

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістра

(освітній ступінь)

на тему: **Методи та засоби управління і моніторингу показників
надійності при розробці комп'ютерних систем**

Виконав: студент (ка) 6 курсу, групи СІм-62
спеціальності 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

	(підпис)	Крисюк І.В. (прізвище та ініціали)
Керівник	(підпис)	Ясній О.П. (прізвище та ініціали)
Нормоконтроль	(підпис)	Тиш Є.В. (прізвище та ініціали)
Завідувач кафедри	(підпис)	Осухівська Г.М. (прізвище та ініціали)
Рецензент	(підпис)	Боднарчук І.О. (прізвище та ініціали)

Тернопіль
2023

Міністерство освіти і науки України
 Тернопільський національний технічний університет імені Івана Пулюя
 (повне найменування вищого навчального закладу)

Факультет комп'ютерно-інформаційних систем і програмної інженерії
 Кафедра комп'ютерних систем та мереж

ЗАТВЕРДЖУЮ

Завідувач кафедри Осухівська Г.М.

«_____» _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр
 (назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»
 (шифр і назва спеціальності)

студенту Крисюку Ігорю Володимировичу
 (прізвище, ім'я, по-батькові)

1. Тема проекту (роботи) Методи та засоби управління і моніторингу показників надійності при розробці комп'ютерних систем

Керівник проекту (роботи) Ясній Олег Петрович, д.т.н., проф.
 (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «01» грудня 2023 року №4/7-1132

2. Термін подання студентом завершеної роботи 28.12.2023

3. Вихідні дані до роботи Показники надійності комп'ютерних систем, критерії оцінювання надійності, типи помилок і дефектів комп'ютерних систем

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ. 1. Аналіз моделей і методів управління та моніторингу показників надійності при розробці комп'ютерних систем 2. Модель та показники надійності комп'ютерних систем
3. Засіб автоматизованого виявлення помилок у програмних модулях комп'ютерної системи
4. Охорона праці та безпека в надзвичайних ситуаціях. Висновки

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

1. Актуальність і мета дослідження. 2. Задачі дослідження, об'єкт і предмет, наукова новизна і практична цінність дослідження. 3. Вплив помилок і дефектів на відмови КС
4. Класифікація помилок на стадіях життєвого циклу КС. 5. Модель Nismo
6. Модифікована модель дефекту. 7. Результати виявлення дефектів 8. Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Охорона праці та безпека в надзвичайних ситуаціях</i>	<i>Осухівська Г.М.</i>		
	<i>Стадник І.Я., проф. каф. ОХ</i>		

7. Дата видачі завдання

20.11.2023

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	<i>Аналіз моделей і методів управління та моніторингу показників надійності при розробці КС</i>	<i>01.12.2023-</i>	<i>виконано</i>
2.	<i>Модель та показники надійності комп'ютерних систем</i>	<i>05.12.2023- 12.12.2023</i>	<i>виконано</i>
3.	<i>Засіб автоматизованого виявлення помилок у програмних модулях комп'ютерної системи</i>	<i>12.12.2023- 17.12.2023</i>	<i>виконано</i>
4.	<i>Охорона праці та безпека в надзвичайних ситуаціях</i>	<i>18.12.2023</i>	<i>виконано</i>
5.	<i>Оформлення пояснювальної записки</i>	<i>20.12.2023</i>	<i>виконано</i>
6.	<i>Оформлення графічного матеріалу</i>	<i>21.12.2023</i>	<i>виконано</i>
7.	<i>Попередній захист кваліфікаційної роботи магістра</i>	<i>22.12.2023</i>	<i>виконано</i>
8.	<i>Захист кваліфікаційної роботи магістра</i>	<i>28.12.2023</i>	

Студент

(підпис)

Крисюк І.В.

(прізвище та ініціали)

Керівник проекту (роботи)

(підпис)

Ясній О.П.

(прізвище та ініціали)

АНОТАЦІЯ

Методи та засоби управління і моніторингу показників надійності при розробці комп'ютерних систем// Кваліфікаційна робота магістра// Крисюк Ігор Володимирович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем та програмної інженерії, група СІМ-62 // Тернопіль, 2023 // с. – 89 , рис. – 33 , табл. –16 , аркушів А1 –8 , додат. – 1, бібліогр. – 21.

Ключові слова: метод, засіб, надійність, комп'ютерна система, моніторинг, управління.

У кваліфікаційній роботі магістра проведено аналіз моделей надійності компонентів комп'ютерних систем, типів помилок та їх ознак, а також впливу на надійність комп'ютерної системи.

Побудовано та обґрунтовано для застосування модель представлення показників надійності, що враховує концептуальну структуру комп'ютерних систем і дає можливість на основі аналізу помилок забезпечувати їх моніторинг та приймати управлінські рішення для усунення загрози відмови чи збою у роботі компонентів і системи в цілому.

Набув подальшого розвитку метод моніторингу дефектів програмних модулів комп'ютерних систем на основі моделі DDHM, що дає змогу проводити спостереження за його розвитком та візуалізувати вплив на надійність комп'ютерної системи у часі з врахуванням версій програмних модулів системи.

Спроектовано архітектуру та розроблено інтерфейси користувачів програмного засобу маніпулювання критеріями надійності, моделями прогнозування дефектів програмного забезпечення, формування імовірних шляхів поширення дефектів на компоненти комп'ютерної системи, що дало змогу забезпечити ефективність виявлення та моніторингу дефектів.

ABSTRACT

Methods and tools for managing and monitoring reliability indicators in the development of computer systems /Master's graduation thesis / Krysiuk Ihor / Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and software engineering, group CIm -62 // Ternopil, 2023// p. - 88, fig. – 33, table. – 16, Sheets A1 – 8, Add – 1, Ref. – 26.

Keywords: method, tool, reliability, computer system, monitoring, management.

In the master's qualification work, an analysis of the reliability models of computer system components, types of errors and their signs, as well as the impact on the reliability of the computer system was carried out.

The model for presenting reliability indicators has been built and justified for use, which takes into account the conceptual structure of computer systems and makes it possible, based on error analysis, to monitor them and make management decisions to eliminate the threat of failure or failure of components and the system as a whole.

The method of monitoring defects of software modules of computer systems based on the DDHM model has been further developed, which makes it possible to observe its development and visualize the impact on the reliability of the computer system over time, taking into account the versions of the software modules of the system.

The architecture was designed and the user interfaces of the software tool for manipulating reliability criteria, models for predicting software defects, and the formation of probable paths for the spread of defects to computer system components were developed, which made it possible to ensure the effectiveness of defect detection and monitoring.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1 АНАЛІЗ МОДЕЛЕЙ І МЕТОДІВ УПРАВЛІННЯ ТА МОНІТОРИНГУ ПОКАЗНИКІВ НАДІЙНОСТІ ПРИ РОЗРОБЦІ КОМП'ЮТЕРНИХ СИСТЕМ.....	12
1.1. Моделі надійності компонентів комп'ютерних систем	12
1.2. Аналіз підходів до управління та моніторингу прогнозування показників надійності комп'ютерних систем	18
1.3. Аналіз типів помилок у програмних складових комп'ютерних систем	21
1.4. Висновки до розділу	26
РОЗДІЛ 2 МОДЕЛЬ ТА ПОКАЗНИКИ НАДІЙНОСТІ КОМП'ЮТЕРНИХ СИСТЕМ	27
2.1. Визначення і математичний опис критеріїв надійності комп'ютерних систем	27
2.2. Модель надійності за наявності збоїв у ПЗ	33
2.3. Обґрунтування моделі представлення помилок у програмному забезпеченні комп'ютерної системи	36
2.4. Висновки до розділу	45
РОЗДІЛ 3 ЗАСІБ АВТОМАТИЗОВАНОГО ВИЯВЛЕННЯ ПОМИЛОК У ПРОГРАМНИХ МОДУЛЯХ КОМП'ЮТЕРНОЇ СИСТЕМИ	47
3.1. Побудова архітектури засобу автоматизованого управління та моніторингу показників надійності комп'ютерних систем	47
3.2. Підсистеми на рівні домену	51
3.3. Компонент візуалізації розвитку помилок	54
3.4. Підсистема моделей помилок, історій і версій	56
3.5. Підсистема формування звітів.....	58
3.6. Проектування схеми бази даних для зберігання даних управління і моніторингу показників надійності.....	60

3.7. Висновки до розділу	65
РОЗДІЛ 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	67
4.1. Охорона праці.....	67
4.2. Оповіщення керівного складу органів виконавчої влади, підприємств установ та організацій, населення про загрозу і виникнення НС природного, техногенного та воєнного характеру.....	70
4.3. Фактори ризику і можливі порушення здоров'я користувачів комп'ютерної мережі	73
ВИСНОВКИ.....	78
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	80
Додаток А Тези конференцій	83

ВСТУП

Актуальність теми. Технологічний розвиток в ІТ індустрії сприяє впровадженню прикладних застосувань у різних галузях виробництва та надання послуг. Основними засобами, які забезпечують автоматизацію як технологічних процесів, так і процесів обліку та управління ресурсами є комп'ютерні системи різного призначення. До структурних компонентів комп'ютерних систем входять апаратні пристрої, програмні модулі управління і передачі даних, а також канали зв'язку на рівні фізичних та логічних з'єднань. Такі системи забезпечують високий рівень функціональності та надійності, зручності використання і продуктивності.

Однак зростаюча складність проектування та реалізації компонентів комп'ютерних систем вимагає застосування нових методів, моделей та інструментів для управління і моніторингу показників надійності під час розробки компонентів системи.

Важливим аспектом при створенні комп'ютерних систем є контроль та своєчасне виявлення помилок у роботі як апаратного, так і програмного забезпечення, тому актуальним є також розробка методів і засобів виявлення помилок, визначення рівня загрози при наявності дефекту в одному з компонентів системи та прогнозуванні імовірності його поширення на інші модулі.

Своєчасно невиявлені помилки або дефекти програмних чи апаратних модулів можуть призводити до збоїв або відмов працездатності системи. Слід відмітити, що сучасний рівень розвитку методів і засобів тестування як апаратного, так і програмного забезпечення, дає змогу забезпечити виявлення дефектів на етапах життєвого циклу, однак внесені логічні помилки у програмне забезпечення можуть проявляти себе на завершальних етапах проектування або під час супроводу комп'ютерних систем. Тому, актуальною задачею є вдосконалення методів і засобів управління та моніторингу показників надійності комп'ютерних систем.

Дослідженню надійності комп'ютерних систем присвячено ряд наукових публікацій українських та закордонних вчених. Серед вітчизняних науковців необхідно відмітити вагомий внесок таких вчених як: Локазюк В.М., Лавріщева К.М., Коваль Г.І., Коротун Т.М., Волочій Б. Ю, Озірковський Л. Д., Сидоров М.О. та ін, серед закордонних – Lakey P.B., Musa J.D., Hecht H., Ohlsson N., Fenton N.E. та ін.

Враховуючи суттєвий внесок науковців у розвиток методів і засобів забезпечення та оцінювання надійності комп'ютерних систем, все ж не до кінця дослідженим залишається ряд питань щодо впливу і розвитку дефектів програмного забезпечення у часі, які можуть призвести до збоїв, на надійність комп'ютерних систем. Тому, задача дослідження впливу збоїв програмного забезпечення на надійність комп'ютерних систем є на сьогодні доволі актуальною задачею.

Мета роботи полягає у дослідженні моделей, методів і засобів управління та моніторингу показників надійності при розробці комп'ютерних систем.

Об'єктом дослідження є процеси управління та моніторингу показників надійності при розробці комп'ютерних систем.

Предметом дослідження є моделі, методи і засоби управління та моніторингу показників надійності комп'ютерних систем.

Задачі, які необхідно вирішити у кваліфікаційній роботі полягають у наступному:

- аналіз наукових підходів і практик дослідження надійності комп'ютерних систем;
- аналітичний огляд методів і засобів виявлення та аналізу помилок у компонентах комп'ютерних систем;
- побудова та обґрунтування моделей надійності комп'ютерних систем;
- побудова та представлення моделей помилок у комп'ютерних системах;

- розробка та обґрунтування методу аналізу впливу помилок у програмному забезпеченні на надійність комп'ютерних систем;
- проектування архітектури засобу аналізу помилок програмного забезпечення та надійності комп'ютерних систем.

Наукова новизна одержаних результатів при виконанні дипломної роботи полягає в наступному:

- уперше побудовано та обґрунтовано модель впливу дефектів програмного забезпечення на надійність комп'ютерної системи, що забезпечує можливість встановлення шляху поширення дефекту програмного забезпечення на інші компоненти комп'ютерної системи і дає змогу прогнозувати ймовірність виникнення негативного впливу на надійність комп'ютерної системи.
- набули подальшого розвитку модель і метод моніторингу дефектів програмного забезпечення комп'ютерних систем на основі модифікованої моделі Nismo, що дає змогу аналізувати розвиток і вплив дефекту програмного забезпечення на надійність комп'ютерної системи у часі та з врахуванням версій програмного забезпечення.

Методи дослідження. При виконанні кваліфікаційної роботи магістра використовувались наступні методи: аналіз та синтез – під час аналізу типів помилок у комп'ютерних системах, їх класифікації та визначенні шляхів управління та моніторингу показниками надійності; формалізація і моделювання – під час побудови моделі надійності комп'ютерної системи та представлення дефектів програмного забезпечення; об'єктно-орієнтовані аналіз, проектування та програмування – під час розроблення засобу аналізу помилок у комп'ютерних системах; експеримент – під час апробації запропонованих методу та засобу.

Практична цінність результатів дослідження. Практична цінність роботи полягає у створенні архітектури та реалізації програмного комплексу підтримки методу визначення впливу дефектів програмного забезпечення на надійність комп'ютерних систем.

Публікації. Результати кваліфікаційної роботи апробовані на XII Міжнародній науково-практичній конференції молодих учених та студентів (6-7 грудня 2023 р.) та XI науково-технічній конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (13-14 грудня 2023 року) як тези конференцій.

1. Ясній О.П., Крисюк І.В. Фактори впливу на надійність комп'ютерних систем в процесі їх розробки. Матеріали XII міжнародної науково-практичної конференції молодих учених та студентів «Актуальні задачі сучасних технологій» (6-7 грудня 2023 року). Тернопіль: ТНТУ. 2023. С. 462.

2. Ясній О.П., Крисюк І.В. Архітектура засобу розрахунку та управління показниками надійності комп'ютерних систем. Матеріали XI науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (13-14 грудня 2023 року). Тернопіль: ТНТУ. 2023. С. 161.

Структура роботи. Кваліфікаційна робота містить розрахунково-пояснювальну записку та графічний матеріал. До складу записки входить вступу, 4 розділи, загальні висновки, список використаних джерел і додатки. Обсяг роботи: розрахунково-пояснювальна записка – 88 арк. формату А4, графічна частина – 8 аркушів формату А1.

РОЗДІЛ 1

АНАЛІЗ МОДЕЛЕЙ І МЕТОДІВ УПРАВЛІННЯ ТА МОНІТОРИНГУ ПОКАЗНИКІВ НАДІЙНОСТІ ПРИ РОЗРОБЦІ КОМП'ЮТЕРНИХ СИСТЕМ

1.1. Моделі надійності компонентів комп'ютерних систем

Комп'ютерні системи представляють собою складні програмно-апаратні комплекси, які виконують перетворення деякої вхідної інформації у формат зручний для користувача і дають змогу приймати відповідні рішення [1]. Надійність апаратних пристроїв можна зобразити за допомогою моделей надійності технічних засобів. Однак ці моделі не можуть бути застосовані для опису характеристик надійності програмних модулів управління комп'ютерною системою, оскільки програмне забезпечення відрізняється від апаратного за своєю природою і типом відмов.

На відмінну від апаратного забезпечення комп'ютерної системи, програмна складова описується більш складними зв'язками та відношення і на неї не впливає фізичне старіння, як один з факторів надійності технічних засобів. Збої у роботі програмних модулів визначаються їх вмістом, кількістю внесених розробниками помилок та середовищем застосування.

Експериментально встановлено, що інтенсивність збоїв або відмов у ПЗ починає знижуватися у процесі аналізу програмних компонентів та ліквідації дефектів. Стабільна робота програмних складових комп'ютерних систем починає проявляється лише приблизно після 4 років після того, як система введена в експлуатацію. Рис. 1.1 демонструє графік інтенсивності відмов, які пов'язані зі збоями у роботі як апаратної, так і програмної складової комп'ютерної системи.

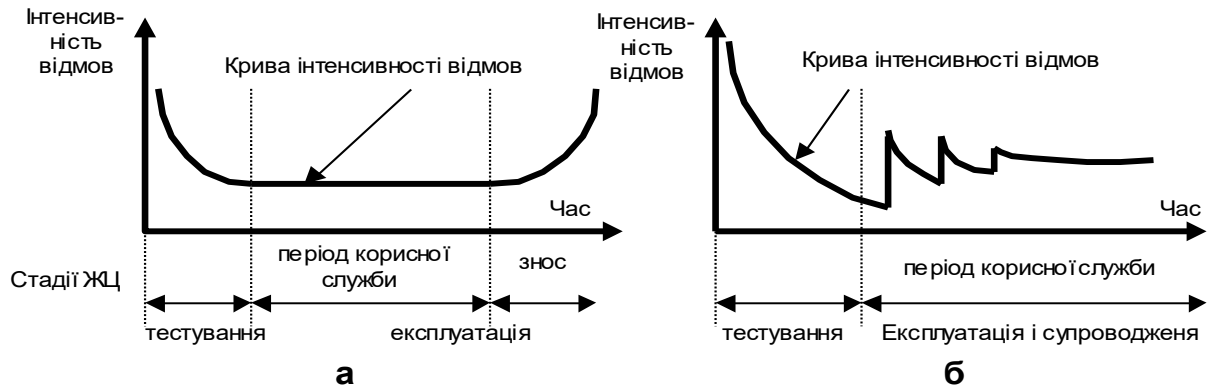


Рис. 1.1. Інтенсивність відмов програмних і апаратних складових комп'ютерної системи

Дефекти, які спричиняють збій у нормальному функціонуванні програмної складової КС зазвичай виникають через помилки розробників при реалізації системи на різних етапах виконання проектів. Графік таких помилок внесених у проект на різних етапах розробки КС за водоспадною моделлю проілюстровано на рис. 1.2.

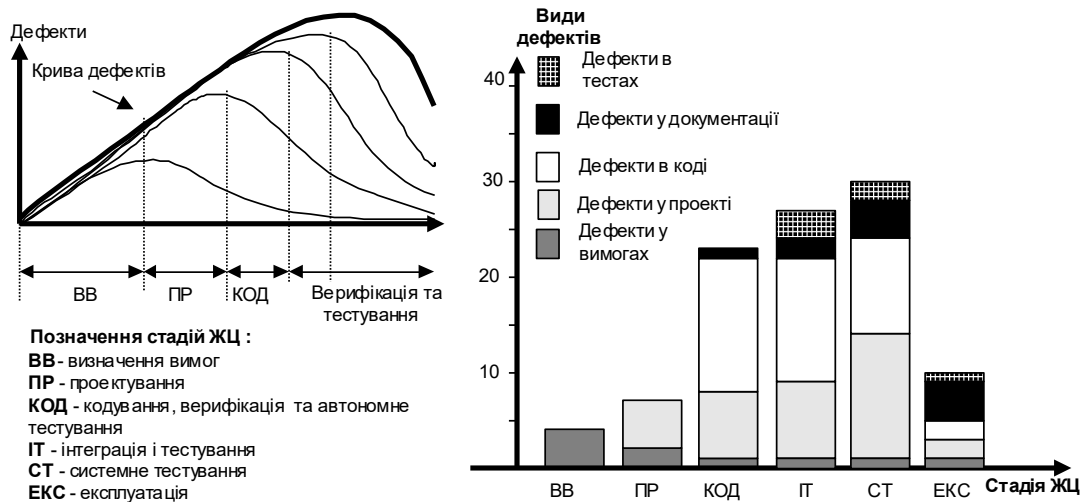


Рис. 1.2. Графік виявлення дефектів при застосуванні водоспадної моделі життєвого циклу

Залежність відмов, які провокують збій ПЗ КС, від дефектів і помилок у програмних компонентах, можуть бути викликані різними факторами і мати різну етіологію. Таку залежність і вплив продемонстровано на рис. 1.3.

