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)

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи)

магістра
(освітній ступінь (освітньо-кваліфікаційний рівень))

на тему: **Методи та засоби оптимізації розробки комп'ютерних систем критичного призначення**

Виконав: студент (ка) 6 курсу, групи СІмз-61
спеціальності (напряму підготовки) 123
«Комп'ютерна інженерія»
(шифр і назва спеціальності (напряму підготовки))
(підпис) Войтина Д.Я.
(прізвище та ініціали)

Керівник (підпис) Яцишин В.В.
(прізвище та ініціали)

Нормоконтроль (підпис) Тиш Є.В.
(прізвище та ініціали)

Рецензент (підпис) Чацкий Н.Б.
(прізвище та ініціали)

м. Тернопіль – 2019

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

Факультет Комп'ютерно інформаційних систем і прикладної інженерії
Кафедра Комп'ютерних систем та мереж
Освітній ступінь
Напрям підготовки
(шифр і назва)
Спеціальність 123 „Комп'ютерна інженерія"
(шифр і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
«02» 09 2019 р.

ЗАВДАННЯ
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Войтині Ліліану Ярославовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Методи та засоби системної розробки комп'ютерних систем критичного призначення

Керівник роботи Яцишин Василь Володимирович к.т.н. доц.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом по університету від «28» 08 2019 року №4/7-736

2. Термін подання студентом роботи 23.12.2019

3. Вихідні дані до проекту роботи Технічні та програмні засоби комп'ютерних систем критичного призначення, методи розробки сучасних систем

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1 Вступ. 2 Аналіз особливостей розробки комп'ютерних систем критичного призначення. 3 Побудова моделі і методу планування процесу розробки комп'ютерних систем критичного призначення. 4 Інструментальні засоби підтримки методу планування розробки комп'ютерних систем критичного призначення. 5 Організаційно економічне обґрунтування. 6 Охорона праці та безпека в надзвичайних ситуаціях. 7 Висновки. 8 Висновки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
Методи та засоби системної розробки комп'ютерних систем критичного призначення. Опис предмета методу системного аналізу. Побудова та аналіз системного методу покращення системи. Узагальнена модель розробки комп'ютерних систем. Функціональна послідовність методу системного аналізу. Інструментальні засоби системного аналізу. Діаграма Ганта. Діаграма послідовності етапів виконання робіт по системі підтримки методу системного аналізу розробки комп'ютерних систем.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата завдання видав	Підпис, дата прийняв
Економічне частина	[illegible]		
Перший спец. розділ	[illegible]		
Економічне	[illegible]		
Охорона праці	[illegible]		
Безпека в НС	Струнок В.С., ст. викл. каф. ОХ		

7. Дата видачі завдання 0.10.19

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів дипломного роботи	Термін виконання етапів роботи	Примітка
1	[illegible]	22.11.19	[illegible]
2	[illegible]	26.11.19	[illegible]
3	[illegible]	6.11.19	[illegible]
4	[illegible]	23.11.19	[illegible]
5	[illegible]	25.11.19	[illegible]
	[illegible]	27.11.19	[illegible]
6	Економічне	28.11.19	[illegible]
	[illegible]	23.12.19	

Студент _____ (підпис) Войтенко Т.О. (прізвище та ініціали)

Керівник роботи _____ (підпис) Дудченко Л.В. (прізвище та ініціали)

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)

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи)

Магістра
(освітній ступінь (освітньо-кваліфікаційний рівень))

на тему: **Методи та засоби оптимізації розробки комп'ютерних систем критичного призначення**

Виконав: студент (ка) 6 курсу, групи СІмз-61
спеціальності (напряму підготовки) 123
«Комп'ютерна інженерія»
(шифр і назва спеціальності (напряму підготовки))
(підпис) Войтина Д.Я. (прізвище та ініціали)

Керівник (підпис) Яцишин В.В. (прізвище та ініціали)

Нормоконтроль (підпис) Тиш Є.В. (прізвище та ініціали)

Рецензент (підпис) (прізвище та ініціали)

м. Тернопіль – 2019

АНОТАЦІЯ

Тема дипломної роботи: “Методи та засоби оптимізації розробки комп’ютерних систем критичного призначення” // Дипломна робота // Войтина Дмитро Ярославович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем та програмної інженерії, група СІмз-61 // Тернопіль, 2019// с. – 125, рис. – 53 , табл. –6, аркушів А1 – 10 , додат. – 1, бібліогр. – 30.

КЛЮЧОВІ СЛОВА: ОПТИМІЗАЦІЯ, МЕТОД, РОЗРОБКА, КОМП’ЮТЕРНА СИСТЕМА, КРИТИЧНА СИСТЕМА.

У дипломній роботі магістра досліджено методи і засоби реалізації та оптимізації процесу розробки комп’ютерних систем критичного призначення, виявлено особливості використання моделей життєвого циклу при розробці таких систем з використанням класичних підходів і гнучких методологій проектування комп’ютерних систем. Проведено аналіз процесів моніторингу за станом розробки комп’ютерних систем критичного призначення на етапах життєвого циклу та методів аналізу вартості проекту. Формалізовано процедуру організації процесів і критеріїв якості, які можна використовувати при розробці комп’ютерних систем критичного призначення на основі класичних моделей життєвого циклу та гнучких методологій.

У першому розділі дипломної роботи магістра проведено аналіз особливостей комп’ютерних систем критичного призначення і встановлено, що технології їх розробки є не гнучкими, довготривалими у часі і вимагають значних фінансових витрат. Окрім цього, проаналізовано процеси життєвого циклу комп’ютерних систем, встановлено вимоги до реалізації процесів згідно стандарту ISO/IEC 12207 та запропоновано шляхи їх оптимізації. Досліджено методи і засоби управління та моніторингу процесів створення комп’ютерних систем критичного призначення, що дало змогу обґрунтувати необхідність

застосування класичних підходів і гнучких методологій розробки комп’ютерних систем.

У другому розділі дипломної роботи магістра проведено аналіз особливостей застосування гнучких методологій розробки комп’ютерних систем, визначено основні їх переваги і недоліки, і як результат запропоновано можливі шляхи оптимізації щодо їхнього застосування при розробці комп’ютерних систем критичного призначення на основі впровадження параметрів якості результатів виконання кожної ітерації. Формалізовано процедуру організації процесів і критеріїв якості, які можна використовувати при розробці комп’ютерних систем критичного призначення на основі класичних моделей життєвого циклу та гнучких методологій, а також запропоновано механізм контролю (метод) за рівнем відповідності реалізованих у комп’ютерній системі властивостей.

У третьому розділі на основі результатів аналізу предметної області та особливостей запропонованого методу, визначено функціональні вимоги до програмного засобу підтримки процесу оптимізації розробки комп’ютерних систем критичного призначення, спроектовано базу даних і побудовано архітектуру програмного засобу, а також розроблено інтерфейси програмного засобу для забезпечення зручності виконання робіт щодо оптимізації процесів розробки комп’ютерних систем критичного призначення.

У четвертому розділі обґрунтовано економічну доцільність розробки методів і засобів оптимізації розробки комп’ютерних систем критичного призначення шляхом розрахунку показників економічної ефективності, що дало змогу протягом приблизно двох років компенсувати витрати на використання запропонованих технічних рішень.

П’ятий розділ присвячено аналізу вимог охорони праці і техніки безпеки при використанні комп’ютерної техніки, визначено шляхи мінімізації негативного впливу при використанні програмного засобу підтримки процесу оптимізації розробки комп’ютерних систем критичного призначення, а також розглянуто питання забезпечення надійності управління і матеріально-технічного постачання суб’єкта господарювання.

У шостому розділі проведено аналіз комплексної оцінки екологічності виробництва та методів узагальнення екологічної інформації.

ABSTRACT

The theme of the thesis: «Methods and tools of critical computer systems development»//Master thesis// Voityna Dmytro Yaroslavovych / Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and software engineering, CImz-61 // Ternopil, 2019 // p. – 125, fig. – 53, table. – 6, Sheets A1 - 10, Add. – 1, Ref. - 30.

KEYWORDS: OPTIMIZATION, METHOD, DEVELOPMENT, COMPUTER SYSTEM, CRITICAL SYSTEM.

The master's thesis investigates methods and means of implementation and optimization of the process of development of computer systems of critical purpose, reveals peculiarities of using life cycle models in the development of such systems using classical approaches and flexible methodologies for designing computer systems. The process of monitoring of the development of computer systems of critical purpose at the stages of the life cycle and methods of cost analysis of the project are analyzed. We have formalized a process for organizing processes and quality criteria that can be used to develop critical mission systems based on classic life cycle models and flexible methodologies.

The first chapter of the master's thesis deals with the analysis of the features of computer systems for critical applications and found that the technologies for their development are not flexible, time-consuming and costly. In addition, the computer systems life cycle has been analyzed, the requirements for the implementation of processes in accordance with ISO / IEC 12207 have been established and ways to optimize them have been proposed. The methods and tools for managing and monitoring the processes of creation of computer systems for critical purposes were investigated, which made it possible to substantiate the necessity of applying classical approaches and flexible methodologies for the development of computer systems.

In the second chapter of the master's thesis the analysis of features of application of flexible methodologies of computer systems development is carried

out, their main advantages and disadvantages are determined, and as a result the possible ways of optimization on their application in the development of critical computer systems on the basis of implementation quality parameters of performance results are proposed on each iteration. The process of organizing the processes and quality criteria that can be used in the development of computer-based critical systems based on classic life-cycle models and flexible methodologies is formalized, and a control mechanism (method) for the level of compliance of computer-implemented properties is proposed.

In the third chapter, based on the results of the analysis of the subject area and features of the proposed method, the functional requirements for the software support for the process of optimization of the development of computer systems of critical purpose are determined, the database is designed and the architecture of the software is developed, as well as the software interfaces are developed for convenience of execution works on optimization of processes of development of computer systems of critical purpose.

The fourth chapter substantiates the economic feasibility of developing methods and tools for optimizing the design of computer critical systems by calculating cost-effectiveness indicators, which made it possible to offset the costs of using the proposed technical solutions for approximately two years.

The fifth chapter is devoted to the analysis of the requirements of occupational safety and health when using computer technology, identifies ways to minimize the negative impact of using software to support the process of optimizing the development of computer systems for critical purposes, and also considers the issues of ensuring the reliability of management and logistics supply of the entity.

In the sixth chapter, an analysis of the integrated environmental assessment of production and methods of generalizing environmental information is analyzed.

ЗМІСТ

ПОЯСНЮВАЛЬНА ЗАПИСКА 4

ПЕРЕЛІК ОСНОВНИХ УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ І СКОРОЧЕНЬ 14

ВСТУП 15

РОЗДІЛ 1 АНАЛІЗ ОСОБЛИВОСТЕЙ РОЗРОБКИ КОМП'ЮТЕРНИХ СИСТЕМ КРИТИЧНОГО ПРИЗНАЧЕННЯ 19

1.1. Аналіз особливостей комп'ютерних систем критичного призначення... 19

1.2. Аналіз процесів життєвого циклу комп'ютерних систем 25

1.3. Аналіз методів управління процесом розробки комп'ютерних систем критичного призначення 30

1.4. Аналіз процесу моніторингу проекту зі створення комп'ютерних систем критичного призначення 34

1.5. Аналіз методів оцінювання вартості проекту 35

1.6. Висновки до розділу 38

РОЗДІЛ 2 ПОБУДОВА МОДЕЛІ І МЕТОДУ ОПТИМІЗАЦІЇ ПРОЦЕСУ РОЗРОБКИ КОМП'ЮТЕРНИХ СИСТЕМ КРИТИЧНОГО ПРИЗНАЧЕННЯ 39

2.1. Аналіз моделей життєвого циклу комп'ютерних систем критичного призначення 39

2.1.1. Waterfall-модель життєвого циклу 39

2.1.2. Модель життєвого циклу критичних комп'ютерних систем на основі інкрементів 41

2.1.3. Особливості моделі життєвого циклу Боема при розробці комп'ютерних систем критичного призначення 43

2.1.4. «Еволюційна модель» 44

2.1.5. Аналіз «V-подібної» моделі життєвого циклу розробки комп'ютерних систем критичного призначення 46

2.2. Обґрунтування методологій гнучкої розробки для оптимізації процесів реалізації комп'ютерних систем критичного призначення 49

2.2.1. Особливості гнучких методологій при розробці комп'ютерних систем критичного призначення 49

2.2.2. Методологія «SCRUM» 52

2.2.3. Метод розробки динамічних систем 54

2.3. Формалізація процесу оптимізації розробки комп'ютерних систем критичного призначення 57

2.4. Побудова моделі і методу оптимізації розробки комп'ютерних систем критичного призначення 61

2.5. Висновки по розділу 66

РОЗДІЛ 3 ПРОГРАМНИЙ ЗАСІБ ПІДТРИМКИ МЕТОДУ ОПТИМІЗАЦІЇ РОЗРОБКИ КОМП'ЮТЕРНИХ СИСТЕМ КРИТИЧНОГО ПРИЗНАЧЕННЯ 67

3.1. Аналіз предметної області 67

3.2. Проектування архітектури засобу підтримки методу оптимізації процесу розробки комп'ютерних систем критичного призначення 70

3.3. Проектування схеми бази даних при оптимізації процесу розробки комп'ютерних систем критичного призначення 72

3.4. Розробка інтерфейсів програмного засобу підтримки методу оптимізації процесів розробки комп'ютерних систем критичного призначення 80

3.5. Висновки до розділу 87

РОЗДІЛ 4 ОБГРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ 88

4.1. Розрахунок норм часу на виконання науково-дослідної роботи 88

4.2. Визначення витрат на оплату праці та відрахувань на соціальні заходи 91

4.3. Розрахунок витрат на електроенергію 93

4.4. Розрахунок витрат на матеріали 94

4.5. Розрахунок суми амортизаційних відрахувань 94

4.6. Обчислення накладних витрат 95
4.7. Складання кошторису витрат та визначення собівартості науково-дослідних робіт 96
4.8. Розрахунок ціни науково-дослідних робіт 97
4.9. Визначення економічної ефективності і терміну окупності капітальних вкладень 98
РОЗДІЛ 5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ 100
5.1. Охорона праці 100
5.2. Забезпечення надійності управління і матеріально-технічного постачання суб’єкта господарювання 103
5.3. Проведення розрахунку режимів роботи об’єкту в умовах радіоактивного забруднення при аваріях на АЕС 107
РОЗДІЛ 6 ЕКОЛОГІЯ 112
6.1. Комплексна оцінка екологічності виробництва 112
6.2. Методи узагальнення екологічної інформації 114
ВИСНОВКИ 117
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 120
Додаток А Тексти наукових публікацій дипломної роботи магістра 123

ПЕРЕЛІК ОСНОВНИХ УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ І СКОРОЧЕНЬ

БД	База даних
ЖЦ	Життєвий цикл
ПЗ	Програмне забезпечення
КС	Комп’ютерна система
CPM	Critical Path Method
CASE	Computer Aided Software Engineering
ER	Entity Relations
PERT	Project Evaluation and Review Technique
UML	Unified Modeling Language

ВСТУП

Актуальність теми. Сучасний стан розвитку галузі комп’ютерної інженерії характеризується впровадженням новітніх технологій, методів і засобів розробки, що дозволяють швидко реалізувати та впровадити технічні рішення у вигляді комп’ютерних систем для автоматизації бізнес-діяльності будь-якої сфери. При цьому такі системи орієнтовані на максимальне задоволення кінцевого споживача у максимально короткі терміни та з максимальним прибутком для замовника.

Сучасні методи і технології швидкої розробки компонентів комп’ютерних систем задовольняють вимоги щодо часу реалізації проекту і забезпечення необхідної функціональності системи. Однак рівень якості та надійності кінцевих продуктів все ж потребує вдосконалення, особливо при реалізації комп’ютерних систем критичного призначення. До систем критичного призначення належать системи керування польотами, засоби діагностики та лікування людини, комп’ютерні системи керування атомними станціями та інші системи, які безпосередньо впливають на життя і здоров’я людей та навколишнього середовища.

Більшість моделей і технологій проектування комп’ютерних систем критичного призначення базуються на моделі «фабрики» (водоспадної моделі життєвого циклу), тобто використовується лінійний процес розробки у якому кожна наступна стадія не розпочинається до закінчення попередньої. Це негативно позначається на термінах розробки комп’ютерних систем, а також не забезпечує гнучкості та масштабованості внесення змін у проект.

З іншого боку, ефективним підходом до реалізації комп’ютерних систем, є застосування методологій розробки: Agile Modeling, Agile Unified Process, Dynamic System Development Method та інших. Перевагою таких технологій є участь замовників або користувачів у процесі розробки комп’ютерних систем, гнучкість процесу розробки, адаптивність процесу при зміні вимог, швидка доставка прототипів системи. Проте, ці технології мають і ряд недоліків, які не дають змогу прямо їх використовувати при проектуванні комп’ютерних систем

критичного призначення. До них належать слабке документування на стадіях розробки системи, відсутність або недосконалість процесу планування ітерацій, низький рівень методів і засобів забезпечення надійності та якості кінцевого продукту.

Дослідженню методів і засобів оптимізації процесів розробки комп'ютерних систем присвячено праці як українських, так і закордонних науковців, зокрема, Лавріщевої К.М., Андона П.І., Дідковської М.В., Боема Б., МакКола Дж., Рейтера Д. та ін. Однак, не зважаючи на ефективність запропонованих науковцями підходів і методів, все ж вдосконалення та оптимізації потребують процеси розробки комп'ютерних систем критичного призначення. Виходячи з цього, актуальними задачами, в контексті оптимізації процесу розробки комп'ютерних систем критичного призначення, є інтеграція методів і засобів гнучких методологій у класичний процес розробки комп'ютерних систем критичного призначення із дотриманням високих вимог щодо їх надійності та якості в цілому.

Мета дипломної роботи полягає у дослідженні методів і засобів оптимізації процесів розробки комп'ютерних систем критичного призначення.

Для досягнення цієї мети у дипломній роботі магістра були поставлені і вирішені **наступні задачі**:

– аналіз наукових публікацій та практик розробки комп'ютерних систем на основі класичних і гнучких методологій проектування;

– визначення особливостей процесів створення комп'ютерних систем критичного призначення;

– побудова та обґрунтування моделі процесу розробки комп'ютерних систем критичного призначення;

– розробка методу оптимізації процесів розробки комп'ютерних систем критичного призначення;

– проектування та реалізація архітектури програмного засобу управління критеріями оптимізації процесу розробки комп'ютерних систем критичного призначення.

Методи дослідження. Для вирішення поставлених задач використано наступні методи: аналіз та узагальнення – при проведенні аналізу існуючих моделей розробки комп'ютерних систем критичного застосування та гнучких методологій їх проектування; формалізації – при побудові моделі і методу оптимізації процесу розробки комп'ютерних систем; проектування та програмування – при розробці програмного засобу підтримки запропонованого методу; експеримент та вимірювання – для апробації розробленого методу і засобу автоматизації.

Об'єктом дослідження є процеси оптимізації розробки комп'ютерних систем критичного призначення.

Предметом дослідження є критерії оптимізації, моделі, методи і засоби розробки комп'ютерних систем критичного призначення.

Наукова новизна одержаних результатів.

– уперше формалізовано процедуру організації процесів і критеріїв якості, які можна використовувати при розробці комп'ютерних систем критичного призначення на основі класичних моделей життєвого циклу та гнучких методологій, що дало змогу оптимізувати часові та якісні характеристики виконання проектів;

– уперше запропоновано механізм контролю (метод) за рівнем відповідності реалізованих у комп'ютерній системі властивостей, що дає змогу застосовувати метод DSDM і методологію Scrum при реалізації комп'ютерних систем критичного призначення.

Практична цінність результатів дослідження. Впровадження програмного засобу підтримки процесів розробки комп'ютерних систем критичного призначення дає змогу оптимізувати за часовими і якісними характеристиками процес їх реалізації та інтегрувати Agile-методології у класичні моделі життєвого циклу.

Публікації. Результати дослідження апробовано на VIII міжнародній науково - технічній конференції молодих учених і студентів «Актуальні задачі сучасних технологій» (27-28 листопада 2019 р.) Тернопільського національного технічного університету імені Івана Пулюя та на VII науково-технічній

конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (11-12 грудня 2019 року) у вигляді тез конференцій:

1. Яцишин В.В., Войтина Д.Я. Оптимізація процесів розробки комп'ютерних систем критичного призначення. Матеріали VIII Міжнародної науково-технічної конференції молодих учених та студентів «Актуальні задачі сучасних технологій» (27 – 28 листопада 2019 р.). Тернопіль:ТНТУ. 2019. с. 128.

2. Яцишин В.В., Войтина Д.Я. Аналіз особливостей комп'ютерних систем критичного призначення. Матеріали VII науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (11-12 грудня 2019 року). Тернопіль: ТНТУ. 2019. с. 27.

Структура роботи. Робота складається з пояснювальної записки та графічної частини. Пояснювальна записка складається з вступу, 6 розділів, висновків, списку використаної літератури та додатків. Обсяг роботи: пояснювальна записка – 125 арк. формату А4, графічна частина – 10 аркушів формату А1.

РОЗДІЛ 1
АНАЛІЗ ОСОБЛИВОСТЕЙ РОЗРОБКИ КОМП'ЮТЕРНИХ СИСТЕМ КРИТИЧНОГО ПРИЗНАЧЕННЯ

1.1. Аналіз особливостей комп'ютерних систем критичного призначення

Характерною особливістю критичних об'єктів, в тому числі і комп'ютерних систем, є те, що у випадках збоїв чи відмов їх роботи, які супроводжуються порушенням або втратою управління над системою, можуть призводити до значних негативних впливів на навколишнє середовище, життя і здоров'я людей або інших наслідків. Комп'ютерні системи, які застосовуються у військовій сфері, на потенційно небезпечних хімічних підприємствах, атомних станціях, на транспорті та в інших сферах, що можуть провокувати негативний вплив на біологічні об'єкти чи приводити до значних збитків належать до критичних комп'ютерних систем.

Сьогодні спостерігається широке використання та розповсюдження комп'ютерних систем критичного призначення у різних сферах діяльності, що супроводжується використання апаратних і програмних компонентів, які забезпечують функції запобігання та захисту від виникнення небезпечних ситуацій і дають змогу уникнути або зменшити негативний вплив на навколишнє середовище.

Як приклад, наведемо медичну систему контролю за інсуліном в організмі людини. Інсуліновий автоматизований пристрій – це медична система, яка імітує роботу підшлункової залози. Програмне забезпечення, що управляє цією системою представляє собою вбудовану критичну комп'ютерну систему, яка збирає інформацію з датчика і управляє насосом, який подає визначену дозу інсуліну для пацієнта.

Люди, які страждають на діабет, використовують цю систему. Діабет є відносно поширеним явищем, що описує стан, коли підшлункова залоза людини не здатна виробляти достатню його кількість і вимагає введення гормону інсулін. Інсулін метаболізує глюкозу (цукор) у крові. Традиційне

лікування діабету включає регулярні ін'єкції інсуліну. Діабетики вимірюють рівень цукру в крові за допомогою зовнішнього пристрою, а потім обчислюють дозу інсуліну, яку їм слід вводити.

Проблема цього лікування полягає в тому, що необхідний рівень інсуліну не просто залежать від рівня глюкози в крові, але і від часу останньої ін'єкції інсуліну. Це може призвести до дуже низького рівня глюкози в крові, якщо занадто багато інсуліну, або дуже високий рівень цукру в крові –якщо інсуліну занадто мало. Низький рівень глюкози в крові є більш небезпечним станом, оскільки це може призвести до тимчасової несправності мозку і, зрештою, втрати свідомості і смерті. У перспективі, постійно високий рівень глюкози в крові може призвести до ураження очей, ураження нирок та проблем із серцем. Тому така система є вбудованою комп’ютерною системою критичного призначення, оскільки безпосередньо впливає на стан хворого, а збої або відмови у роботі системи можуть призвести до летального випадку. На рис. 1.1 представлено архітектуру комп’ютерної системи контролю інсуліну в організмі людини.

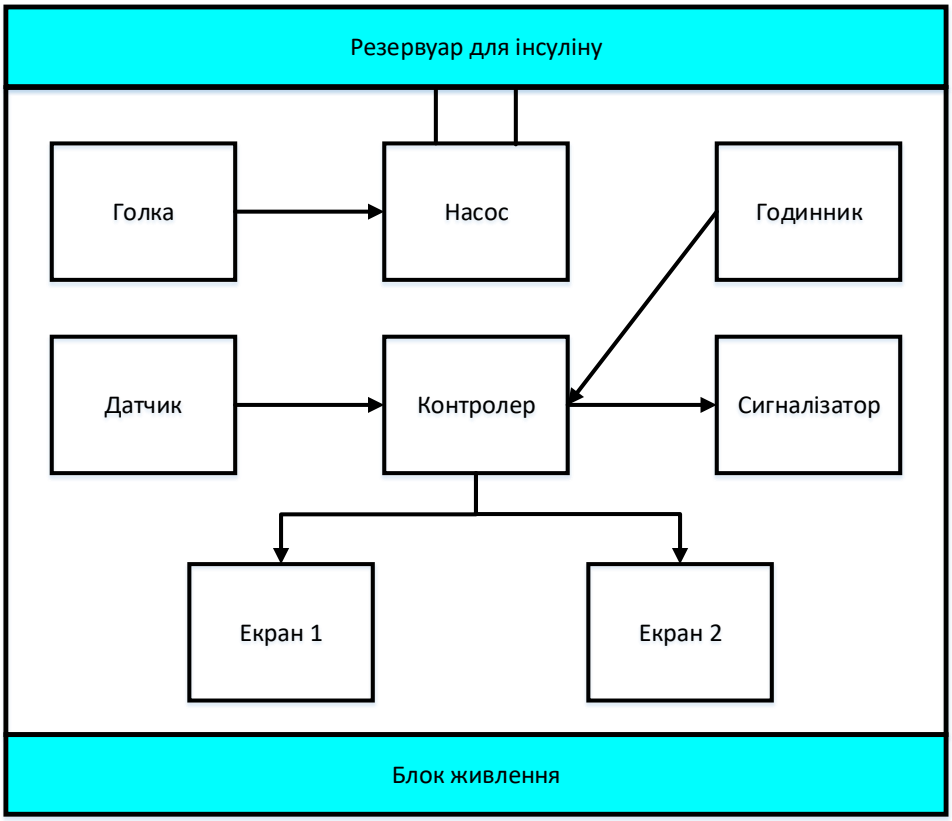

Рис. 1.1. Приклад комп’ютерної системи критичного застосування

Сучасні досягнення в розробці мініатюризованих датчиків означають, що зараз можна розробити автоматизовані системи подачі інсуліну. Ці системи контролюють рівень цукру в крові і можуть подавати відповідну дозу інсуліну, коли потрібно. Системи доставки інсуліну вже існують для лікування пацієнтів лікарні. В майбутньому для багатьох діабетиків ці системи можуть бути постійно прикріплені до свого тіла.

Система постачання інсуліну, що управляється програмним забезпеченням, може працювати, використовуючи вбудований у пацієнта мікросенсор для вимірювання параметрів крові, пропорційного до рівня цукру. Потім він надсилається до контролера насоса. Контролер обчислює рівень цукру та необхідну кількість інсуліну. Потім він надсилає сигнали до мініатюрний насос для доставки інсуліну через постійно прикріплену голку.

На рис 1.1 наведено апаратні компоненти та схему організації системи контролю введення інсуліну. При цьому, датчик крові вимірює характеристики крові при різних умовах, і що ці значення можуть бути пов'язані з рівнем цукру в крові. Інсуліновий насос доставляє одну одиницю інсуліну у відповідь на один імпульс від контролера. Тому, щоб доставити 10 одиниць інсуліну, контролер посилає 10 імпульсів. На рис. 1.2 наведено UML діаграму видів діяльності, яка ілюструє, як програмне забезпечення перетворює вхідний рівень цукру крові у послідовність команд, які забезпечують управління подачі інсуліну.

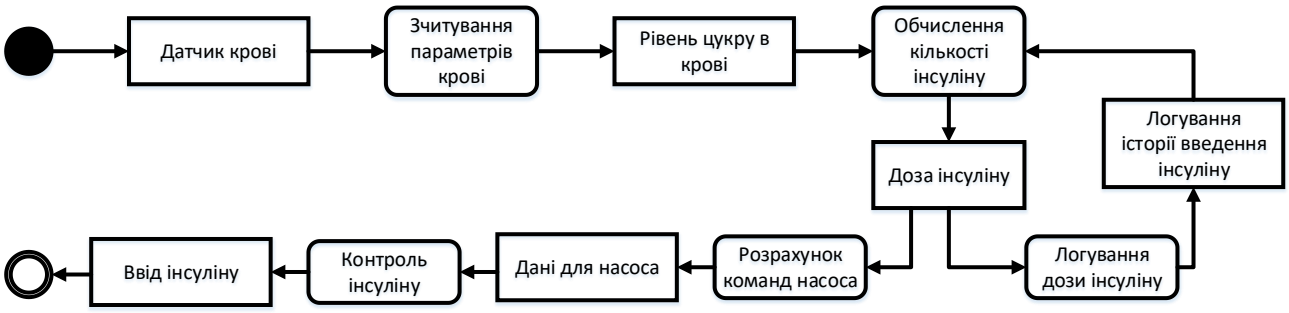

Рис. 1.2. Діаграма видів діяльності при проектуванні комп’ютерної системи критичного призначення

Зрозуміло, що це критично важлива для безпеки система. Якщо насос не працює або працює неправильно, тоді здоров'я користувача може бути під загрозою, що провокує стан коми у зв'язку із занадто високим або надто низьким рівнем інсуліну.

Отже, дві основні вимоги високого рівня, яким повинна відповідати ця система:

– система повинна бути доступною для доставки інсуліну у потрібні моменти часу.

– система повинна надійно працювати і доставляти потрібну кількість інсуліну для протидії поточному рівню цукру в крові.

У випадку використання комп'ютерних систем критичного призначення, за «законами» Лемана компоненти системи можуть перебувати у двох станах:

– процеси внесення змін;

– процес деградації

У роботах [1-3] досліджено, що у процесі супроводу комп'ютерних систем близько 20% часу використовується на усунення дефектів та інших типів помилок, які не були виявленні у процесі розробки і тестування.

За статистикою понад 80% витрат у процесі супроводу комп'ютерних систем критичного призначення, пов'язані з удосконаленням шляхом додавання нових функцій або зміни існуючих.

Найбільш затратним процесом життєвого циклу комп'ютерних систем критичного призначення є процес супроводу. Згідно означень стандарту ISO/IEC 14764, існує 4 види супроводу КС:

– адаптивний супровід (adaptive maintenance) – модифікація комп'ютерної системи і програмних складових, що виконується після введення її в експлуатацію, для збереження придатності і функціональності комп'ютерної системи при внесенні змін у середовищі використання;

– коригуючий супровід (corrective maintenance) – процес супроводу, що характеризується внесенням змін у комп'ютерну систему, що пов'язано з невідповідністю вимог та реалізованих властивостей у КС;

– супровід удосконалення (perfective maintenance) – модифікація комп’ютерної системи після введення в експлуатацію з метою виявлення прихованих помилок, які в майбутньому можуть негативного впливати на надійність системи;

– профілактичний супровід – модифікація комп’ютерної системи з метою виявлення та усунення прихованих дефектів до моменту, коли дефект проявить себе, забезпечення та підвищення здатності комп’ютерної системи до гнучкості та масштабованості під час супроводу.

