

Центральноукраїнський національний технічний університет  
Механіко-технологічний факультет  
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”  
Завідувач кафедри кібербезпеки  
та програмного забезпечення  
д.т.н., професор  
\_\_\_\_\_ Олексій СМІРНОВ  
“ \_\_\_\_ ” \_\_\_\_\_ 2024 р.

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА**  
**за другим (магістерським) рівнем вищої освіти**  
на тему  
**“Дослідження та програмна реалізація системи планування  
модернізації ЦОД”**

КБПЗ - 2024

Виконав здобувач вищої освіти  
II курсу, групи КІ-23М  
ОПП «Комп’ютерна інженерія»  
спеціальності 123 «Комп’ютерна інженерія»  
\_\_\_\_\_ Найдюнов Є.В.  
« \_\_\_\_ » \_\_\_\_\_ 2024 р.

Керівник проекту  
доктор філософії (PhD)  
\_\_\_\_\_ Усік П.С.  
« \_\_\_\_ » \_\_\_\_\_ 2024 р.  
Рецензент \_\_\_\_\_  
\_\_\_\_\_

Центральноукраїнський національний технічний університет  
Факультет Механіко-технологічний  
Кафедра Кібербезпеки та програмного забезпечення  
Рівень вищої освіти магістр  
Галузь знань 12 "Інформаційні технології"  
Спеціальність 123 "Комп'ютерна інженерія"  
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерна інженерія"

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2024 року

## ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Найдьонову Євгену Віталійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи  
планування модернізації ЦОД

2. Керівник роботи Усік Павло Сергійович, доктор філософії (PhD)

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

затверджені наказом вищого навчального закладу № 19-13 від 07.08.2024 року

3. Строк подання студентом роботи до захисту 2.12.2024 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою розробки є  
дослідження та програмна реалізація системи планування модернізації ЦОД

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно  
розробити)

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Маркетингове та економічне  
обґрунтування IT-проєкту.

3. Опис і обґрунтування проектних рішень.

8. Заходи з охорони праці та техніки  
безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову  
експлуатацію

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Наукова новизна

1 аркуш

Структурна схема системи

1 аркуш

Функціональна схема системи

1 аркуш

Діаграма процесів

1 аркуш

Блок-схема алгоритму роботи додатку

2 аркуша

Показники економічної ефективності

1 аркуш

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Доренська А.О.	05.10.2024	14.11.2024
Охорона праці	Марченко К.М., к.т.н., доцент	06.10.2024	16.11.2024

7. Дата видачі завдання « 6 » вересня 2024 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2024 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2024 р.	
3.	Розробка моделі компонента	20.10.2024 р.	
4.	Розробка структур даних	25.10.2024 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2024 р.	
6.	Програмування алгоритмів	10.11.2024 р.	
7.	Розрахунок економічної ефективності	13.11.2024 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2024 р.	
9.	Оформлення ПЗ	17.11.2024 р.	
10.	Попередній захист роботи	2.12.2024 р.	

Дата видачі завдання  
« 6 » вересня 2024 р.

Підпис керівника

\_\_\_\_\_  
(прізвище та ініціали)Завдання прийнято до виконання  
« 6 » вересня 2024 р.

Підпис здобувача

\_\_\_\_\_  
(прізвище та ініціали)

## АНОТАЦІЯ

**Найдьонов Є.В. Дослідження та програмна реалізація системи планування модернізації ЦОД. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2024.**

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи планування модернізації ЦОД.

Метою розробки є дослідження та програмна реалізація системи планування модернізації ЦОД.

Об'єктом дослідження є процес планування модернізації ЦОД.

Предметом дослідження є методи планування модернізації ЦОД.

Методи дослідження базуються на методах комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи планування модернізації ЦОД.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Python.

**Ключові слова:** комп'ютерна інженерія, ЦОД

## ABSTRACT

**Naidyonov E.V. Research and software implementation of the data center modernization planning system. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.**

In this final qualification work for the second (master's) level of higher education, software has been developed, which is intended for the data center modernization planning system.

The purpose of the development is the research and software implementation of the data center modernization planning system.

The object of the research is the data center modernization planning process.

The subject of the research is the methods of data center modernization planning.

The research methods are based on computer network methods, mathematical statistics methods, and software development methods.

The result of the work is the software implementation of the data center modernization planning system.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software are provided.

The program can be used on PCs with Windows 10/11.

The program is developed in the Python environment.

**Keywords:** computer engineering, data center

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ .....	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ .....	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	11
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....	14
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	14
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	43
2.3 Розгорнута постановка завдання .....	45
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....	47
3.1 Опис функціонування системи .....	47
3.2 Розробка структурної схеми.....	48
3.3 Розробка функціональної схеми .....	53
3.4 Розробка діаграми процесів.....	65
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	67
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	67
4.2 Захист розробленого програмного забезпечення.....	83
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....	86
6 НАУКОВА НОВИЗНА .....	92

					ВКРМ-123.24.0027.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	Дослідження та програмна реалізація системи планування модернізації ЦОД	Літ.	Аркуш	Арквівіс
Розроб.	Найдюнов Є.В.					М	1	128
Перев.	Усік П.С.							
Н.контр.	Коваленко А.С.					ЦНТУ КІ-23М		
Затв.	Смірнов О.А.							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ .....	93
7.1	Визначення цільової аудиторії кінцевого готового продукту .....	93
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	94
7.3	Вибір методу оцінки вартості ПЗ .....	96
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	97
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ .....	99
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ .....	99
7.7	Визначення ключових факторів успіху конкретного проєкту.....	101
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ .....	102
8.1	Вступ.....	102
8.2	Шкідливі і небезпечні фактори при роботі з комп'ютером.....	103
8.3	Аналіз санітарно-гігієнічних умов праці на робочому місці програміста .	105
8.4	Розробка заходів з умов поліпшення охорони праці .....	108
8.5	Розрахункова частина .....	109
9	ОСНОВНІ ВИСНОВКИ.....	111
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	112

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ДЦ	–	дата центр
ІБЖ	–	індивідуальний блок живлення
ІТ	–	інформаційні технології
ПЗ	–	програмне забезпечення
СБГЕ	–	система безперебійного гарантованого електропостачання
СКС	–	структуровані кабельні системи
СКУД	–	системи контролю й управління доступом
ЦОД	–	центр обробки даних
ІоТ	–	Internet of Things
QoS	–	механізми контролю й керування якістю

КБПЗ – 2024

					ВКРМ-123.24.0027.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

## ВСТУП

**Актуальність теми.** У якийсь момент центрам обробки даних, особливо дуже великим або приналежним провайдерам хмарних сервісів, будуть потрібні істотно більша щільність розміщення встаткування й значні – до 400 Гбіт/с – швидкості передачі даних. У ЦОДах потрібно підтримувати все більші швидкості передачі даних у рамках заданої структури витрат. Як результат, оператори ЦОДів зіштовхуються з безліччю проблем при підготовці до міграції на встаткування наступного покоління для задоволення зростаючих потреб слідом за стрімким поширенням смартфонів і інших мобільних пристроїв, що генерують величезні обсяги трафіку даних. Переважні сьогодні 10-гігабітні інтерфейси Ethernet замінюються портами 40 і 100 Гбіт/с. Більше того, IEEE уже ініціювала розробку стандарту Ethernet на 400 Гбіт/с. Як операторам ЦОДів забезпечити такі швидкості передачі даних без повної заміни встаткування? Для підтримки більших швидкостей передачі даних і більше високої щільності портів устаткування наступного покоління повинне відповідати твердим вимогам до бюджетів оптичних втрат і володіти більшою енергоефективністю. Чи зможе ЦОД відповідати все зростаючим вимогам без зміни існуючої архітектури? Для спрощення розгортання нового обладнання в майбутньому необхідно вже сьогодні впроваджувати більше гнучку й менш складну архітектуру. Знаходження найкращого рішення неможливо без розуміння прийдешніх викликів і згоди на певні компроміси. Кожний сервер, комутатор, маршрутизатор і систему зберігання знадобиться, цілком можливо, розглядати окремо. Прийняття інформованих рішень і розуміння архітектурних особливостей забезпечать більш швидку, просту й економічну міграцію на встаткування наступного покоління.

**Мета й завдання дослідження.** Метою роботи є дослідження та програмна реалізація системи планування модернізації ЦОД.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем планування модернізації ЦОД.
- Дослідження системи планування модернізації ЦОД.
- Програмна реалізація системи планування модернізації ЦОД.

*Об'єктом дослідження є процес планування модернізації ЦОД.*

*Предметом дослідження є методи планування модернізації ЦОД.*

*Методи дослідження базуються на методах комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.*

**Наукова новизна отриманих результатів.** У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- Удосконалено метод планування модернізації ЦОД.
- Розроблено вітчизняний продукт планування модернізації ЦОД, який має більш широкі можливості, на відміну від існуючих аналогів.

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі планування модернізації ЦОД.

**Достовірність наукових результатів** підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти LV науково-технічної конференції «Наука в ЦНТУ: основні досягнення та перспективи розвитку» (2024 р.), основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №15.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи планування модернізації ЦОД, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

У цей час, інтерес до питань побудови ефективних Центрив Обробки Даних (ЦОД) зріс в усьому світі.

Центри обробки даних (ЦОД) – це відказостійка комплексна централізована система, що забезпечує автоматизацію бізнес-процесів з високим рівнем продуктивності і якістю надаваних сервісів.

Потік інформації, що пропускається корпоративними обчислювальними мережами, постійно збільшується, а її цінність підвищується рік у рік. Поряд із цим, відбувається швидкий розвиток різних інноваційних видів зв'язку: електронної й голосової пошти, IP телефонії, сервісів миттєвого обміну повідомленнями, відеоконференцій і т.п. Якщо у Вашій компанії постійно росте обсяг інформації, яку необхідно обробляти й аналізувати, швидко збільшується кількість використовуваних бізнес-додатків – виникає необхідність централізувати обробку даних у ЦОД. Ці й деякі інші фактори приводять до потреби сучасного бізнесу в збільшенні функціонального числа ЦОД. У цей час у комерційному середовищі поряд з визначенням Центр обробки даних (ЦОД) використовується термін – Дата Центр (ДЦ).

### Концепція побудови інженерних систем ЦОД

ЦОД являє собою технологічне приміщення, і залежно від спрямованості організації, що обслуговується, містить різне «активне» комплектувальне устаткування.

Корпоративні ЦОД уміщають системи зберігання й обробки даних, а так само мережне й телекомунікаційне встаткування.

У ЦОД операторів зв'язку й сервіс – провайдерів розташовуються засоби зв'язку й інші пристрої, необхідні для надання абонентських послуг.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

Необхідність підтримки сприятливого режиму функціонування високотехнологічного сучасного «активного» устаткування ставить перед творцями інженерних рішень проблему організації систем життєзабезпечення ЦОД.

Гарантом загальної працездатності ЦОД, від якої залежить функціонування всієї інформаційної системи в цілому, є правильно організована інженерна інфраструктура.

Сучасний Центр являє собою інтелектуальний будинок у мініатюрі. Всі інженерні системи ЦОД інтегровані в єдиний комплекс, з умовою неодмінного обліку можливих майбутніх змін.

Відновлення телекомунікаційного встаткування відбувається значно швидше змін інженерної інфраструктури.

Основні завдання, розв'язувані встановленими в ЦОД інженерними системами, можна розділити на три групи:

- забезпечення функціонування технологічного встаткування: системи електропостачання, вентиляції й кондиціонуванні, Структуровані Кабельні Системи (СКС).

- захист від технічних збоїв: системи автоматичного оповіщення й гасіння пожежі, система автоматизації й диспетчеризації.

- захист від несанкціонованих дій людини: охоронна сигналізація, відеоспостереження, контроль доступу.

У зв'язку з підвищеним навантаженням, що доводиться на ЦОД, всі комплектуючі і використовувані при обробці й будівництві матеріали повинні проходити обов'язкове тестування на відповідність міжнародним стандартам.

Особливі вимоги пред'являються до кабельної системи ЦОД. Внаслідок специфічних умов експлуатації, СКС, використовувані при будівництві дата центра, трохи відрізняються від звичайних офісних кабельних систем.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

Пасивне телекомунікаційне встаткування, у якому розміщуються сервери, системи зберігання даних і мережних пристроїв повинні бути надійними й зручними для обслуговування пристроїв, що втримуються в них.

### **Кліматична система ЦОД**

Сучасне високотехнологічне обчислювальне й телекомунікаційне встаткування відчутно до самих незначних змін навколишнього середовища. Обов'язковою умовою забезпечення його нормальної працездатності є підтримка строга певних температурних режимів і рівня вологості, тому ЦОД необхідна система кондиціонування й вентиляції.

Ріст обчислювальних потужностей ЦОД і серверних кімнат приводить до збільшення енергетичної щільності встаткування. У результаті, утворюються ділянки перегріву, які можуть стати причиною аварійної зупинки системи.

У загальному випадку, кліматична система ЦОД складається із трьох підсистем:

- система основного кондиціонування, що забезпечує прецизійне регулювання температури й вологості повітря, тим самим підтримуючі параметри навколишнього середовища, прийнятні для нормального функціонування ІТ устаткування;
- система кондиціонування, що встановлює комфортні умови роботи для обслуговуючого ЦОД персоналу;
- система обмінної вентиляції створює надлишковий тиск і постачає ЦОД припливом свіжого повітря.

Для кондиціонування шаф технологічного встаткування зв'язку застосовується система зворотного охолодження, з установкою в комунікаційних шафах повітряно-водяних теплообмінників настінного виконання, і установкою центрального чиллера системи зворотного охолодження.

Можлива реалізація систем комфортного й основного кондиціонування в єдиному комплексі.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

## **Система електропостачання**

Комп'ютерне й телекомунікаційне встаткування, установлене в ЦОД надзвичайно вимогливо до якості електроживлення, що має на увазі установку систем безперебійного гарантованого електропостачання (СБГЕ).

Комплексні рішення для енергопостачання ЦОД містять у собі:

- системи загального електропостачання – внутрішні й зовнішні мережі;
- системи безперебійного електропостачання;
- системи гарантованого електропостачання – дизель-генераторні електростанції;
- підсистеми моніторингу й управління електропостачанням об'єкта, об'єднані в єдину централізовану систему контролю й управління підприємством.

## **Структурована кабельна система**

СКС є основою мережної архітектури ЦОД і повинна відповідати таким вимогам, як надійність, універсальність, масштабованість, гнучкість архітектури й відповідність стандартам.

Пасивне телекомунікаційне встаткування, у якому розміщаються сервери, системи зберігання даних і мережних пристроїв повинні бути надійними й зручними для обслуговування пристроїв, що втримуються в них.

## **Система моніторингу**

Не меншу важливість для забезпечення функцій ЦОД мають системи автоматичного контролю й моніторингу, відповідальні за забезпечення безпеки й управління інженерними системами.

Система автоматизації й диспетчеризації призначена для управління інженерними системами об'єкта й створення комфортних умов для персоналу.

Установка даної системи необхідна в ЦОД для автоматичної підтримки оптимальних режимів роботи встаткування й оперативного прогнозування його виходу з ладу.

Система автоматизації й диспетчеризації дозволяє здійснювати моніторинг стану інженерних систем у режимі реального часу, оперативне

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

управління встаткуванням, розмежування доступу до інформації. При цьому всі функції по управлінню інженерними системами доступні з єдиного користувальницького інтерфейсу.

Система моніторингу робить контроль параметрів:

- Електропостачання центра обробки даних (ЦОД).
- Кліматичної системи центра обробки даних (ЦОД).
- Автоматичного пожежогасіння центра обробки даних (ЦОД).
- Охоронної й пожежної сигналізації центра обробки даних (ЦОД).
- Системи контролю доступу центра обробки даних (ЦОД).

### **Системи безпеки**

Системи безпеки для ЦОД являють собою комплекс технічних засобів, що складає з устаткування систем автоматичного пожежогасіння й оповіщення про пожежу, контролю й управління доступом, охоронної сигналізації, відеоспостереження й ряду інших підсистем.

Контроль роботи систем безпеки і їхніх вузлів здійснюється автоматично.

### **Система контролю й управління доступом**

Застосування на об'єкті системи контролю й управління доступом (СКУД) повністю виключає вплив людського фактора на пропускний режим.

Система контролює дату й час проходження співробітників у приміщення, ідентифікацію співробітника, переміщення співробітників по охоронюваному об'єкті, дозволяє розмежувати доступ у приміщення.

Система контролю може бути інтегрована з іншими системами безпеки. Грамотна інтеграція СКУД із системою відеоспостереження, охоронною сигналізацією, системою пожежної сигналізації дозволяє повністю контролювати ситуацію на об'єкті.

Всі входи й виходи в приміщення оснащуються безконтактними зчитувачами магнітних карт, що дозволяє вести статистику.

Передбачається автоматичне розблокування дверей при пожежі.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Для розблокування дверей в аварійних ситуаціях передбачається установка в малодоступному місці роз'єднувача на ланцюг живлення замків.

Електропостачання встаткування системи контролю й управління доступом забезпечується по вищій категорії надійності.

### **Система відеоспостереження**

Візуальний контроль над обстановкою по периметрі об'єкта й у приміщеннях ЦОД забезпечують камери відеоспостереження.

Системи безпеки й відеоспостереження необхідні для відстеження поточної ситуації, попередження про небезпеки й прийняття необхідних і ефективних заходів при аваріях і пожежах, а також для контролю доступу в будинки й приміщення.

## **1.2 Область застосування**

ЦОДи в цей час є невід'ємною частиною нашого повсякденного життя, будь те бізнес або широка громадськість. З розвитком Інтернету речей (The Internet of Things (IoT)), важливість ЦОДів буде тільки рости, а їхнє число збільшуватися в усьому світі. Однак виробництво й управління розкривають безліч проблем, зв'язаних зі специфікацією дата-центрів. І справа не тільки в тому, щоб брати на себе зобов'язання по дорогому проєкті, але й у рішенні таких складних завдань, як, наприклад, забезпечення функціонування ЦОДа без яких-небудь непередбачених збоїв.

Кожний ЦОД індивідуальний і має свої унікальні проблеми. Те, що добре працює в одному місці, може не підійти в іншому. У результаті, інженери ЦОДів повинні постійно адаптуватися й управляти наявними ресурсами для рішення питань специфікації. Однак існують деякі проблеми, які є типовими для всіх випадків. Хоч і немає єдиного рішення, ми можемо розібратися в типових проблемах специфікацій дата-центрів.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

## Відмова батареї ІБЖ

Найпоширенішими причинами аварій є погане планування й відмова батареї ІБЖ. У сполученні із завищенням потужності ІБЖ це може привести до величезних витрат через простій.

Цього можна уникнути шляхом планування й розумного інвестування в ІБЖ, – це допоможе впоратися з очікуваним навантаженням.

Крім того, дуже важливо регулярно перевіряти батареї й підтримувати їх у робочому стані.

## Недоліки планування інфраструктури

При об'єднанні старих і нових серверів важливо відзначити, що нові сервери (наприклад, blade-сервери) вимагають набагато більше енергії, чим їхні старші версії.

Нові сервери можуть споживати майже в п'ять разів більше енергії, чим старі. Це, у свою чергу, збільшує потребу в охолодженні усередині всієї апаратної, так що це також необхідно враховувати при додаванні нового обладнання.

## Влучення води в пристрій

Може здатися очевидним, що волога не повинна стикатися з електронним устаткуванням, таким як сервери, але все-таки подібне трапляється й викликає серйозні збої. Причини влучення вологи наступні:

- негерметичні клапани;
- газувана вода й кава, які техніки принесли в ЦОД.