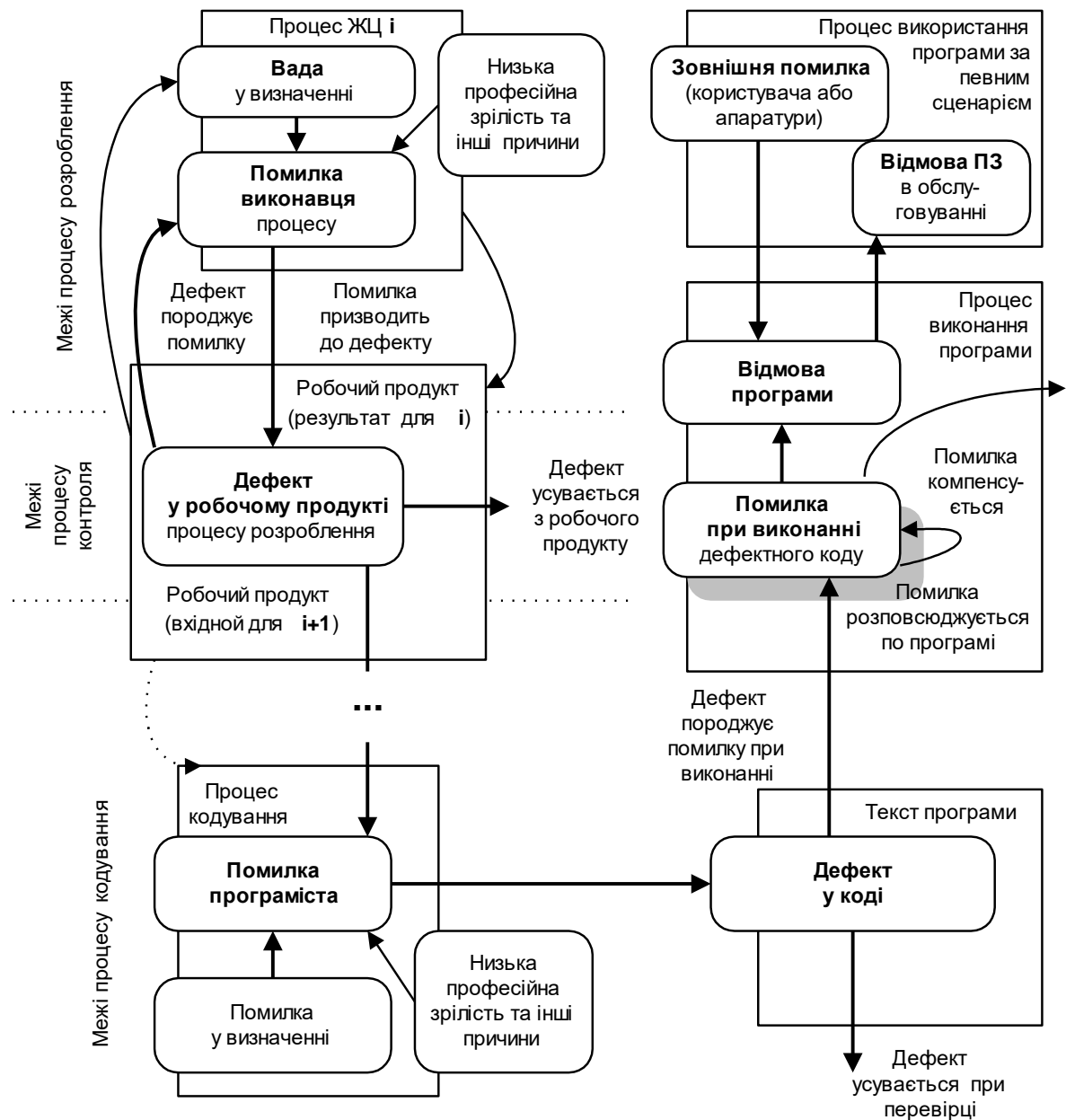


Рис. 1.3. Залежність відмові від помилок і дефектів у програмному забезпеченні комп'ютерної системи

У випадку, коли після проведених процедур рев'ю коду, unit та інтеграційного тестування, у програмних модулях залишаються невиявлені дефекти чи помилки, то це може призвести до пізнішої їх появи при функціонуванні та експлуатації комп'ютерної системи. У випадку, коли помилки проявляють себе після впровадження комп'ютерної системи у середовищі користувача, то це може спричинити ланцюгову реакцію щодо

неправильності її функціонування, а це в свою чергу негативно позначається на надійності всієї комп'ютерної системи і призводить до зниження відмовостійкості.

У зв'язку з наведеними вище фактами та на основі аналізу результатів графіків показників надійності комп'ютерних систем, важливим та актуальним завданням є забезпечення встановленого рівня надійності комплексу програмно-апаратних засобів та мінімізація помилок під час проектування та реалізації системи.

З метою забезпечення та підвищення рівня надійності програмних складових комп'ютерних систем необхідно розв'язати сукупність наступних задач:

- виявлення факторів, які впливають на надійність програмних компонентів з обґрунтуванням вимог надійності до них на різних фазах життєвого циклу;
- формування стратегії і технології, яка б забезпечувала реалізацію попередньо визначених вимог до надійності на кожній фазі проекту зі створення КС;
- виявлення об'єктів, які підлягають обов'язковій перевірці з одночасною розробкою процедур, які б забезпечували кількісне вираження атрибутів системи з метою прийняття оптимізаційних рішень.

Узагальнюючи результати проведених досліджень можна з упевненістю говорити, що надійність є однією з найбільш ключових характеристик якості комп'ютерної системи, а її підхарактеристика «Завершеність» - визначальною у контексті забезпечення стійкості функціонування системи. За допомогою кількісного представлення атрибутів завершеності комп'ютерної системи, особливо її програмної складової, можна встановити потенційний вплив помилок у ПЗ на надійність КС в цілому.

Сьогоднішній стан досліджень у сфері забезпечення надійності як апаратних пристроїв, так і програмних систем характеризуються побудовою

та обґрунтуванням великої кількості моделей, які дають змогу описати та оцінити їх завершеність. Проте існують ще відкриті питання відносно кількісного представлення показників надійності, їхнього обґрунтування, доцільності та адекватності застосування при оцінюванні на різних етапах виконання проекту. Також не до кінця дослідженими залишаються питання впливу відмов ПЗ на комплексний показник надійності комп'ютерних систем.

В залежності від типу і сфери застосування КС, факторів продуктивності та вартісних показників, підхарактеристика надійності «Завершеність» може бути описана за допомогою різних атрибутів.

Один з підходів для опису завершеності програмних модулів КС передбачає визначення єдиного цільового значення надійності для усіх компонентів КС без розподілу. Такий не критеріальний підхід варто застосовувати у випадку існування невизначеності, а за наявності документального опису модулів і доступу до коду, такий підхід не ефективно використовувати. Це пов'язано з необґрунтованістю вимог щодо надійності. Якщо задано надто високі критерії надійності програмних складових КС, то при їх реалізації можливе залучення додаткових коштів, а час реалізації значно зростає [5].

При застосуванні технічних підходів [6, 7] передбачається побудова блок-схем, які інтерпретують надійність апаратно-програмних комплексів або рівномірного пропорційного розподілу надійності відносно складових компонентів, які враховують послідовність і паралельність їх функціонування. Такі підходи є результатом дослідження методів і моделей притаманних для надійності технічних пристроїв і можуть бути використані при реалізації вбудованого ПЗ при умові існування показників надійності усіх компонентів КС.

Підходи, які передбачають використання специфікації програмних модулів мають широкий спектр методів та інструментів орієнтованих на

виявлення та проведення оцінки помилок ПЗ і як наслідок встановлення їх надійності на різних етапах виконання проекту.

Одна з перших методик, яка враховувала розподіл надійності, базувалась на розрахунку імовірності переходів між компонентами комп'ютерної системи, яка відображена у реалізованих алгоритмах функціонування ПЗ. Це дало поштовх для розвитку надійності програмних комплексів..

Іншим підходом [11] передбачено розподіл надійності за профілями, зокрема, щодо забезпечення функціональності програмних підсистем. В основі цього підходу лежить представлення системи у вигляді ієрархії функціональних компонентів, яка враховує особливості різних категорій користувачів, режимів функціонування системи, а також безпосередньо самих операцій. До переваг застосування такого підходу належить здатність оцінювати властивості складних і масштабних КС, створення сценаріїв тестування, які дають змогу врахувати аспекти середовища застосування системи та економія часових ресурсів при проведенні тестування програмних модулів КС. Серед недоліків підходу розподілу надійності за профілями варто виділити високу чутливість моделей при зміні операційного профілю.

Для аналізу надійності програмних складових КС може застосовуватися модель [12], яка орієнтована на розподіл надійності у профілі виконання ПЗ і застосовує принцип моделювання функцій на основі стохастичних процесів старту і функціонування модулів. При такому підході можна визначити патерни за якими виконуються функції ПЗ, оскільки проводиться процедура аналізу переходу між компонентами ПЗ КС. До недоліків практичного застосування підходу на основі розподілу надійності у профілі виконання належить те, що необхідно інтегрувати програмний код служб, які фіксуватимуть частоту звернення до конкретної складової ПЗ.

Серед моделей опису надійності на основі технічних підходів потрібно відмітити модель [12], в основі якої лежить розподіл надійності за критеріями їхньої критичності і специфікою відмов. Така модель використовує метод режимів дефектів програмного забезпечення та їхнього впливу. Перевагою використання цієї моделі є здатність забезпечити врахування аспектів КС, які можна промодельовати UML діаграмами.

Проаналізувавши різні моделі, які описують розподіл надійності ПЗ і дають змогу виявити вплив помилок на виникнення відмов комп'ютерної системи можна сформулювати вимоги до моделі надійності компонентів ПЗ:

- відображення потреб зацікавлених у комп'ютерній системі осіб при визначенні пріоритету і важливості сценаріїв використання системи та при зверненні до відповідних функцій;
- врахування точки зору розробників ПЗ щодо важливості і надійності реалізації функцій КС;
- аналіз і дослідження інтенсивності викликів і значень показників надійності визначених програмних модулів КС;
- забезпечення відповідності показників надійності при функціонуванні комп'ютерної системи очікуваним сподіванням користувачів.

1.2. Аналіз підходів до управління та моніторингу прогнозування показників надійності комп'ютерних систем

Одним з ключових процесів при реалізації комп'ютерних систем є прогнозування критеріїв надійності ПЗ, оскільки вони формують базу щодо керування даними і відповідними процесами у контексті середовища виконання. На початкових стадіях розробки КС доцільність побудови моделі надійності, зумовлена потребою в оцінюванні ресурсів, які б забезпечували досягнення визначеної цілі системи. Як наслідок, оцінка

ресурсів дозволяє забезпечити оптимальність їх розподілу за відповідними фазами виконання проекту з врахуванням потенційних відхилень від календарного плану, ефективність управління процесом реалізації системи з визначеним рівнем надійності її компонентів.

З точки зору практики, процес прогнозування надійності варто організувати, починаючи з ранніх стадій і закінчуючи фінальними фазами. У такому випадку варто застосовувати змішані моделі прогнозування, як для ранніх, так і пізніх етапів, які передбачають деяку задану інтераційність. Це дозволяє уточнювати параметри надійності і реалізувати спостереження щодо розвитку і ліквідації помилок впродовж усього життєвого циклу системи.

Прогнозування показників надійності дає змогу отримати важливу інформацію, зокрема:

- очікувану кількість помилок на різних фазах виконання проекту комп'ютерної системи;
- очікувану частоту та інтенсивність збоїв у програмних модулях комп'ютерної системи;
- потенційні шляхи оптимізації програмних складових і процесів імплементації КС.

Алгоритмічно процес прогнозування показників надійності можна представити у вигляді схеми, представленої на рис. 1.4.

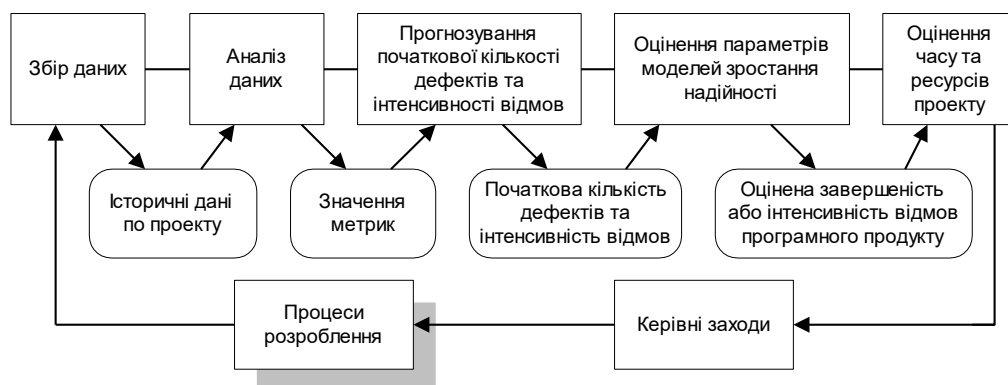


Рис. 1.4. Алгоритм прогнозування показників надійності програмних складових КС

Даним алгоритмом передбачено, що при розробці програмних компонентів КС значення величини показників надійності зазнають змін, а при їх тестуванні відбувається корекція прогнозних значень показників на реальні. Така процедура забезпечує можливість визначення статистичних параметрів щодо оцінок підвищення надійності на протязі усіх етапів ЖЦ імплементації КС.

Перевагами моделі зростання надійності за визначеними відповідними показниками є:

- набори тестових вхідних даних мають невеликий розмір і можуть спричиняти не стійкі чи навіть неможливі величини показників надійності ПЗ. При цьому дані, отримані на основі моделей раннього прогнозування надійності, можна застосовувати при порівнянні чи приведенні до визначених меж прогнозу доповнюючи їх даними, отриманими у результаті тестування;

- у випадку використання ітераційних технологій визначення та обчислення оцінок показників надійності може виникати вплив, який формує залежність щодо початкових оцінок, а при застосуванні апріорних кількісних показників в якості основної початкової оцінки значно скорочується та оптимізується пошук приблизних їх значень;

- для забезпечення планування і розподілу ресурсів при проведенні тестування програмних модулів КС ефективним є застосування історичних даних, які визначаються як апріорні значення надійності.

При виконанні етапів імплементації програмних модулів КС створюється звіт з оцінками показників надійності. Після цього на його основі розраховується міра з якою досягнуто конкретних цілей реалізації КС, критерії щодо оптимальності задіяних у розробку ресурсів, встановлюються недоліки відповідних процесів і формуються рекомендації щодо їх покращення.

У випадку, коли спрогнозовані оцінки показників надійності не задовольняють встановленим вимогам, то це означає, що необхідне втручання та модернізація процесів реалізації програмних модулів.

Практичне використання моделей для прогнозування надійності дає змогу реалізувати процес управління імплементацією і підтримкою КС на основі кількісних показників надійності і спрогнозувати можливий вплив помилок чи відмов у програмних модулях чи компонентах на працездатність КС. Комплекс моделей раннього і пізнього прогнозування надійності дозволяє ефективно керувати ризиками і виконувати балансування проекту за часом виконання, фінансами і критеріями надійності.

1.3. Аналіз типів помилок у програмних складових комп'ютерних систем

Згідно з міжнародною класифікацією, яка використовується для опису типів помилок, які призводять до збоїв у роботі програмного забезпечення визначено три класи помилок.

У загальному випадку під помилкою розуміють ситуацію, яка виникає при роботі програми внаслідок чого отримують недостовірні чи не коректні результати. Така ситуація в першу чергу обумовлена недоліками при програмній реалізації алгоритмів і зазвичай пов'язана з помилками при використанні операторів на етапах проектування і кодування системи.

Під дефектом розуміють такий тип помилки, який одержуються при неякісному написанні програмного коду розробником системи. Варто відмітити, що дефекти також можуть виникати при неточності трактування вимог представлених у специфікації, помилками у проектній і супровідній документації, а також безпосередньо програмним кодом.

Дефекти у більшості випадків виявляють на етапі тестування та супроводу комп'ютерних систем, а це може провокувати збої як в частині програмних модулів, так і системи в цілому.

Під відмовою слід розуміти такий тип помилки, або їх сукупності, що спричиняє перехід системи з працездатного стану до неможливості її використання та функціонування.

Якщо виникає відмова у роботі програмних модулів, то спостерігаються відхилення у її поведінці, відмінні від описаних у документації. Це може бути спровоковано наявністю невиявлених або прихованих дефектів, внесених на етапі проектування, чи проблемами середовища експлуатації [5, 7].

До основних причин виникнення відмов можна віднести:

- недотримання вимог, визначених специфікацією на етапі їх аналізу або неправильне трактування, що спричиняє некоректне їх відображення при проектуванні архітектури чи програмній реалізації та невідповідність очікуванням користувачів;
- несумісність вимог специфікації з вимогами середовища експлуатації чи апаратними пристроями комп'ютерної системи;
- логічні помилки у програмному кодї, які можуть провокувати, як приклад, несанкціонований доступ до інформації чи пристроїв комп'ютерної системи;
- некоректність, недостовірність чи незавершеність імплементації алгоритмів, описаних мовою програмування.

Як висновок, відмови можуть бути спровоковані як одиничними помилками програмного коду, так і набором помилок, а також їх можуть спричинити приховані помилки при проектуванні і програмуванні.

Щодо класифікації помилок у програмних модулях КС, то ІВМ запропонувала підхід ортогональної класифікації дефектів [9]. Даним підходом передбачається, функціями тестувальників КС є виявлення помилок, і їх кластеризація за наперед визначеними групами.

При такій організації, класифікація помилок не містить залежності від виду і застосування програмного компонента, не залежить від способу організації процесу реалізації системи і може бути використана в якості універсальної класифікації. У відповідності до запропонованої ІВМ класифікації, визначають класи помилок і дефектів, які наведено у табл. 1.1.

Таблиця 1.1.

Класи дефектів за ортогональною класифікацією ІВМ

Клас помилки	Опис дефекту
Функція	Помилки щодо відображення даних у інтерфейсах користувачів ПЗ, які пов'язані зі збоями апаратного забезпечення або зовнішніми структурами даних.
Інтерфейс	Помилки методів і засобів інтеграції програмних модулів або систем, механізмів їхньої взаємодії.
Логіка	Логічні помилки у програмно реалізованих алгоритмах опрацювання даних, які не були перевірені, а також неправильна інтерпретація значень змінних .
Присвоювання	Помилки ініціалізації змінних, або не коректне застосування структур представлення даних.
Зациклення	Помилки, пов'язані з продуктивністю програмного забезпечення, які можуть бути викликані неоптимальним розподілом ресурсів.
Середовище	Помилки в репозиторії, у керуванні змінами або в контрольованих версіях проекту
Алгоритм	Помилки, що викликані неефективними алгоритмами опрацювання даних, некоректними структурами даних або помилками в алгоритмах.
Документація	Помилки у документації процесів розробки програмного забезпечення на різних стадіях життєвого циклу

При ортогональній схемі класифікації не допускається приналежність однієї і тієї ж помилки до різних класів, що в свою чергу формує істинність твердження, що помилка завжди відноситься лише до одного класу. У результаті цього, можна сказати, що помилка у програмному модулі повинна міститися лише в одному, програмно реалізованому класі. Це дає змогу забезпечити однозначність та уніфікацію дій розробників у випадку виявлення будь-яких типів помилок.

Альтернативним підходом до класифікації помилок, запропонованої фірмою IBM, є підхід компанії Hewlett–Packard. Базовим при застосуванні цього підходу є класифікація розроблена Г. Бучем, яка дозволяє визначити відповідність для різних типів помилок на етапах реалізації КС. На рис. 1.5 продемонстровано відносний показник помилок за результатами виконання етапів життєвого циклу КС.



Рис. 1.5. Відносні показниками співвідношення між типами помилок ПЗ

Загалом, такий розподіл помилок, який показаний на рис. 1.5, притаманний для більшості корпоративних компаній, однак зараз спостерігаються відмінності для деяких компаній з розробки комп'ютерних систем.

Підхід, який забезпечує фіксацію впливу помилок на відмови описує причинно-наслідковий механізм відношення «помилка-відмова» і дає змогу забезпечувати вибір найбільш ефективних способів та інструментів тестування і моніторингу за станом програмних модулів. Як наслідок це дозволяє знизити рівень негативного впливу збоїв у програмних компонентах на надійність КС.