Процеси супроводу комп’ютерних систем критичного призначення є важливими в контексті їх еволюції та адаптації до нових середовищ експлуатації, розширення функціональних можливостей та ін.

Розробники комп’ютерних систем для забезпечення здатності до еволюціонування таких систем повинні використовувати методи, технології та інструментальні засоби у процесах проектування та супроводу, що давали б змогу ефективно керувати, проводити моніторинг як функціональних, так і нефункціональних критеріїв, особливо щодо гарантоздатності і продуктивності.

Необхідність та важливість критичних комп’ютерних систем до еволюції зумовлено рядом факторів, які стосуються:

– розповсюдження та широкого використання комп’ютерних систем – комп’ютерні системи масово увійшли у побут людей, а збої та відмови у їх функціонуванні безпосередньо впливають на комфорт або життя суспільства;

– комп’ютерні системи і їх складові все більше стають критичними;

– людський фактор, що може зумовлювати негативний вплив на ефективність використання комп’ютерних систем критичного призначення і вимагає застосування все більш складних механізмів людино-машинної взаємодії;

– структурна складність компонентів комп’ютерних систем доволі часто полягає у використанні стороннього зовнішнього впливу та інтеграції з іншими системами, що може негативно позначатись на надійності вузлів і мати критичний вплив на користувачів;

– висока швидкість реакції на ринку комп’ютерних систем вимагає застосування технологій і засобів швидкого внесення змін за короткий проміжок часу та з мінімальними витратами.

Доволі часто, до підкласу критичних систем відносяться системи реального часу, які згідно [2] є «системами, які повинні реагувати на події у зовнішньому по відношенню до системи середовищі або впливати на середовище в межах необхідних тимчасових обмежень».

У зв’язку із стрімким розвитком технологій комп’ютерні системи критичного призначення впроваджуються у різні сфери, зокрема у промисловість, що передбачає управління технологічними процесами без участі людини, системи промислової автоматики, SCADA-системи, випробувальне та вимірювальне обладнання, робототехніку [2].

У медицині комп’ютерні системи критичного призначення та реального часу використовують у:

– томографії;
– радіотерапії;
– приліжковому моніторингу.

У транспортній галузі критичні комп’ютерні системи реального часу використовуються у:

– бортових самописцях;
– системах керування транспортним потоком;
– управління диспетчерськими пунктами повітряного простору;
– аерокосмічній техніці.

Прикладами критичних комп’ютерних систем реального часу є:

– автоматизована система керування реакторами;
– бортова система управління космічного апарату;
– система обробки аудіо-і відеопотоків при трансляції в прямому ефірі;
– інтерактивна комп'ютерна гра.

Комп’ютерні системи критичного призначення набувають великої складності та розмірів, а вимоги до них постійно змінюються, що зумовлює необхідність оптимізації процесів розробки таких систем.

1.2. Аналіз процесів життєвого циклу комп’ютерних систем

За роки формування галузі комп’ютерної інженерії напрацьовано багато алгоритмів та схем щодо впорядкування процесів розробки, які у кінцевому результаті описані у вигляді стандартів, що визначають сукупність процесів, їх характеристики і порядок виконання.

Життєвий цикл комп’ютерних систем критичного призначення, зокрема їх програмних складових, описано в стандарті [5] та реалізовані у більшості моделей ЖЦ.

У стандарті [5] обгрунтовано та описано загальну структуру і суть ЖЦ КС, що починається з виникнення ідеї про створення системи та закінчується виводом її з експлуатації. Даним стандартом передбачено опис характеристик процесів, залежності між ними та сукупності операцій, які необхідно виконати при реалізації комп’ютерних систем. Стандарт не визначає способів і засобів виконання етапів проектування та послідовності їх виконання. У табл. 1.1 наведено категорії процесів ЖЦ комп’ютерних систем, які запропоновано у [5].

Таблиця 1.1

Процеси ЖЦ у стандарті [5]

№ з/п	Процес	
1. «Основні процеси»		
1.1	Замовлення	
	1.1.1	Підготовка замовлення, вибір постачальника
	1.1.2	Моніторинг діяльності постачальника, приймання споживачем
1.2	Постачання (придбання)	
1.3	Розроблення	

Продовж. табл. 1.1.

№ з/п	Процес
1. «Основні процеси»	
1.3.1	Виявлення вимог
1.3.2	Аналіз вимог до системи
1.3.3	Проектування архітектури системи
1.3.4	Аналіз вимог до ПЗ системи
1.3.5	Проектування ПЗ
1.3.6	Конструювання (кодування) ПЗ
1.3.7	Інтеграція ПЗ
1.3.8	Тестування ПЗ
1.3.9	Системна інтеграція
1.3.10	Системне тестування
1.3.11	Інсталяція ПЗ
1.4	Експлуатація
1.4.1	Функціональне використання
1.4.2	Підтримка споживача
1.5	Супроводження
2. «Процеси підтримки»	
2.1	Документування
2.2	Керування конфігурацією
2.3	Забезпечення гарантії якості
2.4	Верифікація
2.5	Валідація
2.6	Загальний огляд
2.7	Аудит
2.8	Вирішення проблем
2.9	Забезпечення застосовності продукту
2.10	Оцінювання продукту
3.1	Керування

Продовж. табл. 1.1

3. «Організаційні процеси»	
3.1.1	Керування на рівні організації
3.1.2	Керування проектом
3.1.3	Керування якістю
3.1.4	Керування ризиком
3.1.5	Організаційне забезпечення
3.1.6	Вимірювання
3.1.7	Керування знаннями
3.2	Удосконалення
3.2.1	Упровадження процесів
3.2.2	Оцінювання процесів
3.2.3	Удосконалення процесів

Основна проблема, що стосується критичних систем – це наслідок збою або відмови системи. Якщо збій системи може призвести до наслідків, які визнаються неприйнятними, тоді система є критично важливою.

Критичні комп’ютерні системи повинні бути сертифікованим регуляторним органом, щоб переконатися у їх придатності до застосування. Це забезпечує належну практику розвитку таких систем для сприяння правильності функціонування. Стандарти у галузі комп’ютерних систем критичного призначення визначають цілі, яких повинна досягати система чи проект, перш ніж вона буде розгорнута у середовищі використання.

Хоча стандарти визначають цілі, яким повинен відповідати будь-який процес розробки критичних систем, однак стандарти не відображають засобів реалізації цих процесів [1].

Команда розробників може використовувати будь-які методи і технології реалізації процесів розробки комп’ютерних систем критичного призначення. Це дає змогу застосовувати гнучкі методи і методології при умові дотримання визначеного рівня надійності, безпеки та ін.

У стандарті, що визначає процеси життєвого циклу комп'ютерних систем розглядається 3 категорії процесів:

– основні процеси;

– процеси підтримки;

– організаційні процеси.

Кожен процес характеризується видами діяльності, можливими шляхами розв'язання задач і вихідними результатами.

До основних процесів при розробці комп'ютерних систем критичного призначення належать:

– процес купівлі – визначає сукупність дій, пов'язаних із поведінкою замовника щодо придбання комп'ютерної системи;

– процес постачання – регулює передачу розробленої комп'ютерної системи замовнику на основі таких видів діяльності як: розробка пропозицій щодо реалізації, підготовка контракту на постачання, терміни планового початку і закінчення робіт та ін.

– процес розроблення – містить рекомендації щодо властивостей процесів безпосередньої реалізації комп'ютерної системи і передбачає виконання підпроцесів. Детальну схема основних процесів ЖЦ наведено на рис. 1.3;

Рис. 1.3. Основні процеси ЖЦ КС

– процес використання – сукупність видів діяльності щодо забезпечення ефективності експлуатації комп’ютерної системи, який включає її обслуговування, надання консультацій користувачам, визначення їхніх потреб та ін.

– процес супроводу – характеризується виконанням видів діяльності щодо управління версіями компонентів комп’ютерних систем, моніторингу функціональної придатності складових системи, інсталяції та утилізації програмного забезпечення.

Допоміжні або додаткові процеси життєвого циклу комп’ютерних систем передбачають сукупність операцій щодо перевірки реалізованих у системі властивостей, менеджменту етапів виконання та управління якістю компонентів. Додаткові процеси стандарту [5] наведено на рис.1.4.

Процеси, що виконують функції допоміжних при розробці програмних складових комп’ютерних систем, включають:

– розробку та аналіз документації на стадіях ЖЦ;
– управління версійністю;
– валідацію та верифікація;
– рев’ю коду, процедури забезпечення якості прототипів системи та ін.

Керування версійністю апаратної і програмної складової комп’ютерних систем відображає суть процесів управління конфігурацією і спрямована на досягнення кінцевих цілей проекту та відповідності вимогам замовників.

Група організаційних процесів життєвого циклу комп’ютерних систем містить опис процесів щодо управління проектом розробки, забезпечення та оцінювання його якості, керування ризиками і т.п.

Реалізацію групи організаційних процесів у проекті розробки комп’ютерних систем виконують члени команди, функції яких полягають у плануванні та моніторингу діяльності щодо реалізації групи основних та допоміжних процесів, обґрунтуванні критеріїв і метрик оцінювання стану виконання проекту.

Рис. 1.4. Група додаткових процесів ЖЦ

Між методами, методологіями і технологіями проектування комп'ютерних систем та розглянутими процесами життєвого циклу немає тісного зв'язку, оскільки перші є інструментами досягнення цілей, описаних у стандарті.

Таким чином, при проектуванні комп'ютерних систем критичного призначення, можливе застосування як класичних підходів до їх розробки, так і гнучких методологій і їх комбінацій.

1.3. Аналіз методів управління процесом розробки комп'ютерних систем критичного призначення

Проведемо аналіз існуючих методів щодо управління процесом розробки комп'ютерних систем з акцентом на визначення їх переваг і недоліків. На практиці широкого застосування набули такі методи керування як:

– Critical Path Method;
– Program Evaluation and Review Technique.

Перший метод запропоновано фахівцями фірми Dupon для забезпечення ефективності процесів планування робіт щодо модернізації заводів з

використанням обчислювальної техніки. Авторами цього методу є Уолкер – Келлі.

В основі запропонованого методу лежить поняття критичного шляху, який описується найдовшою послідовністю дій від початкової діяльності до кінцевої. Для прикладу, візьмемо граф, який показано на рис. 1.5. Критичним шляхом в даному випадку буде шлях «Початок робіт – А1 – А3 – А6 – Кінець робіт».

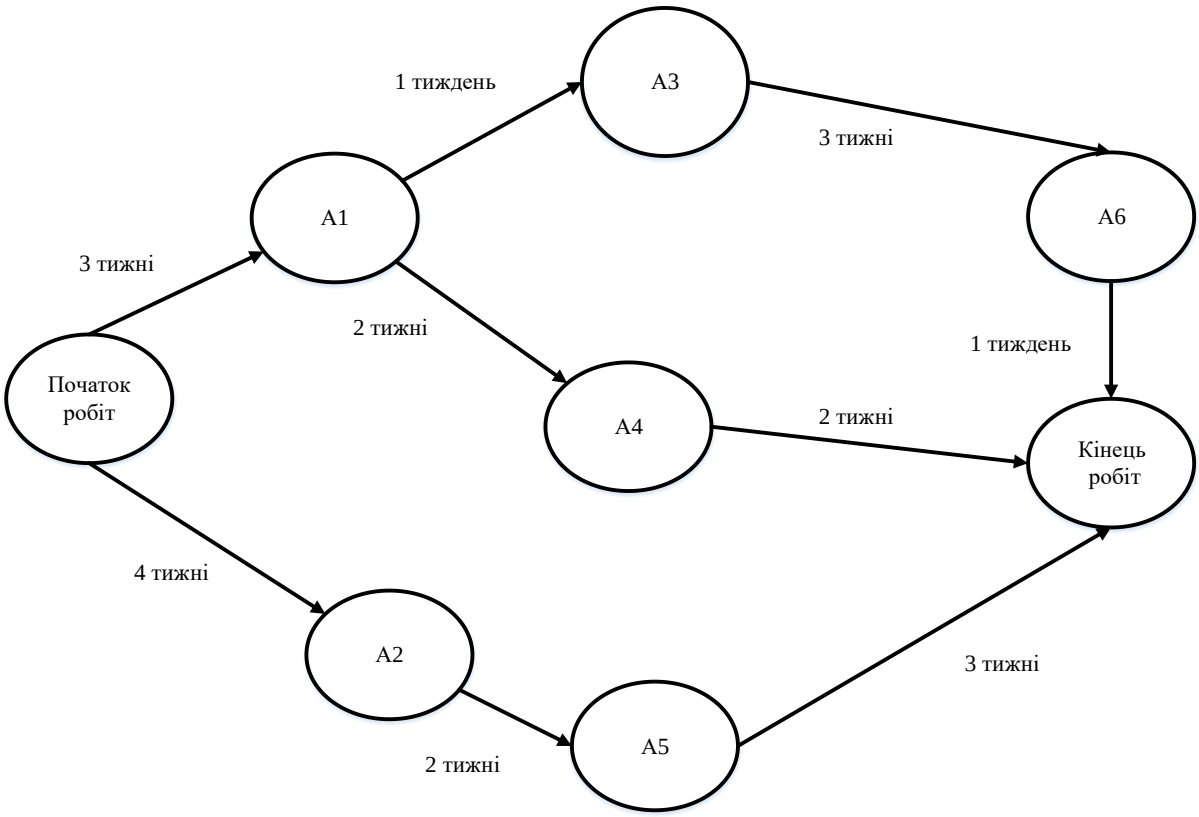

Рис. 1.5. Схема графу для визначення критичного шляху

Модель життєвого циклу комп’ютерних систем критичного призначення можна також представити у вигляді графу, де кожен вузол представлятиме стадію ЖЦ або діяльність на певному етапі проектування. Кожен вузол графу є критичним, а довжина критичного шляху представляє мінімальний загальний час виконання робіт щодо реалізації проекту комп’ютерної системи.

При оптимізації тривалості виконання проекту можна зменшити час виконання видів діяльності, які належать критичній послідовності робіт. Основні задачі проектних менеджерів полягають за моніторингом вчасності

реалізації робіт, що розміщені на критичному шляху, оскільки затримки на кожному з вузлів графу стимулюють затримку тривалості усього проекту комп'ютерної системи.

Основними компонентами при представленні критичного шляху у вигляді графу є вершини, що інтерпретують задачі розробки комп'ютерних систем, та ребра – містять тривалості виконання робіт і зв'язки між видами діяльності.

При розробці комп'ютерних систем критичного призначення з графу вибираються та виконуються роботи, які є незалежними від інших і відповідно не впливають на тривалість видів діяльності.

Іншим методом управління процесом розробки комп'ютерних систем критичного призначення є метод Program Evaluation and Review Technique (PERT). Уперше його застосували при розробці системи «Polaris» над якою працювали близько 4000 розробників, а сама система виконувала близько 60000 операцій.

У методі PERT, при використанні мережевих діаграм, представляються вершини, які інтерпретують події, а ребра інтерпретують події між двома сусідніми подіями, як показано на рис. 1.6.

На мережевій діаграмі (рис. 1.6), початкова точка є одночасно і кінцевою та знаходиться внизу діаграми. Згідно термінології методу PERT:

– початкова точка – сукупність набору подій або елементарна подія, яка виконана до старту деякого процесу у випадку задоволення визначеним критеріям або умовам;

– кінцева точка процесу – момент часу і відповідна подія, яка підлягає перевірці якості результатів виконання процесу;

– тривалість –час успішного виконання проміжної стадії процесу;

– термін – дата повного або часткового завершення процесу або події.

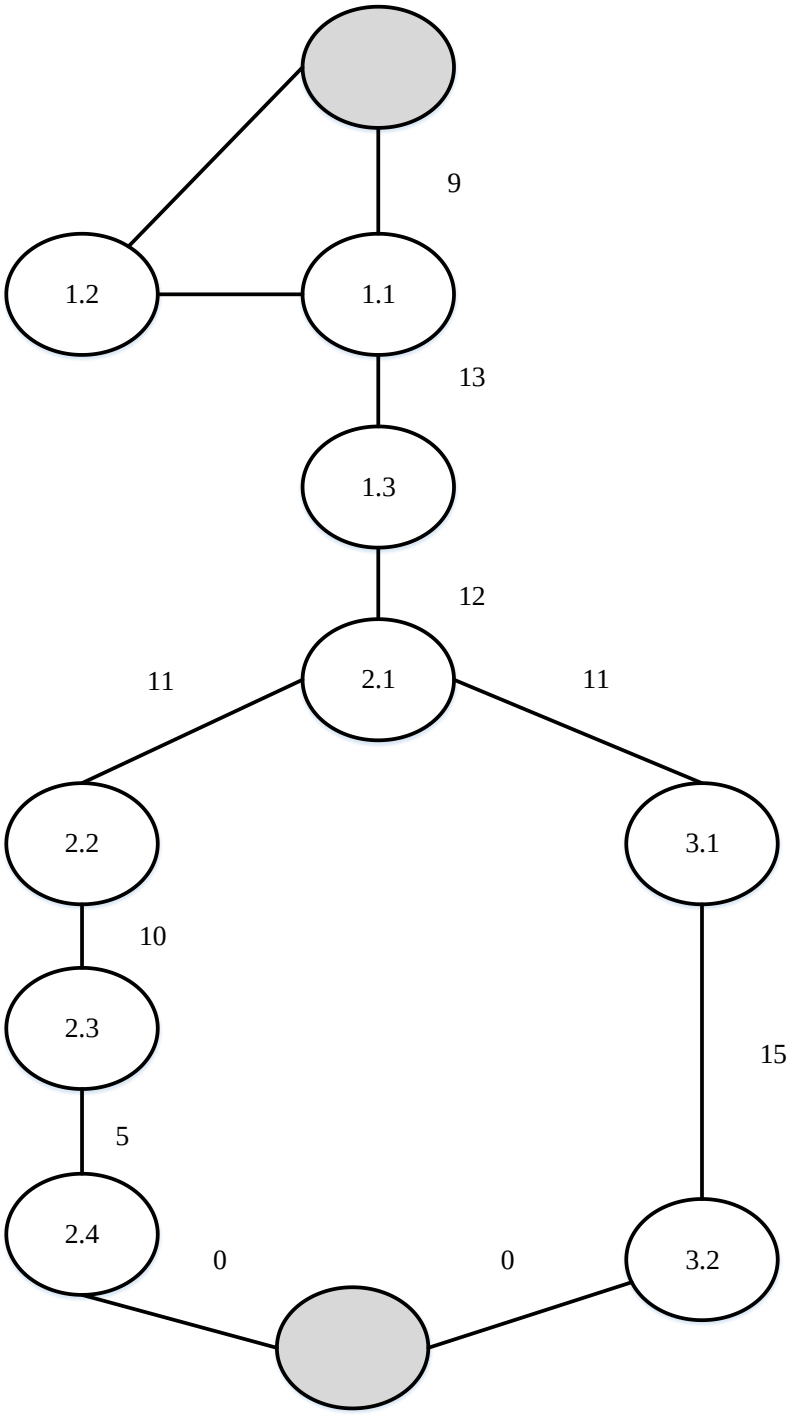

Рис. 1.6. Мережева діаграма та метод PERT

Метод PERT забезпечує гнучкість контролю та управління проектом і дає змогу проводити моніторинг вчасності та якості виконання окремих подій і комплексу видів діяльності.

Часові мітки на ребрах, які виходять з початкового вузла графа або входять у кінцевий вузол мають нульове значення. Для інших ребер передбачено вказання реального часу виконання події чи підпроцесу.

Мережеві діаграми можуть представляти ієрархічні структури, мережні структури, а також їх різновиди такі як цикли і вузли.

Відмінності між методами Program Evaluation and Review Technique та Critical Path Method є незначними. Однак перевагою методу PERT є те, що він враховує невизначеності, які можуть виникати протягом виконання певних операцій.

Однак, при моделюванні складніших типів зв'язків між вершинами графу, зростає складність використання методу Program Evaluation and Review Technique на практиці, що робить його менш ефективним в порівнянні з CPM. Час виконання операцій при використанні методу PERT можна розрахувати на основі трьох оцінок

$$T_{Ex} = \frac{1}{6}(T_{Opt} + T_P + 4T_{Pr}), \qquad (1.1)$$

– де T_{Ex} – час виконання операції;
– T_{Opt} – оптимістична оцінка часу виконання операції;
– T_P – песимістична оцінка часу виконання операції;
– T_{Pr} – імовірнісна оцінка виконання операції.

1.4. Аналіз процесу моніторингу проекту зі створення комп'ютерних систем критичного призначення

Процес моніторингу стадій реалізації комп'ютерних систем критичного призначення орієнтований на контроль і спостереження відповідності реалізації процесів до визначеного плану. У результаті моніторингу визначаються процеси, які необхідно відкоректувати шляхом оптимізації операцій та часу їх виконання. Окрім, цього можуть вноситись зміни у план розробки у разі неврахування складності процесів, недостатності інструментів чи методів розробки.

Моніторинг проводять для того, щоб фактичні результати було простежено за планами проекту з метою повернення проекту в планові рамки. При проведенні моніторингу аналізу підлягають наступні дані:

– фактичні і планові терміни виконання проекту;

– фактичні і планові фінансові та трудові ресурси;

– відсоток виконання проекту від запланованого;

– кількість сторінок документації;

– розмір програмного забезпечення відносно реалізованих функцій окремо взятого компоненту комп’ютерної системи;

– об’єм протестованих складових комп’ютерних систем;

– оцінка надійності, що розраховується як відношення провалених тестів до загальної кількості проведених тестів;

– оцінка інтенсивності дефектів, що виражається як відношення кількості знайдених і виправлених дефектів до загальної кількості помилок.

1.5. Аналіз методів оцінювання вартості проекту

Важливим етапом і процесом при розробці комп’ютерних систем критичного призначення є визначення вартості ресурсів. Факторами, що впливають та визначають кінцеву вартість комп’ютерної системи є ціна апаратних і програмних компонентів, кваліфікація розробників, типи використовуваних технологій та інструментальних засобів.

Окрім основних витрат на створення комп’ютерної системи, у загальну вартість можуть входити додаткові витрати, які виражаються у наданні послуг сторонніми розробниками, застосуванні додаткових засобів верифікації або інших CASE-засобів.

Оцінка витрат щодо створення комп’ютерної системи критичного призначення виражається у людино-днях і формується на початку проекту. Важливими факторами впливу на вартість проекту є типи і величина ресурсів, тривалість виконання робіт на стадіях життєвого циклу.

В якості ресурсів проекту комп’ютерних систем виступають команди розробників, апаратне забезпечення, приміщення, технології розробки, а також форс-мажорні обставини, які призводять до додаткових витрат.

При розрахунку вартості програмного забезпечення комп’ютерних систем та оплати праці програмістів можна використовувати такі апарметри як кількість стрічок програмного коду, або часу, затраченого на виконання поставленого завдання.

Найбільш ефективним при оцінюванні вартості проекту комп’ютерної системи є алгоритмічні методи. В основі алгоритмічних методів закладено використання моделей, які враховують вплив і зв’язок між витратами і факторами, які на них впливають. У такому випадку модель можна представити функцією залежності витрат від факторів впливу

$$(a + bS^{C}) \cdot m(X), \tag{1.2}$$

де S – розмір системи;

a,b,c – константи;

X – фактори, що впливають на вартість системи;

m – регуляризатор.

Іншим способом оцінювання вартості проекту комп’ютерної системи критичного призначення є експериментальна модель, що запропонована у [7]:

$$E = 5.25 \cdot S^{0.91}, \tag{1.3}$$

Дана модель визначена експериментальним шляхом при аналізі програмного забезпечення комп’ютерних систем з кількістю стрічок коду від 4000 до 467000. Кількість мов, які при цьому використовувались, становила 28 одиниць для 66 комп'ютерів. Тривалість розробки систем становила від одного до 10 років.

Окрім цього, можливе застосування техніки, яка виражається рівнянням затрат фірми-розробника:

$$E = 5{,}5 \cdot 0{,}73S^{1{,}16} \qquad (1.4)$$

При оцінюванні вартості комп’ютерних систем, зокрема, програмної складової, наведені моделі враховують і залежать від кількості стрічок коду.

На відміну від попередніх моделей, що враховують фактори вартості, рівень вмінь і знань розробників, характеристики системи і середовищ розробки та розгортання, модель COCOMO оперує трьома комплексними факторами, що впливають на вартість комп’ютерної системи критичного призначення.

Модель COCOMO на першому кроці оцінює ризики, пов’язані з людино-машинною взаємодією, інтеграційних модулях, які наявні у системі, ризики пов’язані з функціональними властивостями та ін. і враховує витрати при комплексному оцінюванні витрат на реалізацію комп’ютерної системи.

При оцінюванні другого комплексного фактору, у моделі Боема передбачено визначення витрат при проектуванні та реалізації функціональних блоків комп’ютерних систем у відповідності до специфікації вимог.

Третій крок моделі COCOMO передбачає оцінку витрат на етапі завершення процесу проектування та реалізації, коли відомі компоненти архітектури комп’ютерної системи критичного призначення, зокрема розмір програмного забезпечення, кількість елементів апаратного забезпечення, типи каналів передачі даних та ін.

Аналітично модель COCOMO можна представити формулою:

$$E = bS^{c} \cdot m(X), \qquad (1.5)$$

де bS^{c} – початкова оцінка;

$m(X)$ – коригуючий вектор вартості.

Модель COCOMO є динамічною моделлю, що враховує розвиток системи за кількістю компонентів на певний момент часу. Значення критерію c у формулі (1.5) на першому кроці моделі належить інтервалу [0,1], для інших – він належить діапазону [1,01…1,26].

1.6. Висновки до розділу

Основні результати, які отримані в даному розділі полягають в наступному:

1. Проведено аналіз характеристик і сфер застосування комп’ютерних систем критичного призначення і виявлено, що процеси їх розробки базуються на класичних моделях життєвого циклу розробки, що супроводжується інертністю та не гнучкістю зміни вимог на етапах життєвого циклу і призводить зростання часу випуску системи. Окрім цього, важливим аспектом комп’ютерних систем критичного призначення є високі вимоги до їх надійності та продуктивності, що вимагає застосування технологій забезпечення та контролю властивостей протягом усього життєвого циклу.

2. Проаналізовано процеси життєвого циклу комп’ютерних систем критичного призначення та виявлено, що стандарт не регламентує способів, методів і технологій їх розробки, що дає змогу оптимізувати процес розробки шляхом інтеграції гнучких методологій у класичні методи побудови таких систем.

3. Проведено аналіз процесів моніторингу за станом розробки комп’ютерних систем критичного призначення на етапах життєвого циклу та методів аналізу вартості проекту, що дало змогу визначити шляхи оптимізації процесів і вартості їх реалізації із застосуванням agile-методологій.

РОЗДІЛ 2
ПОБУДОВА МОДЕЛІ І МЕТОДУ ОПТИМІЗАЦІЇ ПРОЦЕСУ РОЗРОБКИ КОМП'ЮТЕРНИХ СИСТЕМ КРИТИЧНОГО ПРИЗНАЧЕННЯ

2.1. Аналіз моделей життєвого циклу комп'ютерних систем критичного призначення

2.1.1. Waterfall-модель життєвого циклу

Найбільш поширеною моделлю життєвого циклу, що використовується при розробці комп'ютерних систем критичного призначення є каскадна, або водоспадна модель. Характерною особливістю такої моделі є строга послідовність етапів реалізації комп'ютерної системи, що наведена на рис. 2.1.

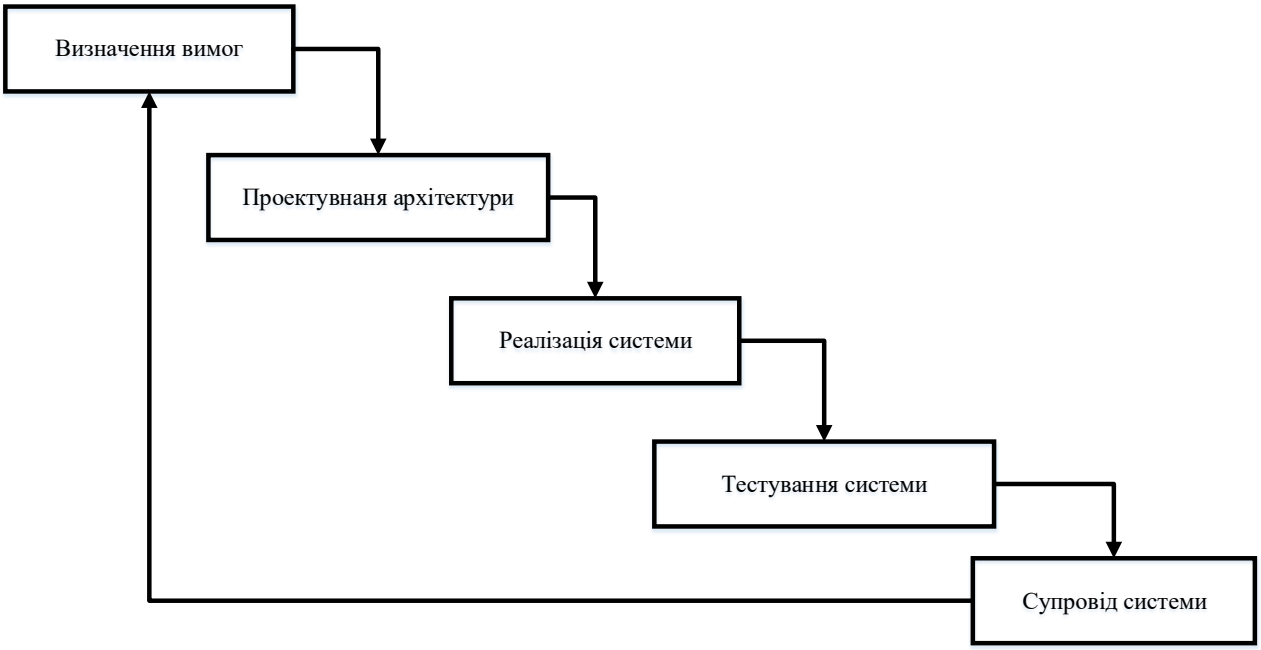

Рис. 2.1. Каскадна модель життєвого циклу

Дана модель передбачає, що процеси на стадіях життєвого циклу виконуються дуже якісно і немає необхідності повертатись до попередньої стадії. Початок наступного етапу розробки відбувається лише після повного

завершення робіт на попередній. Розробники здійснюють перевірку реалізованих властивостей систем відомими методами валідації і верифікації.