Є просте рішення цієї проблеми: контролювати клапани й гарантувати, що всі напої здаються перед входом в апаратну.

## Недостача даних у реальному часі

Найчастіше, коли справа доходить до специфікації ЦОДів, операторами упускається створення системи звітності в режимі реального часу.

Просте читання керівництва, що фахівці з експлуатації перегортали кілька тижнів назад, не допоможе прийняти важливі рішення з таким складним

					ВКРМ-123.24.0027.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

об'єктом, якимось є дата-центр колокейшен. Єдиний спосіб оптимізації ресурсів і силових навантажень – це надійні сінохвилинні показання.

### **Неналежне планування обробки запитів**

Коли справа доходить до дата-центрів, трапляються піки й спади активності, залежно від типів клієнтів. Якщо ви надаєте послуги для фінансового сектора, протягом робочого дня активність буде висока й мінімальна вночі.

А от у сезон відпусток всі інакше, спостерігається значне підвищення активності з боку приватних клієнтів.

Таким чином, оператори ЦОДів повинні планувати й бути готовими до коливань активності.

Таким чином, виходячи з вищеперахованого, дослідження та програмна реалізація системи планування модернізації ЦОД, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ – 2024

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

## 2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

### 2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

#### Комплексність як професійний підхід до створення ЦОДа від Huawei

Сьогодні на ринку інфраструктурних рішень для дата-центрів представлено трохи гравців, продуктовий портфель яких охоплює весь комплекс інженерних рішень і експертизи. Серед них – компанія Huawei, у складі якої є власний інженерний підрозділ, що спеціалізується на розробці комплексних рішень для ЦОДа.

Особливості й переваги подібних рішень як для замовника, так і для постачальника очевидні.

Перша особливість – це обсяг поставки. Якщо, приміром, компанія, що займається виробництвом ІБЖ, може поставити тільки 20-30% усього встаткування, необхідного для створення інженерної інфраструктури ЦОДа, то комплексний постачальник у стані охопити всі 100%. У результаті замовник одержує більше низьку вартість у порівнянні із ціною встаткування розрізнених постачальників.

Для комплексного рішення будуть краще й умови поставки. Що ж стосується самого виробника встаткування, те комплексна поставка значно збільшує його дохід з одного об'єкта.

Друга особливість – глибина пророблення інженерної інфраструктури ЦОДа. Комплексний постачальник, що має власний інженерний підрозділ, здатний детально проробити проект і ідеально підібрати встаткування ще на етапі ескізу. Це значить, що на наступних етапах проектувальник зможе простіше й істотно швидше розробити проекти різних стадій.

					ВКРМ-123.24.0027.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

Компанія Huawei пропонує ринку свій підхід до комплексності, модульності й масштабованості.

Комплексний підхід Huawei містить у собі чотири рівні, кожний з яких представлений глибоко проробленим рішенням. Компанія пропонує чотири рішення, які закривають собою весь комплекс питань по створенню інженерної інфраструктури дата-центрів.

Перший рівень, L0 – це будівельна підготовка ЦОДа. На даному етапі проводяться всі роботи, які можна віднести до загальнобудівельних і оздоблювальних. Як відомо, будівництво великих ЦОДів у будинках, не підготовлених для їхнього розміщення, – дуже складне завдання. Тому не слід недооцінювати важливість цього рівня: вірніше всього буде віддати управління будівництвом тим, хто займається створенням всіх інших систем дати-центра. Не говорячи вже про те, що компанія Huawei має необхідний експертний рівень для рішення подібних завдань.

Другий рівень, L1 – інженерні системи ЦОДа. Для даного рівня Huawei пропонує широкий вибір рішень по встаткуванню для інженерних систем ЦОДа, але на них ми зупинимося трохи пізніше.

Третій рівень, L2 – ІТ-платформа ЦОДа. На цьому рівні в лінійці продукції Huawei для ЦОДів є істотний перелік устаткування:

- сервери (стандартні стійкові сервери серії RH, високопродуктивні сервери серії X, блейд-сервери серії E);
- системи зберігання дані серії OceanStore;
- комутатори серії CloudEngine;
- системи захисту, брандмауер серії USG;
- комплексне рішення, що містить у собі й сервери, і системи зберігання, і комутатори, і системи захисту, – Fusion Cube.

І, нарешті, четвертий рівень, L3 – рівень операційних систем і програмного забезпечення. Тут Huawei пропонує наступні рішення:

- хмарну операційну систему – FusionSphere;

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

- систему управління ресурсами ЦОДа – ManageOne;
- систему управління послугами ЦОДа – CSB;
- рішення розподіленого хмарного ЦОДа – DC2.

Зупинимося більш докладно на встаткуванні для інженерних систем дати-центра. У лінійці Huawei представлені комплексні рішення по інженерній інфраструктурі як для зовнішнього розміщення, так і для внутрішнього.

Для зовнішнього розміщення розроблене рішення IDS1000 – контейнерний ЦОД. Це гнучке модульне рішення поставляється у двох варіантах виконання – «усе в одному» або у вигляді кластера. Воно допоможе реалізувати різний фізичний, сервісні й бізнес-завдання, які не можуть бути виконані в традиційних дата-центрах, розгорнутих у капітальних будинках і будовах. IDS1000 досить швидко (істотно швидше, ніж будівництво нового будинку!) може бути розгорнуте на вуличній площадці, тому дане рішення особливо затребуване при розгортанні ЦОДів операторами зв'язку у віддалених регіонах або при розробці корисних копалин, у зоні стихійних лих і воєнних дій, а також при необхідності швидко реалізувати новий корпоративний ЦОД на площадці, що прилягає до офісу.

Виконання «усе в одному», або IDS 1000-A, являє собою один стандартний морський контейнер, наповнений стійками для ІТ-устаткування й всією інженерною технікою, необхідної для функціонування зони ІТ. Крім стійок для розміщення встаткування комплекс інженерної техніки усередині контейнера містить у собі модульні джерела безперебійного живлення, фреонові межрядні кондиціонери, системи розподілу живлення, кабельні траси, системи протипожежного захисту й т.

Кластерне виконання контейнерного ЦОДа за назвою IDS 1000-C являє собою групу контейнерів, кожний з яких виконує своє безпосереднє завдання. У такій групі контейнерів, що становлять повноцінний ЦОД, присутні ІТ-модуль, модуль системи охолодження, модуль системи електропостачання й диспетчерський модуль. У даному виконанні є ті ж компоненти, що й в IDS 1000-

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

А, за єдиним виключенням: використовується система охолодження на базі моноблочних холодильних машин.

Для внутрішнього розміщення розроблене модульне рішення IDS2000. Воно є так же гнучким, як і IDS1000, і може бути представлено у вигляді трьох виконань різного розміру: малого, середнього й великого. Застосовується IDS2000 у тих випадках, коли в замовника вже їсти де розмістити встаткування або споконвічно передбачене зведення спеціальних приміщень, споруджень і будинків для ЦОДа. Це рішення дозволяє реалізувати різний фізичний, сервісний й бізнес-завдання різного масштабу, етапності й гнучкості. Воно має більший потенціал при будівництві дата-центрів як у вже існуючих спорудженнях (наприклад, при будівництві ЦОДів у колишніх цехах, на складах або автостоянках), так і при масштабному новому будівництві.

Модульний ЦОД IDS2000 великого розміру, або IDS 2000-L, використовується в більших будинках зі спеціальними технологічними приміщеннями, де системи кондиціонування, електропостачання та інше інженерне встаткування спроектоване й розміщене окремо друг від друга. Дане рішення підходить для установки в ЦОДах великих підприємств.

ЦОД середнього розміру, IDS 2000-M , використовується в тих випадках, коли передбачене розміщення систем життєзабезпечення ЦОДа безпосередньо в серверних приміщеннях. Таке рішення підходить для розміщення в різних філіях і представництвах великих компаній або в організаціях і на підприємствах середнього розміру.

Міні-цод, або IDS 2000-S , являє собою дві стійки, у яких безпосередньо розміщене інженерне встаткування. Таке рішення призначене для офісів і серверних кімнат невеликих компаній.

У рішення IDS2000, так само як і в IDS1000, входить все встаткування, необхідне для життєзабезпечення ЦОДа, як то: системи кондиціонування, джерела безперебійного живлення, розподільні пристрої, система ізоляції коридорів,

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

система пожежогасіння й т.п. Все це встаткування може поставлятися як комплексно, так і окремо.

Завдяки такій гнучкості рішення компанії Huawei дозволяють оптимально реалізувати самі різні вимоги замовника й сценарії побудови ЦОДів. Крім того, Huawei намагається відповідати високим вимогам якості й тому регулярно модернізує існуюче встаткування й розробляє нові рішення.

### **Оцінка дата-центра для одержання сертифіката на проект Uptime Institute**

Якщо занадто багато дата-центрів, що заявляють про свою відповідність рівням TierIII або TierIV, на ділі не зможуть забезпечити безперебійну роботу, особливо такі, які надають хостинг високопрофільним хмарним сервісам, що зачіпають широке коло користувачів, то поняття tier втратить зміст для операторів дата-центрів і їхніх клієнтів.

Існують вагомні причини, по яких проектувальники дата-центрів не вправі самостійно оцінювати відповідність свого проекту параметрам сертифікації Uptime, TIA, LEED і інших систем.

Наприклад, система рівнів сертифікації Tier, розроблена Uptime Institute, описує, що у вас є, а чого немає в плані резервування систем дата-центра, а також оцінює ризик і тривалість можливого простою. Однак поняття «рівні Tier» різні проектувальники інтерпретують по-різному. Так звана «самоатестація» може привласнити об'єкту якийсь рівень, але реальна експлуатаційна історія не підтвердить його. Достовірна сертифікація дата-центра по проекті може бути проведена тільки силами Uptime Institute, і їй передує ретельна перевірка.

Сертифікація по проекті працює так само, як і золота, срібна й платинова сертифікація будинків системи LEED. Проектувальники можуть порахувати бали, які, на їхню думку, набирає їхній проект, але остаточне твердження й фактична сертифікація LEED може виходити тільки від Американської Ради по Зелених Будинках (U.S. Green Building Council).

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

Uptime Institute привласнює рівень Tier, керуючись станом самої слабкої ланки інфраструктури. Тому навіть якщо в дата-центрі встановлені системи резервування енергопостачання й охолодження, що відповідають рівню Tier IV (дублюючі джерела безперебійного живлення, чиллери, розподільні щитки, блоки обробки повітря серверного залу та інше), а також дублюючі електричні розподільні ланцюги рівня Tier IV, такий об'єкт однаково одержить сертифікат рівня лише Tier II, якщо, наприклад, у ньому є тільки один контур трубопроводів.

Багато проектувальників незадоволені таким строгим індивідуальним підходом. Сам проектувальник оцінив би одноконтурну систему як цілком надійну для критично важливих робочих навантажень дата-центра, другий контур адже насправді не дуже й потрібний, щоб досягти бажаного рівня надійності. Так що, на думку проектувальника, такий дата-центр заслуговує сертифіката рівня Tier IV.

Дизайнери ЦОДів часто йдуть на так званий інформований компроміс відносно резервування, залежно від бюджету, обмежених площ, або ґрунтуючись на статистику відмов з минулого досвіду. Наприклад, статистика відмов системи охолодження в місцевості, що славиться своєю стійкою системою енергопостачання, може подвигнути оператора суворіше підійти до резервування охолодження, але заощадити на аналогічному резервуванні системи живлення. На підставі цього проектувальник буде вимагати сертифіката рівня Tier III Plus або Near Tier IV, але Uptime Institute такі вимоги категорично не визнає. Будинок у цілому не може бути визнане краще, ніж його сама слабка зона. Сертифікація Uptime бере до уваги тільки те, що в реальності необхідно для досягнення проектних показників надійності і яких модифікацій можливі в межах певного рівня Tier.

### **ТІА проти UptimeInstitute**

Щоб потіснити Uptime, у свій час придумувалися різні системи сертифікації дата-центрів по проекті.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

Асоціація телекомунікаційних компаній (Telecommunications Industries Association – ТІА) розробила стандарт 942-А, до якого додається інформація про параметри рівнів, але споконвічно він був розроблений для кабельної системи в ЦОДі. Додаток F до цього стандарту ґрунтується на базовій термінології Uptime Institute. Щоб не заплутатися, який же сертифікат отриманий тим або іншим ЦОДом, рівні Uptime позначаються римськими цифрами (Tier I, II, III і IV), а ТІА – арабськими (Tier 1, 2, 3 і 4).

Стандарт ТІА є скоріше приписанням, і в цьому його відмінність від сертифікації Uptime. Додаток до стандарту ТІА 942-А включає таблиці з параметрами, яким повинен відповідати дата-центр для одержання певного рівня по архітектурі, електриці, інженерії (охолодженню) і телекомунікаціям (кабелі). На відміну від Uptime Institute, ТІА дозволяє класифікувати кожен групу в загальному проекті ЦОДа окремо. Дата-Центр може мати архітектуру рівня Tier 1, систему електроживлення й інженерію рівня Tier 2, а телекомунікаційну інфраструктуру рівня Tier 4. Такий підхід полегшує проектувальникові завдання відповідності різномірним критеріям і дозволяє окремо підійти до класифікації стійкості кожної зони інфраструктури.

При цьому Uptime Institute вважає часткові класифікації по стандарті ТІА-942-А неправомірними й не мають ніякого значення в оцінці довгострокової надійності й перспектив відмови.

### **Що потрібно для сертифікації по проекту**

Розповсюджена помилка, що для відповідності рівню Tier потрібно всього лише встановити резервування: додаткові джерела живлення, додаткові кондиціонери й насоси. Ніщо не може бути далі від істини. Компоненти резервування не додають надійності, якщо не знайдене правильне місце їхнього розташування в будинку, якщо не приділена увага їхньому правильному підключенню, установці й маркіруванню.

Основне розходження між проектами рівня Tier I або Tier II і рівня Tier III або Tier IV – можливість технічного обслуговування в гарячому режимі: тобто,

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

оператор дата-центра може здійснювати обслуговування всіх системних компонентів без переривання роботи обчислювальних систем.

Компоненти резервування часто встановлюються таким чином, що однаково не усувають ризик збоїти системи. Неправильний варіант установки також обмежує можливість відключити той або інший компонент інфраструктури для ремонту або обслуговування, не торкнувшись критичних обчислювальних навантажень. Інші розповсюджені помилки – це розташування контролерів і вентилів у місцях, куди у випадку аварії не просто добратися, або неправильне маркування, або безсистемне розташування цих компонентів. Уніфіковані проектні рішення затягають проектувальників у пастку й змушують їх робити помилки, яких можна було б уникнути.

Стандарт ТІА 942-А попереджає: «Важливо розуміти, що деякі навмисні або випадкові події, або природні події представляють ризик для операцій дати-центра. Для проектувальника, керуючого й адміністратора дата-центра важливо не тільки осмислити й спробувати зм'якшити ризики, викликані цими факторами, але також мати план на випадок непередбаченої ситуації».

### **Вибір, від якого залежить рівень сертифікації**

Який би сертифікат не одержав ЦОД – Uptime або ТІА – будь-який проект і будь-яке будівництво, націлені на виконання критичної місії, вимагають співробітництва з компетентною командою, щоб уникнути пасток.

Якщо система рівнів tier колись побачить смерть і втратить значимість, то це відбудеться, швидше за все, через неправильне її використання. Якщо занадто багато дат-центри, що заявляють про свою відповідність рівням Tier III або Tier IV, на ділі не зможуть забезпечити безперебійну роботу, особливо такі, які надають хостинг високопрофільним хмарним сервісам, що зачіпають широке коло користувачів, то поняття tier втратить зміст для операторів дата-центрів і їхніх клієнтів.

Цілком можливо побудувати такий дата-центр, що буде відповідати необхідному рівню продуктивності й надійності без проходження складного

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

процесу сертифікації від Uptime, TIA, або LEED (Американської Ради по Зелених Будинках) і інших. Правда, при цьому неминучі витрати, пов'язані з комплексною експертизою проекту на відповідність рівню tier, додатковою деталізацією проектних документів і більше строгим наглядом у процесі будівництва. Ці фактори, як правило, з'їдають весь ефект скорочення витрат і часу, які дає відмова від проходження формальної сертифікації.

Ретельна партнерська перевірка може вилитися в процес, що зовні нагадує процес сертифікації, виявити ризики простою, але при цьому з більшою гнучкістю підійти до реалій проекту, ніж стругаючи сертифікаційна комісія.

Будь-які наміри привласнити собі ранг tier без строгої формальної сертифікації є незаконними. Пройдете ви сертифікацію або не пройдете, ранжирування по tier безумовно дає загальну платформу й принципи для комунікації в індустрії, які всі розуміють.

### **15 основних правил і практичні аспекти вибору рішень**

Тема створення й експлуатації фізичної інфраструктури центрів обробки даних (ЦОД) за останнє десятиліття зробила якісний стрибок. Вона стала досить популярною й з вузькоспеціалізованої області перемістилася у фокус інтересів великої кількості компаній, які тією чи іншою мірою використовують інформаційні технології й устаткування у своєму бізнесі або мають безпосереднє відношення до тематики ЦОД. Де-факто ЦОД стали відігравати роль консолідуючої точки взаємодії компаній різних сфер економіки, до яких можна віднести ІТ-компанії, розроблювачів ПЗ й устаткування, компанії, які працюють у сфері послуг – телекомунікаційних, будівельних, енергетичних, що реалізують інфраструктурні проекти, а також системних інтеграторів.

ЦОД, як і будь-яка складна система, вимагає розгляду завдання проектування в комплексі з урахуванням значної кількості зв'язаних факторів для досягнення компромісів і пошуку оптимальних конфігурацій техніко-економічної властивості. При цьому варто помітити, що застосування в сукупності яких-небудь конкретних рішень в області ЦОД зовсім не гарантує одержання гідного

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

результату. Навіть якщо вони позиціоновані як призначені для рішення даних завдань. Також не завжди можуть бути розглянуті фактори взаємного впливу, реалізації перспективних технологій, їхнє сполучення із класичними архітектурами систем та інші моменти.

Якщо говорити про конкретних фахівців, на яких покладають завдання приватного або комплексного проектування або управління проектами, то тут цікаво відзначити наступні моменти. Як правило, для зручності роботи проєктувальники у своїй практиці формують із вимог великої кількості нормативних документів набір емпіричних правил. Ці правила у свою чергу доповнюються існуючою проєктною практикою й опираються на прецеденти з багатого досвіду. Для головних інженерів проєкту або функціонально аналогічних фахівців, які визначають загальну технічну політику, уміння виділяти й формувати ключові принципи є принципово важливим. Перспективно запропонувати таким фахівцям загальні аналогічні підходи в області технологічного проектування ЦОД, які дозволять їм гнучко підходити до розглянутих питань, приймати ті або рішення, опираючись на більше широкій базис.

Коротко сформулюємо даного правила:

1. Робіть технологічний розділ.
2. Дотримуйте вітчизняних стандартів.
3. Сумнівайтесь у всім.
4. Вирішуйте від завдань.
5. Робіть збалансовані рішення.
6. Уникайте зв'язаних рішень.
7. Проектуйте як можна простіше.
8. Застосовуйте уніфікацію.
9. Створюйте ЦОД добротним.
10. Безпека людей первинна.
11. Не забувайте про дріб'язки.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

12. Розглядайте гірший випадок.
13. Думайте про інтереси всіх сторін.
14. Озвучуйте ризики клієнтові.
15. Проекуйте на перспективу.

