення стає все більш очевидною та важливішою, особливо в контексті ефективного використання метрик якості, таких як середній час між відмовами. Метод MTBF слугує одним з основних показників надійності програмних систем. Принцип роботи, головні цілі застосування та позитивні сторони використання метрики в межах тестування програмного забезпечення ще раз підкреслюють важливість та необхідність її застосування для загальної оцінки роботи ПЗ. Проте, зважаючи на обмеження та недоліки, існує потреба пошуку нових підходів до роботи з метрикою середнього часу між відмовами для проведення успішного тестування ПЗ та ефективного оцінювання його результатів. З урахуванням цих обставин проведення дослідження щодо вдосконалення підходів до роботи з метрикою MTBF є змістовним і актуальним завданням.

Об'єктом дослідження є процес тестування програмного забезпечення, а предметом дослідження виступає методика роботи з метрикою середнього часу між відмовами (MTBF).

Метою дослідження є розроблення удосконаленого методу роботи з метрикою середнього часу між відмовами для забезпечення ефективного оцінювання результатів тестування програмного забезпечення.

Для досягнення мети дослідження було виділено наступні завдання кваліфікаційної роботи:

- виконати детальний аналіз предметної області;
- проаналізувати наявні методи роботи з метрикою середнього часу між відмовами (MTBF) та їх застосування у процесі тестування ПЗ;
- визначити особливості наявних методів та сформулювати пропозиції для їх покращення;

- визначити вимоги та розробити удосконалену методику роботи з метрикою середнього часу між відмовами;
- провести експериментальне дослідження та виконати оцінку ефективності застосування удосконаленої методики;
- описати рекомендації щодо використання удосконаленої методики МТВФ для різних видів програмного забезпечення;
- оцінити перспективи подальшого розвитку та удосконалення методики роботи з метрикою МТВФ.

2 АНАЛІЗ СУЧАСНИХ ПІДХОДІВ ДО УДОСКОНАЛЕННЯ МЕТОДУ РОБОТИ З МЕТРИКОЮ МТВФ

2.1 Аналіз сучасних підходів до покращення методу МТВФ

Використання метрики середнього часу між відмовами базується на різних показниках та роботі програмного забезпечення. Прогрес у сфері розробки ПЗ призводить до появи все більших програмних систем, показники якості та надійності яких стає важче моніторити та контролювати. Зважаючи на це, метрика МТВФ як один з інструментів контролю різних характеристик ПЗ повинна завжди працювати на високому рівні для забезпечення ефективного оцінювання результатів програмного забезпечення. Саме тому дослідники та практики постійно працюють над пошуком різних рішень та використовують в процесі тестування вже наявні сучасні підходи покращення методу МТВФ. Серед існуючих підходів є декілька, кожен з яких має власні особливості, додатковий інструментарій та специфіку використання.

Один із перших підходів для покращеного використання метрики МТВФ полягає в застосуванні статистичних методів [15]. Застосування даного підходу спрямоване на поліпшення результатів розрахунків показника МТВФ та підвищення їх точності. Перший крок до удосконалення в межах даного підходу включає збір та аналіз даних щодо відмов, застосовуючи статистичний аналіз. Такий аналіз може виявити закономірності та тренди в часі між відмовами. Збір даних про відмови можна здійснювати через систему журналів відмов або вимірювання інтервалів часу між ними для виявлення факторів, що впливають на значення показника МТВФ.

Застосування статистичних методів може вдосконалити роботу метрики середнього часу між відмовами шляхом використання різних видів розподілів, які можуть включати регресійний аналіз, кореляційний аналіз, аналіз часових рядів та інші. Кожен розподіл може бути корисним для моделювання відмов та обчислення показника метрики МТВФ. Важливим аспектом при цьому є

правильний вибір розподілу з усіх доступних, оскільки від цього залежить рівень відповідності даним щодо відмов в конкретній системі.

Статистичні методи аналізу надійності можуть бути використані для відповідної оцінки загальної системи та її компонентів. Аналіз такого типу дозволяє виявити вразливі місця програмного забезпечення та ідентифікувати фактори, що сприяють відмовам. Отримуючи результати аналізу надійності, можна приймати заходи для поліпшення МТBF шляхом зміни побудови системи або часткової заміни її основних структурних елементів.

Ще одним зі способів покращення є використання статистичних методів прогнозування. Спираючись на передбачення результатів, можна дістати інформацію про майбутні значення МТBF на основі уже наявних історичних даних. Відповідне прогнозування дозволяє передбачати тенденції та визначати майбутні необхідні ремонтні роботи або профілактику для збереження вимог щодо надійності системи.

Важливими для покращення МТBF також є методи випробування та валідації, що мають статистичний аспект. Вони можуть бути використані для оцінки надійності системи ще на етапі розробки програмного забезпечення, що включає проведення статистичних експериментів, випробувань з навантаженням та аналізу отриманих даних. Результати цих методів можуть бути застосовані для виявлення дефектів, впливу параметрів на надійність та для покращення МТBF перед введенням ПЗ в експлуатацію.

Загалом, використання методів статистики дозволяє більш об'єктивно оцінювати роботу метрики середнього часу між відмовами та здійснювати її покращення. Виявлення факторів, що впливають на надійність системи, та розробка стратегії для поліпшення надійності та безвідмовної роботи є одним з головних завдань даних методів.

Наступним сучасним підходом для покращення роботи з метрикою середнього часу між відмовами є використання машинного навчання [16]. МН є корисним для розрахунку та прогнозування показника МТBF, оскільки може

виявити складні залежності між вхідними параметрами та MTBF, а також врахувати нелінійність та взаємодію між компонентами системи.

На першому кроці застосування машинного навчання відбуваються підготовчі дії по збору інформації. Згідно до цього здійснюється збір та аналіз великого обсягу даних про відмови певного програмного забезпечення. Відповідні дані враховують різноманітні фактори програмної системи, такі як характеристики компонентів, умови експлуатації, технічні характеристики ПЗ, дата відмови, тип відмови, обставини, в яких вона відбулася та інші.

На наступному етапі відбувається вибір моделей машинного навчання, які найкраще підходять для аналізу даних про відмови в роботі ПЗ. До таких моделей можна віднести нейронні мережі, лінійні моделі, дерева рішень, алгоритми класифікації, регресії або кластеризації тощо. Вибір моделей залежить від особливостей даних та типу задачі.

Після збору інформації та вибору необхідної моделі машинного навчання відбувається її тренування на основі історичних даних. Важливим аспектом перед початком навчання моделі є вибір набору вхідних функцій та гіперпараметрів, що відображають важливі властивості та конфігурації програмної системи та її компонентів. Під час тренування модель буде шукати залежності між вхідними даними та MTBF, адаптуючи свої ваги та параметри для найкращої прогнозної точності. Після навчання важливим аспектом є перевірка отриманої моделі на незалежних даних, щоб оцінити її прогнозну точність та здатність узагальнення. Готова навчена модель машинного навчання може бути використана для прогнозування MTBF на основі нових даних, враховуючи взаємозв'язки між різними факторами, що впливають на надійність програмної системи.

Результати прогнозування можуть бути використані для аналізу та оптимізації якості та надійності програмного забезпечення. Згідно цього можуть бути виявлені фактори, що негативно впливають на MTBF, та прийняті заходи для їх усунення. Також можна здійснювати розробку стратегії профілактики та обслуговування для забезпечення більш довгого періоду безвідмовної роботи системи. Машинне навчання дозволяє системі адаптуватися

до змін в умовах використання програмного забезпечення. Якщо з'являються нові фактори, що впливають на надійність, моделі можуть бути оновлені, щоб враховувати ці зміни.

Загалом, застосування машинного навчання для покращення методу MTBF дозволяє більш об'єктивно оцінювати надійність програмного забезпечення і приймати інформовані рішення щодо покращення його безвідмовної роботи. МН здатне здійснювати точні прогнози часу успішної роботи ПЗ та виявляти складні залежності між його компонентами, які не можуть бути розпізнані за допомогою традиційних аналітичних методів. Однак, важливо мати на увазі, що машинне навчання також потребує великої кількості якісних даних та уважної обробки для досягнення надійних результатів.

Наступний загальний сучасний підхід до покращення метрики середнього часу між відмовами полягає у застосуванні моделювання програмного забезпечення на основі архітектури [17]. Відповідний підхід заснований на використанні детальної схеми побудови програмної системи на загальному рівні та окремих її компонентів.

Для початку використання підходу важливим є визначення архітектурної концепції програмного забезпечення, що включає вивчення його компонентів, взаємодій та залежностей між ними, а також інших ключових аспектів, які визначають структуру системи. На початковому етапі також можливий попередній розгляд вразливих місць та точок відмови в архітектурі.

Після вивчення архітектури відбувається визначення ризиків. Відповідний процес включає пошук потенційних ризиків та факторів, які можуть спричинити відмови в програмному забезпеченні. Такими небажаними аспектами можуть бути некоректні дані введення, недостатня обробка помилок, нестабільні мережеві з'єднання тощо.

Безпосереднє моделювання відмов на основі архітектури застосовується з використанням усіх попередньо встановлених даних. Різні методи моделювання використовуються для ідентифікації можливих відмов та їх причин. До таких

методів належать діаграми причинно-наслідкових зв'язків, моделі, що засновані на блоках, мережах Петрі, моделях залежності та інших формальних показниках.

Після створення моделей відмов на основі результатів аналізу і моделювання відбувається розробка та впровадження заходів з покращення. Зокрема, можуть бути застосовані різні заходи, такі як вдосконалення контролю введення, розширення механізмів обробки помилок, покращення алгоритмів та оптимізація роботи з пам'яттю тощо. Внесення змін в програмне забезпечення залежить від специфіки його роботи та даних моделі про потенційні помилки в ході виконання основних його функцій. Після будь-якої зміни програмної системи необхідно провести відповідне тестування, щоб переконатися, що внесені покращення дійсно сприяють збільшенню MTBF. Важливим аспектом є застосування розширеного тестування ПЗ, щоб покрити якомога більше можливих сценаріїв використання та забезпечити стабільну роботу програми. Після цього необхідно встановити різні механізми моніторингу програмного забезпечення для відстеження відмов та збору даних про продуктивність. Аналіз відповідної інформації може бути використаний для подальшого вдосконалення архітектури ПЗ та процесу розробки.

Моделювання на основі архітектури допомагає розробникам системи визначити можливі проблеми, які можуть вплинути на MTBF, та приймати заходи для їх вирішення. Такий підхід сприяє підвищенню рівня надійності програмного забезпечення, яке відповідає вимогам щодо MTBF та забезпечує стабільну роботу програмної системи.

Ще одним із сучасних підходів до покращення методу роботи з метрикою середнього часу між відмовами є використання емпіричних даних [18]. В основі даного підходу покладено ідею застосування різноманітної інформації про успішну роботу програмного забезпечення та час, коли система знаходиться в неробочому стані, для визначення показника метрики.

На початковому етапі використання підходу до покращення відбувається збір відповідних даних, що можуть включати інформацію про час відмови, причину відмови, відповідність до архітектурних компонентів тощо. Терміни

збору інформації повинні бути якомога більшими, в той же час сама інформація має бути чітко систематизована.

Після початкового етапу необхідно виконати аналіз зібраних даних, щоб виявити ключові тенденції та визначити часті причини відмов, фактори їх появи, а також компоненти, які найчастіше відмовляють, та їх взаємозв'язки. Відповідний процес допоможе зрозуміти, де саме потрібно зосередитися для покращення показника метрики середнього часу між відмовами.

Емпіричні дані можуть бути використані для виявлення паттернів або повторюваних проблем, що призводять до відмов. Такими паттернами можуть бути конкретні послідовності дій, окремі компоненти або залежності між ними. Визначення цих паттернів допоможе усунути відмови та покращити MTBF.

На наступному кроці на основі аналізу емпіричних даних та встановлених паттернів відбувається внесення змін до проблемних апаратних компонентів або кодової частини програми. Після застосування змін необхідно використати доступні техніки для спостереження за новими даними щодо відмов. Аналіз та порівняння цих даних з попередніми покликаний для забезпечення оцінювання ефективності внесених змін та встановлення подальших дій для покращення показника MTBF.

Емпіричні дані грають ключову роль у виявленні проблем, покращенні надійності та збільшенні MTBF програмного забезпечення. Важливо постійно збирати дані та оновлювати процес покращення на основі нових відомостей, щоб забезпечити стабільну та надійну роботу ПЗ.

Останнім з сучасних підходів покращення MTBF є розробка адаптивних методів оцінки надійності програмного забезпечення. Адаптивні методи дозволяють враховувати змінні умови та контекст роботи програмного забезпечення для більш точної оцінки його надійності.

Для застосування даного підходу важливою складовою виступає достатньо великий обсяг емпіричних даних про відмови програмного забезпечення в різних умовах роботи. Такими даними може бути інформація про роботу в різних операційних системах, апаратних платформах, мережевих середовищах тощо.

Аналіз цих даних виступає ефективним процесом для виявлення залежностей між умовами роботи та відмовами. Для визначення факторів, які мають найбільший вплив на надійність програмного забезпечення, використовуються статистичний аналіз та машинне навчання. До таких факторів може належати використання пам'яті, завантаження процесора, мережеві навантаження, типи введення тощо. Необхідно визначити, які фактори мають найсильніші кореляції з відмовами. Далі на основі зібраних даних та визначених факторів потрібно розробити адаптивні моделі для оцінки надійності програмного забезпечення. Для цього застосовуються різні методи побудови моделей, які дозволяють враховувати змінні умови та контекст роботи. Після цього відбувається навчання моделі за вже відомим сценарієм для прогнозування ймовірності відмови на основі поточних умов роботи. На останньому кроці застосування адаптивних методів будується механізм моніторингу програмного забезпечення для збору поточних даних про умови роботи та відмови та їх порівняння з прогнозованими. Якщо спостерігається розбіжність, то модель оновлюється, враховуючи нові дані. В залежності від прогнозованої надійності системи здійснюється розробка адаптивних стратегій на основі її оцінки для збільшення часу безвідмовної роботи програмного забезпечення в різноманітних умовах.

Адаптивні методи оцінки надійності програмного забезпечення дозволяють більш точно прогнозувати надійність в реальному часі, враховуючи змінні умови роботи. Їх використання може сприяти покращенню методу МТВФ та забезпеченню більш стабільної роботи програмного забезпечення.

Кожен з сучасних підходів покращення методу роботи з метрикою середнього часу між відмовами є корисним та важливим, оскільки особливості їх застосування так чи інакше допомагають покращити показники безвідмовної роботи ПЗ. Правильне застосування різних підходів або їх комбінацій має позитивний вплив на загальний процес оцінювання роботи програмних систем під час безпосереднього їх тестування або експлуатації. Також наявні рішення для покращення мають велике значення для пошуку нової методики удосконалення роботи з метрикою з метою підвищення надійності та якості роботи ПЗ.

2.2 Інтеграція MTBF з іншими метриками тестування

Метрика середнього часу між відмовами слугує одним з основних показників надійності програмного забезпечення. Поряд з цим існує велика кількість інших показників якості програмних систем, кожен з яких відповідає за відображення певних їх характеристик. Для отримання точніших розрахунків часу безвідмовної роботи та загальної оцінки надійності роботи ПЗ можна використовувати різне поєднання метрик [19, 20]. Така інтеграція метрики MTBF та інших числових показників покликана покращити проведення процесу тестування програмного забезпечення в цілому та окремих його компонентів.

Першою для інтеграції з метрикою MTBF виступає метрика MTTR. MTTR (Mean Time To Repair) – це метрика, яка вимірює середній час, який потрібно для відновлення системи або програмного забезпечення після виникнення відмови. Таким чином, MTTR визначає тривалість часу, який потрібен для вирішення проблеми та приведення програми до робочого стану після того, коли в роботі ПЗ стався збій або певна помилкова ситуація.

Інтеграція метрики MTBF з показником MTTR дозволяє отримати більш повний огляд надійності системи або програмного забезпечення. Відповідне поєднання метрик може бути важливим при аналізі продуктивності та ефективності системи з точки зору користувача.

Сумарна метрика MTBF та MTTR визначає середній час між відмовами та час відновлення після відмови. Дана інтеграція дозволяє оцінити, скільки часу система працює та як швидко вона відновлюється після відмови. Знання MTBF та MTTR допомагає розробляти плани обслуговування та ремонту програмного забезпечення, що дозволяє покращити надійність, доступність та якість системи в цілому, надаючи можливість для швидкого відновлення після відмов та запобігаючи їх виникненню. Аналіз MTBF та MTTR може допомогти ідентифікувати основні фактори, які впливають на безперебійну роботу системи. Відповідна оцінка показників є важливою для розробки стратегій поліпшення надійності програмного забезпечення.

Загалом, інтеграція MTBF з MTTR розширює можливості для проведення процесів аналізу та управління надійністю системи та допомагає забезпечити більш ефективну їх роботу.

Наступна метрика, яка може бути використана для комбінації з MTBF, є MTTF. Метрика MTTF (Mean Time To Failure) – це метрика, яка представляє собою середній час до відмови програмної системи після переходу в робочий стан. Обидві метрики є схожими між собою, оскільки основним їх завданням є вимірювання часу роботи програмного забезпечення з врахування можливих відмов та збоїв. Основна відмінність між метриками полягає в тому, що MTTF не включає час, який витрачається на відновлення системи після відмови, проте MTBF включає цей час.

Інтегруючи відповідні метрики, можна отримати краще розуміння роботи системи і можливість оптимізувати управління надійністю. MTTF вказує, як довго система може працювати без відмови в умовах нормальної роботи. Поєднуючи MTTF з MTBF, можна визначити, чи перебуває система в робочому стані і чи варто розпочинати профілактичне обслуговування. Інтеграція MTTF з MTBF дозволяє планувати ремонти та заміни компонентів в оптимальний час, щоб максимізувати час безвідмовної роботи системи. Розуміння взаємозв'язку між MTTF та MTBF допомагає уникнути позапланових відмов та зменшити простій системи, забезпечуючи стабільну роботу. Коли MTTF перевищує MTBF, це свідчить про те, що система функціонує ефективно в умовах нормальної роботи і зберігається інформація про те, що може пройти ще певний час до наступної відмови. Відповідна інформація є корисною та важливою для оцінки загального стану функціонування програмного забезпечення.

Отже, інтеграція MTTF з MTBF дозволяє краще управляти надійністю системи та оптимізувати роботу компонентів. Враховуючи обидві метрики, можна більш об'єктивно визначити оптимальний час для проведення ремонтних робіт та збільшити безперебійність роботи системи.

Третьою для поєднання з показником MTBF є метрика SER. SER (Survival Event Rate) – це показник, що використовується для вимірювання інтенсивності

відмов в системі, та вона показує, як часто відмови відбуваються протягом певного періоду часу. Метрика SER вказує на імовірність того, що система здатна певний період часу працювати без відмов. Відповідний показник також використовується для оцінки надійності та стійкості системи в часі.

Інтеграція метрики MTBF з метрикою SER може допомогти отримати більш повний огляд надійності системи. Використання MTBF та SER дозволяє розробляти плани обслуговування та ремонту системи на основі інтенсивності відмов та часу між ними. Відповідна діяльність допомагає забезпечити безперебійну роботу системи та мінімізувати перерви у роботі. Інтеграція вказаних метрик дозволяє більш точно аналізувати зміни в надійності системи в часі. Виявлення збільшення інтенсивності відмов (SER) може попередити про можливі проблеми з надійністю у майбутньому. Комбінування показників дозволяє оцінити загальну надійність системи, враховуючи як інтенсивність збоїв ПЗ та певних його компонентів, так і час успішної роботи системи.

Інтеграція SER з MTBF допомагає отримати більш об'єктивний погляд на стійкість системи в часі і може бути корисною при прийнятті рішень щодо обслуговування, модернізації та удосконалення системи.

Ще одним числовим показником для комбінації з MTBF є метрика TTF. TTF (Time To Failure) – це метрика, що визначає час, який залишився до очікуваної відмови системи або компонента на певний момент часу. Вона вимірюється у відповідних одиницях часу, таких як години, дні чи інші визначені періоди.

Метрика TTF є важливою, оскільки дозволяє передбачити, скільки часу залишилося для безвідмовної роботи системи або компонента, і враховувати цю інформацію при прийнятті рішень щодо технічного обслуговування та ремонту. Інтеграція метрики TTF з метрикою MTBF дозволяє отримати більш об'єктивне розуміння надійності системи чи компонента. MTBF визначає середній час між відмовами після їх відновлення, тоді як TTF показує, скільки часу залишилося до наступної відмови. Якщо TTF значно менший за MTBF, це може свідчити про наближення відмови і вимагати негайної дії. Іншими словами, інтеграція цих

метрик допомагає забезпечити більше ефективне управління надійністю системи і вчасно реагувати на можливі відмови.

Загалом, інтеграція MTBF і TTF допомагає приймати рішення з питань відновлення роботи ПЗ та забезпечує більш повне уявлення про надійність системи в контексті часу і відмов.

Метрика покриття коду є ще одним показником для інтеграції з метрикою середнього часу між відмовами. Метрика покриття коду (Code Coverage) – це метрика, яка визначає обсяг виконання та перевірки програмного коду тестами під час тестування програмного забезпечення. Вона вимірює, скільки частин програмного коду було виконано в процесі виконання тестових сценаріїв. Дана метрика також може включати різні види покриття: покриття рядків коду (Line Coverage), що вимірює кількість рядків коду було виконано під час виконання тестових сценаріїв; покриття рішень (Decision Coverage), що вказує на те, які умовні конструкції були виконані під час виконання тестів; покриття гілок (Branch Coverage), що вимірює кількість гілок та розгалужень коду було пройдено під час виконання тестових сценаріїв; покриття функцій (Function Coverage), що показує, які функції були викликані і перевірені під час проведення тестування.

Комбінація метрики покриття коду з метрикою середнього часу між відмовами може бути корисною для оцінки якості тестування та надійності програмного забезпечення. Високий рівень покриття коду свідчить про те, що багато частин коду було перевірено тестами, що може зменшити ймовірність виявлення відмов в реальному середовищі.

Інтеграція показника покриття коду з MTBF дозволяє оцінити, наскільки добре тестовий набір взагалі охоплює різні частини програми, включаючи ті, які можуть бути критичними для надійності системи. Низький рівень покриття коду може свідчити про те, що певні частини програми не були протестовані і можуть містити потенційні помилки, які можуть вплинути на MTBF.

Отже, інтеграція метрики покриття коду з MTBF може допомогти вдосконалити процес тестування та забезпечити високу стабільність в роботі програмного забезпечення

Наступною для комбінації з МТВФ може бути використана метрика вад. Метрика вад (Defect Metric) – це показник, який визначає кількість виявлених дефектів або помилок в програмному забезпеченні під час розробки, тестування або експлуатації. Метрика вад може включати різні параметри: кількість виявлених дефектів в загальному вимірі та окремо на певну одиницю коду, важкість та пріоритетність дефектів, кількість виправлених та перевірених помилок, загальний розмір невиправлених дефектів. Відповідний показник допомагає визначити загальну якість програмного продукту, слугуючи індикатором того, наскільки багато помилок або недоліків було виявлено відносно загальної кількості розроблених чи випущених функцій.

Інтеграція метрики вад з метрикою МТВФ може бути корисною для оцінки зв'язку між кількістю виявлених дефектів та середнім часом між відмовами програмного забезпечення. Даний аналіз може вказати на те, як виявлені дефекти впливають на надійність програми. Якщо кількість вад зростає, це може свідчити про погіршення якості та може вплинути на зменшення МТВФ.

Комбінація вказаних метрик допомагає зрозуміти, як виправлення дефектів та підвищення якості програмного забезпечення можуть позитивно вплинути на надійність системи. Вимірювання обох метрик в динаміці та їх аналіз допомагають удосконалювати процес розробки, тестування та підтримки програмного забезпечення з метою покращення МТВФ та загальної якості продукту. Важливо зазначити, що інтеграція цих метрик дозволяє розробникам та тестувальникам вчасно виявляти та виправляти дефекти, що сприяє підвищенню та стійкості загальних необхідних характеристик для роботи ПЗ.

Наступною метрикою, яка також може бути поєднана з МТВФ, виступає метрика ефективності тестування. Метрика ефективності тестування (Testing Efficiency Metric) визначає, наскільки ефективно використовуються ресурси та зусилля під час процесу тестування програмного забезпечення. Вона надає інформацію про те, як проходить тестування враховуючи встановлені обмеження і ресурси, а також чи вдається досягти поставлених цілей та завдань. Метрика ефективності тестування може включати в себе різні параметри, такі як кількість

успішно виконаних тест-кейсів, тривалість тестування, кількість виявлених дефектів, вартість тестування, кількість фальшивих спрацювань у тестових звітах і багато інших показників.

Інтеграція даної метрики з метрикою середнього часу між відмовами важлива для оцінки та поліпшення надійності програмного забезпечення. Метрика MTBF визначає час безперебійної роботи програмного забезпечення, тоді як метрика ефективності тестування допомагає зрозуміти, наскільки ефективно виявляються дефекти під час безпосереднього процесу тестування.

Поєднання цих метрик може допомогти виявити, які аспекти тестування та виправлення дефектів впливають на збільшення MTBF. Наприклад, висока ефективність тестування, що призводить до раннього виявлення та виправлення дефектів, може сприяти підвищенню MTBF шляхом зменшення кількості відмов. Таким чином, інтеграція цих метрик допомагає забезпечити більш надійну та якісну роботу ПЗ.