Особливістю водоспадної моделі ЖЦ є лінійна послідовність виконання процесів, що ведуть до реалізації комп'ютерної системи критичного призначення та інтегровані в основні процеси групи допоміжних та організаційних процесів. Окрім цього, дана модель використовує принцип «фабрики» – зміна властивостей комп'ютерної системи можлива лише після випуску готового продукту.

При проектуванні комп'ютерних систем критичного призначення застосування каскадної моделі характеризується наступними недоліками:

– на практиці, при розробці системи, можливі значні відхилення від послідовності та форми процесів, визначених на етапах життєвого циклу;

– дана модель не забезпечує можливості зміни вимог під час розробки комп'ютерної системи, а також погано враховує зміну середовища використання системи;

– значний час виправлення помилок чи дефектів системи, які внесені на ранніх етапах проектування, а також великі фінансові та часові ресурси, що використовуються на доопрацювання комп'ютерної системи.

Ризики, пов'язані з використанням водоспадної моделі життєвого циклу, полягають в наступному:

– не чітке формулювання вимог до комп'ютерних систем, що не в повній мірі враховує особливості еволюції системи та середовища її використання;

– велика монолітна архітектура комп'ютерної системи не дає змоги провести її декомпозиції, що може вплинути на неефективність використання апаратних ресурсів;

– не адаптованість моделі до швидкого внесення змін, що може знизити процес розробки компонентів комп'ютерних систем критичного призначення;

– неефективне та обмежене використання ресурсів при використанні водоспадної моделі життєвого циклу;

– реалізована комп’ютерна система критичного призначення може не відповідати очікуванням замовників або користувачів у зв’язку з нерозумінням вимог розробниками системи.

Однак, крім недоліків і ризиків використання каскадної моделі, вона володіє і певними перевагами:

– одночасне виконання основних процесів і пов’язаних з ним допоміжних та організаційних, що забезпечую стійкість процедур реалізації комп’ютерної системи критичного призначення;

– наявність повністю документованих результатів розробки системи забезпечує простоту супроводу, тестування та планового внесення змін.

Ефективність застосування водоспадної моделі життєвого циклу проявляється лише при створені першого прототипу комп’ютерної системи критичного призначення і дає змогу визначити та зрозуміти її основні функціональні властивості.

Однак виявлення помилок на етапах супроводу чи експлуатації комп’ютерної системи при застосуванні каскадної моделі передбачає повторення усіх процесів розробки, починаючи з визначення та аналізу вимог.

2.1.2. Модель життєвого циклу критичних комп’ютерних систем на основі інкрементів

Інкрементна модель передбачає розробку комп’ютерної системи на основі ітерацій, а результат виконання якої характеризується випуском проміжного прототипу. Тривалість ітерацій та їх кількість визначається на основі встановлення відповідності реалізованих властивостей комп’ютерної систем, яка враховує усі вимоги, які до неї висунуті, тобто реалізацією остаточної версії продукту.

Характерною особливістю інкрементних моделей є те, що на кожній ітерації послідовно виконуються процеси життєвого циклу, наведені у стандарті [5]. На рис. 2.2 наведено схему виконання процесів на ітераціях та самі ітерації .

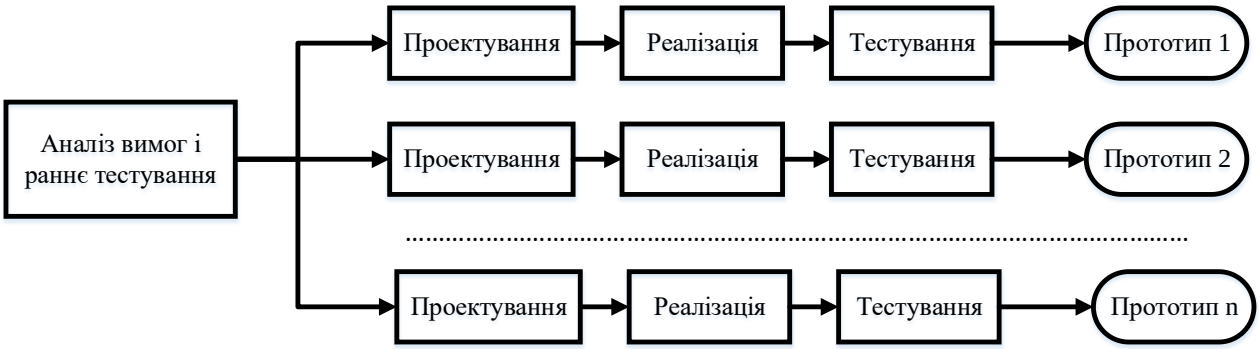

Рис. 2.2. Модель на основі інкрементів

На основі визначених вимог створюється перша версія комп'ютерної системи у якій реалізовані основні функціональні вимоги, і з часом на наступних ітераціях відбувається нарощування функціональності та інших вимог, які передбачені специфікацією. Такий процес триває до тих пір, поки не буде створена система, що задовольняє вимоги замовника чи користувачів.

Перелік процесів, які виконуються на кожній ітерації наведено на рис. 2.2.

Для моделі на основі інкрементів характерні наступні фактори ризику:

– всі вимоги до системи реалізовуються одночасно, що може призвести до виникнення конфліктних ситуацій;

– відкритим залишається питання стійкості архітектури комп'ютерної системи, пов'язана із швидкими змінами технологій розробки та вимог до системи;

– ризики, пов'язані з часовими рамками виконання проекту в умовах обмежених ресурсів.

Інкрементну модель, при розробці комп'ютерних систем, варто використовувати тоді, коли:

– існує необхідність реалізувати певні функціональні властивості за короткий проміжок часу у вигляді проміжного прототипу;

– існує можливість використання окремих компонентів комп'ютерної системи, як незалежних підсистем за рахунок гнучкої декомпозиції системи;

– бюджет на розробку комп'ютерної системи або її компонентів може зростати.

2.1.3. Особливості моделі життєвого циклу Боема при розробці комп’ютерних систем критичного призначення

Для забезпечення можливості зміни у процесах розробки комп’ютерних систем, а також у проміжні прототипи Б. Боемом запропоновано спіральну модель життєвого циклу, яку можна візуалізувати як показано на рис. 2.3.

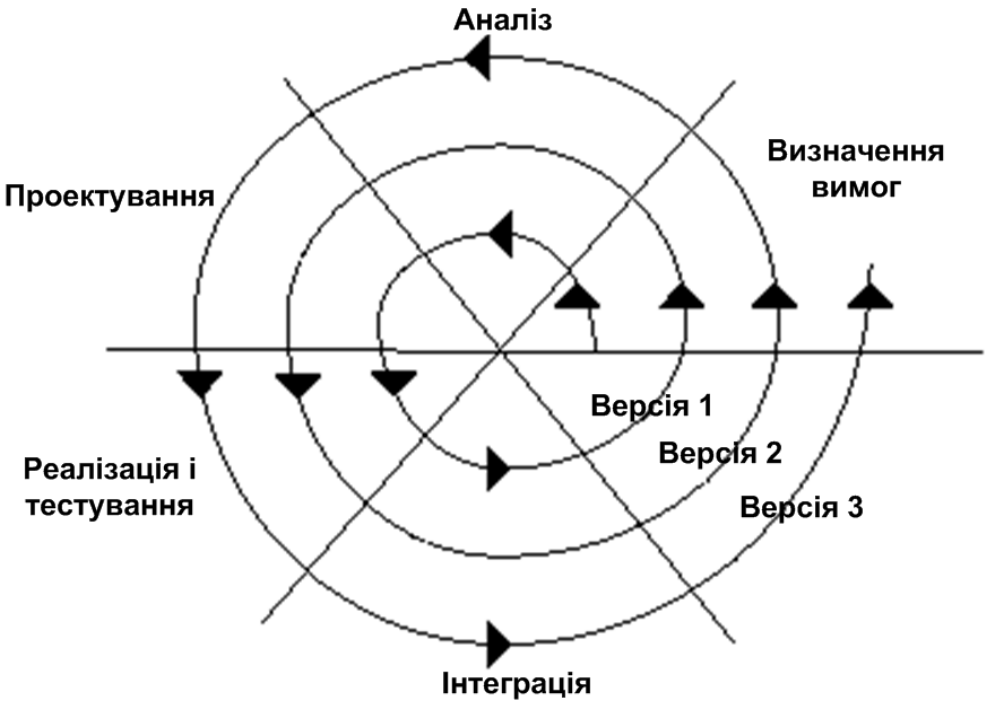

Рис. 2.3. Спіральна модель життєвого циклу

Спіральна модель забезпечує адаптивне та гнучке внесення змін, пов’язане з невідповідністю реалізованих у прототипі властивостей, артефактів чи опису в документації, виявлених замовником або користувачем комп’ютерної системи.

Витки спіральної моделі розгортаються від середини і включають аналіз властивостей системи, її верифікації і валідацію, оцінювання адекватності щодо переходу на наступний етап, або повернення до попереднього.

Модель Боема забезпечує можливість багаторазового переходу між витками спіралі, що дає змогу змінювати прототипи системи з будь-якого етапу розробки.

Для реалізації комп’ютерних систем критичного призначення спіральна модель не є ефективною за рядом причин.

Перша причина полягає в тому, що зміна вимог, зумовлена користувачами або замовниками, призводить до їх конфліктності та великої кількості повторень процесів на витках спіралі з внесенням змін у попередні, що може призвести до невідповідності реалізованих властивостей очікуваним сподіванням стейкхолдерів.

Друга причини пов’язана з еволюцією комп’ютерної системи і у випадку необхідності внесення змін або розширення функціональності системи, відбувається проходження по усіх витках спіралі з уточненням або доповненням зафіксованих вимог, що породжує складність управління вимогами та результатів реалізації прототипів.

2.1.4. «Еволюційна модель»

Еволюційна модель, при реалізації комп’ютерної системи, характеризується послідовністю розробки конструктивних блоків системи, що у порівнянні з інкрементною моделлю, не вимагає формування усіх вимог до системи. При використанні еволюційної моделі передбачається часткове визначення вимог, які з часом на проміжних стадіях уточнюють і зазнають змін.

Еволюційна модель, або по-іншому модель прототипування, передбачає можливість проведення аналізу предметної області для якої розробляється комп’ютерна система для визначення можливості її застосування.

З практики відомо, що модель прототипування є ефективною при реалізації простих не критичних систем, оскільки критерії надійності та продуктивності не є пріоритетними вимогами, а акцент робиться лише на забезпеченні необхідної функціональності комп’ютерної системи.

Еволюційна модель реалізується шляхом ітеративного виконання процесів, починаючи від управління вимогами аж до реалізації кінцевого прототипу комп’ютерної системи. Дана модель лежить в основі більш

розвинутої моделі еволюційного прототипування на етапах життєвого циклу КС, яка наведена на рис. 2.5.

Імплементацію у процес розробки комп'ютерних систем стимулювало розвиток методологій швидкої розробки комп'ютерних систем – Rapid Application Development (RAD).

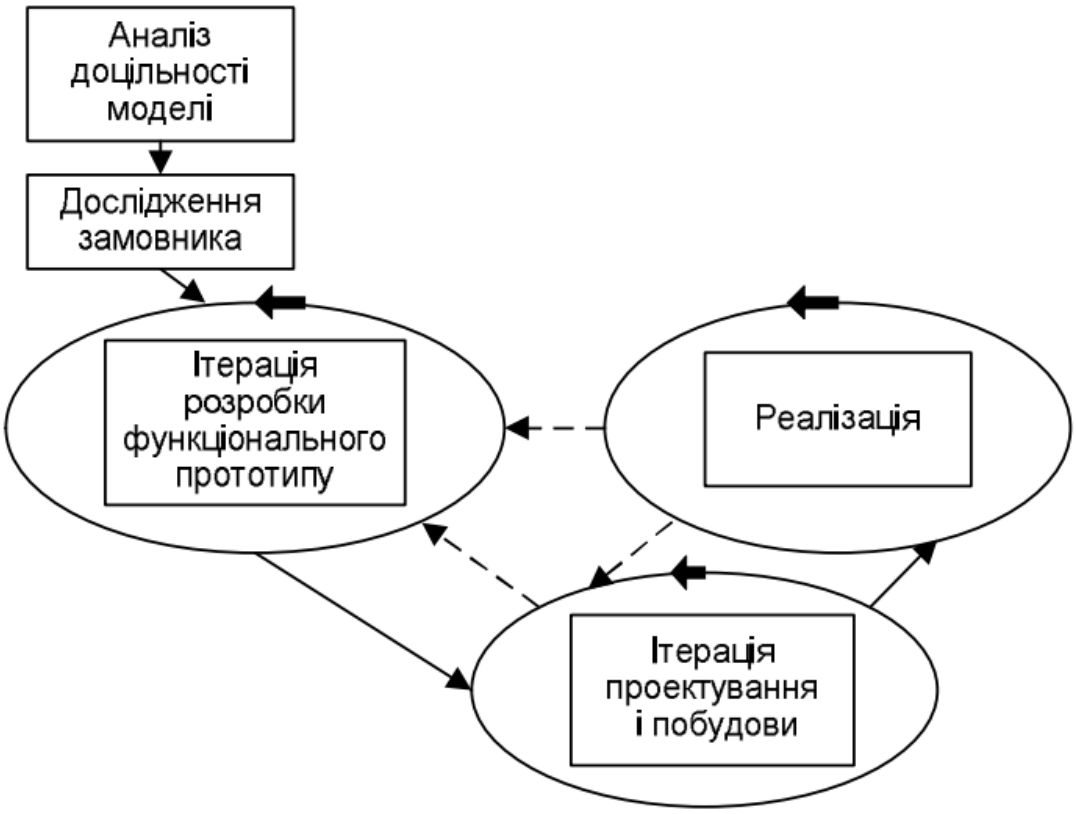

Рис. 2.4. «Еволюційне» прототипування

Основними процесами та етапами еволюційної моделі є проектування і реалізація, у результаті виконання яких встановлюється відповідність реалізованих властивостей функціональним вимогам і вимогам якості. Ці два процеси виконується ітераційно до поповненого задоволення вимог.

Як наслідок, одержують декілька прототипів комп'ютерної системи, кожен наступний відрізняється від попереднього кількістю реалізованих функцій.

Перевага еволюційної моделі при розробці комп'ютерних систем полягає в отриманні за короткий проміжок часу повнофункціонального продукту, у

якому реалізовано найбільш пріоритетні для замовника або користувача функціональні вимоги.

Еволюційна модель характеризуються наступними факторами ризику:

– складність архітектури системи, пов’язана з одночасною реалізацією великої кількості вимог;

– зайнятість команди розробників на протязі великого терміну часу.

До переваг використання еволюційної моделі належить:

– швидке прототипування основних функцій комп’ютерної систем і доставка проміжного продукту замовнику;

– еволюційність прототипів, що характеризується нарощуванням функціональності у кожній наступній версії системи;

– можливість використання прототипів у виглядів окремих підсистем комп’ютерної системи;

– гнучкий бюджет щодо реалізації комп’ютерної системи;

– наявність комунікації замовника та розробників щодо зміни або уточнення функцій системи;

– гнучкість зміни властивостей системи у відповідності до зміни функціональних вимог.

Еволюційна модель передбачає нарощування функціональності та якості шляхом додавання нових або зміни існуючих властивостей системи, в тому числі і реалізації вимог продуктивності, надійності, безпечності та ін.

2.1.5. Аналіз «V-подібної» моделі життєвого циклу розробки комп’ютерних систем критичного призначення

V-подібна модель представляє собою організацію сукупності процесів розробки системи з одночасним та ефективним застосуванням методів тестування на кожному етапі проектування комп’ютерної системи.

Як видно з рис. 2.5, ітерації у моделі виконується зліва направо і зверху вниз, що дає змогу на ранніх етапах проектування комп’ютерної системи одержати сукупність тестів для проведення системного та інтеграційного

тестування, а також ефективно використовувати час при створенні системи.

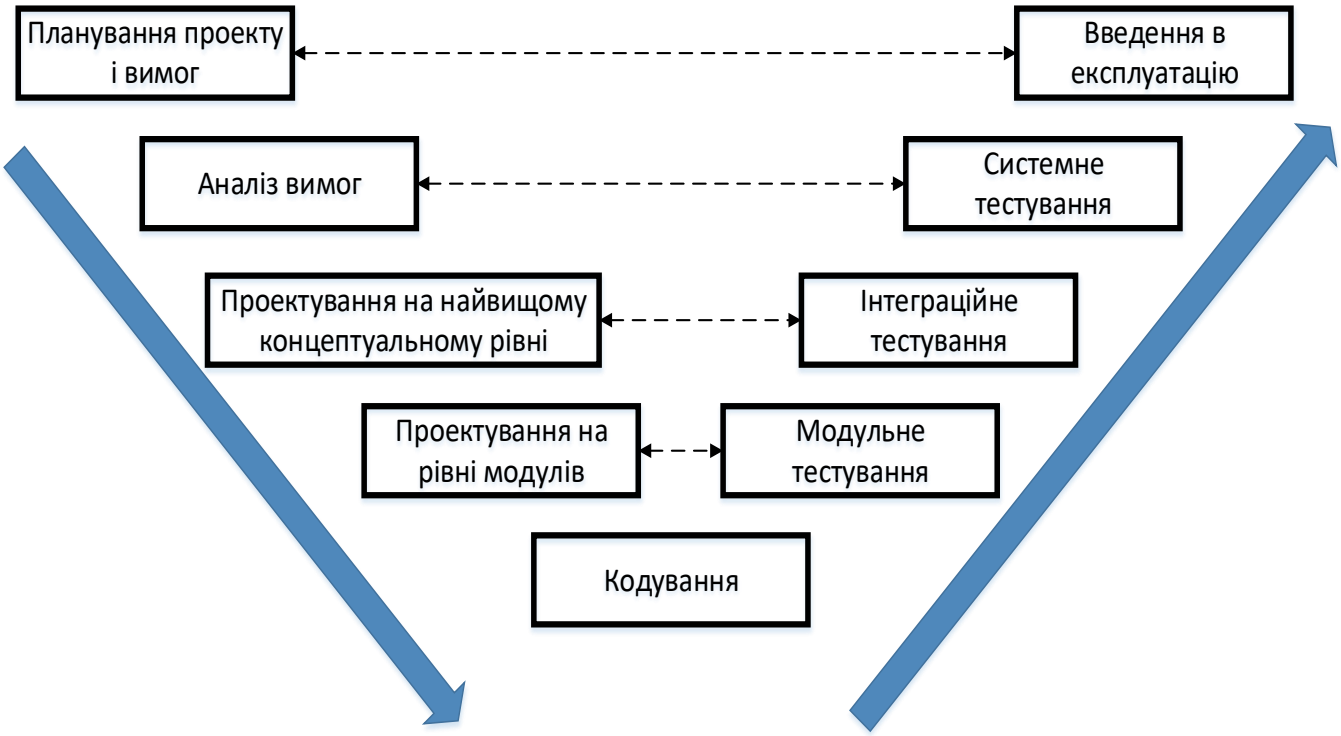

Рис. 2.5. «V-подібна» модель життєвого циклу

При застосуванні V-подібної моделі значно зростає якість процесів та відповідно системи в цілому, оскільки кожна ітерація передбачає перевірку відповідності результатів виконання із застосуванням різних методів валідації, верифікації і тестування. Це дає змогу забезпечити контроль відповідності реалізованих властивостей комп'ютерної системи щодо виконання вимог, визначених у специфікації.

Для V-подібної моделі характерними є недоліки, які притаманні водоспадній моделі, оскільки вона є її модифікованим екземпляром. Зокрема, це стосується управління та контролю за вимогами при їх зміні.

У випадку розробки складної комп'ютерної системи, що вимагає тривалого часу реалізації, та з використанням V-подібної моделі, кінцевий продукт може не відповідати бізнес-вимогам, які за цей час могли змінитись.

Змінюваність вимог, пов'язана також з швидкими темпами розвитку інформаційних технологій, оптимізацією та розробкою нових методів і засобів

реалізації комп’ютерних систем, впровадженням більш ефективних апаратних засобів і каналів передачі даних.

V-подібна модель слабо враховує вимоги якості такі як продуктивність, надійність, зрозумілість, безпечність, переносимість, оскільки на ранніх етапах проектування практично неможливо спрогнозувати ефективність методів, які будуть реалізованими у комп’ютерній системі.

Однак, V-подібна модель володіє значними перевагами, які полягають в наступному:

– можливість застосування методів валідації та верифікації комп’ютерної системи на ранніх етапах проектування;

– ефективність застосування методів unit-тестування щодо перевірки правильної реалізації детальних функціональних вимог;

– можливість створення архітектури комп’ютерної системи на найвищому концептуальному рівні з одночасним виконанням процесів інтеграційного тестування;

– узгодженість вимог та комунікація вимог між етапами виявлення та аналізу вимог та системного тестування.

До недоліків застосування V-подібної моделі належить:

– підтримка лише послідовного виконання подій;

– слабка гнучкість внесення змін у процеси на стадіях життєвого циклу;

– відсутність або недосконалість процедур щодо аналізу та управління ризиками виконання проекту;

– прототип системи одержується лише після виконання усіх етапів, передбачених V-подібною моделлю.

2.2. Обґрунтування методологій гнучкої розробки для оптимізації процесів реалізації комп’ютерних систем критичного призначення

2.2.1. Особливості гнучких методологій при розробці комп’ютерних систем критичного призначення

В основі гнучких методологій розробки комп’ютерних систем лежить принцип швидкої реакції на зміни у вимогах до системи та участь зацікавлених сторін безпосередньо при виконанні проекту. При застосуванні таких методологій передбачено тісну комунікацію як всередині команди розробників, так і між користувачами чи замовниками, що дає змогу із значно вищою ефективністю, в порівнянні з іншими технологіями, реалізовувати властивості комп’ютерних систем. У такому випадку, процес розробки комп’ютерної системи стає гнучким, оскільки базується на принципах ітеративного підходу з покроковою реалізацією визначених вимог до системи.

При застосуванні гнучких методологій розробки комп’ютерних систем використовуються кращі практики і методи їх проектування, а сам процес включає найбільш важливі аспекти, що ведуть до створення готового продукту.

На рис. 2.6 схематично представлено структуру процесів і зв’язків між ними у гнучких методологіях розробки.

Основна задача, яку вирішують такі методології розробки комп’ютерних систем, полягає в тому, що розробники можуть ефективно працювати в умовах частої зміни вимог і нестабільності предметної області завдяки здатності управління ними на усіх етапах розробки системи. Ітераційність виконання процесів та постійна інтеграція нових реалізованих властивостей за короткий проміжок часу сприяє забезпеченню якості кінцевого продукту з максимальним задоволенням потреб замовника.

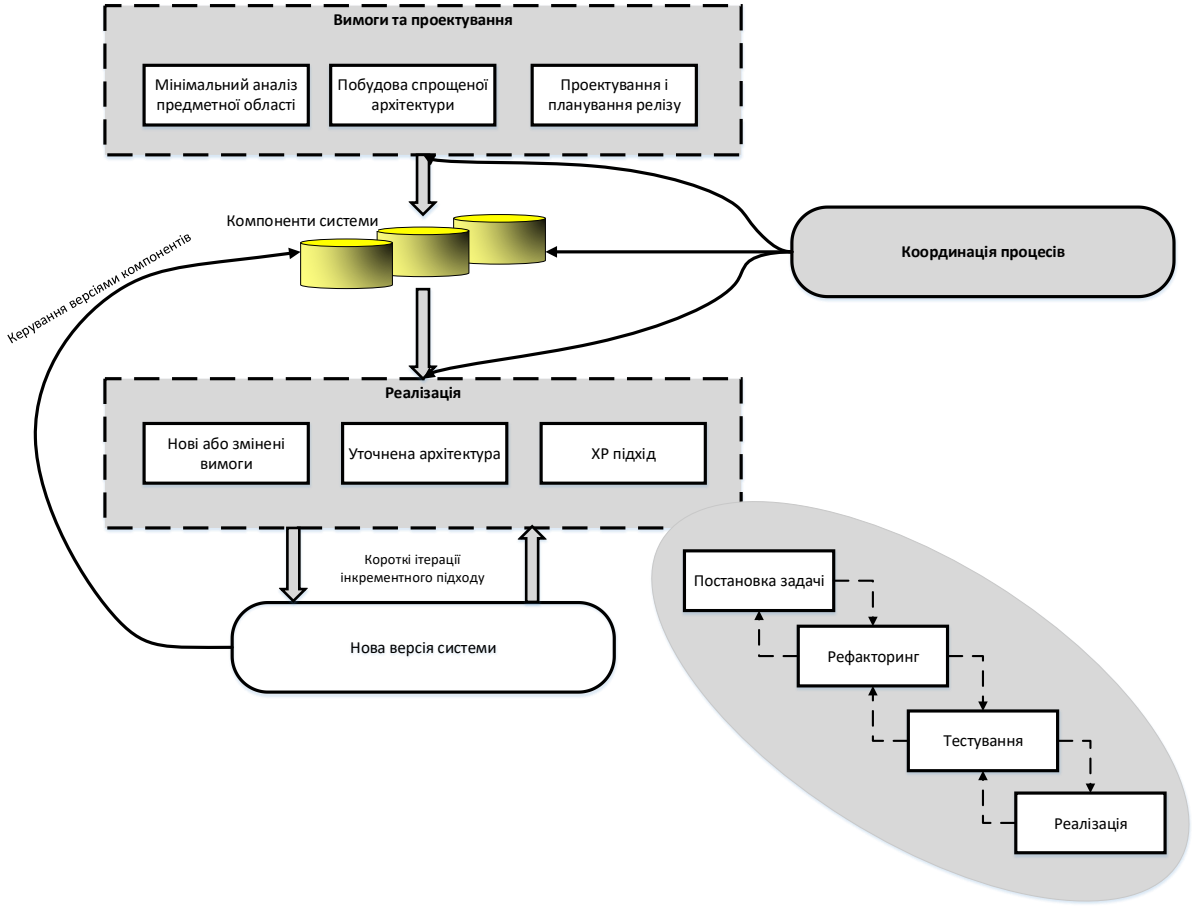

Рис. 2.6. Узагальнена модель процесу розробки в гнучких методологіях

До переваг гнучких методологій також належить:

– ефективне керування ризиками, шляхом постійної комунікації з користувачами або замовником системи;

– застосування методів і засобів continues integration та continues delivery, а також безперервного тестування, сприяє виявленню дефектів на ранніх стадіях реалізації комп’ютерної системи;

– саморганізація та самокерованість команди розробників, що проявляється в ефективності прийняття рішень при проектуванні комп’ютерних систем;

– вироблення стратегії реалізації найбільш важливих функціональних властивостей системи, що є спільним баченням сторони замовника і розробника;

– мінімізація максимально можливої тривалості виконання проекту комп’ютерної системи за рахунок створених прототипів підсистем, які готові до експлуатації.

У гнучких методологіях ітерації відрізняються складом і тривалістю виконання задач, однак у результаті виконання будь-якої з них зростає функціональність комп’ютерної системи.

За рахунок тісної співпраці розробників та замовників досягається належна якість виконання задач, що забезпечує еволюцію внесення змін у компоненти системи, а також значно зростає одночасність і точність трактування того, що повинно бути реалізовано у системі. Як наслідок, задоволеність кінцевого бенефеціара досягається внаслідок реалізації найбільш важливих функцій системи, які з часом також еволюціонують і переоцінюються на кожній ітерації.

Однак, велика кількість переваг гнучких методологій розробки комп’ютерних систем в порівнянні з іншими підходами, володіє і рядом недоліків.

Перша проблема застосування гнучких методологій полягає у необхідності зміни існуючої в компанії культури процесу розробки і складнощів, пов’язаних з долученням до процесів розробки користувачів і замовника. Це зумовлює необхідність встановлення командних правил гри, створення особливих правил спілкування із зацікавленими сторонами.

Також запровадження гнучких методологій розробки комп’ютерних систем може спричиняти спротив з боку розробників, оскільки вони повинні постійно вдосконалювати свої знання і вміння (skills).

Окрім цього, виникають проблеми із оптимальністю розподілу ресурсів проекту, що проявляється у дисбалансі між витратами і доходами.

До недоліків застосування гнучких методологій також належить нестабільність вимог, що проявляється у можливості невірного їх трактування на кожній окремо взятій ітерації.

Щодо архітектури комп’ютерної системи, то при застосування гнучких методологій, вона не завжди є стійкою, оскільки агрегується з окремих

компонентів, розроблених на кожній ітерації і доволі часто не узгоджується. Також складним є процес рефакторингу програмного забезпечення комп'ютерної системи.

Застосування гнучких методологій є не ефективним у випадку, коли кількість розробників комп'ютерної системи налічує понад сорок осіб, або комп'ютерна система представляє собою цілу екосистему. Окрім цього, фахівці з розробки комп'ютерних систем не рекомендують застосовувати цей підхід при реалізації систем з високим рівнем надійності та безпеки.

У результаті аналізу класичних моделей життєвого циклу і гнучких методологій розробки комп'ютерних систем виникає необхідність побудови моделі і розробки методу, який давав би змогу оптимізувати процеси розробки комп'ютерних систем, що відносяться до класу критичних. Для цього, пропонується інтегрувати критерії, які характеризують переваги гнучких методологій у моделі життєвого циклу з постійним моніторингом якості їх реалізації, що дозволить забезпечити необхідний рівень їх надійності та безпеки.

2.2.2. Методологія «SCRUM»

Однією з найбільш ефективних методологій Agile є методологія Scrum, яка дає змогу швидко реалізувати комп'ютерну систему з врахуванням можливих змін вимог. При реалізації проектів на основі Scrum, участь у розробці беруть невеликі команди, кількість членів яких не перевищує 5-9 осіб, а сам процес поділений на три стадії.

Перша стадія (Pregame) передбачає створення концептуального плану виконання робіт щодо реалізації комп'ютерної системи, у результаті якого формується сукупність вимог (ProductBacklog) і проектується архітектура системи на найвищому концептуальному рівні. Після завершення першої стадії відбувається перехід до наступної – стадія реалізації (Game).

Друга стадія передбачає декомпозицію задач у вигляді ітерацій, тривалість яких коливається в межах від одного до двох тижнів. Такі ітерації у

методології Scrum прийнято називати спринтами. Кожен спринт характеризується виконанням процесів, які нагадують послідовність виконання стадій у каскадній моделі життєвого циклу. Результат виконання ітерації завжди реалізує та відображає деяку функціональність комп'ютерної системи зі збереженням працездатності створеного прототипу.

Планування спринта відбувається на сесії Sprint Planning Meeting і містить інформацію про те, які функціональні вимоги будуть реалізовані протягом одного двох тижнів.

Методологією Scrum під час виконання спринта передбачено щоденні наради, так звані скрам-сесії (DailyScrumMeeting), що забезпечують комунікацію всередині команди розробників щодо поточного стану виконання проекту, обговорення проблем, які виникають на певному конкретному етапі реалізації. Останній день спринта, зазвичай, присвячений демонстрації одержаних результатів (SprintReviewMeeting), які отримані кожним членом команди щодо реалізації функціональності або інших вимог замовника.