Відмови програмних складових комп'ютерних систем можна класифікувати також наступним чином:

- відмова апаратного пристрою, яка призводить до втрати працездатності системного ПЗ;
- відмова на інформаційному рівні, яка спричинена помилками введення даних чи при спробі їх передачі;
- відмова на рівні програмного компоненту, яка зумовлена наявністю дефекту і його впливом на комп'ютерну систему;
- відмова на рівні ергономіки, яка спровокована людським фактором, наприклад, некоректним використанням апаратних пристроїв або елементів керування програмного інтерфейсу.

Як видно з результатів проведеного аналізу щодо факторів, які спричиняють, або зумовлюють помилки на різних етапах виконання проектів і впливають на надійність системи, універсального рішення для управління та моніторингу показників надійності комп'ютерних систем поки не запропоновано. Тому актуальним на сьогодні є проведення додаткових досліджень, які б давали змогу виявляти помилки і дефекти як у програмних, так і у апаратних складових КС, а також створення та впровадження процедур встановлення рівня впливу відмов на загальний показник надійності комп'ютерних систем.

1.4. Висновки до розділу

Основні результати даного розділу полягають в наступному:

1. Проведено аналіз моделей надійності компонентів комп'ютерних систем у результаті якого встановлено, що більшість відмов у роботі систем пов'язано з наявністю помилок і дефектів у їхніх програмних складових.

2. Проведено аналіз типів помилок, їх ознак та впливу на надійність комп'ютерної системи, що дало змогу враховувати їх розвиток та обґрунтувати необхідність застосування моделей прогнозування показників надійності на різних стадіях життєвого циклу і спостереження за ними під час експлуатації і супроводу.

3. Проведено аналіз моделей надійності комп'ютерних систем з акцентом на програмне забезпечення і виявлено, що застосування моделей оцінювання надійності технічних засобів є не ефективним для представлення показників надійності програмного забезпечення у зв'язку з різною природою та механізмами відмов.

РОЗДІЛ 2

МОДЕЛЬ ТА ПОКАЗНИКИ НАДІЙНОСТІ КОМП'ЮТЕРНИХ СИСТЕМ

2.1. Визначення і математичний опис критеріїв надійності комп'ютерних систем

Опис критеріїв надійності комп'ютерних систем у кваліфікаційній роботі пропонується виконувати у термінах та згідно із структурою, яка рекомендована стандартом ISO 25010. У відповідності до рекомендацій надійність, як комплексну характеристику можна зобразити у вигляді підхарактеристик, які визначають відмовостійкість системи, її завершеність та здатність до відновлення. На рис. 2.1 проілюстровано залежність між надійністю КС та її характеристиками.

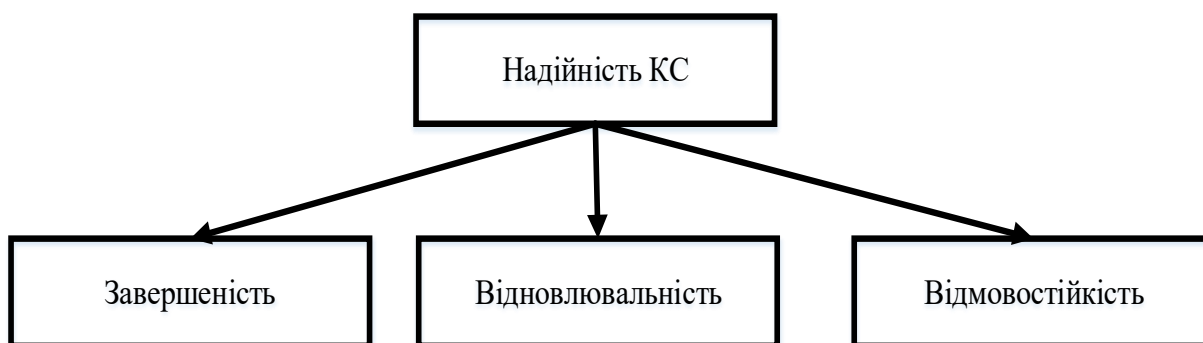


Рис. 2.1. Структура характеристики надійності КС

Згідно означення стандарту, підхарактеристика «Завершеність» відноситься до однієї з найважливіших при визначенні надійності комп'ютерної системи. Суть цієї підхарактеристики полягає у відображенні впливу на систему з точки зору безвідмовності її компонентів, як програмних, так і апаратних та комунікаційних.

«Завершеність» дає змогу описати такі атрибути системи, які дозволяють уникати збоїв через приховані або не виявлені дефекти у процесі проектування системи або її реалізації.

Кількісне представлення характеристики «Надійність» забезпечує множина метрик атрибутів кожної підхарактеристики, яка представлена на рис. 2.1. Згідно означення стандарту [2] метрика представляє собою метод і шкалу вимірювання.

Для різних стадій розробки комп'ютерної системи можна використовувати різні метрики, які дають змогу кількісно виразити надійність на зовнішньому і внутрішньому рівні, а також з точки зору експлуатації системи.

Для того, щоб виміряти надійність кожного компоненту комп'ютерної системи застосовуються внутрішні метрики. Якщо комп'ютерна система функціонує у середовищі розробників, то для цього доцільно скористатися набором зовнішніх метрик. Надійність на рівні експлуатації у середовищі користувача визначається експлуатаційними метриками.

Існує вплив і залежність показників надійності між собою, а також на загальну надійність комп'ютерної системи. На рис. 2.2 продемонстровано вплив на різних рівнях визначення надійності комп'ютерної системи.

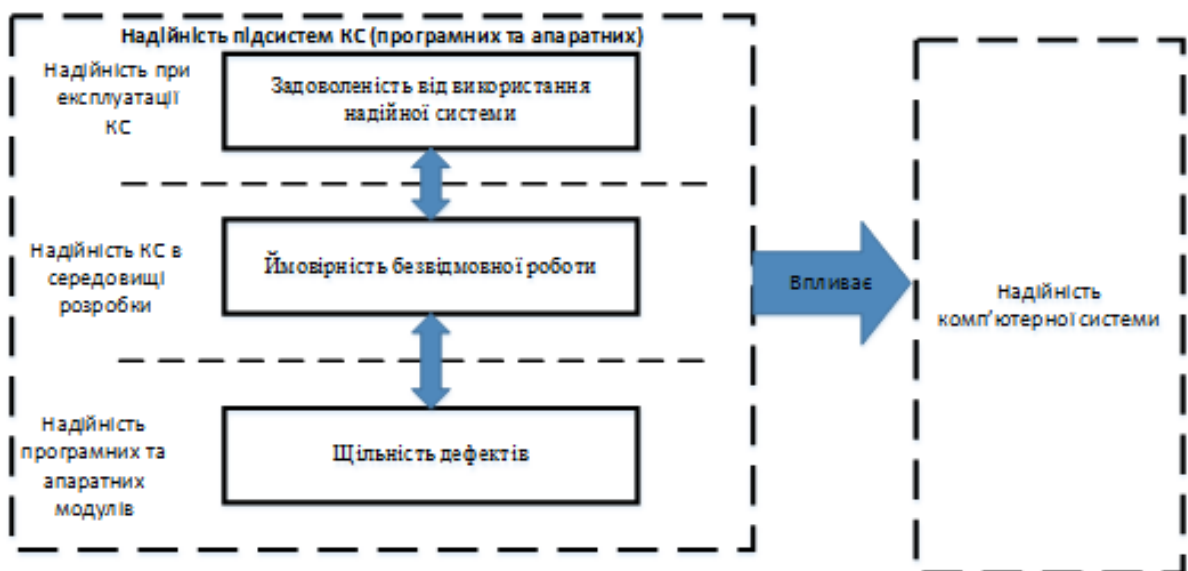


Рис. 2.2. Взаємозв'язок між представленням рівнів надійності

Метрики підхарактеристики «Завершеність» можна описати атрибутами і метриками, які, для прикладу, визначають завершеність програмних компонентів системи:

- частота появи (інтенсивність) відмов ;
- частота (інтенсивність) ліквідації виявлених дефектів;
- точність виявлення помилок при проведенні перевірок.

Для кількісного вираження інтенсивності появи дефектів у компонентах комп'ютерної системи доцільно використовувати формулу (2.1)

$$Def^{int} = \frac{Def^{find}}{Def^{pend}} \quad (2.1)$$

Def^{int} – атрибут і метрика, що виражає інтенсивність виявлення помилок;

Def^{find} – кількість визначених помилок або дефектів;

Def^{pend} – кількість помилок, які очікують розробники.

Формально для опису інтенсивності виправлення та ліквідації помилок чи дефектів у КС можна скористатися прямим показником їх кількості, або ж відношенням кількості ліквідованих помилок до кількості визначених під час проведення перевірок

$$Def^{likv} = n \quad (2.2)$$

Def^{likv} – інтенсивність виправлення помилок;

n – кількість виправлених помилок.

Для випадку рівності значень виявлених та ліквідованих помилок застосовується наступна рівність

$$Def^{find} = Def^{likv} \quad (2.3)$$

Для представлення метрики виправлення помилок, або інтенсивності їх усунення на внутрішньому рівні застосовується відношення наступного вигляду:

$$Def^{likv} = \frac{n}{Def^{find}} \quad (2.4)$$

З формули (2.4) видно, що при значенні Def^{likv} , яке прямує до 1, кількість помилок і дефектів починає зменшуватися.

Для формального опису атрибуту точності перевірок застосовується відношення кількості класів помилок, перевірку яких планується виконати, до кількості класів помилок, які треба виконати для забезпечення повноти перевірки

$$Check^{accur} = \frac{Cat^{plan}}{Cat^{need}} \quad (2.5)$$

$Check^{accur}$ – атрибут з метрикою точності перевірок;

Cat^{plan} – кількість класів помилок, які були запланованими для перевірки;

Cat^{need} – кількість перевірок, які потрібно провести, щоб забезпечити повноту визначення помилок.

Зовнішніми атрибутами і метриками характеристики «Завершеність» в контексті програмних складових комп'ютерних систем є:

- щільність очевидно не виявлених помилок;
- щільність відмов у роботі комп'ютерної системи;
- щільність помилок неякісної реалізації компоненті комп'ютерної системи;
- інтенсивність з якою усуваються помилки, допущені розробниками КС;
- середній час між збоями (відмовами працездатності) у роботі компонентів комп'ютерної системи.

Щільність очевидно не виявлених помилок можна обчислити шляхом розрахунку відношення загальної кількості очікуваних помилок за винятком визначених помилок під час перевірок та абсолютним значенням кількості компонентів комп'ютерної системи, тобто розміру КС

$$Density^{Def^{Pend}} = \frac{|Def^{pend} - Def^{find}|}{V} \quad (2.6)$$

$Density^{Def^{Pend}}$ – щільність очевидно не виявлених помилок;

Def^{pend} – кількість помилок, які очікуються;

Def^{find} – кількість фактичних визначених помилок під час перевірок;

V – кількість складових комп'ютерної системи, наприклад, програмних.

Щільність відмов у роботі комп'ютерної системи визначається за допомогою відношення: кількість відмов за деякий час до кількості проведених перевірок

$$Density^{Failure} = \frac{|Failure|}{|Tests|} \quad (2.7)$$

$Density^{Failure}$ – щільність збоїв у роботі КС за деякий тестовий інтервал часу;

$|Failure|$ – кількість збоїв протягом тестового періоду часу;

$|Tests|$ – кількість перевірок, які виконано за тестовий період часу.

На зовнішньому рівні атрибут та метрика щільності помилок визначається у вигляді відношення загальної кількості визначених помилок або збоїв у роботі комп'ютерної системи до кількості компонентів системи

$$Density^{Def} = \frac{|Def^{find}|}{V} \quad (2.8)$$

$Density^{Def}$ – щільність помилок;

Def^{find} – кількість встановлених помилок дефектів при проведенні перевірок;

V – кількість програмних та апаратних складових системи.

На зовнішньому рівні, інтенсивність з якою усуваються помилки, допущені розробниками КС, можна розрахувати за допомогою наступного відношення:

$$Def_{int}^{ext} = \frac{|Def^{change}|}{|Def^{find}|} \quad (2.9)$$

Def_{int}^{ext} – інтенсивність помилок зовнішнього рівня;

$|Def^{change}|$ – кількість усунутих помилок;

$|Def^{find}|$ – кількість визначених помилок.

Середній час між збоями або відмовами у працездатності комп'ютерної системи розраховується як відношення інтервалу часу

нормальної працездатності системи до кількості зафіксованих збоїв та відмов. Інша метрика для кількісного вираження цього показника може бути обчислена у як відношення суми тривалості безперебійної роботи системи до кількості встановлених протягом цього інтервалу часу помилок

$$\begin{aligned} Avg_{time}^{Failure} &= \frac{T^{func}}{|Failure|} \\ Avg_{time}^{Failure} &= \frac{\sum T^{func}}{|Failure|} \end{aligned} \quad (2.10)$$

$Avg_{time}^{Failure}$ – середній час між збоями у працездатності системи;

$|Failure|$ – кількість збоїв протягом деякого інтервалу часу;

T^{func} – час, протягом якого система перебуває у стані працездатності.

Експертні технології є ефективним інструментом для забезпечення кількісного представлення підхарактеристики завершеність на рівні експлуатації системи, тобто з точки зору користувача системи у його середовищі використання.

Однак, для встановлення міри впливу та аналізу помилок і збоїв у роботі комп'ютерної системи важливо побудувати модель представлення надійності за її структурними компонентами. Особливо важливу роль при цьому відіграє програмне забезпечення, що забезпечує процес управління апаратними складовими.

2.2. Модель надійності за наявності збоїв у ПЗ

Типова комп'ютерна система складається з апаратних компонентів, програмного забезпечення на рівні системного і прикладного опрацювання даних, а також комунікаційних каналів взаємодії. Тому модель системи можна представити у наступному вигляді:

$$CompSyst = \{ HW, SW, Chan \} \quad (2.11)$$

CompSyst - конкретна КС;

HW – множина апаратних складових системи;

SW – набір ПЗ управління;

Chan – множина каналів обміну даними.

Загалом, надійність на надійність комп'ютерної системи впливає надійність її структурних елементів, що може бути виражена у вигляді зваженої суми показників у формулі (2.11) .

Надійність компонентів ПЗ КС пропонується описати таким виразом:

$$R(SW) = \{ SW_i, Def_{ij} \} \quad (2.12)$$

$R(SW)$ – надійність програмних компонентів КС;

SW_i – конкретний компонент ПЗ, $i = 1..k$, k – кількість компонентів;

Def_{ij} – помилка, яка виникла при функціонуванні i -го компонента ПЗ, $j = 1..m$, m – кількість помилок.

$$R(HW) = \{ HW_i, Def_{ij} \} \quad (2.13)$$

$R(HW)$ – надійність апаратних пристроїв КС;

HW_i – конкретний апаратний пристрій, $i = 1..l$, l – кількість апаратних пристроїв;

Def_{ij} – помилка, яка виникла при функціонуванні конкретного апаратного пристрою $j = 1..p$, p – кількість помилок.

Враховуючи те, що архітектура комп'ютерної системи передбачає побудову зв'язків-відношень між її компонентами як на рівні програмного,

так і апаратного забезпечення, то для формального їх відображення, а також можливості застосування метрики зчеплення між програмними модулями необхідно формалізувати таке відношення:

$$Rel_{sw} = \{ SW_i, SW_j \} \quad (2.14)$$

SW_i – деякий компонент або модуль ПЗ;

SW_j – компонент ПЗ, відмінний від SW_i та взаємодіє з ним,
 $i, j \in K, i \neq j$.

Представлення зв'язку між апаратними пристроями комп'ютерної системи можна зобразити аналогічно до формули (2.14)

$$Rel_{hw} = \{ HW_i, HW_j \} \quad (2.15)$$

HW_i – деякий апаратний пристрій комп'ютерної системи;

HW_j – пристрій відмінний від HW_i , але який перебуває у взаємодії з ним,
 $i, j \in L, i \neq j$.

Зв'язки між апаратними пристроями і програмним забезпеченням КС пропонується записати у наступному вигляді:

$$Rel_{hw}^{sw} = \{ \{ SW_i, HW_j \}, Chan_s \} \quad (2.16)$$

Rel_{hw}^{sw} – сукупність зв'язків між програмними модулями і апаратними пристроями КС;

$\{ SW_i, HW_j \}$ – кортеж «програмний модуль-фізичний пристрій», між якими відбувається комунікація, $i \in K, j \in L$;

$Chan_s$ – канал передачі і прийому даних при взаємодії програмних модулів та апаратних пристроїв, $s = 1..S$, S – кількість каналів взаємодії.

Враховуючи запропонований формалізм структурних компонентів комп'ютерної системи, можна побудувати ланцюг комунікації між програмними модулями та апаратними пристроями, що дає можливість забезпечити моніторинг помилок та визначити причини, а також вплив помилок на надійність комп'ютерної системи.

Одним із шляхів підвищення надійності КС, який пропонується у кваліфікаційній роботі, є мінімізація зчеплення між програмними модулями та зв'язків апаратного і програмного забезпечення на основі парадигми функційного програмування. Далі потрібно обґрунтувати модель, для представлення помилок у комп'ютерній системі та розробити відповідний метод для їх моніторингу.

2.3. Обґрунтування моделі представлення помилок у програмному забезпеченні комп'ютерної системи

Модель представлення помилки ПЗ комп'ютерної системи повинна забезпечувати опис компонента у якому виник дефект.

Нехай $SW = \{ sw_1, sw_2, sw_3, \dots, sw_k \}$ – сукупність програмних складових КС. Множина помилок у конструктивних елементах ПЗ представляється як сукупність $Def = \{ def_1, def_2, def_3, \dots, def_p \}$, що проявляються у структурних компонентах ПЗ. Виходячи з цього, модель помилки можна зобразити як кортеж функцій [20]

$$\langle Func_{Def}, Avg_{Def} \rangle \quad (2.17)$$

$Func_{Def}$ – функція, що визначає рівень розвитку помилки;

Avg_{Def} – функція, яка інтерпретує середню інтенсивність ознак помилки.

Для того, щоб побудувати модель помилки ПЗ необхідно виконати кроки наступного алгоритму:

1. Задати правила для опису конструктивних елементів ПЗ. Множина правил $Rules = \{r_1, r_2, r_3, \dots, r_p\}$ формується на основі документів специфікації мови, технології і парадигм програмування. Помилка у конструктивному елементі ПЗ інтерпретується як порушення правила. Нехай $def \in Def$ – помилка ПЗ, а $r \in Rules$ – правило, що описує вимоги до конструкції елемента, тоді $def = violate(r)$ вказує на факт не дотримання правила r і спричиняє виникнення дефекту def [20].

2. Встановити ознаки $S_d = \{s_{d,1}, s_{d,2}, s_{d,3}, \dots, s_{d,m}\}$, які вказують на недотримання правила при проведенні аналізу конструктивного елемента ПЗ.

3. З набору доступних метрик вибрати таку, яка дає змогу визначити інтенсивність ознак помилок ПЗ у комп'ютерній системі. Така ознака може бути елементарною або складатися з інших ознак. Під елементарною ознакою потрібно розуміти таку ознаку, інтенсивність якої вимірюється на основі прямої метрики [15]. Кожній елементарній ознаці $s \in S$ ставиться у відповідність метрика $M_s : E \rightarrow Q$, Q – область допустимих значень.

4. Встановити граничну межу метрики. Граничне значення $Z_s \in Q$ дає змогу трактувати результат вимірювання значення ознаки в контексті її розвитку.

5. Визначити $Func_{def}$ та Avg_{def} . Для того, щоб оцінити інтенсивність елементарної ознаки $s \in S$ з врахуванням типу границі, можна скористатися наступними формулами:

– якщо границя відповідає верхній межі:

$$I_s(sw) = \text{HigherThan}(M_s(sw), Z_s) = \frac{M_s(sw)}{Z_s} \cdot 100 \quad (2.18)$$

– якщо границя відповідає нижній межі:

$$I_s(sw) = \text{LowerThan}(M_s(sw), Z_s) = \frac{Z_s}{M_s(sw)} \cdot 100 \quad (2.19)$$

При розрахунку інтенсивності складених ознак пропонується скористатися агрегатними функціями:

– якщо складена ознака $s \in S$ одночасно проявляє елементарні ознаки $s_1, s_2 \in S$, згідно з [15], то можна записати:

$$I_s(e) = \min(I_{s_1}(e), I_{s_2}(e)) \quad (2.20)$$

– якщо у складеній ознаці $s \in S$ проявляється хоча б одна елементарна $s_1, s_2 \in S$, то (2.20) трансформується у

$$I_s(e) = \max(I_{s_1}(e), I_{s_2}(e)) \quad (2.21)$$

Будь-яку модель, в тому числі і моделі складових комп'ютерної системи, можна представити як сукупність сутностей, їх властивостей та відношень між ними.