Розглянемо кожне з даних правил більш докладно. При цьому відразу варто звернути увагу на розходження термінів «проектна документація» і «проект». Останній термін буде застосовуватися до опису повного циклу реалізації ЦОД – від ідеї до експлуатації.

### **Правило № 1. Робіть технологічний розділ**

Часто серед IT-Фахівців або пов'язаних із ЦОД професіоналів виникає улюблене й риторичне питання філософського плану: чим ЦОД відрізняється від серверної кімнати? Залежно від світогляду колег по індустрії і їхні визначення понять ЦОД і серверної може бути даний широкий діапазон відповідей: від ніякої різниці до ґрунтового гідного переліку нюансів.

Якщо припустити, що різниця в термінах все-таки існує, то одним з універсальних відповідей, на який варто звернути увагу, є наступний: у підході до постановки завдання, методах і принципах проектування. У ЦОД всі технічні рішення повинні бути підлеглі конкретним цілям, об'єднані єдиним логічним змістом функціонування й прив'язані до забезпечення основного бізнесу-завдання.

Приведемо аналогію. При розробці проектної документації на спеціалізовані об'єкти, будь те виробничий цех або, наприклад, ресторан, у главу кута ставиться бізнес-завдання, що формулюється технологом у складі роздягнуло ТХ. Дана марка, поряд з архітектурними рішеннями, є відправною для інших розділів проектної документації.

Такий же підхід доцільно зпроектувати на завдання створення ЦОД.

Розділ ТХ допоможе зв'язати й гармонізувати IT-системи ЦОД і телекомунікаційну інфраструктуру (у першу чергу як елементи ЦОД, що приносять реальний дохід) з фізичною інфраструктурою, що забезпечує.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

Дозволить уникнути білих плям, розрізненого існування систем, «звалених у купу» в одному приміщенні, проектування інфраструктури заради інфраструктури.

У технологічному розділі повинні бути відбиті наступні питання:

- проведено загальний опис технологічних рішень;
- виділено основні технологічні зони;
- зазначено перелік, розміщення й логіка роботи технологічного

встаткування;

– сформульовано вимоги до інженерного забезпечення технологічного встаткування й варіанти забезпечення даних вимог;

- показано взаємозв'язок між всіма елементами системи;
- визначено основні технологічні параметри, хоча б у макропоказниках.

Основою при створенні технологічного розділу є планувальні рішення ЦОД, або Room Layout (в англійській термінології).

Розробка технологічного розділу несе пряму практичну користь для всіх учасників проекту: замовника, проектної команди, третьої й наступної сторін, які можуть бути залучені в проект на будь-якому етапі.

– ЦОД представляється як єдиний об'єкт, конкретні технічні рішення приймаються в канві єдиної логіки й з урахуванням інтересів всіх учасників проекту;

- є технічний опис для підтримки прийняття економічних рішень;
- розділ сприяє збалансованості проекту в цілому;
- розділ обмежує максимальні потреби в ресурсах;
- ключові топології й схеми затверджуються для подальшої розробки;
- розділ відповідає на запитання, чому ті або інші рішення були прийняті;
- клієнт має формальне подання про те, що він одержить.

Українська система розробки проектної документації в будівництві для технічно складних об'єктів історично має досить налагоджену й формалізовану процедуру стадійного проектування:

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

- ескіз або концепція;
- стадія «Проект» або «Проектна документація» (раніше – техніко-економічне обґрунтування);
- робоча документація.

Робити або не робити всі кроки, визначення повноти кожного розділу або їхнє об'єднання – все це залежить від особливостей і масштабу проекту, бажання клієнта, культури проектування виконавця. Як описувати й формувати розділ, названий тут «технологічним», особиста справа кожного фахівця. Проте при розробці документації повинен бути знайдений формат документа, що відіб'є всі вищевикладені питання.

## **Правило № 2. Дотримуйте вітчизняних стандартів**

У спадщину від радянської системи нормативної документації нам дісталися десятки тисяч стандартів. В українських реаліях уже більше десяти років триває реформа в області технічного регулювання. Все це накладає певний специфічний відбиток на процес проектування й використання стандартів як точки опори при прийнятті рішень.

Реформа змінила обов'язковий характер дотримання стандартів у бік рекомендаційного, додала імпульс до руху у бік світової практики, гармонізації й запозичення. З'явилися нові види нормативних документів. Виконання більшості стандартів при цьому в цей час носить добровільний характер, обов'язковими залишилися стандарти, які впливають на безпеку життєдіяльності. Ситуація збільшується тим, що профільних вітчизняних нормативних документів практично немає, а практичне використання вимагає певної навички, уміння аналізувати й трактувати. Немаловажним є постійне відстеження поточної ситуації в області технічного регулювання й розуміння фактичних і юридичних наслідків застосування того або іншого документа.

Варто дати кілька рекомендацій, які дозволять уникнути проблем і в той же час одержати бажані технічні кондиції. Необхідно:

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

- прописувати стандарти або необхідні вимоги в договорах і технічних завданнях;
- при порівнянні вимог використовувати найбільш тверді;
- розробляти власні стандарти;
- строго впливати обов'язковим державним стандартам (архітектура й конструктив, пожежна безпека, екологія, електропостачання).

### **Правило № 3. Сумнівайтеся у всіх**

Як уже говорилося вище, при створенні ЦОД інженери-проектувальники зіштовхуються із цілим рядом проблем інформаційно-нормативної властивості. Цьому сприяють постійно, що розвиваються технології, підходи, віяння й тенденції, актуалізація й перегляд стандартів і інших нормативних документів. Наявність різних міжнародних експертних і консультативних організацій, які формують додаткові нормативні документи у вигляді, наприклад, кращих практик або технічні документи. Виклад проблем і питань, пов'язаних із принципами побудови ЦОД конкурентним образом, аж до суперечливих поглядів і підходів усередині професійного середовища. Додають плутанини агресивні моделі продажів устаткування й рішень від вендорів, які відзначають достоїнства, але приховують недоліки, поверхово ставляться до питань інтеграції, роблять некоректні порівняння маркетингового плану на користь свого рішення. Все це збільшується принциповою складністю завдання комплексного проектування, а також необхідністю враховувати специфіку українського законодавства й регулюючих документів.

Корисний вивід, якому можна із усього це зробити: абсолютних рішень не існує, потрібно сумніватися у всіх.

Тезовим образом можна сформулювати деякі практичні ради, які допоможуть поліпшити якість проекту. Кожний з них має свої за й проти.

- Підтримка професійного рівня фахівців, які зможуть бути в курсі новинок і проводити самостійний аналіз.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

– Скептичне відношення до застосування нових «смажених» технологій, пошук сильних і слабких сторін, аналіз «мнимої» вигоди в порівнянні із класичними рішеннями.

– Звертання до зовнішньої експертизи, що можуть надати незалежні спеціалізовані або комерційні організації, вендори, а також приватно експерти-фрілансери.

– Використання готових комплексних рішень, які при цьому є однозначно більше дорогими й накладають певні обмеження на перспективні модернізації.

– Облік технічних, юридичних, економічних і інших наслідків впровадження того або іншого рішення.

#### **Правило № 4. Вирішуйте від завдань**

Раніше згадувалося, що одним з типових оман є створення фізичної інфраструктури заради її самої. При побудові ЦОД вибираються й створюються системи електроживлення й кондиціонування, газового пожежогашіння й т.д. Кожна окремо й, як правило, всі разом вони є «верхи досконалості».

Подібна робота, безумовно, важлива, але як можна розглядати питання проектування ЦОД без оцінки потреб ІТ-устаткування, що буде розміщене в ньому? Подібна ситуація часто зустрічається при проектуванні комерційних ЦОД, коли неясні передбачувані потреби орендарів або корпоративних клієнтів, коли невірно оцінені власні потреби. У цьому випадку проявляється стійка тенденція робити все по досяжному максимумі ресурсів, які доступні для конкретної площадки. Однак ЦОД, побудований по максимуму, може виявитися зовсім не затребуваним, а це значить, що ККД буде низьким і економіка бізнес-процесу буде страждати.

Для ілюстрації можна сформулювати дві тези.

Могутніше – не значить краще. Домірність завданням.

Найважливішим критерієм є вибір тих або інших рішень відповідно до передбачуваних завдань. Ніж вірніше будуть оцінені потреби, тим більш успішний буде проект. Що означає змусити шафами всю площу, розподілити всю

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

доступну електроенергію між шафами, вивести ЦОД на рівень максимально можливого Tier? Це значить примусити до необґрунтованих витрат клієнта й поставити під погрозу реалізацію проекту. Чи вірний це підхід?

Типовий приклад: некритичний до часу простою ЦОД, завдання якого за рівнем надійності цілком відповідають Tier I або Tier II, про всякий випадок позиціонують як Tier III або Tier IV. За результатами проектування й визначення кошторису бюджет і бізнес-план починає тріскотіти по швах. Починається зменшення вимог, наприклад, по частині нібито некритичних систем. Проект перекроюється на ходу, стає незбалансованим, спрощується настільки, що вже не відповідає ніякому Tier. Такий ЦОД (або це вже ІТ-смітник) потрібний клієнтові? Може бути, ринку? У цьому випадку й проектній команді, і замовники займалися самообманом.

З іншого боку, при певних моделях бізнесу доступність і надійність ІТ-устаткування може бути підвищена іншими методами, наприклад, шляхом створення розподілених площадок з резервуванням, дуплікування даних і т.д. Тобто варто відсторонитися від фізичної інфраструктури ЦОД, глянути на проблему надійності ширше, вийти за рамки штампів, що нав'язуються, по фізичній інфраструктурі й використовувати можливості телекомунікаційних і ІТ-систем.

У якості ще одного приклада відповідності рішень перспективним завданням можна дуально проаналізувати популярну останнім часом ідею реалізації ЦОД з використанням високонавантажених стійок, потужність кожної з яких становить від 15 до 30 кВт, а те й вище. Немає сумнівів у тому, що поточний розвиток ІТ-устаткування створює перспективи для таких потреб. Однак на противагу потрібно враховувати й такі широко відомі факти: у світі середня потужність на одну стійку становить близько 5 кВт, і часто орендарі, особливо в Україні, не хочуть сусідити в одній шафі з іншими орендарями, воліючи викуповувати цілу шафу, нехай він і використовується тільки наполовину або на дві третини.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

## Вирішуйте від економіки

Будівництво ЦОД – досить витратна річ. Якщо ж говорити про інвестиції, то будь-яке вкладення у високі технології – це проекти, що володіють високими ризиками. Крім технічних або топологічних параметрів в аналіз проектних рішень вступають фактори економічного плану, які варто приймати в розрізі: застосовність – функціональність – економічна доцільність.

Варто оцінювати витрати як капітальні (CAPEX), так і операційні (OPEX). Звичайно вони мають полярні залежності, і потрібно вирішувати завдання по пошуку оптимуму.

Як ілюстрація розглянемо популярну тему впровадження нових «зелених» технологій. Питання для України не очевидний. Вартість ресурсів ще низька, а витрати ґрунтовні. Строк окупності технології може бути зіставимо з життєвим циклом ЦОД. Чи так коштує їх впроваджувати? У кожному випадку потрібно вирішувати індивідуально.

Ще одним важливим фактором є параметри, які заставляються в бізнес-план, коли вибудовується модель окупності ЦОД. Що реально продається або на чому працють гроші у випадку кінцевого клієнта? Може трапитися, що вважати в реках і юнітах не дуже правильно, якщо в бізнес-моделі гроші працють на бітах і флопах. І якщо ми оцінюємо ЦОД як інформаційний масив, те можливе знаходження цікавих технічних рішень у сфері фізичної інфраструктури, які можуть відрізнитися від класичних схем.

### Правило № 5. Робіть збалансовані рішення

Під збалансованістю рішень розуміється наскрізна відповідність якомусь заданому рівню всіх складових ЦОД у декількох розрізах і з метою реалізації основного задачі-бізнесу-завдання. Основні параметри ЦОД повинні бути збалансовані, тобто відповідати один одному. Тут важливо помітити, що збалансовані рішення – не обов'язково рішення зв'язані, і це питання буде розглянутий далі в Правилі № 6.

					ВКРМ-123.24.0027.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Обмеження по одному із ключових параметрів зведе функціональні можливості ЦОД до цього самого обмеження. У якості основних можна виділити збалансованість у розрізах:

- по макропоказниках;
- по технічних рішеннях;
- за вартістю рішень.

### **Збалансованість по макропоказниках**

Для аналізу ЦОД можуть бути застосовані загальнопоширені або певні самостійно коефіцієнти й показники.

Кожний з таких показників добре відомий і, як правило, фігурує в описі технічних характеристик ЦОД. Тут можна відзначити різноманітні питомі показники: площа на шафу, потужність електроенергії на шафу й т.д., а також найбільш популярний на справжній момент макропоказник – PUE (або DCiE (DCE, DCPE) – величина, зворотна PUE), що по популярності останнім часом поза конкуренцією.

Розглянемо приклад. Обмеження, що накладаються при плануванні/розрахунку PUE. Для одержання «гарних» (близьких до одиниці) значень PUE очевидно потрібно підвищити споживану встаткуванням потужність і знизити потужність, що витрачають інші споживачі. Однак для повноти аналізу не можна оперувати тільки двома максимальними цифрами – потрібно розглядати систему в цілому, проводити комплексний аналіз при різних режимах роботи, при неповнім завантаженні й т.д. Інакше можна одержати «віртуальне» прийнятне значення PUE, що на практиці буде далеко від ідеалу. А практичного результату при цьому прийде чекати цілий рік.

Більше цікавими показниками з погляду виконання реальних бізнес-завдань є параметри по продуктивності, які останнім часом набирають вагу. DCP (продуктивність дата-центра), що показує, що ж на виході видає ЦОД стосовно витрачених ресурсів.

$$\text{Datacenter Productivity} = \text{Useful Work} / \text{Total Facility Power}$$

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

## **Збалансованість по технічних рішеннях**

Різні системи оцінки надійності (Tier I-IV, Tier 1-4, Critically Levels, STARS, інші клони й варіації) так чи інакше відповідають один одному й ніякому образу не гарантують певний час простою.

Тоді чому ж вони так популярні, особливо Tier, в Україні?

У рамках розгляду даного правила можна запропонувати провокаційна теза, що використання даних класифікацій саме й формує баланс по технічних рішеннях і тим самим поліпшує архітектуру ЦОД. При цьому бездумне проходження пунктам метрик найчастіше неефективно. Воно не враховує можливих локальних умов, модифікація яких можлива тільки після трактування того або іншого показника. При аналізі важливо зрозуміти підхід і логіку, чому так зроблено, що коштує за прийнятими рекомендаціями.

Який рівень технічних рішень варто вибирати? Це залежить від рівня й важливості завдань. Tier III або Tier IV, до яких прагнуть «про всякий випадок», виправданий, тільки коли мова йде про життя людей і системах життєзабезпечення, системообrazуючих державних і фінансових інститутах, об'єктивно критичних сервісах.

## **Збалансованість за вартістю рішень**

Вартість рішення є не тільки економічним параметром – це й один з найважливіших параметрів оцінки якості рішення.

Для збалансованості можна припустити, що всі складові ЦОД повинні лежати в одному ціновому сегменті. ЦОД у смітнику (Trash Data Centers) – дуже гарна ідея й рішення. Але це ідея, що ілюструє нижній ціновий сегмент некритичних до часу простою сервісів.

На закінчення Правила приведемо кілька ідей, як одержати збалансовані рішення. Необхідно:

- виробити рішення, опираючись на обраний рівень;
- застосувати й скористатися ідеологією оптимізованого по продуктивності ЦОД (POD), що просувається деякими організаціями;

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

– максимально використовувати рішення від одного виробника або від спільних альянсів.

### **Правило № 6. Уникайте зв'язаних рішень**

Попереднє Правило рекомендує робити збалансовані рішення, а поточне, на перший погляд, суперечить йому. Тонка грань розходження між цими двома принципами й уміння правильно їх застосовувати визначає одну зі сторін мистецтва проектувальника.

Основний зміст Правила № 6 – зробити рішення гнучкими і масштабованими, закласти на етапі проектування кілька ступенів волі.

ЦОД є складним багатofакторним комплексом. На етапі проектування можна розробити й збалансувати ЦОД, однак це не дає гарантії, що моделювання було проведено успішно або були враховані всі фактори. Також на стадії реалізації проекту можуть вноситися зміни. Все це приводить до того, що при управлінні змінами в проекті при наявності зв'язаних рішень ЦОД починає «тріскотіти по швах». Якщо в проекті параметри були задані жорстко, а системи так само жорстко були зв'язані між собою по цих параметрах, то рух одного з них у кожній із систем може зажадати перепроєктування ЦОД у цілому.

### **Уникайте технічно зв'язаних рішень**

При проектуванні варто задати запас по варіативних параметрах, але не на розвиток, як у нас часто прийнято обґрунтовувати, а на балансування ЦОД і адаптацію під конкретні потреби. У практичному плані величини запасу повинні лежати в межах 10-25 %.

Як приклад можна привести популярну останнім часом ідею ЦОД змінної щільності (Density Zone). Тут формуються три рівні завантаженості ЦОД по продуктивності ІТ-устаткування (наприклад, у рамках модуля):

- зона високої щільності;
- зона середньої щільності;
- зона низької щільності.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

Вигоди від такого підходу полягають у мінімізації забезпечення ресурсами про запас, зменшенні експлуатаційних витрат на 30-40 %. Висока щільність забезпечується тільки там, де необхідно. При цьому рішення повинні дозволяти швидко й без капітальних вкладень реконфігурувати зону на один щабель нагору або долілиць.

### **Уникайте економічно зв'язаних рішень**

Заміна невірної застосованого технічного рішення може зажадати капітальних витрат на модернізацію. Аналогічні витрати знадобляться у випадку неправильного позиціонування ЦОД в економічній моделі. Замість того, щоб приносити доходи, ЦОД стане збитковим, тяжким тягарем.

Прикладом реалізації гнучкого з економічної точки зору ЦОД є дата-центр змінного рівня (Multi-Tier DC). У рамках одного об'єкта формуються модулі ЦОД – кожний зі своєю прив'язкою до рівня Tier. Відбувається диверсифікованість ЦОД під різні ІТ-завдання й сервіси зі своїм рівнем цінової політики.

Потреба в тому або іншому рівні сервісу із пропорційними витратами визначається ринком. Вигоди від даного підходу полягають у витратах на формування тільки базового рівня сервісу, зменшуються капітальні й експлуатаційні витрати, з'являється можливість гнучких реконфігурацій, а виходить, росте економічна ефективність використання ЦОД.

Рішення повинні дозволяти швидко й без капітальних витрат реконфігурувати модулі під необхідний рівень сервісу на один щабель нагору або долілиць.

### **Правило № 7. Проектуйте як можна простіше**

Кожний активний або нестационарний елемент інженерних систем ЦОД є потенційною точкою відмови. Такими елементами можуть бути як великі блоки, так і локальні: контакти, рознімання, стики, переходники, пристрою комутації й т.д. Ступінь впливу кожної такої точки відмови на працездатність ЦОД нерівномірна й повинна бути оцінена в частині потенційних наслідків.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

Перебільшуючи, можна вказати, як часто при проектуванні проста схема, на догоду додатковим (найчастіше незатребуваним) сервісам або просто ідеям і бажанням, обростає додатковими елементами, рішеннями. Негативні ознаки, через які треба у свою чергу йти на чергові ускладнення. А є чи дійсна вигода від ускладнення схем, топологій, рішень і т.д.? Мудрована схема дорожче, навряд чи надійніше, суужніше в розумінні, при цьому вповільнюється час реакції в аварійній ситуації й ухвалення рішення по її ліквідації або мінімізації збитку.