Для комбінації з показником MTBF може також бути використана метрика вартості відмов. Метрика вартості відмов (Failure Cost Metric) – це числовий показник, який використовується для вимірювання фінансових наслідків відмов програмного забезпечення. Вона оцінює витрати та втрати, пов'язані з відмовами, такі як втрати прибутку, витрати на відновлення програми, витрати на підтримку користувачів і витрати на компенсацію клієнтам. Дана метрика допомагає приймати рішення щодо вдосконалення надійності програмного забезпечення, так як вона дозволяє оцінити вплив відмов на бізнес-процеси та прибуток з розробки програмного забезпечення.

Інтеграція метрики вартості відмов (FCM) з метрикою середнього часу між відмовами (MTBF) дозволяє комплексно оцінювати як технічну надійність, так і економічний аспект програмного забезпечення. Така діяльність означає, що при аналізі надійності програми враховуються не лише технічні показники, але і витрати, пов'язані з можливими відмовами. Інтеграція цих метрик допомагає зрозуміти співвідношення впливу збоїв на фінансову виправданість покращень в надійності програмного забезпечення.

Отже, MTBF та FCM разом надають комплексний погляд на надійність програми з технічної та економічної точок зору, допомагаючи приймати обґрунтовані рішення щодо покращення властивостей ПЗ.

Останньою для комбінації з MTBF може виступати метрика швидкості виправлення вад. Метрика швидкості виправлення вад (Defect Resolution Speed Metric) вимірює час, необхідний для виявлення, документування, виправлення і верифікації відмов або дефектів у програмному забезпеченні. Умовні одиниці часу можуть бути представлені у вигляді хвилин, годин, діб тощо. Дана метрика оцінює ефективність процесу управління вадами та важлива для вимірювання тимчасової продуктивності команди розробників.

Інтеграція метрики швидкості виправлення вад з метрикою MTBF полягає в тому, що вона допомагає визначити, наскільки швидко команда може реагувати на виявлені проблеми в програмному забезпеченні та вирішувати їх.

Спільне використання цих двох метрик дозволяє оцінювати ефективність процесу виявлення та виправлення вад в контексті загальної надійності програмного забезпечення. Якщо команда швидко реагує на виявлені дефекти і виправляє їх, то це може сприяти підвищенню MTBF, оскільки менше часу витрачається на вирішення проблем. На загальному рівні інтегровані метрики допомагають забезпечити надійність та якість програмного продукту в умовах швидкої реакції на виявлені помилкові ситуації.

Інтеграція метрики середнього часу між відмовами з усіма зазначеними показниками слугує важливим аспектом для визначення та підвищення рівня надійності програмного забезпечення. Метрика MTBF може успішно компонуватись зі всіма іншими метриками на етапі тестування та під час подальшого супроводу програмних систем, що допомагає забезпечити високоякісне їх обслуговування та швидке виправлення дефектів, в разі їх появи. Комбінація MTBF та різнотипних показників підтримує успішну та безперебійну роботу системи та допомагає мінімізувати ризики виникнення збоїв у функціонування програми.

2.3 Автоматизація процесу розрахунку МТBF

Розрахунок показника метрики середнього часу між відмовами може відбуватись різними методами та інструментами в залежності від типу програмного забезпечення та особливостей відмов, які стаються під час його роботи. Важливим аспектом для обчислення МТBF може слугувати автоматизація даного процесу. Існують декілька способів автоматизації розрахунку показника безвідмовної роботи ПЗ, що здатні значно спростити оцінювання його надійності.

Для виконання автоматичного розрахунку МТBF можливе використання різних алгоритмів та моделей. Їх застосування виступає одним з перших способів для проведення полегшених обчислень відповідного показника. Одним з алгоритмів, який здатний оцінювати надійність програмної системи, виступає експоненціальний розподіл [21]. Використання даного розподілу дозволяє розраховувати МТBF на основі статистики відмов. Алгоритм може бути налаштований для збору і аналізу інформації про відмови в системі. Для основного розрахунку він використовує формулу експоненціального розподілу, яка враховує кількість відмов і загальний час спостереження для обчислення МТBF. Відповідний процес може бути повністю автоматизованим за допомогою спеціалізованих інструментів і програм (ReliaSoft, MATLAB, RELEX та ін.).

Іншим алгоритмом для автоматичного розрахунку показника часу безвідмовної роботи є конкурентний алгоритм. Відповідний алгоритм використовується для оцінки надійності системи, яка має декілька компонентів або підсистем. Він враховує конкурентний вплив елементів один на одного та дозволяє враховувати взаємодію між ними при розрахунку МТBF.

Метод Монте-Карло також використовується для автоматизації процесу розрахунку показника метрики [22]. Даний алгоритм базується на стохастичних симуляціях та випадкових величинах для оцінки МТBF. Він може бути корисним для складних та великих програмних систем, де використання інших методів може бути неможливим або ускладненим.

Поряд з використанням алгоритмів застосовуються різні моделі для обчислення показника середнього часу між відмовами ПЗ в автоматичному порядку. Зокрема, однією з таких є модель часових рядів. Моделювання часових рядів дозволяє виявити тенденції та патерни у даних про відмови. Програми для аналізу часових рядів можуть визначати параметри моделі та робити прогнози на основі нових даних.

Марковські моделі та модель станів та переходів також є загальними абстрактними інструментами для обчислення МТВФ. Для складних систем, які можуть перебувати в різних станах та ситуаціях, моделі допомагають визначити ймовірності відмов для кожного стану та переходів між ними. Хоча специфіка та особливості застосування даних моделей може відрізнятися, проте обидва типи моделей поряд з наявними алгоритмами автоматизації здатні оцінювати роботу програмного забезпечення або окремих його компонентів для виявлення потенційних помилкових ситуацій.

В цілому, використання різних алгоритмів та моделей автоматизації значно полегшує оцінку МТВФ і покращує надійність системи шляхом розуміння її найбільш критичних точок та виявлення можливостей для оптимізації.

Автоматизація розрахунку показника середнього часу між відмовами може бути здійснена за допомогою іншого загального способу, а саме з використанням системи керування вадами програмного забезпечення. Відповідна система допомагає відстежувати та керувати виявленими вадами або незначними помилковими ситуаціями під час тестування та протягом експлуатації ПЗ.

В контексті автоматизації розрахунку МТВФ система керування вадами дозволяє збирати та аналізувати інформацію про відмови та дефекти, які виникають в програмному забезпеченні. Вона фіксує дані про час і причини відмов, а також про заходи, вжиті для їх усунення. Відповідна інформація є основою для розрахунку показника МТВФ.

Для отримання вірних результатів показника метрики середнього часу між відмовами важливо безпомилково інтегрувати відповідний процес обчислення до певної системи керування вадами. Зокрема, одним з варіантів є звичайне

додавання поля MTBF до записів про вади чи дефекти в системі керування вадами. Також можна інтегрувати інструменти або скрипти для автоматичного розрахунку MTBF на основі даних, зібраних системою керування. Іншим варіантом поєднання метрики та системи керування є встановлення зв'язків між вадами та оцінками MTBF для визначення, які вади найбільше впливають на надійність системи, що дозволить приділити пріоритети виправленню та покращенню вад, які негативно впливають на MTBF.

Система керування вадами дозволяє в автоматичному режимі допомагати ідентифікувати тенденції в змінах MTBF з часом та виокремлювати основні причини відмов. До загальновідомих систем керування дефектами відносяться JIRA, Bugzilla, Redmine, Trello, Mantis. Завдяки вказаним системам можливо автоматизовано генерувати звіти та аналізи про надійність програми, що спрощує розрахунок MTBF та дозволяє розробникам та тестувальникам швидше реагувати на виявлені проблеми.

В загальному, використання системи керування вадами сприяє ефективнішому та автоматизованому процесу розрахунку MTBF та покращенню надійності програмного забезпечення.

Ще одним загальним підходом для автоматичного розрахунку метрики MTBF виступає використання інструментів автоматизації тестування. Даний спосіб також спрощує обчислення показника та використовується для оцінки надійності ПЗ.

Серед інструментів автоматизації тестування програмних систем існують такі: Selenium, Appium, JUnit, TestNG, Jenkins, Postman тощо. Кожен зі вказаних інструментів призначений для автоматизованого виконання, моніторингу і аналізу тестових сценаріїв та відслідковування результатів виконання тестів.

Інструменти автоматизації тестування можуть створювати автоматичні журнали або реєстри відмов під час тестування. Відповідні дані можуть включати час, причини та походження відмов. Збір таких даних в автоматичному режимі допомагає у точній реєстрації відмов для подальшого аналізу MTBF.

Після збору даних інструменти можуть бути налаштовані для автоматичного розрахунку MTBF на основі відомої інформації. Вони можуть використовувати стандартні формули та алгоритми для розрахунку середнього часу між відмовами, спрощуючи відповідний процес.

Розрахувавши показник часу безвідмовної роботи, механізми автоматизації можуть бути налаштовані на постійний моніторинг програмного забезпечення. Інструменти можуть відстежувати тривалість кожного тесту та інтервали між відмовами системи. Такий процес допомагає точно визначити часові параметри, які використовуються для розрахунку MTBF. Також для покращеного спостереження за процесом тестування ПЗ механізми автоматизації можуть створювати звіти та генерувати сповіщення про значення показника середнього часу безвідмовної роботи. Така діяльність спрощує загальний моніторинг та сприяє оперативному реагуванню на покращення або погіршення надійності програмного забезпечення.

Загалом, інструменти автоматизації допомагають значно прискорити процес тестування програмного забезпечення, а загальне їх використання допомагає спростити та поліпшити процес оцінки та покращення MTBF, що є важливим для забезпечення високої надійності програмного забезпечення.

Для автоматизації процесу розрахунку показника метрики середнього часу між відмовами існує ще один зі способів. Даний підхід заснований на відстеженні показника в реальному часі. Вказаний спосіб включає в себе використання спеціальних програмних інструментів і систем для постійного нагляду за роботою ПЗ та автоматичного обліку часу між відмовами.

Для роботи даного способу першочергово вказуються критерії відмов. Спочатку визначаються та програмуються критерії відмови, які вказують на те, коли система вважається надійною або, навпаки, коли відмова сталася. Відповідні критерії можуть бути пов'язані з певними подіями, помилками, зупинками роботи, а також з іншими параметрами, що свідчать про несправність. Спеціальні засоби або програмні модулі відстежують роботу програмного забезпечення в реальному часі. Вони реєструють відмови, помилки, а також час між відмовами. Зібрані дані

про відмови та час між ними агрегуються для подальшого аналізу. Інструменти моніторингу можуть автоматично визначати MTBF на основі цих даних.

Системи спостереження, такі як Prometheus, Grafana, Nagios, Zabbix та інші, надають можливість встановлення моніторингу для програмного забезпечення і операційної системи. Вони дозволяють визначати параметри, які можуть свідчити про відмови або неполадки в роботі системи, та автоматично реєструвати ці події.

Логування подій у програмному забезпеченні також може бути представлено як один зі способів відстеження даних про роботу ПЗ. Воно дозволяє фіксувати і аналізувати важливі події, помилки та відмови. Лог-файли можуть бути аналізовані в реальному часі для виявлення аномалій.

Також для відслідковування за роботою програмних систем в реальному часі може бути використаний більш традиційний спосіб. Моніторинг ресурсів, такий як використання центрального процесора, обсяг оперативної пам'яті, обсяг дискового простору та інші параметри, дозволяє виявляти можливі проблеми, пов'язані з ресурсами, які можуть вплинути на надійність програмної системи.

Використання різних систем, інструментів і методів моніторингу та відстеження MTBF в реальному часі дозволяє автоматизувати процес розрахунку цього показника, забезпечити постійний контроль за надійністю програмного забезпечення та дозволяють оперативно реагувати на будь-які проблеми, що можуть вплинути на час безвідмовної роботи.

Автоматизація процесу розрахунку показника середнього часу між відмовами є необхідним кроком для оптимізації та полегшення виконання даного етапу під час тестування програмного забезпечення. Використання автоматизації в цьому контексті охоплює різні аспекти, включаючи застосування різних алгоритмів, моделей, систем, інструментів та методів, які сприяють швидшому та більш ефективному обчисленню MTBF. Завдяки автоматизації, процес розрахунку MTBF стає менш витратним з точки зору ресурсів та часу, що дозволяє команді розробників та тестувальників більш ефективно використовувати свій потенціал для отримання точних результатів оцінки надійності програмних продуктів.

2.4 Порівняльний аналіз сучасних підходів до удосконалення методу МТВФ

Основна мета кожного сучасного підходу до удосконалення методики роботи з метрикою середнього часу між відмовами в загальному полягає у забезпеченні високого рівня розробки та тестування програмних систем. Різні методи та підходи мають свої особливості та специфіку виконання та використання. Проте від результатів їх застосування залежить надійність та успішність роботи програмних продуктів в цілому та окремих його компонентів. Кожен з описаних підходів має власні позитивні сторони та обмеження, що так чи інакше впливають на доцільність їх використання в певних ситуаціях. Для ознайомлення з перевагами та негативними сторонами описаних підходів необхідним є проведення порівняльного аналізу.

Використання статистичних методів для аналізу відмов дозволяє отримати об'єктивну інформацію про тенденції та закономірності в роботі системи. Вони дозволяють аналізувати великий обсяг історичних даних та виявляти закономірності. Також вони використовують об'єктивний підхід для виявлення факторів, що впливають на надійність. Однак дані методи можуть не враховувати незрозумілі або неочікувані залежності. Ще одним обмеженням є те, що вони вимагають великої кількості історичних даних про роботу ПЗ для статистично значущих результатів.

Машинне навчання може ефективно використовувати складні моделі для прогнозування МТВФ на основі великої кількості даних. Такий процес є особливо корисним у випадках, коли існує велика кількість параметрів та факторів, що впливають на стабільність роботи системи. Також МН має здатність враховувати складні залежності та швидко адаптуватися до змін у функціонуванні ПЗ. Проте воно вимагає великої кількості даних для тренування, а також може бути складними для налаштування та інтерпретації.

Моделювання архітектури системи може допомогти виявити можливі точки відмов та слабкі місця. Також воно дозволяє виправити проблеми та удосконалити архітектурні рішення ще до реалізації. Моделі можуть бути використані для

аналізу різних сценаріїв та ідентифікації потенційних проблем. Проте недоліком даного процесу в контексті розрахунку MTBF є вимоги значних ресурсів для моделювання та аналізу. В такому випадку точність моделей буде залежати від правильності архітектурного опису, що в підсумку може вплинути на загальний показник надійності ПЗ.

Оцінка надійності на основі емпіричних даних є важливою складовою удосконалення методу MTBF. Збір та аналіз історичних даних дозволяють враховувати специфіку конкретної системи та її експлуатаційних умов. Оцінювання стабільності на основі реальних даних є дійсним та об'єктивним процесом. Однак обмеженням є те, що для визначення рівня надійності та виявлення змін даним способом може бути недостатньо даних, що додає вимогу систематичного збору та обробки інформації.

Адаптивні підходи дозволяють системі реагувати на зміни у роботі та умовах експлуатації, що дозволяє підвищити надійність у реальному часі. Ці методи можуть включати автоматичне налаштування параметрів, перевірку відмов в негайному режимі та гнучке керування ресурсами. Проте вони вимагають складного та чіткого управління даними ресурсами для реалізації адаптивних стратегій та можуть бути складними для загального налаштування.

Комбінація різних метрик з показником MTBF також має свої позитивні сторони та обмеження. Зокрема, інтеграція MTBF з MTTR дозволяє отримувати повноцінне бачення процесу відновлення системи після відмови. Даний аспект допомагає краще розуміти час відновлення та його вплив на надійність. Проте важливо враховувати, що MTTR та MTBF можуть бути некерованими факторами. Інтеграція може бути складною при відсутності детальних даних про час відновлення програмного забезпечення.

Інтеграція MTBF та MTTF дозволяє краще розуміти час між відмовами та час безвідмовної роботи системи. Такий момент є корисним для оцінки загальної продуктивності системи. Однак необхідно розуміти, що MTBF та MTTF можуть мати різні розподіли та фактори впливу на них. Поєднання даних метрик потребує обґрунтування спільної використаної методології.

Спільне використання метрики MTBF з SER допомагає зосередитися на аналізі помилок програмного забезпечення, які можуть призвести до відмов, що дозволяє покращити якість програмного забезпечення. Але метрику SER може бути складною для вимірювання. Комбінування показників вимагає збору та аналізу додаткових даних про відмови.

Інтеграція MTBF з TTF дозволяє визначити час до очікуваної відмови та планувати обслуговування, що є позитивним моментом. Проте ускладненням для використання даної комбінації може виступати факт важкого передбачення точного часу відмови. Інтеграція вимагає ретельного аналізу даних та створення детальних моделей компонентів програмного забезпечення.

Інтеграція з метрикою покриття коду дозволяє виявляти кодові ділянки, які можуть впливати на надійність системи. Позитивною стороною цього є можливість зосередити зусилля безпосередньо на виправленні проблемних місць. Однак покриття коду не завжди корелює з реальною надійністю системи, що відповідно потребує систематичного аналізу коду.

Інтеграція показника часу безвідмовної роботи з метрикою покриття вад дозволяє ідентифікувати та слідкувати за вадами у програмному забезпеченні. Проте недоліком є те, що комбінація метрик не завжди може вказати на точний час відмов та тривалість їх виправлення. Спільне використання вимагає систематичного відстеження та виправлення вад.

Інтеграція MTBF з метрикою ефективності тестування допомагає визначити, наскільки добре вдалося виявити вади під час тестування. Проте відповідне застосування метрик може не відображати реальну надійність системи. Комбінація даних показників вимагає аналізу результатів тестування та їх використання в контексті MTBF.

Інтеграція з метрикою вартості відмов дозволяє оцінити фінансові наслідки відмов та планувати обслуговування. Важливо враховувати, що реальну вартість відмов може бути важко оцінити точно. Інтеграція MTBF з метрикою вартості відмов вимагає ретельного аналізу фінансових даних.

Інтеграція показника МТBF з метрикою швидкості виправлення вад допомагає визначити, наскільки швидко та вірно здійснюється ремонт ПЗ та усунення виявлених проблем. Проте недоліком комбінації є те, що метрика швидкості виправлення збоїв не враховує час між відмовами. Спільне використання метрик потребує додаткового аналізу даних та часу виправлення.

Автоматизація процесу розрахунку як ще один підхід до удосконалення методу МТBF також має свої особливості. Різні алгоритми та моделі по-різному виконують свої завдання та мають позитивні і негативні сторони.

Експоненціальний розподіл відзначається своєю математичною простотою та можливістю аналітичних обчислень, зокрема й показника МТBF. Такий аспект дозволяє використовувати його для моделювання часу між відмовами, особливо, коли відмови є незалежними подіями. Проте, він не завжди враховує складні залежності та фактори, що впливають на відмови в реальних системах.

Конкурентні алгоритми, незважаючи на більшу складність, можуть враховувати багато даних про відмови та параметри системи для точнішого моделювання МТBF. Вони підходять для систем зі складними залежностями та факторами ризику, але вимагають великих обчислювальних ресурсів та високої кваліфікації для здійснення різних операцій за допомогою їх використання.

Метод Монте-Карло використовує стохастичне моделювання на основі випадкових величин та параметрів системи. Він підходить для врахування випадкових факторів та складних залежностей, що впливають на МТBF, але може витратити велику кількість часу та значні ресурси, що може позначитись на результатах розрахунків.

Модель часових рядів дозволяє враховувати часові залежності відмов та аналізувати зміни в часі. Вона корисна для виявлення трендів та сезонності в даних про відмови, враховуючи показник МТBF. Однак дана модель обмежена вимогами до наявності великої кількості історичних даних та може бути обмеженою в здатності точно моделювати складні взаємозв'язки між збоями.

Марковська модель дозволяє відтворювати стани системи та переходи між ними. Вона здатна враховувати залежності між відмовами та переходами ПЗ з

одного стану в інший, що може бути корисним для систем зі складною структурою та взаємодією компонентів. Проте відповідна модель вимагає складних математичних розрахунків і детального аналізу даних про збої та успішну роботу програмних систем.

Модель станів та переходів зосереджена на точному моделюванні станів системи та їх відповідних переходах. Вона часто використовується для оцінки надійності складних та великих систем, де показник МТВФ є критично важливим. Однак вона також вимагає точних даних про стани функціонування програмного забезпечення та ймовірності переходів між ними.

Кожен з сучасних підходів до удосконалення методики роботи з метрикою МТВФ є хорошим для використання його в процесі розрахунку часу безвідмовної роботи. Проте застосування того чи іншого способу залежить від конкретної ситуації, що може включати специфіку поставлених завдань, характеристики системи, різні вимоги проекту, наявність необхідних даних та ресурсів. В багатьох випадках комбінування різних підходів може бути найефективнішим рішенням для поліпшення методу МТВФ.

2.5 Загальні пропозиції щодо удосконалення методу МТВФ на основі проведеного аналізу

Різні позитивні сторони застосування сучасних підходів для розрахунку МТВФ однозначно мають важливий вплив на обчислення даного показника, підвищення стабільності та надійності роботи програмного забезпечення та проведення етапу його тестування в цілому. Наявні способи значно полегшують вирішення поставлених задач та отримання необхідних результатів. В той же час існуючі обмеження різних підходів обчислення показника безвідмовної роботи ПЗ дають можливість для пошуку нових напрямків або вдосконалення вже наявних з метою спрощення процедури обчислення та забезпечення високої якості різнотипних програмних систем.

Для подальшого розвитку нових моделей та алгоритмів для розрахунку MTBF необхідно розглянути кілька важливих аспектів. По-перше, важливо спрямувати зусилля на створення більш точних та адаптивних моделей, здатних враховувати широкий спектр умов експлуатації системи. Для цього може бути корисним використання нестандартних розподілів та розширення підходів, зокрема робота з функціями надійності та інтенсивності відмов.

По-друге, важливо активно впроваджувати аналіз статистичних даних та використання методів машинного навчання для виявлення закономірностей у відмовах та впливу різних факторів на надійність системи. Застосування глибинного навчання та нейронних мереж може покращити точність розрахунків MTBF та забезпечити аналіз в реальному часі.

Важливим аспектом має бути вдосконалення інструментів для моделювання програмного забезпечення на основі архітектури з підтримкою різних типів тестування та віртуальних середовищ. Така діяльність допоможе при аналізі взаємодії компонентів та ідентифікації потенційних проблем у дизайні системи.

Крім того, важливо набагато активніше використовувати емпіричні дані для налаштування та валідації моделей надійності та оцінки їх впливу на MTBF. Робота з реальними даними допомагає вдосконалити моделі та робити їх більш адаптивними до конкретних умов.

Інтеграція метрики MTBF з іншими метриками тестування повинна бути ще одним ключовим аспектом для покращення надійності та доступності програмного забезпечення. Окрім зазначених раніше метрик, існують інші метрики, які можуть бути інтегровані з MTBF для глибшого розуміння та покращення надійності системи.

Метрика навантаження надає інформацію про те, як система працює під навантаженням, що є критичним для оцінки MTBF. Вона дозволяє встановити, як зміни в навантаженні впливають на надійність системи та якість її роботи.

Метрика покриття вимог допомагає визначити, наскільки тестування охопило всі вимоги до системи. Даний процес є важливим, оскільки неперевірені вимоги можуть призвести до відмов. Інтеграція з MTBF може допомогти

визначити, чи є залежність між покриттям вимог і показниками надійності програмної системи.

Метрика ризиків і припустимих відмов дає можливість ідентифікувати потенційні ризики та визначити, наскільки припустимі рівні відмов можуть бути порушені. Відповідний аспект дозволяє зосередитися на важливих факторах тестування та розробки для підвищення надійності системи.

Метрика витрат на тестування та ремонт допомагає керувати витратами та ресурсами, спрямованими на підвищення надійності. Інтеграція з МТВФ дозволить визначити, як ефективно витрачаються кошти на підвищення стабільності системи та як це впливає на загальну вартість і результати.

Для полегшеного розрахунку МТВФ важливо також розглядати напрямок автоматизації та додаткові підходи для його реалізації. Одним з перших має бути розвиток спеціалізованих платформ для надійності ПЗ, які включають в себе автоматичний збір та аналіз даних від віддалених систем. Відповідна діяльність допоможе в реальному часі відстежувати показники надійності без значних зусиль та затрат ресурсів.

Другий напрямок – розробка інтегрованих інструментів для оцінки інших метрик, які впливають на надійність, разом із МТВФ. Дана діяльність може включати в себе автоматизовані засоби для вимірювання різних числових показників та ключових параметрів, які можуть бути інтегровані для більш комплексного аналізу.

Необхідно також розглянути можливості використання блокчейн-технологій для забезпечення документації та перевірки результатів розрахунків МТВФ. Дане нововведення може забезпечити надійність та недекларовану зміну даних, що є важливим для об'єктивності результуючих метрик.

Для подальшого розвитку методів та інструментів моніторингу та відстеження МТВФ в реальному часі можна розглянути використання сучасних технологій Internet of Things (IoT). Інтеграція IoT-пристроїв у системи, де вимагається висока надійність, дозволить збирати дані в реальному часі з різних компонентів системи та передавати їх на аналіз до центральної системи.

Відповідно такі процеси дозволять отримувати найсвіжішу інформацію про стан системи і виявляти відмови навіть на ранніх стадіях.

Для реалізації такої системи необхідно розробляти високоефективні датчики та IoT-пристрої, які можуть надавати дані щодо стану обладнання і систем в режимі реального часу. Також важливо розробити потужний аналітичний програмний засіб, який може обробляти великі обсяги даних та аналізувати їх для передбачення можливих відмов.