В якості моделі запропоновано скористатися За основу пропонується взяти модель [20], що передбачає застосування моделі «Режим історії недоліків проектування» (англ. DDMH), яка є доповненням до моделі на мета рівні – Nismo. Структура базової моделі Nismo представлена на рис. 2.3. Важливою перевагою цієї моделі є те, що вона забезпечує

можливість аналізу історичних даних при виявленні та виправленні помилок ПЗ.

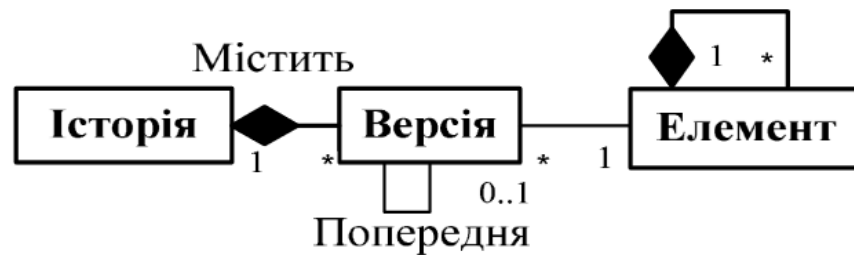


Рис. 2.3. Базова структура мета моделі Nismo

Модель Nismo включає:

- елемент, який підлягає заміні іншим конструктивним елементом, поміщається в історію і досліджується;
- версія конструктивного елемента ПЗ передбачає додавання мітки часу до його мета опису, при цьому переміщаючи його в історію; версія елемента однозначно визначається відповідною міткою та унікальним ідентифікатором історії, що сформований в тому числі і з мітки часу елемента;
- до історії конструктивного елемента входить набір версій, однак при цьому дотримуються правила, що одна версія належить тільки одній історії, через це відношення історії і версії представляється їхньою композицією.

Модель «Режим історії недоліків проектування» будується шляхом доповнення типової моделі Nismo елементами, які містять опис версії елемента та історію помилок.

Тоді таку доповнену модель можна графічно зобразити, як показано на рис. 2.4.

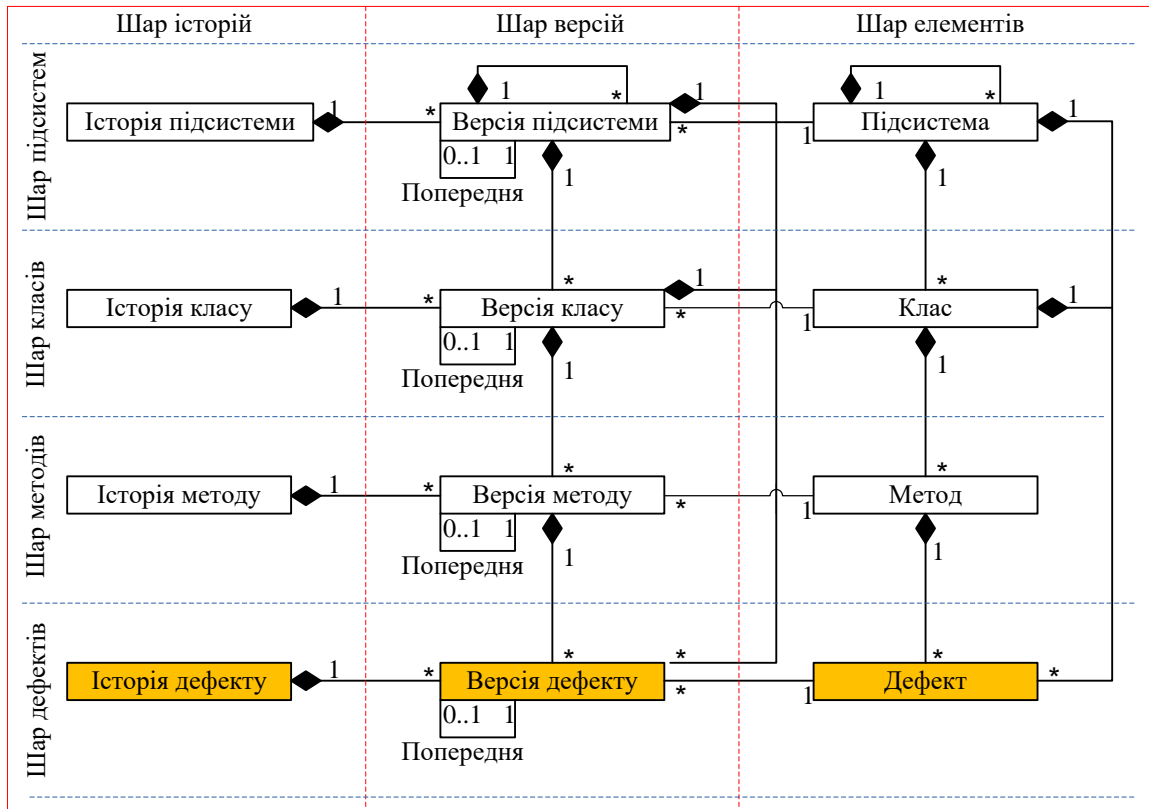


Рис. 2.4. Мета модель історії дефектів проектування

Як видно з рис. 2.4 у вертикальні шари моделі включено узагальнюючі сутності, зокрема, конструктивні елементи, їх версія, історія. До горизонтальних шарів включено об'єкти, які представляють конструктивні елементи ПЗ, яке моделюється разом з їхніми версіями та історією змін.

Особливістю представленої на рис. 2.4 моделі є те, що окрім версій та історій конструктивних елементів ПЗ до неї також входять версії й історії помилок внесених розробниками.

Якщо говорити формально, то модель DDHM представляє собою орієнтований мультиграф:

$$MModel = \langle N, E \rangle, \quad (2.22)$$

де $N = \{n_1, n_2, n_3, \dots, n_m\}$ – представляють вузли опису сутностей;

$E = \{e_1, e_2, e_3, \dots, e_k\}$ – ребра, які задають зв'язок між сутностями.

Для мультиграфа характерним є те, що зв'язок між двома вершинами може бути організований за допомогою кількох ребер. Для того, щоб задати модель застосовується типізація. Тип ребра «*contains*» визначає вкладеність вершин графа. Те що мультиграф є орієнтованим, вказує на те, що кожне ребро належить лише до одного вузла – джерела і одного вузла – цілі.

Навантаженість мультиграфа визначає властивість що для кожного вузла чи ребра може бути означена будь-яка кількість властивостей.

Нехай η – сукупність усіх типів вершин, η_i – довільна вершина з набору представленого у табл. 2.1, ρ – сукупність усіх властивостей вершин, ρ_i – будь-яка властивість вершини, а ε – набір усіх типів з'єднань вершин, і ε_i – будь-який тип з'єднання, який визначений у табл. 2.2.

Для прикладу, «*system*» - вершина мультиграфа, яка представляє собою комплексну комп'ютерну систему, «*method*» - вершина. Яка інтерпретує функцію описану в класі, тобто його метод. Тоді процедуру композиції чи агрегації, яка відображається стрілкою від контейнера до вмісту, формує ребро графу або по іншому «*contains*».

Таблиця 2.1

Типи вузлів моделі

Вид вузла	Опис
<i>system, systemV, systemH</i>	Програмна система, її версія та історія
<i>pack, packV, packH</i>	Пакет, його версія та історія
<i>class, classV, classH</i>	Клас, його версія та історія
<i>method, methodV, method</i>	Метод, його версія та історія
<i>attribute, attributeV, attributeH</i>	Атрибут класу, його версія та історія
<i>defect, defectV, defectH</i>	Дефект, його версія та історія

Таблиця 2.2

Типи ребер

Тип ребра	Опис	Можливі пари типів <джерело, мета>
<i>contains</i>	Джерело містить мету	$\langle \text{system}, \text{pack} \rangle$, $\langle \text{system}, \text{class} \rangle$, $\langle \text{pack}, \text{pack} \rangle$, $\langle \text{pack}, \text{class} \rangle$, $\langle \text{class}, \text{method} \rangle$, $\langle \text{class}, \text{attribute} \rangle$, $\langle \text{pack}, \text{defect} \rangle$, $\langle \text{class}, \text{defect} \rangle$, $\langle \text{method}, \text{defect} \rangle$, $\langle \text{packV}, \text{packV} \rangle$, $\langle \text{packV}, \text{classV} \rangle$, $\langle \text{classV}, \text{methodV} \rangle$, $\langle \text{classV}, \text{attributeV} \rangle$, $\langle \text{packV}, \text{defectV} \rangle$, $\langle \text{classV}, \text{defectV} \rangle$, $\langle \text{methodV}, \text{defectV} \rangle$, $\langle \text{packH}, \text{packV} \rangle$, $\langle \text{classH}, \text{classV} \rangle$, $\langle \text{methodH}, \text{methodV} \rangle$ $\langle \text{attributeH}, \text{attributeV} \rangle$, $\langle \text{defectH}, \text{defectV} \rangle$, $\langle \text{systemH}, \text{systemV} \rangle$, $\langle \text{systemV}, \text{packV} \rangle$
<i>version</i>	Джерело є версією мети	$\langle \text{packV}, \text{pack} \rangle$, $\langle \text{classV}, \text{class} \rangle$, $\langle \text{methodV}, \text{method} \rangle$, $\langle \text{attributeV}, \text{attribute} \rangle$ $\langle \text{defectV}, \text{defect} \rangle$, $\langle \text{systemV}, \text{system} \rangle$
<i>prev</i>	Джерело є версією, наступною після версії-мети у довільній історії	$\langle \text{packV}, \text{packV} \rangle$, $\langle \text{classV}, \text{classV} \rangle$, $\langle \text{methodV}, \text{methodV} \rangle$, $\langle \text{defectV}, \text{defectV} \rangle$, $\langle \text{systemV}, \text{systemV} \rangle$
<i>call</i>	Джерело викликає мету	$\langle \text{method}, \text{method} \rangle$, $\langle \text{methodV}, \text{methodV} \rangle$
<i>inherit</i>	Мета є суперкласом історії	$\langle \text{class}, \text{class} \rangle$ $\langle \text{classV}, \text{classV} \rangle$

Для того, щоб позначити ребро графа, яке протилежно напрямлене, використовується стрілка над його назвою. Як приклад, $\overleftarrow{\text{contains}}$ – зв'язок включення, що є ребром напрямленим від вмісту до контейнера. Враховуючи те, що мультиграф є навантаженим, то для будь-якої вершини вузла чи ребра можна додавати спеціальні властивості.

Приклади можливих застосувань властивостей вершин представлено у табл. 2.3.

Таблиця 2.3

Властивості вузлів

Тип вузла	Властивість	Опис
<i>systemV, packV</i> <i>classV, methodV,</i> <i>defectV</i>	<i>date</i> <i>rank</i>	Дата створення версії Номер версії в історії
Тип вузла	Властивість	Опис
<i>defectV</i>	<i>dpd</i> <i>msi</i> <i>defType</i>	Ступінь розвитку дефекту (defect progress degree) Середня інтенсивність ознак дефекту (mean sign intencity) Тип дефекту (defect type)

Функції, представлені у табл. 2.4, використовуються для забезпечення сигнатури, тобто виконання операцій над моделлю.

Таблиця 2.4

Функції маніпулювання над моделлю

Функція	Опис
$source: E \rightarrow N$	Для отримання вузла, джерела ребра
$target: E \rightarrow N$	Для отримання вузла, мети ребра
$type: N \rightarrow \eta$	Для отримання типу вузла
$type: E \rightarrow \varepsilon$	Для отримання типу ребра
$property: N, \rho \rightarrow PROPV$	$PROPV$ – множина значень властивостей вузла

Множина V , наведена нижче, застосовується також для роботи з моделлю DDHM

$$V = \{n \in N \mid type(n) \in \{packV, classV, methodV\}\}, \quad (2.8)$$

$$DT = \{x/x - \text{тип помилки}\}$$

Тоді, предикат $isEdge: E \times N \times \varepsilon \times N \rightarrow Boolean$, що виконує перевірку наявності ε_1 – зв'язок між вершинами n_1 і n_2 , де $n_1, n_2 \in N$, $\varepsilon_1 \in \varepsilon$ буде представлений як:

$$isEdge(e, n_1, \varepsilon_1, n_2) : source(e) = n_1 \wedge target(e) = n_2 \wedge target(e) = \varepsilon_1, \quad (2.23)$$

Функція $Dpd: V \times D \rightarrow [0,1000]$ визначає рівень розвитку конкретної помилки $d \in D$, яка включена у версію конструктивного елемента $v \in V$

$$Dpd(v, dt) = \begin{cases} property(dv, dpd), \text{ якщо} \\ (\exists dv \in \{n \in N \mid type(n) = defecV \wedge \\ \wedge property(dv, defType = dt)\} \\ (\exists e \in E isEdge(e, v, contains, dv)); \\ 0, \text{ в інших випадках} \end{cases} \quad (2.24)$$

Функція $Maxdpd: V \rightarrow [0,1000]$ обчислення рівня розвитку найбільше розвинутої помилки програмно може бути визначена, як показано на рис. 2.5.

Визначальну роль у процесі моніторингу за розвитком помилок відіграє візуалізація, яка дає змогу інтерпретувати рівень її розвитку.

```

procedure Maxdpd(v)
  Maxdpd(v):=0
  for each
     $dv \in \{n \in N \mid \text{type}(n) = \text{defec}V \wedge$ 
     $\wedge \exists e \in E \text{ isEdge}(e, v, \text{contains}n)\}$ 
  do
    if Maxdpd(v) < property(dv, dvd) do
      Maxdpd(v):= property(dv, dvd)
    end if
  end for
end procedure

```

Рис. 2.5. Процедура обчислення функції *Maxdpd*

Запропонована модель і метод візуалізації помилок дозволяють графічно відобразити помилки у конструктивних елементах програмних складових комп'ютерних систем на різних рівнях абстракції. Це стосується представлення помилок на рівні методу, на рівні класу, а також на рівні підсистем. Окрім цього існує можливість аналізу історичних аспектів розпаду ПЗ, розвитку помилок у процесі проектування, розвитку як елементарних, так і складених ознак помилки проектування.

2.4. Висновки до розділу

Основні результати даного розділу полягають в наступному:

1. На основі результатів аналізу моделей раннього і пізнього прогнозування помилок програмного забезпечення обґрунтовано їх застосування при проектуванні комп'ютерних систем, що дає змогу побудувати комплексний підхід до визначення впливу дефектів програмного забезпечення на надійність експлуатації комп'ютерних систем з врахуванням історії виникнення помилок у ПЗ.

2. Побудовано та обґрунтовано для застосування модель представлення показників надійності, що враховує концептуальну

структуру комп'ютерних систем і дає можливість на основі аналізу помилок забезпечувати їх моніторинг та приймати управлінські рішення для усунення загрози відмови чи збою у роботі компонентів і системи в цілому.

3. Набув подальшого розвитку метод моніторингу дефектів програмних модулів комп'ютерних систем на основі моделі DDHM, що дає змогу проводити спостереження за його розвитком та візуалізувати вплив на надійність комп'ютерної системи у часі з врахуванням версій програмних модулів системи.

РОЗДІЛ 3

ЗАСІБ АВТОМАТИЗОВАНОГО ВИЯВЛЕННЯ ПОМИЛОК У
ПРОГРАМНИХ МОДУЛЯХ КОМП'ЮТЕРНОЇ СИСТЕМИ3.1. Побудова архітектури засобу автоматизованого управління та
моніторингу показників надійності комп'ютерних систем

Архітектура системи автоматизації процесів управління та моніторингу надійності комп'ютерних систем повинна забезпечувати діагностику програмних компонентів. Однак перед безпосереднім проектуванням її структури потрібно навести розширену структуру метамоделі DDHM, яка проілюстрована на рис. 3.1.

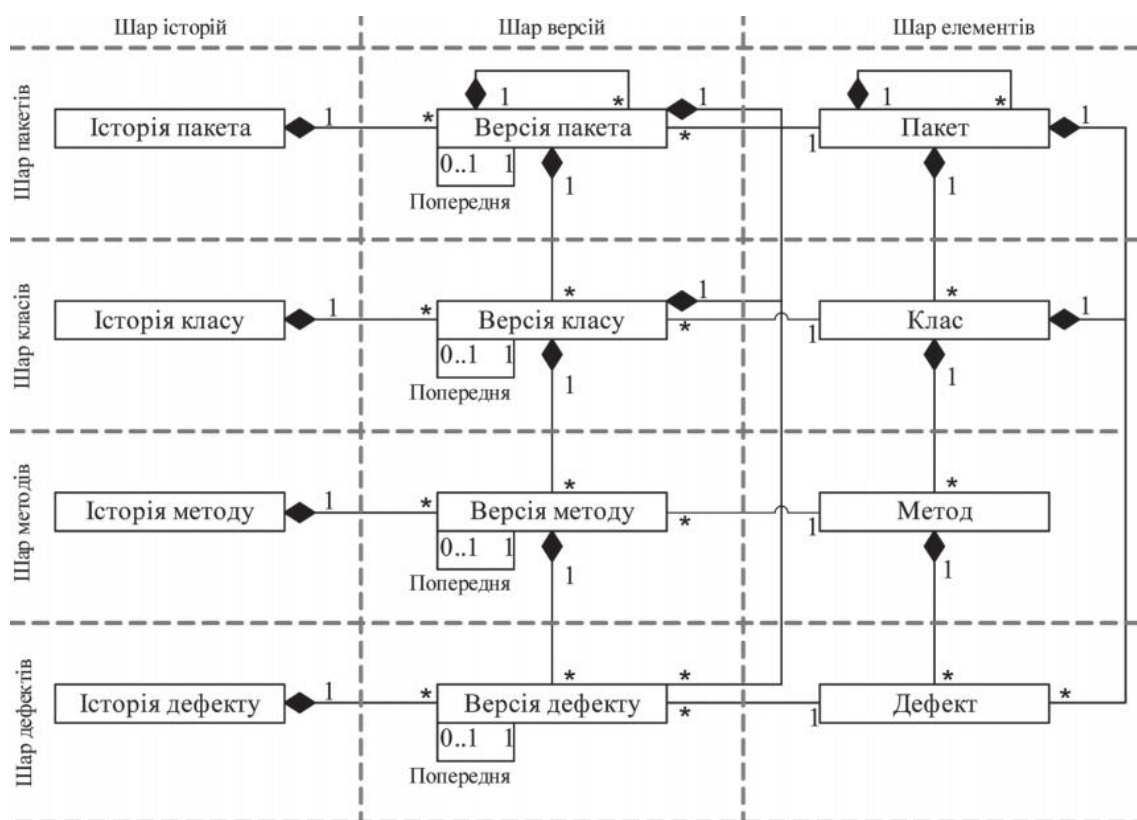


Рис. 3.1. Представлення моделі DDHM у вигляді шарів

При такій організації метамоделі вертикальні шари представляють собою набір сутностей, які є екземплярами більш загальних об'єктів у

моделі HISMO [12]. У горизонтальних шарах представлено набір сутностей, які відображають компоненти моделі програмних модулів, їх версії та історичні дані.

Вертикальні шари метамоделі складаються з таких рівнів:

- рівень програмних елементів – сформований з об'єктів, які інтерпретують конструктивні елементи програмних модулів для дослідження їх історії;
- рівень версій конструктивних елементів – відображає наявні версії конструкцій програмних елементів;
- рівень історії – включає об'єкти метамоделей, які представляють історію конструктивних програмних елементів .

До складу горизонтальних рівнів метамоделі включено рівень пакетів класів (підсистем), рівень класів в термінах об'єктно-орієнтованого програмування, рівень функцій описаних у класі та рівень дектів або помилок.

Аналізуючи структуру моделі, представленої на рис. 3.1, побудовано структуру програмного засобу для автоматизації процедур виявлення, моніторингу та управління розвитком помилок програмних модулів у комп'ютерних системах. Графічне представлення такої архітектури наведено на рис. 3.2.

Конструктивні модулі архітектури дають можливість забезпечити процес моніторингу помилок в елементах програмних модулів на різних абстрактних рівнях та аспектах із застосуванням таких функцій, як:

- побудова моделі програмного модуля, який необхідно діагностувати;
- формування структури моделі (ей) помилок при проектуванні програмного модуля;
- добування екземпляра моделі (рис. 3.1) на основі аналізу програмного коду, який підлягає діагностуванню;
- зберігання параметрів екземпляра моделі у сховищі даних;

- візуальне представлення різних аспектів процесу моніторингу;
- інтерактивність користувачів при роботі з графічним представленням помилок при їх моніторингу;
- формування структури звітної інформації щодо результатів діагностики.