Після завершення реалізації комп'ютерної системи проводиться комплекс заходів щодо її впровадження –Postgame. Взаємозв'язок між процесами методології Scrum наведено на рис. 2.7, структру спринта – а на рис. 2.8.

Рис. 2.7. Схема процесу Scrum

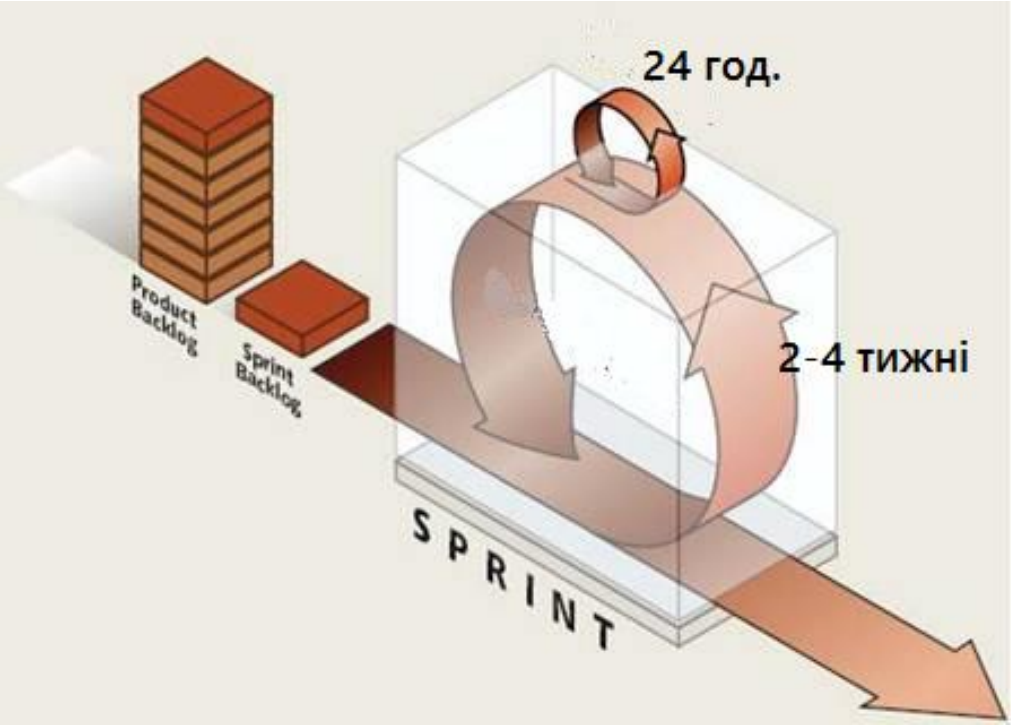

Рис. 2.8. Спринт у Scrum

Структурою процесів та організацією його виконання всередині ітерації займаються лише розробники комп’ютерної системи, а узгодження та синхронізація з управління проектом проводиться на зборах після завершення ітерації. Тому, як такого, розподілу ролей між учасниками проекту не існує.

2.2.3. Метод розробки динамічних систем

Не менш ефективним серед гнучких методів і методологій розробки комп’ютерних систем є метод, запропонований консорціумом англійських компаній, який має назву Dynamic System Development Method (DSDM). Основна перевага даного методу полягає у швидкій розробці системи із збереженням термінів виконання та бюджету, передбаченого на реалізацію комп’ютерної системи.

Методе розробки динамічних систем притаманні аспекти найбільш сучасних методів управління проектами, що дають змогу забезпечити простоту, масштабованість та ефективність виконання проектів.

Однак застосування методу DSDM не завжди можливе для керування розробкою особливих типів проектів, наприклад, критичних чи систем реального часу.

До недоліків використання методу розробки динамічних систем при реалізації комп’ютерних систем слід віднести:

– складність впровадження, що зумовлена високим порогом входу;

– довготривала у часі і фінансово дорога процедура зміни процесів розробки та управління проектом.

Зрілість процесів методу DSDM проявляється в ефективності реалізації процесів, зокрема при розробці програмного забезпечення комп’ютерних систем і є найбільш оптимальним засобом досягнення цілей серед гнучких методологій.

При використанні методу DSDM встановлюється можливість і доцільність його використання при реалізації конкретної комп’ютерної системи, а також визначається середовище та аналізується предметна область застосування системи. Результат цього етапу передбачає вибір або проектування майбутньої архітектури і плану реалізації комп’ютерної системи.

Одним із основних принципів, на яких базується даний метод гнучкої розробки, характеризується активним залученням користувачів до процесу реалізації комп’ютерної системи, що дозволяє скоротити витрати на уточнення і деталізацію вимог, виправлення дефектів і мінімізації ризиків. При цьому, передбачено участь у проекті невеликої кількості представників сторони замовника, які формують цільову аудиторію комп’ютерної системи і перебувають у постійній комунікації із розробниками.

Інший принцип, який сповідує DSDM, стосується надання можливостей розробникам самостійно приймати рішення щодо реалізації тієї чи іншою функціональності, визначення пріоритетності виконання задач та інших деталей, пов’язаних з технічними аспектами комп’ютерної системи.

Методом DSDM передбачено часті поставки функціональних прототипів комп’ютерної системи, що забезпечує раннє виявлення невідповідності реалізованих властивостей очікуванням замовника чи користувачів. Крім того,

документація, опис моделей та алгоритмів також узгоджується із зацікавленою стороною.

Орієнтація на реалізацію підтримки основних бізнес-процесів предметної області є пріоритетом при створенні комп'ютерної системи. Це забезпечує швидкість реакції на зміни у середовищі та подальшу оптимізацію або рефакторинг архітектури чи окремих компонентів системи.

Гнучкість ітерацій та інкрементний підхід дають змогу нарощувати функціональність комп'ютерної системи на кожному наступному кроці та забезпечувати ефективність керування проектом.

Метод DSDM передбачає, що будь-яка дія при виконанні ітерації може бути скасована і цим самим дає змогу невеликими кроками розробляти комп'ютерну систему без значної втрати реалізованої функціональності.

Складність керування вимогами в умовах їх змінюваності несе потенційно високі ризики щодо часу і якості виконання проекту, тому при використанні методу розробки динамічних систем передбачено узгодження та стійкість деяких вимог високого рівня.

Процес тестування реалізується на кожній ітерації, що визначена методом DSDM, і включає не тільки тестування, наприклад, програмного коду, але й супровідної і робочої документації.

Загальний процес розробки комп'ютерної системи за методом DSDM передбачає виконання трьох взаємозалежних процесів, які наведено на рис. 2.9.

Рис. 2.9. Розробка системи на основі методу DSDM

У даному випадку (рис. 2.9) цикл функціональної моделі передбачає розробку документації щодо аналізу предметної області і властивостей

подібних комп’ютерних систем прототипів. Розробляється і аналізується функціональний прототип системи на основі якого формується перелік функцій якими повинна володіти майбутня система.

Наступний цикл передбачає виконання процесів, що ведуть до створення робочого прототипу системи, як результату проектування і конструювання комп’ютерної системи. Слід зазначити, що у даному циклі забезпечено виконання процесів гарантування та оцінювання якості.

Завершальним етапом при використанні методу DSDM є цикл реалізації, що характеризується впровадженням комп’ютерної системи у середовищі замовника. При цьому відбувається формуванні інструкцій для користувачів, їх навчання, які можуть бути виконані за кілька ітерацій.

2.3. Формалізація процесу оптимізації розробки комп’ютерних систем критичного призначення

Пряме застосування Agile-методологій при проектуванні комп’ютерних систем критичного призначення неможливе, тому пропонується модифікувати і формалізувати процеси розробки таких систем.

Перший етап розробки комп’ютерних систем стосується визначення та аналізу потреб цільової аудиторії, тому перш за все необхідно математично обґрунтувати представлення вимог до комп’ютерної системи критичного призначення. Для цього пропонуємо С-вимоги записати у вигляді множини

$$Needs = \{ n_i, t_j \} \tag{2.1}$$

n_i – С-вимоги до комп’ютерної системи критичного призначення у визначений момент часу, $i = 1, N$ – потужність множини С-вимог у конкретний момент часу;

t_j – час виявлення С-вимоги, $j = 1, T$ – загальна кількість моментів часу.

Далі відбувається перетворення С-вимог у D-вимоги, які можна представити у наступному формалізованому вигляді

$$Req = \{ r_i, Needs_{ij} \} \tag{2.2}$$

r_i – D-вимоги до комп'ютерної системи критичного призначення;

$i = 1, R$ – кількість D-вимог;

$Needs_{ij}$ – С-вимоги до комп'ютерної системи, $j \in 1..N$ – кількість С-вимог, які мають зв'язок з D-вимогою.

Кожна D-вимога, що формує набір Req, повинна містити пріоритет щодо ваги при реалізації комп'ютерної системи критичного призначення, тоді формулу (2.2) трансформуємо у вигляд (2.3)

$$Req = \{ r_i, Needs_{ij}, Weight_i \} \tag{2.3}$$

$Weight_i$ – коефіцієнт пріоритету, що визначає важливість *i*-ої D-вимоги.

При встановленні значень коефіцієнтів пріоритету можна скористатись методами експертного аналізу. Найбільш широко використовуваними є методи, які базуються на середній зваженій оцінці, застосуванні методів Сааті, Когера і Ю та ін.

Наступний етап реалізації комп'ютерної системи критичного призначення полягає у проектуванні її архітектури. Для цього необхідно визначити структурні компоненти і побудувати зв'язки між ними. На найвищому рівні проектування, архітектуру можна подати у вигляді

$$ArcHigh = \{ comp_i \} \tag{2.4}$$

$comp_i$ – структурні елементи архітектури комп'ютерної системи на найвищому рівні проектування;

Кожен компонент архітектури комп’ютерної системи критичного призначення забезпечує виконання однієї або кількох D-вимог з множини Req, що дозволяє записати

$$\{Req_i\} \rightarrow comp_j \text{ або } comp_i \rightarrow Req_{ij} \qquad (2.5)$$

$\{Req_i\} \in Req, i = 1..H$ – сукупність D-вимог, які імплементовані у деякому компоненті комп’ютерної системи критичного призначення;

H – потужність підмножини D-вимог щодо конкретного компоненту архітектури на найвищому рівні проектування.

Кожен компонент комп’ютерної системи критичного призначення є більш, або менш важливим. Тому при обчисленні ваги компоненту архітектури комп’ютерної системи запропоновано скористатись усередненням зваженої суми пріоритетів D-вимог

$$Weight(comp_i) = \frac{1}{N}\sum_{j=1}^{H} Weight_j , \qquad (2.6)$$

де $Weight(comp_i)$ – пріоритет підсистеми архітектури комп’ютерної системи;

$Weight_j$ – пріоритет *j-ої* D-вимоги, що планується імплементувати в *i-ій* підсистемі.

Пріоритети вимог і компонентів архітектури є важливими при плануванні спринтів у гнучких методологіях розробки комп’ютерних систем, в тому числі і критичного призначення. Ці показники обов’язково будуть враховані при виконанні інших процесів.

Як зазначалось раніше, безпосередня реалізація компонентів архітектури відбувається шляхом виконання задач, визначених у конкретному спринті. Тому пропонується здійснити розбиття підсистем (2.4) на більш елементарні і додати їх у задачі при реалізації комп’ютерних систем критичного призначення.

У результаті одержимо множину задач, що визначає послідовність реалізації компонентів архітектури

$$Task = \{ comp_i \{ ecomp_{ik} \}, task_{ikj} \} \tag{2.7}$$

$ecomp_{ik}$ – найпростіший модуль *i-го* компонента комп'ютерної системи критичного призначення, $k = 1..K$, K – кількість найпростіших елементів;

$task_{ikj}$ – завдання, передбачені для виконання при реалізації *i-го* компоненту критичної комп'ютерної системи.

Для Agile-методологій характерним є поняття «спринта», що представляє собою деякий період часу і сукупність задач, у результаті якого одержують працездатний проміжний прототип компоненту комп'ютерної системи критичного призначення. Такий період часу пропонується визначити сукупністю, елементами якої є розробники КС, завдання, час початку і завершення етапу реалізації компонента системи

$$Sprint = \{ Task_i, \{ Person \}, t_{start}, t_{end} \} \tag{2.8}$$

$Task_i$ – набір завдань запланованих до виконання протягом «спринта»;

$\{ Person \}$ – сукупність розробників, призначених для виконання завдань на протязі «спринта»;

t_{start} – час старту «спринта»;

t_{end} – час завершення «спринта»;

Таким чином, розроблено метод оптимізації основних процесів при реалізації комп'ютерних систем критичного призначення за допомогою Agile-методологій і класичних моделей життєвого циклу.

2.4. Побудова моделі і методу оптимізації розробки комп’ютерних систем критичного призначення

Більшість моделей ЖЦ комп’ютерних систем, які апробовані часом та проектами, дають змогу виявляти відхилення від заданих вимог або помилки у комп’ютерній системі критичного призначення при завершенні проекту. Це погано впливає на тривалість розробки та відповідно підвищує фінансову складову і кількість ризиків щодо виправлення недоліків. На противагу класичним статичним моделям, динамічні моделі, в основі яких лежать ітераційні підходи, забезпечують зростання ступеню якості процесів розробки і комп’ютерної системи, як продукту, в цілому. До недоліків застосування таких моделей належить низький рівень імплементації методів забезпечення необхідного рівня якості і надійності при проектуванні систем критичного призначення.

Враховуючи недоліки проаналізованих моделей життєвого циклу, доцільним є розробка процедур інтеграції методів забезпечення їх надійності і якості шляхом впровадження засобів управління та моніторингу процесів розробки на різних стадіях життєвого циклу.

Причинами, які стимулюють необхідність забезпечення якості в процесі розробки комп’ютерних систем критичного призначення, є широкий спектр функціонально-подібних систем однакового призначення. Окрім цього, розробники систем основну увагу концентрують на функціональності продукту, і не застосовують методів і засобів гарантування і перевірки якості. Це призводить до того, що реалізовані у комп’ютерній системі властивості не відповідають очікуванням кінцевих користувачів.

Сьогодні при оптимізації процесів розробки комп’ютерних систем критичного призначення наукові дослідження і практики проектування спрямовані на вдосконалення технологій тестування та менеджменту проектів. Проте комплексного підходу, який би забезпечував управління і контроль критеріями якості на кожній стадії впродовж життєвого циклу поки що не запропоновано.

На етапі розробки вимог якість залежить від коректності формалізованого представлення потреб замовник, що можуть зображатися як у текстовому, так і графічному вигляді. У наукових працях [16, 17] такі алгоритми містять велику кількість операцій та не обгрунтовані математичними моделями і методами, а це негативно впливає на процеси реалізації комп'ютерних систем. У роботі [19] наведено підхід до представлення вимог комп'ютерних систем, що використовує термінологію моделей якості та забезпечує математичне обґрунтування процесів життєвого циклу. При використанні моделей якості стандарту ISO/IEC 25010 можна уникнути ситуацій щодо підміни або неоднозначності представлення вимог до комп'ютерних систем і досягти допустимого рівня адекватності і стандартизованості результатів представлення. Пропонується за основу взяти модель, що враховує аспекти якості комп'ютерних систем і застосувати її до усіх стадій життєвого циклу. При використанні даної моделі необхідно розробити критерії прийнятності щодо відповідності результатів проектування, конструювання та інших основних процесів розробки. Графічно модель можна зобразити, як показано на рис. 2.10.

Використання процедури перевірки результатів кожного етапу розробки комп'ютерних систем дозволяє забезпечити трасування вимог. Це дає змогу ефективно керувати реалізацією комп'ютерних систем критичного призначення з одночасним досягненням необхідного рівня якості кінцевого продукту.

На основі запропонованої процедури зручно вносити та проводити моніторинг змін у вимогах до комп'ютерних систем, здійснювати маніпулювання ними, а також формувати та аналізувати стани їх виконання на усіх етапах життєвого циклу. А це в свою чергу дає змогу оптимізувати та уніфікувати задачі проектування комп'ютерних систем критичного призначення на початкових етапах розробки.

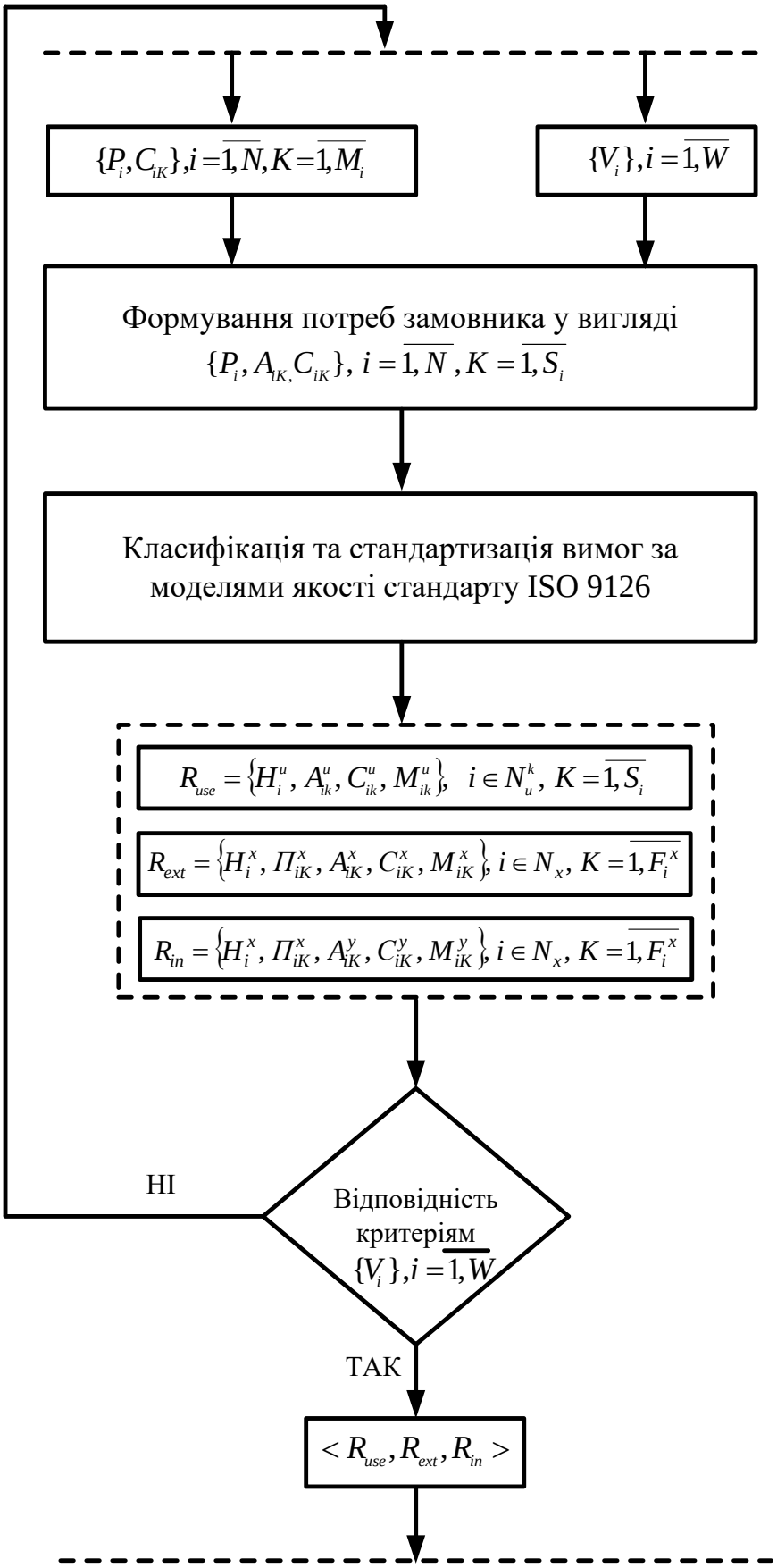

Рис. 2.10. Послідовність виконання процесів життєвого циклу з інтегрованими критеріями якості

На рис. 2.11 представлено у графічному вигляді модель оптимізації процесу розробки комп’ютерних систем критичного призначення із заданим рівнем якості кінцевого продукту.

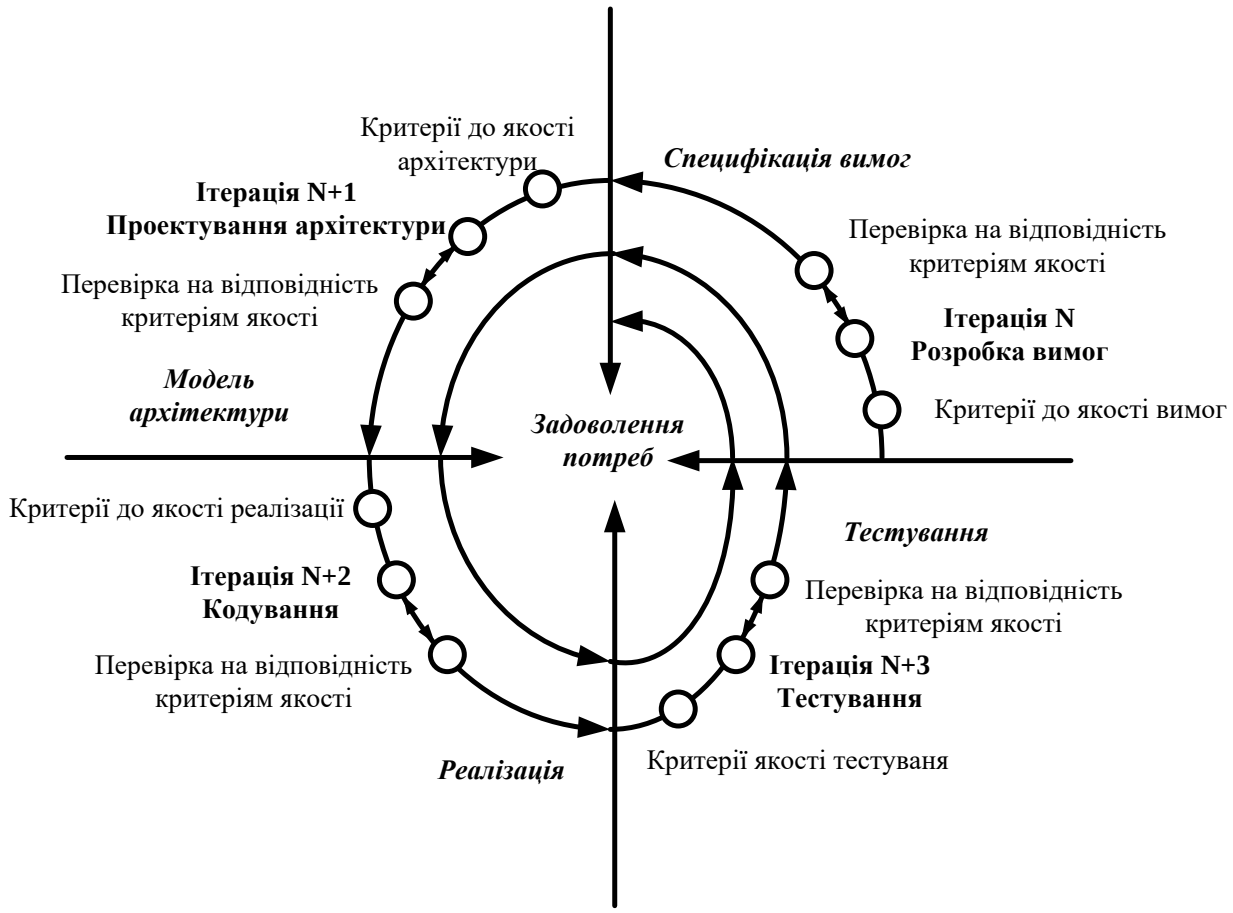

Рис. 2.11. Модель ЖЦ комп’ютерних систем критичного призначення

Оскільки, на рис. 2.10, наведено процедуру інтеграції критеріїв якості на етапі визначення та аналізу вимог до комп’ютерних систем критичного призначення, то його слід імплементувати для усіх основних процесів розробки, які наведено на рис. 2.11.

Оптимізація процесу розробки комп’ютерних систем полягає у тому, що при запропонованій моделі забезпечується паралельне асинхронне виконання задач на різних етапах розробки критичних систем і дає змогу забезпечувати управління та моніторинг стану виконання проектів на усіх стадіях життєвого циклу.

Окрім цього, при використанні гнучких методологій розробки комп'ютерних систем та запропонованого алгоритму і моделі життєвого циклу, можна ефективно проводити оцінювання якості результатів одержаних на кожній ітерації.

На рис. 2.12 наведено sequence діаграму, що визначає послідовність виконання процесів реалізації комп'ютерних систем критичного призначення при використанні моделей якості стандарту ISO/IEC 25010. У цьому випадку вимоги на кожній стадії відповідають структурованому стандартизованому набору атрибутів моделей якості.

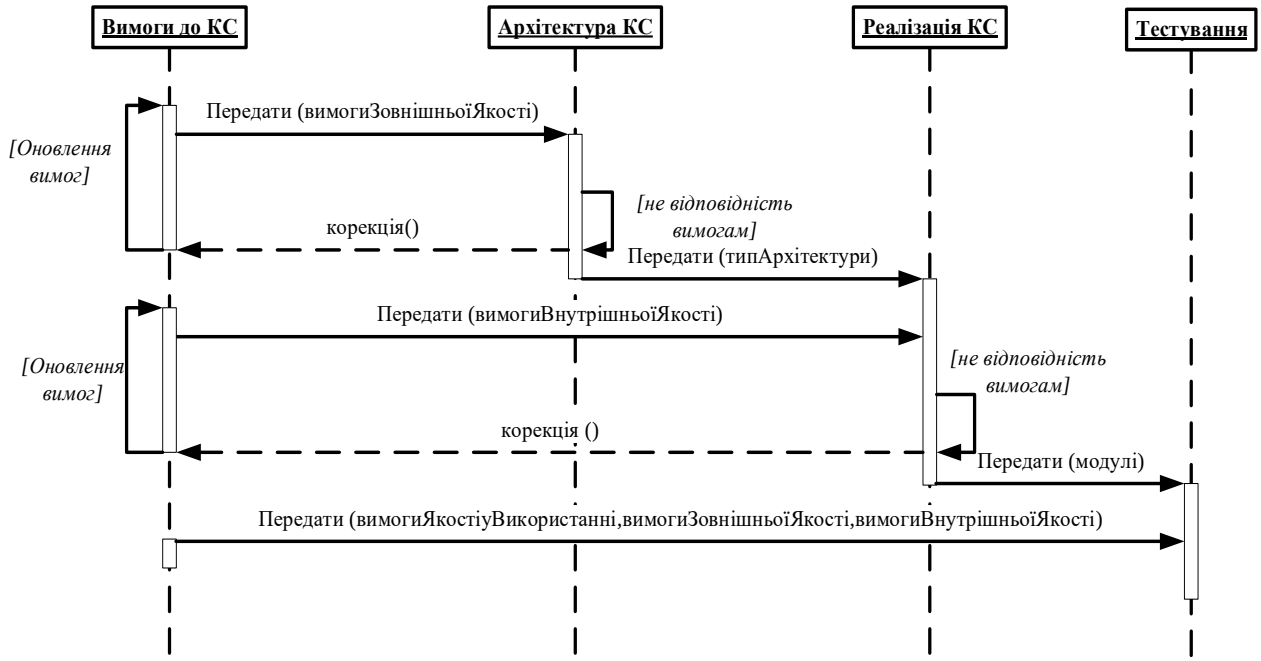

Рис. 2.12. Метод інтеграції критеріїв якості на стадіях ЖЦ

На основі запропонованого методу і моделей якості оптимізовано модель ЖЦ, яка на відміну від відомих, дозволяє підвищити якість КС, стандартизувати та уніфікувати процеси розробки на кожній із стадій ЖЦ, при цьому забезпечити виконання вимог стандарту ISO 12207 щодо контролю, забезпечення та керування якістю.

2.5. Висновки по розділу

1. Проведено аналіз класичних моделей життєвого циклу розробки комп’ютерних систем критичного призначення у результаті якого встановлено, що основними їх недоліками є велика часова тривалість і не здатність процесів до адаптації в умовах зміни вимог, а це в свою чергу вимагає розробки методів оптимізації для їх усунення.

2. Проведено аналіз особливостей застосування гнучких методологій розробки комп’ютерних систем, визначено основні їх переваги і недоліки, і як результат запропоновано можливі шляхи оптимізації щодо їхнього застосування при розробці комп’ютерних систем критичного призначення на основі впровадження параметрів якості результатів виконання кожної ітерації.

3. Формалізовано процедуру організації процесів і критеріїв якості, які можна використовувати при розробці комп’ютерних систем критичного призначення на основі класичних моделей життєвого циклу та гнучких методологій, що дало змогу оптимізувати часові та якісні характеристики виконання проектів.

4. Запропоновано механізм контролю (метод) за рівнем відповідності реалізованих у комп’ютерній системі властивостей, що дає змогу застосовувати метод DSDM і методологію Scrum при реалізації комп’ютерних систем критичного призначення.

РОЗДІЛ 3
ПРОГРАМНИЙ ЗАСІБ ПІДТРИМКИ МЕТОДУ ОПТИМІЗАЦІЇ РОЗРОБКИ КОМП'ЮТЕРНИХ СИСТЕМ КРИТИЧНОГО ПРИЗНАЧЕННЯ

3.1. Аналіз предметної області

Аналіз предметної області, процесів життєвого циклу, зокрема тих, які належать до групи основних процесів, дозволить спроектувати програмний засіб, орієнтований на забезпечення та оптимізацію виконання проекту комп'ютерних систем критичного призначення.

На практиці для аналізу предметної області можна застосовувати різні підходи: структурний, об'єктно-орієнтований, візуальний та ряд інших, які володіють певним набором засобів опису предметної області.

Оскільки, об'єктно-орієнтований підхід і всі похідні від нього, є найбільш ефективним, тому скористаємось технологією графічного об'єктно-орієнтованого моделювання UML (Unified Modelling Language).

Для опису та налізу предметної області у даній технології застосовуються декілька видів діаграм: діаграма прецедентів, діаграма класів, діаграма компонентів, діаграма видів діяльності.

Оскільки, предметну область нам необхідно описати з точки зору функціональних вимог, тому скористаємось таким видом діаграм, як діаграма прецедентів, або діаграма варіантів використання.

Для реалізації методу оптимізації процесів розробки комп'ютерних систем критичного призначення необхідно залучити три групи фахівців:

– системний аналітик – виконує роль аналізу предметної області для якої створюється проект шляхом визначення, аналізу і трансформації потреб у продукті на його властивості;

– експерт з якості – формує критерії якості вимог до комп'ютерних систем та реалізує процес управління ними;

– менеджер проекту – здійснює контроль та моніторинг процесів виконання проекту і показників якості системи.

На рис. 3.1 у вигляді діаграми прецедентів представлено основні функціональні можливості експерта з якості.