Крім цього, важливо розглянути розробку та впровадження інтелектуальних програмних інструментів, які здатні передбачати можливі відмови на основі аналізу динаміки параметрів системи. Такі засоби повинні включати використання різноманітних методів для виявлення аномалій та прогнозування показника МТBF.

Кожен з можливих напрямків для розробки удосконаленого методу розрахунку МТBF є дуже перспективним та заслуговує на велику увагу. Успішна їх реалізація допоможе полегшити виконання процесів на етапі тестування та створити ідеальні умови для забезпечення високої якості програмних систем. З урахуванням стрімкого розвитку технологій збільшується складність програм і залежність від них. Саме тому покращення МТBF стає ключовим завданням, яке потребує ефективних сучасних високотехнологічних рішень.

3 РОЗРОБКА УДОСКОНАЛЕНОЇ МЕТОДИКИ РОБОТИ З МЕТРИКОЮ МТВФ ДЛЯ ЕФЕКТИВНОГО ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ ТЕСТУВАННЯ ПЗ

3.1 Визначення вимог до удосконаленої методики роботи з МТВФ

Різноманітні сучасні підходи для удосконаленої роботи з метрикою середнього часу між відмовами показують, що їх застосування здатне покращити як сам процес оцінювання якості та надійності програмного забезпечення, так і загальне виконання етапу тестування системи. Для здійснення пошуку нових методів роботи з метрикою МТВФ важливо використовувати вже набутий досвід та аналіз наявних способів її покращення, що дозволить охопити велику частину різних аспектів, які є необхідними для функціонування показника часу безвідмовної роботи. Також удосконалений метод роботи з показником МТВФ повинен задовольняти певним вимогам, які представляють загальні особливості його використання під час виконання тестування програмного забезпечення.

Визначення вимог до удосконаленої методики роботи з метрикою середнього часу між відмовами – це процес створення чітких та конкретних критеріїв і очікувань для того, як повинна функціонувати та які результати повинна надавати методика аналізу та розрахунку надійності систем, апаратного чи програмного забезпечення. До основних вимог до удосконаленої методики роботи з МТВФ можна включити такі аспекти:

- точність та адекватність оцінювання;
- простота реалізації та використання;
- адаптивність до особливостей та специфіки функціонування програмного забезпечення та тестових даних;
- можлива інтегрованість з іншими метриками тестування;
- автоматичний розрахунок та моніторинг МТВФ;
- можливість налаштування та оптимізації параметрів;
- чіткість та зрозумілість результатів;

– підтримка прийняття рішень.

Перша вимога, що стосується точності та адекватності оцінювання, є однією з найбільш значущих вимог до удосконаленої методики роботи з МТВФ. Дана вимога передбачає, що методика повинна забезпечувати високу точність та відображати реальний стан надійності програмного забезпечення. Точність розрахунків МТВФ є важливою, оскільки від неї залежать прийняття рішень щодо планування обслуговування та запобігання відмовам. Результати розрахунків повинні бути якомога ближчими до реальних значень, щоб забезпечити ефективні стратегії обслуговування.

Адекватність моделей, використаних для розрахунків МТВФ, є ще одним ключовим фактором. Моделі повинні враховувати складну структуру системи та взаємозв'язок між компонентами. Якщо моделі не відображають відповідні особливості, то результати можуть бути неточними.

Адекватність даних також грає важливу роль. Для оцінки МТВФ потрібні належні дані про відмови та вимоги до збору цих даних. Неповні або неточні дані можуть призвести до неправильних розрахунків. Коректність аналізу даних та точність при інтерпретації результатів розрахунку метрики є обов'язковими. Отримані результати повинні враховувати всі важливі фактори, включаючи зміни в умовах експлуатації та різні види відмов.

Друга вимога – простота реалізації та використання – є ще однією важливою вимогою до покращення методу роботи з показником часу безвідмовної роботи. Дана вимога передбачає, що методика повинна бути доступною і зрозумілою для користувачів, а також надзвичайно простою в реалізації та використанні.

Простота реалізації означає, що сам процес збору та аналізу даних для розрахунку МТВФ повинен бути легким у виконанні. Даний процес може включати в себе використання різних традиційних або спеціальних інструментів збору даних, а також наявність рекомендацій щодо процедур розрахунків.

Простота використання означає, що методика повинна бути інтуїтивно зрозумілою для користувачі. Відповідний аспект дозволить тестувальникам,

інженерам та операторам ефективно використовувати методику без значних зусиль і спеціалізованої підготовки.

Адаптивність до специфіки програмного забезпечення та тестових даних є також необхідною вимогою до удосконаленої методики роботи з показником середнього часу між відмовами. Оскільки програмне забезпечення може бути дуже різноманітним і розроблятися для різних цілей, методика повинна бути адаптованою до конкретних вимог та особливостей.

Адаптивність до специфіки програмного забезпечення передбачає, що методика повинна бути в змозі працювати з різними типами програм, незалежно від їх складності, архітектури, мов програмування тощо. Вона також повинна бути здатною аналізувати програми різних класів та враховувати їхні особливості під час розрахунку МТВФ.

Адаптивність до тестових даних означає, що методика має працювати з різними наборами тестових даних, включаючи дані з реального використання програмного забезпечення. Важливим аспектом також повинна бути робота з різним обсягом даних, незалежно від їх складності та формату представлення.

Можлива інтегрованість з іншими метриками тестування виступає ще одним аспектом, що здатен допомогти при розрахунку МТВФ за удосконаленою методикою. Включення інших метрик тестування до аналізу МТВФ дозволяє отримувати більш повну та об'єктивну інформацію про надійність та якість програмного забезпечення.

Залучення інших метрик тестування також допомагає визначити, наскільки відповідний МТВФ є адекватним показником якості програмного забезпечення. Узгодженість між різними метриками тестування здатне забезпечити цілісний та надійний аналіз стану програмного забезпечення. Інтеграція метрик дозволяє здійснювати оцінювання та планування покращень в роботі з МТВФ та в загальному процесі розробки та тестування програмного забезпечення.

Автоматичний розрахунок та моніторинг МТВФ також належить до вимог до удосконаленої методики роботи з показником часу безвідмовної роботи. Основна мета цієї вимоги – забезпечити зручний та ефективний спосіб визначення

та відстеження показників надійності програмного забезпечення без значних зусиль та втрат часу.

Автоматичний розрахунок МТBF дозволяє отримувати результати безпосередньо з даних, які вже збираються під час роботи програмного забезпечення або тестування. Відповідний процес полегшує здійснення оцінки надійності та дозволяє отримувати оновлену інформацію в режимі реального часу. Моніторинг МТBF дозволяє відстежувати зміни в цьому показнику та вчасно реагувати на можливі проблеми.

За даною вимогою необхідно мати вбудовані інструменти, які можуть автоматично здійснювати розрахунок МТBF на основі існуючих даних та надавати результати в доступному та зрозумілому форматі. Такі інструменти дозволяють значно економити час і зусилля на оцінці надійності, що особливо важливо в умовах прогресуючого середовища розробки та тестування ПЗ.

Наступна вимога – можливість налаштування та оптимізації параметрів – представляє ще одну потребу для виконання функції розрахунку МТBF відповідно до удосконалення. Дана вимога передбачає можливість адаптувати покращену методика до конкретних потреб та особливостей проекту, програмного забезпечення або тестових даних.

Налаштування параметрів дозволяє користувачам враховувати специфіку їхнього програмного продукту згідно аналізу основних аспектів його роботи та потенційно вразливих місць, а також враховувати певні умови експлуатації. Дана діяльність особливо корисна в тих випадках, коли стандартні підходи можуть бути недостатньо адекватними або неефективними.

Оптимізація параметрів важлива для покращення загальної точності розрахунків МТBF. Відповідний фактор дозволить уникнути хибних розрахунків, які можуть виникати через некоректний вибір параметрів або методології. Правильний вибір та налаштування параметрів може покращити відповідність результатів реальним умовам експлуатації, що є критично важливим для оцінки надійності програмного забезпечення.

Чіткість та зрозумілість результатів також належить до вимог, які стосуються удосконаленої методики роботи метрикою середнього часу між відмовами. Відповідна вимога передбачає, що користувачі (розробники, тестувальники) повинні легко розуміти та інтерпретувати результати отриманих розрахунків. Для досягнення цієї мети, методика повинна бути чітко описана, а її методологія повинна бути зрозумілою для користувачів. Вхідні дані мають бути визначені в строгому порядку, а тестувальники повинні володіти інформацією про те, які саме дані необхідно зібрати для розрахунків МТBF.

Зрозумілість параметрів і результатів також важлива. Кожен параметр та результат розрахунків повинні бути чітко позначені та пояснені, щоб користувачі могли правильно інтерпретувати отриману інформацію.

Нарешті, результати розрахунків мають бути легко інтерпретованими без необґрунтованих труднощів. Такий аспект може бути досягнуто за допомогою графічних зображень, діаграм або текстових описів, які допомагають користувачам легко визначити тенденції та зробити інформовані висновки.

Підтримка прийняття рішень є останньою вимогою до удосконаленої методики роботи з МТBF. Вона передбачає, що методика повинна надавати користувачам інформацію та інструменти, необхідні для прийняття обґрунтованих рішень з покращення надійності системи.

Для досягнення цієї вимоги методика повинна забезпечувати можливість аналізу різних альтернатив та їх впливу на значення МТBF. Даний процес включає в себе можливість моделювання різних сценаріїв, проведення детального аналізу та ідентифікації ключових параметрів, які впливають на надійність системи. Також важливо, щоб на основі результатів роботи методики було можливим формулювання загальних рекомендацій щодо можливих заходів по покращенню показника МТBF.

Для підтримки прийняття рішень методика повинна надавати зручні засоби візуалізації результатів та можливість порівнювати різні сценарії. Важливо, щоб користувачі мали можливість швидко оцінювати наслідки впровадження покращень та вибирати найбільш обґрунтовані рішення.

Сформований перелік вимог чітко окреслює необхідний набір ключових сторін удосконаленої методики роботи з метрикою середнього часу між відмовами. Встановлені пункти повинні регулювати сам процес створення покращеного методу та в подальшому здійснювати оцінку його роботи під час безпосереднього використання. Створена методика за визначеними вимогами буде мати хороші якості для виконання розрахунку показника MTBF, визначення та підвищення рівня надійності роботи програмного забезпечення та окремих його компонентів, а також в цілому позитивно впливти на проведення тестування програмних систем та послідовне отримання та оцінювання його результатів.

3.2 Опис критеріїв оцінки ефективності удосконаленої методики MTBF

Розрахунок показника середнього часу між відмовами є одним з найбільш відповідальних процесів на етапі тестування програмних систем. Від отриманих результатів згідно обчислення метрики залежить рівень стабільності та надійності роботи програми, а також визначення подальшого плану дій з підтримки та обслуговування ПЗ під час безпосередньої експлуатації. Саме тому важливо встановити деякі критерії оцінки ефективності удосконаленої методики роботи з MTBF для перевірки її функціонування та можливості подальшого застосування на етапі тестування ПЗ.

Збільшення точності прогнозування відмов є одним із перших критеріїв оцінки ефективності удосконаленої методики MTBF. Точні та адекватні прогнози відмов мають величезне значення для будь-якої системи або продукту, оскільки вони дозволяють попереджати можливі проблеми, планувати робочі процеси та ефективно витратити ресурси.

З точними прогнозами відмов можна уникнути непередбачуваних ситуацій, забезпечити безперебійну роботу системи та забезпечити надійність для користувачів. Точне передбачення допомагає вчасно планувати роботи з технічного обслуговування та ремонту, які мають на меті збільшити MTBF. Також

даний аспект сприяє оптимізації запасів та ресурсів, що покращує фінансову ефективність підприємства.

Для забезпечення високої точності прогнозування відмов необхідно враховувати багато чинників, які можуть впливати на надійність системи. Важливо також мати можливість моніторити та оновлювати прогнози на основі нових даних та змінних умов. Даний аспект забезпечує постійну адаптацію методики до реальних обставин і підвищує її ефективність. Чіткість та зрозумілість результатів прогнозування є також важливими, оскільки це дозволяє тестувальникам ефективно реагувати на передбачені сценарії та приймати обґрунтовані рішення щодо технічного обслуговування та ремонту.

Скорочення часу на аналіз та оцінку результатів тестування є другим критерієм оцінки роботи покращеного методу МТВФ. Важливим аспектом має бути зменшена тривалість часу, яка необхідна для обробки великої кількості даних та виконання аналізу. В традиційних методиках відповідні процеси можуть вимагати значно більше умовних часових одиниць та людських ресурсів, що призводить до затримок у видачі результатів і прийняття рішень.

Зменшення часу на аналіз і оцінку результатів також дозволяє оперативно реагувати на відмови та події, що може бути критичним у сферах, де надійність має вирішальне значення. Вчасне виявлення проблем та можливостей вдосконалення системи забезпечує підвищену надійність і безперебійну роботу.

До того ж, скорочення часу на аналіз і оцінку дозволяє заощадити ресурси та витрати, що пов'язані з тестуванням і моніторингом, і сприяє підвищенню продуктивності та ефективності інших процесів розробки.

Зменшення кількості відмов є третім та одним із найбільш ключових критеріїв для оцінки ефективності удосконаленої методики МТВФ. Зниження кількості відмов є прямим показником підвищення надійності програмного забезпечення або технічних систем в цілому. Згідно за даним критерієм висока ефективність розрахунку МТВФ має бути підкреслена можливістю виявляти ризики та проблеми у роботі програмної системи на ранніх етапах, що дозволяє вжити заходів для їх попередження.

Зменшення кількості відмов за рахунок розрахунку та аналізу показника MTBF на загальному рівні також сприяє покращенню задоволення користувачів програмного забезпечення або продукту в загальному. Ефективна методика MTBF дозволяє спрогнозувати та попередити збої, що робить процес обслуговування більш раціональним та зменшує час та ресурси, витрачені на виправлення відмов або помилкових ситуацій.

Відповідний критерій також наголошує, що зниження кількісного рівня збоїв може бути важливим у вимірювальних та регулюючих системах, де точність та надійність є критичними. У таких випадках зменшення кількості відмов може сприяти підвищенню точності вимірювань та якості регулювання.

Покращення якості програмного забезпечення є наступним суттєвим критерієм оцінки ефективності удосконаленої методики MTBF. Висока якість ПЗ означає, що програма працює без суттєвих відмов та помилок, що дозволяє користувачам впевнено та ефективно використовувати її. За даним критерієм покращення надійності ПЗ перекладається на менші втрати часу, ресурсів та грошей, які раніше були витрачені на усунення дефектів та вирішення проблем та дефектів під час функціонування ПЗ.

Покращення якості програмного забезпечення на загальному рівні також важливе для збереження та підвищення конкурентоздатності компаній. Висока якість програмного забезпечення може призвести до задоволення клієнтів, що сприяє збільшенню їхньої відданості та лояльності до програмної системи та відповідної команди розробників.

Зручність використання та інтеграції представляє собою ще один критерій оцінки покращеного методу роботи з метрикою середнього часу між відмовами. Важливо, щоб методика була легко впроваджувана в різні проекти та відповідала потребам різних команд розробників, тестувальників та інших учасників процесу розробки програмного забезпечення.

Зручність використання означає, що методика MTBF повинна бути простою, доступною та зрозумілою для тестувальників. Відповідно до цього командам легше впроваджувати методику та користуватися нею на практиці.

Щодо інтеграції, методика МТBF повинна бути сумісною з існуючими інструментами та системами, які вже використовуються в процесі розробки. Інтеграція спрощує збір та аналіз даних, наслідком чого є підвищення точності та цінності результатів.

Зручність використання та інтеграції важливі для того, щоб зробити методику МТBF доступною та використовуваною в широкому спектрі проектів та організацій. Відповідна практика сприяє розповсюдженню надійних практик у галузі розробки програмного забезпечення та допомагає досягти кращих результатів в плані покращення якості та надійності ПЗ.

Економічна ефективність є одним з останніх критеріїв оцінки удосконаленої методики МТBF. Однією з головних цілей даної методики є зменшення ризику та витрат, пов'язаних з відмовами програмного забезпечення. Якщо методика МТBF дозволяє заздалегідь виявити можливі проблеми та виправити їх на ранніх стадіях розробки, тоді це призводить до економії витрат на подальше усунення вад та додаткове тестування. Зменшення збоїв у промисловому виробництві може призвести до економії значних коштів, пов'язаних з ремонтом, заміною або відшкодуванням збитків.

Крім того, ефективність удосконаленої методики МТBF може бути оцінена через покращення якості продукту, що може призвести до збільшення задоволеності клієнтів ПЗ на глобальному рівні. Виправлення вад після випуску програмного продукту на ринок може бути надзвичайно дорогим процесом.

Також слід враховувати, що покращення надійності програмного забезпечення може сприяти зменшенню часу, витрат та ресурсів, які споживаються на усунення збоїв та підтримку програмного продукту під час його експлуатації. Відповідні фактори можуть серйозно вплинути на економічну сторону розробки програмного забезпечення.

Встановлені критерії оцінки ефективності удосконаленої методики роботи з метрикою середнього часу між відмовами покликані забезпечити належний рівень вимірювання показника з метою отримання точних результатів для здійснення управління надійністю програмної системи. Критерії є визначеними в межах рівня

розробки та тестування програм, а також відносяться до глобальних факторів експлуатації різних програмних продуктів користувачами. Згідно з усіма встановленими пунктами покращений метод роботи з показником часу безвідмовної роботи програмного забезпечення повинен працювати злагоджено та представляти загальні та конкретні відомості про стан якості та стабільності функціонування системи.

3.3 Засоби та технології для розробки удосконаленого методу роботи з MTBF

Розробка удосконаленої методики роботи з метрикою середнього часу між відмовами потребує використання сучасних інструментів та технологій. Зокрема, для створення відповідного методу було обрано мову високого рівня програмування Python та відповідні компоненти.

Мова програмування Python – це потужна, високорівнева мова, яка набула величезної популярності завдяки своїй простоті та читабельності коду [23]. Однією з головних переваг Python є його легкість вивчення, завдяки зрозумілому синтаксису та великій кількості доступних ресурсів та документації.

Python підтримує багато галузей програмування, включаючи веб-розробку, наукові обчислення, штучний інтелект, обробку даних, автоматизацію та багато інших. Інтерпретований характер мови дозволяє виконувати код на різних операційних системах без необхідності компіляції. Багатство сторонніх бібліотек та фреймворків робить Python дуже потужним інструментом для вирішення різноманітних завдань.

Ще однією суттєвою перевагою Python є активна спільнота розробників, яка постійно розширює функціональність та підтримку мови. Python також має велику екосистему додаткових інструментів для розробки, включаючи редактори коду, середовища розробки та системи контролю версій. Усе це робить Python

потужним та дружнім інструментом для розробників усіх рівнів, незалежно від їхніх цілей та поставлених задач.

Для виконання поставленого завдання з розробки удосконаленого методу роботи з МТВФ використовуються різні бібліотеки обраної мови. Однією з таких виступає бібліотека NumPy. NumPy (Numerical Python) – це основна бібліотека для наукових обчислень та обробки даних у середовищі мови програмування Python [24]. Вона створена для спрощення роботи з чисельними даними та забезпечення високої швидкості різноманітних обчислень. NumPy надає численні корисні функції та структури даних, які роблять роботу з числами, масивами та матрицями набагато зручнішою.

Бібліотека також містить багато функцій для виконання операцій над масивами, включаючи арифметичні, статистичні, тригонометричні, лінійної алгебри та багато інших. NumPy є популярним інструментом у галузях наукового обчислення, машинного навчання, обробки сигналів, графіки та багатьох інших сферах. Вона часто використовується в поєднанні з іншими бібліотеками для створення комплексних програмних рішень для обробки даних та виконання наукових досліджень. NumPy дозволяє вам легко створювати, опрацьовувати та аналізувати чисельні дані в середовищі Python, забезпечуючи високу продуктивність обчислень.

Іншою бібліотекою, яка застосовується в процесі розробки методу, є Pandas. Бібліотека Pandas – це ще один потужний інструмент для обробки та аналізу даних в мові програмування Python [25]. Вона надає зручні та ефективні структури даних, які дозволяють працювати з таблицями та часовими рядами, що містять інформацію у вигляді різноманітних даних. Pandas дозволяє виконувати завдання, які пов'язані з обробкою даних, включаючи імпорт, фільтрацію, об'єднання та агрегацію даних.

Основні структури даних в Pandas – це DataFrame і Series. DataFrame представляє собою табличну структуру даних зі стовпцями та рядками, Series визначається як одновимірний рядок даних. Обидві структури надають багато методів для зручної роботи з даними.

Pandas також дозволяє працювати з даними в різних форматах, включаючи CSV, Excel, SQL-бази даних та багато інших. Вона є незамінним інструментом для аналізу даних та підготовки даних для моделювання та візуалізації. Бібліотека Pandas широко використовується для забезпечення зручності та продуктивності роботи зі складними даними, що допомагає вирішувати різноманітні завдання у світі аналітики та наукових досліджень.

Для представлення даних роботи удосконаленого методу при розробці використовується ще одна бібліотека – Matplotlib. Бібліотека Matplotlib є одним із найпопулярніших інструментів для візуалізації даних в мові програмування Python [26]. Вона дозволяє створювати різноманітні графіки, діаграми, графіки розсіювання та інші візуальні елементи для подання даних у зрозумілій та інформативній формі. Matplotlib є потужним інструментом для відображення даних у наукових дослідженнях, статистичному аналізі, інженерії тощо.

Основною ідеєю Matplotlib є створення графіків з використанням простих та зрозумілих команд. Бібліотека надає багато можливостей для налаштування стилю графіків, включаючи кольори, шрифти, маркери та інше. Вона підтримує створення графіків у різних форматах, включаючи зображення для вставки в наукові публікації, а також інтерактивні графіки для веб-застосунків.

Matplotlib інтегрується з іншими бібліотеками Python, такими як NumPy та Pandas, що дозволяє легко відображати дані з цих бібліотек. Багата функціональність Matplotlib робить її незамінним інструментом для тих, хто працює з візуалізацією даних та графічними представленнями результатів досліджень та різноманітних розрахунків.

Поряд з використанням мови Python та її компонентів для розробки важливим є використання одного з доступних середовищ. Для створення удосконаленого методу роботи з MTBF було обрано середовище Jupyter Notebook. Jupyter Notebook – це інтерактивне середовище розробки та обчислень, яке стало популярним серед науковців, аналітиків даних та розробників [27]. Воно надає зручний інтерфейс для виконання коду Python та відображення результатів у вигляді тексту, графіків, таблиць та інших візуальних представлень.

Jupyter Notebook базується на веб-технологіях і запускається в браузері. Він дозволяє розділити код на окремі комірки, які можна виконувати незалежно одна від одної. Даний аспект робить процес розробки та експериментів дуже зручним, оскільки дозволяє переглядати результати виконання кожної комірки без необхідності повного перезапуску коду.

Jupyter Notebook також має підтримку візуалізації даних за допомогою бібліотеки Matplotlib та інших. Він надає можливість легко створювати графіки, гістограми та інші графічні представлення прямо в середовищі та відразу відображати їх у документі.

Ще однією важливою особливістю Jupyter Notebook є можливість створення текстових комірок з використанням розмітки Markdown, що дозволяє розробляти документацію, описи аналізу та пояснення результатів. Відповідний фактор робить середовище відмінним інструментом для роботи з даними, створення наукових звітів та спільної роботи.

Вказані інструменти та технології виступають хорошими складовими для здійснення розробки різних програмних систем. Для вирішення поставленої задачі у вигляді створення удосконаленого методу роботи з МТВФ мова Python, її бібліотеки та середовище розробки виступають найкращим рішенням, оскільки злагоджене застосування всіх компонентів надає широкий спектр можливостей для роботи з різноманітними даними та їх візуалізацією, а також для проведення розрахунків різної складності. Саме ці аспекти та їх використання є ключовими для створення та послідовної підтримки удосконаленого методу.

3.4 Реалізація удосконаленого методу роботи з МТВФ

Реалізація покращеного методу роботи з метрикою середнього часу між відмовами була здійснена згідно встановлених задач та з використанням інструментів Python. Кодова частина складається з різних складових, кожна з яких відповідає за власну функціональність.

Для початку роботи удосконаленої методики важливим є підключення необхідних бібліотек. Залучення до проекту відповідних елементів відбувається за допомогою ключового слова `import`. Саме підключення бібліотек здійснюється наступним чином:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Загальна робота методики зосереджується на опрацювання інформації про відмови програмного забезпечення. Основними даними методики виступають час роботи програмного забезпечення відповідно до проведення тестових випробувань та кількість збоїв, яка сталась з вказану тривалість функціонування програми. Саме тому необхідним для методики є отримання відповідних даних для здійснення їх опрацювання. Для цього було обрано один зі способів, а саме читання необхідної інформації з файлу, що має розширення `.csv` та зберігає дані у табличному форматі з використанням роздільників. Даний тип передачі інформації до удосконаленої методики широко використовується під час розробки різного програмного забезпечення за допомогою мови Python, оскільки належна йому бібліотека Pandas дозволяє легко здійснювати як процес читання даних, так і подальшу їх обробку. Безпосередній процес читання даних з файлу виглядає наступним чином:

```
data = pd.read_csv('test_data.csv')
```

Необхідні для роботи методики дані зберігаються у змінну `data`. За читання даних з файлу відповідає метод `read_csv`, який викликається через скорочення назви відповідної бібліотеки Pandas. В даному випадку файл має назву `test_data`, проте вона може бути іншою в залежності від контексту використання методики або інших нюансів.