Компоненти засобу автоматизації процесів управління та моніторингу показників надійності комп'ютерних систем можуть функціонувати у двох основних режимах:

- акумулювання та аналіз інформації – проводяться процедури аналізу вихідного коду досліджуваного програмного модуля, виконується розрахунок значень метрик та побудова моделі DDHM, результати якої записується у БД;

- діагностика – забезпечує для користувача можливість спостереження за помилками, допущеними при проектуванні системи на основі дослідження візуального представлення розвитку помилок.

Реалізація вище перелічених функцій можлива при програмній імплементації запропонованої архітектури, яка зображена на рис. 3.2.

Представлення і реалізацію архітектури виконано за допомогою шарів Фаулера, для забезпечення гнучкості, можливості міграції та масштабування компонентів засобу автоматизації. В архітектурі визначено наступні рівні:

- рівень подання (представлення) – орієнтований на забезпечення інтерактивної взаємодії користувачів при введенні і виведенні інформації з підсистем рівня домену;

- рівень домену – представляє підсистеми, які забезпечують функціональність щодо виявлення, управління та моніторингу показників надійності, а також управління розвитком помилок;

- рівень доступу до даних – фізичний рівень для зберігання даних, необхідних для двох верхніх рівнів.

Рівень доступу до даних складається з єдиної підсистеми, що забезпечує зв'язок на рівні фізичної БД з логічними та функціональними підсистемами виявлення та моніторингу помилок у програмних модулях. Рівень подання містить також одну підсистему, яка представляє собою користувацький інтерфейс і забезпечує відповідні елементи керування для взаємодії із системою.

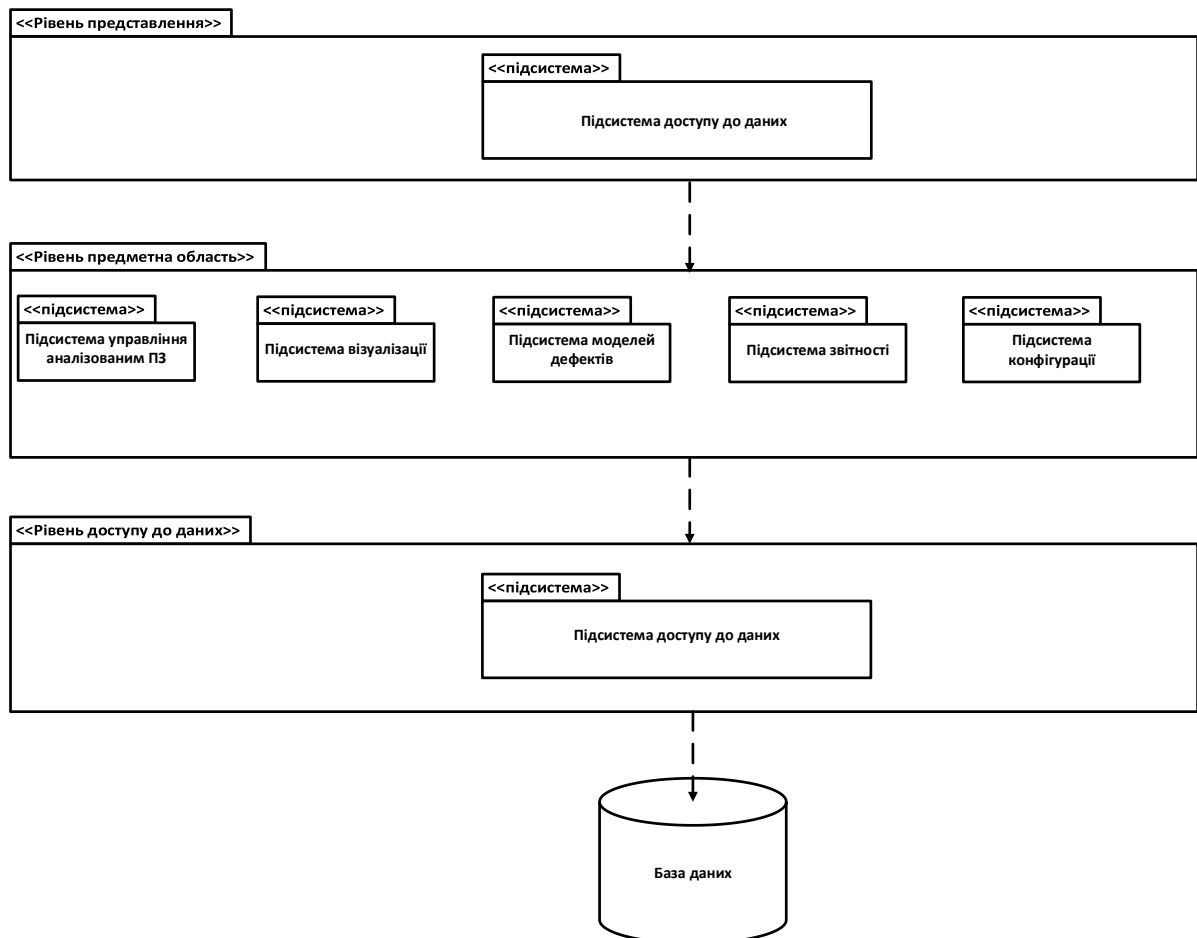


Рис. 3.2. Архітектура програмного комплексу підтримки методу виявлення та контролю дефектів проектування ПЗ

До складу підсистем рівня домену входять:

- підсистема завантаження коду діагностованого програмного модуля, що дає можливість аналізувати моделі, їхні версії та конструктивні елементи;

- підсистема візуального представлення, основна функція якої полягає у побудові графіків з можливістю використання функцій масштабування, переходу та ін.;

- підсистема моделей помилок, яка дає можливість заповнювати відповідні довідники і налаштовувати моделі помилок;

- підсистема формування звітів – дає змогу забезпечити налаштування параметрів при формуванні звітної документації щодо результатів діагностики програмного модуля КС;

- підсистема конфігурації орієнтована на ввід параметрів, які визначають режим роботи засобу автоматизації.

Добуті екземпляри моделі, результати вимірювань, параметри моделей помилок, конфігураційні параметри і службові дані зберігається у сховищі даних.

3.2. Підсистеми на рівні домену

Властивість інкапсуляції в об'єктно-орієнтованій парадигмі програмування забезпечує приватність та розділення доступу до класів та їх екземплярів. При розділенні рівня інтерфейсів та логічного рівня використано патерн проектування «Фасад» [13]. У даному випадку цей патерн забезпечує простоту доступу до кожної з підсистем.

Підсистема керування опрацьовує запити від користувацького інтерфейсу і дає можливість організувати сумісне функціонування компонентів на рівні кожної з підсистем.

На підсистему керування покладено виконання наступних функцій:

- вставки, оновлення, знищення моделей програмних модулів за якими здійснюється моніторинг;

- вставки, оновлення, знищення версій програмних модулів, які діагностуються;

- перегляду і навігації по конструктивних елементах у кожній версії програмного модуля;
- пошуку за визначеними критеріями.

Виконання описаних вище задач забезпечують компоненти підсистем, структуру яких представлено на рис. 3.3

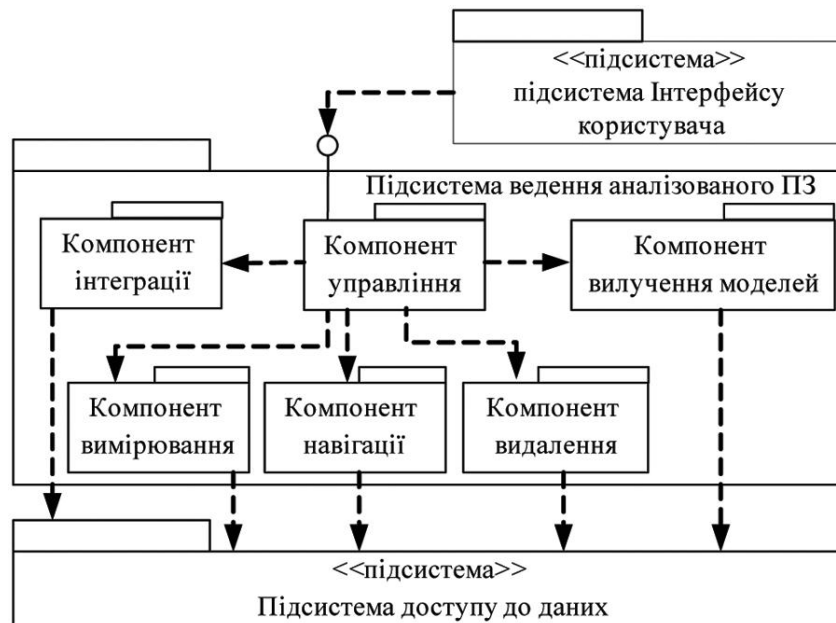


Рис. 3.3. Підсистема управління програмним забезпеченням, що діагностується

Вхідними даними для компоненту добування моделей виступає стрічка у якій вказано шлях до вихідних кодів конкретної версії програмного модуля, який підлягає аналізу. Після цього, відбувається зчитування коду, проводиться його аналіз і формується екземпляр метамоделі FAMIX. В якості ядра компонента добування моделей помилок застосовується iPlasma [11]. Модель зберігається у файлі з форматом MSE.

Підсистема інтеграції може бути викликана підсистемою управління після того, як створений файл з моделлю стає доступним для використання після опрацювання у підсистемі добування моделей. Наступним кроком є передача файлу на вхід до підсистеми інтеграції, що виконує над ним таку послідовність кроків:

- завантаження моделі, що зберігається у файлі, до сховища даних;
- проведення аналізу і пошуку сутностей з врахуванням їх походження на основі моделі, яка описує версію програмного модуля;
- побудова відношення між ідентичними сутностями, які уже збережені у сховищі.

Ідентичні сутності – це сутності, які не відрізняються між собою протягом часу формування кількох історій та версій програмного модуля комп’ютерної системи [13]. Підсистема для проведення вимірювань в якості вхідного аргументу отримує унікальний ідентифікатор системи і проводить розрахунок кількісних показників на основі збереженого у сховищі екземпляра моделі DDHM і моделі помилок. Атрибути об’єктів моделі змінюються на основі результатів аналізу та записуються у сховище даних. Далі виконується розрахунок мір, які інтерпретують метрики конструктивних елементів та їх історію. Атрибутами і метриками історії можуть бути вік конструктивного елемента програмного модуля, кількість внесених у нього змін, кількість розробників, які оновлювали елемент. На основі моделей помилок можна визначити атрибути помилок [15, 16].

Навігаційний компонент забезпечує користувачам можливість перегляду атрибутів досліджуваного програмного модуля у вигляді ієрархії. Вершинами ієрархії є програмні модулі, їх структурні елементи та версії. Навігацію представлено на рис. 3.4.

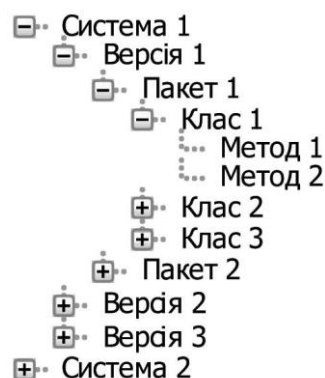


Рис. 3.4. Система навігації при дослідженні помилок у програмному модулі

Використовуючи ієрархічне дерево розробник може досліджувати структуру програмного модуля для якого необхідно забезпечити моніторинг розвитку помилок чи дефектів. Також при використанні навігації передбачено можливість пошуку по вершинах дерева.

3.3. Компонент візуалізації розвитку помилок

Програмний компонент візуалізації помилок повинен дозволяти вирішувати наступні завдання:

- препроцесинг даних для забезпечення можливості візуалізації помилок і конструктивних елементів з врахуванням обмежень щодо типу і кількості помилок, які можуть бути одночасно відображеними на графіку;
- визначення патернів оновлення помилок;
- відображення графіків;
- інтерактивність користувачів із візуальним представленням помилок.

Компонент візуалізації помилок і дефектів програмного модуля повинен мати можливість генерувати «Рентгенограми», «Історії дефектів», «Історії властивостей помилки».

Компоненти, які забезпечують автоматизацію процесів візуалізації помилок у програмному модулі комп'ютерної системи зображені на рис. 3.5.

Модуль препроцесингу даних при візуалізації параметрів помилок отримує в якості аргументу дані щодо типу візуалізації. На основі екземпляра моделі DDHM і вхідних параметрів щодо зображення, яке потрібно згенерувати, компонент візуалізації формує на виході відповідні дані у визначеному форматі.

Підсистема визначення класів помилок отримує на вхід згенеровані підсистемою препроцесингу структури даних і проводить їх аналіз для знаходження категорії помилок з мітками класу, які інтерпретують розвиток помилки, її спадання, пульсацію чи стабільність.

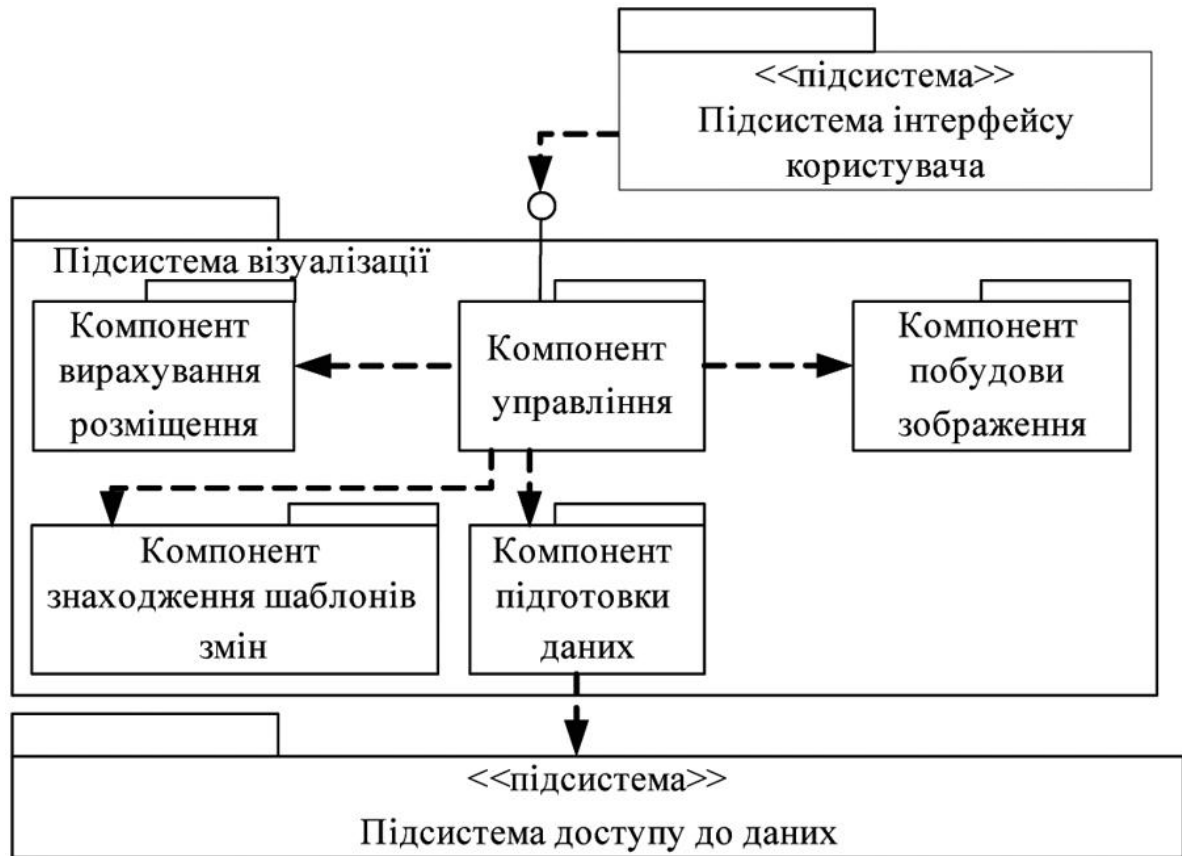


Рис. 3.5. Структура підсистеми візуалізації

У результаті функціонування підсистеми пошуку категорій помилок виконується оновлення даних таким чином, щоб при їх візуалізації можна було виділити мітки відповідних класів і передати їх на вихід підсистеми.

Підсистема розрахунку локації для розміщення візуального представлення помилок у вигляді обраного типу графіка виконує обчислення координат кожного елемента, який буде візуалізований.

Далі керування передається до компоненту генерації зображення, який виконує вивід на екран графіків розвитку помилок у програмному модулі комп'ютерної системи на основі результатів, одержаних з попередніх компонентів підсистеми.

3.4. Підсистема моделей помилок, історій і версій

Підсистема моделей помилок повинна забезпечити автоматизацію і розв'язок таких задач:

- доступ до набору моделей помилок та їх описів, які наявні у сховищі даних;
- визначення області значень метрик інтенсивності ознак помилок;
- маніпуляції із додавання, видалення, зміни, пошуку і тестування моделей помилок;
- доступ до сукупності метрик, які використовуються при аналізі помилок у програмних модулях комп'ютерних систем;
- можливість вибору метрик для оцінювання показників надійності конкретного програмного модуля;
- маніпулювання властивостями ознак помилок.

Розв'язок таких задач покладено на підсистему, структуру якої представлено на рис. 3.6.

Модуль маніпулювання властивостями моделей помилок в якості аргументів приймає ідентифікатори моделі та операції маніпулювання. У випадку, коли ідентифікатори мають значення Null, то спрацьовує режим перегляду моделей, наявних у сховищі даних. Модуль побудови моделі виконує підтвердження створення або зміни у структурі моделі помилки.

Візуальне створення або внесення змін у моделі помилок забезпечує модуль побудови моделей, надаючи користувачу перелік і метаопис метрик. Користувач має можливість задати ознаки помилки, здійснити їх декомпозицію на більш простіші ознаки. Розбиття ознак відбувається до тих пір, доки не буде одержано таку сукупність ознак, щоб можна було виміряти та оцінити їхню на основі метрик, наявних у сховищі даних.

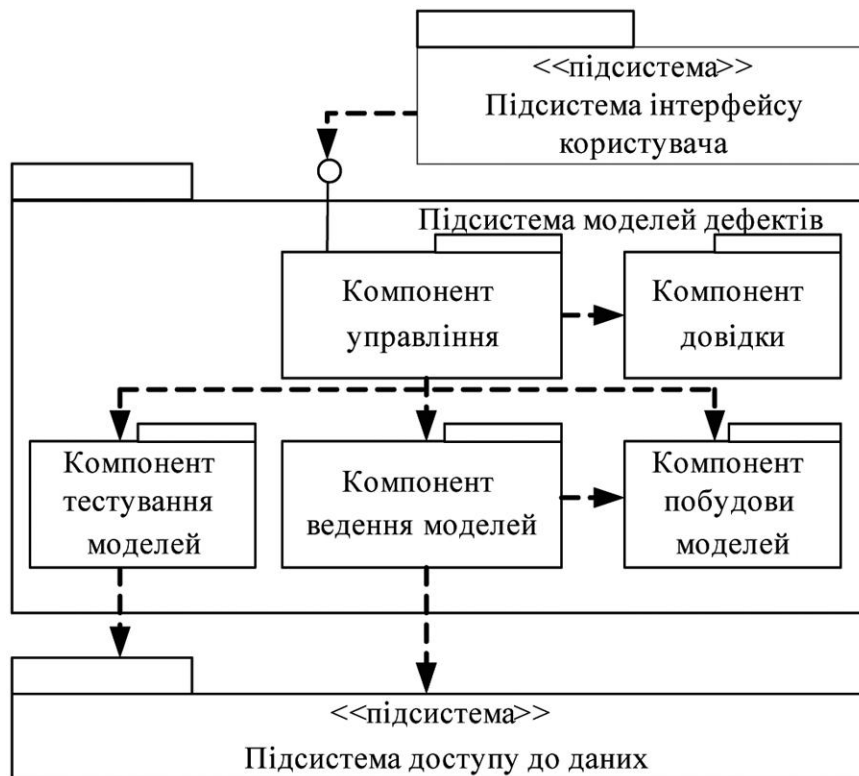


Рис. 3.6. Структура підсистеми моделей дефектів

Якщо декомпозиція ознаки помилки неможлива, то така ознака є простою. Наступний крок полягає у створенні зв'язку між простою ознакою помилки та метрикою, задавши її граничне значення. Вікно формування простої ознаки та зв'язку з метрикою показано на рис. 3.7.