Як приватна ілюстрація (без претензії на рекламу) можна привести рішення з коминковою трубою над шафою, при якому в системі повітрообміну відсутні активні елементи, крім тих, які перебувають у прецизійних кондиціонерах.

### **Правило № 8. Застосовуйте уніфікацію**

У продовження попереднього Правила хотілося б відзначити очевидне рішення, що допоможе зробити ЦОД простіше. Очевидно, що пройшли ті часи, коли підряд на створення ЦОД був разовим замовленням, а сам ЦОД – штучним виробом. Дозволити зараз таку розкіш можуть тільки при будівництві великих ЦОД з реалізацією певної ідеї. Більшість ЦОД, які створюються в цей час, мають досить стандартне компонування й устаткування. Уніфікація є наслідком масштабного промислового виробництва встаткування й великої кількості проектів.

Уніфікація ЦОД може бути реалізована на будь-якому рівні – від однакових болтів для кріплення встаткування в шафах до закінчених модульних або контейнерних рішень під ключ.

Вигода від використання уніфікованих рішень очевидне: зменшення часу розгортання й відновлення після аварії, сумісність і взаємозамінність компонентів, економія на запасному встаткуванні й матеріалах.

У черговий раз можна проілюструвати дане правило прикладом з життя. Припустимо, що на одному об'єкті існує кілька установок газового пожежогасіння. Відповідно до вітчизняних норм в області пожежної безпеки, на

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

об'єкті повинен зберігатися запас балонів з газовою сумішшю для відновлення працездатності системи протягом доби. Якщо при проектуванні застосувати уніфікацію газових балонів, то для виконання вимоги по резерві досить буде тримати балони тільки на саму більшу установку. Тим самим досягається досить істотна економія й зменшується місце на складі, необхідне під розміщення резерву.

### **Правило № 9. Створюйте ЦОД добротним**

ЦОД повинен бути зроблений доботно. У це слово заставляється й забезпечення надійності, і необхідна якість, і закладений запас міцності. Надійність системи в цілому є сукупністю характеристик надійності її компонент, їхнього зв'язку й впливи один на одного. Всі застосовувані рішення бажані для реалізації в одному рівні якості (це питання вже зачіпалося при описі збалансованості ЦОД). Варто мінімізувати кількість точок відмови. Аналізувати можливий запас міцності на працездатність навіть при відсутності належного обслуговування.

При застосуванні того або іншого встаткування або матеріалів, якщо з'являються сумніви, варто вивчити технічні характеристики, що заявляються, і показники надійності. Основними показниками є термін служби й ресурс. Також потрібно уточнити статистичні дані, розрахункові й прецедентні: імовірність безвідмовної роботи, середній час наробітку до відмови або на відмову. При схожих економічних параметрах варто робити вибір на користь більше відказостійких рішень. Немаловажним фактором тут є неформальна репутація встаткування на ринку, що, безумовно, потрібно брати до уваги.

Виходячи з даних по відказостійкості, а також логіки функціонування ЦОД, варто ухвалювати рішення щодо резервуванні, дублюванні й інших практичних і відомих схемах підвищення надійності. Як відомо, скупий платить двічі. Реалізація тих або інших схем забезпечення надійності повинна опиратися на цифри очікуваних втрат у випадку простою. Тут слід дотримуватися

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

розумного балансу й розраховувати ймовірності виникнення події з величиною збитку, що може бути нанесений цією подією.

І ще один немаловажний фактор, що у меншому ступені ставиться до етапу проектування, але може бути застосований у рамках авторського нагляду. Варто стежити за добротною і якісною реалізацією проектних рішень. Навіть саме надійне встаткування може бути халтурно змонтовано, а неповний комплекс пусконаладжувальних робіт не дозволить реалізувати всі функціональні можливості.

### **Правило № 10. Безпека людей первинна**

Однієї з очікуваних характеристик правильно побудованого ЦОД є безпека. При цьому виділяють безпека фізичну, протипожежну, інформаційну.

Ще однією найважливішою стороною реалізації безпеки є забезпечення нормального, безінцидентного знаходження й роботи в ЦОД персоналу й відвідувачів.

Цікаво виділити два випадки.

Забезпечення безпеки людей не вступає в конфлікт із технічними системами безпеки. У цьому випадку необхідні заходи повинні бути реалізовані в проектних рішеннях або виконуватися адміністративними механізмами при експлуатації ЦОД. Прикладом першої тези може бути, наприклад, установка захисних конструкцій, що обгороджують, які б перешкождали потенційним травмам на виробництві. Так, при проектуванні часто забувають установлювати поручні на перепадах висот фальшполу, пандусах або сходах. Для ілюстрації другої тези можна привести випадок, коли відкрита плитка фальшполу приводила до перелому ніг співробітника.

Другий інцидент виникає, коли безпека людей залежить від працездатності ЦОД або зачіпає логікові роботи технічних систем безпеки. У цьому випадку варто вказати, що безпека людей первинна. Як правило, основна увага даного питання лежить в області пожежної й електробезпечності. Відомо, що вихід газу установки пожежогасіння повинен бути заблокований у випадку

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

знаходження в ЦОД людей. Шляхи евакуації повинні бути вільні, навіть якщо це порушує концепцію фізичної безпеки об'єкта. У цьому випадку варто приймати додаткові міри, що компенсують.

Також додаткові або спеціальні рішення повинні бути реалізовані, якщо за умовами функціонування ЦОД зобов'язаний забезпечити безперервну роботу. Наприклад, для запобігання техногенної аварії. Модель забезпечення безпеки в цьому випадку також повинна виходити з основного принципу – безпеки людей.

### **Правило № 11. Не забувайте про дріб'язки**

Як відомо, є достатня кількість прислів'їв і приказок про важливість незначних на перший погляд речей. Наприклад, «диявол криється в дріб'язках» або «малий золотник, так доріг».

Що відбувається із системами або компонентами ЦОД, користь і потреба яких не зовсім очевидна? Виконавець починає про них забувати, а клієнт – заощаджувати!

В одному із ЦОД проектувальники встановили прозорий фальшпол, щоб відвідувачі дата-центра бачили логічну, прекрасно організовану структуру комунікацій, розташованих у підпільному просторі. Зрозуміло їхнє бажання підкреслити достоїнства свого ЦОД, особливо в тому місці, де можна розслабитися й зробити як потрапило – однаково ніхто нічого не побачить. Ознака професіоналізму – доводити справа до логічного кінця, до останньої деталі – того самого «дріб'язку».

Схожа категорія рішень по якості повинна бути наскрізна для всіх систем ЦОД. Відмова незначного або малопомітного компонента може привести до збою роботи ЦОД, з однієї сторони. Неуважність до дріб'язків при експлуатації може також привести до аварійної ситуації, що може бути спровокована в тому числі й людському факторі. Незначна в технічному плані або за вартістю частина системи або комплексу ЦОД може стати досить реальною точкою відмови або істотно знизити сервісні функції.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

Як типові приклади безпам'ятності, особливо при реалізації українських проектів, можна вказати, наприклад, відсутність правильно зробленого маркування ЦОД або функціонального заземлення.

### **Правило № 12. Розглядайте гірший випадок**

Дане правило не призиває читача мати рішення на всі випадки життя, включаючи також ядерну війну або навалу інопланетян. Проте прийняті рішення повинні опиратися на аналіз потенційних подій або модель ризиків, якщо вона існує. З одного боку, варто обмежувати коло зовнішніх подій, які можуть трапитися в доступному для огляду майбутньому стосовно до доцільності підтримки бізнес-процесів і працездатності ЦОД як з погляду економіки, так і здорового глузду. З іншого боку, не слід бути безтурботними, забувати або не взяти до уваги ризику, від яких ніхто не застрахований.

Другий аргумент на користь даного правила – принцип підтримки прийняття проектних рішень. Якщо їсти можливість зробити добре, то навіщо робити погано? Припущення гіршої ситуації буде додатковим критерієм і дозволить зробити правильний вибір при рівності за й проти декількох проектних рішень.

Наприклад, знаходження ЦОД нижче рівня землі або в заплаві водойми навіть за умови наявності прекрасної гідроізоляції й зливової каналізації, а також сигналізації про протічках, коли в стандартних умовах площадці не загрожує затоплення, не є правильним. Що може трапитися, – рано або пізно відбудеться. Персонал, що обслуговує ЦОД, повинен виконувати всі регламенти або приписання, бути уважним і зосередженим, але це не виходить, що не потрібно вбудовувати у свої рішення «захист від дурня» або систему оповіщення про критичну ситуацію, до якої може привести можлива недбалість персоналу.

### **Правило № 13. Думайте про інтереси всіх сторін**

Звичайно при проектуванні й будівництві ЦОД доводиться вирішувати величезна кількість поточних завдань. У цьому вирі подій на задній план іде розгляд питання експлуатаційної придатності дата-центра і його окремих систем.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

Питання ж про капітальну реконструкцію або взагалі конверсії площадки представляється чимсь із області фантастики.

Облік сподівань інженерів по експлуатації ЦОД може бути зрозумілий або тими, хто побував у їхній шкірі, або придбаний з багаторічним досвідом. Розгляд даного правила може бути представлене в різних ракурсах.

Часто при розробці об'ємно-планувальних і інших рішень на угоду одержання максимальних показників забуваються вимоги по елементарних сервісних зонах обслуговування, зручності доступу до замінних елементів системи й т.д. Доходить до того, що заміна великого блоку якої-небудь системи неможлива без зупинки ЦОД або проведення спеціальних будівельних робіт.

При виборі рішень потрібно враховувати, які з них є менш витратними при експлуатації, наскільки доступні запчастини й видаткові матеріали, наскільки складно знайти фахівців, які будуть обслуговувати спроектовану екзотичну систему, скільки становить параметр часу відновлення після збою або аварії.

У проекті обов'язково потрібно враховувати ЗПП, формувати його состав.

У проект також необхідно закласти необхідний сервісний інструмент: присоски для підняття фальшполу, викрутки, ліхтарі. При цьому визначити їхнє розташування й навігацію, що дозволить одержати швидкий доступ у випадку аварійної ситуації.

Як уже говорилося раніше, при проектуванні варто закладати можливості регулювання параметрів систем, що також є важливим для експлуатації.

Схеми автоматики й диспетчеризації повинні бути зрозумілі й логічні. Може бути реалізована система підтримки ухвалення рішення. Можна вмонтувати у функціонала нагадування про сервісне обслуговування. Якщо клієнтові потрібно, бажано надати допомогу в розробці процедур, регламентів, ситуаційних карт.

#### **Правило № 14. Озвучуйте ризики клієнтові**

Багато рішень ЦОД є інноваційними. При цьому, як уже говорилося, вони можуть бути не до кінця доробленими, і навряд чи клієнт захоче виступати в ролі

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>40</b>

«бета-тестера». А можуть бути самопальними, створеними кустарним способом, що теж ризиковано, тому що надійність непромислових рішень може викликати питання. Ініціація суперсучасних технологій звичайно йде з боку клієнта, що «понаслухався, поначитався» або хоче, щоб усе було «круто».

У боротьбі за ексклюзивні показники енергоефективності або інші «фішки» у технічних завданнях іноді прописуються рішення й параметри, які або не мають під собою бази, або суперечать самі собі. Та й реалізація їх перебуває під великим сумнівом.

Іноді в главу кута ставиться одна із систем, що накладає велику кількість обмежень на суміжні системи, архітектуру, весь ЦОД у цілому. Можливо, у якихось конкретних ситуаціях це непогано, але при цьому проектувальникам категорично не рекомендується займатися самовільним прийняттям рішень. Варто озвучувати всі видимі й передбачувані ризики клієнтові, докладно працювати з ним на кожному етапі проекту. Тим самим можна домогтися більше рівного плину проекту, убезпечити себе від подиву клієнта на останньому етапі, коли він побачить, що це зовсім не той ЦОД, що йому представлявся.

При будь-якому сумніві дуже корисно сформулювати проблему перед клієнтом. По-перше, формальне формулювання питання вже, як відомо, дає половину відповіді. По-друге, якщо проектувальник не перекладає відповідальність на клієнта, те як мінімум утягує його в процес ухвалення рішення.

Як приклад цікаво привести розповсюджений міф про те, що установка пересувних ДГУ на колісній базі не вимагає ніяких проектних рішень і погоджень. Іноді, дійсно, такий ДГУ може стояти як завгодно довго й не викликати питань. На противагу цьому – для клієнта вже на етапі експлуатації може стати неприємним сюрпризом перевірка фахівців і повідомлення про нелегальну установку ДГУ.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

## Правило № 15. Проекуйте на перспективу

Інформаційні технології, а слідом, мабуть, і інфраструктурні рішення міняються настільки швидко, що є ризик їхнього морального старіння за час циклу реалізації проекту. Єдиний спосіб якось урахувати й компенсувати цю ситуацію – дивитися ледве далі, на перспективу, і приймати відповідні технічні рішення. Планування ЦОД в Україні вперед на 3-4 роки, на погляд автора, є оптимальним. З урахуванням реалізації проектів ЦОД строком до двох років у практичному змісті цілком достатньо цікавитися поточними реалізаціями в США й інших розвинених країнах миру. Хоча при цьому потрібно враховувати, що строки відставання й запозичення рішень і технологій скорочуються, що тенденцій досить багато – одні з них більше популярні в даний момент, інші будуть більш-менш популярні в майбутньому, і не все з них дають однозначно кращий результат.

Приводити в рамках даної статті поточні тенденції є зайвим, віяння й мода проникнули в усі сфери нашої діяльності. Найчастіше за гарною вивіскою криється або добре забуте старе, або утопічна, але заворожлива ідея, а може бути, і класичне рішення, ажіотажно роздуге маркетологами.

Набагато важливіше розуміти, на що опирається дана тенденція, чи затребувана вона, є чи в неї бізнес-рушій, чи можуть бути запропоновані конкуруючі ідеї. Реалізація рішень, які ще не були апробовані, є відчутним ризиком як з погляду техніки, так і економіки.

Щоб одержати інформацію, куди рухається ринок ЦОД, досить взяти участь у тематичних конференціях, почитати статті, для особливо недовірливих – дослідження шановних консалтингових компаній, наприклад Garther. У століття інформаційних технологій це не є проблемою – досить простого бажання.

					ВКРМ-123.24.0027.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

## 2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – високорівнева мова програмування, яку називають другою за популярністю в світі. Її використовують для розробки вебзастосунків, програмного забезпечення, машинного навчання. Python застосовують для вирішення робочих завдань у компаніях Google, Instagram, Facebook, IBM, NASA, Dropbox, Netflix та інших. Розробники цінують цю мову програмування за простоту у вивченні, ефективність та мультиплатформність.

Python – скриптова мова програмування з досить простим синтаксисом. Для розуміння достатньо порівняти принципи написання найпростішої програми, яка виводить на екран текстове повідомлення. Саме тому мова програмування Python більш доступна для новачків, а професіонали встигли адаптувати її для вирішення великої кількості завдань. Це мультиплатформне рішення, тому знання Python дає можливість працювати у різних сферах: від розробки мобільних застосунків до ігрової індустрії та штучного інтелекту.

У мови програмування динамічна типізація: є можливість передавати до функцій будь-який тип даних без попереднього вказання. Інтерпретованість дозволяє знаходити помилки у коді ще до повної збірки у робочий застосунок. При цьому Python дуже чітко дає зрозуміти, де та через що виникла помилка.

Це мова об'єктноорієнтованого програмування (ООП). Програмне забезпечення на Python оформлене у вигляді моделей, які можуть бути зібраними у пакети. Тип та структуру кожного об'єкта можна запитати під час виконання програми. Для кожного з об'єктів можна отримати всю інформацію щодо його внутрішньої структури. Окрім того:

- у мови логічний синтаксис, завдяки чому вихідний код легко читати та розуміти;
- гнучкість та масштабованість Python дозволяє адаптувати високорівневу логіку та розширяти складні застосунки, як тільки виникне така необхідність;

- розробка на Python у більшості випадків проходить швидше, ніж на інших мовах програмування;
- Python – інтерпретована мова програмування. Це значить, що код можна написати у будь-якому текстовому файлі на будь-якій платформі, і потім успішно запусити;
- у Python – колосальна спільнота однодумців. Тож будь-які складнощі конкретних розробників вирішуються колективно.

Проте є декілька особливостей, які можна віднести до недоліків. Це повільність (ця мова програмування хоч і універсальна, проте повільніша за інші), велика кількість ресурсів, необхідних для роботи та «прив’язаність» до системних бібліотек.

Мова програмування Python використовується у наступних сферах:

1. Розробка програмних застосунків будь-якого напрямку.
2. Розробка серверної частини мобільних застосунків (найпопулярніший напрямок).
3. Ігри. Багато сучасних ігор для комп’ютерів (наприклад, World of Tanks) частково чи повністю написані на Python.
4. Вбудовані системи для різних пристроїв. Дуже часто Python використовують для написання внутрішніх платформ управління банкоматами.
5. Скрипти та плагіни до уже реалізованих програм для автоматизації процесів чи створення інших рішень.
6. Тестування (автоматизація цього процесу).
7. Машинне навчання. – основна мова для написання алгоритмів і аналітичних застосунків у сфері Machine Learning.

### **Бібліотеки Python**

Різні бібліотеки Python використовують для виконання конкретних завдань. Наприклад, Matplotlib підходить для відображення даних у двовимірній та тривимірній графіці. Pandas підходить для зручної роботи з даними. NumPy дозволяє створювати масиви та керувати ними. Requests використовується для

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>44</b>

веброзробки. OpenCV-Python відкриває можливості для обробки зображень з метою оптимізації систем «машинного зору».

### **Найвідоміші фреймворки для мови програмування Python**

Фреймворки Python допомагають створити зручне та функціональне середовище для розробки. У них міститься набір інструментів, модулів та бібліотек, корисних для виконання конкретних завдань. Це значно полегшує роботу: наприклад, дає змогу не витратити час на розписування дій, які повторюються, а використати релевантний інструмент. Тож є можливість позбутися рутинних процесів та сконцентруватися на логіці проєкту.

Серед найпопулярніших фреймворків для Python:

- Django – найстаріший та найвідоміший. Створений для реалізації великих інтерактивних проєктів;

- Pyramid – зручний у налаштуваннях, і дає можливість реалізувати складні нестандартні ідеї;

- Web2py – підходить в першу чергу для вебзастосунків і може використовуватись на будь-яких архітектурах.

### **Популярні Python IDE**

IDE або інтегровані середовища розробки – це програмне забезпечення, яке надає розробникам необхідні інструменти для написання, редагування, тестування та налаштування коду. Для розробки на Python найчастіше використовують IDE PyCharm, IDLE, Spyder та Atom.

## **2.3 Розгорнута постановка завдання**

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи планування модернізації ЦОД.

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

- а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;
- б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;
- в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;
- г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;
- д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;
- е) провести розрахунки по визначенню економічної ефективності розробленої системи;
- ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;
- з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

Планування ЦОДа – це постійний пошук балансу між вимогами до швидкості й дальності передачі даних, а також до гнучкості, причому остання умова є ключовим чинником. Більша гнучкість забезпечує більше простий перехід до 40, 100 і навіть 400 Гбіт/с.