Після читання необхідної інформації в роботу вступає метод розрахунку показника МТВФ. Оголошення даного методу відбувається за допомогою ключового слова `def` та виглядає так:

```
def calculate_mtbf(failures, operating_time):
    if failures == 0:
        return float('inf')
    return operating_time / failures
```

Відповідний метод має назву `calculate_mtbf`. Основний принцип його роботи полягає у використанні переданих йому даних у вигляді параметрів кількості відмов та часу загальної роботи ПЗ та розрахунку середнього часу між збоями. Метод здійснює обчислення тривалості безвідмовної роботи лише у випадку, якщо такі збої відбувались. Якщо ж програма працювала успішно під час проходження певного тесового сценарію, то метод пропускає даний випадок та переходить до ситуації, де виникнення помилкових ситуацій мало місце. Після розрахунку показника метод повертає відповідне значення до точки його виклику. Дана точка є командою для запуску роботи даного методу. В кодовій частині дана команда має такий вигляд:

```
mtbf_values = data.apply(lambda row: calculate_mtbf(row['failures'],
row['operating_time']), axis=1)
```

Передача даних про відмови та час загальної роботи під час виконання тестів почергово відбувається за допомогою методу `apply`, що включає безпосередній виклик методу `calculate_mtbf`, додатковий параметр `lambda`, що в даному випадку представляє приховану функцію проходження по рядках даних, що зберігаються у вигляді двовимірного масиву у відповідній змінній, та параметр `axis`, що вказує на кількість осей, які потрібно використовувати під час кожного проходження по масиву. Результати розрахунків показника середнього часу між відмовами зберігаються у змінній `mtbf_values`.

Наступний метод, який використовується в межах загальної удосконаленої методики, дозволяє здійснити розрахунок показника надійності системи. Даний метод має назву `Na`. На вхід він приймає два параметри, один з яких представляє показник часу безвідмовної роботи, інший – базове його значення, яке може змінюватись в залежності від типу та специфіки роботи ПЗ. Безпосередній розрахунок надійності здійснюється за допомогою функції `exp` бібліотеки NumPy через її умовне скорочення та відповідного ділення базового показника тривалості

безвідмовної роботи на реальний. Код методу є представлений за допомогою наступного фрагменту:

```
def system_reliability(mtbf, t):
    reliability = np.exp(-t / mtbf)
    return reliability
```

Виклик відповідного методу відбувається звичайним чином та є таким:

```
reliability_values = system_reliability(mtbf_values, 1000)
```

Змінна `reliability_values` зберігає значення показників надійності відповідно до проходження тестових сценаріїв.

Наступний метод методики є основним з-поміж інших. Він представляє удосконалений спосіб роботи з показником середнього часу між відмовами програмного забезпечення. Метод має назву `improved_mtbf_method`. Лістинг коду даного методу є таким:

```
def improved_mtbf_method(data, threshold):
    filtered_data = data[data['failures'] >= threshold]
    mtbf_values = filtered_data.apply(
        lambda row:
            calculate_mtbf(row['failures'], row['operating_time']), axis=1)
    return mtbf_values
```

Удосконалений метод приймає для своєї роботи два параметри. Перший параметр представляє змінну `data`, яка зберігає дані про відмови та операційний час роботи ПЗ, другий параметр містить значення, що відповідає за кількісний поріг збоїв у програмній системі. На початку роботи методу відбувається фільтрація даних про збої за встановленим пороговим значенням. Відповідні результати фільтрації зберігаються у змінній `filtered_data`. Після цього схоже до звичайного способу відбувається безпосередній розрахунок показника часу безвідмовної роботи з використанням вже відфільтрованих даних. Результати роботи покращеного методу зберігаються у змінну `improved_mtbf_values`, яка представлена під час безпосереднього виклику удосконаленого методу. Рядок коду, що відповідає за виклик має такий вигляд:

```
improved_mtbf_values = improved_mtbf_method(data, 5)
```

Поряд з існуючими методами для розрахунку показників середнього часу між відмовами та надійності системи в межах удосконаленої методики реалізовано можливість для візуалізації даних у вигляді графіків. Перший з таких методів має назву `data_vizualization` та відповідає за візуальне представлення вхідних даних. Лістинг коду відповідного методу є таким:

```
def data_vizualization(dataset, xlabel, ylabel):
    plt.plot(dataset, label=ylabel)
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.legend()
    plt.show()
```

Основою для роботи даного методу є бібліотека `Matplotlib`, яка застосовується через умовне скорочення. Для виконання основних функцій відображення інформації метод використовує три параметри, які повинні бути передані на вхід. Перший параметр відповідає за набір даних для відображення, другий – за назву осі X, третій – за назву осі Y та позначення назви лінії графіку в легенді. Створення образу відображення відбувається за допомогою бібліотечної властивості `plot`. Також за допомогою пакетних властивостей `xlabel` та `ylabel` відбувається визначення осей графіка та легенди графіка за допомогою опції `legend`. Функція `show` відповідає за остаточний виклик відображення графіка відповідно до визначених характеристик.

Для візуалізації за удосконаленою методикою реалізовано декілька представлень різних даних. Для виклику загального методу візуалізації `data_vizualization` відповідають різні, але схожі за сигнатурою команди. Перші дві з них мають такий вигляд:

```
data_vizualization(data['failures'], '№ of Test Case', 'Failures')
data_vizualization(data['operating_time'], '№ of Test Case', 'Operating Time')
```

Перша команда відповідає за візуалізацію даних про відмови згідно виконання тестових кейсів. Відповідно до цього відбувається передача інформації

до методу у вигляді сукупності даних про відмови та назв основних компонентів графіка. У випадку другої команди передача даних відбувається в такому ж порядку, проте самі дані дещо відрізняються, оскільки даний рядок коду відповідає за відображення загального операційного часу програмної системи згідно виконання тестових сценаріїв.

Також удосконалена методика здатна візуально відображати результуючі дані, що були утворені в ході розрахунків. Зокрема, було реалізовано візуалізацію показників надійності системи. Загальне представлення даних відбувається за допомогою методу `data_vizualization`. Виклик даного методу для відображення стану надійності програмного забезпечення є схожим до попередніх команд та відбувається наступним чином:

```
data_vizualization(reliability_values, '№ of Test Case', 'Reliability Values')
```

В якості першого параметру визначено показники надійності програми, які були розраховані за допомогою методу `system_reliability`. Наступні параметри команди відповідають за назви структурних елементів графічного подання.

Іншим результуючим показником, для кого було реалізована можливість візуального представлення, є удосконалений показник часу безвідмовної роботи. Саме представлення відбувається за допомогою методу `mtbf_vizualization`. Лістинг коду методу є таким:

```
def mtbf_vizualization(first_dataset, second_dataset, first_label, second_label,
xlabel, ylabel):
    plt.plot(first_dataset, label=first_label)
    plt.plot(second_dataset, label=second_label)
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.legend()
    plt.show()
```

Метод `mtbf_vizualization` є схожим за своєю структурою та принципом роботи з попереднім методом для візуалізації `data_vizualization`. Єдиною відмінністю є більша кількість параметрів, яка приймається на вхід. В даному випадку таких властивостей шість: перші дві відповідають за основну інформацію

для відображення, що містить показники звичайного та удосконаленого методу розрахунку часу безвідмовної роботи, наступні два вказують на назви ліній графіка, що подаються в легенді, останні два вказують на назви осей графічного подання. Виклик методу `mtbf_vizualization` також відбувається за схожою до попередніх процедурою. Код виклику є таким:

```
mtbf_vizualization(mtbef_values, improved_mtbef_values, 'Standard MTBF', 'Processed
MTBF', '№ of Test Case', 'MTBF')
```

Відповідно до виконання вказаних команд виклику методів візуалізації буде здійснено представлення необхідних графіків. Їх подання буде виконане з усіма відповідними структурними компонентами згідно до контексту даних для здійснення подальшого їх опрацювання.

Удосконалена методика роботи з метрикою середнього часу між відмовами представляє загальний підхід для виконання розрахунків показника та здійснення його послідовного аналізу. Покращена методика може використовуватись автономно або з використанням інших додаткових інструментів для виконання обчислень. Інтеграція з іншими показниками якості також можлива. Для цього необхідно додати необхідні вхідні дані та відповідну складову розрахунку вибраного показника до кодової частини. Всі відповідні реалізовані елементи є адаптованими до змін та можливих доповнень в разі необхідності. Удосконалена методика має на меті покращити процес розрахунку часу безвідмовної роботи, полегшити процес аналізу та оцінювання результатів обчислень та допомогти у прийнятті важливих рішень для підвищення та підтримки рівня надійності ПЗ.

3.5 Рекомендації щодо провадження удосконаленої методики MTBF в процес тестування програмного забезпечення

Удосконалена методика роботи з метрикою середнього часу між відмовами може бути використана для виконання тестування різних програмних продуктів. Її застосування здатне позитивно впливати на перебіг процесів тестування, розробку

тестових стратегій та перевірки загального функціонування програмних систем та їх компонентів. Впровадження удосконаленої методики в загальний процес тестування не вимагає проведення серйозних заходів та може бути здійснена в декілька етапів.

Адаптація методики МТВФ до конкретного проекту є одним з перших етапів впровадження покращеної методики в процес тестування програмного забезпечення. Даний крок є необхідним, оскільки кожен проект може мати свої унікальні вимоги, особливості та обмеження, які потрібно враховувати для ефективного визначення та підвищення надійності програми.

Під час адаптації методики до конкретного проекту, перш за все, необхідно провести детальний аналіз вимог та специфікацій проекту. Важливо визначити, які вимоги до надійності встановлені для даного програмного забезпечення, а також які критичні функції та процеси впливають на його надійність.

Необхідно також враховувати конкретні типи відмов, які можуть бути критичними для проекту, та здійснювати їх постійний моніторинг. Правильно налаштована методика має включати в себе вимоги до збору даних про збої, зокрема, час їх виникнення та тривалість відновлення.

Удосконалена методика має враховувати розмір та складність програмного забезпечення, обсяг тестування, доступність ресурсів та технічні можливості. Методика повинна бути адаптована до обставин проекту з урахуванням доступних інструментів та підходів. Важливо також систематично оцінювати результати тестування ПЗ та вживати заходи для вдосконалення, які дозволять підвищити рівень надійності програми.

Інтеграція удосконаленої методики МТВФ з існуючими процесами тестування є ще одним етапом у впровадженні покращеної методики для оцінки надійності програмного забезпечення. Відповідний процес передбачає об'єднання нового підходу з існуючими структурами тестування з метою забезпечення максимальної ефективності та співпраці.

Один із ключових аспектів інтеграції – це впровадження нових етапів та процедур у процес тестування програмної системи відповідно до його специфіки з

метою постійного отримання інформації для розрахунку показника МТBF. Така інтеграція дозволяє отримати більш повну та достовірну інформацію про відмови програмного забезпечення.

Навчання команди тестування та розробки на загальному рівні також є важливим етапом впровадження удосконаленої методики МТBF в процес тестування програмного забезпечення. Впровадження нових методів та інструментів для оцінки надійності ПЗ може вимагати від команди специфічних знань та навичок для здійснення ефективного оперування нововведеннями.

Для впровадження удосконаленої методики розрахунку МТBF команді тестувальників не потрібно докладати багато зусиль. Базові знання та навички роботи зі звичайним способом розрахунку показника є необхідною та достатньою умовою для використання удосконаленої методики. Для повного розуміння роботи методики необхідно провести аналіз ключових нюансів роботи всіх розроблених методів. Особливо важливо звернути увагу на покращений метод розрахунку середнього часу між відмовами та ознайомитись з фільтрацією даних. Відповідні дії не є складними, що дозволить команді розробників швидко залучити удосконалену методику до етапу тестування ПЗ.

Моніторинг та аналіз результатів є важливим кроком впровадження покращеної методики МТBF в процес тестування ПЗ. Після навчання команди та адаптації методики до конкретного проекту, необхідно систематично стежити за її впровадженням та оцінювати її ефективність.

На цьому етапі важливо встановити ключові показники продуктивності, які дозволять визначити, чи досягнуті поставлені цілі та об'єктиви удосконаленої методики. Для забезпечення моніторингу можна використовувати спеціалізовані інструменти та програми, які дозволять збирати та аналізувати необхідну інформацію. Завдяки цьому команда зможе в режимі реального часу слідкувати за показниками та вчасно виявляти будь-які аномалії чи невідповідності. На основі результатів аналізу, команда може вносити корективи та вдосконалювати методику та постійно покращувати процес тестування програм та забезпечувати високу якість розробленого продукту.

Корекція та удосконалення методики на основі отриманих результатів є останнім кроком, який може бути активним під час впровадження або вже повноцінного використання удосконаленої методики МТВФ на етапі тестування програмних систем. Після того, як було проведено моніторинг та аналіз результатів впровадження методики, виявлені ключові показники, прийшов час можливе додаткове налаштування методики для досягнення більшої ефективності та відповідності поставленим цілям.

Спершу, команда повинна проаналізувати результати спостережень та виявити основні проблеми чи області, де методика може бути покращена. Після ідентифікації слабких місць команда повинна розробити конкретний план корекції та удосконалення методики. Відповідний перелік дій може включати в себе розробку нових засобів, а також оновлення інструментів та програм для моніторингу та аналізу. Важливо враховувати особливості конкретного проекту та його потреби.

Після внесення коректив та удосконалень методику слід впроваджувати знову та повторно проводити моніторинг та аналіз результатів. Відповідний ітеративний підхід дозволяє поступово покращувати ефективність методики та досягати більш точних та адекватних результатів в роботі з МТВФ.

Впровадження удосконаленої методики в процес тестування програмних продуктів є клопітким видом діяльності для забезпечення високого рівня вимірювання та підтримки надійності ПЗ. Злагоджене впровадження методики відповідно до визначених етапів дозволить отримати більш ефективний та продуктивний спосіб розрахунку показника часу безвідмовної роботи, що безпосередньо дасть можливість здійснювати аналіз отриманих результатів та приймати рішення щодо керування якістю системи. Успішний результат впровадження удосконаленої методики дозволить команді тестувальників здійснювати точну оцінку функціонування програмного забезпечення та відкрити нові можливості для пошуку інших покращень.

3.6 Вибір конкретного проекту для валідації роботи удосконаленої методики МТВФ

Етап тестування різноманітних програмних проектів потребує різних рішень для підвищення загальної ефективності розробки. Удосконалена методика роботи з метрикою середнього часу між відмовами може слугувати хорошим інструментом для відстеження та підтримки якості програмного забезпечення будь-якого типу та призначення. Загальна перевірка роботи удосконаленої методики роботи з МТВФ може бути здійснена на одному з вже розроблених проектів. Для виконання відповідного процесу було обрано мобільний додаток «Smart Health» згідно до специфіки його роботи та особливостей виконання різних функцій. Важливим для перевірки покращеної методики є використання даних про проведення загального тестування додатку.

Мобільний додаток «Smart Health» – це програмний продукт, призначений для покращення доступності та ефективності в галузі охорони здоров'я. Додаток надає можливості для моніторингу здоров'я, консультацій з лікарем, відстеження медичних показників та інші функції для взаємодії з медичними послугами.

«Smart Health» дозволяє користувачам швидко отримувати відповіді від лікарів онлайн та вести електронні медичні записи. Завдяки інтеграції з сучасними технологіями, додаток надає зручний і ефективний спосіб процесу отримання допомоги та догляду за здоров'ям.

Вибір проекту мобільного додатку «Smart Health» для валідації удосконаленої методики роботи з МТВФ є розумним та стратегічним кроком. Даний проект має свої унікальні властивості та особливості, які можуть стати важливими в контексті валідації та апробації вдосконаленої методики.

Вказаний мобільний додаток та інше програмне забезпечення такого типу мають високі критерії та вимоги до відтворення власного функціоналу, зважаючи на рід діяльності та сферу їх застосування. Зважаючи на це, існують певні аспекти, які відіграли важливу роль при виборі проекту мобільного додатку «Smart Health» для здійснення перевірки роботи удосконаленої методики.

Одним з перших аспектів є висока вимога до якості та надійності ПЗ. Особливістю вибору є потреба в надійному та безперебійному функціонуванні мобільного додатку, оскільки він взаємодіє з важливими аспектами здоров'я та медичними послугами користувачів.

Медичні додаток «Smart Health» має важливу роль у наданні доступу до медичної інформації та послуг в будь-який час. Враховуючи відповідний нюанс, вимагається високий рівень якості, щоб уникнути можливих помилок та забезпечити надійність медичної інформації та функцій, наданих додатком.

Вибір додатку за відповідним аспектом дозволяє не лише перевірити ефективність удосконаленої методики МТВФ у реальному середовищі високого стандарту, але й гарантує, що програмне забезпечення відповідає високим вимогам до якості та надійності, необхідним для успішного функціонування в галузі охорони здоров'я.

Наступною важливою особливістю, що стала причиною для вибору додатку для валідації, є його різноманітна функціональність. Розмаїття функціональності мобільного додатку «Smart Health» визначається його широким спектром сервісів та можливостей, спрямованих на забезпечення повноцінного медичного обслуговування та підтримки користувачів у галузі охорони здоров'я. Враховуючи відповідний аспект, додаток стає ідеальною кандидатурою для валідації удосконаленої методики МТВФ.

Мобільний додаток «Smart Health» може включати в себе функції телемедицини, моніторингу здоров'я, нагадувань про прийом ліків, спостереження за фізичною активністю, аналіз медичних показників та інші інструменти, спрямовані на поліпшення якості медичного обслуговування та підтримки пацієнтів. Така різноманітність функціональності викликає велику кількість сценаріїв використання та можливих точок відмов, що робить додаток ідеальним для перевірки покращеної методики роботи МТВФ в реальних умовах.

Забезпечення стійкості та надійності в умовах такого різноманіття функцій стає критичною вимогою. Висока кількість різних функцій створює виклик для

удосконаленої методики МТВФ та вимагає її адаптації до різних варіантів використання для забезпечення оптимальної надійності та ефективності додатку.

Іншим аспектом вибору програмного продукту «Smart Health» для перевірки методики є складність проекту та його розмір. Велика ієрархічна структура компонентів та значний розмір ПЗ робить його важливим об'єктом для здійснення процесу перевірки методики МТВФ. Складність проекту визначається не лише кількістю функцій, але й їх взаємодією, що віддзеркалює складну внутрішню архітектуру та логіку додатку.

Великий розмір проекту означає значну кількість кодової бази та ресурсів, що додає складності управління та визначення його надійності. Існування значної кількості функцій і модулів також ускладнює процес тестування та вимагає високої ефективності методів, що оцінюють надійність.

Валідація удосконаленої методики МТВФ на такому рівні складності дає можливість оцінити її ефективність у великих та різнорідних проектах. Здатність методики пристосовуватися до розміру проекту та управління його складністю стає важливою характеристикою в контексті загального процесу валідації.

Висока динаміка розвитку є ще однією ключовою особливістю вибору медичного додатку для перевірки функціонування покращеної методики МТВФ. Зміни та оновлення в додатку відбуваються на регулярній основі, що вимагає постійного моніторингу та адаптації тестових стратегій. Динаміка розвитку також означає появу нових функцій та модулів, а також оптимізацію існуючих, що впливає на стабільність та надійність системи.

Відповідний аспект також полягає в здатності швидко адаптуватися до змін у функціоналі додатку та забезпечувати надійність під час інтенсивного розвитку. Валідація на проекті «Smart Health» дозволяє перевірити, наскільки методика справляється з вимогами до надійності в умовах постійної еволюції ПЗ.

Враховуючи високу динаміку розвитку, важливо, щоб удосконалена методика МТВФ була гнучкою та готовою до швидкої інтеграції нових елементів та змін, що стає ключовим фактором у контексті проекту «Smart Health».

Наступною особливістю, яка стала причиною вибору додатку для валідації методики, є підтримка різних платформ та операційних систем. Відповідний аспект робить додаток «Smart Health» хорошим варіантом для перевірки роботи удосконаленої методики MTBF, оскільки він вказує на розширений спектр тестування в різних середовищах з використанням багатьох пристроїв.

Здатність додатку працювати на різних операційних системах, таких як Android і iOS, а також на різних типах пристроїв, включаючи смартфони та планшети, створює складні умови для тестування його надійності. Удосконалена методика MTBF повинна забезпечити стабільність та надійність функціонування додатку на всіх цих платформах, враховуючи їхні особливості та вимоги. Універсальність роботи на різних апаратних та програмних складових також означає, що покращена методика повинна бути достатньо адаптованою для різних технологій та ефективно працювати в різноманітних умовах експлуатації.

Останнім аспектом додатку «Smart Health», що вплинув на вибір для валідації, є наявність довгострокових даних про відмови. Надійні дані про відмови важливі для ретельного тестування та оцінки продуктивності методики на протязі тривалого періоду та є критичною умовою для використання покращеної методики роботи з метрикою середнього часу між відмовами ПЗ.

Можливість аналізувати довгострокові відмітки часу та взаємодію з іншими факторами в додатку надає можливість ретельно дослідити поведінку системи протягом тривалого періоду експлуатації. Даний фактор дозволяє ефективно оцінити, наскільки удосконалена методика MTBF відповідає реальним умовам експлуатації додатку та як ефективно вона передбачає та управляє відмовами.

На основі довгострокових даних можна виявити тенденції та патерни відмов, що допомагає зрозуміти, які чинники впливають на стабільність додатку в залежності від тривалості його роботи. Такий аналіз є ключовим для переконання в тому, що удосконалена методика MTBF є не лише ефективною в короткостроковій перспективі, але й забезпечує стійкість та надійність протягом тривалого періоду використання.

Кожен з аспектів мобільного додатку «Smart Health» вказує на його високу функціональність та широкий спектр програмних можливостей, що є ключовою підставою для вибору його як основного валідатора удосконаленої методики роботи з МТВФ. Для додатку важливим є забезпечення стабільного та продуктивного виконання всіх його функцій. Відповідні фактори визначають основні завдання для загальної роботи покращеної методики. Перевірка її функціонування за допомогою мобільного додатку «Smart Health» допоможе оцінити її ефективність, здатність впливати на процеси тестування ПЗ та керувати надійністю та якістю програмних продуктів.

3.7 Опис процесу тестування обраного проекту із застосуванням удосконаленої методики роботи з МТВФ

Для перевірки роботи удосконаленої методики МТВФ необхідною умовою є її застосування на етапі тестування обраного проекту мобільного додатку «Smart Health». Важливим аспектом для вимірювання показника часу безвідмовної роботи є збір та аналіз даних про збої програмного забезпечення на протязі різних тестових ситуацій. Одним з перших етапів тестування є розробка відповідних тестових сценаріїв для перевірки роботи функціоналу додатку.

Високі вимоги додатку «Smart Health» визначають важливі рамки для створення різних тестових випадків. Для оцінки роботи додатку аналізуються позитивні та негативні моменти, які можуть виникнути під час його функціонування, та на основі цього здійснюється формування повного переліку тестів. Список розроблених тестових сценаріїв згідно всіх важливих нюансів мобільного додатку «Smart Health» є таким:

а) Авторизація користувача:

- 1) Успішний вхід з вірними обліковими даними;
- 2) Невдалий вхід з невірними обліковими даними;
- 3) Відновлення забутого пароля.

б) Реєстрація нового користувача:

- 1) Успішна реєстрація з валідними даними;
- 2) Невдалий спроба реєстрації з хибними даними;
- 3) Перевірка унікальності електронної пошти.

в) Створення та редагування профілю користувача:

- 1) Додавання особистої інформації (вік, стать, зріст, вага);
- 2) Зміна налаштувань приватності профілю;
- 3) Завантаження фотографії профілю.

г) Відстеження фізичної активності:

- 1) Додавання нової фізичної активності;
- 2) Редагування відомостей про активність;
- 3) Видалення активності;
- 4) Синхронізація зі зовнішніми пристроями.

д) Відстеження харчування:

- 1) Додавання продуктів харчування та створення прийомів їжі;
- 2) Редагування інформації про прийом їжі;
- 3) Видалення прийому їжі;
- 4) Пошук продуктів харчування в базі даних.

е) Нагадування та сповіщення:

- 1) Налаштування нагадувань про фізичну активність;
- 2) Налаштування нагадувань про прийоми їжі;
- 3) Налаштування сповіщень про досягнення цілей (наприклад, кількість кроків на день, спожиті калорії);
- 4) Управління налаштуваннями сповіщень (вібрація, звук, повідомлення).

ж) Цілі та статистика:

- 1) Встановлення особистих цілей (кількість кроків, спожиті калорії, час тренувань);
- 2) Відображення статистики за день, тиждень, місяць;
- 3) Створення звітів про досягнення та відхилення від цілей;

- 4) Аналіз прогресу та рекомендації щодо покращення результатів.
- 3) Взаємодія з технічною підтримкою:
 - 1) Надсилання запитів або повідомлень про проблеми;
 - 2) Отримання відповідей або рекомендацій від технічної підтримки;
 - 3) Зворотний зв'язок та пропозиції щодо покращення додатка.

Розроблені тестові сценарії виступають базовими критеріями для оцінювання роботи мобільного додатку. Їх застосування та послідовне відтворення дозволяє перевірити функціонування програми в різних умовах, а також виявити та виправити потенційні проблеми з продуктивністю, безпекою, стабільністю та надійністю його функціонування.

На наступному етапі тестування відбувається його безпосередній процес. На основі сформованих тестових сценаріїв відбувається проведення різних типів випробувань. Функціональні тести включають у себе перевірку основних функцій додатку. Інтеграційні тести оцінюють взаємодію різних компонентів додатку для забезпечення їх спільної роботи та вирішення можливих конфліктів. Стресові тести спрямовані на визначення меж системи, шляхом викладення її ресурсів на великі навантаження та перевірку того, наскільки добре додаток витримує інтенсивні умови використання.