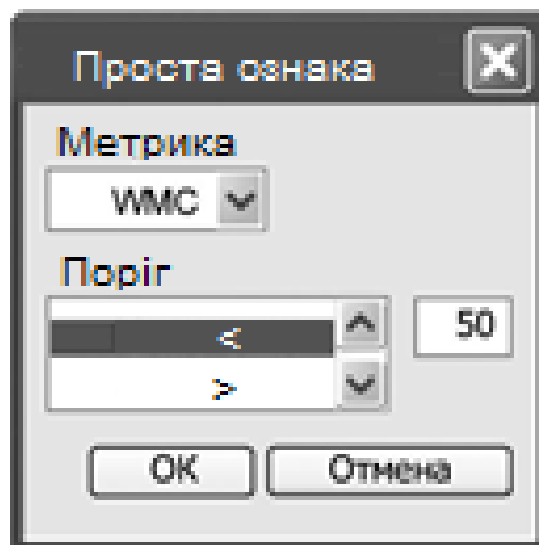


Рис. 3.7. Редактор простої ознаки

За допомогою логічних операторів «І» та «АБО» із простих ознак можна формувати більш складні.

Підсистема тестування моделей забезпечує одержання необхідних тестових наборів даних у користувача та забезпечує виконання розрахунків для моделі помилки у програмному модулі.

Компонент для одержання інформації щодо використання засобу автоматизації виявлення помилок, моніторингу та управління показниками надійності забезпечує інтерактивність пошуку даних за вказаними ключовими словами та повернення результатів у зручному для користувача вигляді. Програмно довідка реалізована за допомогою технології XML.

3.5. Підсистема формування звітів

На підсистему формування звітів покладено виконання наступних задач:

- створення, редагування або знищення шаблону звіту;
- виконання операцій маніпулювання звітами;
- маніпулювання запитом щодо результатів аналізу помилок;
- генерація запитів;
- автоматичне створення звітів за зразком;
- перегляд інформації у звітах;
- пошук даних у звітах;
- перегляд даних щодо вимірювань інтенсивності ознак помилок у

програмному модулі, який діагностувався.

Структура підсистеми формування звітної документації представлена на рис. 3.8.

Найбільш важливим компонентом у підсистемі формування звітної документації є модуль генерації звітів, який в якості вхідного аргументу одержує ідентифікатор типу шаблону, а після цього автоматично генерує звітний документ.

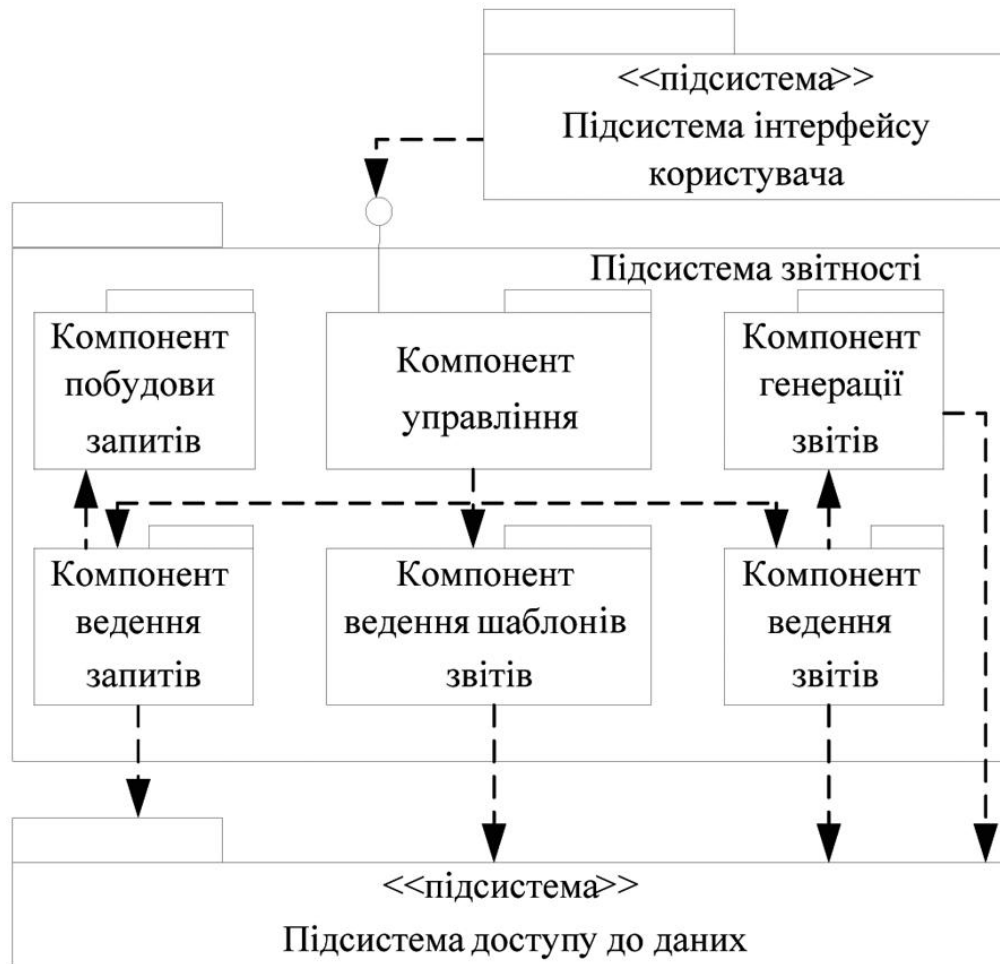


Рис. 3.8. Структура підсистеми формування звітної документації

Алгоритм функціонування даного модуля полягає в наступному:

- знаходження ідентифікатора звіту у сховищі даних, який відповідає значенню вхідного аргументу типу шаблону;
- виконання запитів, необхідних для заповнення полів шаблону;
- запис даних у відповідні поля шаблону звіту;
- зберігання звіту у вигляді окремого документу у сховищі даних;
- передача або друк звіту.

3.6. Проектування схеми бази даних для зберігання даних управління і моніторингу показників надійності

Сховище даних, або у даному конкретному випадку реляційна база даних необхідна для зберігання даних про екземпляри моделі DDHM, моделі помилок, структури шаблонів і форми звітів, що використовуються при аналізі програмного модуля комп'ютерної системи.

Для розв'язку наведених вище задач, спроектовано схему реляційної бази даних, яка представлена на рис. 3.9 у вигляді діаграми сутностей і зв'язків. Така структура забезпечує зберігання необхідних даних для проведення процедур щодо визначення показників надійності та управління і моніторингу за розвитком помилок у програмному модулі.

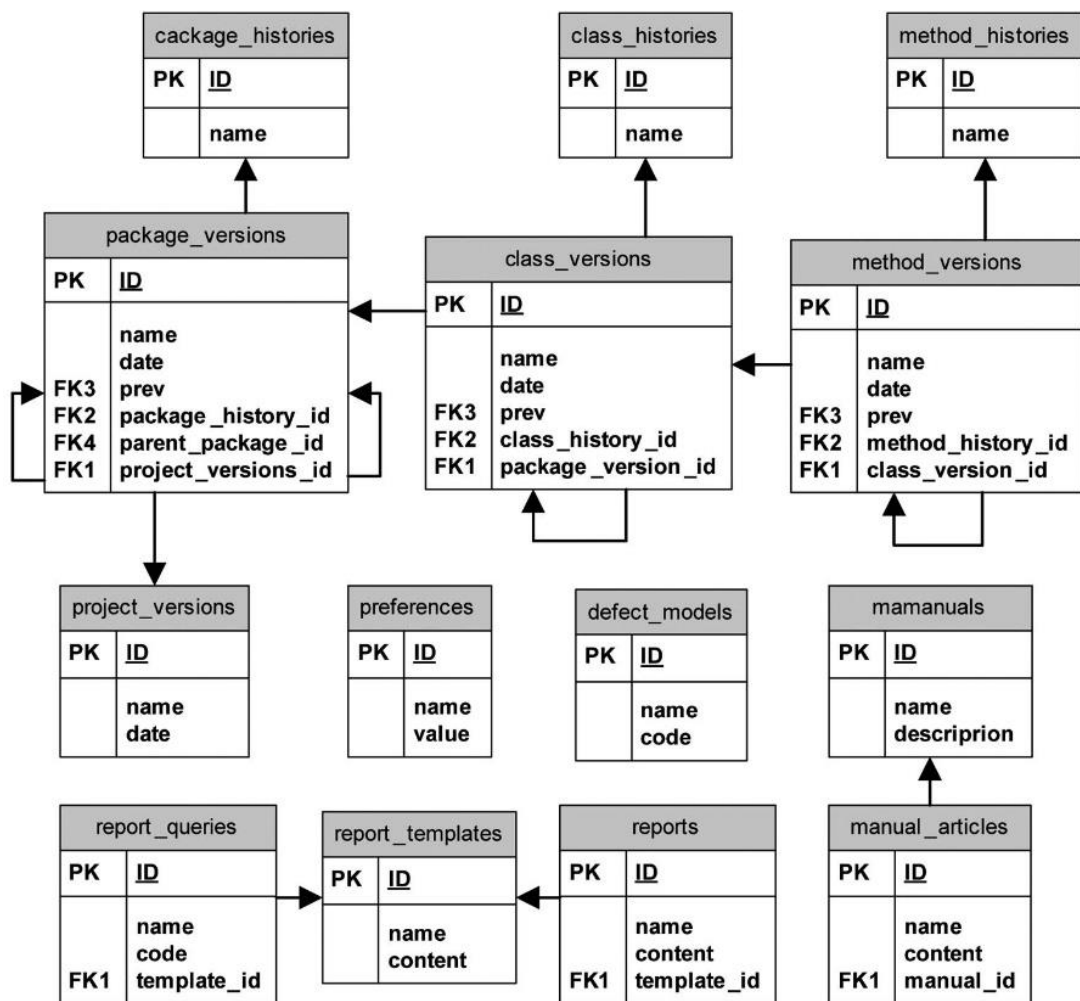


Рис. 3.9. ER діаграма бази даних

Далі доцільно провести експерименти щодо виявлення та моніторингу помилок на основі запропонованих моделей і методів. В якості прикладу візьмемо програмне забезпечення з відкритим вихідним кодом. Моніторинг помилок проводиться у класі «God Class», який є частиною інструменту ArgoUML.

Характеристики ArgoUML одержано з використанням описаних вище підсистем засобу автоматизації процесу аналізу помилок у програмних модулях комп'ютерних систем (табл. 3.1), а рис. 3.10 – демонструє інтерфейс і графіки помилок, які містить обраний програмний модуль.

Таблиця 3.1

Характеристики ArgoUML

Характеристика	Значення
Кількість рядків коду	136,000
Кількість методів/інтерфейсів	14,221
Кількість класів	2,522
Кількість пакетів	143

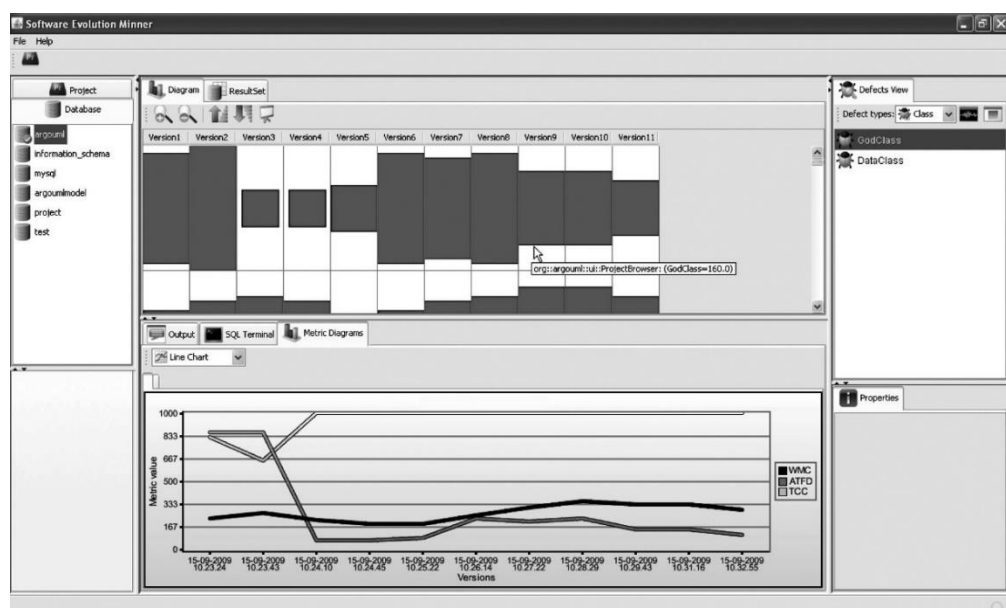


Рис. 3.10. Знімок засобів інтерфейсів користувача

Як видно з рис. 3.10 помилки класу «God Class» поширюються на інші класи, які забезпечують основну функціональність програмного модуля.

При детальному дослідженні класів, які показані на рис. 3.11, можна спостерігати збільшення кількості структур у яких наявні помилки, внесені розробниками програмного забезпечення. Вважається, що клас містить помилку в тому випадку, якщо рівень її розвитку перевищує 100%. На рис. 3.11 спостерігається лінійний тренд до зростання.

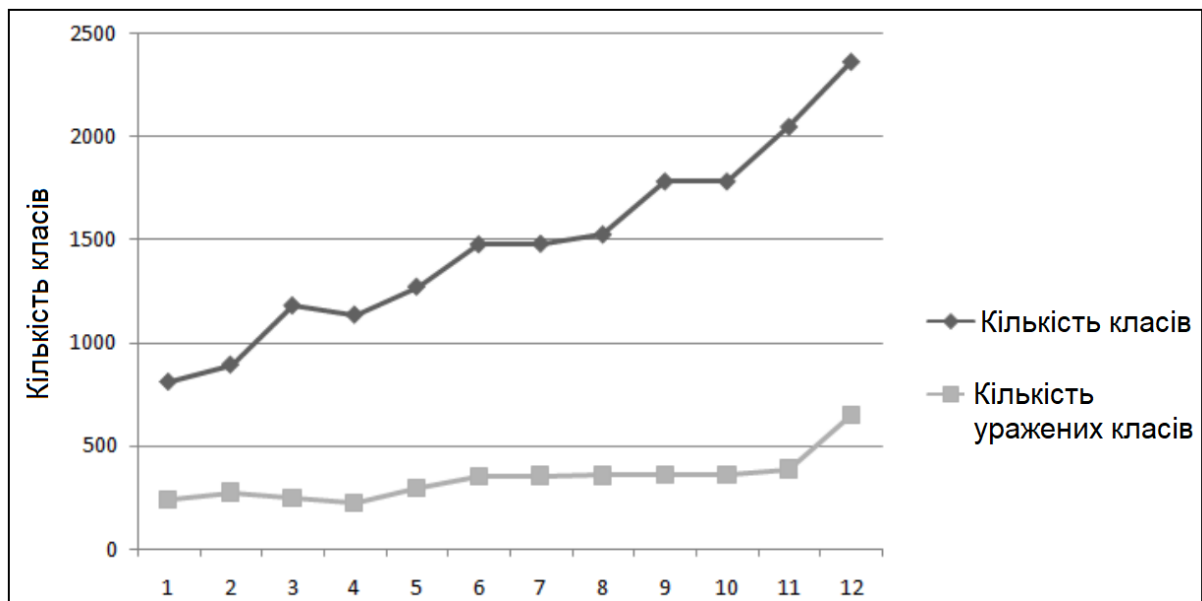


Рис. 3.11. Кількість дефектів версіях ПЗ КС

Кількість класів, які містять помилки на протязі усієї історії є відносно стабільною величиною, однак спостерігається видиме їх збільшення в останньому релізі системи. Окрім цього, аналізуючи рис. 3.13 можна помітити тенденцію швидшого зростання кількості працездатних класів у порівнянні з класами, які містять помилки. При цьому помилки містяться у ранніх початкових версіях системи.

Розподіл помилок, наявних у класах програмного модуля комп'ютерної системи на основі запропонованих міток класів показано на рис. 3.12. Як видно з діаграми, близько 77% помилок на протязі усієї історії не зазнавали змін і формують найбільш потужний клас.

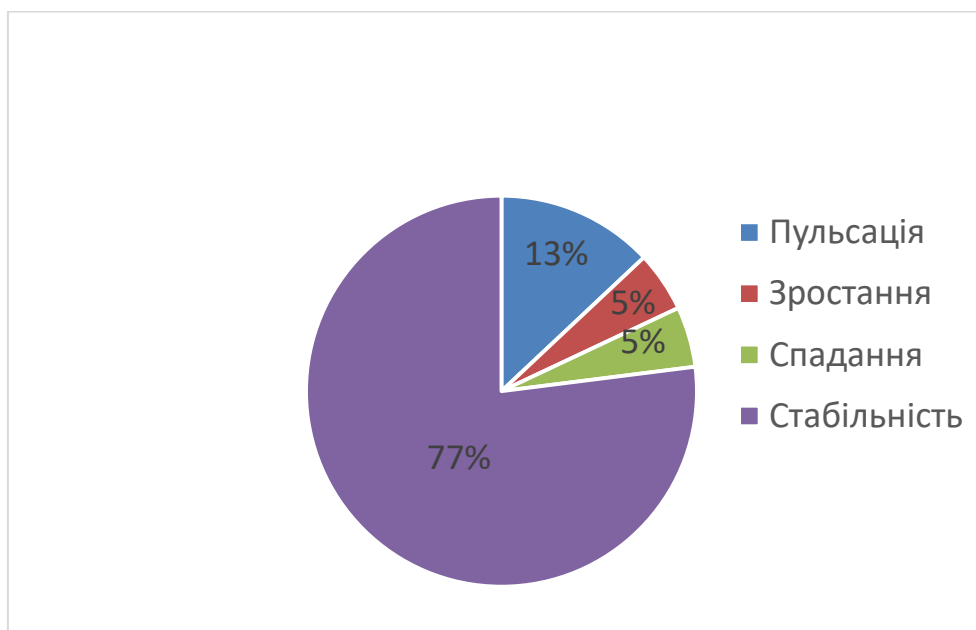


Рис. 3.12. Розподіл дефектів за основними категоріями

На рис. 3.13 та рис. 3.14 наведено розподіл помилок за підкласами «Пульсації» і «Зростання».

Аналізуючи результати, представлені на рис. 3.14, можна спостерігати, що найбільш потужним є нижній клас, який містить помилки на стадії їх зародження, тобто рівень їхнього розвитку менше 100%.

Найменший клас включає в себе найбільш небезпечні помилки: «Верхня границя зростання» - 4% (8), «Верхній рівень пульсації» - 4% (16). Отже, пріоритетними завданнями мають бути застосування дій щодо реструктуризації двадцяти чотирьох класів, які містять помилки такого типу.

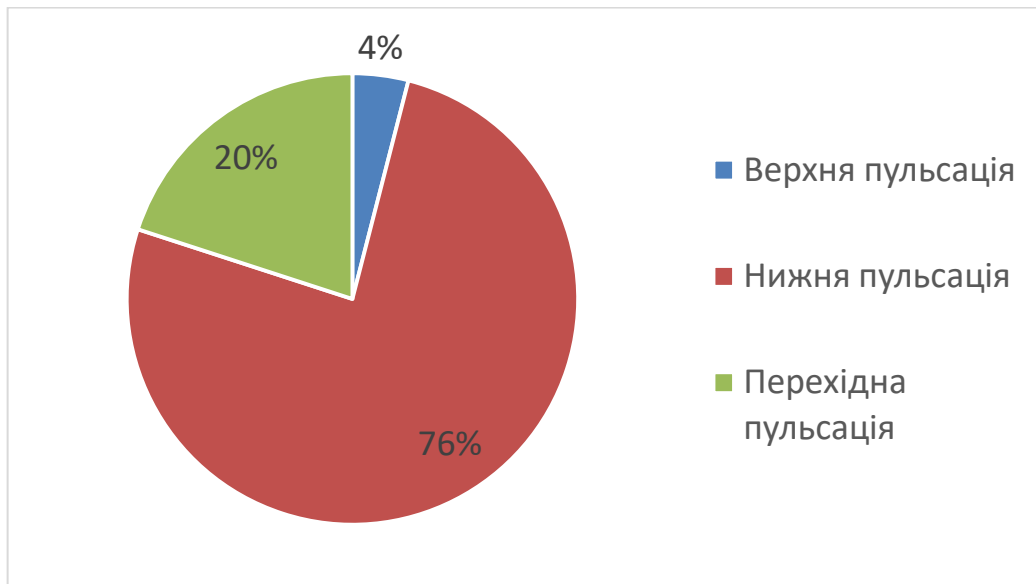


Рис. 3.13. Розподіл дефектів за категорією «Пульсація»



Рис. 3.14. Розподіл дефектів за категорією «Зростання»

Історія помилок при реалізації програмного модуля показана на рис. 3.15. Тут, верхній елемент, який позначає пульсацію помилки міститься у класі «FigNodeModelElement», а рівень його розвитку більший за 100% на протязі усєї історії.