Як очікується, протягом найближчих чотирьох-п'яти років більшість серверів будуть переведені із з'єднань 10 Gigabit Ethernet на 40 Гбіт/с (і частково 100 Гбіт/с), а через 10 років, можливо, і на 400 Гбіт/с. Модернізація від 10 Gb до 40 і 100 Gb може бути забезпечена за рахунок застосування 24-волоконних сполучних кабелів MPO зі знизеними оптичними втратами. Крім того, наступну міграцію на встаткування з більше високими швидкостями передачі даних необхідно тримати в розумі при виборі оптичних кросів, панелей і кабеленесучих систем.

Сьогодні в ЦОДах найчастіше використовується багатомодове оптичне волокно, що забезпечує розумний баланс між продуктивністю, щільністю й вартістю. Однак для забезпечення більше високих швидкостей або більшої дальності передачі застосовуються й одномодові волокна, які доцільно прокладати між точкою уведення й головним кросовим приміщенням (Main Distribution Area, MDA), а також між поверхами в проектах мегаЦОДів. Тому в проектувальників виникає питання: у якому місці варто організувати перехід із багатомодового на одномодове волокно, щоб знайти золоту середину між вартістю й відстанню? При подальшому збільшенні швидкості або дальності передачі, можливо, буде потрібно перейти на інше середовище передачі.

Наприклад, наявні стандарти визначають лінії до 100 м, 150 м і 10 км. Але навіть у великих ЦОДах довжину більшості ліній не перевищує 500 м, що

					ВКРМ-123.24.0027.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

створює потенційний попит на продукти й стандарти для малопотужної одномодової оптики для відстаней до 2 км. Поки немає стандартів ні для 500 м, ні для 2 км, а з «зазором» від 150 м до 10 км бюджет оптичної потужності може бути надлишковим. Стандарти для одномодових ліній довжиною 500 м або 2 км дозволять знизити витрати, енергоспоживання й тепловиділення в порівнянні із традиційними одномодовими лініями, розрахованими на дальність передачі 10 км.

Щоб спроектована інфраструктура ЦОДа була максимально гнучкої, необхідно розглянути безліч факторів. Наприклад, потрібно чи проектувати кожен з ліній, довжина яких перевищує 150 м? Якщо таких ліній багато, чи варто відразу переходити на одномодову інфраструктуру? При наявності планів переходу на одномодове волокно наскільки довго варто використовувати багатомодову інфраструктуру, щоб одержати вигоду від низьких амортизаційних відрахувань і зменшити вартість активного встаткування?

Багато чого залежить від подань власника ЦОДа про зміни в технологіях, стандартах і вартості. Хоча проектні рішення визначаються бюджетом і оцінками перспектив подальшого розвитку, у кожному разі корисно передбачити максимально гнучку фізичну інфраструктуру в рамках заданого бюджету, характеристик об'єкта й поточних потреб.

### 3.2 Розробка структурної схеми

#### Питання планування

У ході планування модернізації ЦОДа необхідно дати відповіді на наступні питання:

- Які швидкості передачі даних потрібні?
- Яка довжина ліній, що прокладаються?
- Який монтажний простір доступно?
- Наскільки важлива гнучкість?

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

Ключовий момент – забезпечення для магістральної підсистеми більше високої пропускну здатності, чим буде потрібно серверам у найближчі два роки. При достатній гнучкості підсистеми міграцію можна безболісно здійснити в доступному для огляду майбутньому завдяки створеній інфраструктурі.

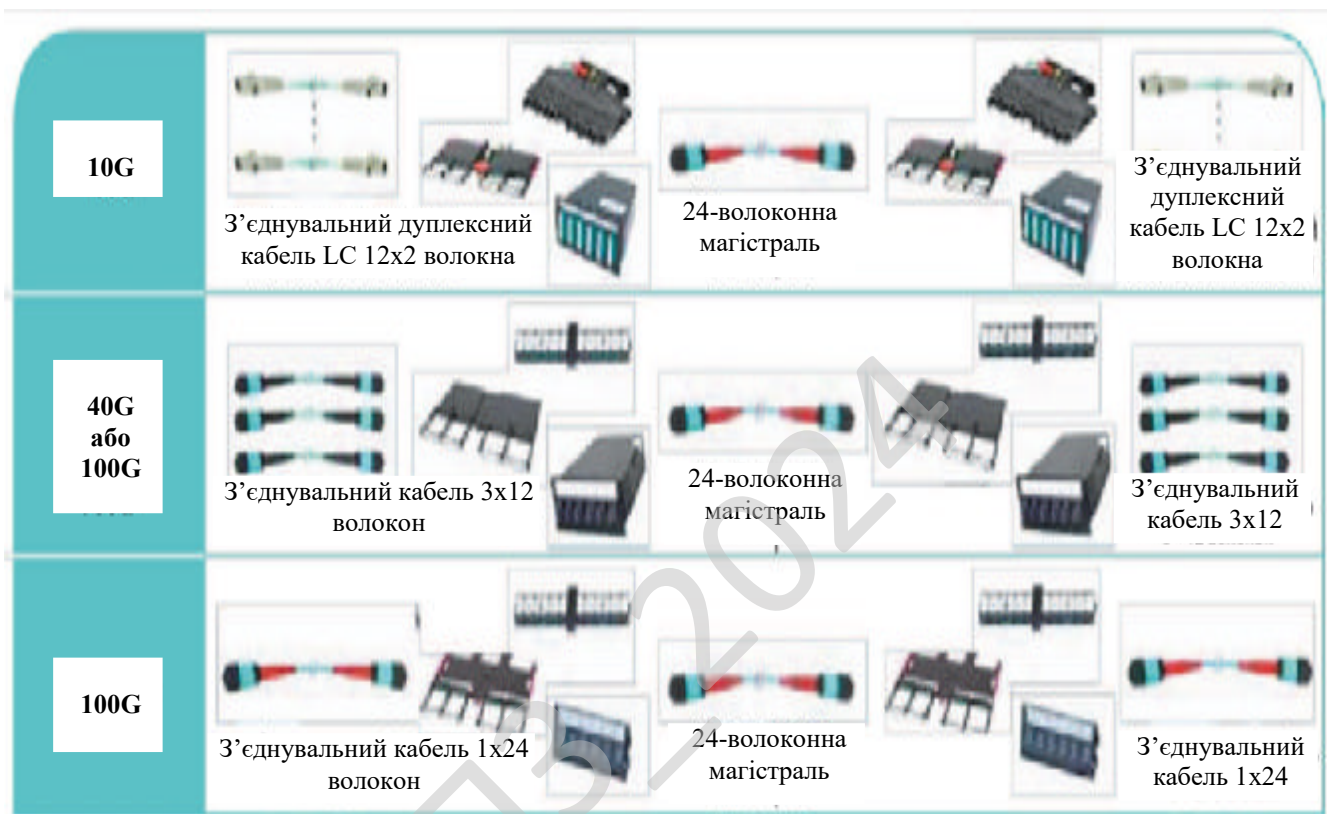


Рисунок 3.1 – Структурна схема системи

На структурній схемі (рисунок 3.1) показаний типовий шлях міграції 10/40/100 Гбіт/с для гнучкої інфраструктури, у рамках якої можуть бути реалізовані прогнозовані сценарії змін. Якщо гнучкість передбачається споконвічно, це може принести згодом значну вигоду, оскільки ніхто не в змозі точно пророчити, що чекає спереду. Безсумнівно, у якийсь момент більшості ЦОДів (особливо провайдерів хмарних сервісів і операторам мегаЦОДів) будуть потрібні більше висока щільність портів і значні швидкості передачі даних, аж до 400 Гбіт/с.

Дальність є вирішальним чинником при визначенні того, який вид волокна буде використовуватися в ЦОДі – чи можна й далі обходитися багатомодовим або варто перейти на одномодове. Багатомодова інфраструктура простіше, дешевше й перевірена часом. Звичайно, якщо довжина з'єднань між більшістю портів перевищує 150 м, для збільшення швидкості передачі даних у майбутньому потрібна одномодова інфраструктура, але чим довше стануть використовуватися багатомодові лінії, тим вигідніше це виявиться в плані амортизації й через імовірне здешевлення одномодового встаткування.

Незважаючи на те що для підтримки 400 Gigabit Ethernet напевно буде використовуватися одномодове волокно, багато аспектів 400Gb до кінця не визначені. Один з можливих підходів – реалізація 16 ліній по 25 Гбіт/с кожна, хоча проектування, розгортання й підтримка такого рішення можуть виявитися непростю справою. Інший варіант – 8 ліній по 50 Гбіт/с кожна, що здійснити значно простіше. Перешкоди серйознішають проблемою при більшій кількості з'єднань, тому для об'єднаних плат мережного встаткування розробляються спеціальні рішення з низьким рівнем шуму.

Більшість операторів сегментують ЦОД, розміщаючи встаткування мережі зберігання даних в одній області, а комутатори й сервери – в іншій. Протягом життєвого циклу ЦОДа зміни середовищ передачі в обох областях стануть відбуватися, швидше за все, у різний час. Чи буде мережа перебудовуватися відразу по одному інтерфейсі, пристрою, шафі, ряду або залу? Незалежно від обраного кванта змін, ключову роль грає планування, а найкращий підхід – забезпечення найбільшої гнучкості в рамках наявного бюджету.

Щільність є ще одним елементом «пазла» міграції. Смуга пропущення сама по собі впливає на щільність. Наприклад, для досягнення більшої щільності портів один порт 40 Gigabit Ethernet може бути зконфігурований як чотири працюючих паралельно порти 10 Гбіт/с, а порт 100 Gigabit Ethernet – як 10 паралельних портів 10 Гбіт/с. Це лише опція, але вона може вплинути на капітальні й операційні витрати розраховуючи на один порт.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

Багато мереж ЦОДів переводяться на архітектуру комутуючої матриці (fabric), що істотно збільшує вимоги до гнучкості з'єднань. Ця міграція вплине на те, як буде конфігуруватися ЦОД. Наприклад, заміна встаткування в одній зоні завжди зажадає відповідних змін в іншій.

Важливо забезпечувати гнучкість у тих областях ЦОДа, де відбуваються переміщення, додавання й зміни й де вплив на міграцію буде найбільшим. Наприклад, якщо згодом буде потрібно міграція на одномодові лінії, кроси й оптичні панелі повинні підтримувати як багатомодові, так і одномодові з'єднання за рахунок модульності.

### **Енергоспоживання**

Необхідність досягнення більше високої швидкості уступає тільки потребі в більшій енергоефективності. Багато хто ЦОДи витрачають стільки енергії, скільки необхідно для висвітлення невеликого міста. Кожна модернізація з метою підтримки більшої швидкості передачі даних припускає підключення додаткової потужності. Побічним результатом енергоспоживання є тепловиділення. Для скорочення першого розробляються оптичні випромінювачі малої потужності. Зниження споживання навіть на декілька ватт на кожний випромінювач приводить до помітної економії енергії при великій кількості малопотужних випромінювачів.

За прогнозами IEEE802.org, кількість портів комутаторів буде подвоюватися кожні півтора року, а щільність портів серверів – кожні два роки. Тому відкладати перехід на волоконну оптику з більше високими швидкостями передачі даних зовсім не вигідно. У порівнянні з існуючими рішеннями новітні оптичні випромінювачі на 25 Гбіт/с й споживають енергії на 60% менше, що може дати істотну економію на системному рівні. Крім того, кожний ватт заощадженої енергії дозволяє додатково знизити експлуатаційні витрати на підтримку інфраструктури.

Виробники активного встаткування прискорено впроваджують оптикові на рівні як введення-виводу, так і внутрішніх з'єднань. Основним завданням

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

стосовно до уведення-виводу є досягнення необхідної щільності для реалізації швидкостей передачі даних наступного покоління. Крім того, рішення повинне бути збалансованим – одночасно забезпечувати мінімально можливу споживану потужність і мінімальну вартість розраховуючи на порт. Оскільки в устаткуванні багато зовнішніх портів, тут фактор вартості набагато важливіше, ніж у випадку внутрішнього з'єднання. Всі частіше волоконна оптика надає найкраще сполучення продуктивності й вартості як для зовнішніх портів, так і для внутрішніх з'єднань в активному встаткуванні.

Іншим ключовим фактором зниження операційних витрат ЦОДів в умовах підвищення вимог до пропускну здатності й швидкості обробки даних є щільність. Кожна одиниця площі ЦОДа має відповідну вартість, тому необхідно одержати максимальну вигоду від кожного юніта вертикального простору шафи. Високоплотні рішення для кросів дають безліч переваг за рахунок збільшення кількості з'єднань на одиницю площі. Термінізовані максимально можливого числа з'єднань між мережним устаткуванням на малій площі надзвичайно важливо, особливо в багатокористувальницьких ЦОДах.

У порівнянні з існуючими багатОВОЛОКОННИМИ системами, розроблювальні 64-волоконні кабельні складання здатні збільшити щільність на 33%, а термін служби встаткування – в 10 разів. Ці інновації допоможуть швидше виконувати інсталяцію, спростять управління й обслуговування. Не можна недооцінювати й кабельні органайзери, адже щільність кабелів збільшується в кожній шафі, коридорі й у ЦОДі в цілому.

Кількість користувачів смартфонів і інших мобільних пристроїв стрімко росте. Ці бездротові пристрої стимулюють соціальні комунікації, підвищують ефективність економічної діяльності й поліпшують якість життя. Але породжувані ними величезні обсяги трафіку змушують операторів адаптувати свої ЦОДи до нових умов.

У результаті збільшується число центрів обробки даних і з'являється необхідність у постійній міграції на технології нового покоління. Оператори

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

ЦОДів повинні справлятися з усе більше складними завданнями, для чого їм необхідні рішення, що забезпечують необхідну гнучкість при внесенні змін і адаптації. Єдиний спосіб підтримки стійкого розвитку ЦОДа складається в досягненні більше високих швидкостей економічно ефективним образом, керованому росту й контролі за енергоспоживанням.

Дуже важливо, щоб ваша мережа була досить гнучкою для адаптації до структури організації, до поточних і майбутніх вимог бізнесу, а виникаючі при цьому технічні ризики щонайкраще співвідносилися б з фінансовими можливостями. У досягненні цих цілей ЦОДи опираються на пропозиції виробників устаткування й компонентів, а також на їхню допомогу по впровадженню інноваційних технологій і перебудові архітектури з метою пошуку найбільш ефективних шляхів міграції на наступне покоління встаткування. Гнучкі інфраструктурні рішення гарантують реалізацію «гладкої» міграції ЦОДа.

### 3.3 Розробка функціональної схеми

Розглянемо існуючі методи підвищення вірогідності експертної оцінки основних технічних параметрів, режимів роботи й характеристик вхідних у поняття стійкості ЦОД на поточному етапі життєвого циклу, а також технічні засоби аналізу ризиків провідних виробників, як прототипи розроблювальних засобів підвищення вірогідності експертної оцінки стійкості, їхнього достоїнства й недоліки, а також аналізується проблематика використання в питаннях експертизи стійкості ЦОД на поточному етапі життєвого циклу.

Розглядається стійкість ЦОД, що:

- поєднує в собі такі категорії, як стійкість, функціональність і інформаційна безпека;
- виробляється аналіз принципової схеми ЦОД із погляду побудови живучого програмно-апаратного комплексу;
- здійснюється вимір основних параметрів функціонування ЦОД;

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

– у складі окремого проекту визначаються значення стійкості ЦОД і зіставляються отримані значення із заданими в ТЗ;

– у складі окремого проекту виробляється побудова моделі загроз і порушників, виробляється аналіз ризиків і їхнє зіставлення з заданим у ТЗ рівнем інформаційної безпеки;

– виробляється розбивка життєвого циклу проекту на послідовність ітерацій у вигляді спіральної (spiral) моделі Боема, що являє собою систему координат, що рухається спіралі по, яка розвертається, що поєднує чотири фази, що повторюються на кожному новому витку часу; здійснюється реалізація моделі на етапі впровадження в промислову експлуатацію шляхом розбивки етапу на п'ять стадій:

– стадія розробки концепції впровадження системи;

– стадія проекту впровадження ЦОД;

– стадія макетування;

– стадія першої версії робочої ЦОД;

– стадія готової ЦОД.

З переходом від стадії до стадії в експертній групі відбувається нагромадження інформації, досвіду й історичних даних; вірогідність експертизи починає зростати, але разом з її зростанням змінюється й стан стійкості ЦОД. Взаємний вплив перерахованих факторів приводить до потреби корекції результатів експертизи.

Для забезпечення експертизи існує множину технічних засобів, таких, як:

– COBRA, розробки компанії C & A Systems Security Ltd.;

– RA Software Tool, засоби розробки компанії RiskWatch.

В основі розглянутих програмних комплексів лежать статистичні й імовірнісні методи, що вимагають великої кількості історичних даних про роботу ЦОД. В умовах неповноти інформації розглянуті засоби виявляються неефективними.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

Узагальнюючи вищесказане, при розгляді завдань оцінки стійкості телекомунікаційних систем і комп'ютерних мереж (ЦОД) в умовах неповноти інформації з урахуванням поточного етапу життєвого циклу необхідно виділити фактори, що роблять на вірогідність експертної оцінки безпосередній вплив:

- неповнота історичних даних вимірів основних параметрів функціонування ЦОД на різних режимах роботи (видах трафіків);
- неповнота історичних даних про стійкість апаратних компонентів і функціональності програмних засобів ЦОД;
- ризики інформаційної безпеки ЦОД;
- особливості поточного етапу життєвого циклу і його стадій.

Для успішного рішення перерахованих вище проблем, представляється доцільним розгляд ряду питань, серед яких можна виділити наступні:

- розгляд існуючих методів оцінки стійкості ЦОД;
- експлікація (роз'яснення) експертних оцінок стійкості ЦОД в умовах неповноти історичних даних на поточному етапі життєвого циклу ЦОД;
- розробка моделі життєвого циклу для умов неповноти інформації;
- розробка моделі забезпечення вірогідності експертизи стійкості ЦОД на поточному етапі життєвого циклу (етапі впровадження в промислову експлуатацію) у вигляді знаходження погрішності експертних оцінок;
- імітаційне моделювання проведення експертизи стійкості ЦОД на поточному етапі життєвого циклу за рахунок застосування моделі, створеної в середовищі, здатної забезпечити процеси збору й аналізу історичних даних, у тому числі про граничні режими роботи ЦОД.

Розробимо метод експлікації експертних оцінок стійкості ЦОД, що реалізує можливість роз'яснення точки зору експертів на об'єкт дослідження й враховуючий розкид їхніх думок. У його основі лежить визначення комплексних показників стійкості телекомунікаційних систем і комп'ютерних мереж (ЦОД) на основі логіко-лінгвістичних методів оцінки, нечітких поліхроматичних множин,

застосовуваних в умовах неповноти інформації, що робить вплив на вірогідність експертної оцінки стійкості ЦОД на поточному етапі життєвого циклу.

Застосування для оперування з невизначеними величинами апарата теорії імовірності приводить до того, що фактично невизначеність, незалежно від її природи, ототожнюється з випадковістю, тим часом як основним джерелом невизначеності в багатьох процесах прийняття рішень є нечіткість (неточність), або розпливчастість (fuzzines).

У питаннях забезпечення стійкості ЦОД виявляються неефективними й іншими кількісними методами прийняття рішень, такі, як максимізація очікуваної корисності, мінімаксна теорія, методи максимальної правдоподібності, теорія ігор, аналіз "витрати – ефективність" і інші. Розглянуті методи допомагають вибрати найкращі з множини можливих рішень лише в умовах одного конкретного виду невизначеності або в умовах повної визначеності.

У зв'язку з вищесказаним пропонується реалізувати метод експлікації експертних оцінок стійкості телекомунікаційних систем і комп'ютерних мереж (ЦОД) на основі методів нечіткої логіки.