Важливим є виконання тестів в реальних умовах, оскільки нерідко бувають ситуації, коли робота додатку залежить від зовнішніх факторів. Відповідний аспект сприяє виявленню та усуненню проблем, які можуть виникнути безпосередньо при реальному використанні додатку.

Під час процесу тестування відбувається збір та аналіз різної інформації. Зокрема, проводиться фіксація даних про роботу додатку в контексті виникнення відмов або певних помилкових ситуацій функціонування. Відповідна інформація є критичною для визначення стабільності та надійності додатку під час реального використання, а також є необхідною для вимірювання часу успішної роботи за допомогою як традиційних методів, так і удосконаленої методики.

Аналіз отриманих даних про відмови дозволяє ідентифікувати конкретні причини неполадок. Їх детальний аналіз є ключовим етапом для вдосконалення

загальної якості та надійності мобільного додатку «Smart Health». Під час вивчення часових інтервалів між відмовами виявляються патерни та тенденції, пов'язані з виникненням різних проблем, що дозволяє розробникам програмного забезпечення ефективно втручатися та вирішувати виявлені дефекти.

Після збору необхідної інформації відбувається безпосередній розрахунок MTBF за допомогою удосконаленої методики. Результати цього розрахунку надають об'єктивну міру часу роботи додатку у безвідмовному стані.

Разом із MTBF використовуються інші показники для комплексної оцінки ефективності тестування. Аналіз цих показників дозволяє не тільки визначити, наскільки часто відбуваються відмови, але і виявити ефективність процесів виявлення та виправлення дефектів.

На подальших етапах загального процесу тестування відбувається подальший моніторинг та загальні дії щодо покращення якості мобільного додатку. Після проведення розрахунку MTBF та використання інших інструментів отримані результати систематично відстежуються протягом усього проекту. Мета даного процесу полягає в оцінці впливу удосконаленої методики на ефективність процесу тестування. Зокрема, аналізуються показники, що відображають зниження кількості відмов, скорочення часу виявлення та виправлення дефектів, а також поліпшення загальної надійності програми.

На основі проведеного моніторингу та аналізу його результатів вносяться корективи в процес тестування для подальшого підвищення ефективності роботи програмного забезпечення. Даний процес включає в себе модифікації тестових сценаріїв, оптимізацію процесів тестування, а також оновлення вимог до програмного продукту «Smart Health».

Після завершення всіх етапів валідації та внесення змін відбувається оцінка всіх отриманих результатів для перевірки ефективності удосконаленої методики MTBF. Особлива увага приділяється аналізу показників ефективності тестування, надійності ПЗ та відповідності заявленим вимогам та критеріям.

Загальний процес тестування мобільного додатку «Smart Health» відбувається відповідно до строго визначених етапів. Злагоджені дії на кожному з

кроків дозволяють виконати точне оцінювання роботи додатку та в підсумку підвищити якість та надійність програмного продукту. Застосування удосконаленої методики роботи з МТBF здатне покращити здатність виявлення слабких місць додатку та підвищити час його безвідмовного функціонування. Результати застосування удосконаленої методики та відповідний їх аналіз представлено в наступному розділі.

4 РЕЗУЛЬТАТИ ЗАСТОСУВАННЯ УДОСКОНАЛЕНОЇ МЕТОДИКИ РОБОТИ З МЕТРИКОЮ МТВФ ТА ШЛЯХИ ПОДАЛЬШОГО ПОКРАЩЕННЯ

4.1 Представлення результатів застосування удосконаленої методики

Застосування удосконаленої методики роботи з метрикою середнього часу між відмовами дозволяє провести ефективне оцінювання роботи програмного забезпечення. Під час проведення тестування обраного для валідації проекту мобільного додатку «Smart Health» було здійснено збір даних, які є необхідними для роботи методики. Зокрема, було взято загальний операційний час роботи додатку відповідно до кожного тестового сценарію та кількість збоїв, які виникали під час проведення тестів.

Удосконалено методика роботи з показником МТВФ здатна представляти не лише кінцеві результати розрахунків, але й здійснювати відтворення вхідних даних. Відповідно до цього була реалізована можливість візуалізації вхідної інформації часу роботи та відмов. Загальні результати візуалізації відповідних даних мобільного додатку за удосконаленою методикою подані на рисунку 4.1 та рисунку 4.2.

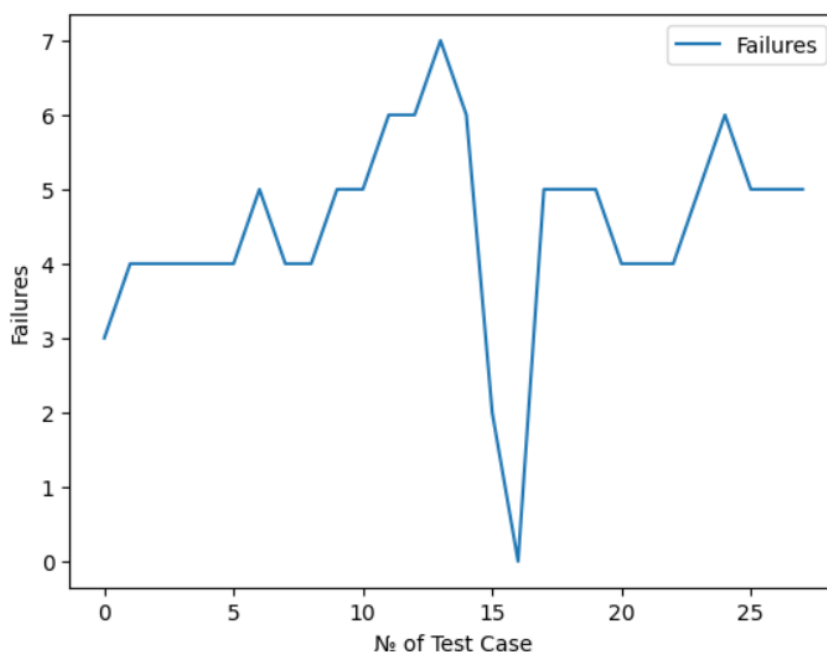


Рисунок 4.1 – Результат візуалізації вхідних даних про відмови ПЗ

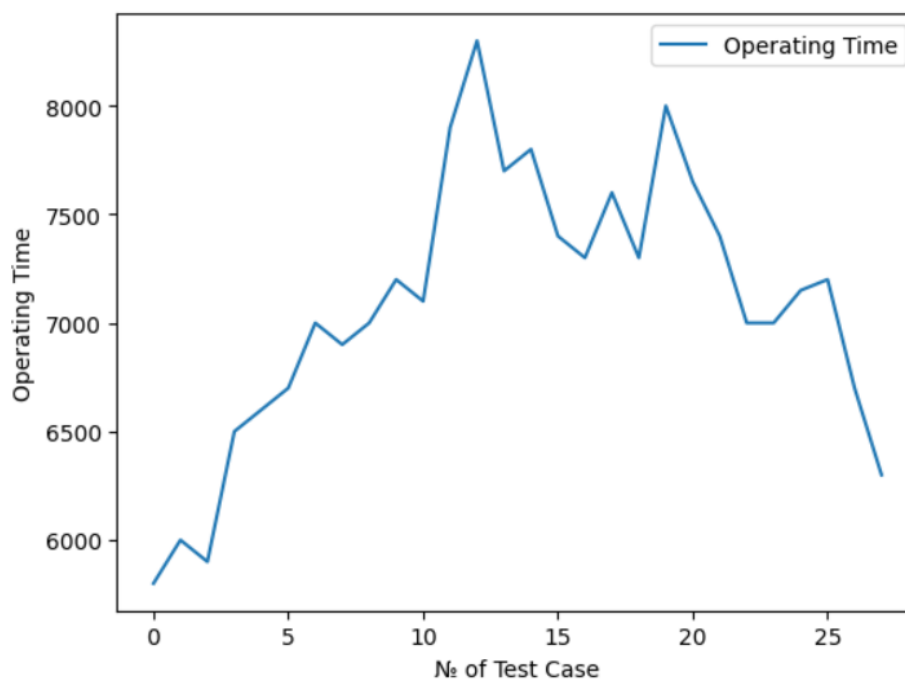


Рисунок 4.2 – Результат візуалізації вхідних даних про загальний час роботи програмного забезпечення

Після візуалізації вхідних даних за удосконаленою методикою можливий розрахунок значень надійності роботи мобільного додатку відповідно до виконання тестових сценаріїв. Відповідні числові параметри також можуть бути представлені наочно в результаті роботи удосконаленої методики. Візуалізація значень надійності подана на рисунку 4.3.

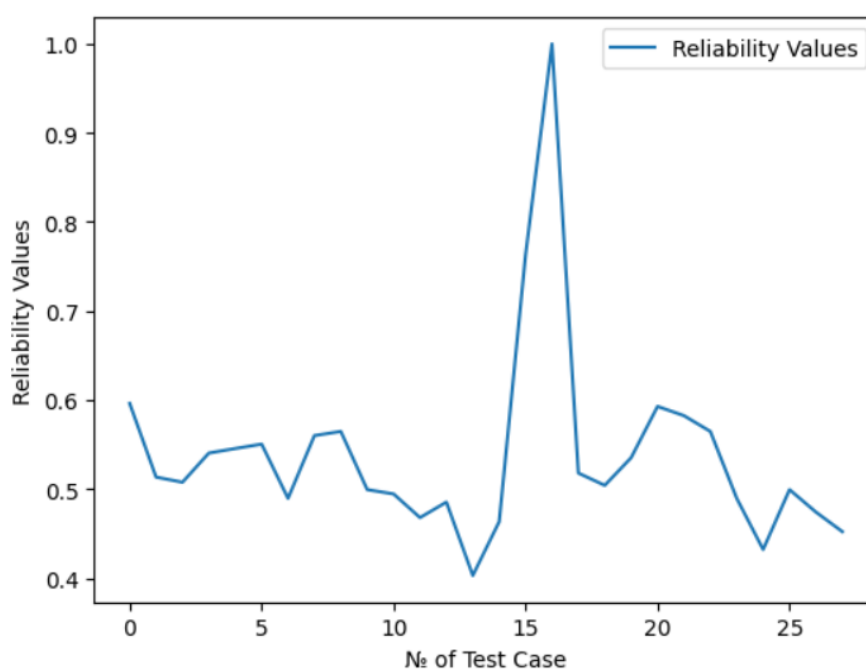


Рисунок 4.3 – Результат візуалізації значень надійності

Подальша робота удосконаленої методики зосереджується на покращеному способу розрахунку показника середнього часу між відмовами. Обчислення здійснюються на основі зібраних даних про помилкові ситуації роботи мобільного додатку. Представлення результатів розрахунку показника MTBF також відбувається в наочному вигляді за допомогою графіка. Відповідна візуалізація результатів подана на рисунку 4.4.

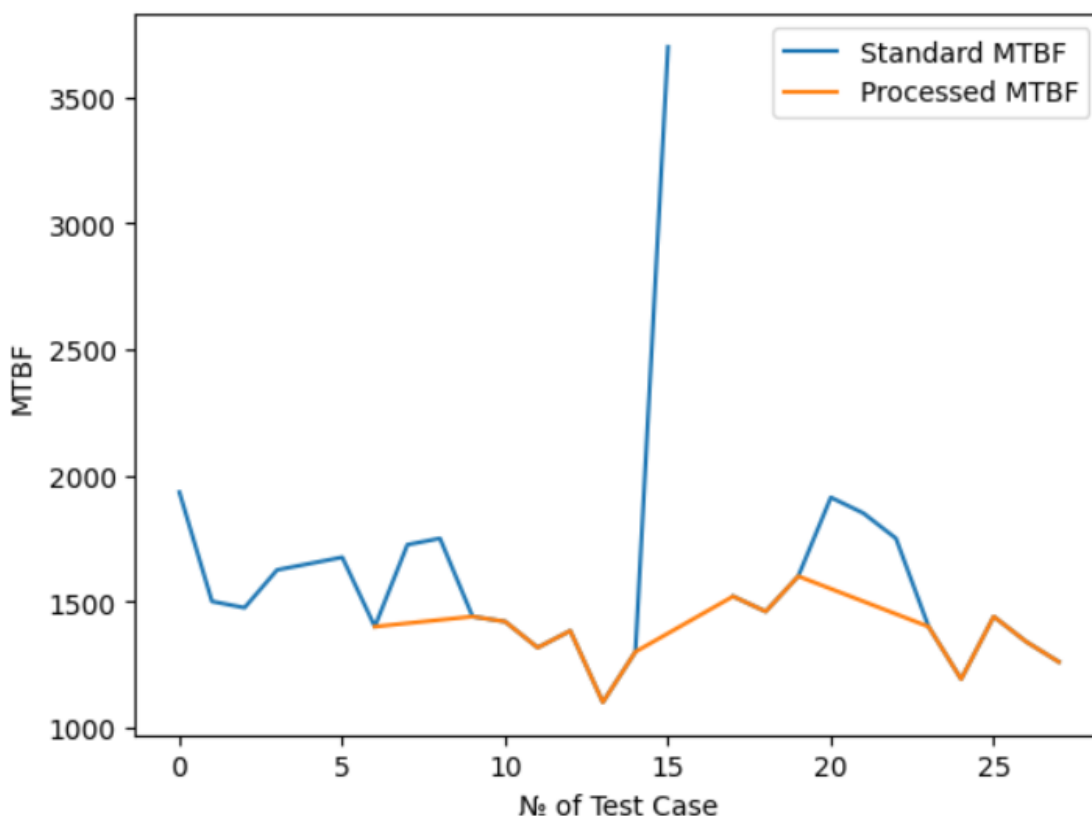


Рисунок 4.4 – Результати роботи удосконаленої методики

Здійснення розрахунку показника часу безвідмовної роботи за удосконаленою методикою відбувається реалізованим способом. Відповідні обчислення здійснюються у визначеному порядку. Візуалізація вхідних даних та отриманих результатів розрахунків дозволяє швидко виявляти слабкі та вразливі місця в роботі програмного забезпечення та виконувати загальну оцінку функціонування програмного продукту. Відтворення відповідних процесів надає можливість здійснювати ефективне управління надійністю та стабільністю роботи програмного забезпечення.

4.2 Аналіз результатів застосування удосконаленої методики

Представлення результатів роботи удосконаленої методики дозволяє виконати їх аналіз та описати ключові аспекти її застосування. Методика добре проявила себе під час виконання всіх необхідних розрахунків з використанням даних обраного проекту для валідації. Загальна робота покращеної методики є швидкою та точною, що позитивно впливає на процес перевірки роботи ПЗ.

Після впровадження удосконаленої методики МТBF у проекті мобільного додатку «Smart Health» виявлено позитивну динаміку роботи. Встановлено, що відповідна методика здатна швидко аналізувати функціонування програмного забезпечення в цілому та його компонентів окремо. В загальному вона впливає на підвищення якості програмного забезпечення та працює продуктивніше порівняно з класичними методами розрахунку МТBF.

Додатково в результаті аналізу було виявлено наступні переваги удосконаленої методики:

- глибше розуміння взаємозв'язку між різними аспектами тестування дозволяє розробникам точніше визначати проблемні фактори та встановлювати пріоритети для виправлення дефектів;

- підвищення швидкості виявлення та усунення відмов можливе внаслідок інтеграції удосконаленої методики МТBF з сучасними автоматизованими системами збору та аналізу даних, що сприяє оперативному виявленню нових відмов і дозволяє команді швидше реагувати на них;

- впровадження змін у процес розробки та тестування програмного забезпечення відбувається з метою зменшення кількості відмов та покращення надійності продукту, що на загальному рівні сприяє підвищенню задоволеності користувачів та довіри до програми.

Загалом, аналіз результатів використання удосконаленої методики підтверджує її ефективність у вдосконаленні процесу тестування програмного забезпечення. Впровадження удосконаленої методики МТBF на основі реального проекту призвело до покращення якості продукту, оптимізації витрат на

тестування та усунення відмов, а також сприяло поліпшенню загального рівня розробки програмного забезпечення.

На основі отриманих результатів можна рекомендувати удосконалену методику MTBF для використання в інших проектах з розробки програмного забезпечення, особливо в тих випадках, де висока якість та надійність є критичними факторами. Однак перед впровадженням удосконаленої методики MTBF в інші проекти важливо провести аналіз специфіки конкретного проекту та можливості адаптації методики до конкретних умов розробки.

4.3 Оцінка ефективності удосконаленої методики MTBF в реальних умовах

Застосування удосконаленої методики має позитивний вплив на керування рівнем якості програмного забезпечення. Результати використання та їх аналіз служать відправною точкою для залучення методики до етапу тестування більшої кількості програмних проектів. Іншим аспектом, що підтверджує відповідне твердження, є оцінка ефективності даної методики в реальних умовах.

За результатами застосування удосконаленої методики було виявлено, що вона здатна позитивно впливати на роботу мобільного додатку «Smart Health». Зокрема, її використання дозволяє зменшити кількість відмов функціонування додатку. Результат зниження відмов відображений на рисунку 4.5.

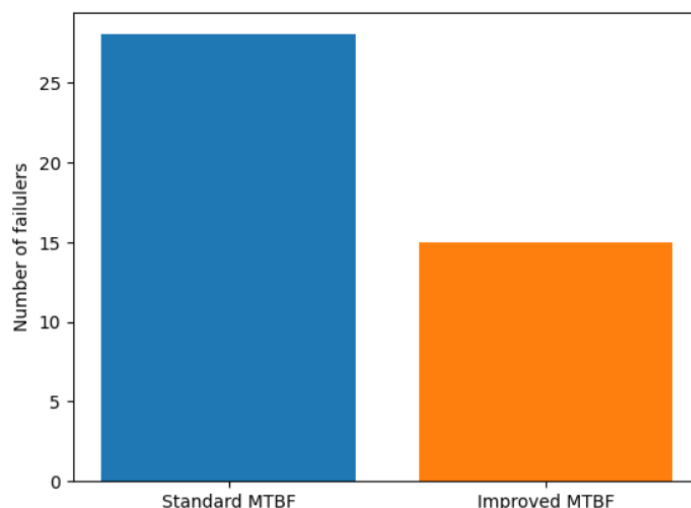


Рисунок 4.5 – Результат зниження відмов ПЗ після застосування методики

Після застосування методики спостерігається зниження кількості відмов у додатку на 35%. Даний аспект свідчить про покращення стабільності та надійності програмного забезпечення.

Методика здатна збільшити загальний час між відмовами програмного забезпечення. Відповідно до застосування такий показник може сягати 30% в порівнянні зі звичайними способами вимірювання. В результаті користувачі будуть зіштовхуватись з меншою кількістю проблем та збоїв у роботі додатка.

Важливим аспектом удосконаленої методики також виступає її здатність до ефективнішого виконання різних процесів на етапі тестування. Завдяки покращеній методиці час та витрати на тестування можуть зменшитися на 25%, що дозволяє розробникам більш ефективно використовувати свій час на розвиток та покращення додатка. Удосконалена методика МТВФ здатна допомогти краще контролювати процес перевірки функціонування ПЗ, аналізувати результати та приймати своєчасні рішення щодо вдосконалення продукту. Відповідні фактори допомагають забезпечити більш ефективне управління ресурсами та підвищити продуктивність розробки.

Удосконалена методика МТВФ володіє можливістю гнучкої адаптації до змін у процесі розробки та вимогам проекту. Вона методика може бути застосована в різних проектах, незалежно від їхнього рівня складності, враховуючи особливості роботи команди та наявні ресурси.

Удосконалена методика МТВФ може також працювати в поєднанні методами оцінки якості, такими як ємнісне тестування, стрес-тестування та перевірка безпеки. Відповідне застосування надає можливість отримати більш повне розуміння стану проекту та забезпечити комплексний підхід до покращення надійності програмного забезпечення.

Використання покращеної методики на загальному рівні використання програмного забезпечення здатне збільшити кількість користувачів та підвищити їхній рівень задоволеності та відданості програмному продукту. Оцінка ПЗ серед користувачів може бути підвищена за рахунок більшої якості роботи програми.

Також може спостерігатись збільшення загальної аудиторії програмного забезпечення, а також кількість активних його користувачів.

Застосування удосконаленої методики МТВФ в реальних умовах та оцінка ефективності відповідного процесу вказує на те, що вона сприяє більш точній оцінці результатів тестування та забезпечує створення стабільних та надійних програмних продуктів. Використання методики дозволяє досягти покращення якості програмного забезпечення за рахунок зниження кількості відмов та збільшення часу безвідмовної роботи. Дані аспекти підтверджують факт рекомендації удосконаленої методики роботи з МТВФ для використання в різних проектах розробки ПЗ з метою удосконалення процесу тестування та забезпечення якомога більшої тривалості успішної роботи програмних продуктів. Важливо лише пам'ятати, що для досягнення максимальної користі від покращеної методики роботи з МТВФ необхідно постійно аналізувати її результати та працювати над її подальшим удосконаленням.

4.4 Рекомендації щодо використання удосконаленої методики МТВФ для різних видів програмного забезпечення

Використання удосконаленої методики роботи з метрикою середнього часу між відмовами може бути здійснене в процесі тестування різнотипного програмного забезпечення. Для кожного виду ПЗ існують власні аспекти та ключові особливості використання методики.

В контексті вбудованих систем, де безпека та надійність відіграють ключову роль, використання удосконаленої методики МТВФ може стати важливою стратегією для покращення функціональності та забезпечення оптимального рівня стійкості до відмов. Знаходження та виправлення потенційних проблем в роботі вбудованих систем стає ефективнішим завдяки аналізу, який здатна надавати удосконалена методика.

Застосування удосконаленої методики МТВФ також може враховувати особливості вбудованих систем, такі як обмежені ресурси та особливості середовища. Адаптація методики до конкретних вимог та умов вбудованих систем грає важливу роль у забезпеченні їхньої ефективності та надійності.

Удосконалена методика МТВФ також може стати основою для розробки оптимальних стратегій тестування та підтримки вбудованих систем. Даний аспект включає в себе не лише виявлення відмов, але й розробку ефективних методів управління ризиками та реагування на можливі проблеми з метою забезпечення стабільності та безпеки вбудованих систем.

Використання покращеної методики МТВФ для веб-додатків та мобільних програм має важливе значення для забезпечення їхньої стабільності та продуктивності. Однією з ключових переваг цієї методики є спроможність адаптуватися до особливостей веб- та мобільних додатків, враховуючи різноманітність пристроїв та платформ, на яких вони працюють.

Удосконалена методика МТВФ враховує різні сценарії використання та роботу з різними пристроями, включаючи веб-браузери різних типів та версій, а також мобільні пристрої з різними операційними системами. Відповідний аспект дозволяє ефективно виявляти та вирішувати проблеми, які можуть виникнути в різних експлуатаційних умовах.

Використання удосконаленої методики МТВФ для корпоративних систем та бізнес-додатків визначається потребою забезпечення високої працездатності та якості обслуговування, щоб забезпечити ефективність та успіх бізнесу. Однією з ключових рекомендацій є акцент на зменшенні часу простою системи, оскільки це безпосередньо впливає на продуктивність та результат виконання різноманітних корпоративних процесів.

Удосконалена методика МТВФ сприяє покращенню доступності систем та додатків, забезпечуючи їхню стабільність та відмінну продуктивність. Дана особливість особливо є важливою для корпоративних клієнтів, які очікують надійного та безперебійного функціонування систем упродовж тривалого часу.

Подальшою рекомендацією є активна підтримка користувачів та впровадження швидких заходів у разі виявлення проблем. Використання удосконаленої методики МТВФ дозволяє оперативно реагувати на відмови та забезпечувати ефективні процеси підтримки, що позитивно впливає на задоволення користувачів та довіру до системи.

У системах з високим навантаженням, де кожна секунда та кожна операція має велике значення, покращена методика МТВФ може важливим інструментом для забезпечення високої продуктивності. Однією з головних рекомендацій є адаптація методики до конкретних потреб системи, враховуючи велику кількість оброблюваної інформації та високі навантаження на обчислювальні ресурси.

Застосування удосконаленої методики МТВФ для систем з високим навантаженням передбачає ретельний аналіз режимів роботи та завдань, які вони виконують. При цьому слід враховувати особливості обробки великих обсягів даних, оптимізацію алгоритмів та використання паралельних обчислень для забезпечення ефективності та швидкодії.

Рекомендації для використання удосконаленої методики МТВФ в контексті відкритого програмного забезпечення також спрямовані на створення умов для підтримки та підвищення надійності продукту, розробленого за участю спільноти. Перший аспект включає в себе систематичний аналіз результатів тестування та їх зворотний зв'язок зі спільнотою. Активний обмін інформацією про виявлені відмови, їх причини та шляхи виправлення сприяє швидкому вдосконаленню продукту. Удосконалена методика МТВФ може слугувати основою для ефективного аналізу даних та розробки стратегій управління якістю відкритого програмного забезпечення.

Інший аспект передбачає взаємодію зі спільнотою для реалізації нових функцій та виправлення виявлених проблем. Застосування удосконаленої методики МТВФ до відкритого програмного забезпечення дозволяє зрозуміти, які аспекти системи потребують уваги та покращення. Комунікація з користувачами та розробниками допомагає визначити пріоритети та ефективно впроваджувати зміни для покращення стабільності та надійності.

Згадані аспекти та рекомендації застосування удосконаленої методики роботи з МТВФ є важливими для використання в різних типах програмного забезпечення. Застосування методики може сприяти оптимізації процесу тестування та підвищенню якості та надійності різних програмних продуктів. У кожному конкретному випадку важливо враховувати унікальні особливості проекту та встановлені вимоги до програмного забезпечення для досягнення максимально ефективних результатів.