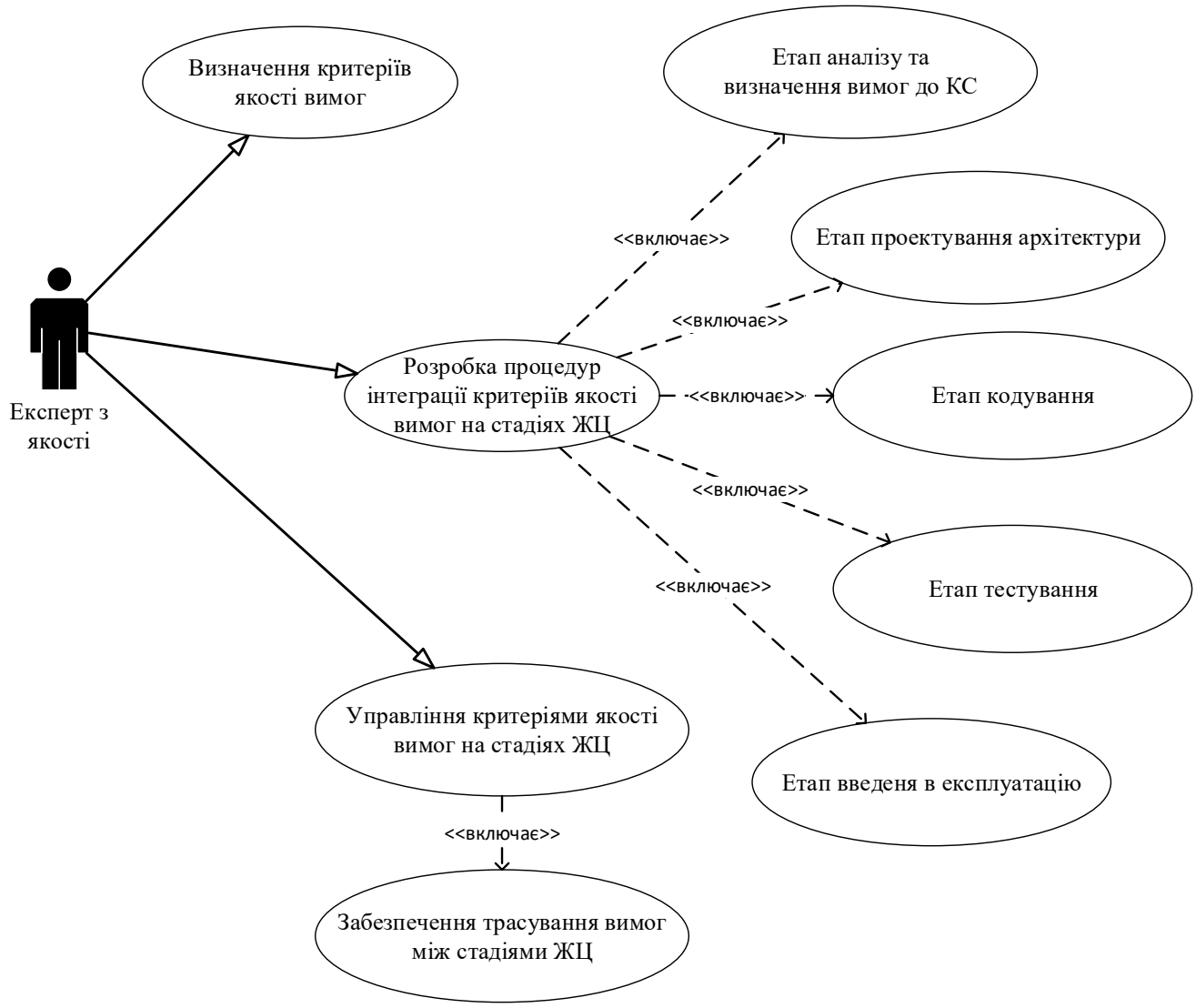

Рис. 3.1. Діаграма прецедентів «Експерт з якості»

Основні функціональні обов’язки експерта з якості полягають у розробці та формалізації критеріїв якості вимог до комп’ютерної системи критичного призначення та реалізації процедур їх інтеграції в основні процеси життєвого циклу.

Участь у проекті при використанні гнучких методологій беруть менеджери проекту, які, зазвичай, займаються організацією та контролем виконання проекту. На рис. 3.2 наведено додаткові функції, які необхідно виконувати менеджеру проекту, для оптимізації процесів у проекті.

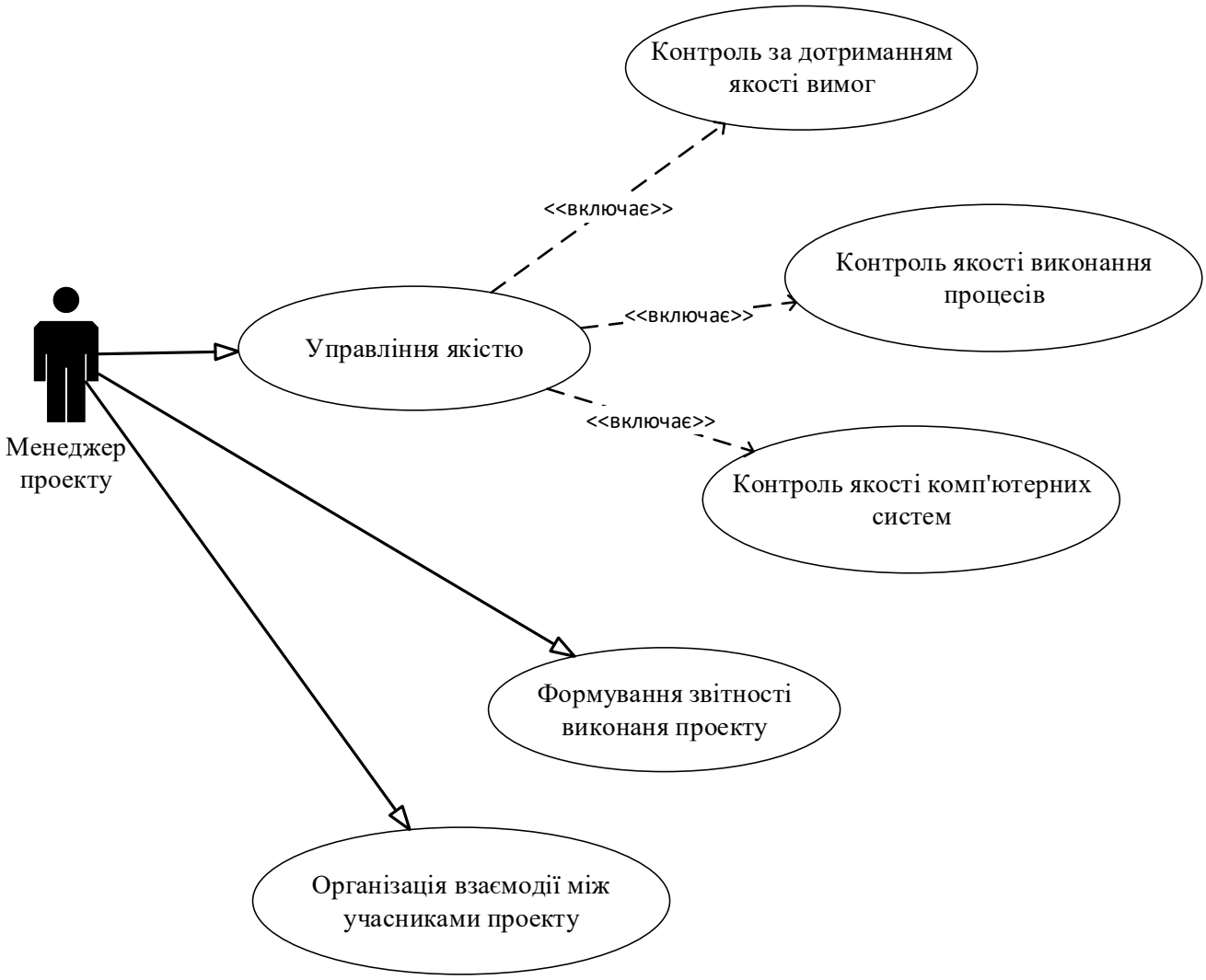

Рис. 3.2. Діаграма прецедентів «Менеджер проекту»

Основними функціональними вимогами щодо автоматизації діяльності менеджерів проектів з орієнтацією на підвищення якості їх виконання, є контроль та управління процесами розробки з врахуванням критеріїв якості вимог до процесів та комп’ютерної системи критичного призначення, які сформовані експертом з якості та системним аналітиком.

Системний аналітик, при впровадженні модифікованої моделі життєвого циклу оптимізації процесів розробки комп’ютерної системи критичного призначення повинен відобразити потреби у системі на вимоги до продукту. На рис. 3.3 наведено базові процеси, автоматизацію яких необхідно забезпечити для системного аналітика.

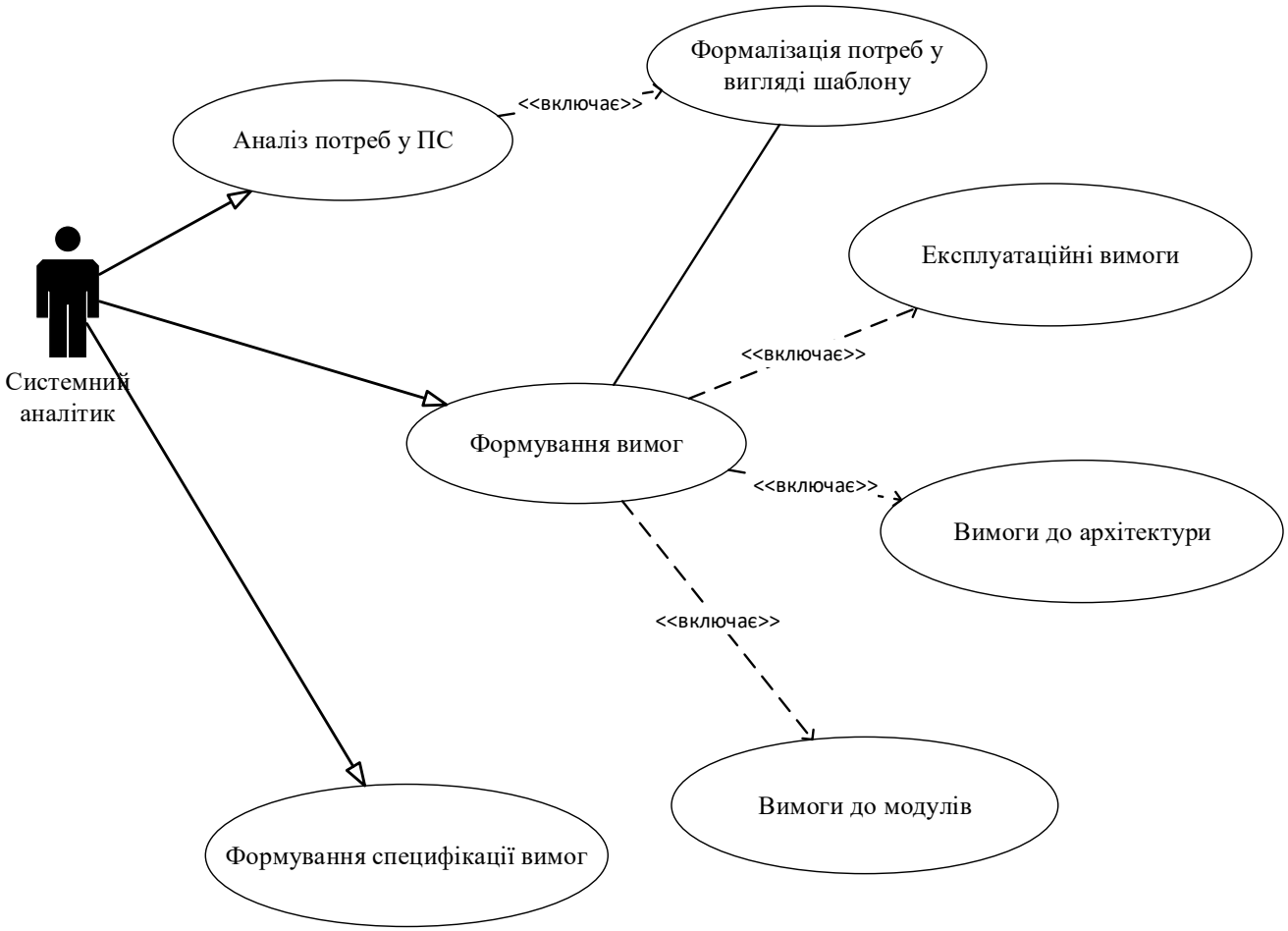

Рис. 3.3. Діаграма прецедентів «Системний аналітик»

До функціональних обов’язків системного аналітика входить формування вимог до комп’ютерної системи критичного призначення на рівнях вимог до експлуатації системи, вимог до архітектури та до окремих модулів системи. При цьому необхідно забезпечити трасування вимог з одного етапу життєвого циклу на інший.

Таким чином, розроблено діаграми прецедентів для учасників проекту, які забезпечують оптимізацію процесів у проекті, що дає змогу в подальшому спроектувати архітектуру засобу підтримки запропонованої моделі і методу.

3.2. Проектування архітектури засобу підтримки методу оптимізації процесу розробки комп’ютерних систем критичного призначення

Виходячи з висунутих у попередніх розділах вимог щодо оптимізації процесу розробки комп’ютерних систем критичного призначення та вимог до

архітектури було спроектовано структуру програмного засобу на концептуальному рівні, яка представлена на рис. 3.5.

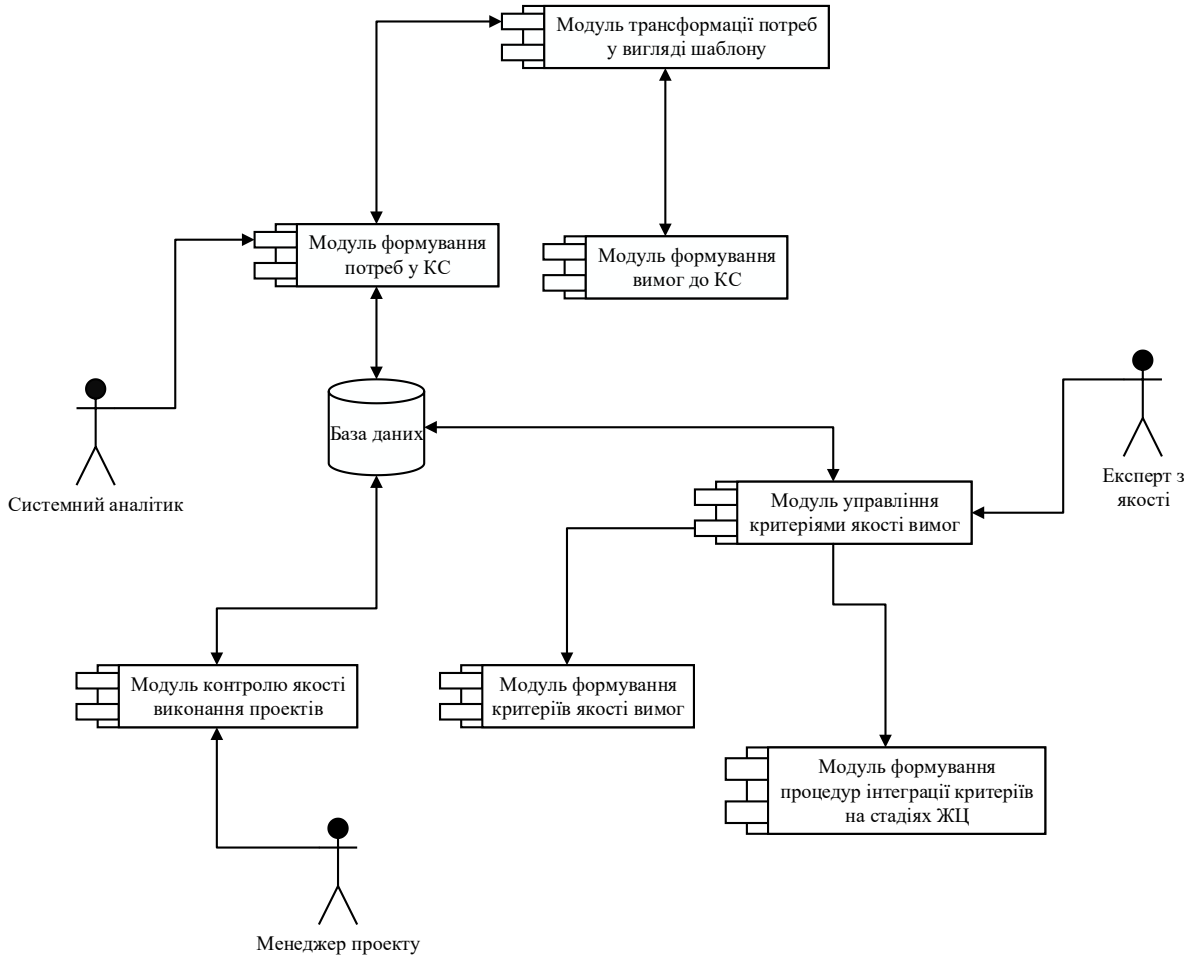

Рис. 3.5. Архітектура програмного засобу підтримки методу оптимізації процесу розробки комп’ютерних систем критичного призначення

Як видно з рис. 3.5 в архітектурі передбачено інтерфейси для трьох видів користувачів системи та відповідні компоненти, які реалізують процеси формування та управління як критеріями якості, так і формування самих вимог на різних етапах життєвого циклу. Це дозволяє забезпечити оптимізацію процесів розробки комп’ютерних систем критичного призначення. При проектуванні архітектури програмного засобу, що розробляється, кількість компонентів становить 7, що задовольняє загальні вимоги до архітектури програмного забезпечення. Вкладеність компонентів реалізується шляхом композиції відповідних класів.

У базі даних міститься інформація про критерії якості вимог, задоволення яких необхідно враховувати на стадіях життєвого циклу, посилання на потреби користувачів (С-вимоги) та вимоги до комп'ютерної системи критичного призначення.

3.3. Проектування схеми бази даних при оптимізації процесу розробки комп'ютерних систем критичного призначення

Для забезпечення здатності зберігання та маніпулювання даними про процес розробки комп'ютерних систем, спроектовано та реалізовано реляційну базу даних. Основні сутності та їх інтерпретація у термінах реляційної алгебри представлено на рис. 3.6 – 3.22. Як видно, з рис. 3.6 у даній таблиці існує три поля: ідентифікатор проекту, назва проекту та його опис.

ID_Project	int	☐
ProjectName	varchar(150)	☑
ProjectDescription	varchar(MAX)	☑

Рис. 3.6. Таблиця Project

Таблиця «Model» (рис. 3.7) містить інформацію про ідентифікатор моделей стандарту ISO/IEC 25010, який є первинним ключем даного відношення, зовнішній ключ на таблицю «Project», назву моделі якості та її короткий опис.

Column Name	Data Type	Allow Nulls
ID_Model	int	☐
ID_Project	int	☑
ModelName	varchar(120)	☑
Description	varchar(MAX)	☑

Рис. 3.7. Таблиця Model

На рис. 3.8 представлено таблицю «Characteristic», що містить ідентифікатор характеристики моделі якості, посилання на конкретний тип моделі (зовнішній ключ) і назву характеристики.

	Column Name	Data Type	Allow Nulls
🔑	ID_Characteristic	int	☐
	ID_Model	int	☑
	CharacteristicName	varchar(150)	☑

Рис. 3.8. Characteristic

Таблиця «SubCharacteristic», що представлена на рис. 3.9, має унікальний ідентифікатор ID_SubCharacteristic, зовнішній ключ на таблицю «Characteristic», назву підхарактеристики і її опис.

	Column Name	Data Type	Allow Nulls
🔑	ID_SubCharacteristic	int	☐
	ID_Characteristic	int	☑
	SubCharacteristicName	varchar(150)	☑
	Description	varchar(MAX)	☑

Рис. 3.9. SubCharacteristic

Оскільки підхарактеристики можуть одночасно належати кільком характеристикам, то утворюється зв’язок багато-до-багатьох і необхідно провести його розбиття на два простіших типи зв’язку – один-до-багатьох та багато-до-одного. На рис. 3.10 наведено таблицю, що забезпечує цілісність зв’язку характеристика-підхарактеристика, шляхом використання двох зовнішніх ключів – ID_Characteristic та ID_SubCharacteristic.

	Column Name	Data Type	Allow Nulls
🔑	ID_Rel_Char_SubChar	int	☐
	ID_Characteristic	int	☑
	ID_SubCharacteristic	int	☑

Рис. 3.10. Проміжна таблиця Rel_Char_SubChar

На рис. 3.11 наведено структуру таблиці Attribute, що містить ідентифікатор атрибуту, зовнішній ключ на таблицю Rel_Char_SubChar, назву атрибуту та його призначення.

	Column Name	Data Type	Allow Nulls
🔑	ID_Attribute	int	☐
	ID_Rel_Char_SubChar	int	☑
	AttributeName	varchar(150)	☑
	Description	varchar(MAX)	☑

Рис. 3.11. Attribute

Для вимірювання атрибуту, що описує конкретну вимогу комп'ютерної системи критичного призначення створено таблицю Metric, схема якої наведено на рис. 3.12.

	Column Name	Data Type	Allow Nulls
🔑	ID_Metric	int	☐
	MetricName	varchar(150)	☑
	Formula	varchar(100)	☑
	Description	varchar(MAX)	☑

Рис. 3.12. Metric

В якості первинного ключа використовується поле ID_Metric. Інші атрибути описують властивості назви метрики, формули для її обчислення та опис щодо застосування.

Оскільки, між атрибутом і метрикою може існувати зв'язок багато-до-багатьох, то доцільним є створення проміжної таблиці Attr_Metric, яка дозволяє уникнути цього не бажаного явища. На рис. 3.13 представлено вигляд цієї таблиці.

	Column Name	Data Type	Allow Nulls
🔑	ID_Attr_Metric	int	☐
	ID_Attribute	int	☑
	ID_Metric	int	☑

Рис. 3.13. Attr_Metric

Для опису потреб зацікавлених у розробці комп’ютерної системи осіб, визначено таблицю Need, яка наведена на рис. 3.14.

	Column Name	Data Type	Allow Nulls
🔑	ID_Need	int	☐
	NeedNumber	int	☑
	NeedDescription	varchar(MAX)	☑

Рис. 3.14. Need

Оскільки, потреби відображені на вимоги, тобто атрибути комп’ютерної системи, то необхідно забезпечити зв’язок між ними шляхом визначення таблиці Need_Attribute, яка наведена на рис. 3.15.

	Column Name	Data Type	Allow Nulls
🔑	ID_Need_Attribute	int	☐
	ID_Attribute	int	☑
	ID_Need	int	☑

Рис. 3.15. Need_Attribute

У даному випадку (рис. 3.15) існує два зовнішніх ключі: ID_Attribute та ID_Need.

Згідно запропонованої формалізації проектування комп’ютерних систем критичного призначенні із застосуванням гнучких методологій, необхідно визначити сутність, яка б описувала архітектурні компоненти системи. Для цього визначено структуру реляційного відношення Component, який наведений на рис. 3.16.

	Column Name	Data Type	Allow Nulls
🔑	ID_Component	int	☐
	ComponentName	int	☑
	ComponentType	varchar(120)	☑
	ComponentDescription	varchar(MAX)	☑

Рис. 3.16. Component

Таблиця, наведена на рис. 3.16., містить поле ідентифікатор компоненту, назву компоненту, його тип та примітку щодо використання.

Для створення зв’язку між програмним модулем (компонентом) та вимогами, які він реалізовує, необхідно створити таблицю, яка б давала змогу його зобразити (рис. 3.17).

	Column Name	Data Type	Allow Nulls
🔑	ID_Component_Requirenment	int	☐
	ID_Component	int	☑
	ID_Attribute	int	☑

Рис. 3.17. Component_Requirement

У даному випадку, таблиця Component_Requirement містить зовнішні ключі на таблиці Component та Attribute.

Визначивши сутності предметної області такі, як потреби, вимоги і компоненти, для забезпечення відповідності процесів гнучкої методології проектування комп’ютерних систем критичного призначення, потрібно описати такий артефкт як Завдання (Task). На рис. 3.18 показано структуру реляційного відношення для опису цієї сутності.

	Column Name	Data Type	Allow Nulls
🔑	ID_Task	int	☐
	ID_Sprint	int	☑
	TaskName	varchar(150)	☑
	TaskDescription	varchar(MAX)	☑
	ID_Developer	int	☑

Рис. 3.18. Task

На рис. 3.18 первинним ключем виступає поле ID_Task, зовнішнім – ID_Sprint, що забезпечує зв’язок із сутністю Sprint та ID_Developer – для зв’язку між задачею і її виконавцем. Інші поля описують задачу і її назву.

Для того, щоб побудувати зв’язок між компонентами на певному конкретному спринті та задачами, необхідно визначити таблицю – Task_component, структуру якої наведено на рис. 3.19.

	Column Name	Data Type	Allow Nulls
🔑	ID_Task_Component	int	☐
	ID_Component_Requirenment	int	☑
	ID_Task	int	☑

Рис. 3.19. Task_component

Зв’язок між таблицями Task і Component утворено шляхом сторення двх зовнішніх ключів – ID_Task та ID_Component_Requirement.

Ще одна сутність, яку потрібно описати у предметній області, визначає розробника комп’ютерної системи критичного призначення. Ця сутність наведена на рис. 3.20.

	Column Name	Data Type	Allow Nulls
🔑	ID_Developer	int	☐
	DeveloperName	varchar(MAX)	☑
	ID_DeveloperPosition	int	☑
	ID_Project	int	☑

Рис. 3.20. Developer

З рис. 3.20 видно, що ідентифікатором реляційного відношення є поле ID_Developer, два інші ідентифікатори виступають у ролі зовнішніх ключів – ID_DeveloperPosition та ID_Project.

Для того, щоб ефективно керувати проектом розробки комп'ютерної системи необхідно володіти інформацію про навики розробників, зокрема, про його посаду. Для цього визначено таблицю Position (рис. 3.21).

	Column Name	Data Type	Allow Nulls
🔑	ID_Position	int	☐
	PositionName	varchar(100)	☑
	Description	varchar(MAX)	☑

Рис. 3.21. Position

Для представлення спринта, тобто часових рамок виконання визначених задач певним розробником, створено таблицю Sprint, структура якої показана на рис. 3.22.

	Column Name	Data Type	Allow Nulls
🔑	ID_Sprint	int	☐
	ID_Project	int	☑
	SprintName	varchar(150)	☑
	DateOfStart	date	☑
	DateOfFinish	date	☑

Рис. 3.22. Sprint

Таблиця Sprint (рис. 3.22) містить первинний ключ ID_Sprint, зовнішній ключ ID_Project, поле назва спринта, дата його початку і завершення.

У результаті аналізу предметної області та з врахуванням особливостей запропонованої моделі і методу, спроектовано базу даних у середовищі MS SQL Server та автоматично згенеровано ER-діаграму бази даних, яка представлена на рис. 3.23.

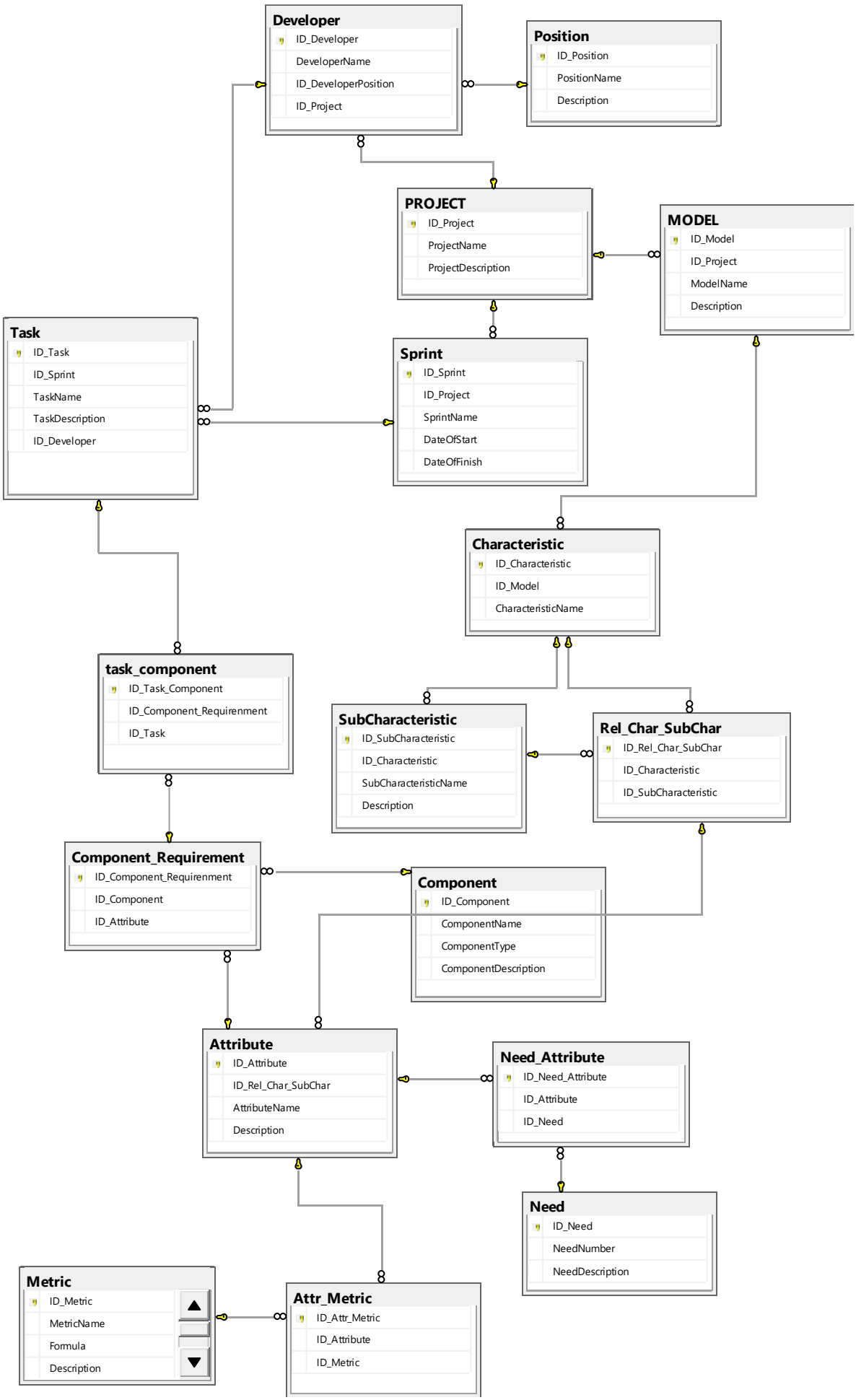

Рис. 3.23. Схема спроектованої бази даних

Спроектована та реалізована засобами мови SQL база даних забезпечує підтримку оптимізованого використання гнучких методологій розробки комп’ютерних систем критичного призначення і дає змогу ефективно оперувати значеннями показників якості та проводити моніторинг стану виконання задач.