На першому етапі на підставі IDEF0 моделі загроз і порушників здійснюється аналіз можливих загроз стійкості й аналізується їхній вплив один на одного.

На другому етапі на підставі кожної сформульованої загрози й внутрішніх факторів формуються вихідні дані, що служать компонентами вектора запиту.

Своєрідною бальною шкалою для формування єдиних думок експертів є таблиця визначення ризику залежно від трьох факторів, що складена на основі статистичних розробок компанії Gartner Group.

Такі таблиці використовуються як в "паперових", так і в програмних методиках оцінки ризиків. В останньому випадку матриця оцінки ризиків поставляється разом із системою оцінки ЦОД і не підлягає зміні.

На третьому етапі виконується ранжирування загроз (вихідних даних) через визначення коефіцієнтів важливості. Експерти вказують свої суб'єктивні

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

оцінки у вигляді чисельних значень від 0 (максимально необхідна, оптимальна, стійкість – точка "обрію") до 1 (повний крах системи).

Результатом ранжирування нечіткої множини коефіцієнтів важливості на множини вихідних даних, може стати функція приналежності (ФП – відображення множини вихідних даних в одиничний відрізок  $[0, 1]$ ).

Для визначення базової (еталонної) функції приналежності (ФП) нечіткої множини коефіцієнтів важливості в магістерській роботі пропонується використовувати метод побудови експонентної ФП (Гаусса –  $\text{gaussmf}$ , в інтерпретації). З аналізу різних джерел, присвячених методам побудови функцій приналежності, розглянута метод доцільніше всього використовувати для рішення завдань виробітку й оцінки альтернатив, а також подання нормально розподілених випадкових величин, таких, як загрози (збурювання) інформаційної безпеки. Відповідно, реакція експертів на розглянуті загрози, також являє собою нормально розподілену випадкову величину, що спрощує подальший перехід до стохастичних методів оцінки.

У магістерській роботі пропонується поняття колір нечіткої множини розглядати як експлікацію (роз'яснення, пояснення) понять:

- властивість;
- атрибут;
- характеристика.

У нечіткій поліхроматичній множини, що моделює складний об'єкт, який володіє різними властивостями, кожному елементу поставимо у відповідність множину персональних кольорів або розфарбувань, що роз'яснюють ці властивості, а множини в цілому – множину унітарних кольорів або розфарбувань самої множини.

На четвертому етапі пропонується, щоб уникнути некоректності нечітких моделей, в описі множини вказувати умови існування персональних кольорів елементів і унітарних кольорів самої множини, а також ураховувати їхній вплив один на одного.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

На підставі аналізу значимості лінгвістичних термов мнемонічне подання бінарної матриці буде показувати наявність фізичного впливу загроз на стан стійкості ЦОД у цілому.

Персональні розфарбування всіх елементів множини описуються матрицею бінарних відносин між елементами й квітами, що представляється у вигляді підмножини декартова добутку.

Ідеальним є випадок, коли всі загрози рівноцінні й не залежать друг від друга, однак для реальних систем існують більше й менш небезпечні й залежні загрози, які, у свою чергу, можуть бути домінуючими й домінуємі.

При позитивному (домінуючому) впливі персональні кольори розглянутого елемента розширюють унітарне розфарбування множини за рахунок появи нових тіл унітарних кольорів, у які входить розглянутий елемент.

При негативному (домінуємому) впливі кольору розглянутого елемента негативно впливають на відповідні персональні кольори інших елементів і на унітарні кольори множини, що приводить до припинення їхнього існування.

При нейтральному впливі умови існування кольорів інших елементів і унітарних кольорів множини залишаються незмінними.

На п'ятому етапі пропонується, на підставі умовиводів, уважати ступенем впливу в унітарному розфарбуванні її персональних складову міру близькості між розфарбуваннями сусідніх елементів поліхроматичної множини. Способи визначення міри близькості засновані на понятті відстані Хеммінгу.

У роботі експертів зручно використовувати відносну відстань Хеммінгу.

Очевидно: чим більше відносна відстань Хеммінгу, тим менше одне розфарбування впливає на сусіднє розфарбування в розглянутому універсумі.

Нечіткий вивід займає центральне місце в нечіткій логіці й системах нечіткого керування. Нечітким логічним виводом називається одержання висновку у вигляді нечіткої множини, що відповідає поточним значенням входів, з використанням нечіткої бази знань і нечітких операцій.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

Оснoву нечіткого лoгічного виводу становить композиційне правило Заді: якщо відомо нечітке відношення  $F$  між вхідною ( $x$ ) і вихідний ( $w$ ) змінними, то при нечіткому значенні вхідний змінної  $x = A$  нечітке значення вихідний змінної визначається як  $w = F_o$ , де  $o$  – максіміма композиція, або функція максимізації мінімуму можливих рішень.

Механізм, або алгоритм, виводу є наступною важливою частиною базової архітектури систем нечіткого виводу. Одним з найбільше широко розповсюджених алгоритмів нечіткого виводу є алгоритм Мамдані (Mamdani). Зазначений алгоритм, у порівнянні з іншими методами нечіткого виводу (Цукамото, Ларсена, Сугено), має економічність алгоритмічної реалізації, щонайкраще застосуємо для нечітких множин, що відповідають термам висновків, що ставляться до тим самим вихідним лінгвістичним змінних.

Подальші експериментальні дослідження застосування методів нечіткого виводу Цукамото, Ларсена, Сугено для рішення завдань експлікації експертних оцінок показали, що різниця отриманих значень не перевищує 15%. З огляду на ту обставину, що механізм нечіткого виводу має фізичний сенс "мір і ваг", у роботі пропонується використовувати саме алгоритм Мамдані з умовою, що зазначений метод необхідно буде застосовувати у всіх подібних випадках.

### **Алгоритм Мамдані (Mamdani)**

На підставі реалізації приведення до чіткості, чітке значення  $w_0$  змінної  $x_i$  визначається, як центр ваги кривої результуючої функції приналежності.

1. Нечіткість (правило 1): перебувають ступені істинності персональних кольорів і унітарних кольорів у розфарбуванні множини для передумов (конкретних значень) кожного правила:

2. Нечіткий вивід (правило 2): перебувають рівні "відсікання" для передумов кожного із правил (з використанням операції МІНІМУМ).

3. Композиція (правило 3): з використанням операції МАКСИМУМ виробляється об'єднання знайдених усічених функцій, що приводить до



Невизначеність (нечіткість) оцінки експертами кожної з загроз і невизначеність експертизи стійкості ЦОД у цілому на етапі впровадження в промислову експлуатацію буде становити.

Розробимо метод підвищення вірогідності експертизи стійкості ЦОД на основі систем дуального керування.

Для систем керування з невизначеністю в застосуванні розглянутої теорії до засобів оцінки стійкості ЦОД основна мета складається у відтворенні впливів, що задають, і фільтрації збурювань, викликаних умовами невизначеності загроз.

В умовах неповноти апріорної інформації стратегія експертизи, що забезпечує гарантоване досягнення мети (необхідного рівня вірогідності) тільки шляхом створення надмірності (уточнення інформації, одержуваної експертом) неефективна, а в багатьох випадках (для досить масивних класів невизначеності) нереалізована.

Природно очікувати, що ціль синтезу засобів експертизи стійкості ЦОД досягається використанням стратегій, що забезпечують бажану вірогідність керування зі зменшенням невизначеності у вихідному описі об'єкта й загроз.

Для спіральної моделі Боема в умовах невизначеності функція кінцевого стану системи (розвитку експертної групи в часі), без обліку стану невизначеності, визначається передатною функцією запізнілої ланки.

Гранично досяжна швидкість збіжності нечітких оцінок для досить масивних класів розподілів загроз обмежена невизначеністю. Ці умови забезпечуються параметричним регулятором, структурна схема якого представлена нижче.

Для розглянутої схеми приймається допущення, що результат оцінки стійкості ЦОД одержують один раз наприкінці кожної стадії. Отже, базові характеристики стійкості, отримані в попередній главі для кожної загрози залежно від етапу життєвого циклу (ЖЦ) будуть змінюватися, становлячи траєкторії експертних оцінок.

Чисельно вірогідність експертизи зручно представити у вигляді виправлення функціонала середніх витрат:

$$\Delta J_T[U_0^x(*)] = \{\Delta J_T[U_1^x(*)], \Delta J_T[U_2^x(*)], \dots, \Delta J_T[U_n^x(*)]\} = \frac{J_T[U_0^x(*)]}{J_n[U_0^x(*)]} \cdot 100\%$$

Виправлення (допуски) до оцінок стійкості перебуває як відношення поточного значення вірогідності керування, до її максимального значення й виражаються у відсотках для кожної стадії поточного етапу життєвого циклу

Розглянемо функціональну схему системи у вигляді імітаційної моделі засобів підвищення вірогідності експертної оцінки стійкості ЦОД. Моделювання дозволяє відслідковувати будь-які зміни зовнішніх деструктивних впливів на стан стійкості ЦОД і оперативно реагувати на виявлені деструктивні впливи. Отримана імітаційна модель може служити електронним шаблоном стійкості ЦОД. У четвертому розділі пропонується програмно-апаратний засіб інтерпретації рівня стійкості ЦОД, що моделює використання системи дуального керування, реалізоване в середовищі.

Вихідними даними для формування функціональної схеми яка представлена на рисунку 3.2, є експертні оцінки, представлені в зручному для уведення виді.

Далі, виконується уведення експертних оцінок у графічному режимі програми:

- у графічному інтерфейсі користувача редактори програми встановлюються позначення змінних, кожне з яких окремо утворить елементи нечіткої поліхроматичної множини загроз;

- викликається редактор функцій приналежності, у якому задаються для кожної із шести вхідних загроз значення параметрів, що відповідають настроювання виконуються й для всіх значень оцінки;

- викликається редактор правил нечіткого виводу, у якому задаються для кожного із шести вхідних параметрів правила нечіткого виводу;

- для одержання результатів розподілу нечіткого виводу викликається програма перегляду правил нечіткого виводу;

- розглянута програма дає можливість переглянути функції приналежності в графічному режимі, але не змінювати їх;
- результати експертизи наочніше всього переглядати у вигляді поверхні нечіткого виводу, так, наприклад, графічно залежності еталонної функцій приналежності загрози відмови доступу від загрози виходу з ладу активного мережного встаткування представлена на рисунку 3.2 у вигляді поверхні нечіткого виводу;
- для відображення був викликаний графічний інтерпретатор нечіткого виводу командою;
- як видно з рисунка, загроза відмови доступу при виході з ладу активного мережного встаткування, по оцінках експертів, найбільшу небезпеку для стійкості ЦОД може представляти в епіцентрі отриманої поверхні.

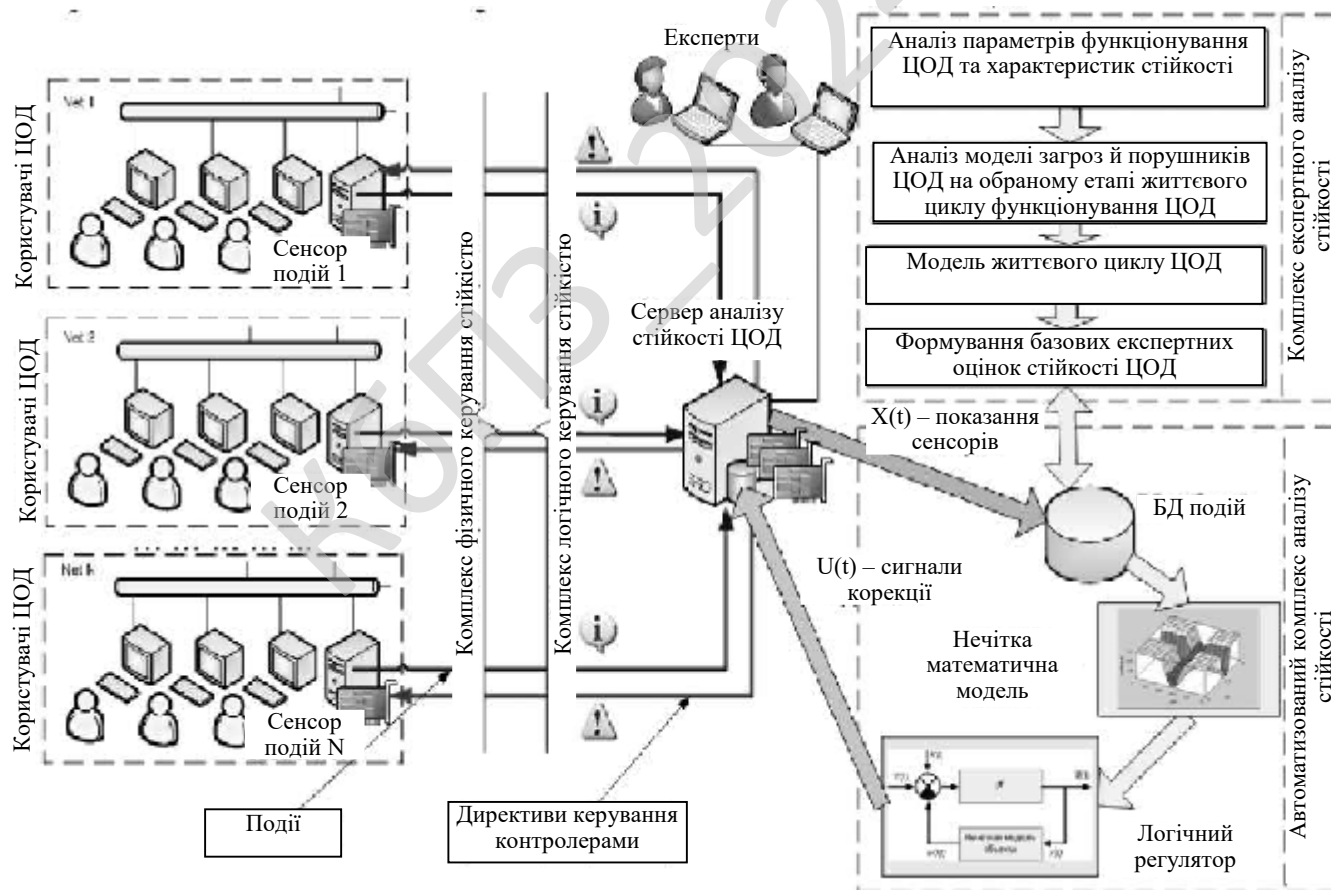


Рисунок 3.2 – Функціональна схема системи

Найменш критичними є загрози виходу з ладу пасивного мережного устаткування й загрози порушення електроживлення. Це обумовлено дублюванням пасивного мережного встаткування й застосуванням джерел безперебійного живлення.

Найбільш критичними, за результатами експлікації експертних оцінок, протягом усього етапу впровадження в промислову експлуатацію залишаються загрози відмови доступу в результаті виходу з ладу активного мережного встаткування. Це пов'язане з високим ступенем ризиків збоїв програмного забезпечення ЦОД у результаті помилок передачі даних.

З аналізу отриманих результатів видно, що вірогідність експертних оцінок зростає від стадії до стадії. Темпи росту вірогідності щодо кожної з загроз приблизно однакові. Однак, вірогідність оцінок кожної з загроз не однакова, тому що різним загрозам експерти із самого початку експертизи приділяли не однакову увагу, а виділяли, на свій розсуд, більш-менш значимі.

Наприклад, ризику порушення конфіденційності, що має високе значення для оцінки стійкості ЦОД, експерти споконвічно приділили меншу увагу, через велику кількість необхідної для аналізу документації. Найменшу увагу експерти приділяли аналізу ризиків загрози порушення електроживлення й загрози виходу з ладу пасивного мережного встаткування, через високий ступінь резервування зазначених компонентів ЦОД. Допуски виміру основних параметрів функціонування ЦОД для зазначених стадій етапу впровадження ЦОД, отримані в результаті імітаційного моделювання представлені для двох стадій етапу впровадження ЦОД. Як показало практичне застосування розроблених засобів підвищення вірогідності експертизи для редактора програми, час на уведення вихідних даних, обробку інформації й аналіз отриманих результатів не перевищує 0,5 значення, відведеного для ухвалення рішення), що скорочує час нагромадження історичних даних на 50%. Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

### 3.4 Розробка діаграми процесів

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування). Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи.

Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється. Діаграма процесів розробленої системи зображена на рисунку 3.3.

При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі.

Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

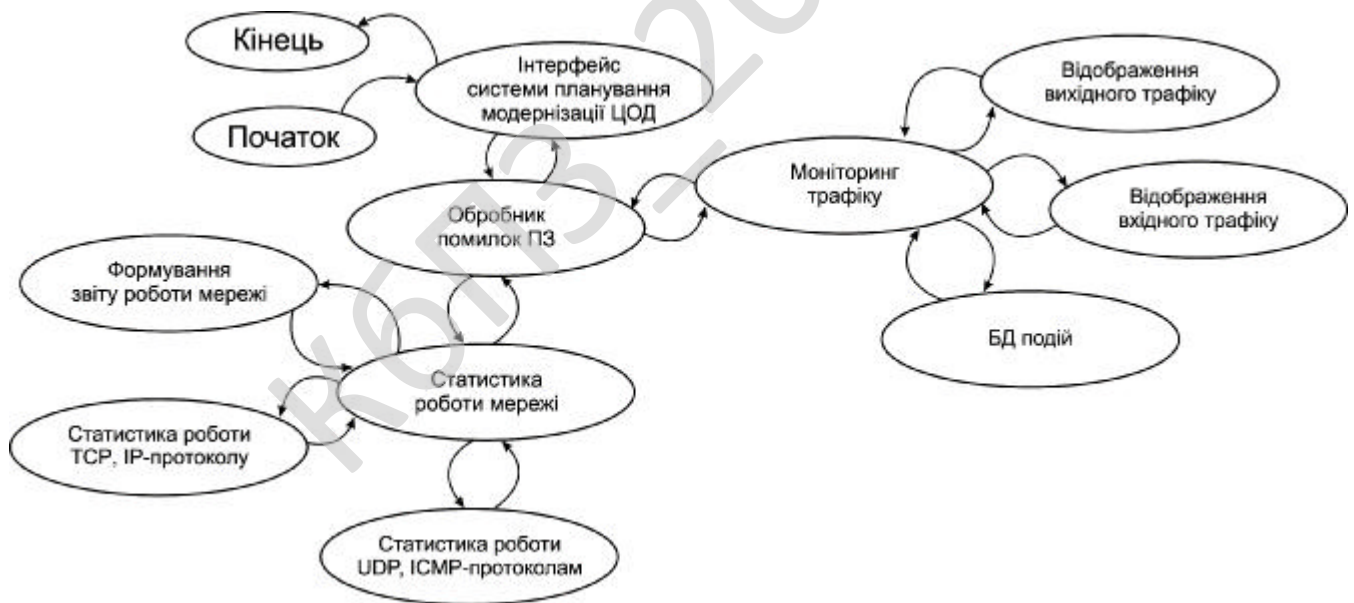


Рисунок 3.3 – Діаграма взаємодії процесів

Діаграми потоків даних містять чотири типи елементів:

– Процеси які являють собою трансформацію даних в рамках описуваної системи.

– Сховища даних (репозиторії).

– Зовнішні по відношенню до системи сутності.

– Потоки даних між елементами трьох попередніх типів.

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

КБПЗ\_2024

					VKPM-123.24.0027.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

## 4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

### 4.1 Блок-схеми та опис алгоритмів функціонування системи

Блок-схеми є основою ПЗ. Тому від точності і детальності проробки блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає оптимізації, також те, що при розробці програми слід надати особливу увагу модулю системи планування модернізації ЦОД.

Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

Функціональні блоки можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірки поточного стану та поверненням на початок схеми чи з завершенням роботи розробленого ПЗ.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

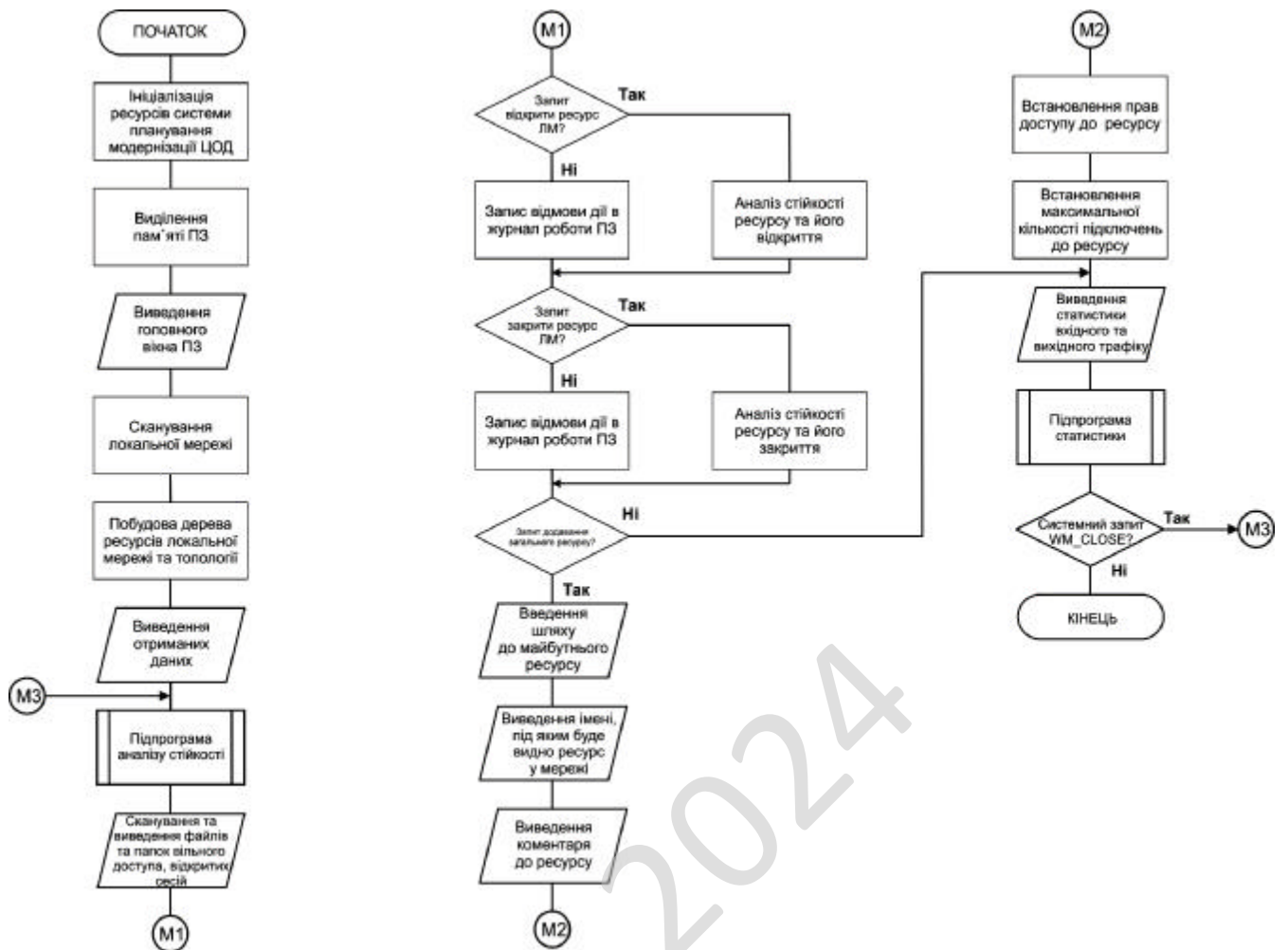


Рисунок 4.1 – Блок-схема основної програми

При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю.

UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем.



програм і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

- набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
- набір клієнтів, які використовують сервіси, що надаються серверами;
- мережа, яка забезпечує взаємодію між клієнтами та серверами.

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

Дуже важливо ясно уявляти, хто або що розглядається як «клієнт». Можна говорити про клієнтський комп'ютер, з якого відбувається звернення до інших комп'ютерів. Можна говорити про клієнтське та серверне програмне забезпечення. Нарешті, можна говорити про людей, які бажають за допомогою відповідного програмного та апаратного забезпечення отримати доступ до тієї чи іншої інформації.

Загальноприйнятим є положення, що клієнти та сервери – це перш за все програмні модулі. Найчастіше вони знаходяться на різних комп'ютерах, але бувають ситуації, коли обидві програми – і клієнтська, і серверна, фізично розміщуються на одній машині; в такій ситуації сервер часто називається локальним.

Модель клієнт-серверної взаємодії визначається перш за все розподілом обов'язків між клієнтом та сервером. Логічно можна відокремити три рівні операцій:

- рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд;

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

- прикладний рівень, який реалізує основну логіку ПЗ і на якому здійснюється необхідна обробка інформації;
- рівень управління даними, який забезпечує зберігання даних та доступ до них.

Дворівнева клієнт-серверна архітектура передбачає взаємодію двох програмних модулів – клієнтського та серверного. В залежності від того, як між ними розподіляються наведені вище функції, розрізняють:

- модель тонкого клієнта, в рамках якої вся логіка ПЗ та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;
- модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта. Товстими клієнтами часто також називають пристрої з обмеженою потужністю: кишенькові комп'ютери, мобільні телефони та ін.

Типовим прикладом клієнт-серверної взаємодії є WWW. Існує величезна кількість веб-серверів, на яких розміщується та чи інша інформація. У найпростішому випадку ця інформація являє собою набір веб-сторінок, які можуть зберігатися на сервері у вигляді файлів, розмічених за допомогою мови розмітки HTML. Але ситуація, як правило, є складнішою; значна частина веб-ресурсів на сучасному етапі є динамічними, тобто вони не існують в заздалегідь підготовленому вигляді, а створюються безпосередньо в процесі обробки запиту від користувача.

Для того, щоб людина, яка працює в Інтернеті, могла переглянути ту чи іншу сторінку, на її комп'ютері повинно бути встановлено відповідне програмне забезпечення. Програми для перегляду веб-сторінок називаються браузерями.

Але, крім браузерів, до серверів можуть звертатися і інші клієнти, а саме – автономні програми. Вони можуть передбачати взаємодію з людиною, а можуть працювати в цілком автоматичному режимі. Типовим класом таких програм є роботи, призначені для автоматичного перегляду веб-ресурсів. Зокрема, роботи є

важливим елементом пошукових систем і використовуються ними для перегляду сторінок і збору інформації про них.

Для запиту до веб-сервера клієнтська програма повинна задати місцезнаходження комп'ютера, на якому розміщується серверна програма, назву потрібного документа і, можливо, інші дані, які специфікують запит. Мережа забезпечує знаходження сервера і передачу йому клієнтського запиту. Серверні програми обробляють цей запит, відповідь пересилається по мережі клієнтові.

Трирівнева клієнт-серверна архітектура, яка почала розвиватися з середини 90-х років, передбачає відділення прикладного рівня від управління даними. Відокремлюється окремий програмний рівень, на якому зосереджується прикладна логіка ПЗ. Програми проміжного рівня можуть функціонувати під управлінням спеціальних серверів ПЗ, але запуск таких програм може здійснюватися і під управлінням звичайного веб-сервера. Нарешті, управління даними здійснюється сервером даних.

Для роботи з системою користувач використовує стандартне програмне забезпечення –звичайний браузер. Це позбавляє його необхідності завантажувати та інсталювати спеціальні програми (хоча інколи така необхідність все-таки виникає).

Але користувачеві слід надати в розпорядженні інтерфейс, який дозволяв би йому взаємодіяти з системою і формувати запити до неї. Форми, що визначають цей інтерфейс, розміщуються на веб-сторінках та завантажуються разом з ними.

Веб-оглядач формує запит та пересилає його до сервера, який здійснює обробку. При необхідності сервер викликає серверні програмні модулі, які забезпечують обробку запиту і в разі потреби звертаються до сервера даних. Сервер даних здійснює операції з даними, що зберігаються в системі та складають її інформаційну основу.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

Зокрема, він може здійснити вибірку з інформаційної бази відповідно до запиту та передати її модулю проміжного рівня для подальшої обробки. Дані, з якими працює сервер даних, найчастіше організовані як реляційна база даних.

Найчастіше веб-сервер і серверні модулі проміжного рівня розміщуються на одному комп'ютері, хоч і являють собою окремі і логічно незалежні програмні модулі.

На сучасному етапі для програмування модулів проміжного рівня використовується мова серверних сценаріїв PHP, а для управління даними – СУБД MySQL. Таким чином, зв'язку PHP-MySQL слід розглядати як стандартний інструмент для створення порівняно простих інтерактивних веб-сайтів та систем електронної комерції; близько 90% комерційних систем сьогодні створюється саме на цій основі. Водночас як засоби управління даними, так і middleware-засоби можуть бути найрізноманітнішими. Так, для створення серверних програм, крім PHP, широко застосовуються Java, Perl, Python, Delphi.

Взагалі, технології створення розподілених, зокрема веб-програм, стрімко розвиваються. Слід згадати про технології EJB (Enterprise Java Beans), CORBA, а також про .NET – порівняно нову ініціативу компанії Microsoft. Для зберігання даних та їх передачі часто використовується так звана розширювана мова розмітки XML (Extensible Markup Language).

При розробці ПЗ було використано підходи ризик-менеджменту – це система управління ризиками, яка включає в себе стратегію та тактику управління, направлені на досягнення основних цілей. Ефективний ризик-менеджмент включає:

- систему управління;
- систему ідентифікації і вимірювання;
- систему супроводження (моніторингу та контролю).

Сучасна наука представляє ризик як вірогідну подію, в результаті настання якої можуть відбутися позитивні, нейтральні або негативні наслідки. Якщо ризик припускає наявність як позитивних, так і негативних результатів, він відноситься

до спекулятивних ризиків. Якщо ж наслідки негативні, або відсутні взагалі, такий ризик іменується чистим.

Мета ризик-менеджменту – підвищення конкурентоспроможності господарюючих суб'єктів за допомогою захисту від реалізації чистих ризиків.

Теорія ризик-менеджменту ґрунтується на трьох базових поняттях: корисності, регресії і диверсифікації.

У 1738 швейцарський математик Даніель Бернуллі доповнив теорію вірогідності методом корисності або привабливості того або іншого результату подій. Ідея Бернуллі полягала в тому, що в процесі ухвалення рішення люди приділяють більше уваги розміру наслідків різних результатів, ніж їх вірогідність.

В кінці XIX століття англійський дослідник Ф. Гальтон запропонував вважати регресію або повернення до середнього значення універсальною статистичною закономірністю. Суть регресії трактувалася ним як повернення явищ до норми з часом. Згодом було доведено, що правило регресії діє в найрізноманітніших ситуаціях, починаючи з азартних ігор та розрахунку вірогідності виникнення нещасних випадків, і закінчуючи прогнозуванням коливань економічних циклів.

У 1952 аспірант Університету Чикаго Гарі Марковіц в статті «Диверсифікація вкладень» («Portfolio Selection») математично обґрунтував стратегію диверсифікації інвестиційного портфеля, зокрема, він показав, як шляхом продуманого розподілу вкладень мінімізувати відхилення прибутковості від очікуваного показника. У 1990 Г. Марковіцу присуджена Нобелівська премія за розробку теорії і практики оптимізації портфеля фондових активів.

### **Етапи ризик-менеджменту**

У ризик-менеджменті прийнято виділяти декілька ключових етапів:

– на першому етапі відбувається виявлення ризику з супутньою оцінкою вірогідності його реалізації і масштабу наслідків;

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

– на другому етапі здійснюється розробка ризик-стратегії з метою зниження вірогідності реалізації ризику і мінімізації можливих негативних наслідків;

– на третьому етапі вибираються методи і інструменти управління виявленим ризиком;

– на четвертому етапі проводиться безпосереднє управління ризиком;

– на завершальному етапі оцінюються досягнуті результати і коректується ризик-стратегія.

За ключовий етап ризик-менеджменту вважається етап вибору методів і інструментів управління ризиком.

### **Методи і інструментарій ризик-менеджменту**

Базовими методами ризик-менеджменту є відмова від ризиків, зниження, передача і ухвалення.

Ризик-інструментарій значно ширший. Він включає політичні, організаційні, правові, економічні, соціальні інструменти, причому ризик-менеджмент як система допускає можливість одночасного застосування декількох методів і інструментів ризик-управління.

Найбільш часто вживаним інструментом ризик-менеджменту є страхування. Страхування припускає передачу відповідальності за відшкодування передбачуваного збитку сторонній організації (страхової компанії).

Прикладами інших інструментів можуть бути відмова від надмірно ризикової діяльності (метод відмови), профілактика або диверсифікація (метод зниження), аутсорсинг витратних ризикових функцій (метод передачі), формування резервів або запасів (метод ухвалення).

Під час роботи над програмою також використовувались підходи RUP.

**Rational Unified Process(RUP)** є ітеративним процесом розробки програмного забезпечення створеним Rational Software – підрозділом IBM з 2003.

RUP не є єдиним, конкретним розпорядчим процесом, а скоріше фреймворком процесу, що має бути адаптованим організаціями які займаються

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

розробкою та командами розробників які оберуть елементи процесу, які підходять під їх потреби.

Rational Unified Process (RUP) являє собою продукт, спочатку розроблений Rational Software, яка була придбана компанією IBM в лютому 2003 року. Продукт містить у собі базу знань з гіперпосиланнями, та прикладами артефактів і докладні описи для різних видів діяльності. RUP входить в продукт IBM Rational Method Composer (RMC), який дозволяє налаштування процесу.

До 1997 року, Rational придбав Verdix, Objectory, Requisite, SQA, Performance Awareness, та Pure–Atria. Поєднання баз досвіду цих компаній привело до вироблення семи «найкращих практик» сучасної програмної інженерії:

1. Розробляти ітеративно, керуючись ризиками.
2. Управляти вимогами.
3. Використовувати компонентну архітектуру.
4. Моделювати програмне забезпечення візуально.
5. Постійно перевіряти якість.
6. Контролювати зміни.
7. Підлаштовуватись.

Ці найкращі практики рухали розробку продуктів Rational, та використовувались польовими командами Rational, щоб допомогти клієнтам вдосконалити якість, та передбачуваність їх розробницьких зусиль. Щоб зробити ці знання доступнішими, Філіпу Крачтену, було поставлено завдання збирати явні фреймворки сучасної розробки програмного забезпечення. Ці зусилля використовував заснований на HTML механізм доставки процесів розроблений Objectory.

У результаті «Раціональний уніфікований процес» (RUP) завершив стратегічну опору для Rational:

- Адаптований процес що направляє розробку
- Інструменти, що автоматизують використання цього процесу

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

– Сервіси, що прискорюють впровадження і процесу, і інструментів.

## Будівельні блоки RUP

RUP заснований на наборі будівельних блоків, чи містить елементи, що описують те, що повинно бути зробленим, необхідні навички, та покрокове пояснення того, як досягаються конкретні цілі розробки. Основними будівельними блоками, чи елементами вмісту, є наступні:

– Ролі (хто). Роль визначає набір навичок, компетенції та відповідальності.

– Робочі продукти (що). Робочий продукт являє собою щось отримане з завдання, в тому числі всі документи і моделі, випущені під час роботи впродовж процесу.

– Завдання (як). Завдання описує одиницю роботи, яка доручена ролі, яка забезпечує значущий результат.

У кожній ітерації, завдання діляться на дев'ять дисциплін: шість «інженерних дисциплін» (бізнес-моделювання, вимоги, аналіз і проектування, реалізація, тестування, розгортання) і трьох допоміжних дисциплін (конфігурація і керування змінами, управління проектами, середовища).

## Чотири фази життєвого циклу проекту

RUP визначає життєвий цикл проекту, що складається з чотирьох фаз. Ці фази дозволяють процесу, бути представленим на високому рівні, подібно до того як представляються проекти у «водоспадному» стилі, хоча, по суті, ключем до процесу є ітерації розробки, які простягаються вздовж всіх фаз. Крім того, кожен етап має одну ключову ціль, та віху в кінці, яка позначає досягнення цілі.

### Початкова фаза

Первинною ціллю є адекватна оцінка системи, як база для обчислення початкових розцінок та бюджету. На цьому етапі встановлюються бізнес випадки, які включають бізнес-контекст, фактори успіху (очікувані доходи, визнання на ринку, і т.д.), а також фінансовий прогноз. На додаток до бізнес випадку генерується базова модель прецедентів, план проекту, попередня оцінка ризику і опис проекту (основні вимоги до проекту, обмеження та основні характеристики).

					ВКРМ-123.24.0027.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

Після їх завершення проект перевіряється на відповідність наступним критеріям:

– Зацікавленими сторонами досягають згоди з визначення масштабів і оцінкою вартості/термінів.

– Розуміння вимог як свідчення якості первинних прецедентів.

– Достовірність оцінок вартості/термінів, пріоритетів, ризиків, та процесу розробки.

– Глибина і ширина будь-якого архітектурного прототипу, який був розроблений.

– Встановлення базової лінії за допомогою якої можна порівняти фактичні витрати в порівнянні із запланованим витратам.

Якщо проект не пройде цей етап, що називається віхою життєвого циклу, він може бути як скасований так і повторений після переконструювання з метою кращого задоволення критеріїв.

#### **Фаза уточнення**

Основна мета полягає в пом'якшенні ключових ризиків, виявлених на основі аналізу до кінця цієї фази. Фаза уточнення – фаза де проект починає набувати форми. На цьому етапі робиться аналіз предметної області, і архітектура проекту отримує свою базову форму.

Ця фаза має пройти віху життєвого циклу архітектури (LCA), задовольняючи такі критерії:

– Модель прецедентів, в якій ідентифікуються прецеденти та актори, та розробляється більшість описів прецедентів. Модель прецедентів повинна бути завершена на 80%.

– Опис архітектури програмного забезпечення в процесі розробки програмної системи.

– Виконувана архітектура, яка реалізує архітектурно значимі прецеденти.

– Бізнес це випадки та список ризиків що переглядаються.

– План розвитку проекту в цілому.

					ВКРМ-123.24.0027.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

– Прототипи, що явно зменшили кожен виявлений технічний ризик.

Якщо проект не може переступити цю віху, ще є час для того, щоб він був скасований або змінений. Тим не менше, після закінчення цього етапу, проект переходить в операцію з високим ступенем ризику, де зміни набагато складніші та згубні, при здійсненні.

Системна архітектура є ключовим елементом розробки, що отримується з аналізу предметної області.

### **Фаза конструювання**

Основна мета полягає в створенні програмної системи. На цьому етапі основна увага приділяється розробці компонентів та інших характеристик системи. Це етап, коли відбувається основна частина кодування. У більш великих проектах, може бути кілька фаз конструювання, в спробі поділити прецеденти на керовані сегменти, які можуть утворити презентабельні прототипи.

Цей етап створює перший реліз програмного забезпечення. Його завершення позначає віха початкової боєготовності.