4.5 Перспективи подальшого розвитку та удосконалення методики роботи з метрикою МТВФ

Удосконалена методика роботи з метрикою середнього між відмовами здатна слугувати хорошим рішенням для забезпечення високої якості та надійності програмного забезпечення. Враховуючи всі фактори та ключові сторони роботи методики, важливо також здійснювати підтримку її потенціалу. Для цього можуть бути взяті до уваги різні перспективи подальшого розвитку та удосконалення методики МТВФ.

Поряд із наявними способами до покращення методики, що полягають в інтеграції з іншими методами та підходами оцінювання якості програмного забезпечення, розробці автоматизованих інструментів для роботи з метрикою та використанням штучного інтелекту та машинного навчання існують інші, які описують глобальні та загальні способи удосконалення. Адаптація методики роботи з метрикою МТВФ до специфічних вимог та особливостей різних галузей промисловості є однією з перших перспектив для подальшого розвитку та удосконалення цієї методології в тестуванні програмного забезпечення. Кожна галузь промисловості має свої унікальні особливості, технічні вимоги та критичні аспекти, які впливають на процеси тестування та надійність ПЗ.

Адаптація методики МТВФ до конкретної галузі дозволяє враховувати специфічні характеристики та вимоги цієї галузі, що в свою чергу сприяє більш

точному вимірюванню та оцінці надійності програмного забезпечення. Враховуючи вимоги конкретної галузі, можна розробити спеціалізовані тестові сценарії, які відобразатимуть реальні умови експлуатації програмного продукту. Даний аспект дозволяє забезпечити більш реалістичне та вичерпне тестування, що в свою чергу сприяє покращенню стабільності системи в конкретній сфері.

Підвищення рівня знань та компетенцій фахівців у сфері тестування програмного забезпечення також є важливою перспективою для подальшого розвитку методики МТВФ. Основною причиною є те, що компетентність та здатність виконання експертної оцінки спеціалістів безпосередньо впливають на якість проведення тестувань та аналізу надійності програмного продукту.

Збільшення рівня знань спеціалістів може включати в себе ознайомлення з сучасними тенденціями в галузі тестування, оволодіння новітніми інструментами та технологіями, а також розширення розуміння технічних та методологічних аспектів вимірювання надійності програмного забезпечення. Навчання спеціалістів необхідно орієнтувати не лише на технічні аспекти, але й на розвиток аналітичних та комунікаційних навичок. Комунікація між командами розробки та тестування важлива для правильного розуміння вимог, ефективної обміну інформацією та спільного досягнення мети – високоякісного програмного продукту з надійною та успішною роботою.

Розробка методів для врахування впливу апаратного забезпечення на МТВФ є ще однією з перспектив подальшого прогресу методики. Апаратне забезпечення впливає на надійність програмного продукту, і його роль у підвищенні або зменшенні середнього часу між відмовами є значущою.

Однією з можливих стратегій є інтеграція аналізу апаратного забезпечення в процес тестування. Дана діяльність включає в себе розгляд та оцінку функціонування різних видів апаратури, таких як пристрої введення-виведення, обчислювальні ресурси, системи охолодження тощо. Проведення тестів на апаратне забезпечення може надати цінні дані про його робочу стабільність та рівень витривалості.

Додатковою перспективою є розробка алгоритмів, які автоматично враховують характеристики апаратного забезпечення при розрахунках MTBF. Відповідний фактор дозволить здійснювати більш точні та реалістичні оцінки надійності програмного продукту, враховуючи специфіку апаратної інфраструктури, на якій він працює. Важливою частиною цього напрямку є також розробка стандартів та методологій, які нададуть загальноприйняті принципи врахування апаратного забезпечення у вимірюванні MTBF. Даний процес сприятиме стандартизації підходів та обміну даними між різними командами розробників та тестування.

Врахування впливу людського фактору на MTBF може також бути однією з перспектив подальшого розвитку та покращення методики роботи з метрикою MTBF. Людський фактор включає в себе не лише фахові навички та знання спеціалістів, але й їхнє сприйняття та взаємодію з програмним забезпеченням, що може суттєво впливати на його надійність та час між відмовами.

Одним із шляхів врахування людського фактору є включення аналізу взаємодії користувачів з програмою в процес тестування. Відповідна діяльність передбачає врахування різних сценаріїв використання, рівень навичок користувачів та їхню здатність ефективно взаємодіяти з програмним продуктом. Використання методів, що моделюють реальні умови користування, може допомогти виявити можливі проблеми, пов'язані з людським фактором.

Розробка інструментів для моніторингу та аналізу роботи користувачів може стати корисним елементом врахування людського фактору в MTBF. Такі інструменти можуть надавати дані про типові варіанти використання, помилки та труднощі, що виникають під час взаємодії, що дозволить пристосовувати до цього тестові ситуації та покращувати якість програмного забезпечення.

Різні перспективи удосконалення методики роботи з метрикою MTBF є важливими та значущими, оскільки завдання покращення способу розрахунку та підвищення часу безвідмовної роботи є загальним для кожної з них. Використання різних сценаріїв удосконалення з метою підвищення рівня функціонування метрики середнього часу між відмовами дозволить збільшити

ефективність виконання загальних процесів розробки та тестування будь-яких програмних систем. Досягнення бажаних результатів в даному напрямку також буде важливим кроком до забезпечення та постійної підтримки високого рівня якості, стабільності та надійності програмного забезпечення.

ВИСНОВКИ

Виконання кваліфікаційної роботи було здійснене в межах предметної області, що пов'язана з тестуванням програмного забезпечення та застосуванням метрики середнього часу між відмовами для розрахунку тривалості успішної роботи ПЗ. На початку виконання було виокремлено всі ключові аспекти роботи, які були проаналізовані та досліджені. Подальше виконання було здійснене відповідно до строго визначених етапів.

В першому розділі було виконано загальний аналіз предметної області з представленням інформації щодо тестування ПЗ та метрики МТВФ. Зокрема, було описано різні види тестування та їх особливості. Поряд з цим було представлено метрику середнього часу між відмовами програмного забезпечення як основний елемент дослідження, виокремлено ключові особливості її застосування та позитивні сторони. Також було визначено основні недоліки та обмеження використання метрики, що слугувало основою для формулювання постановки задачі для виконання подальших досліджень.

У другому розділі було виконано аналіз сучасних підходів до роботи з метрикою МТВФ та способів їх покращення. Першочергово було представлено опис наявних підходів, що включають використання статистичних розподілів та методів прогнозування. Далі було описано шляхи покращення метрики МТВФ, що полягають у застосуванні особливостей машинного навчання, різнотипного моделювання та розробці адаптивних методів оцінки надійності. На наступному кроці було здійснено опис потенційної роботи метрики МТВФ в інтеграції з іншими видами відповідних показників якості ПЗ. Після цього було виконано представлення різних способів автоматизації розрахунку показника часу безвідмовної роботи, що включає використання різних алгоритмів, моделей та додаткових обчислювальних інструментів. На останніх двох етапах було здійснено порівняльний аналіз вказаних сучасних підходів до покращення роботи з метрикою МТВФ та представлено загальні пропозиції щодо подальшого удосконалення на основі проведеного аналізу.

У третьому розділі було представлено особливості розробки удосконаленої методики роботи з метрикою МТВФ. Спочатку було визначено та описано вимоги до покращеної методики та представлено критерії оцінки її ефективності після розробки. На наступному кроці було виконано безпосередню реалізацію удосконаленої методики з послідовним описом основних її компонентів та складових структур. Далі було створено та визначено рекомендації щодо впровадження розробленої методики в процес тестування програмного забезпечення. На подальшому етапі відбувся вибір конкретного проекту для валідації удосконаленої методики та було представлено основні особливості та фактори, які стали причиною для вибору відповідного програмного проекту. На останньому етапі було виконано загальний опис процесу тестування обраного програмного проекту із застосуванням удосконаленої методики.

В четвертому розділі було здійснено оцінку застосування покращеної методики. Першочергово було представлено результати використання удосконаленої методики роботи з метрикою середнього часу між відмовами на прикладі даних тестування обраного проекту для валідації. Далі було виконано аналіз відповідних результатів та виокремлення ключових особливостей застосування методики. На наступному кроці була представлена кількісна та загальна оцінка ефективності застосування удосконаленого підходу. На останніх етапах було представлено рекомендації щодо застосування покращеної методики для різних видів програмного забезпечення та подальші перспективи розвитку та удосконалення методики роботи з метрикою середнього часу між відмовами.

Розроблена методика роботи з метрикою МТВФ може бути використана для впровадження в процес тестування різноманітного програмного забезпечення. Її застосування здатне полегшити процес розрахунку показника метрики, виявити критичні місця роботи ПЗ, а також на загальному рівні підвищити ефективність проведення тестування програмних систем та здійснення оцінки їх результатів. Правильне використання методики дозволить контролювати якість ПЗ та приймати важливі рішення для забезпечення та підтримки високого рівня надійності програмних продуктів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Дипломна робота магістра: методичні вказівки щодо її виконання для студентів спеціальності «Інженерія програмного забезпечення» / Г. І. Радельчук. – Хмельницький : ХНУ, 2017. – 80 с.
2. Birolini A. Reliability Engineering: Theory and Practice. Heidelberg: Springer Berlin, 2013. 626 p.
3. Mili A., Tchier F. Software Testing: Concepts and Operations. New Jersey: Wiley, 2015. 400 p.
4. Dingsøyr T., Dybå T., Moe N. B. Agile Software Development: Current Research and Future Directions. Heidelberg: Springer Berlin, 2014. 238 p.
5. Langford G. O. Engineering Systems Integration: Theory, Metrics, and Methods. Boca Raton: CRC Press, 2016. 406 p.
6. Lienig J., Bruemmer H. Fundamentals of Electronic Systems Design. *Reliability Analysis*. Edinburgh, 2017. P. 45-73.
7. Fenton N., Bieman J. Software Metrics: A Rigorous and Practical Approach, 2nd ed. Boca Raton: CRC Press, 2020. 618 p.
8. Lewis W. E. Software Testing and Continuous Quality Improvement, 3rd ed. Boca Raton: Auerbach Publications, 2008. 684 p.
9. Kumar S., Rathore S. S. Software Fault Prediction: A Road Map (SpringerBriefs in Computer Science). Singapore: Springer, 2018. 72 p.
10. Schieferdecker I., Hoffmann A. Model-Based Testing. New Jersey: Wiley, 2010. 89 p.
11. Spetzler C., Winter H., Meyer J. Decision Quality: Value Creation from Better Business Decisions. New Jersey: Wiley, 2016. 252 p.
12. Yamada S. Software Reliability: Modeling Fundamentals and Applications. Berlin: Springer Science & Business Media, 2013. 90 p.
13. Tech S. Quality Assurance: Software Quality Assurance Made Easy. Charleston: CreateSpace Independent Publishing Platform, 2016. 50 p.
14. Dekker S. Drift into Failure. London: Routledge, 2011. 234 p.
15. Forbes C., Evans M., Hastings N., Peacock B. Statistical Distributions 4th ed. New Jersey: Wiley, 2011. 230 p.

16. Lenz A. R., Pozo A., Vergilio S. R. Linking software testing results with a machine learning approach. *Engineering Applications of Artificial Intelligence*. Amsterdam, 2013. № 5-6. P. 1631-1640.
17. Uzun B., Tekinerdogan B. Model-driven architecture based testing: A systematic literature review. *Information and Software Technology*. Amsterdam, 2018. № 102. P. 30-48.
18. Fraser G., Arcuri A. Sound empirical evidence in software testing. Proceedings - 34th International Conference on Software Engineering. New York, 2012. P. 178-188.
19. Basu A. Software quality assurance, testing and metrics. Hoboken: Prentice Hall, 2015. 301 p.
20. Kharb L. Reliable Software Development with Proposed Quality Oriented Software Testing Metrics. *International Journal of Computer Technology and Applications*. New Delhi, 2011. P. 798-802.
21. Oliver C. I. Fundamentals of Applied Probability and Random Processes, 2nd ed. Cambridge: Academic Press, 2014. 434 p.
22. Kroese D. P., Brereton T., Taimre T., Botev Z. I. Wiley Interdisciplinary Reviews: Computational Statistics. *Why the Monte Carlo method is so important today*. New Jersey: Wiley, 2014. №6. P. 386–392.
23. Dobesova Z. Programming language Python for data processing. *Electrical and Control Engineering: materials of the international conference*. Yichang, 2011. P. 4866–4869.
24. Johansson R. Numerical Python: A Practical Techniques Approach for Industry. California: Apress Berkeley, 2015. 487 p.
25. Chen D. Y. Pandas for Everyone: Python Data Analysis. Boston: Addison-Wesley Professional, 2017. 406 p.
26. Yim A., Chung C., Yu A. Matplotlib for Python Developers: Effective techniques for data visualization with Python, 2nd ed. Birmingham: Packt Publishing, 2018. 300 p.
27. Toomey D. Learning Jupyter. Birmingham: Packt Publishing, 2016. 238 p.

ДОДАТОК А
(обов'язковий)

КОПІЇ НАУКОВИХ ПУБЛІКАЦІЙ

<https://doi.org/10.31891/2219-9365-2023-75-14>

УДК 004.052.32

СВЕРБА Артем

Хмельницький національний університет
gloriks273@gmail.com

МАРТИНЮК Валерій

Хмельницький національний університет
<https://orcid.org/0000-0001-5758-4244>
e-mail: martynyuk.valeriy@gmail.com

УДОСКОНАЛЕННЯ МЕТОДУ РОБОТИ З МЕТРИКОЮ СЕРЕДНЬОГО ЧАСУ МІЖ ВІДМОВАМИ ДЛЯ ЗАБЕЗПЕЧЕННЯ ЕФЕКТИВНОГО ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Метрика середнього часу між відмовами (MTBF) широко використовується на етапі тестування програмного забезпечення для оцінки надійності та стабільності його роботи. Проте традиційні методи розрахунку MTBF мають певні недоліки, що знижують ефективність оцінювання результатів тестування. В статті запропоновано шляхи удосконалення методу роботи з метрикою MTBF, серед яких застосування статистичних методів, машинного навчання, моделювання програмного забезпечення, інтеграція з іншими метриками, автоматизація розрахунків та моніторинг в реальному часі. Ці підходи дозволять підвищити точність розрахунку MTBF, врахувати взаємозв'язки між компонентами системи, оперативно реагувати на зміни показників надійності. Впровадження запропонованих удосконалень методу роботи з MTBF сприятиме ефективнішому оцінюванню результатів тестування та підвищенню загальної надійності програмного забезпечення.

Ключові слова: Метрика, MTBF, тестування програмного забезпечення, надійність, моделювання, інтеграція метрик, автоматизація, моніторинг в реальному часі.

SVERBA Artem, MARTYNYUK Valeriy
Khmelnitskyi National University

IMPROVING THE MEAN TIME BETWEEN FAILURE METRIC FOR ENSURING EFFECTIVE EVALUATION OF SOFTWARE TESTING RESULTS

The mean time between failures (MTBF) metric is widely used in software testing to assess reliability and stability. However, traditional methods for calculating MTBF have certain drawbacks that reduce the effectiveness of evaluating testing results. The paper proposes ways to improve the MTBF metric method, including applying statistical methods, machine learning, software modeling, integrating with other metrics, automating calculations, and real-time monitoring. These approaches will increase the accuracy of MTBF calculation, account for interconnections between system components, and promptly respond to changes in reliability indicators. Implementing the proposed improvements to the MTBF method will promote more effective evaluation of testing results and increase overall software reliability.

Keywords: Metric, MTBF, software testing, reliability, modeling, metric integration, automation, real-time monitoring.

Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями

Використання метрики середнього часу між відмовами є важливим рішенням на етапі тестування програмного забезпечення, оскільки вона надає інформацію про стабільність роботи системи та її надійність. Існуючі способи роботи з метрикою MTBF дозволяють використовувати її для обчислення основного показника, проте такий процес виконується за допомогою більш традиційного та загального підходу. Саме тому існує потреба удосконалення методу роботи з даною метрикою для покращення її розрахунку та результатів тестування програмного продукту в цілому.

Аналіз досліджень та публікацій

Основою для дослідження є праці різних науковців, зокрема Ф. Нормана, А. Біроліні, Х. Брюммера, Є. Лініга та інших, а також загальні публікації про етап тестування програмного забезпечення, метрики перевірки роботи ПЗ, включаючи методи роботи з метрикою середнього часу між відмовами (MTBF), її особливості та важливі сторони її використання. Наявні публікації та дослідження відповідно до тематики містять опис та ключові моменти вже наявних підходів до покращення методу роботи з метрикою середнього часу між відмовами. Дослідники виділяють різноманітні покращення, серед яких є такі: використання різних статистичних методів для точнішого прогнозу надійності роботи програмного забезпечення, застосування методів машинного навчання для якіснішого знаходження та представлення результатів процесу оцінки надійності, застосування різних архітектурних моделей програмних систем для деталізації процесу взаємодії різних компонентів системи та їх вплив на надійність загальної роботи ПЗ, використання емпіричних даних про відмови в роботі програми та їх послідовний моніторинг для встановлення певних закономірностей у разі зупинки роботи ПЗ. Аналіз останніх публікацій та досліджень є

дуже важливим, оскільки основні їх аспекти будуть слугувати для розробки удосконаленої методики роботи з метрикою середнього часу між відмовами.

Формулювання цілей статті

Головною метою статті є опис метрики середнього часу між відмовами (MTBF), її методів розрахунку, переваг та недоліків, а також представлення способів удосконалення методу роботи з даною метрикою, опис їх ключових нюансів та напрямків для покращення.

Виклад основного матеріалу

Тестування є одним з важливих етапів розробки програмних систем. Тестування програмного забезпечення — це процес перевірки ПЗ на наявність помилок та багів, а також перевірка відповідності реальних та очікуваних результатів роботи програми. Тестування є фундаментальним процесом створення надійних і придатних для використання програмних продуктів. Вся важливість даного кроку підкреслюється загальною оцінкою розробленої програми про її працездатність та відповідність заявленим вимогам та прийняттям рішення про її випуск для цільового користувача або доопрацювання її компонентів з врахуванням усіх встановлених дефектів [1].

Існує велика кількість видів тестування, кожен з яких визначається об'єктом тестування, власним набором інструментів та алгоритмом їх застосування для перевірки компонентів програмного продукту або його роботи в цілому. Результат виконання даного етапу розробки програмного забезпечення може бути різним в залежності від того, як працює програмний продукт та його структурні елементи. Для оцінки отриманих результатів тестування застосовуються метрики.

Метрика тестування — це показник тестування програмного забезпечення, що визначається як кількісна ознака, яка допомагає оцінити хід і якість процесу тестування програмного забезпечення [2]. Застосування метрики включає в себе кроки, що допомагають дати оцінку результатам тестування, провести їх аналіз та дійти висновку щодо подальших дій в межах розробки ПЗ.

В загальному метрики тестування є дуже важливими для етапу тестування, оскільки вони визначають критерії відповідного процесу та допомагають чітко окреслити його межі. Також відповідні метрики допомагають прийняти рішення щодо подальших кроків розробки, оскільки вони полегшують загальну процедуру прийняття рішення та вказують на конкретні компоненти програмної системи, які повинні бути доопрацьовані в разі необхідності.

В процесі виконання тестування програмного забезпечення може бути застосована велика кількість різних метрик, які обираються в залежності від поставленої мети тестування. Однією з таких метрик є метрика середнього часу між відмовами (MTBF). Даний кількісний показник відноситься до типу метрик, що стосуються успішного часу виконання програми та її локальних збоїв, що називаються відмовами.

MTBF (Mean Time Between Failures) — це метрика, яка вимірює очікувану середню тривалість безвідмовної роботи програмного забезпечення до наступної відмови [3]. Основним показником метрики є значення, що вказує на те, як довго програмна система може працювати без збоїв. Схематичне зображення метрики середнього часу між відмовами подане на Рисунку 1.

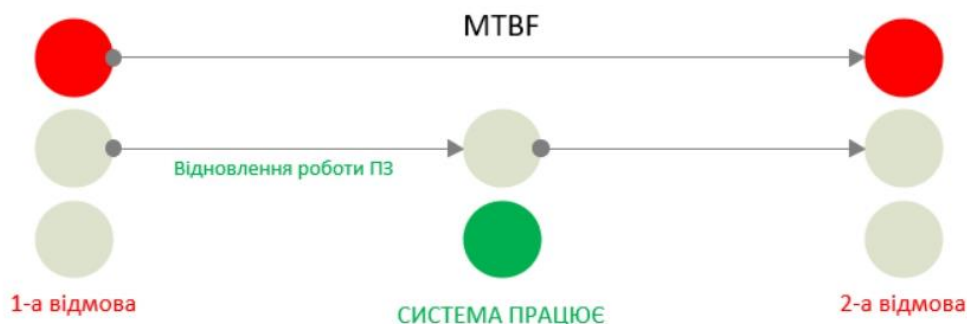


Рис. 1. Метрика середнього часу між відмовами

Середній час між відмовами є важливим показником процесу тестування ПЗ для вимірювання продуктивності, безпеки та працездатності. Також дана ознака використовується для визначення надійності розробленої програмної системи, що безпосередньо пов'язано з успішною та коректною роботою програми. Ще одним важливим аспектом метрики середнього часу між відмовами є її можливість надавати декілька типів оцінок: оцінку продуктивності, безпеки та архітектури ПЗ.

Метрика MTBF спирається на показник безвідмовної роботи програмної системи. Існує декілька методів розрахунку показника даної метрики. Перший з них — історичний. Даний метод є найбільш традиційним та використовуваним для обчислення значення показника. Принцип його роботи є дуже

простим, оскільки для розрахунку необхідна лише загальна інформація про роботу програмної системи. Загальна формула розрахунку середнього часу між відмовами за історичним методом є такою:

$$MTBF = \frac{\text{Загальна кількість робочих годин}}{\text{Загальна кількість відмов}}$$

Основними компонентами формули є показники загальної кількості робочих годин та кількості відмов. Перший числовий показник вказує на кількість годин, коли програмна система працювала успішно та безперебійно. При визначенні фактичного значення даного показника не враховується час виконання відладки ПЗ та приведення його до робочого стану повторно. Значення загальної кількості відмов вказує на кількість помилкових збоїв в роботі програмного забезпечення за певний проміжок часу [4]. Результуючий показник MTBF представляє кінцеве значення проведення вимірювання метрикою середнього часу між відмовами.

Наступним методом розрахунку показника даної метрики є метод на основі статистичних розподілів. Він використовується для оцінки середнього часу між відмовами системи з використанням певного статистичного розподілу. Цей підхід дозволяє більш точно моделювати час між відмовами та прогнозувати MTBF на основі характеристик та властивостей цього розподілу [5]. Для того, аби почати розрахунок за даним методом необхідно спочатку обрати статистичний розподіл, що найбільше підходить для моделювання часу між відмовами програмного забезпечення. Його вибір може базуватися на припущеннях про природу відмов системи або аналізі даних попередньо вказаного історичного методу. Після вибору розподілу потрібно провести оцінку параметрів вибраного статистичного розподілу на основі наявних даних про відмови. Оцінка параметрів дозволяє налаштувати вибраний розподіл таким чином, щоб він якнайкраще відповідав спостережуваним даним. На наступному етапі роботи методу розраховується саме значення метрики MTBF. Воно може бути розраховане за допомогою інструментів обраного статистичного розподілу або за іншими формулами, зокрема формулою історичного методу. Останнім кроком роботи методу є перевірка того, наскільки добре розподіл відповідає спостережуваним даним про відмови. Результат перевірки буде вказувати на адекватність моделі, яка може бути використана для моделювання та прогнозування часу між відмовами в майбутньому.

Ще одним методом розрахунку показника метрики MTBF є метод на основі тестування. Даний метод застосовується для оцінки надійності системи шляхом проведення спеціальних тестів з метою виявлення відмов. Основний принцип його роботи базується на випробуванні системи під реалістичними умовами та визначення часу між відмовами [6]. Для початку роботи з цим методом потрібно визначити конкретні тестові сценарії, що включають навантаження та умови, які найбільш відповідають реальному використанню системи. Далі необхідно виконати безпосередньо процес тестування, визначаючи при цьому час відмови для кожного тесту. На наступному кроці потрібно використати формулу історичного методу для розрахунку загального показника MTBF. Останній етап включає аналіз отриманих результатів тестування, враховуючи MTBF та інші показники надійності, що дозволить виявити слабкі місця системи та визначити можливі шляхи поліпшення надійності ПЗ.

Метод розрахунку MTBF на основі надійності є ще одним способом розрахунку показника метрики. Його можна використовувати для оцінки середнього часу між відмовами системи на основі аналізу надійності її компонентів. Даний метод передбачає збір даних про відмови компонентів системи та використання різних аналітичних технік для оцінки MTBF. Для роботи даного методу спочатку потрібно зібрати дані про відмови програмного забезпечення. Така інформація може бути зібрана будь-яким з доступних способів. Головними даними виступають час безвідмовної роботи та час відмов. Після цього необхідно виконати аналіз надійності системи та її компонентів на основі зібраних інформації. Наступним кроком є розрахунок показника MTBF шляхом комбінування надійності компонентів, використовуючи формули або аналітичні моделі. На останньому кроці відбувається перевірка отриманих результатів, порівнюючи їх з фактичними даними про відмови системи. Перевірка може включати порівняння прогнозованого MTBF з фактичними спостереженнями або порівняння прогнозу надійності компонентів з реальним значенням даного показника.