3.4. Розробка інтерфейсів програмного засобу підтримки методу оптимізації процесів розробки комп’ютерних систем критичного призначення

Інтерфейс програмної системи підтримки методу оптимізації процесів розробки комп’ютерних систем критичного призначення повинен бути максимально простим та зручним у використанні. Тому, при розробці враховувались вимоги ергономіки при проектуванні користувацьких інтерфейсів. Головне вікно програмної системи представлено на рис. 3.24.

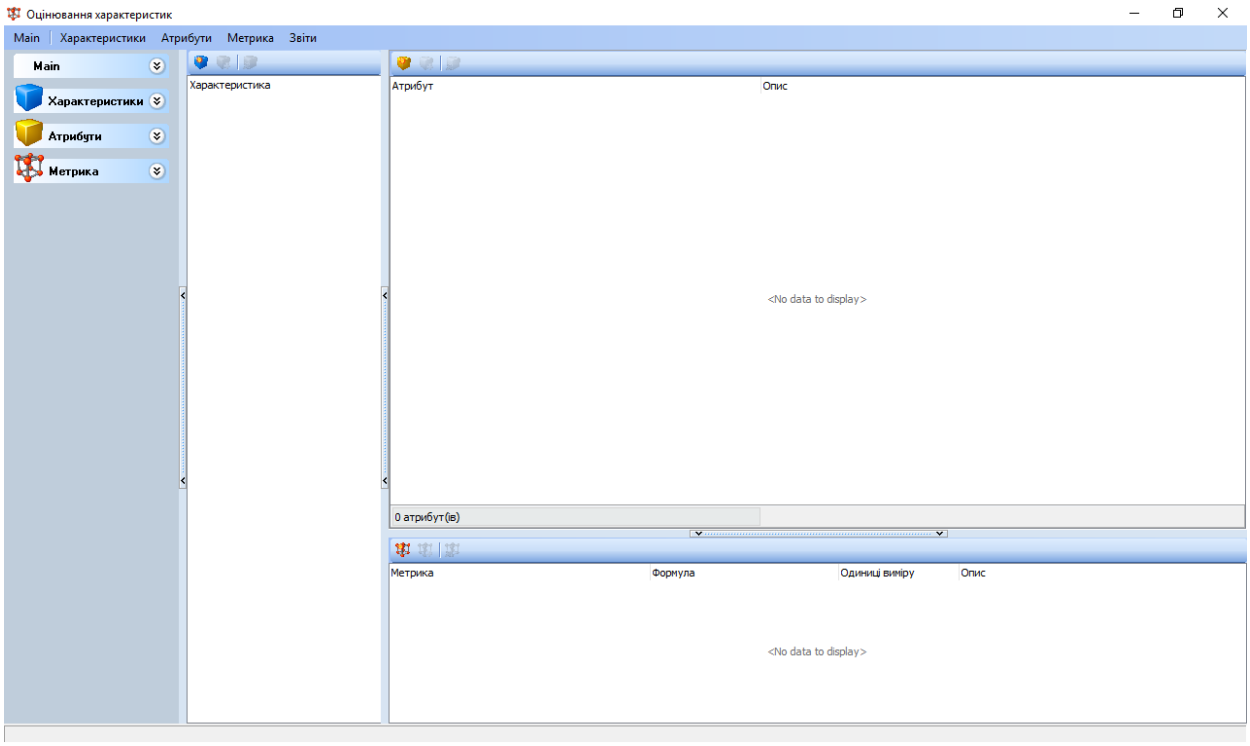

Рис. 3.24. Головне вікно програмного засобу

У вікні, наведеному на рис. 3.24, передбачено використання як основного меню, так і панелі, розміщеної зліва, для формування як критеріїв якості вимог, так і самих вимог до програмних продуктів. На рис. 3.25 показано структуру меню «Main».

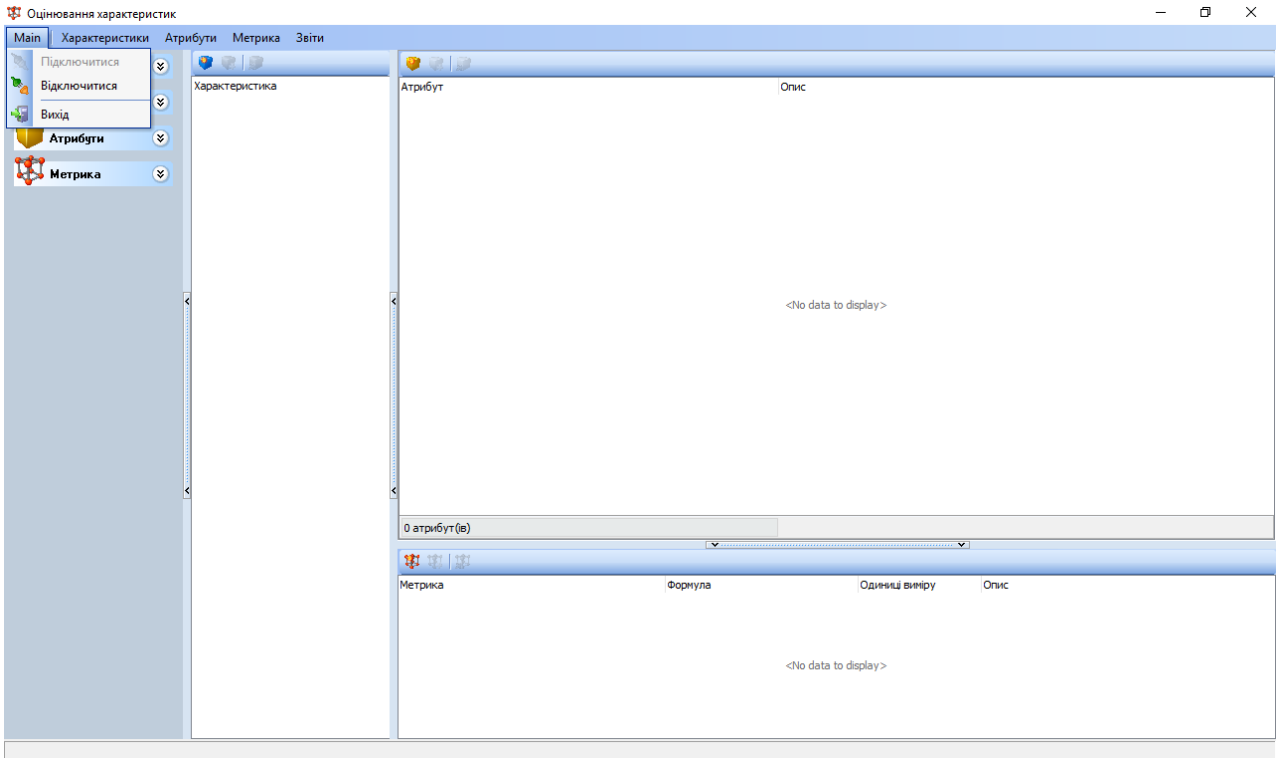

Рис. 3.25. Структура меню «Main»

Основними пунктами меню «Main» є функції з підключення та відключення бази даних, де зберігаються критерії якості іта вимоги до програмного продукту та вихід з програми.

На рис.3.26 наведено структуру меню «Характеристики». Пункти цього меню дають змогу керувати характеристиками комп’ютерних систем критичного призначення. Форма додавання характеристик наведена на рис. 3.27.

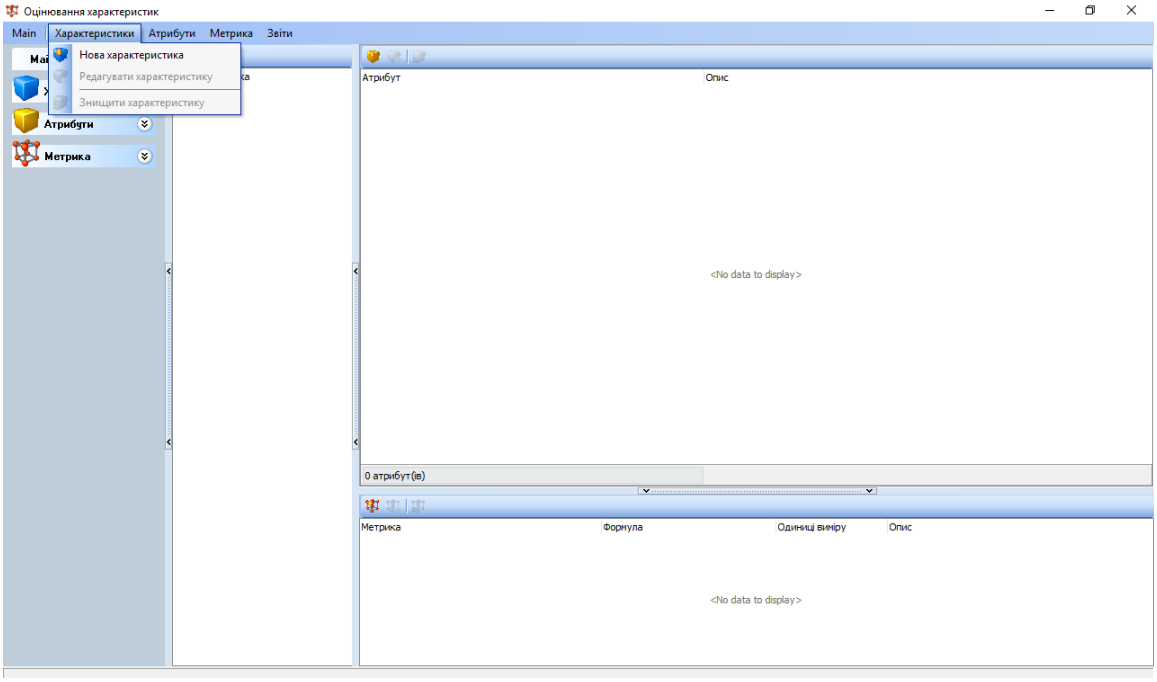

Рис. 3.26. Меню «Характерстики»

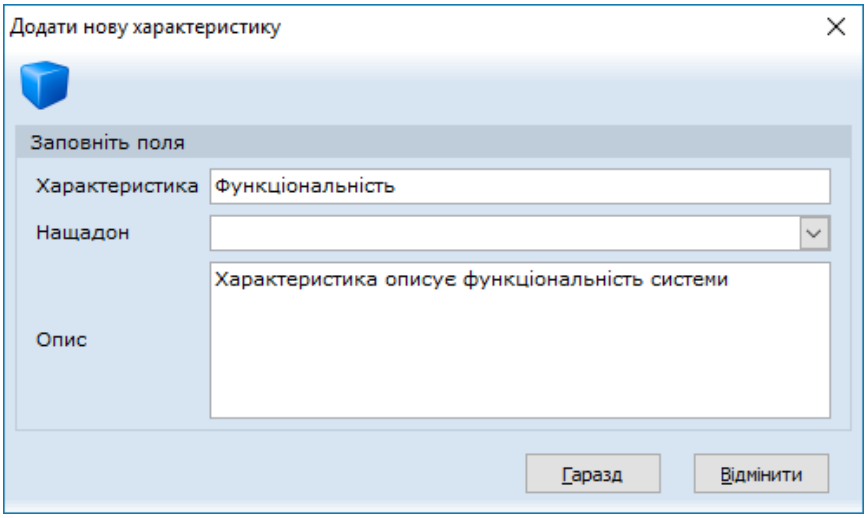

Рис. 3.27. Форма додавання нової характеристики якості

Меню «Атрибути» призначено для формування, редагування та видалення атрибутів, тобто вимог до комп’ютерних систем критичного призначення. Однак додати чи змінити атрибут можна лише після створення відповідної характеристики, до якої його класифікував експерт з якості. На рис. 3.28 наведено меню для роботи з атрибутами комп’ютерних систем критичного призначення, а на рис. 3.29 – вікно додавання та редагування атрибута.

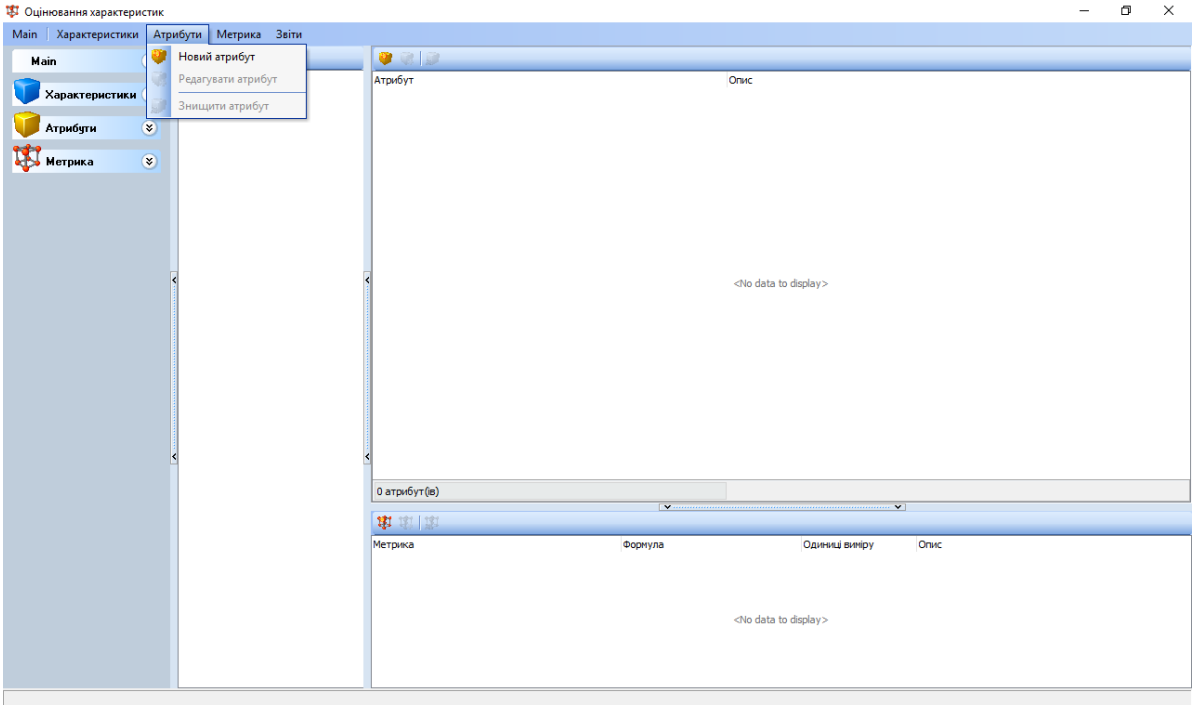

Рис. 3.28. Меню «Атрибути»

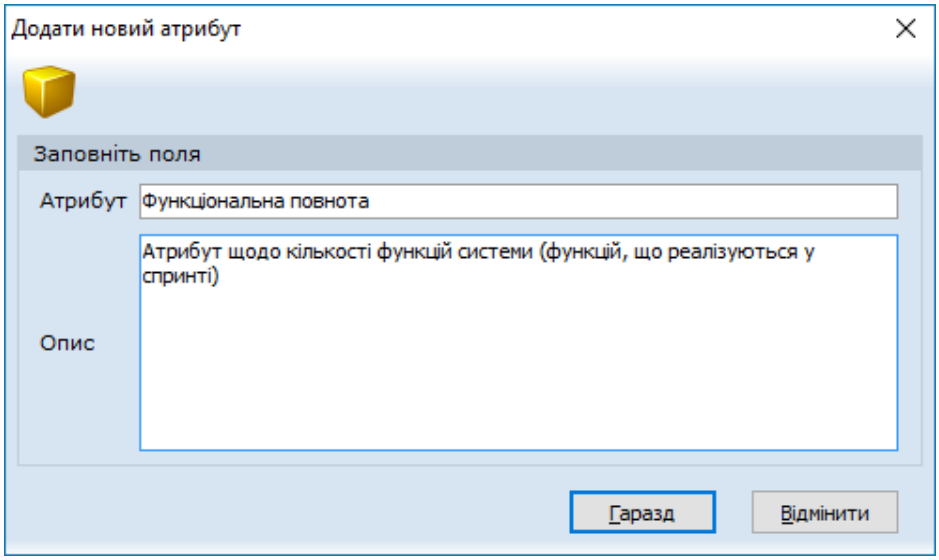

Рис. 3.29. Вікно додавання та редагування атрибутів

Для того, щоб оцінити міру реалізації атрибута та відповідності його вимогам, призначене меню «Метрика», яке дає змогу додати, редагувати та знищити метрики відповідних атрибутів. Слід зазначити, що створити метрику без атрибуту неможливо, оскільки метрика виражає у кількісному вигляді ступінь реалізації атрибуту, тобто вимоги. На рис. 3.30 наведено структуру меню «Метрика», а на рис. 3.31 – вікно додавання та редагування метрики.

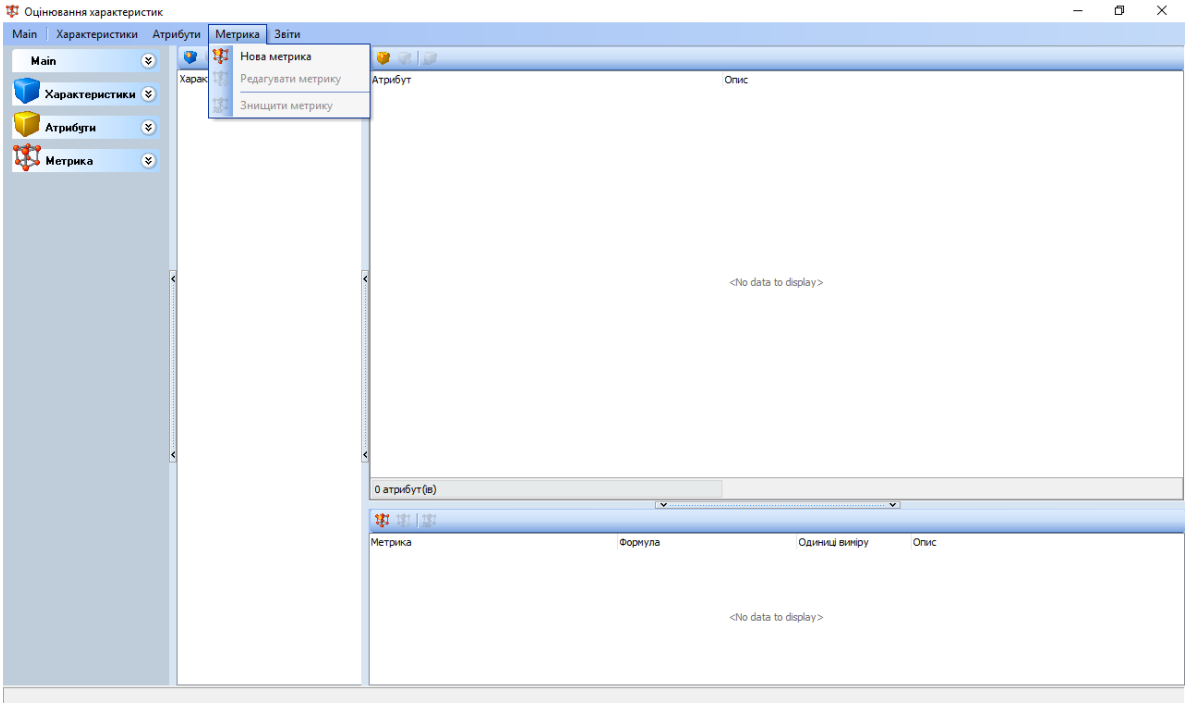

Рис. 3.30. Структура меню «Метрика»

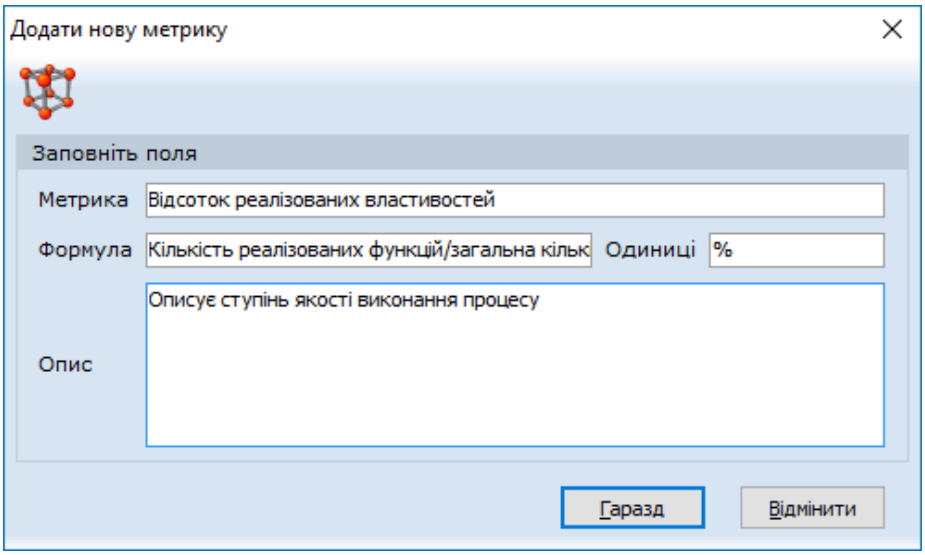

Рис. 3.31. Вікно додавання та редагування метрик

У результаті створення характеристики «Функціональність», атрибуту «Функціональна повнота» і метрики «Відсоток реалізованих функцій» автоматично формується вікно, вигляд якого представлено на рис. 3.32.

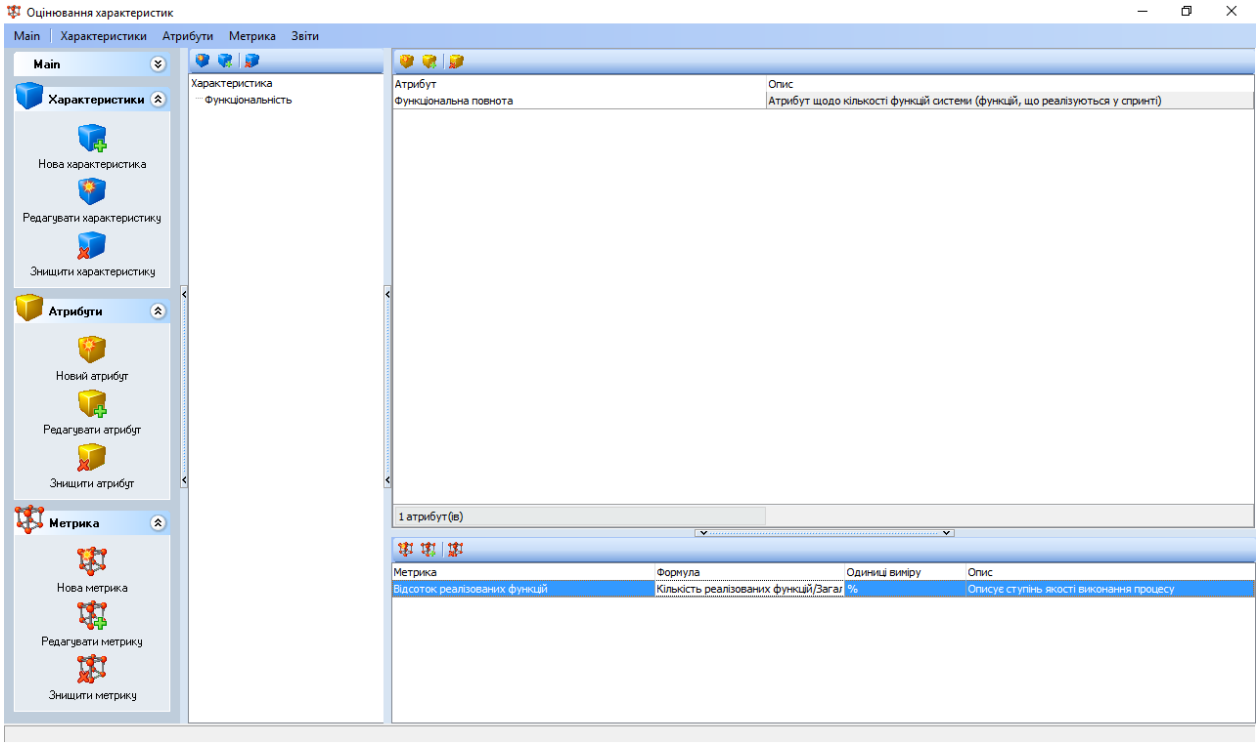

Рис. 3.32. Форма заповненими тестовими даними

Редагування тестових даних можна виконувати за допомогою панелі інструментів, що знаходиться зліва. Повністю дерево (ієрархічну структуру) характеристик комп’ютерних систем критичного призначення можна побачити між панеллю інструментів і табличним відображенням атрибутів і метрик.

Дана форма в залежності від типу користувача, дає змогу формувати критерії якості вимог до комп’ютерних систем критичного призначення та до процесів їх реалізації. Тому актуальним також є розробка інтерфейсу для гнучкого формування звітної документації, по типу специфікації вимог до комп’ютерних систем критичного призначення та результатів реалізації критеріїв якості вимог на стадіях життєвого циклу.

На рис. 3.34 та рис. 3.35 наведено форми для генерації документації по проекту.

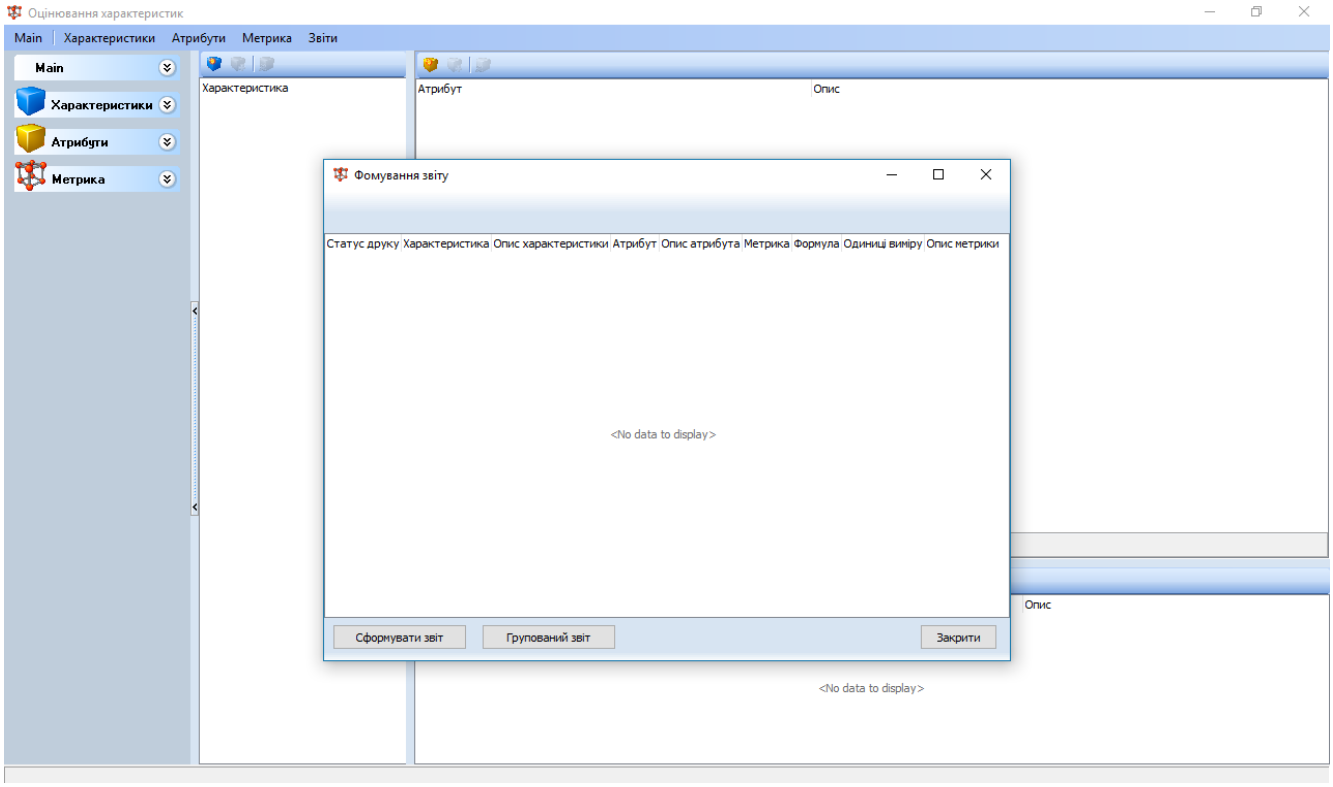

Рис. 3.34. Форма налаштування параметрів формування звітної документації

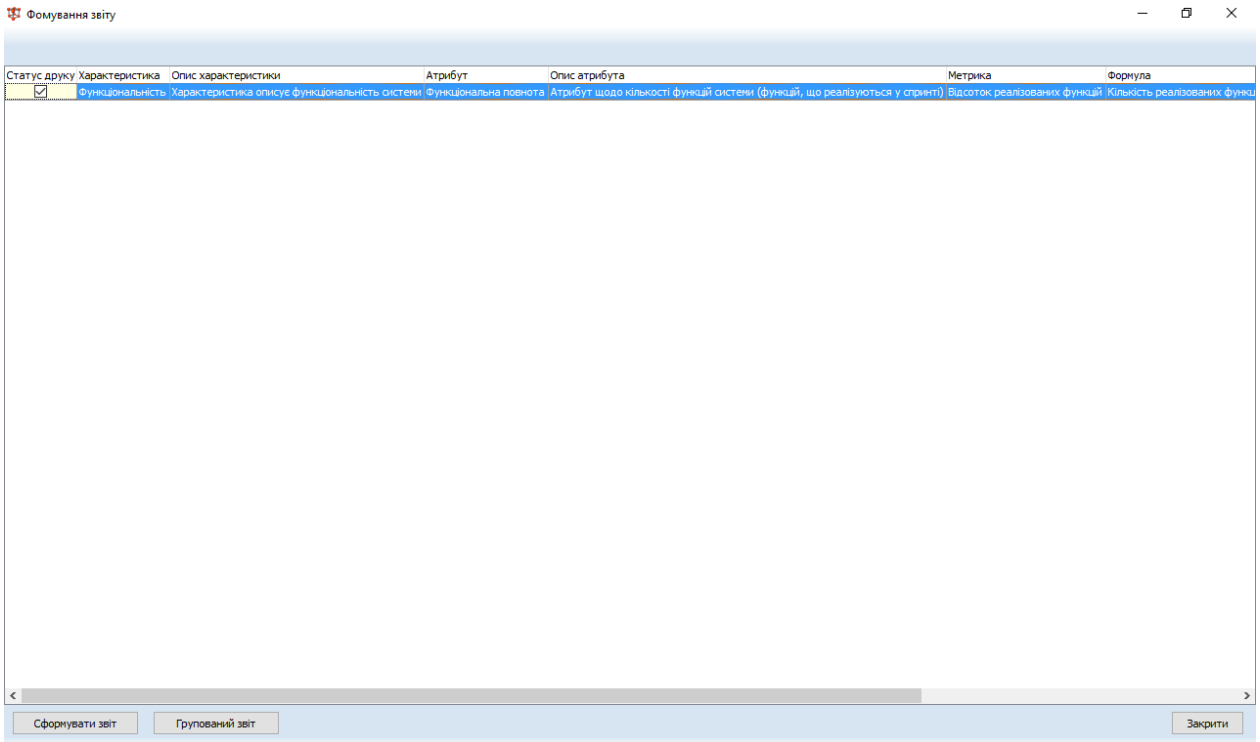

Рис. 3.35. Вікно формування контенту звітної документації

Таким чином, розробивши функціональні вимоги до програмного засобу оптимізації процесів розробки комп’ютерних систем критичного призначення, спроектувавши її архітектуру та користувацькі інтерфейси одержано

повнофункціональний програмний продукт, орієнтований на автоматизацію процесів підвищення якості реалізації таких проектів. У подальшому планується розробка API для інтеграції із системами керування процесом розробки комп’ютерних систем.