Рис. 3.15. Історія дефектів ArgoUML

Таким чином, розроблений метод і засіб дають змогу проводити моніторинг та управління показниками надійності шляхом аналізу помилок у програмних модулях комп'ютерних систем з використанням історії їх розвитку та аналізу версій модуля. Метод спостереження за помилками та візуалізації класів помилок дозволяє виявляти потенційно небезпечні помилки, які розвиваються у часі і вживати заходів щодо своєчасного їх усунення.

3.7. Висновки до розділу

Основні результати даного розділу полягають в наступному:

1. На основі реляційного підходу спроектовано базу даних для зберігання та маніпулювання даними при визначенні впливу дефектів

програмного забезпечення на надійність комп'ютерної системи та реалізовано її у середовищі MS SQL Server.

2. Спроектовано архітектуру та розроблено інтерфейси користувачів програмного засобу маніпулювання критеріями надійності, моделями прогнозування дефектів програмного забезпечення, формування імовірних шляхів поширення дефектів на компоненти комп'ютерної системи, що дало змогу забезпечити ефективність виявлення та моніторингу дефектів.

3. Проведено експериментальні дослідження щодо застосування запропонованих та обґрунтованих у роботі моделей, методів і програмного засобу, що дало змогу підтвердити доцільність їх використання.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Охорона праці

Кваліфікаційна робота магістра присвячена дослідженню методів і засобів управління та моніторингу показників надійності при розробці комп'ютерних систем. Важливим та необхідним у процесі розробки та експлуатації засобів керування показниками надійності КС є дотримання вимог техніки безпеки на робочому місці, пожежної безпеки, відповідно до діючих нормативно-правових актів і встановлених норм.

Перед початком експлуатації систем необхідно проводити інструктаж з техніки безпеки на робочому місці відповідно до вимог Типового положення про інструктажі, спеціальне навчання та перевірку знань з питань пожежної безпеки на підприємствах, в установах та організаціях України згідно з ДБН В.1.1-7-2016 „Пожежна безпека об'єктів будівництва та НПАОП 0.00-4.36-05.

Розробка комп'ютерних систем та проведення досліджень відбувались відповідно до Закону України "Про охорону праці" та "Правил охорони праці при експлуатації електронно-обчислювальних машин" та Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, а також НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [23].

Робоче місце відповідає ергономічним, організаційним вимогам, вказаних у вищеперахованих документах.

Електробезпека на робочому місці, повинна відповідати вимогам Правил безпечної експлуатації електроустановок споживачів, затверджених наказом Держнаглядохоронпраці від 09.01.98 N 4 , зареєстрованих у Міністерстві юстиції України 10.02.98 за N 93/2533 [23].

Для забезпечення електробезпеки при роботі з ЕОМ під час проведення досліджень електромережа відповідала правилам:

- лінія електромережі живлення виконана як окрема групова трипровідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників;
- нульовий захисний провідник використовується для заземлення(занулення) електроприймачів;
- електричне устаткування, електропроводи та кабелі мають апаратуру захисту від струму короткого замикання та інших аварійних режимів;
- під час монтажу та експлуатації електромережі необхідно повністю унеможливити виникнення електричного джерела загоряння внаслідок короткого замикання та перевантаження проводів, обмежено застосування проводів з легкозаймистою ізоляцією та застосовано негорючу ізоляцію;
- електричне устаткування підключено до мережі лише за допомогою справних штепсельних з'єднань і розеток заводського виготовлення;
- у штепсельних з'єднаннях та електророзетках є спеціальні контакти для підключення нульового захисного провідника [23].

Електробезпека під час розробки прототипу системи підтримки методів і засобів управління та моніторингу показників надійності комп'ютерних систем забезпечувалась завдяки використанню безпровідних технологій та напруги живлення в діапазоні 3.3 – 9 В, що зменшує шанс ураження струмом та його вплив при виникненні контакту з мережею чи аварійних ситуацій.

Пожежна безпека будівель та приміщень, де розміщені робочі місця, обладнані ЕОМ з ВДТ і ПП відповідає вимогам, встановленим:

– державними будівельними нормами ДСТУ Б.В.1.1-36:2016 „Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою”;

– ДБН В.1.1-7-2016 „Пожежна безпека об’єктів будівництва”;

Приміщення для роботи з ВДТ повинні мати природне та штучне освітлення відповідно до ДБН В.2.5-28-2006. Природне освітлення забезпечується через світлові прорізи, орієнтовані переважно на північ чи північний схід і коефіцієнт природною освітленості (КПО) не нижче ніж 1,5%. Розраховується КПО за методикою, викладеною в ДБН В.2.5-28-2006.

Штучне освітлення в приміщенні з робочими місцями, обладнаними ВДТ ЕОМ та ПЕОМ, забезпечувалось системою загального рівномірного освітлення. В якості джерела світла для штучного освітлення використані люмінесцентні лампи типу ЛБ.

Також під час дослідження методів і засобів управління та моніторингу показників надійності при розробці комп’ютерних систем не було порушено наступних вимог:

- самостійного ремонту ЕОМ, ВДТ чи ПП;
- внесення змін до конструкції ЕОМ;
- не торкатись задньої панелі системного блоку при включеному живленні;
- вживання напоїв та їжі на робочому місці;
- нагромадження біля ЕОМ, ВДТ, ПП предмети, які не використовуються для поточної роботи.

Дослідження методів і засобів управління та моніторингу показників надійності при розробці комп’ютерних систем проведено з дотриманням вимог техніки безпеки та охорони праці, електробезпеки, пожежної безпеки та санітарних норм при роботі з ЕОМ та ВДТ.

4.2. Оповіщення керівного складу органів виконавчої влади, підприємств установ та організацій, населення про загрозу і виникнення НС природного, техногенного та воєнного характеру

При загрозі чи виникненні надзвичайних ситуацій природного, техногенного чи воєнного походження, згідно [25], правовою основою організації оповіщення населення є:

- Конституція України;
- Кодекс Цивільного захисту України;
- Постанови Кабінету Міністрів "Про затвердження Положення про організацію оповіщення про загрозу виникнення або виникнення надзвичайних ситуацій та зв'язку у сфері цивільного захисту";
- "Положення про єдину державну систему цивільного захисту";
- накази центрального органу виконавчої влади з питань НС, відповідні розпорядження обласної державної адміністрації та інші акти [25].

Система централізованого оповіщення області представляє собою комплекс організаційно-технічних заходів, апаратури і технічних засобів оповіщення, засобів та каналів зв'язку, мереж проводового, радіо, телевізійного мовлення призначених для своєчасного доведення сигналів та інформації з питань цивільної оборони (цивільного захисту) до центральних і місцевих органів виконавчої влади, підприємств, установ, організацій і населення [26]. Для зосередження уваги громадян перед передачею інформації вмикаються сирени, інші сигнальні засоби. Їх звук означає попереджувальний сигнал "УВАГА ВСІМ".

Взагалі система оповіщення складається із загальнодержавної, регіональних і спеціальних систем централізованого оповіщення; локальних та об'єктових систем оповіщення, систем циркулярного виклику [26]. Ці системи забезпечують оповіщення і подальше інформування:

- чергових служб міністерств та інших центральних органів виконавчої влади по службових телефонах;
- чергових служб місцевих органів виконавчої влади;
- чергових аварійно-рятувальних служб.

Для виконання основних завдань оповіщення, які визначені керівними документами, а саме: забезпечення своєчасного проходження інформації між органами управління щодо ступенів готовності; оповіщення керівного складу, населення про загрозу радіоактивного, хімічного і бактеріологічного ураження, про загрозу і виникнення надзвичайних ситуацій у мирний і особливий період та постійне інформування його про наявну обстановку [26].

Система оповіщення працює за принципом відбору каналів з єдиної національної системи зв'язку. Апаратура оповіщення розташована на відповідних об'єктах органів управління, електрозв'язку, чергових відділах МВС, на радіо-теле-передавальних центрах та інших визначених підприємствах і установах [26].

Для оперативного доведення відповідної інформації до керівного складу по телефонам застосовуються стійки циркулярного виклику та апаратура автоматизованого багатоканального оповіщення [26].

Для передачі попереджувального сигналу "УВАГА ВСІМ" застосовуються електричні сирени централізованого і автономного включення, наявна кількість яких в основному забезпечує озвучення території де проживає населення області [26].

Інформація до населення доводиться через радіотрансляційні вузли, радіо-теле-передавальні центри по проводовому мовленню до якого підключено радіоточки і вуличні гучномовці, по визначеним радіо та телевізійним каналам.

На випадок виникнення надзвичайної ситуації безпосередньо на потенційно небезпечних підприємствах за їх рахунок створюються об'єктові системи оповіщення [26]. Локальні системи оповіщення створюються на

потенційно небезпечних об'єктах, зона ураження від яких, у разі виникнення на них надзвичайної ситуації, досягає заселених територій або інших підприємств, установ, організацій. До їх складу входять абонентські радіоточки мережі радіомовлення та відомчих радіотрансляційних вузлів, вуличні гучномовці, пристрої запуску електросирен та самі електросирени, система централізованого виклику, магнітофони, магнітні стрічки із записаними текстами звернень [26].

Спеціальні системи оповіщення створюються і функціонують:

- на атомних електростанціях;
- на гідротехнічних спорудах Дніпровського та Дністровського каскадів та в зонах їх можливого катастрофічного затоплення;
- на магістральних продуктопроводах.

Такі системи передбачають взаємодію з відповідними територіальними та місцевими автоматизованими системами централізованого оповіщення.

На атомних електростанціях спеціальні системи оповіщення повинні забезпечувати:

- передачу сигналу “Увага всім”;
- передачу повідомлень на території атомної електростанції та її промислової зони;
- оповіщення міста - супутника атомної електростанції;
- оповіщення відповідних оперативно-чергових (чергових) служб місцевих органів виконавчої влади (органів місцевого самоврядування), територіальних органів ДСНС та Національної поліції.

Готовність систем оповіщення забезпечено шляхом:

- організованої цілодобової чергової відповідних служб;
- налагодження телефонного зв'язку чергових служб потенційно небезпечних підприємств, зона ураження яких може поширюватися на заселені території або території інших підприємств, установ, організацій з

оперативно-черговою службою пункту управління облдержадміністрації, чергових служб органів МВС в містах та районах області;

- завчасної підготовки персоналу чергових служб до дій у надзвичайних ситуаціях;

- впровадження автоматизованих систем оповіщення з використанням сучасних технологій;

- якісного експлуатаційно-технічного обслуговування апаратури і технічних засобів оповіщення та системи зв'язку.

Забороняється відключати радіотрансляційні точки та абонентські лінії, через які здійснюється запуск електросирен від мереж радіомовлення, демонтувати вуличні гучномовці без погодження з відповідними органами управління з питань ЦЗН [26].

4.3. Фактори ризику і можливі порушення здоров'я користувачів комп'ютерної мережі

На перший погляд, робота за комп'ютером здається безпечною, але саме легковажність до неї може призвести до певних проблем у здоров'ї людини. Професія програміста та інших фахівців ІТ-технологій пов'язана з колосальним розумовим напруженням. Розробники – настільки захоплені люди, що навіть відволікаючись від роботи над проектом, продовжують думати про роботу. Нерідко відпочинком вони вважають паралельну заміну основної діяльності, наприклад, читання профільної літератури, верстку сайтів, вивчення нових мов програмування.

Однак мозок не може до безкінечності приймати виключно корисну інформацію, яку розробник прагне направляти в русло особистісного та професійного зростання. Адже мозок людини не машина: він не може нескінченно зберігати і переробляти дані практично не втрачаючи продуктивності.

Інтенсивна робота з персональними комп'ютерами, обладнаними візуальними дисплейними терміналами є причиною виникнення багатьох захворювань. Стан здоров'я користувачів ПК за суб'єктивними і об'єктивними показниками залежить від типу роботи і умов її виконання.

Робота користувача ПК виконується в одноманітній позі в умовах обмеження загальної м'язової активності при рухливості кистей рук, великому напруженні зорових функцій і нервовоемоційній нарузі під впливом наступних фізичних факторів:

- електростатичного поля;
- електромагнітних випромінювань у наднизькочастотному, низькочастотному та середньо частотному діапазонах (5 Гц – 400 кГц);
- рентгенівського, ультрафіолетового та інфрачервоного випромінювань;
- випромінювань видимого діапазону; акустичного шуму; незадовільного рівня освітленості;
- незадовільних метеорологічних умов.

Особливе місце серед профзахворювань користувачів ПК займають порушення зору, викликані:

- нераціональним освітленням;
- світлотехнічною специфікою робочих місць із ПК;
- недотриманням режиму праці.

Світлотехнічна специфіка обумовлена світлотехнічною різноманітністю об'єктів зорової роботи користувача ПК: екрана, документації і клавіатури, розташованих у різних зонах спостереження, що вимагає багаторазового переміщення лінії зору від одного об'єкта до іншого. Об'єкти відмінності мають як негативний (темні об'єкти на світлому фоні) так і позитивний (світлі об'єкти на темному фоні) контраст. Тому відбувається постійна переадаптація від яскравих об'єктів з позитивним контрастом на темні з негативним контрастом.

За восьмигодинний робочий день за монітором користувач кидає приблизно 30000 поглядів на екран, око працює з перевантаженням і не може достатньо адаптуватися до цієї ситуації. Такі особливості призводять до напруження м'язового та світло-сприймаючого апарату очей, що є однією з причин виникнення астенопічних явищ (різь в очах, біль в очах, ломить у надбрівній ділянці, розпливчастість контурів, нечіткість зображення).

Постійний погляд на матове скло екрана монітора зменшує частоту кліпання очей, що призводить до висихання та викривлення роговиці ока, погіршує зір (синдром Сікка). Робота користувача за пульсуючим екраном монітора, що не відповідає нормативним вимогам щодо обмеження пульсації (блмання), викликає дискомфорт і втоми (загальну і зорову).

Робота з дзеркальною відбиваючою і неплоскою зовнішньою поверхнею екрана монітора, на якій з'являються численні відбиті відблиски, призводить до виникнення у користувача астенопічних явищ та функціональних змін ока. Неправильний розподіл яскравості в полі зору, тобто поверхні периферії (стеля, стіни, меблі і т.п.) висвітлені краще ніж центр поля зору, призводить до порушення основних зорових функцій ока.

Засліплююча дія світильників у приміщенні, на робочому місці з ПК викликає не тільки астенопічні явища, але й функціональні порушення очей користувача. Кольоровий шрифт збільшує навантаження на зір, оскільки складові кольорів мають різні довжини хвиль і видимі на різній віддалі. Око потребує точнішої адаптації, ніж при чорно-білому зображенні.

Робота користувача ПК вимагає тривалого статичного напруження м'язів спини, шиї, рук і ніг, що призводить до втоми і специфічних скарг. Можливе ушкодження хребта, в результаті недостатнього рівня ергономічності робочого місця користувача, тобто якщо крісло неправильно підтримує згин хребта. При цьому плечі і шия напружені і затікають, внаслідок неприродного положення, тому виникають болі в області шиї, спини і голови.

В середньому працівник, який використовує ПК, просиджує в такому положенні за все своє життя до 80000 годин (8 років). Неправильне положення рук при введенні даних за допомогою клавіатури призводить до перетискання нервів у вузьких місцях зап'ястя.

Синдром RSI (хронічне розтягнення зв'язок) – це пошкодження, що виникає в результаті постійного напруження м'язів кистей рук як результат неправильно обладнаного з погляду ергономіки робочого місця користувача ПК. Це хронічне захворювання може непомітно розвиватися протягом декількох років. Такі перевантаження призводять до перенапруження всієї м'язової системи людини.

Найбільш небезпечним є те, що внаслідок концентрації уваги на екрані монітора притуплюється своєчасне попередження про болі, які є тривожним сигналом для тіла. Захворювання рук і кистей рук спостерігається у працюючих за ПК у 7-12 разів частіше, ніж у інших, і досить часто помилково діагностується як запалення сухожилів.

Отже, працюючи за комп'ютером, рекомендується дотримуватися деяких правил:

- за будь-яких умов безперервна робота за комп'ютером для дорослої людини не повинна перевищувати двох годин.
- слідкувати за поставою: ноги твердо стоять на підлозі чи на спеціальній підставці; стегна розташовані під прямим кутом до тулуба, а гомілки – під прямим кутом до стегон;
- сидіти потрібно прямо або злегка нахилившись вперед;
- пальці рук знаходяться на рівні зап'ястків або трохи нижче – у такому положенні вони найбільш рухливі; плечі мають бути розслаблені та вільно опущені, що сприяє розслабленню рук;
- відстань від очей до екрану монітора – не менше 55-60 см;
- центр екрану має знаходитися на рівні очей чи трохи нижче; рекомендується хоча б раз на день виконувати гімнастику для очей;

- для попередження „синдрому сухого ока”, потрібно моргати кожні 3-5 секунд;
- у процесі роботи за комп’ютером обов’язково потрібно звертати увагу на дихання: воно має бути рівномірним, без затримок;
- у процесі роботи рекомендується періодично (приблизно раз на 20 - 30 хвилин) переводити погляд з екрану на найбільш віддалений предмет у кімнаті, а ще краще – на віддалений об’єкт за вікном;
- якщо з’явилося відчуття втоми, напруження, сонливості, тяжкості в очах, потрібно припинити роботу та хоча б трохи відпочити.

Висновок.

Оповіщення керівного складу органів виконавчої влади, підприємств, установ та організацій, населення про загрозу і виникнення НС природного, техногенного та воєнного характеру є важливим аспектом життєзабезпечення і досягається шляхом організації локальних, територіальних та загальнодержавних систем оповіщення. Такі системи функціонують з використанням радіоканалу передачі інформації про надзвичайні події, звукових сирен на території підприємства або міста, а також засобів телебачення чи інших систем зв’язку.

Дотримання правил режиму праці користувачами комп’ютерів і комп’ютерної мережі забезпечує збереження їхнього здоров’я, а правильна організація робочого місця сприяє зниженню негативного впливу на всі системи організму людини.

ВИСНОВКИ

Основні наукові та практичні результати полягають в наступному.

1. Проведено аналіз моделей надійності компонентів комп'ютерних систем у результаті якого встановлено, що більшість відмов у роботі систем пов'язано з наявністю помилок і дефектів у їхніх програмних складових.

2. Проведено аналіз типів помилок, їх типів та впливу на надійність комп'ютерної системи, що дало змогу враховувати їх розвиток та обґрунтувати необхідність застосування моделей прогнозування показників надійності на різних стадіях життєвого циклу і спостереження за ними під час експлуатації і супроводу.

3. Проведено аналіз моделей надійності комп'ютерних систем з акцентом на програмне забезпечення і виявлено, що застосування моделей оцінювання надійності технічних засобів є не ефективним для представлення показників надійності програмного забезпечення у зв'язку з різною природою та механізмами відмов.

4. На основі результатів аналізу моделей раннього і пізнього прогнозування помилок програмного забезпечення обґрунтовано їх застосування при проектуванні комп'ютерних систем, що дає змогу побудувати комплексний підхід до визначення впливу дефектів програмного забезпечення на надійність експлуатації комп'ютерних систем з врахуванням історії виникнення помилок у ПЗ.

5. Побудовано та обґрунтовано для застосування модель представлення показників надійності, що враховує концептуальну структуру комп'ютерних систем і дає можливість на основі аналізу помилок забезпечувати їх моніторинг та приймати управлінські рішення для усунення загрози відмови чи збою у роботі компонентів і системи в цілому.

6. Набув подальшого розвитку метод моніторингу дефектів програмних модулів комп'ютерних систем на основі моделі DDHM, що дає змогу проводити спостереження за його розвитком та візуалізувати вплив

на надійність комп'ютерної системи у часі з врахуванням версій програмних модулів системи.