Метод експертних оцінок розрахунку MTBF є заключним серед інших та використовується для оцінки середнього часу між відмовами системи на основі експертних знань та досвіду фахівців. Особливістю даного методу є те, що він застосовується тоді, коли недостатньо даних про відмови або коли система є новою і немає історичних даних. Принцип його роботи заснований на суб'єктивному сприйнятті роботи системи. Спочатку відбувається формування експертної групи, що володіє необхідною інформацією про роботу програмного забезпечення. Далі відбувається розгляд різних аспектів системи, що можуть впливати на її надійність, та визначаються параметри, які можуть бути використані для оцінки надійності. На наступному кроці група експертів здійснює оцінювання середнього часу між відмовами програмної системи, що може бути представлено числовими значеннями або діапазонами. На останніх двох кроках

відбувається аналіз встановлених оцінок часу та здійснюється їх перевірка за допомогою додаткових методів, таких як порівняння з існуючими даними або використання інших незалежних експертних оцінок.

Враховуючи особливості метрики MTBF, існують як позитивні, так і негативні її сторони. До переваг даної метрики відносять:

1. Простота вимірювання. MTBF обчислюється за допомогою легких розрахунків, що робить його досить простим і зрозумілим для вимірювання і порівняння надійності різних систем або компонентів;
2. Показник надійності. Метрика може слугувати показником надійності системи. Високе значення MTBF вказує на довгий період безвідмовної роботи системи, що відображає високий рівень надійності.
3. Прогнозування терміну роботи без відмов. Метрика MTBF може бути використаний для прогнозування тривалості роботи системи, що допомагає встановити терміни технічного обслуговування для запобігання збоїв.

Також дана метрика може має свої негативні сторони:

1. Не враховує ремонтпридатність. MTBF не враховує можливість відновлення системи після виникнення відмови, а вимірює лише тривалість безвідмовної роботи і не дає повної картини про загальну доступність системи.
2. Не враховує серйозність відмови. Метрика не дає інформації про серйозність або вплив відмови на функціонування системи, оскільки просто вказує на середню тривалість між відмовами, не розмежовуючи відмови за їх важкістю або наслідками.
3. Не враховує випадковість відмов. MTBF передбачає регулярний розподіл відмов, що не завжди відповідає реальності. Фактичні відмови можуть мати випадковий характер, і MTBF не враховує таку випадковість.
4. Непридатність для деяких видів програмного забезпечення. Метод MTBF може бути менш придатним для оцінки надійності програмного забезпечення, що характеризується високою складністю роботи, динамічною поведінкою системи або взаємодією зі значною кількістю зовнішніх компонентів. У таких випадках, оцінка надійності на основі MTBF може не враховувати вплив різних факторів, що можуть спричинити відмову.

Позитивні сторони метрики MTBF свідчать про те, що правильне використання метрики середнього часу між відмовами дає можливість налаштувати програмний продукт на довгу та безперебійну роботу, мінімізуючи затрати та ресурси для вирішення помилкових ситуацій [7]. Проте сукупність недоліків спонукає до пошуку кращого способу використання метрики середнього часу між відмовами для кращого оцінювання результатів тестування програмного забезпечення.

Для покращення методу MTBF та його ефективності оцінювання результатів тестування програмного забезпечення можуть бути використані різні сучасні підходи та способи. Зокрема, для покращення методу роботи метрики можуть бути використані як додаткові методи математичних розрахунків, так і застосування суміжних галузей розробки програмного забезпечення [8].

Одним з перших підходів, що може бути використаний для покращення роботи з метрикою MTBF, є використання статистичних методів. Застосування даного підходу може покращити результати розрахунків показників метрики та підвищити їх точність. Першим кроком до удосконалення в межах даного підходу може бути збір і аналіз даних про відмови з використанням статистичного аналізу. Такий тип аналізу даних про відмови може допомогти виявити закономірності та тренди у часі. Збір даних про відмови можна здійснювати через систему журналів відмов або вимірювання проміжків часу між ними для виявлення факторів, що впливають на значення MTBF.

Використання статистичних методів може покращити роботу методу метрики середнього часу за допомогою застосування різних типів розподілів. Розподіли можуть бути корисними для моделювання відмов і розрахунку MTBF. Вибір правильного розподілу може бути важливим, оскільки різні розподіли можуть краще відповідати даним про відмови в конкретній системі.

Статистичні методи аналізу надійності можуть бути використані для відповідної оцінки загальної системи та її компонентів. Аналіз такого типу дозволяє виявити вразливі місця програмного забезпечення та ідентифікувати фактори, що сприяють відмовам [9]. Отримуючи результати аналізу надійності, можна приймати заходи для поліпшення MTBF шляхом зміни побудови системи або часткової заміни її основних структурних елементів.

Ще одним зі способів покращення є використання статистичних методів прогнозування. Спираючись на передбачення результатів, можна дістати інформацію про майбутні значення MTBF на основі уже наявних історичних даних. Відповідне прогнозування дозволяє передбачати тенденції та визначати майбутні необхідні ремонтні роботи або профілактику для збереження вимог щодо надійності системи.

Важливими для покращення MTBF також є методи випробування та валідації, що мають статистичний аспект. Вони можуть бути використані для оцінки надійності системи ще на етапі розробки програмного забезпечення, що включає проведення статистичних експериментів, випробувань з

навантаженням та аналізу отриманих даних. Результати цих методів можуть бути застосовані для виявлення дефектів, впливу параметрів на надійність та для покращення MTBF перед введенням ПЗ в експлуатацію.

В загальному використанні методів статистики дозволяє більш об'єктивно оцінювати роботу метрики середнього часу між відмовами та здійснювати її покращення. Виявлення факторів, що впливають на надійність системи, та розробка стратегії для поліпшення надійності та безвідмовної роботи є одним з головних завдань даних методів.

Інший загальний підхід, що може використовуватись для покращення методу роботи з метрикою MTBF, є застосування машинного навчання. Воно може допомогти виявляти складні залежності між вхідними параметрами та середнім часом між відмовами, а також враховувати нелінійні зв'язки між компонентами ПЗ [10].

На шляху до використання моделей машинного навчання важливою складовою є збір даних. Достовірна інформація про характеристики ПЗ та різні умови використання стане основою для тренування моделей машинного навчання. Відповідні моделі будуть застосовуватись для розв'язання задач прогнозування часу успішної роботи та часу відмов ПЗ. Вибір таких моделей залежить від даних про систему та конкретно поставленої задачі. Також важливим аспектом є вибір правильних функцій моделі, яка буде оперувати в межах програми, що проходить тестування. Тренування моделей на зібраних даних з використанням різних алгоритмів здатне шукати залежності між вхідними параметрами та MTBF, здійснюючи адаптацію своїх ваг та параметрів для найкращої прогнозовної точності. Важливим моментом після тренування є валідація отриманих результатів на незалежних даних, щоб оцінити точність прогнозу та здатність узагальнення. Для покращення самих моделей також можна застосовувати різні варіанти їх оптимізації, що в кінцевому результаті призведе до покращення отриманих вихідних показників метрики MTBF.

На загальному рівні, застосування машинного навчання для прогнозування MTBF може удосконалити точність прогнозів та допомогти виявити складні відносини та залежності між компонентами, які складно виявити за допомогою традиційних аналітичних методів.

Важливим рішенням для покращення методу роботи MTBF є ще один підхід, який базується на моделюванні програмного забезпечення на основі архітектури. Даний підхід є корисним, оскільки вхідними даними для виконання методу MTBF є добре вивчена та налаштована модель програмної системи [11]. Застосування моделей програмного забезпечення, заснованих на архітектурі та особливостях окремої системи, може допомогти краще враховувати взаємодію між компонентами програмного забезпечення та їх вплив на надійність. Також одним з кроків моделювання може бути створення моделей відмов. Згідно цього раніше побудована модель архітектури програмного забезпечення буде взаємодіяти з моделлю відмов, що дозволить точніше спрогнозувати час успішної роботи програмної системи та час можливих її збоїв.

Одним зі способів покращення методу MTBF також може бути використання емпіричних даних. Послідовний збір таких даних дозволить виконувати постійний їх моніторинг та аналіз, що матиме позитивний вплив на прогнозування середнього часу між відмовами. Безпосередньо під час процесу передбачення буде відбуватись обробка зібраних даних, їх порівняння з відомими паттернами, що дозволить виявити природу різних збоїв та повторюваних дефектів ПЗ. В загальному емпіричні дані грають ключову роль у виявленні проблем, покращенні надійності та збільшенні MTBF програмного забезпечення.

Для поліпшення методу роботи з метрикою середнього часу між відмовами можлива також розробка адаптивних методів оцінки надійності програмних систем. Вони дозволяють враховувати зміну умов та концепт роботи програмного забезпечення для більш точної оцінки його надійності [12]. В межах розробки таких методів можуть використовуватись підходи машинного навчання або моделювання програм, оскільки важливим є визначення факторів впливу на працездатність ПЗ та створення відповідних моделей прогнозування. Адаптивні методи оцінки надійності програмного забезпечення можуть автоматично коригувати параметри методу MTBF на основі отриманих даних про відмови, що дозволить більш точно прогнозувати надійність в реальному часі.

Всі розглянуті підходи до покращення методу роботи MTBF є досить важливими, оскільки кожен з них має свої особливості та здатен вносити позитивні якості в роботу методу, що може покращити точність розрахунку основного показника MTBF та надійність роботи програмного забезпечення в цілому. Проте також суттєвим кроком є пошук нових способів удосконалення методу роботи з метрикою середнього часу між відмовами, що можуть широко застосовуватись в майбутньому на етапі тестування програмних систем.

Одним з перших таких способів удосконалення роботи з MTBF є розвиток нових моделей та алгоритмів для розрахунку показника середнього часу між відмовами. Прогрес в дослідженні нових структур може внести суттєвий вклад у покращення методу MTBF та оцінки надійності програмного забезпечення. Перший напрямок для розвитку може бути пов'язаним зі створенням моделей та алгоритмів, що спираються на комплексні залежності між компонентами програмного забезпечення. Будь-яка програма може бути представлена у вигляді системи, взаємозалежні компоненти якої впливають один на одного, що має безпосередній ефект при визначенні відмов та збоїв ПЗ. В такому випадку також можливе застосування

уже готових або новостворених методів статистики, що можуть дати точну оцінку надійності ПЗ на основі емпіричних даних та різних впливів між його компонентами.

Ще одним напрямком в розвитку нових структурних складових є покращене моделювання. Його основна особливість в тому, що застосування основних його принципів може бути здійснене протягом всіх етапів розробки програмного забезпечення. Оцінка MTBF повинна застосовуватись ще на ранніх етапах створення ПЗ, що допоможе виявляти більшу кількість вразливих місць ПЗ та вирішувати проблеми потенційних відмов в майбутній його роботі. Також рання оцінка та створення моделей на перших етапах дасть можливість програмі бути стійкішою до змін різнотипних умов її роботи, що неабияк підвищить її надійність.

Іншим перспективним напрямком розвитку є використання штучного інтелекту. Для прогнозування показника MTBF доступне використання різних нейронних мереж, моделей машинного та глибокого навчання, що здатні обробляти великі обсяги даних про ПЗ та на основі них здійснювати оцінювання роботи програми, зокрема надійності та здатності до якомога тривалішої безперебійної діяльності програмного продукту. Штучний інтелект також допоможе при вирішенні складних та незрозумілих ситуацій, які стосуються залежностей між компонентами ПЗ, що безпосередньо впливають на показник середнього часу між відмовами.

В загальному розвиток нових моделей та алгоритмів для розрахунку MTBF є дуже важливим, оскільки існує необхідність постійно оновлювати методи та враховувати специфічні особливості програмного забезпечення, щоб досягти більш точних оцінок надійності та покращення MTBF.

Інтеграція з іншими метриками тестування є ще одним новим способом удосконалення методу роботи з метрикою середнього часу між відмовами. Комбінація різних метрик дозволить отримати більш повне уявлення про надійність системи та виявити проблеми, які можуть бути пропущені, якщо розглядати лише MTBF. Різні метрики спираються на різні показники та по-своєму дають оцінку в конкретній ситуації, що в спільній роботі з MTBF може слугувати надійним інструментом оцінювання надійності та всієї роботи програми.

Першою для комбінації з MTBF може слугувати метрика MTTF. MTTF – це метрика середнього часу до першої відмови програмного забезпечення. Інтеграція MTTF дозволить отримати більш широке представлення надійності системи. Також можна використовувати MTTF з MTBF для визначення того, які частини життєвого циклу системи перебувають в стані стабільної роботи перед першою відмовою та на якій з них відбувається збій роботи ПЗ.

Другою для інтеграції з MTBF є SER-метрика. SER – це метрика, яка дозволяє визначити кількість помилок, виявлених у програмному забезпеченні протягом певного періоду часу. Інтеграція SER з MTBF дасть можливість оцінити вплив помилок на надійність системи. В загальному випадку основним є твердження: чим нижче SER, тим більш надійне програмне забезпечення, що сприяє збільшенню MTBF.

Третя метрика, що може працювати в комбінації з MTBF, є TTF. TTF – це метрика, що визначає час до відмови окремого компонента або модуля системи. Інтеграція TTF з MTBF дозволить визначити, які компоненти чи модулі мають більшу ймовірність відмови та впливають на загальний MTBF системи. Застосування даної метрики допоможе зосередити увагу на вразливих або критичних елементах системи для подальшого покращення надійності.

Четвертою метрикою для інтеграції з MTBF може виступати показник MTTR. MTTR – це метрика середнього часу відновлення ПЗ після відмови. Комбінація MTTR та MTBF дозволить оцінити, наскільки швидко система може бути відновлена після збою та перейти в робочий режим. Низький MTTR сприяє скороченню часу простою системи та підвищенню загальної надійності.

Метрика MTBF також може поєднуватись з динамічними метриками для покращення вимірювання різних показників. Застосування динамічних метрик тестування, таких як Code Coverage (покриття коду), Fault Detection Rate (швидкість виявлення помилок) або Failure Rate (частота відмов), може допомогти в оцінці надійності програмного забезпечення та впливати на метод MTBF. Інтеграція цих метрик з MTBF дозволить зосередитися на ефективності тестування та виявленні помилок, що сприяє покращенню збільшенню середнього часу між відмовами.

В підсумку інтеграція MTBF з іншими метриками тестування дозволить більш широко оцінювати працездатність системи в цілому та окремих її компонентів, а також дасть можливість спрямовувати більше зусиль на покращення конкретних аспектів, що здатні впливати на MTBF.

Ще одним оновленим способом удосконалення методу роботи MTBF може слугувати впровадження автоматизації розрахунку показника метрики. Таке впровадження може суттєво полегшити процес обчислення, удосконалити метрику середнього часу між відмовами ПЗ та підвищити його надійність. Використання спеціалізованих програмних інструментів є першим напрямком даного способу. Відповідно до нього для розрахунку показника середнього часу між відмовами можуть бути використані програмні інструменти, які на основі введених або емпіричних даних будуть швидко та точно надавати значення MTBF для програмної системи. Також для розрахунків може бути розроблений власний програмний

продукт або звичайний скрипт, який буде задовольняти потребу пошуку значення MTBF для конкретного програмного забезпечення.

За іншим напрямком удосконалення методу роботи MTBF може бути здійснене за допомогою інтеграції з системами моніторингу та журналювання. Впровадження таких систем допоможе на автоматичних засадах збирати інформацію про відмови та інші необхідні параметри програмного забезпечення. Завдяки цьому можливе використання цих даних безпосередньо для розрахунку MTBF без необхідності ручного введення, що сприятиме збільшенню точності обчислень та заощадження зусиль розробників ПЗ. Важливим аспектом в автоматичному зборі інформації є налаштування періодичності такого процесу. Для підтримки актуальності даних про систему та показник MTBF необхідно встановити регулярне оновлення відповідної інформації. Воно допоможе оперативно отримувати наявну інформацію про відмови, якщо вони сталися, а також здійснювати послідовне обчислення середнього часу між відмовами.

Автоматизація розрахунку MTBF допоможе зменшити ризик помилок, полегшить процес оцінки надійності та дозволить ефективно використовувати час та ресурси. Вона сприятиме покращенню методу MTBF та дозволить швидко отримувати оновлені результати розрахунків для прийняття відповідних рішень.

Останнім способом для удосконалення методу роботи з метрикою середнього часу між відмовами може виступати розробка методів та інструментів для моніторингу MTBF в реальному часі. В поєднанні з попереднім способом такий процес буде мати абсолютно новий підхід до збору інформації про відмови та загальну роботу програмної системи, оскільки вона буде отримуватись автоматичним шляхом в режимі реального часу, що допоможе своєчасно реагувати на будь-які зміни та постійно відстежувати показник MTBF. Також своєчасний збір інформації допоможе швидко виконувати процес її аналізу для прийняття різних рішень, що стосуються роботи ПЗ. Для кращого аналітичного процесу в режимі реального часу може бути розроблено шаблон звітності або здійснено візуалізацію отриманої інформації. Враховуючи автоматичний розрахунок MTBF та постійний потік інформації про відмови, необхідним елементом буде зв'язок з іншими системами попередження та автоматичного сповіщення розробника про зміну умов працездатності програмного забезпечення. Такі додаткові компоненти разом з іншими особливостями способу удосконалення допоможуть забезпечити актуальну інформацію про надійність програмного забезпечення та швидко відповідати на зміни, що в свою чергу буде сприяти покращенню методу MTBF та надійності системи.

Кожен з наведених способів удосконалення покликаний виконати головне завдання – здійснити удосконалення методу роботи з MTBF. З урахуванням ключових особливостей всі з них можуть бути використані для створення умов покращення параметрів метрики середнього часу, що на загальному рівні призведе до покращення результатів всього процесу тестування ПЗ, підвищення показника середнього часу безвідмовної роботи та збільшення надійності будь-якої програмної системи.

Висновки з даного дослідження і перспективи подальших розвідок у даному напрямі

Удосконалення методу роботи з метрикою середнього часу між відмовами є важливим завданням для підвищення ефективності оцінювання результатів тестування програмного забезпечення. Традиційні підходи до розрахунку MTBF мають певні обмеження, що не дозволяють повною мірою врахувати всі фактори, які впливають на надійність системи. Запропоновані в статті способи удосконалення методу, такі як використання статистичних методів, машинного навчання, моделювання ПЗ, інтеграції з іншими метриками, автоматизації та моніторингу в реальному часі, дозволять подолати ці обмеження. Застосування даних підходів надасть можливість точніше розраховувати показник MTBF з урахуванням усіх взаємозв'язків між компонентами системи, швидше реагувати на зміни показників надійності, ефективніше оцінювати результати тестування в цілому.

Отже, впровадження запропонованих у статті способів удосконалення методу роботи з метрикою MTBF дозволить підвищити якість оцінювання результатів тестування програмного забезпечення та забезпечити створення більш надійних програмних систем. Подальші дослідження можуть бути спрямовані на практичну реалізацію та валідацію ефективності запропонованих підходів.

Література

1. В.Е. Льюїс, Тестування програмного забезпечення та постійне покращення якості (3-є видання), 2016.
2. Ф. Норман, Метрики програмного забезпечення: суворий і практичний підхід (3-тє видання), 2014.
3. С. Лініг, Х. Брюммер, «Аналіз надійності», Основи проектування електронних систем, 2017.
4. А. Біроліні: Розробка надійності: теорія та практика, Берлін 2013.
5. І. Шифердеккер, А. Гофман, Тестування на основі моделі, 2010.
6. Дж. Бернанд, А. Прайс, Тестування програмного забезпечення: поняття та операції, 2016.
7. С. Кумар, С. Ратор, Прогноз програмних збоїв: дорожня карта, 2017.

8. Т. Дінгсойр, Т. Дйба, Н. Бреде, Гнучка розробка програмного забезпечення: поточні дослідження та майбутні напрямки, 2018.
9. Дж. О'Ріган, Практичний підхід до якості програмного забезпечення, 2014.
10. Ш. Ямада, Моделювання надійності програмного забезпечення: Основи та застосування, 2014.
11. П. Джалоте, Інтегрований підхід до розробки програмного забезпечення (2-ге видання), 2017.
12. Т. ДеМарко, Т. Лістер, Вальс з ведмедами: Управління ризиками програмного забезпечення, 2014.

References

1. W.E. Lewis, Software Testing and Continuous Quality Improvement (3rd ed.), 2016.
2. F. Nomman, Software metrics: a rigorous and practical approach (3rd ed.), 2014.
3. J. Lienig, H. Bruemmer, "Reliability Analysis", Fundamentals of Electronic Systems Design, 2017.
4. A. Birolini: Reliability Engineering: Theory and Practice, Berlin 2013.
5. I. Schieferdecker, A. Hoffmann, Model-Based Testing, 2010.
6. J. Barnard, A. Price, Software Testing: Concepts and Operations, 2016.
7. S. Kumar, S. Rathore, Software Fault Prediction: A Roadmap, 2017.
8. T. Dingsoyr, T. Dyba, N. Brede, Agile Software Development: Current Research and Future Directions, 2018.
9. J. O'Regan, A Practical Approach to Software Quality, 2014.
10. Sh. Yamada, Software Reliability Modeling: Fundamentals and Applications, 2014.
11. P. Jalote, An Integrated Approach to Software Engineering (2nd ed.), 2017.
12. T. DeMarco, T. Lister, Waltzing with bears: Managing risk on software projects, 2014.

Міністерство освіти і науки України
Хмельницький національний університет



ЗБІРНИК НАУКОВИХ ПРАЦЬ
за матеріалами XV Всеукраїнської науково-практичної конференції
«Актуальні проблеми комп'ютерних наук АПКН-2023»

17-18 листопада 2023

Хмельницький 2023

АКТУАЛЬНІ ПРОБЛЕМИ КОМП'ЮТЕРНИХ НАУК - 2023*XV Всеукраїнська науково-практична конференція*

Метою конференції є висвітлення актуальних проблем комп'ютерних наук, інформатики та інформаційних технологій.

СЕКЦІЇ КОНФЕРЕНЦІЇ:

1. Комп'ютерні науки та прикладні інформаційні технології.
2. Комп'ютерна інженерія та системи захисту інформації.
3. Математичне моделювання та інженерія програмного забезпечення
4. Телерадіокомунікації, медійні та комунікаційні системи.
5. Проблеми впровадження інформаційних технологій у виробництво та управління.

Робочі мови конференції: українська, англійська

ОРГКОМІТЕТ:

Олег СИНЮК – голова оргкомітету, проректор Хмельницького національного університету з наукової роботи, доктор технічних наук, професор

Олег САВЕНКО – заступник голови оргкомітету, декан факультету Інформаційних технологій ХНУ, доктор технічних наук, професор

Олександр БАРМАК – заступник голови оргкомітету, завідувач кафедри Комп'ютерних наук ХНУ, доктор технічних наук, професор

Тетяна ГОВОРУЩЕНКО – завідувач кафедри Комп'ютерної інженерії та інформаційних систем ХНУ, доктор технічних наук, професор

Олена ВИСОЦЬКА – доктор технічних наук, завідувач кафедри Радіоелектронних та біомедичних комп'ютеризованих засобів і технологій Національного аерокосмічного університету ім. М. Є. Жуковського «Харківський авіаційний інститут», професор

Євгеній ЛАВРОВ – доктор технічних наук, професор (Сумський державний університет)

Людмила ТІМОФЄЄВА – відповідальна за студентську науково-дослідну роботу ХНУ

Олександр МАЗУРЕЦЬ – секретар конференції, к.т.н., доцент кафедри Комп'ютерних наук ХНУ

Марина МОЛЧАНОВА – секретар конференції, викладач кафедри Комп'ютерних наук ХНУ

КОНТАКТНА ІНФОРМАЦІЯ:

e-mail для листування: apkt.khnu@gmail.com

УДК 004.052.32

Сверба А.А.

*Хмельницький національний університет***УДОСКОНАЛЕННЯ МЕТОДУ РОБОТИ З МЕТРИКОЮ СЕРЕДНЬОГО ЧАСУ МІЖ ВІДМОВАМИ ДЛЯ ЗАБЕЗПЕЧЕННЯ ЕФЕКТИВНОГО ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

Розглянуто основні аспекти метрики середнього часу між відмовами, що застосовується на етапі тестування програмного забезпечення. Представлено підходи до удосконалення методу роботи з даною метрикою з використанням різних додаткових компонентів та аспектів сфери інженерії програмного забезпечення. Запропоновані способи удосконалення мають на меті покращити результати розрахунку показника метрики, дати можливість швидкого реагування на зміни параметрів надійності та забезпечити ефективніше оцінювання результатів тестування в цілому.

The main aspects of the mean time between failures metric, which is used at the stage of software testing, are considered. Approaches to improving the method of working with this metric using various additional components and aspects of software engineering are presented. The proposed methods of improvement are aimed at improving the results of the calculation of the metric indicator, enabling a quick response to changes in reliability parameters, and ensuring a more efficient evaluation of test results in general.

Інформаційний прогрес суспільства виступає рушійною силою для розробки різноманітного програмного забезпечення, яке використовується у всіх сферах нашого життя. Життєвий цикл ПЗ поділений на певну кількість етапів, кожен з яких є однаково важливими для загального функціонування програми. Одним з них є тестування. Тестування програмного забезпечення є етапом розробки, що забезпечує виправлення помилок, виявлення вразливостей ПЗ і забезпечення відповідності програми вимогам користувачів [1].

Процес безпосереднього тестування відбувається за допомогою різних засобів та додаткових компонентів, які сприяють збільшенню якості отриманих результатів тестування. Одним з таких компонентів виступають метрики тестування. Метрики – це вимірювані параметри та показники, які використовуються для оцінки процесу та результатів тестування програмного забезпечення [2]. Вони надають об'єктивну інформацію про якість програми, ефективність тестування та його прогрес, що допомагає розробникам та тестувальникам приймати обґрунтовані рішення та вдосконалювати процес розробки.