3.5. Висновки до розділу

1. У результаті аналізу предметної області та особливостей запропонованого методу, визначено функціональні вимоги до програмного засобу підтримки процесу оптимізації розробки комп’ютерних систем критичного призначення та визначено ролі учасників проекту, що дало змогу врахувати їх при проектуванні архітектури даного засобу.
2. Спроектовано базу даних і побудовано архітектуру програмного засобу підтримки методу оптимізації процесу розробки комп’ютерних систем критичного призначення, що дало змогу забезпечити ефективність керування параметрами якості на різних ітераціях життєвого циклу із застосуванням agile-методологій.
3. Розроблено інтерфейси програмного засобу для забезпечення зручності виконання робіт щодо оптимізації процесів розробки комп’ютерних систем критичного призначення з врахуванням особливостей класичних моделей життєвого циклу та гнучких методологій розробки.

РОЗДІЛ 4
ОБГРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ

Встановлення економічної доцільності дослідження методів і засобів оптимізації процесу розробки комп’ютерних систем критичного призначення є основною метою виконання розрахунків щодо обґрунтування економічної ефективності. Проведемо обчислення критеріїв для визначення економічної доцільності розробки та впровадження розроблених у роботі методу і засобу.

4.1. Розрахунок норм часу на виконання науково-дослідної роботи

При оптимізації процесу розробки комп’ютерних систем критичного призначення ефективності виконання проектів систем реального часу даних необхідно обґрунтувати економічну доцільність науково-дослідної роботи.

Основні етапи при виконанні дипломної роботи магістра можна визначити наступним чином:

– визначення актуальності теми ДР магістра;

– аналіз наукових публікацій та практик розробки комп’ютерних систем на основі класичних і гнучких методологій проектування;

– визначення особливостей процесів створення комп’ютерних систем критичного призначення;

– побудова та обґрунтування моделі процесу розробки комп’ютерних систем критичного призначення;

– розробка методу оптимізації процесів розробки комп’ютерних систем критичного призначення;

– проектування та реалізація архітектури програмного засобу управління критеріями оптимізації процесу розробки комп’ютерних систем критичного призначення;

– створення інструкції з розгортання розробленого програмного засобу;

– оформлення інструкцій.

При оцінюванні тривалості виконання окремих робіт використовують нормативи часу або попередній досвід. До таких нормативів відносять тривалість написання операцій (команд), які для окремих підприємств становлять: для однієї операції – 0,5-1,6 год та 8 годин для п'яти операцій (тривалість зміни).

У разі їх відсутності звертаються до експертних оцінок по встановленню тривалості кожного етапу, яка при трьох оцінках обчислюється за формулою [29]

$$T_{вс} = (t_{min} + 4t_{н.й} + t_{max})/6, \quad (4.1)$$

При двох оцінках, експертна оцінка обчислюється за формулою:

$$T_{вс} = (3t_{min} + 2t_{max})/5, \quad (4.2)$$

де $T_{вс}$ – очікуване (середнє) значення тривалості виконання етапу (стадії);

t_{min} – мінімальна оцінка тривалості виконання етапу;

$t_{н.й}$ – найбільш імовірна оцінка тривалості виконання етапу;

t_{max} – максимальна оцінка тривалості виконання етапу.

Дані витрат часу на виконання окремих стадій (етапів) можна звести у табл. 4.1.

Таблиця 4.1

Основні етапи виконання НДР

№ та назва етапу	Середній час виконання стадії (етапу) інженером, год.	
	Інженер	Керівник
1. Визначення актуальності теми ДР магістра	10	8

Продовж. табл. 4.1

№ та назва етапу	Середній час виконання стадії (етапу) інженером, год.	
	Інженер	Керівник
2. Аналіз наукових публікацій та практик розробки комп'ютерних систем на основі класичних і гнучких методологій проектування	16	6
3. Визначення особливостей процесів створення комп'ютерних систем критичного призначення	14	4
4. Побудова та обґрунтування моделі процесу розробки комп'ютерних систем критичного призначення	18	10
5. Розробка методу оптимізації процесів розробки комп'ютерних систем критичного призначення	28	6
6. Проектування та реалізація архітектури програмного засобу управління критеріями оптимізації процесу розробки комп'ютерних систем критичного призначення	48	2
7. Створення інструкції з розгортання розробленого програмного засобу	18	2
8. Оформлення інструкцій	8	2
Разом	200	40

Витрати часу керівника на виконання окремих стадій (етапів) при недостатній кількості інформації доцільно приймати в межах 5% сумарних витрат часу інженерів на виконання цих стадій (етапів).

4.2. Визначення витрат на оплату праці та відрахувань на соціальні заходи

Відповідно до Закону України «Про оплату праці» заробітна плата – це «винагорода, обчислена, як правило, у грошовому виразі, яку власник або уповноважений ним орган виплачує працівникові за виконану ним роботу».

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та господарської діяльності підприємства. Заробітна плата складається з основної та додаткової оплати праці.

Основна заробітна плата нараховується на виконану роботу за тарифними ставками, відрядними розцінками чи посадовими окладами і не залежить від результатів господарської діяльності підприємства.

Додаткова заробітна плата – це складова заробітної плати працівників, до якої включають витрати на оплату праці, не пов'язані з виплатами за фактично відпрацьований час. Нараховують додаткову заробітну плату залежно від досягнутих і запланованих показників, умов виробництва, кваліфікації виконавців. Джерелом додаткової оплати праці є фонд матеріального стимулювання, який створюється за рахунок прибутку.

Основна з/п складається із прямої з/п і доплати, яка при укрупнених розрахунках становить 25% – 35% від прямої з/п. При розрахунку з/п кількість робочих днів в місяці слід приймати – 21 дні/міс., що відповідає 168 год./міс. Розмір місячних окладів керівника та інженерів слід приймати згідно існуючих на даний час норм. Основна заробітна плата розраховується за формулою:

$$З_{осн.} = T_c \cdot K_Г \qquad (4.3)$$

де T_c – тарифна ставка, грн.;

$K_Г$ – кількість відпрацьованих годин.

Посадові оклади (тарифні ставки) за розрядами Єдиної тарифної сітки визначаються шляхом множення окладу (ставки) працівника 1 тарифного розряду на відповідний тарифний коефіцієнт. У разі коли посадовий оклад (тарифна ставка) визначені у гривнях з копійками, цифри до 0,5 відкидаються, від 0,5 і вище – заокруглюються до однієї гривні. У 2019 році посадові оклади (тарифні ставки) розраховуються згідно з Законом України «Про Державний бюджет України на 2019 рік».

Мінімальна зарплата у 2019 р. складає 4173,00 грн., в погодинному розмірі 25,13 грн., приймемо 90,00 грн. для інженера, для керівника – 150,00 грн.

Для інженера: $З_{осн.} = 90{,}00 \cdot 160 = 14400{,}00$ грн.

Для керівника: $З_{осн.} = 150{,}00 \cdot 40 = 6000{,}00$ грн.

Додаткова заробітна плата становить 10 – 15% від суми основної заробітної плати і обчислюється за формулою

$$З_{дод.} = З_{осн.} \cdot K_{допл.} \tag{4.4}$$

де $K_{допл.}$ – коефіцієнт додаткових виплат (0,1).

Для інженера: $З_{дод.} = 14400{,}00 \cdot 0{,}1 = 1440{,}00{,}00$ грн.

Для керівника: $З_{дод.} = 6000{,}00 \cdot 0{,}1 = 600{,}00$ грн.

Звідси загальні витрати на оплату праці ($B_{ОП.}$) визначаються за формулою

$$B_{ОП.} = З_{осн.} + З_{дод.} \tag{4.5}$$

Для інженера: $B_{ОП.} = 14400{,}00 + 1440{,}00 = 15840{,}00$ грн.

Для керівника: $B_{ОП.} = 6000{,}00 + 600{,}00 = 6600{,}00$ грн.

Таким чином загальна сума становить 22440,00 грн. Крім того, необхідно визначити відрахування на соціальні заходи:

– податок на доходи фізичних осіб: 18% – 4039,20 грн.;
– військовий збір: 1,5% – 336,60 грн.;

– єдиний внесок: 22% – 4936,80 грн.

У сумі зазначені відрахування становлять 41,5%. Отже, загальна сума відрахувань на соціальні заходи становитиме:

$$B_{С.З.} = ФОП \cdot 0{,}415 \qquad (4.6)$$

де $ФОП$ – фонд оплати праці, грн.

У даному випадку сума відрахувань становить: $B_{С.З.} = 22440{,}00 \cdot 0{,}415 = 9312{,}60$ грн.

Проведені розрахунки витрат на оплату праці зведемо у табл. 4.2.

Таблиця 4.2

Зведені витрати на заробітну плату

Категорія працівників	Основна заробітна плата, грн.			Додаткова заробітна плата, грн.	Нарахування на ФОП, грн.	Всього витрати на оплату праці, грн.
	Тарифна ставка, грн.	К-сть відпрацьованих	Фактично нарах. з/пл., грн.			
Інженер	90	160	14400,00	1440,00	6573,60	22413,60
Керівник проекту	150	40	6000,00	600,00	2739,00	9339,00
Разом			20400,00	2040,00	9312,60	31752,60

4.3. Розрахунок витрат на електроенергію

Затрати на електроенергію при використанні обладнання визначаються за формулою:

$$З_е = W \cdot T \cdot S \qquad (4.7)$$

де W – необхідна потужність, кВт;

T – кількість годин роботи обладнання;

S – вартість кіловат-години електроенергії.

Згідно з постановою НКРЕКП України від 05.10.2018 р. № 1177 вартість електроенергії становить 243,71 коп./кВт.год.

Потужність комп'ютера – 800 Вт, а кількість годин роботи обладнання згідно табл. 4.1 – 200 години.

Затрати на електроенергію становлять: $З_e = 0{,}8 \cdot 200 \cdot 2{,}4371 = 389{,}94$ грн.

4.4. Розрахунок витрат на матеріали

Результати розрахунку затрат на матеріали наведено у табл. 4.3.

Таблиця 4.3

Визначення величини затрат на матеріал

Найменування матеріальних ресурсів	Одиниця виміру	Норма витрат	Ціна за одиницю, грн	Затрати матеріалів, грн	Транспортно-	Загальна сума витрат на матеріали, грн
Компоненти MS Visual Studio	шт.	1	24200	24200	-	24200
Компакт-диски	шт.	2	10	20	-	20
Разом						24220

4.5. Розрахунок суми амортизаційних відрахувань

Характерною особливістю застосування основних фондів у процесі виробництва є їх відновлення. Для відновлення засобів праці у натуральному

виразі необхідне їх відшкодування у вартісній формі, яке здійснюється шляхом амортизації.

Амортизація – це процес перенесення вартості основних фондів на вартість новоствореної продукції з метою їх повного відновлення.

Комп'ютери та оргтехніка належать до четвертої групи основних фондів. Для цієї групи річна норма амортизації дорівнює 60 % (квартальна – 15 %).

Для визначення амортизаційних відрахувань застосовуємо формулу:

$$A = \frac{Б_в \cdot Н_А}{100\%} \tag{4.8}$$

де A – амортизаційні відрахування за звітний період, грн.,

$Б_в$ – балансова вартість комп'ютера, на початок звітного періоду, грн..

$Н_А$ – норма амортизації, яку приймемо на рівні 15%.

Амортизаційні відрахування при балансовій вартості ПК у 25000 грн. та нормі амортизації на рівні 15%, амортизаційні відрахування становитимуть: $A = \frac{25000 \cdot 15\%}{100\%} = 3750{,}00$ грн.

4.6. Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці.

Накладні витрати можна встановити на рівні 20% від суми основної та додаткової заробітної плати працівників:

$$Н_В = В_{ОП} \cdot 0{,}2 \tag{4.9}$$

де $Н_В$ – накладні витрати, грн.,

$В_{ОП}$ – суми основної та додаткової заробітної плати працівників, грн..

У даному випадку накладні витрати становитимуть:

$H_B = 22440{,}00 \cdot 0{,}2 = 4488{,}00$ грн.

4.7. Складання кошторису витрат та визначення собівартості науково-дослідних робіт

Собівартість (C_B) науково-дослідних робіт розрахуємо за формулою:

$$C_B = B_{о.п.} + B_{с.з} + З_{м.в.} + З_е + Т_в + А + Н_в \qquad (4.10)$$

У даному випадку собівартість (СВ) науково-дослідних робіт розрахуємо за формулою:

$C_B = 22440{,}00 + 9312{,}60 + 24220{,}00 + 389{,}94 + 3750{,}00 + 4488{,}00 = 64600{,}54$ грн

Результати проведених вище розрахунків зведемо у табл. 4.4.

Таблиця 4.4

Кошторис витрат на науково-дослідних робіт

Зміст витрат	Сума, грн.	В % до загальної суми
Витрати на оплату праці (основну і додаткову заробітну плату)	22440,00	34,74%
Відрахування на соціальні заходи	9312,60	14,42%
Матеріальні витрати	24220	37,49%
Витрати на електроенергію	389,94	0,60%
Амортизаційні відрахування	3750,00	5,80%
Накладні витрати	4488,00	6,95%
Собівартість	64600,54	100,00%

4.8. Розрахунок ціни науково-дослідних робіт

Ціну науково-дослідних робіт можна визначити за формулою:

$$Ц = \frac{C_в \cdot (1 + P_{рен.}) + K \cdot B_{н.і.}}{K} \cdot (1 + ПДВ), \qquad (4.11)$$

де $P_{рен.}$ – рівень рентабельності, 26 %;

K – кількість замовлень, од. (встановлюється лише при розробці програмного продукту та мікропроцесорних систем);

$B_{н.і.}$ – вартість носія інформації, грн. (встановлюється лише при розробці програмного продукту);

ПДВ – ставка податку на додану вартість, (20 %).

Оскільки розробка є прикладною, і використовуватиметься тільки для одного підприємства, то для розрахунку ціни не потрібно вказувати коефіцієнти K та $B_{н.і.}$, оскільки їх в даному випадку не потрібно.

Тоді, формула для обчислення ціни розробки буде мати вигляд:

$$Ц = C_в \cdot (1 + P_{рен.}) \cdot (1 + ПДВ) \qquad (4.12)$$

Ціна НДР становитиме: $Ц = 64600{,}54 \cdot (1 + 0{,}26) \cdot (1 + 0{,}2) = 97676{,}01$ грн.

Визначимо величину прибутку:

$$П = Ц - C_в \qquad (4.13)$$

Прибуток буде становити: $П = 97676{,}01 - 64600{,}54 = 33075{,}47$ грн.

4.9. Визначення економічної ефективності і терміну окупності капітальних вкладень

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу.

Економічна ефективність (E_p) полягає у відношенні результату виробництва до затрачених ресурсів:

$$E_p = \frac{П}{C_в} \tag{4.14}$$

де $П$ – прибуток;

$C_В$ – собівартість.

Економічна ефективність становить:

$$E_p = \frac{33075,47}{97676,01} = 0,51.$$

Поряд із економічною ефективністю розраховують термін окупності капітальних вкладень (T_p):

$$T_p = \frac{1}{E_p} \tag{4.15}$$

В даному випадку термін окупності становить: $T_p = \frac{1}{0,51} = 1,95$ року.

Про доцільність розробки програми можна сказати при врахуванні критеріїв, які наведені у табл. 4.5.

Таблиця 4.5

Техніко-економічні показники НДР

№ з/п	Показник	Значення
1	Собівартість, грн	64600,54
2	Плановий прибуток, грн	33075,47
3	Ціна, грн	97676,01
4	Економічна ефективність	0,51
5	Термін окупності, рік	1,95

При розробці та впровадженні методу і засобу підтримки процесу оптимізації розробки комп’ютерних систем критичного призначення плановий очікуваний прибуток становить 33075,47 грн., при собівартості 64600,54 грн. Термін окупності капітальних вкладень становить близько двох років, що дає змогу обґрунтувати економічну доцільність проведення НДР.

РОЗДІЛ 5
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

5.1. Охорона праці

Тема дипломної роботи магістра пов’язана із дослідженням методів і засобів оптимізації процесів розробки комп’ютерних систем критичного призначення. Розробники таких систем повинні дотримуватись вимог нормативних документів щодо охорони праці і безпеки експлуатації електронно-обчислювальних пристроїв, оскільки це безпосередньо впливає на стан їхнього здоров’я.

Основним нормативним документом, який регламентує норми і правила експлуатації електронно-обчислювальної техніки є НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями». Правила встановлюють вимоги безпеки до обладнання робочих місць операторів ЕОМ. Вимоги НПАОП 0.00-7.15-18 є обов'язковими для роботодавців, операторів електронно-обчислювальних машин, операторів комп'ютерного набору, операторів комп'ютерної верстки та працівників інших професій, які у своїй роботі застосовують ЕОМ з ВДТ і ПП [25].

Згідно з [25] площа, на якій розташовується одне робоче місце, облаштоване комп’ютером, повинна становити не менше як 6.0 м^2, об'ємом не менше як 20 м^3.

Робочі місця інженерів та експертів при визначенні пріоритетів характеристик комп’ютерних систем обладнанні ПК, що розташовуються на відстані від стін – на 1 м; між собою на відстані 1,7 м, що відповідає вимогам до організації робочих місць, обладнаних комп’ютерами.

З метою запобігання нещасним випадкам та організації охорони праці на виробництві розробляються інструкції з охорони праці і техніки безпеки при використанні комп’ютерної техніки.

До роботи на ПК допускаються особи, які пройшли спеціальне навчання, медичне обстеження, вступний інструктаж з охорони праці, інструктаж на робочому місці та інструктаж з пожежної безпеки.

Згідно із розробленою інструкцією, відповідальний працівник за охорону праці і техніку безпеки при експлуатації системи оптимізації процесу розробки комп’ютерних систем критичного призначення, повинен дотримуватись правил внутрішнього трудового розпорядку, не допускати в робочу зону сторонніх осіб, не виконувати вказівок, які суперечать правилам охорони праці та пам'ятати про особисту відповідальність за виконання правил охорони праці та безпеку товаришів по роботі.

Для усіх працівників, задіяних у процесі оптимізації розробки комп’ютерних систем критичного призначення, поставлено вимоги щодо вміння надавати першу медичну допомогу потерпілим від нещасних випадків, користування первинними засобами пожежогасіння та дотримання правил особистої гігієни.

Основними небезпечними та шкідливими виробничими факторами, які впливають на користувачів системи підтримки процесу оптимізації комп’ютерних систем критичного призначення є:

– підвищений рівень шуму на робочому місці (від вентиляторів, процесорів та аудіоплат);

– можливе підвищене значення напруги в електричному колі, замикання якого може статися через тіло людини;

– підвищений рівень статичної електрики;

– підвищений рівень електромагнітного випромінення;

– підвищена напруженість електричного поля;

– прямий та відбитий від екранів блиск; несприятливий розподіл яскравості в полі зору;

– фізичні перевантаження статичної та динамічної дії;

– нервово-психічні перевантаження (розумове перенапруження, перенапруження аналізаторів, монотонність праці, емоційні перевантаження).

При виборі місць розташування ПК враховано ступінь відбиття світла на екранах дисплеїв, яке проходить через вікна і яке може викликати значне осліплення в тих, хто сидить перед ними, особливо влітку та в сонячні дні. Тому, ПК і оргтехніка розміщені біля стін, які не знаходяться біля вікон або навпроти них.

Оскільки, при незадовільному освітленні знижується продуктивність праці розробників комп'ютерних систем критичного призначення, і можливі негативні впливи на здоров'я такі, як короткозорість, швидка втомленість, тому всі приміщення медзакладу, які облаштовані робочими місцями з ПК, мають природне і штучне освітлення. Не допускається розташування робочих місць з ПК в підвальних приміщеннях.

Робочі місця з ПК при виконанні творчої роботи, яка потребує значної розумової напруги чи великої концентрації уваги, ізольовано одне від одного перегородкою висотою 1,6 м.

Поверхня підлоги у приміщеннях повинна бути оздоблена керамічною плиткою і бути рівною та зручною для очищення та вологого прибирання.

Штучне освітлення у приміщеннях повинно бути виконано у вигляді комбінованої системи освітлення з використанням люмінесцентних джерел світла у світильниках загального освітлення, які розташовувати над робочими поверхнями у рівномірно-прямокутному порядку. Штучне освітлення забезпечує на робочих місцях з ПК освітленість 300 – 500 лк.

Для запобігання засвітленню екранів ПК прямими світловими потоками лінії світильників розташовані з достатнім бічним зміщенням відносно рядів робочих місць, а також паралельно до світлових отворів. При цьому кожне вікно повинно мати світлорозсіюючі штори з коефіцієнтом відбивання 0,7.

У приміщенні також необхідно забезпечити і природнє освітлення, при цьому на кожному вікні закріплені жалюзі з вертикальними ламелями, що регулюються для зменшення прямого попадання сонячного світла на екран комп'ютерів.

З метою мінімізації відблисків від клавіатури та екрану застосовуються спеціальні захисні козирки і джерела штучного світла розташовано паралельно

напрямку погляду на екран ПК з обох сторін.

Для запобігання засліпленню користувачів світильники місцевого освітлення молочного кольору. Захисний кут відбивача світильника становить 60^0.

Для оздоблення приміщень використовують дифузно-відзеркалюючі матеріали з коефіцієнтами відбиття: стелі - 0,8; стін - 0,5; підлоги - 0,3, що відповідає чинним вимогам [25].

На робочому місці розробників комп’ютерних систем критичного призначення розміщується монітор, клавіатура, робочий стіл, крісло. При розташуванні елементів робочого місця враховано наступні фактори:

– робочу позу користувача;
– простір для розміщення користувача;
– можливість огляду елементів робочого місця;
– можливість огляду простору за межами робочого місця;
– можливість робити записи, розміщення документації і матеріалів, які використовує користувач.

Взаємне розташування елементів робочого місця не заважає виконанню всіх необхідних рухів та переміщень для експлуатації ПК. Отже, при розробці методів і засобів оптимізації процесу розробки комп’ютерних систем критичного призначення, враховано необхідні вимоги щодо охорони праці при використанні електронно-обчислювальної техніки і забезпечено умови для зручної та ефективної роботи працівників.

5.2. Забезпечення надійності управління і матеріально-технічного постачання суб’єкта господарювання

Надійність управління і матеріально-технічне постачання суб’єкта господарювання впливає на стійкість роботи об’єкта господарської діяльності при виникненні надзвичайних ситуацій. Стійкість об’єкта визначає його спроможність в умовах надзвичайної ситуації випускати продукцію в запланованому обсязі та номенклатурі, а при отриманні середніх руйнувань або

порушенні зв'язків з кооперації та поставок відновлювати виробництво у мінімальні терміни.

На стійкість роботи ОГД в умовах НС впливають наступні фактори:

– надійність захисту робітників та службовців;

– спроможність інженерно-технічного комплексу об'єкта протистояти у визначеному ступеню уражаючих факторів стихійного лиха, аварій, катастроф та сучасних видів зброї;

– захищеність об'єкта від вторинних уражаючих факторів (пожеж, вибухів, зараження ОР та СДОР);

– надійність системи забезпечення об'єкта всім необхідним для виробництва (сировиною, паливом, комплектуючими вузлами і деталями, електроенергією, водою, газом та іншим);

– стійкість та безперервність управління виробництвом та ЦО;

– підготовленість об'єкта до ведення РІНР та робіт щодо порушеного виробництва.

Перелічені фактори є основними загальними для усіх ОГД шляхи підвищення стійкості роботи в умовах НС, а саме:

– забезпечення надійного захисту робітників та службовців від уражаючих факторів сучасної зброї, аварії, катастрофи і стихійного лиха;

– захист основних виробничих факторів від уражаючих факторів, в тому числі і від вторинних, які виникають в умовах НС;

– стійке забезпечення всім необхідним для випуску запланованої продукції;

– підготовка до відновлення порушеного виробництва;

– підвищення надійності та оперативності управління виробництвом та ЦО.

Захист робітників та службовців досягається чотирма основними способами:

– укриття людей в захисних спорудах;

– проведення евакозаходів;

– радіаційно-хімічний захист;

– медичний і біологічний захист.

Надійно захистити виробничий персонал об'єкта можливо лише при комплексному використанні усіх основних способів захисту.

Захист виробничих фондів полягає у підвищенні протидії будинків, споруд і конструкції об'єкта до уражаючих факторів та захисті технологічного обладнання, верстатів, систем і комунікацій та інших засобів, що формують основу виробничого процесу.

Створення надійних систем електро-, водо- та теплозабезпечення об'єктів:

а) підвищення стійкості електрозабезпечення:

– розподіл схеми електромереж на незалежно працюючі частини;

– закільцювання електромереж та підключення їх до декількох джерел енергозабезпечення;

– створення резерву дизельних електростанцій;

б) підвищення стійкості систем водопостачання:

– водопостачання від двох незалежних джерел, одне з яких підземне;

– захист вододжерел та резервуарів чистої води;

– створення обвідних (байпасних) ліній навколо водонапірних веж;

в) підвищення стійкості систем газо, тепло- та паливо- забезпечення:

– розподільні газопроводи робити підземними та передбачати їх кільцювання:

– газорозподільні станції та опорні пункти обвідних газопроводів передбачати в підземному варіанті;

– встановлювати в основних вузлових точках систем газозабезпечення;

– автоматичні вимикаючі пристрої, як. спрацьовують при аваріях.

Підвищення протипожежної стійкості:

– максимальне скорочення запасів паливо та вибухонебезпечних речовин;

– проведення профілактичних протипожежних заходів;

– підготовка сил і засобів пожежогасіння.

Створення стійкості системи матеріально-технічного постачання. На ОГД створюють запаси сировини, палива, комплектуючих вузлів і деталей, обладнання, які дозволяють продовжувати роботу на випадок дезорганізації постачання.

Створення стійкості системи управління:

– підготовка ПУ (захищених);
– забезпечення ПУ засобами зв'язку;
– використання автоматизованої системи управління.

Підготовка до прискореного (негайного) відновлення порушеного виробництва:

– розробка необхідної технічної та технологічної документації;
– створення запасів матеріальних засобів для встановлення робіт;
– розробка розрахунків сил і засобів для відновлюючих робіт;
– визначення вірогідної черговості робіт по відновленню виробництва з урахуванням наявних ресурсів та місцевих умов.

Крім того, на стійкість роботи ОГД буде впливати наявність підготовленої робочої сили. Підвищення надійності та оперативності управління виробництвом:

– створення на об'єкті стійкої системи зв'язку;
– висока підготовка керівного складу;
– своєчасне прийнятті ,ірних рішень та постановка завдань підлеглим у відповідності до обстановки, що склалася.

Підвищення стійкості роботи ОГД досягається завчасним проведенням комплексу інженерно-технічних, технологічних та організаційних заходів, які спрямовані на максимальне зниження дії уражаючих факторів і створення умов для ліквідації наслідків АС.

Інженерно-технічні заходи — це комплекс робіт, що забезпечують підвищення стійкості виробничих будинків і споруд, обладнання, комунально-енергетичних, систем.

Технологічні заходи забезпечують підвищення стійкості роботи об'єкті шляхом зміни технологічного процесу, що сприяє спрощенню виробництва продукції та усуває можливість виникнення вторинних уражаючих факторів'.

Організаційні заходи передбачають розробку і планування дій керівного командно-начальницького складу штабу, служб і формування ЦО при захисті робітників і службовців, відновленні виробництва.

5.3. Проведення розрахунку режимів роботи об’єкту в умовах радіоактивного забруднення при аваріях на АЕС

Головними показниками ступеня небезпеки радіоактивного зараження для населення є розміри зон радіоактивного забруднення та рівні радіації. Розміри зон радіоактивного забруднення та еталонний рівень радіації у разі ядерного вибуху визначаються залежно від потужності, різновиду вибуху та швидкості вітру за допомогою таблиць, радіаційних РЛ, дозиметричних ДЛ розрахункових лінійок.

Згідно з табл. 5.1 залежно від потужності вибуху визначаються розміри забруднених зон із навітряної сторони, а у відповідності до табл. 5.2. залежно від потужності вибуху і швидкості середнього вітру визначаються розміри забруднених зон із підвітряної сторони й еталонний рівень радіації на 1-у годину.

Таблиця 5.1

Радіуси зон забруднення в районі наземного вибуху з навітряної сторони

Потужність вибуху, Мт	Зони забруднення			Потужність вибуху, Мт	Зони забруднення		
	А	Б	В		А	Б	В
0,02	735	450	340	0,2	1040	735	595
0,05	865	560	430	0,5	1220	865	710
0,10	970	645	510	1,0	1290	930	770

У разі аварії на АЕС прогнозування радіаційної обстановки здійснюється визначенням зон (індекс зони) радіаційного забруднення та їхніх розмірів залежно від типу реактора, виходу активності, метеоумов (СВСА, швидкість вітру) за табл. 5.2. і табл. 5.3.

Таблиця 5.2

Розміри прогнозованих зон радіоактивного забруднення місцевості за слідом хмари після аварії на АЕС

Потужність вибуху, Мт	Швидкість середнього вітру, $V_в$, км/год.	Розмір зон і еталонний рівень радіації на зовнішній межі зон на 1-у годину після вибуху, км		
		А – 8 Р/год	Б – 80 Р/год	В – 240 Р/год
0,1	25	116*12	49*6,1	31*4
	50	150*14	60*6,4	25*3,9
	75	175*15	64*6,3	35*3,8
0,2	25	157*15	67*7,8	43*5,3
	50	200*18	83*8,3	50*5,3
	75	233*20	90*8,4	50*5,0
0,3	25	190*18	80*8,0	52*6,0
	50	240*21	98*9,6	60*6,2
	75	275*23	100*9,8	60*6,0
0,5	25	231*21	100*10	65*7,4
	50	300*25	121*12	78*7,7
	75	340*27	140*13,5	83*7,7
1,0	25	309*20	132*13,0	83*9,5
	50	402*31	170*15,0	109*10,0
	75	466*34	192*16,0	118*10,0
2,0	25	413*32	182*17	121*12
	50	536*39	231*19,6	149*13
	75	626*43	262*21	165*13
3,0	25	495*37	218*19	145*14
	50	630*45	275*23	180*15
	75	750*50	310*24	200*16

Продовж. табл. 5.2

Потужність вибуху, Мт	Швидкість середнього вітру, $V_в$, км/год.	Розмір зон і еталонний рівень радіації на зовнішній межі зон на 1-у годину після вибуху, км		
		А – 8 Р/год	Б – 80 Р/год	В – 240 Р/год
5,0	25	772*52	343*27	225*19
	50	920*58	393*28	253*20
	75	1035*62	436*30	270*20

*Примітка. Перша група чисел – довжина зони, друга – ширина

Визначені зони радіоактивного забруднення з урахуванням напряму вітру наносяться на карту або схему в певному масштабі, а з навітряної сторони на рівні вісі сліду радіоактивної хмари у чисельнику вказуються у разі аварії на АЕС –тип реактора і процент виходу активності (у разі ядерного вибуху – різновид вибуху і його потужність), у знаменнику – час і дату вибуху або викиду.