7. На основі реляційного підходу спроектовано базу даних для зберігання та маніпулювання даними при визначенні впливу дефектів програмного забезпечення на надійність комп'ютерної системи та реалізовано її у середовищі MS SQL Server.

8. Спроектовано архітектуру та розроблено інтерфейси користувачів програмного засобу маніпулювання критеріями надійності, моделями прогнозування дефектів програмного забезпечення, формування імовірних шляхів поширення дефектів на компоненти комп'ютерної системи, що дало змогу забезпечити ефективність виявлення та моніторингу дефектів.

9. Проведено експериментальні дослідження щодо застосування запропонованих та обґрунтованих у роботі моделей, методів і програмного засобу, що дало змогу підтвердити доцільність їх використання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сергієнко І.В. Становлення і розвиток досліджень з інформатики. К.: Наукова думка. 2008. 205 с.
2. ДСТУ 3918-99 Інформаційні технології. Процеси життєвого циклу програмного забезпечення. Київ. Держстандарт України. 2000. 49 с.
3. IEEE 982.1 Standard Dictionary of Measures to produce Reliable Software. IEEE. 1988. 56 p.
4. Highsmith J. Retiring Lifecycle Dinosaurs. Software Testing & Quality Engineering. 2010. URL: <http://www.stqemagazine.com> (дата звернення 08.11.2023 р.)
5. Lakey P.B., Neufelder A.M. System and software reliability assurance notebook. Rome Laboratory Report, Griffiss Air Force Base, Rome NY. 2017. 186 p.
6. Musa J.D. Operational Profiles in Software Reliability Engineering. IEEE Software. V.10. N.2. 2003. P. 14 - 32.
7. Munson J., Elbaum S. Software reliability as a function of user execution patterns. Proc. of the 32nd Hawaii International Conference on System Sciences. 1999. P.1-12.
8. Hecht H. An Alternative Software Reliability Assessment. Proceedings 14th International Symposium on Software Reliability Engineering (ISSRE 2003), Denver, Colorado. 2003. P. 293 - 295.
9. Ohlsson N., Helander M., Wohlin C. Quality Improvement by Identification of Fault-Prone Modules using Software Design Metrics. Proceedings Sixth International Conference on Software Quality. 1996. P. 1 – 13.
10. Fenton N.E. A critique of software defect prediction models. IEEE Trans. On Soft. Eng. V. 25. N.5. 1999. P. 675 – 689.
11. Malaiya Y.K., Denton J. What do the Software Reliability Growth Model Parameters Represent. Proc. IEEE-CS Int. Symp. on Software Reliability Engineering ISSRE. 1997. P. 124 - 135.

12. Chulani S. Constructive quality modeling for defect density prediction: COQUALMO. International Symposium on Software Reliability Engineering (ISSRE'99). Boca Raton. 1999.

13. Пасічник В., Резніченко В. Організація баз даних та знань. К.: Видавнича група BHV, 2006. 384 с.

14. Cheung R. A User-oriented Software Reliability Model. IEEE Trans. Soft. Eng. SE-6, N. 2. 1980. P. 11- 125

15. Kharchenko A., Galay I., Yatsyshyn V. The method of quality management software. 2011 Proceedings of 7th International Conference on Perspective Technologies and Methods in MEMS Design, MEMSTECH 2011 . Polyana. 2011. pp. 82-84.

16. Yatsyshyn V., Pastukh O., Lutskiv A., Tsymbalistyy V., Martsenko N. A Risks management method based on the quality requirements communication method in agile approaches / Information technologies: theoretical and applied problems, 2022, pp. 1-10.

17. Harchenko A., Bodnarchuk I., Yatsyshyn V. The modeling and optimization of software engineering processes. Modern Problems of Radio Engineering, Telecommunications and Computer Science - Proceedings of the 11th International Conference, TCSET'2012. Lviv - Slavske , 2012. p. 326.

18. Yasniy O., Pastukh O., Didych I., Yatsyshyn V., Chykhira I. Application of machine learning for modeling of 6061-T651 aluminum alloy stress–strain diagram. Procedia Structural Integrity. 48. 2023. pp. 183–189.

19. Yatsyshyn V., Pastukh O., Palamar A., Zharovskyi R. Technology of relational database management systems performance evaluation during computer systems design. Scientific Journal of TNTU (Tern.). Vol. 109. No 1. 2023. pp. 54–65.

20. Ясній О.П., Крисюк І.В. Фактори впливу на надійність комп'ютерних систем в процесі їх розробки. Матеріали XII міжнародної науково-практичної конференції молодих учених та студентів «Актуальні

задачі сучасних технологій» (6-7 грудня 2023 року). Тернопіль: ТНТУ. 2023. С. 462.

21. Ясній О.П., Крисюк І.В. Архітектура засобу розрахунку та управління показниками надійності комп'ютерних систем. Матеріали XI науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (13-14 грудня 2023 року). Тернопіль: ТНТУ. 2023. С. 161.

22. НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». Київ. 2018.

23. Бедрій Я. Основи охорони праці користувачів персональних комп'ютерів: навчальний посібник для студентів ВНЗ та інженерів-практиків. Навчальна книга-Богдан. 2014. 144 с.

24. Жидецький В.Ц. Охорона праці користувачів комп'ютерів. Львів: Афіша, 2011. 176 с.

25. Желібо Е.Н. Безпека життєдіяльності: Навчальний посібник/ За редакцією Е.П. Желібо, В.М. Львів: «Новий світ - 2000», 2011. 320с.

26. Стадник І.Я., Зварич Н.М. Оцінка хімічної обстановки при аваріях на хімічно небезпечних об'єктах викидом (виливом) небезпечних хімічних речовин та застосуванні хімічної зброї. ТНТУ. 2020. 36 С.

Додаток А
Тези конференцій

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

АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ

Збірник
тез доповідей

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



УКРАЇНА
ТЕРНОПІЛЬ – 2023

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

- | | | |
|-----|--|-----|
| 70. | О. П. Ясній, І. В. Крисюк
ФАКТОРИ ВПЛИВУ НА НАДІЙНІСТЬ КОМП'ЮТЕРНИХ СИСТЕМ В ПРОЦЕСІ ЇХ РОЗРОБКИ | 462 |
| 71. | О. П. Ясній, М. М. Галас
АРХІТЕКТУРА ІНТЕЛЕКТУАЛЬНОЇ КОМП'ЮТЕРНОЇ СИСТЕМИ УПРАВЛІННЯ ДОСТУПНІСТЮ ПАРКОМІСЦЬ | 463 |
| 72. | В. В. Яцишин, Ю. О. Рапацький, Вік. В. Яцишин
МЕТОДОЛОГІЯ QUALITY FUNCTION DEPLOYMENT У ПРОЦЕСІ ОПТИМІЗАЦІЇ РОЗРОБКИ КЛІЄНТ-СЕРВЕРНИХ КОМП'ЮТЕРНИХ СИСТЕМ | 464 |
| 73. | С. А. Жураковський, В. Ю. Олійник, В. Р. Ковалишин
ДОСЛІДЖЕННЯ СВІТОВОГО ДОСВІДУ ВПРОВАДЖЕННЯ НОВИХ МЕРЕЖЕВИХ ТЕХНОЛОГІЙ В КОНТЕКСТІ ІНДУСТРІЇ 5.0 | 465 |
| 74. | В. Р. Ковалишин, С. В. Марценко
ДОСЛІДЖЕННЯ ПЕРСПЕКТИВ ВИКОРИСТАННЯ ТЕХНОЛОГІЇ 5G В УКРАЇНІ | 466 |
| 75. | І. Р. Плавуцька, Сас Д. В.
ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ЯК ІННОВАЦІЇ У СФЕРІ АВТОМАТИЗАЦІЇ ТА КОМП'ЮТЕРНО-ІНТЕГРОВАНИХ ТЕХНОЛОГІЙ | 467 |
| 76. | І. Р. Плавуцька, Я. Р. Гриневич
РОБОТИЗАЦІЯ ТА АВТОМАТИЗАЦІЯ ЗАДЛЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ВИРОБНИЦТВА | 469 |
| 77. | В. Б. Сендецький, М. Ю. Степанюк, В. С. Форгель, І. Ю. Делів
ЗАДАЧА ПРОЕКТУВАННЯ АНТЕН ДЛЯ СИСТЕМ СУПУТНИКОВОГО ЗВ'ЯЗКУ | 471 |
| 78. | І. М. Недошвітко, М. В. Багрій, Я. В. Мельник, І. Ю. Делів
ЗАХИСТ ВІД КОМБІНОВАНИХ ЗАВАД ДЛЯ РАДІОЛОКАЦІЙНИХ СИСТЕМ | 472 |
| 79. | О. А. Делів, Я. В. Липницький, Л. Є. Делів, В. Г. Дозорський, О. Ф. Дозорська
ЗАДАЧА СИНХРОНІЗАЦІЇ ПРОЦЕДУРИ СВІЛЛОТЕРАПІЇ ІЗ РОБОТОЮ СЕРЦЕВО-СУДИННОЇ СИСТЕМИ | 473 |
| 80. | Б. В. Галенда, М. М. Кузнецов, Л. Є. Делів
ЗАДАЧА РОЗРОБЛЕННЯ СИСТЕМИ ОБМІНУ ДАНИМИ З ВІДКРИТИМ КАНАЛОМ | 474 |
| 81. | А. І. Маняк, І. Ю. Делів
СПОСІБ ПЕРЕДАЧІ СИГНАЛУ В СИСТЕМАХ СУПУТНИКОВОГО ЗВ'ЯЗКУ | 475 |

УДК 004.38

О. П. Ясній докт. техн. наук, І. В. Крисяк

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

ФАКТОРИ ВПЛИВУ НА НАДІЙНІСТЬ КОМП'ЮТЕРНИХ СИСТЕМ В ПРОЦЕСІ ЇХ РОЗРОБКИ

О. Р. Yasniy, Dr, Prof., I. V. Krysiuk

EFFECTS RELIABILITY FACTORS OF COMPUTER SYSTEMS IN THE PROCESS OF THEIR DEVELOPMENT

Надійність комп'ютерних систем у більшості випадків визначається за допомогою показників безвідмовної роботи її апаратного і програмного забезпечення протягом деякого періоду часу. При цьому доцільно застосовувати моделі надійності технічних засобів при обчисленні комплексного критерію характеристики надійності. Проте, для визначення надійності програмної складової комп'ютерної системи такі моделі не варто використовувати, оскільки вони мають різну природу відмов і збоїв.

Різниця між технічним і програмним забезпеченням полягає у реалізації складних і розгалужених відношень між структурними елементами на рівні програмного управління, а також відсутністю фізичного зносу. Збої та відмови у програмному забезпеченні можуть бути спровоковані лише змістом і кількістю дефектів, і також спричинені особливостями середовища експлуатації.

Експериментально встановлено, що кількість та інтенсивність відмов спадає у процесі визначення та ліквідації дефектів, а стійке функціонування програмної складової комп'ютерної системи забезпечується після 4-ох років експлуатації.

На рис. 1 продемонстровано візуальне представлення інтенсивності при апаратних і програмних відмовах протягом різного періоду використання комп'ютерних систем.

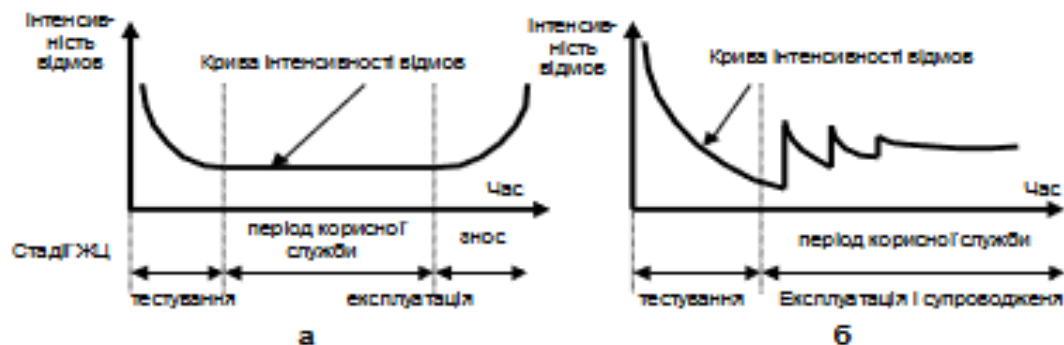


Рисунок 1. Інтенсивність відмов апаратного та програмного забезпечення

Дефекти, що призводять до збоїв програмного забезпечення пов'язані з помилками розробників на різних етапах створення продукту. У випадку, коли процедури перевірки та рев'ю коду, unit та інтеграційного тестування, залишили не визначені дефекти, то існує велика ймовірність того, що вони проявляться пізніше і як наслідок може бути спровокована ланцюгова реакція щодо некоректності роботи комп'ютерної системи в цілому.

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ ІМЕНІ ІВАНА ПУЛЮЯ**

МАТЕРІАЛИ

XI НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



13-14 грудня 2023 року

**ТЕРНОПІЛЬ
2023**

Ясків О.П., Крисюк І.В. ФАКТОРИ ВПЛИВУ НА НАДІЙНІСТЬ КОМП'ЮТЕРНИХ СИСТЕМ В ПРОЦЕСІ ЇХ РОЗРОБКИ Yaskiv O.P., Krysiuk I.V. EFFECTS RELIABILITY FACTORS OF COMPUTER SYSTEMS IN THE PROCESS OF THEIR DEVELOPMENT	161
Василь Яцишин, Іван Кучма КЛАСИФІКАЦІЯ ОНТОЛОГІЙ В ПРОЦЕСІ МОДЕЛЮВАННЯ КОМП'ЮТЕРНИХ МЕРЕЖ Vasyl Yatsyshyn, Ivan Kuchma CLASSIFICATION OF ONTOLOGIES IN THE PROCESS OF COMPUTER NETWORK MODELING	162
І.В. Лилік, А.М. Паламар КОМП'ЮТЕРИЗОВАНА СИСТЕМА МОНІТОРИНГУ РІВНЯ УЛЬТРАФІОЛЕТОВОГО ВИПРОМІНЮВАННЯ НА ОСНОВІ ІНТЕРНЕТУ РЕЧЕЙ I.V. Lylyk, A.M. Palamar COMPUTERIZED ULTRAVIOLET RADIATION LEVEL MONITORING SYSTEM BASED ON THE INTERNET OF THINGS	163
Андрій Луцьків, Сергій Макогон ТИПИ АРХІТЕКТУР НЕЙРОННИХ МЕРЕЖ ДЛЯ ПЕРЕТВОРЕННЯ ТЕКСТОВИХ ПОВІДОМЛЕНЬ У ЗВУКОВИЙ ПОТІК Andriy Lutskiv, Serhii Makohon TYPES OF NEURAL NETWORK ARCHITECTURES FOR TEXT TO SPEECH	164
Андрій Луцьків, Юрій Мельничук МУЛЬТИАГЕНТНА ОРГАНІЗАЦІЯ СЕРВЕРА ОНЛАЙН АУКЦІОНІВ Andriy Lutskiv, Yuriy Melnychuk MULTI-AGENCY ONLINE AUCTION SERVER ORGANIZATION	165
Галина Осухівська, Денис Муштин КОМП'ЮТЕРИЗОВАНА СИСТЕМА КОНТРОЛЮ ЗА МЕТЕОДАНИМИ ДЛЯ ОБПРИСКУВАЧА Halyna Osukhivska, Denys Mushryn COMPUTERIZED METEODATA CONTROL SYSTEM FOR SPRAYER	166
Т.А. Озарків, Р.О. Жаровський МЕТОД ОПТИМІЗАЦІЇ EIGRP ПРОТОКОЛУ ДЛЯ ПІДВИЩЕННЯ ПРОДУКТИВНОСТІ ПЕРЕДАЧІ ДАНИХ В КОМП'ЮТЕРНИХ МЕРЕЖАХ T. A. Ozarkiv; R.O. Zharovskyi THE METHOD OF OPTIMIZING THE EIGRP PROTOCOL TO INCREASE THE PRODUCTIVITY OF DATA TRANSMISSION IN COMPUTER NETWORKS	167
Андрій Луцьків, Андрій Острозький ОРГАНІЗАЦІЯ ДОСТУПУ ДО МОДЕЛІ GPT-3 ЗАСОБАМИ МОВИ PYTHON Andriy Lutskiv, Andriy Ostrovskiy ORGANIZING ACCESS TO THE GPT-3 MODEL USING PYTHON	168
А.М. Паламар, Р.О. Романчук, М.В. Дрогобицький КОМП'ЮТЕРИЗОВАНА СИСТЕМА ДЛЯ ДИСТАНЦІЙНОГО КОНТРОЛЮ РІВНЯ КОНЦЕНТРАЦІЇ ПИЛУ НА ОСНОВІ ІНТЕРНЕТУ РЕЧЕЙ A.M. Palamar, R.O. Romanchuk, M.V. Drohobytzkyi COMPUTERIZED SYSTEM FOR REMOTE MONITORING OF DUST CONCENTRATION LEVEL BASED ON THE INTERNET OF THINGS	169
Ярослав Паньчущин СТРУКТУРА СИСТЕМИ КОНТРОЛЮ ПАРАМЕТРІВ МІКРОКЛІМАТУ МІНІ-ТЕПЛИЦІ Yaroslav Panchushyn STRUCTURE OF THE MINI-GREENHOUSE MICROCLIMATE PARAMETER CONTROL SYSTEM	170

УДК 004.052.32

Олег Ясний докт. техн. наук, професор, Ігор Кривсюк

Тернопільський національний технічний університет імені Івана Пулюя

АРХІТЕКТУРА ЗАСОБУ РОЗРАХУНКУ ТА УПРАВЛІННЯ ПОКАЗНИКАМИ НАДІЙНОСТІ КОМП'ЮТЕРНИХ СИСТЕМ

Oleh Yasniy Dr., Prof., Ihor Krysiuk

ARCHITECTURE OF THE TOOL FOR CALCULATION AND MANAGEMENT OF COMPUTER SYSTEMS RELIABILITY INDICATORS

Одним з важливих аспектів розробки комп'ютерних систем є автоматизація процесу оцінювання їх надійності та визначення ефективності результатів імплементації системи. Це в подальшому забезпечить можливість управління та контролю за критеріями надійності та стабільністю функціонування комп'ютерної системи в цілому. Процес автоматизації представляє собою відображення методу визначення та розрахунку надійності компонентів КС за допомогою програмної імплементації на основі обраної технології. Зважаючи на те, що алгоритмічна складність при реалізації такої системи автоматизації є доволі високою, то це передбачає необхідність забезпечення максимальної ефективності при побудові її архітектури з врахуванням особливостей комунікації зовнішніх інтерфейсів та функціональних компонентів опрацювання критеріїв, що впливають на надійність системи.

На рис. 1 представлено архітектуру інструментального засобу обчислення показників надійності та оцінювання ефективності комп'ютерних систем.

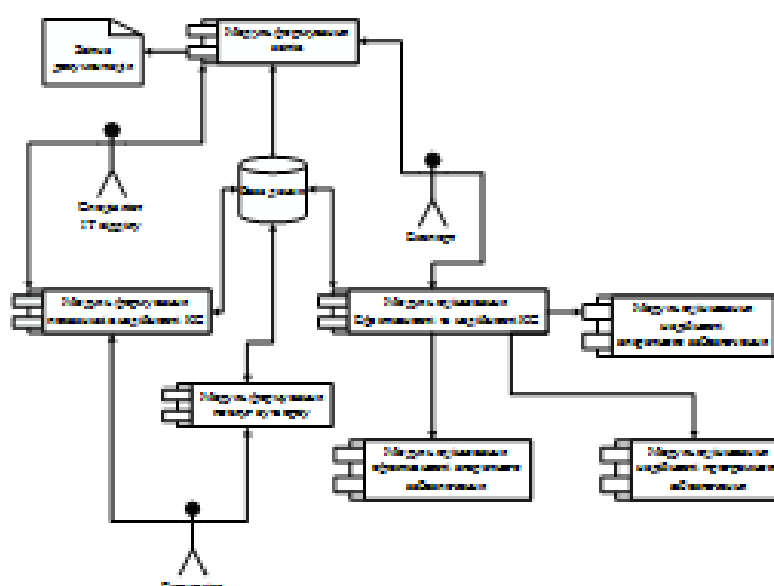


Рис. 1. Архітектура засобу обчислення надійності та ефективності комп'ютерних систем

Як видно з представленої на рис. 1 архітектури, компоненти системи можна умовно поділити на два класи: компоненти інтерфейсу та компоненти логічного опрацювання даних.