Важливою метрикою етапу тестування програмного забезпечення є метрика середнього часу між відмовами (MTBF), яка використовується для оцінки надійності ПЗ та вимірювання часу його безпомилкової роботи [2]. Існують різні варіанти вимірювання даної метрики, кожен з яких описує власний підхід до перевірки стабільності роботи програми. Проте наявні способи розрахунку використовують більш традиційний підхід знаходження результатів метрики, що може призвести до зниження ефективності тестування програмного забезпечення. Саме тому існує потреба удосконалення методу роботи з метрикою MTBF для покращення надійності роботи ПЗ.

Доступні на сьогодні публікації та дослідження даної тематики описують певні способи, що можуть покращити процес роботи з метрикою MTBF. Серед наявних покращень вказані такі: використання різних статистичних методів для точнішого прогнозу надійності роботи програмного забезпечення, застосування методів машинного навчання для якіснішого знаходження та представлення результатів процесу оцінки надійності [3], застосування різних архітектурних моделей програмних систем для деталізації процесу взаємодії різних компонентів системи та їх вплив на надійність загальної роботи ПЗ [4], використання емпіричних даних про відмови в роботі програми та їх послідовний моніторинг для встановлення певних закономірностей у разі зупинки роботи ПЗ [5]. Аналіз останніх публікацій та досліджень є дуже важливим, оскільки основні їх моменти будуть слугувати для досягнення мети дослідження.

Основною метою даного дослідження є опис та представлення способів удосконалення методу роботи з метрикою середнього часу між відмовами, висвітлення їх ключових аспектів та напрямків для подальшого прогресу.

Враховуючи основне завдання роботи, можна сформулювати декілька нових підходів до роботи з метрикою середнього часу між відмовами. Одним з перших способів удосконалення є розвиток нових моделей та алгоритмів для розрахунку показника середнього часу між відмовами. Прогрес в дослідженні нових структур може внести суттєвий вклад у покращення методу MTBF та оцінки надійності програмного забезпечення. Основні напрямки розвитку можуть включати: розробку моделей та алгоритмів, які базуються на складних взаємозв'язках між компонентами програмного забезпечення; покращене моделювання, основні принципи якого разом із оцінкою MTBF повинні бути використані ще на ранніх етапах розробки програмного забезпечення; використання штучного інтелекту для якіснішого прогнозування показника MTBF на основі вже відомих даних про роботу програмної системи.

Наступним загальним способом удосконалення методу роботи з метрикою MTBF є інтеграція з іншими метриками. Поеднання різних метрик допомагає сформувати більш докладне уявлення про стабільність системи і виявити можливі проблеми, які могли бути пропущені у випадку використання лише метрики MTBF. Різні метрики ґрунтуються на різних показниках і призначені для надання оцінки в конкретних ситуаціях. У спільному використанні з MTBF вони можуть стати

надійним інструментом для оцінки надійності та загальної продуктивності програми. Серед метрик, які можуть бути використані в комбінації з метрикою середнього часу між відмовами, існують такі: MTTF (метрика середнього часу до першої відмови), SER (метрика, яка дозволяє визначити кількість помилок, виявлених у програмному забезпеченні протягом певного періоду час), TTF (метрика, що визначає час до відмови окремого компонента або модуля системи), MTTR (метрика середнього часу відновлення ПЗ після відмови), динамічні метрики для покращення вимірювання різних показників (Code Coverage, Fault Detection Rate, Failure Rate).

Ще одним способом для удосконаленої роботи з метрикою MTBF може бути впровадження автоматизації розрахунку показника метрики. Введення такого заходу може значно спростити розрахунки, вдосконалити метрику середнього часу між відмовами програмного забезпечення і підвищити його ступінь надійності. В межах даного напрямку покращення важливим може бути використання спеціалізованих програмних інструментів для швидкого та точного розрахунку показника MTBF на основі емпіричних даних про ПЗ, а також впровадження інтеграції з системами моніторингу та журналювання для автоматичного збору інформації про помилки у роботі ПЗ та інші необхідні параметри.

Останнім способом удосконалення методу роботи з метрикою середнього часу між відмовами може виступати розробка методів та інструментів для моніторингу MTBF в реальному часі. У поєднанні з попереднім методом дана процедура представляє собою абсолютно новий підхід до збору інформації про всю роботу програмної системи. Така інформація надходить автоматично в режимі реального часу, що дозволяє реагувати на будь-які зміни та постійно відстежувати показник MTBF. Також швидкий збір інформації сприяє негайному проведенню її аналізу для прийняття різних рішень, що пов'язані з роботою програмного забезпечення.

Отже, інтеграція запропонованих у статті підходів до вдосконалення методики використання метрики MTBF сприятиме підвищенню точності оцінки результатів тестування програмного забезпечення та забезпечить створення більш надійних програмних систем. Подальші дослідження можуть бути спрямовані на практичне впровадження та перевірку ефективності запропонованих підходів.

Перелік посилань:

1. В.Е. Льюїс, Тестування програмного забезпечення та постійне покращення якості (3-є видання), 2016.
2. Ф. Норман, Метрики програмного забезпечення: суворий і практичний підхід (3-тє видання), 2014.
3. Є. Лініг, Х. Брюммер, «Аналіз надійності», Основи проектування електронних систем, 2017.
4. А. Біроліні, Розробка надійності: теорія та практика, Берлін 2013.
5. С. Кумар, С. Ратор, Прогноз програмних збоїв: дорожня карта, 2017.

ДОДАТОК Б
(обов'язковий)

ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА

ТЕМА: УДОСКОНАЛЕННЯ МЕТОДУ РОБОТИ З МЕТРИКОЮ
СЕРЕДНЬОГО ЧАСУ МІЖ ВІДМОВАМИ ДЛЯ ЗАБЕЗПЕЧЕННЯ
ЕФЕКТИВНОГО ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ ТЕСТУВАННЯ
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

СТУДЕНТ: СВЕРБА АРТЕМ АНДРІЙОВИЧ

КЕРІВНИК: БЕДРАТЮК ЛЕОНІД ПЕТРОВИЧ, Д-Р, ФІЗ.-МАТ. НАУК, ПРОФЕСОР

АКТУАЛЬНІСТЬ РОБОТИ

- Актуальність теми роботи зумовлена постійним розвитком технологій та зростанням вимог до надійності програмного забезпечення та полягає у пошуку методів для підвищення якості програмних продуктів та зменшення ризиків, пов'язаних із виникненням відмов.

ОБ'ЄКТ, ПРЕДМЕТ ТА МЕТА РОБОТИ

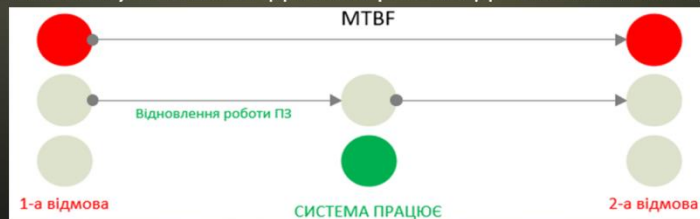
- Об'єктом дослідження є процес тестування програмного забезпечення.
- Предметом дослідження виступає методика роботи з метрикою середнього часу між відмовами (MTBF).
- Мета дослідження – розроблення удосконаленої методики роботи з метрикою середнього часу між відмовами для підвищення ефективності оцінювання результатів тестування програмного забезпечення.

ЗАВДАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

- Для досягнення поставленої мети необхідно виконати такі завдання:
 - ✓ виконати детальний аналіз предметної області;
 - ✓ проаналізувати наявні методи роботи з метрикою середнього часу між відмовами (MTBF) та їх застосування у процесі тестування ПЗ;
 - ✓ визначити особливості наявних методів та сформулювати пропозиції для їх покращення;
 - ✓ визначити вимоги та розробити удосконалену методику роботи з метрикою середнього часу між відмовами;
 - ✓ провести експериментальне дослідження та виконати оцінку ефективності застосування удосконаленої методики;
 - ✓ описати рекомендації щодо використання удосконаленої методики MTBF для різних видів програмного забезпечення;
 - ✓ оцінити перспективи подальшого розвитку та удосконалення методики роботи з метрикою MTBF.

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

- Тестування програмного забезпечення – це процес, що спрямований на перевірку та оцінку якості програмного продукту з метою виявлення помилок, недоліків та відхилень від вимог.
- Метрика середнього часу між відмовами (MTBF) – це метрика, яка визначає середню тривалість часу між відмовами або збоями в програмному забезпеченні, системі або в будь-якому її компоненті. MTBF використовується для оцінки надійності та стійкості об'єкта та вказує на середній інтервал часу, протягом якого очікується безвідмовна робота до настання наступної відмови.



ІСНЮЮЧІ ПІДХОДИ ДО ПОКРАЩЕННЯ МЕТОДУ РОБОТИ З МЕТРИКОЮ MTBF

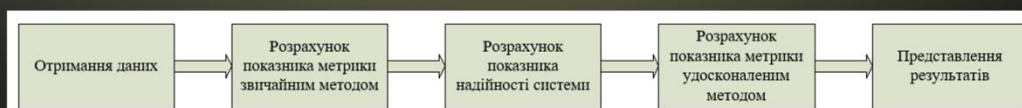
- Існують різні сучасні підходи до покращення методу роботи з метрикою:
 - використання статистичних методів;
 - застосування машинного навчання;
 - інтеграція з іншими методами оцінки надійності ПЗ;
 - моделювання ПЗ на основі архітектури;
 - розробка адаптивних методів оцінки;
 - використання алгоритмів та моделей;
 - застосування систем керування вадами;
 - використання інструментів для автоматизації розрахунку.

АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ДО ПОКРАЩЕННЯ

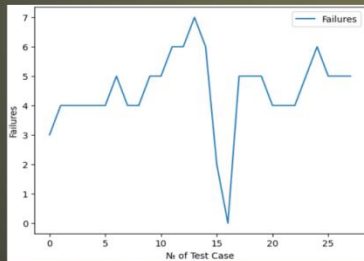
- Існуючі підходи до покращення методу роботи з метрикою середнього часу вимагають додаткових навичок та різноманітних інструментів для свого застосування. Деякі зі способів вимагають складної алгоритмізації або ускладненого моделювання, інші – великої кількості обчислювальних ресурсів та історичних даних про відмови програмного забезпечення, що на загальному рівні можуть бути значними обмеженнями для використання в межах тестування певних програмних продуктів.

РЕАЛІЗАЦІЯ УДОСКОНАЛЕНОЇ МЕТОДИКИ РОБОТИ З МЕТРИКОЮ МТВФ

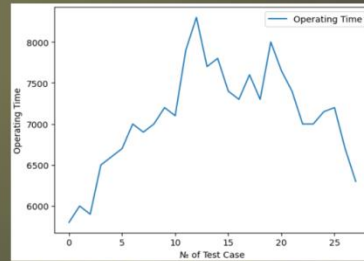
- Реалізація удосконаленої методики роботи з метрикою середнього часу між відмовами була здійснена за допомогою високорівневої мови Python, відповідних їй бібліотек Pandas, NumPy, Matplotlib та середовища для розробки Jupyter Notebook.
- Основний принцип роботи удосконаленої методики відповідає наступній поданій схемі:



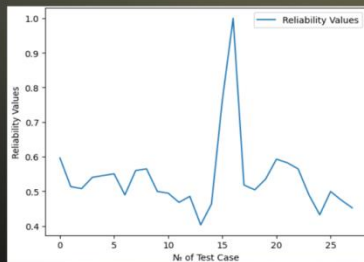
РЕЗУЛЬТАТИ РОБОТИ УДОСКОНАЛЕНОЇ МЕТОДИКИ



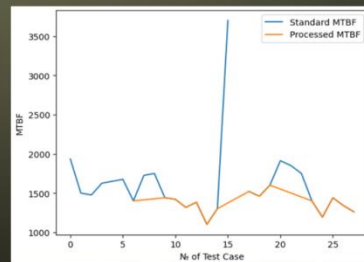
Вхідні дані про відмови



Вхідні дані про загальний час роботи ПЗ



Результат розрахунку значень надійності ПЗ

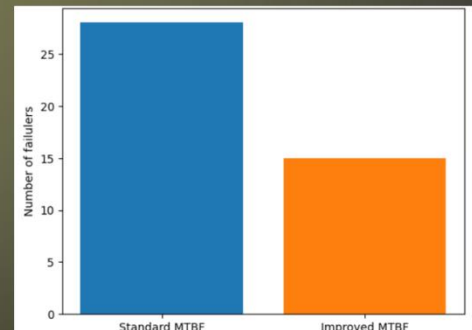


Результат роботи удосконаленої методики

АНАЛІЗ РЕЗУЛЬТАТІВ ТА ОЦІНКА ЕФЕКТИВНОСТІ ЗАСТОСУВАННЯ УДОСКОНАЛЕНОЇ МЕТОДИКИ

■ За результатами роботи удосконаленої методики було визначено:

- методика здатна виявляти критичні та вразливі місця роботи ПЗ;
- після застосування методики спостерігається зниження кількості відмов у додатку на 35%;
- методика здатна збільшити загальний час між відмовами програмного забезпечення;
- завдяки покращеній методиці час та витрати на тестування зменшуються на 25%;
- в загальному методика позитивно впливає на підвищення якості та надійності програмного забезпечення.



ПЕРСПЕКТИВИ ПОДАЛЬШОГО РОЗВИТКУ ТА УДОСКОНАЛЕННЯ МЕТОДИКИ РОБОТИ З МТВФ

- Загальні перспективи подальшого розвитку методики є наступними:
 - розробка продуктивних та водночас малозатратних алгоритмів та моделей;
 - посилене використання машинного навчання для прогнозування МТВФ;
 - інтеграція метрики МТВФ з новими показниками надійності ПЗ;
 - адаптація методики до конкретної галузі застосування;
 - розробка методів для врахування впливу апаратного забезпечення на МТВФ;
 - врахування впливу людського фактору на МТВФ.

НАУКОВА НОВИЗНА ОТРИМАНИХ РЕЗУЛЬТАТІВ

- Наукова новизна отриманих результатів дослідження є такою:
 - вперше розроблено комплексний підхід до удосконалення методу роботи з метрикою МТВФ, що враховує різну специфіку тестування програмного забезпечення, що дозволяє підвищити точність прогнозування надійності програмного забезпечення та забезпечити більш ефективне оцінювання результатів відповідного етапу;
 - отримав подальший розвиток метод врахування контексту програмного забезпечення при роботі з метрикою МТВФ, що передбачає збір, обробку та аналіз додаткових даних про програмне забезпечення, його архітектуру, мову програмування, зовнішні залежності та інші характеристики, які можуть впливати на надійність та результати тестування.

ПРАКТИЧНЕ ЗНАЧЕННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

- Практичне значення результатів полягає у наступному:
 - удосконалена методика роботи з метрикою MTBF може бути впроваджена в процес тестування програмних продуктів різних видів для забезпечення високої їх надійності;
 - результати дослідження можуть бути використані при розробці тестових планів, стратегій та різних сценаріїв, оскільки вони дають змогу оптимізувати процес тестування та забезпечити більш ефективне виявлення та усунення дефектів програмного забезпечення;
 - рекомендації щодо використання удосконаленої методики MTBF можуть бути використані при навчанні спеціалістів з тестування ПЗ.

НАУКОВІ ПУБЛІКАЦІЇ

- Сверба А. А., Мартинюк В. В., Удосконалення методу роботи з метрикою середнього часу між відмовами для забезпечення ефективного оцінювання результатів тестування програмного забезпечення. Вісник Хмельницького національного університету – 2023, №3 – 120-127 с.
- Сверба А. А., Мартинюк В. В., Удосконалення методу роботи з метрикою середнього часу між відмовами для забезпечення ефективного оцінювання результатів тестування програмного забезпечення. Збірник наукових праць конференції АПКН-2023.

ВИСНОВКИ

- Виконання кваліфікаційної роботи було здійснене відповідно до сформованих завдань. Кожна з поставлених задач була виконана в повній мірі. Спочатку було виконано детальний аналіз предметної області. Відповідно до цього було представлено загальну інформацію щодо тестування ПЗ та ключових особливостей даного етапу. Далі було описано метрику середнього часу між відмовами, основні аспекти її застосування, позитивні та негативні сторони.
- На наступному етапі було виконано опис та аналіз наявних методів роботи з метрикою MTBF та основних факторів її застосування під час тестування ПЗ. На основі поданого аналізу було виокремлено особливості кожного методу та подано пропозиції щодо подальшого їх покращення.
- На наступному кроці була проведена розробка удосконаленої методики роботи з метрикою MTBF. Для цього спочатку було визначено вимоги покращеної методики, далі – виконано безпосередній процес реалізації. Після цього згідно розробленої методики було виконано дослідження результатів її застосування та представлено оцінку її ефективності.
- На останніх етапах виконання роботи було представлено рекомендації щодо використання удосконаленої методики MTBF для різних видів програмного забезпечення та визначено перспективи щодо подальшого її розвитку та удосконалення.

Завідувачу кафедри інженерії програмного
забезпечення проф. Леоніду БЕДРАТЮКУ
здобувача вищої освіти
Сверби Артема Андрійовича
факультет ІТ, 2 курс, група ІПЗм-22-1

ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності в Хмельницькому національному університеті, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку й збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

06.12.2023
Дата

Artem
Підпис

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 19.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 6%

ID: 121873 Назва: Удосконалення методу роботи з метрикою середнього часу між відмовами для забезпечення ефективного оцінювання результатів тестування програмного забезпечення Додано в БД: 2023-12-05 Автора: Сверба А. А. Керівники: д-р фіз.-мат. наук, професор Л. П. Бедратюк Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	179713	1319	34581 (19%)	269 (20%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми
119396	Назва: Звіт з науково-дослідної практики Додано в БД: 2023-10-19 Автора: Сверба А.А. Керівники: Бедратюк Л.П. Консультанти: Опоненти:	34011 (19.0%)	260 (20.0%)



Ім'я користувача:
ІПЗ

ID перевірки:
1015974502

Дата перевірки:
06.12.2023 00:27:31 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
06.12.2023 00:31:24 EET

ID користувача:
100012953

Назва документа: КвР_Сверба А. А._ІПЗм-22-1

Кількість сторінок: 105 Кількість слів: 23752 Кількість символів: 190041 Розмір файлу: 396.28 KB ID файлу: 1015653883

4.39% Схожість

Найбільша схожість: 3.7% з Інтернет-джерелом (<https://vottp.khmnu.edu.ua/index.php/vottp/article/download/155/156>)

4.36% Джерела з Інтернету

114

Сторінка 107

0.32% Джерела з Бібліотеки

81

Сторінка 107

0.18% Цитат

Цитати

2

Сторінка 108

Не знайдено жодних посилань

0% Вилучень

Немає вилучених джерел

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувач Сверба Артем АндрійовичТема Удосконалення методу роботи з метрикою середнього часу між відмовами для забезпечення ефективного оцінювання результатів тестування програмного забезпеченняСпеціальність 121 «Інженерія програмного забезпечення»**Обсяг кваліфікаційної роботи:**

Кількість листів креслень _____; кількість сторінок записки _____ 130

1. Короткий зміст роботи та прийнятих рішень У кваліфікаційній роботі виконано аналіз предметної області з послідовним представленням основних аспектів метрики середнього часу між відмовами. Також було представлено сучасні підходи до покращення роботи з метрикою та подано пропозиції щодо майбутнього їх використання. Зважаючи на всі аспекти, було визначено вимоги щодо розробки удосконаленої методики, критерії оцінки її використання та здійснено безпосередню її реалізацію. На основі застосування удосконаленої методики було представлено відповідні результати, виконано їх аналіз та подано загальну оцінку її ефективності. На завершення було представлено особливості використання методики для різного програмного забезпечення та подано рекомендації щодо подальшого її удосконалення.

2. Висновок про відповідність роботи поставленому завданню Кваліфікаційна робота освітнього ступеня «магістр» повністю відповідає тематиці та є розробленою згідно поставлених завдань, вимог та стандартів.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У розділі «Вступ» було визначено актуальність та мету дослідження, сформульовано завдання роботи, описано її наукову новизну та практичне значення результатів. У першому розділі було виконано детальний аналіз предметної області, представлено загальну інформацію про метрику середнього часу між відмовами, виокремлено ключові особливості її застосування, позитивні сторони та обмеження, здійснено постанову задачі дослідження щодо удосконалення методики роботи з метрикою. В другому розділі було представлено наявні підходи до покращення показника часу безвідмовної роботи, виконано їх порівняльний аналіз та на основі нього сформульовано загальні пропозиції щодо подальших покращень. У третьому розділі було визначено вимоги щодо розробки удосконаленої методики, представлено основні критерії оцінки її застосування, виконано безпосередню її реалізацію, а також обрано проект для її валідації та здійснено опис процесу його тестування з використанням методики. У четвертому розділі було представлено результати роботи удосконаленої методики, виконано їх аналіз та загальну оцінку ефективності її застосування, представлено рекомендації щодо використання методики для різних видів програмного забезпечення та наведено перспективи її подальшого розвитку та удосконалення.

4. Позитивні сторони роботи Кваліфікаційна робота представляє удосконалену методику роботи з метрикою середнього часу між відмовами та містить загальні рекомендації щодо її використання та розвитку в майбутньому. Результати роботи з методикою вказують на доцільність її застосування під час проведення тестування програмного забезпечення, що матиме позитивний вплив на здійснення оцінювання надійності ПЗ та прийняття рішень щодо її підвищення та підтримки.

5. Негативні сторони роботи Для функціонування удосконаленої методики роботи з метрикою середнього часу між відмовами необхідна достатня кількість історичних даних про відмови та час загального оперування програмного забезпечення. У разі відсутності необхідної інформації методика не зможе виконувати покладені завдання щодо вимірювання та прийняття рішень для підвищення рівня надійності ПЗ. Також для подальшого покращення важливо працювати в напрямку пошуку та впровадження нових способів отримання інформації для роботи методики.

6. Оцінка графічного оформлення та пояснювальної записки роботи Графічне оформлення виконано згідно теми кваліфікаційної роботи та представлено за допомогою відповідних рисунків. Пояснювальна записка оформлена згідно вимог чинних стандартів.

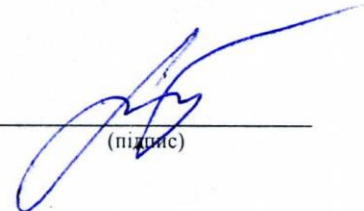
7. Відгук про роботу в цілому Кваліфікаційна робота заслуговує на позитивну оцінку. Пояснювальна записка дипломної роботи є розробленою згідно всіх вимог та положень стандартів. Матеріал записки є чітко визначеним та структурованим, що допомагає в повній мірі відобразити весь викладений матеріал відповідно до обраної тематики та визначених завдань. Графічні матеріали та подання надають можливість для візуального представлення різних складових роботи з метою відображення основних прийнятих рішень для виконання поставленої задачі дослідження.

8. Інші зауваження _____

9. Оцінка кваліфікаційної роботи Кваліфікаційна робота повною мірою виконана та оформлена згідно вимог, правил та стандартів. Проаналізувавши всі сторони роботи, ступінь її відповідності тематиці та поставленим завданням дослідження, було зроблено висновок, що вона заслуговує на оцінку «відмінно».

РЕЦЕНЗЕНТ (прізвище, ім'я, по-батькові, посада, місце роботи) _____
 Говорущенко Т. О., д. т. н., проф., завідувач кафедри комп'ютерної інженерії та інформаційних систем

« 6 » _____ грудня 202_3 р.


 (підпис)

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатами звіту/звітів подібності щодо роботи, продуктованими програмно-технічним засобом(ами) перевірки текстів на плагіат.

Назва: «Удосконалення методу роботи з метрикою середнього часу між відмовами для забезпечення ефективного оцінювання результатів тестування програмного забезпечення»

Автор: Сверба Артем Андрійович

Спеціальність: 121 – Інженерія програмного забезпечення

Освітня програма: Освітньо-професійна програма «Інженерія програмного забезпечення»

Науковий керівник: Бедратюк Леонід Петрович, д-р фіз.-мат. наук, професор

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені у розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за два дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої й електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені у розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того, як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Запозичення, виявлені у роботі, є законними і не є плагіатом, оскільки:

1) у тексті кваліфікаційної роботи системою перевірки на плагіат Unicheck виявлено схожість з деякими документами у частині загальноживаних обов'язкових словосполучень у стандартних бланках (титулка, бланк завдання), у структурі змісту, у назвах розділів/підрозділів, у назвах публікацій переліку джерел посилання тощо;

2) в якості запозичень системою Unicheck було зафіксовано деякі послідовності вихідного коду і посилання на бібліотеки, які є стандартними мовними конструкціями програмування та не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;

3) запозичення, виявлені в тексті роботи, є фрагментарними або мають належним чином оформленні посилання;

4) виявлені модифікації тексту не впливають на відсоток схожості.

Максимальний обсяг запозичень, визначений системою Anti-Plagiarism, складає 19.0%. Обсяг запозичень, визначений системою Unicheck виявлення збігів ідентичності/схожості, складає 4.39% і адресується до 114 джерел з інтернету і 81 джерела з бібліотеки, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь кваліфікаційної роботи.

Дата 06.12.2023

Завідувач кафедри ІПЗ

Гарант освітньої програми

Керівник кваліфікаційної роботи



Леонід БЕДРАТЮК

Оксана ЯШИНА

Леонід БЕДРАТЮК