Таблиця 5.3

Розміри прогнозованих зон радіоактивного забруднення місцевості за слідом хмари після аварії на АЕС (категорія стійкості –конвекція, швидкість вітру *Vв* = 2 м/с)*

Вихід активності, %	Індекс зони	Реактор			
		РВПК-1000		ВВЕР-1000	
		Довжина, км	Ширина, км	Довжина, км	Ширина, км
3	М	62,5	12,1	82,5	16,2
3	А	14,1	2,7	13	2,2
10	М	140	29,9	185	40,2
10	А	28	5,9	39,4	6,8
10	Б	6,8	0,8		
30	М	249	61,8	338	82,9
30	А	62,6	12,1	82,8	15,4
30	Б	13,9	2,7	17,1	2,5

Продовж. табл. 5.3

Вихід активності, %	Індекс зони	Реактор			
		РВПК-1000		ВВЕР-1000	
		Довжина, км	Ширина, км	Довжина, км	Ширина, км
30	В	6,9	0,8		
50	М	324	81,8	438	111
50	А	883	18,1	123	24,6
50	Б	18,3	3,6	20,4	3,7
50	В	9,21	1,5	838	1,07

*Примітка. Відсутність даних про розміри зон радіоактивного забруднення свідчить про те, що зони не утворюються.

На рис. 5.1 наведено схему прогнозування зон радіоактивного забруднення

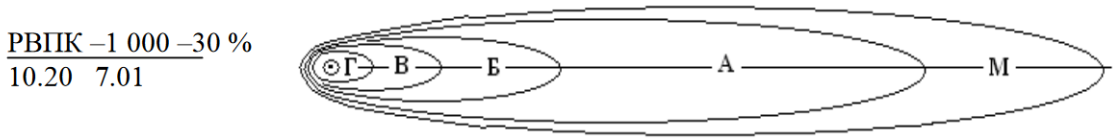

Рис. 5.1. Схема прогнозу зон радіоактивного забруднення при аварії на АЕС

Варто зазначити, що залежно від ступеня радіоактивного забруднення прийнято всю заражену територію внаслідок ядерного вибуху умовно поділяти на 4 зони забруднення Г, В, Б, А. Ці зони на схемі прогнозу позначаються певним кольором: зона А – синім, Б – зеленим, В – коричневим, Г – чорним. Необхідно відмітити, що у разі радіоактивного забруднення внаслідок аварії на АЕС передбачена також зона М –зона радіаційної небезпеки, яка на схемі прогнозу радіоактивного забруднення позначається червоним кольором.

Середнє значення потужності дози випромінювання (рівень радіації) на зовнішніх межах зон забруднення залежно від часу після викиду у разі аварії на АЕС може бути визначне за спеціальними таблицями.

Далі, якщо відомий виміряний рівень радіації шляхом порівняння з його граничним (середнім) значенням можливо орієнтовно визначити зону, де знаходиться потрібний об’єкт або формування цивільного захисту.

РОЗДІЛ 6
ЕКОЛОГІЯ

6.1. Комплексна оцінка екологічності виробництва

Під комплексною оцінкою екологічності виробництва (ЕВ) розуміють інтегральний показник, який виражає рівень екологічності господарської діяльності з урахуванням чинника техногенної безпеки у взаємозв'язку з виробничими ресурсами, умовами і фінансово-економічними результатами господарської діяльності [29]. Комплексну оцінку ЕВ підприємства розглядають як характеристику, отриману в результаті дослідження, що містить висновки про результати екологічної діяльності підприємства, галузі, регіону.

Комплексна оцінка ЕВ підприємства використовується як:

– інструмент обліку, аналізу, планування і регулювання; показник еколого-економічного стану господарського об'єкта;

– критерій порівняльної оцінки екологічності виробництва різних об'єктів;

– показник ефективності прийнятих управлінських рішень у сфері природокористування й охорони навколишнього середовища, а також повноти їх реалізації.

Таким чином, оцінювання екологічної діяльності суб'єктів господарювання, ЕВ і стану соціально-еколого-економічної системи проводиться за одним показником (критерієм), який характеризує всі сторони функціонування об'єкта [29]. Отримання комплексної оцінки екологічної діяльності підприємства та ЕВ на основі системи показників має елемент порівняння (як і комплексна оцінка господарської діяльності).

Однак комплексне оцінювання ЕВ має задовольняти такі вимоги:

– виражати сутність виробничих та еколого-економічних відносин;

– охоплювати головні сторони виробничо-господарської та екологічної діяльності підприємства;

– використовувати обмежену кількості узагальнених еколого-економічних показників;

– бути еластичною - побічно визначати динаміку суспільне необхідних (повних) витрат у сфері природокористування і охорони навколишнього середовища;

– забезпечувати порівнянність показників у часі та просторі.

Методологічною основою оцінювання складових ЕВ виступає індексний метод. За допомогою індексів (у межах від 0 до 1) можна характеризувати наближення того чи іншого показника до необхідного.

Процедура комплексної порівняльної оцінки ЕВ виконується у вигляді таких відносно самостійних етапів:

– поставлення цілей і завдань комплексного оцінювання ЕВ, включаючи вибір підприємств і видів їх виробничо-економічної діяльності;

– обґрунтування та вибір системи еколого-економічних і фінансово-економічних показників;

– організація збирання вихідної інформації, розрахунку і оцінки окремих показників і вагових коефіцієнтів;

– вибір об'єкта як бази для порівняння;

– розроблення алгоритму і розрахунку комплексних показників ЕВ;

– перевірка адекватності комплексних узагальнених оцінок еколого-економічної ситуації;

– аналіз і використання порівняльних комплексних рейтингових оцінок у процесі прийняття управлінських рішень щодо екологізації промислового виробництва.

Реалізація методів порівняльної комплексної рейтингової оцінки передбачає наявність бази порівняння. В економічному аналізі використовуються поняття підрозділу-еталона, підприємства-еталона або об'єкта-еталона [30].

Коефіцієнт відносної вагомості (важливості) тих чи інших еколого-економічних показників може визначатися статистичним, математичним методами або методом експертних оцінок.

Вимірювання рівня екологічності виробництва та екобезпечності підприємств здійснюється за системою показників, що є основою для вивчення, регулювання та вдосконалення ЕВ та *еколого*-економічного рівня господарювання (ЕЕРГ).

Доцільність використання комплексного показника ЕВ, на відміну від інтегрального показника впливу виробництва на навколишнє середовище у формі економічного збитку, для цілей екологічного регулювання виробництва пояснюється тим фактом, що величина збитку тісно пов'язана з організаційно-технічним рівнем виробництва, з місцем розміщення підприємства тощо [30].

6.2. Методи узагальнення екологічної інформації

Екологічні дослідження складаються з трьох етапів: збору інформації, обробки інформації та узагальнення інформації. Останній етап – це та стадія роботи з екологічною інформацією, коли оброблений екологічний матеріал потребує узагальнення, наочного подання і відображення складних екологічних ситуацій.

Методами наочного подання та викладання фізичних величин, що використовують для більш раціонального та систематизованого викладення цифрової інформації є статистичні таблиці і статистичні графіки, тобто табличний і графічний метод узагальнення екологічної інформації.

Статистичні таблиці є формою узагальнення, раціонального та систематизованого викладення цифрової інформації. Основною перевагою цифрової інформації, зведеної в таблиці, є компактність, наочність, виразність. Інформація стає легкодоступною і рельєфною, компактною і раціональною.

За допомогою таблиць можна досягти узагальнення та систематизації цифрової інформації, полегшити і прискорити ефект її сприйняття,

інтенсифікувати пізнавальний процес явища, зекономити місце при викладенні інформації.

З точки зору узагальнення інформації, таблиця за своїм логічним змістом розглядається як «статистичне речення», що має свій підмет і присудок [30]. Підмет таблиці характеризує об'єкт дослідження, а присудок - це система показників, що відображає підмет як об'єкт.

Статистична таблиця має ряд горизонтальних рядків і вертикальних граф. Перетин рядків і граф утворює клітини таблиці. Ліві бічні і верхні клітини призначені для словесних заголовків, а решта – для числових.

Кожна таблиця має три заголовки:

– загальний – відображає зміст таблиці (його місце над таблицею);

– ліві (бічні) – найменування рядків, розкривають зміст підмета;

– верхні – найменування граф, розкривають зміст присудка.

Оптимальний за розміром об'єм таблиці складає 25-30 графо-клітин, який є добутком рядків і граф.

Іншим способом узагальнення екологічної інформації є застосування графіків, які будуються на основі статистичних даних. Статистичний графік є рисунком, який описує статистичні сукупності умовною мовою геометричних знаків тієї чи іншої форми: крапок, ліній, площин, фігур та різних їх комбінацій. Статистичні графіки лежать в основі графічного методу узагальнення інформації.

Графічний метод узагальнення екологічної інформації використовують для популяризація цифрових даних та забезпечення доступності їх сприйняття.

У більшості випадків статистичні графіки використовують не об’ємне зображення, складне за побудовою, а площинне.

Графіки за призначенням поділяють на аналітичні, які дають можливість порівняти графічні образи, ілюстративні – допомагають при порівняні геометричних фігур та показують зміну розмірів явищ, а також інформаційні, які містять інформацію лише про об'єкт вивчення. Графічні образи на графіках

поділяють на крапкові, лінійні; площинні (квадратні, кругові), просторові (об'ємні), зображувальні (фігурні).

Застосування табличних і графічних методів узагальнення екологічної інформації дають змогу наочно представити результати екологічних досліджень, порівняти цільові показники в часі, а також задовольнити вимоги статистичного опрацювання вхідних (досліджуваних) об’єктів чи явищ.

ВИСНОВКИ

У дипломній роботі магістра досліджено методи і засоби оптимізації розробки комп’ютерних систем критичного призначення. Отримані наступні основні наукові і практичні результати.

1. Проведено аналіз характеристик і сфер застосування комп’ютерних систем критичного призначення і виявлено, що процеси їх розробки базуються на класичних моделях життєвого циклу розробки, що супроводжується інертністю та не гнучкістю зміни вимог на етапах життєвого циклу і призводить зростання часу випуску системи. Окрім цього, важливим аспектом комп’ютерних систем критичного призначення є високі вимоги до їх надійності та продуктивності, що вимагає застосування технологій забезпечення та контролю властивостей протягом усього життєвого циклу.

2. Проаналізовано процеси життєвого циклу комп’ютерних систем критичного призначення та виявлено, що стандарт не регламентує способів, методів і технологій їх розробки, що дає змогу оптимізувати процес розробки шляхом інтеграції гнучких методологій у класичні методи побудови таких систем.

3. Проведено аналіз процесів моніторингу за станом розробки комп’ютерних систем критичного призначення на етапах життєвого циклу та методів аналізу вартості проекту, що дало змогу визначити шляхи оптимізації процесів і вартості їх реалізації із застосуванням agile-методологій.

4. Проведено аналіз класичних моделей життєвого циклу розробки комп’ютерних систем критичного призначення у результаті якого встановлено, що основними їх недоліками є велика часова тривалість і не здатність процесів до адаптації в умовах зміни вимог, а це в свою чергу вимагає розробки методів оптимізації для їх усунення.

5. Проведено аналіз особливостей застосування гнучких методологій розробки комп’ютерних систем, визначено основні їх переваги і недоліки, і як результат запропоновано можливі шляхи оптимізації щодо їхнього

застосування при розробці комп’ютерних систем критичного призначення на основі впровадження параметрів якості результатів виконання кожної ітерації.

6. Формалізовано процедуру організації процесів і критеріїв якості, які можна використовувати при розробці комп’ютерних систем критичного призначення на основі класичних моделей життєвого циклу та гнучких методологій, що дало змогу оптимізувати часові та якісні характеристики виконання проектів.

7. Запропоновано механізм контролю (метод) за рівнем відповідності реалізованих у комп’ютерній системі властивостей, що дає змогу застосовувати метод DSDM і методологію Scrum при реалізації комп’ютерних систем критичного призначення.

8. Обґрунтовано доцільність проведення науково-дослідної роботи шляхом розрахунку техніко-економічних показників, що дало змогу протягом приблизно двох років компенсувати витрати на використання технології планування проектів систем реального часу.

9. У результаті аналізу предметної області та особливостей запропонованого методу, визначено функціональні вимоги до програмного засобу підтримки процесу оптимізації розробки комп’ютерних систем критичного призначення та визначено ролі учасників проекту, що дало змогу врахувати їх при проектуванні архітектури даного засобу.

10. Спроектовано базу даних і побудовано архітектуру програмного засобу підтримки методу оптимізації процесу розробки комп’ютерних систем критичного призначення, що дало змогу забезпечити ефективність керування параметрами якості на різних ітераціях життєвого циклу із застосуванням agile-методологій.

11. Розроблено інтерфейси програмного засобу для забезпечення зручності виконання робіт щодо оптимізації процесів розробки комп’ютерних систем критичного призначення з врахуванням особливостей класичних моделей життєвого циклу та гнучких методологій розробки.

12. Обчислено показники економічної ефективності проведення дослідження та, на основі одержаних результатів, обгрунтовано доцільність

впровадження запропонованих методів і засобів оптимізації розробки комп’ютерних систем критичного призначення.

13. Проведено аналіз вимог нормативних документів з охорони праці користувачів комп’ютерів і визначено необхідність їх впровадження та дотримання при експлуатації програмного засобу підтримки процесу оптимізації розробки комп’ютерних систем критичного призначення.

14. Проаналізовано забезпечення надійності управління і матеріально-технічного постачання суб’єкта господарювання та принципів проведення розрахунку режимів роботи об’єкту в умовах радіоактивного забруднення при аваріях на АЕС.

15. Розглянуто питання комплексної оцінки екологічності виробництва та методів узагальнення екологічної інформації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Астелс Д., Миллер Г., Новак М. Практическое руководство по экстремальному программированию. М.: «Вильямс». 2012. 320с.
2. ДеМарко Т., Листер Т. Вальсируя с медведями: управление рисками в проектах по разработке программного обеспечения. Компания p.m. Office. М. 2015. 190 с.
3. Основные положения по разработке и применению систем сетевого планирования и управления. М. 1974. 216 с.
4. Алферов П. Роль бизнес-заказчика в ИТ-проекте. Управление проектами. №4. 2008. URL: http://www.pmmagazine.ru/document.asp?ob_no=771 (дата звернення 15.11.2019 р.).
5. Фатрелл Р., Шафер Д., Шафер Л. Управление программными проектами: достижение оптимального качества при минимуме затрат. М.: Издательский дом «Вильямс». 2013. 1136 с.
6. Брауде Э. Технология разработки программного обеспечения. СПб: Питер, 2014. 655 с.
7. Зыль С. Проектирование, разработка и анализ програмного обеспечения критического назначения. СПб: БХВ-Петербург. 2010. 336 с.
8. Соммервилл И. Инженерия программного обеспечения. Издательский дом “Вильямс”. 2015. 624 с.
9. ISO/IEC 12207:2008. Systems and software engineering – Software life cycle processes
10. ISO/IEC 33001:2015. Information technology – Process assessment
11. Кантор М. Управление программными проектами. Практическое руководство по разработке успешного программного обеспечения. М.: Вильямс. 2012. 176 с.
12. Симонов С. Технологии и инструментарий для управления рисками. Jet Info. № 2. 2003. 34 с.

13. Fenton N., Neil M. Decision Support Software for Probabilistic Risk Assessment Using Bayesian Networks. IEEE Software. 2014. № 31 (2). P. 21-26.

14. Fenton N., Neil M. Risk Assessment and Decision Analysis with Bayesian Networks. CRC Press. 2012. – P. 33-38.

15. Soumya Krishnan M. Software Development Risk Aspects and Success Frequency on Spiral and Agile Model. Int. Journal of Innovative Research in Computer and Comm. Eng. 2015. Vol. 3. P. 122-129.

16. Abdullahi M., Basri Sh., Osman Ali H. Strength and Weakness of Software Risk Assessment Tools. International Journal of Software Engineering and Its Applications. 2014. Vol. 8. № 3. P. 389-398.

17. Woody C., Ellison R. Supply-Chain Risk Management: Incorporating Security into Software Development. Software Engineering Institute Carnegie Mellon University. 2010. P. 166-178.

18. Britkin A.I. Model estimate the duration of the iterative software development process. Open education. 2009. №4. P. 75.

19. Seacord R. Secure Coding in C and C++ 2nd Edition by Addison-Wesley Professional. Part of the SEI Series in Software Engineering series Published. 2013. 256 p.

20. William R. Project Management Body of Knowledge. PMI Standard Committee. 2006. 182 p.

21. Sommerville I. Software Engineering. Addison-Wesley Publ. Company. 2011. 866 p.

22. Hijazi H. A Review of Risk Management in Different Software Development Methodologies. Int. Journal of Computer Appl. 2013. Vol. 48. № 3. P.1441-1453.

23. Сідун В., Пономарьова Ю. Економіка підприємства. К. 2013. 254 с.

24. НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації ЕОМ». Наказ Держгірпромнагляду від 26.03.2010 No 6.

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

http://sop.zp.ua/norm_npaop_0_00-7_15-18_01_ua.php (дата звернення: 10.12.2019).

26. Жидецький В.Ц. Охорона праці користувачів комп’ютерів / В.Ц. Жидецький. Львів: Афіша. 2009. 176 с.

27. Желібо Є., Заверуха Н., Зацарний В. Безпека життєдіяльності. К.: 2001. 483 с.

28. Атаманчук П.С., Мендерецький В.В., Панчук О.П., Чорна О.Г. Безпека життєдіяльності та охорона праці (Практичний курс): Навчальний посібник. Кам'янець-Подільський: "Думка". 2010. 152 с.

29. Джигирей В.С. Екологія та охорона навколишнього природного середовища. Навчальний посібник. К.: Знання. 2000. 356 с.

30. Тарасова В.В. Екологічна статистика. К.: Центр учбової літератури. 2008.-392 с.

Додаток А

Тексти наукових публікацій дипломної роботи магістра

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет імені Івана Пулюя (Україна)
Національна академія наук України
Університет імені П'єра і Марії Кюрі (Франція)
Маріборский університет (Словения)
Технічний університет у Кошице (Словаччина)
Вільнюський технічний університет ім. Гедимінаса (Литва)
Шяуляйська державна колегія (Литва)
Жешувський політехнічний університет ім. Лукасевича (Польща)
Білоруський національний технічний університет (Республіка Білорусь)
Міжнародний університет цивільної авіації (Марокко)
Національний університет біоресурсів і природокористування України (Україна)
Наукове товариство ім. Шевченка
ГО «Асоціація випускників Тернопільського національного технічного університету імені Івана Пулюя»

АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ

Збірник

тез доповідей

Том II

VIII Міжнародної науково-технічної конференції молодих учених та студентів

27-28 листопада 2019 року

УКРАЇНА
ТЕРНОПІЛЬ – 2019

ДАНИХ НА БАЗІ СИНХРОННИХ МУЛЬТИПЛЕКСОРІВ AXD155

89. **В.М. Юзьвак**
ПАТЕРНИ РОБОТИ З БАЗАМИ ДАНИХ: ООП-ПІДХІД 122

90. **В.М. Юзьвак**
АНТИПАТЕРНИ ПРИ РОЗРОБЦІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 123

91. **О.М. Яковенко, О.І. Забігайло, І.С. Ячменьов**
СТАНДАРТИ РОЗУМНОГО МІСТА 125

92. **О.М. Яковенко**
СТАНДАРТИ РОЗУМНОГО МІСТА В УКРАЇНІ 126

93. **О.П. Ясній, І.І. Голуб**
МЕТОДИ ТА ЗАСОБИ ПОБУДОВИ МЕРЕЖЕВИХ КОМУТАТОРІВ З ПІДТРИМКОЮ ТЕХНОЛОГІЙ GEPON ТА LTE 127

94. **В.В. Яцишин, Д.Я. Войтина**
ОПТИМІЗАЦІЯ ПРОЦЕСІВ РОЗРОБКИ КОМП'ЮТЕРНИХ СИСТЕМ КРИТИЧНОГО ПРИЗНАЧЕННЯ 128

95. **В.В. Яцишин, В.В. Нестор**
АЛГОРИТМ КЛАСИФІКАЦІЇ АТРИБУТІВ ЯКОСТІ КОМП'ЮТЕРНИХ СИСТЕМ 129

96. **В.В. Яцишин, Я.О. Чирський**
ОСОБЛИВОСТІ ПРОЕКТУВАННЯ ІНТЕРФЕЙСІВ ЛЮДИНО-МАШИННОЇ ВЗАЄМОДІЇ 130

97. **І.Г. П'ятківський, І.С. Ячменьов**
ВИКОРИСТАННЯ ЗАСОБІВ GOOGLE MAPS ДЛЯ ВІЗУАЛІЗАЦІЇ ДАНИХ 131

УДК 004.05
В.В. Яцишин, канд. техн. наук, доц., Д.Я. Войтина
Тернопільський національний технічний університет імені Івана Пулюя, Україна

ОПТИМІЗАЦІЯ ПРОЦЕСІВ РОЗРОБКИ КОМП'ЮТЕРНИХ СИСТЕМ КРИТИЧНОГО ПРИЗНАЧЕННЯ

V.V. Yatsyshyn PhD, Assoc. Prof., D.Y. Voityna

OPTIMIZATION OF CRITICAL SYSTEMS DEVELOPMENT PROCESSES

Сучасні тенденції розвитку технологій проектування комп'ютерних систем спрямовані на максимальне задоволення кінцевого споживача в максимально короткі терміни та з максимальним прибутком.

Хоча й спостерігається стрімкий ріст засобів, методів та методологій розробки комп'ютерних систем, однак рівень якості виконання проектів все ж залишається не дуже високим, що особливо небезпечно при реалізації комп'ютерних систем критичного призначення. До таких систем належать системи керування польотами, засоби діагностики та лікування людини, комп'ютерні системи керування атомними станціями та інші системи, які безпосередньо впливають на життя і здоров'я людей та навколишнього середовища.

Для підвищення якості та оптимізації процесу розробки комп'ютерних систем критичного призначення пропонується поєднати класичні моделі життєвого циклу і технології agile-проектування. Для цього необхідно провести формалізацію процесів та побудувати концептуальну модель виконання окремих видів діяльності на кожному з визначених етапів життєвого циклу. Концептуальну модель розробки програмного забезпечення комп'ютерних систем критичного призначення запропоновано реалізовувати як показано на рис. 1.

Рисунок 1. Модель розробки ПЗ комп'ютерних систем критичного призначення

У даному випадку розробка критичних комп'ютерних систем виконується ітераційно з можливістю інтеграції елементів agile-технологій.

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

М А Т Е Р І А Л И

VII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ »

11–12 грудня 2019 року

ТЕРНОПІЛЬ
2019

СЕКЦІЯ 3. **КОМП'ЮТЕРНІ СИСТЕМИ ТА МЕРЕЖІ**

В. Владика, Д. Величко, Г. Осухівська
ТЕХНОЛОГІЇ ПЕРЕДАВАННЯ ДАНИХ В СИСТЕМІ «ЦИФРОВА ЛІКАРНЯ» 109
В. Барбарич, Ю. Івануса
ШКІДЛИВИЙ ВПЛИВ WI-FI НА ОРГАНІЗМ ЛЮДИНИ 110
М. Бедрийчук
РЕАЛІЗАЦІЯ МЕТОДУ АДАПТИВНОЇ СЕЛЕКЦІЇ КАНАЛІВ ЗВ'ЯЗКУ ДЛЯ СИСТЕМИ «РОЗУМНИЙ БУДИНОК» 112
Д. Войтина, В. Яцишин
АНАЛІЗ ОСОБЛИВОСТЕЙ КОМП'ЮТЕРНИХ СИСТЕМ КРИТИЧНОГО ПРИЗНАЧЕННЯ 113
Р. Гаван, В. Яцишин
ОСОБЛИВОСТІ АНАЛІЗУ ТОНАЛЬНОСТІ ТЕКСТУ НА ОСНОВІ АСПЕКТІВ ПРИ ВДОСКОНАЛЕННІ КОМП'ЮТЕРНИХ СИСТЕМ 114
С. Галан, В. Яцишин
ОСОБЛИВОСТІ ПРОЕКТУВАННЯ «РОЗУМНИХ СИСТЕМ» З МОЖЛИВІСТЮ ВЗАЄМОДІЇ З ХМАРНИМИ СЕРВІСАМИ 115
Р. Гайдук, Д. Михалик
РОЗРОБКА СОЦІАЛЬНОЇ МЕРЕЖІ ДЛЯ БІБЛІОТЕК З ВИКОРИСТАННЯМ .NET ТЕХНОЛОГІЙ 116
Ю. Голояд
МЕТОДИ ТА АЛГОРИТМИ РОЗПІЗНАВАННЯ ОБРАЗІВ 117
І. Голуб, О. Ясній
МЕТОДИ ТА ЗАСОБИ ПОБУДОВИ МЕРЕЖЕВИХ КОМУТАТОРІВ ЗПІДТРИМКОЮ ТЕХНОЛОГІЙ GEPON ТА LTE 118
А. Джинджиристий, М. Паламар
АНАЛІЗ МЕТОДІВ КОЛАБОРАТИВНОЇ ФІЛЬТРАЦІЇ ПРИ ПІДБОРІ КОМАНДИ РОЗРОБНИКІВ КОМП'ЮТЕРНИХ СИСТЕМ 119
П. Євтух, В. Храпа
МЕТОДИ ТА ЗАСОБИ УЗГОДЖЕННЯ ЦИФРОВИХ ПОТОКІВ ПРИ ЗАСТОСУВАННІ ІОТ ДО МЕДИЧНИХ ПРИЛАДІВ 120
А. Жуйвода
МЕТОД ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ПЕРЕДАЧІ ГОЛОСОВИХ СИГНАЛІВ В КОМП'ЮТЕРНИХ МЕРЕЖАХ 121
А. Жуйвода
ПІДВИЩЕННЯ ЯКОСТІ ПЕРЕДАЧІ ПОТОКОВОГО ТРАФІКУ 122
О. Зимницький
ВРАЗЛИВОСТІ РЕАЛІЗАЦІЇ КРИПТОГРАФІЧНИХ МЕТОДІВ ЗАХИСТУ ПРОТОКОЛУ SSL/TLS 123
Б. Калиниченко, І. Грод
ДОСЛІДЖЕННЯ ВРАЗЛИВОСТЕЙ МЕРЕЖІ ОФІСУ "ZoomSupport" ТА МЕТОДІВ ЇХ УСУНЕННЯ 124
В. Ковальов, С. Лупенко
МЕТОДИ ТА ЗАСОБИ ПОБУДОВИ КОМП'ЮТЕРИЗОВАНИХ ДІАЛОГОВИХ СИСТЕМ ТОРГОВОГО ЦЕНТРУ 125
І. Купратий
НЕЙРОМЕРЕЖІ У СИСТЕМАХ БІОМЕТРИЧНОЇ АУТЕНТИФІКАЦІЇ ЗА КЛАВІАТУРНИМ ПОЧЕРКОМ 126
О. Ліщук, Є. Тиш
ПЕРЕВАГИ ВИКОРИСТАННЯ ТЕХНОЛОГІЇ АГРЕГАЦІЇ КАНАЛІВ КОМП'ЮТЕРНОЇ МЕРЕЖІ 127

УДК 004.05
Д. Войтина, В. Яцишин
Тернопільський національний технічний університет імені Івана Пулюя

АНАЛІЗ ОСОБЛИВОСТЕЙ КОМП'ЮТЕРНИХ СИСТЕМ КРИТИЧНОГО ПРИЗНАЧЕННЯ

UDC 004.05
D. Voityna, V. Yatsyshyn
(Ternopil Ivan Puluj National Technical University, Ukraine)

ANALYSIS OF FEATURES OF CRITICAL COMPUTER SYSTEMS

Сучасні методи і технології швидкої розробки компонентів комп'ютерних систем задовольняють вимоги щодо часу реалізації проекту і забезпечення необхідної функціональності системи. Однак рівень якості та надійності кінцевих продуктів все ж потребує вдосконалення, особливо при реалізації комп'ютерних систем критичного призначення.

Характерною особливістю критичних об'єктів, в тому числі і комп'ютерних систем, є те, що у випадках збоїв чи відмов їх роботи, які супроводжуються порушенням або втратою управління над системою, можуть призводити до значних негативних впливів на навколишнє середовище, життя і здоров'я людей або інших наслідків. Комп'ютерні системи, які застосовуються у військовій сфері, на потенційно небезпечних хімічних підприємствах, атомних станціях, на транспорті та в інших сферах, що можуть провокувати негативний вплив на біологічні об'єкти чи приводити до значних збитків належать до критичних комп'ютерних систем.

Сьогодні спостерігається широке використання та розповсюдження комп'ютерних систем критичного призначення у різних сферах діяльності, що супроводжується використанням апаратних і програмних компонентів, які забезпечують функції запобігання та захисту від виникнення небезпечних ситуацій і дають змогу уникнути або зменшити негативний вплив на навколишнє середовище.

Розробники комп'ютерних систем для забезпечення здатності до еволюціонування таких систем повинні використовувати методи, технології та інструментальні засоби у процесах проектування та супроводу, що давали б змогу ефективно керувати, проводити моніторинг як функціональних, так і нефункціональних критеріїв, особливо щодо гарантоздатності і продуктивності.

Необхідність та важливість критичних комп'ютерних систем до еволюції зумовлено рядом факторів, які стосуються:

- розповсюдження та широкого використання комп'ютерних систем – комп'ютерні системи масово увійшли у побут людей, а збої та відмови у їх функціонуванні безпосередньо впливають на комфорт або життя суспільства;
- комп'ютерні системи і їх складові все більше стають критичними;
- людський фактор, що може зумовлювати негативний вплив на ефективність використання комп'ютерних систем критичного призначення і вимагає застосування все більш складних механізмів людино-машинної взаємодії;
- структурна складність компонентів комп'ютерних систем доволі часто полягає у використанні стороннього зовнішнього впливу та інтеграції з іншими системами, що може негативно позначатись на надійності вузлів і мати критичний вплив на користувачів;
- висока швидкість реакції на ринку комп'ютерних систем вимагає застосування технологій і засобів швидкого внесення змін за короткий проміжок часу та з мінімальними витратами.