### **Фаза впровадження**

Основна мета полягає в переведенні системи з розробки у продукт, зробивши її доступною та зрозумілою для кінцевого споживача. Діяльність у рамках цієї фази включає навчання кінцевих користувачів та обслуговуючого персоналу, бета-тестування системи для перевірки її на відповідність очікуванням користувачів. Продукт також перевіряється на відповідність рівню якості, встановленого в початковій фазі.

Якщо всі вимоги задоволені, досягається віха релізу продукту, і цикл розробки завершується.

## **Шість інженерних дисциплін**

### **Дисципліни бізнес-моделювання**

Бізнес-моделювання пояснює, як описати бачення організації, в якій буде розгортатись система і як використати це бачення для виділення процесу, ролей та обов'язків.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

Організації стають все залежнішими від ІТ систем, що вимагає від інженерів інформаційних систем знання того, як застосунок що вони розробляють вписується в організацію. Підприємства інвестують в ІТ, коли вони розуміють, конкурентні переваги і вартість що додає технологія. Метою бізнес-моделювання є по-перше встановити глибше розуміння та комунікаційний канал між бізнес інженерією та програмною інженерією. Розуміння бізнесу означає, що програмісти повинні розуміти структуру і динаміку цільової організації (клієнта), нинішні проблеми в організації, а також можливі удосконалення. Вони повинні також забезпечити загальне розуміння цільової організації між клієнтами, кінцевими користувачами та розробниками.

### **Дисципліни вимог**

Вимоги пояснюють, як виявити запити зацікавлених осіб і перетворити їх в набір вимог, робочих продуктів, що досягають створювану систему й надають детальні вимоги до того, що система повинна робити.

### **Дисципліна аналізу та проектування**

Метою аналізу і проектування, є показати, яким чином система буде реалізована. Ціллю є створення системи, яка:

- Виконує – в особливому середовищі реалізації – задачі та функції описані в описах прецедентів.
- Виконує всі свої вимоги.
- Легко змінити, коли змінюються функціональні вимоги.

Проектування дає в результаті модель проектування, а аналіз відповідно модель аналізу. Модель дизайну служить абстракцією вихідного коду; тобто модель дизайну працює «синьою», розміткою того як буде структурований та написаний вихідний код. Дизайн моделі складається проектування класів структурованих в пакети і підсистеми з чітко визначеними інтерфейсами, які представляють, що стане компонентами у реалізації.

Він також містить опис того, як об'єкти цих сконструйованих класів співпрацюють для виконання прецедентів.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

## Дисципліна реалізації

Метою реалізації є:

- Визначити організацію коду з точки зору реалізації підсистем, які організовані в шари.
- Реалізація класів та об'єктів у термінах компонентів (вихідних файлів, виконуваних файлів, та інших).
- Для тестування розроблених компонент та модулів.
- Для інтеграції результатів, отриманих окремими виконавцями (чи групами) у виконувану систему.

Системи реалізуються через реалізацію компонентів. Процес описує як повторно використати існуючі компоненти, чи реалізувати нові компонентни з чітко визначеними відповідальностями, роблячи систему легше підтримуваною і збільшуючи можливості для повторного використання.

## Дисципліна тестування

Цілі тестування:

- Перевірити взаємодії між об'єктами.
- Перевірити належну інтеграцію всіх компонентів програмного забезпечення.
- Щоб переконатися, що всі вимоги були правильно виконані.
- Щоб визначити та переконатись що дефекти будуть розглянуті до розгортання програмного забезпечення.
- Переконатись, що всі дефекти виправлені, повторно перевірені та закриті.

Раціональний уніфікований процес пропонує ітеративний підхід, а це означає, що тестування відбувається протягом всього проекту. Це дозволяє виявляти дефекти якомога раніше, що радикально знижує вартість виправлення дефекту. Тести проводяться за чотирма вимірами якості: надійності, функціональності, продуктивності додатків і продуктивності системи. Для

					ВКРМ-123.24.0027.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

кожного з цих вимірів критеріїв якості, процес описує як пройти життєвий цикл планування, проектування, виконання і оцінки тесту.

### **Дисципліна розгортання**

Метою розгортання є успішно робити релізи продукту, та постачати програмне забезпечення для кінцевих користувачів. Вона охоплює широке коло заходів, у тому числі виробництво зовнішніх версій програмного забезпечення, запаковування програмного забезпечення та бізнес-додатків, розповсюдження програмного забезпечення, встановлення програмного забезпечення та надання допомоги і підтримки для користувачів.

Хоча діяльність по впровадженню в основному зосереджена на перехідному етапі, багато які з цих заходів повинні бути включені в більш ранні етапи, щоб підготуватись до розгортання в кінці фази конструювання. Процеси розгортання та середовища із RUP містять менше деталей, ніж інші робочі процеси.

### **Шість найкращих практик**

Шість найкращих практик як описані в RUP є парадигмою програмної інженерії, яка перераховує шість ідей яким варто слідувати при конструюванні будь-якого проекту щоб мінімізувати провали, та збільшити продуктивність. Цими практиками є:

1. Ітеративна розробка. Найкраще було б знати всі вимоги наперед; тим не менш, часто це не той випадок. Існує кілька процесів розробки програмного забезпечення, які мають справу з рішеннями які дозволяють зменшувати вартість в термінах фаз розробки.
2. Управління вимогами. Завжди пам'ятати вимоги встановлені користувачами.
3. Використання компонент.
4. Розбиття складного проекту не тільки пропонується, а є фактично неминучим. Це дає можливість тестувати окремі компоненти до того, як вони

						<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			82

будуть інтегровані в більшу систему. Також, повторне використання коду є великим плюсом та може бути здійснене легше через використання ООП.

5. Візуальне моделювання. Використовуйте діаграми щоб представити всі основні компоненти, користувачів, та їх взаємодії. «UML», скорочено від Unified Modeling Language, є інструментом що може зробити це завдання більш здійсненим.

6. Перевірка якості. Завжди робіть тестування більшої частини проекту в будь-який момент часу. Тестування стає важчим з розростанням проекту, та воно має бути постійним фактором в будь-якому створенні програмного продукту.

7. Контроль змін. Багато проектів створюються багатьма командами, іноді з різним місцезнаходженням, використовуючи різні платформи, і т.п. Як результат є важливим переконатись, що зміни які вносяться в систему синхронізуються та перевіряються постійно.

#### 4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм ДСТУ 8845:2019 – алгоритм симетричного потокового перетворення. В основі ДСТУ 8845:2019 лежить класична схема підсумовуючого генератора подібна генератору SNOW 2.0, SNOW 3.0 та SNOW V. В основі ДСТУ 8845:2019 збережені всі базові операції шифрів сімейства SNOW, а раунд шифрування AES замінений на функцію нелінійної підстановки T, що реалізовує перестановку елементів скінченного поля  $GF(2^{64})$  за допомогою компонентів національного стандарту симетричного криптоперетворення ДСТУ 7624:2014.

ДСТУ 8845:2019 використовує 256-бітний вектор ініціалізації IV та 256-бітний або 512-бітний секретний ключ K і забезпечує високий та надвисокий рівень стійкості із врахуванням можливого застосування квантового криптографічного аналізу. При розробці алгоритму ДСТУ 8845:2019 орієнтувалися на сучасні 64-бітні обчислювальні системи, тому розмір слова

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

обрано рівним 8 байт. запису байтів застосовують подання від старшого до молодшого. Генератор ключових потоків ДСТУ 8845:2019 у режимі генерації гами шифру схематично приведено у стандарті.

Як впливає з генератора ключових потоків ДСТУ 8845:2019 у режимі генерації гами шифру основними компонентами генератору є регістр зсуву з лінійним зворотнім зв'язком та скінчений автомат на базі якого виконується нелінійне перетворення  $T$ . Вхідні дані (ключ шифрування  $K$  та вектор ініціалізації  $IV$ ) використовуються для ініціалізації змінної стану  $S_i (i \geq 0)$ , яка складається із двох компонент до складу яких входять [12]:

- 16 змінних  $s^{(i)}$  – комірок регістра зсуву з лінійним зворотнім зв'язком:  $s^{(i)} = (s_{15}^{(i)}, s_{14}^{(i)}, s_{13}^{(i)}, s_{12}^{(i)}, s_{11}^{(i)}, s_{10}^{(i)}, s_9^{(i)}, s_8^{(i)}, s_7^{(i)}, s_6^{(i)}, s_5^{(i)}, s_4^{(i)}, s_3^{(i)}, s_2^{(i)}, s_1^{(i)}, s_0^{(i)})$ ;
- Два регістри скінченного автомату  $r^{(i)} : r^{(i)} = (r_2^{(i)}, r_1^{(i)})$ . На виході отримуємо ключовий потік (гамма), який формується з 8-байтних слів  $Z_i$ .

З рисунку слідує, що відводи регістра зсуву з лінійним оберненим зв'язком побудовані за примітивним над полем  $GF(2^{64})$  поліномом:  $f(x) = x^{16} + x^{13} + \alpha^{-1}x^{11} + \alpha$ , де  $\alpha$  є коренем примітивного над полем  $GF(2^8)$  поліному  $g(z) = z^8 + \beta^{170}z^7 + \beta^{166}z^6 + \beta^2 z^5 + \beta^{224} z^4 + \beta^{70}z^3 + \beta^2$ . Поле  $GF(2^8)$  як і в ДСТУ 7624:2014 побудовано за примітивним на полем  $GF(2)$  поліномом  $p(y) = x^8 + x^4 + x^3 + x^2 + 1$ , а коефіцієнти  $g(z)$  подаються через ступінь примітивного елементу  $\beta$  поля  $GF(2^8)$ , тобто  $\beta$  корінь поліному  $p(y)$ . Тобто, у нас є вежа полів:  $GF(2) \subset GF(2^8) \subset GF(2^{64}) \subset GF(2^{1024})$ , де:

- поле  $GF(2^{1024})$  задається відводами зворотного зв'язку як фактор кільце  $GF(2^{64})[x]/(f(x))$ ;
- поле  $GF(2^{16424})$  задається як фактор кільце  $GF(2^8)[z]/(g(z))$ ;
- поле  $GF(2^{1024})$  задається як фактор кільце  $GF(2)[y]/(p(y))$ .

З вищезазначеного, слідує що період вихідної послідовності становить  $2^{1024}$ .

Структурно в алгоритмі симетричного потокового перетворення ДСТУ 8845:2019 виділяють три основні функції:

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>84</b>

– функція ініціалізації, яка приймає в якості вхідних даних 256-бітний вектор ініціалізації IV та 256-бітний або 512-бітний секретний ключ K, і виробляє початкове значення змінної стану  $S_0 = (s^{(0)}, r^{(0)})$ ;

– функція наступного стану Next, яка приймає на вхід зміну стану  $S_i = (s^{(i)}, r^{(i)})$  та виробляє наступне значення змінної стану  $S_{i+1} = (s^{(i+1)}, r^{(i+1)})$ ;

– функція ключового потоку Strm, що приймає на вході змінну стану  $S_i = (s^{(i)}, r^{(i)})$  та виробляє на виході 64-бітний ключовий потік  $Z^i$ .

Також функція Next може виконуватися в двох режимах, в залежності від способу виконання ітерації, як частина реалізації ініціалізації алгоритму ДСТУ 8845:2019 або як частина функції ключового потоку Strm.

КБПЗ\_2024

					ВКРМ-123.24.0027.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

## 5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ системи планування модернізації ЦОД яке зображено на рисунку 5.1. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

- Навігаційне меню: Система; Налаштування; Довідка.
- Розділу виведення результату роботи системи.
- Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.
- Функціональних кнопок ПЗ.

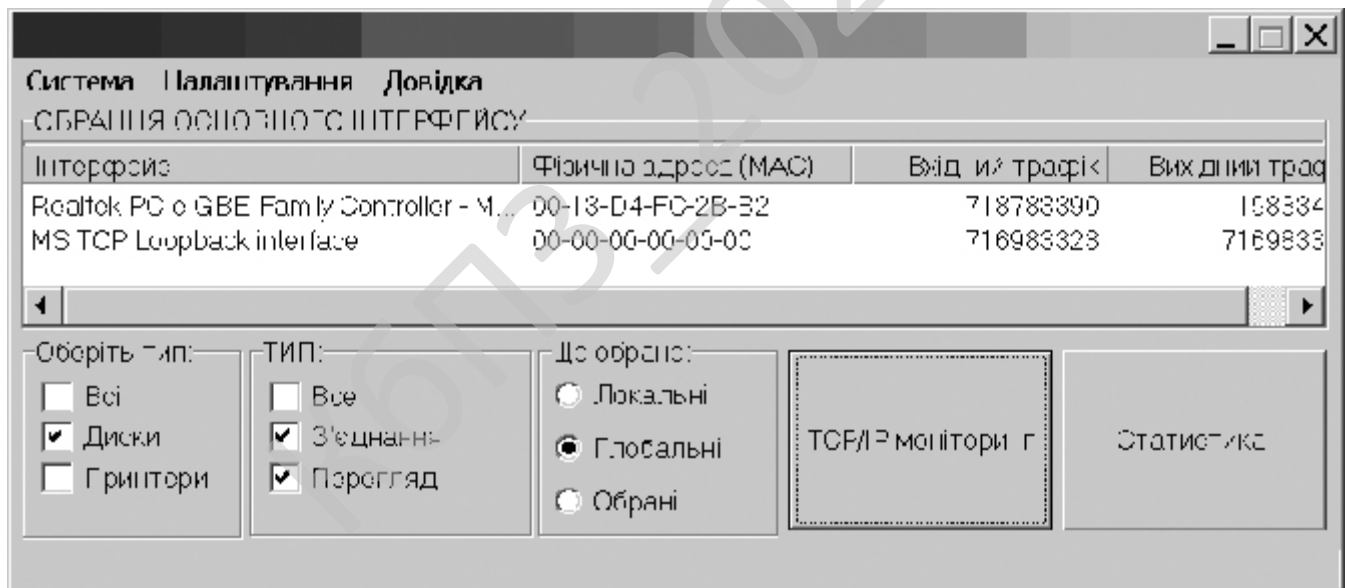


Рисунок 5.1 – Головне вікно ПЗ

Розроблена програма має дуже простий і інтуїтивно зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий.

Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

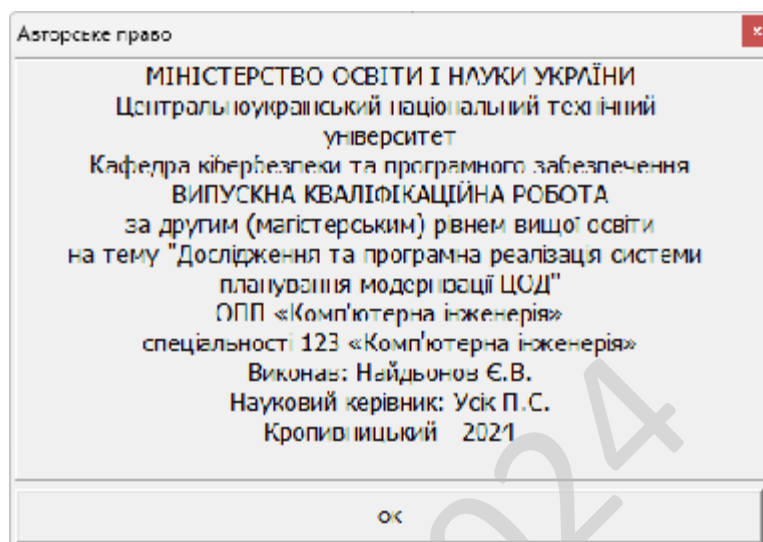


Рисунок 5.2 – Авторське право

Розглянемо процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частиною життєвого циклу програмного забезпечення.

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника так і з боку споживача. Оскільки кожна програмна система є унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, "розгортання" можна трактувати як загальний процес відповідно до певних вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

						<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			87

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

- Випуск.
- Встановлення та активація.
- Деактивація.
- Адаптація.
- Оновлення.
- Вмонтування.
- Відстежування версій.
- Видалення.
- Вилучення з обігу.

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

– Виділення критичних, з точки зору загального результату, процедур в діяльності організації. Коли набір таких процедур визначений, необхідно в першу чергу використовувати ІТ рішення для автоматизації операцій усередині саме цих процедур. Таким чином, розроблене ІТ рішення автоматично стає життєво важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження;

– Розширення нормативної бази організації шляхом включення до неї регламентів, що описують порядок виконання процедур автоматизованих процесів. В іншому випадку є небезпека виникнення неузгодженості між автоматизованими процедурами та іншими процесами організації.

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються в ІТ рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

- відповідності поставленим вимогам;
- правильна відповідь для усіх можливих вхідних даних;
- виконання функцій за прийнятний час;
- практичність;
- сумісність з ОС та стороннім ПЗ.

Оскільки число можливих тестів для програмних компонент практично нескінченне, тому стратегія тестування полягала в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів.

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів.

Проводилось тестування чорної скриньки.

Основне місце програми тестів «чорної скриньки» – інтерфейс ПЗ. Відомі: функції програми. Досліджується: робота кожної функції на всій області визначення.

Ці тести демонструють:

- Як виконуються функції програми.
- Як приймаються вихідні дані.
- Як виробляються результати.
- Як зберігається цілісність зовнішньої інформації.

При тестуванні «чорної скриньки» розглядаються системні характеристики програм, ігнорується їхня внутрішня логічна структура. Вичерпне тестування, як правило, неможливе.

Наприклад, якщо в програмі 10 вхідних величин і кожна приймає по 10 значень, то кількість тестових варіантів становитиме  $10^{10}$ . Тестування «чорної скриньки» не реагує на багато особливостей програмних помилок.

Тестування «чорної скриньки» (функціональне тестування) дозволяє отримати комбінації вхідних даних, які забезпечують повну перевірку всіх функціональних вимог до програми.

Програмний виріб тут розглядається як «чорна скринька», чію поведінку можна визначити тільки дослідженням його входів та відповідних виходів. При такому підході бажано мати:

– Набір, утворений такими вхідними даними, які призводять до аномалій у поведінці програми (назвемо його ІТс).

– Набір, утворений такими вхідними даними, які демонструють дефекти програми (назвемо його ОТ).

Будь-який спосіб тестування «чорної скриньки» повинен:

– Виявити такі вхідні дані, які з високою ймовірністю належать набору ІТс;  
– Сформулювати такі очікувані результати, які з високою ймовірністю є елементами набору ОТ.

Принцип «чорної скриньки» не альтернативний принципу «білої скриньки». Скоріше це доповнює підхід, який виявляє інший клас помилок.

Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

– Некоректних чи відсутніх функцій.  
– Помилки інтерфейсу.  
– Помилки у зовнішніх структурах даних або в доступі до зовнішньої бази даних.

– Помилки характеристик (необхідна ємність пам'яті і т.д.).

– Помилки ініціалізації та завершення.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

Обрано умови розповсюдження – Freeware.

Це власницьке програмне забезпечення, котре можна Безоплатно використовувати протягом необмеженого терміну без обмежень у функціональності, і поширюване без сирцевих кодів.

Автори такого програмного забезпечення, як правило, хочуть «дати щось спільноті», але хочуть також контролювати його подальшу розробку. Іноді, коли програмісти вирішують припинити розробку, вони передають сирцевий код іншим програмістам, або ж спільноті як вільне програмне забезпечення.

Дуже часто плутають поняття «безплатне програмне забезпечення» та «вільне програмне забезпечення», хоча вони суттєво відрізняються.

Безплатне програмне забезпечення можна безоплатно встановлювати та використовувати (іноді з певними обмеженнями, як, наприклад, «безплатне для домашнього або некомерційного вжитку»), в той час як вільне програмне забезпечення можна продавати за будь-яку суму, але при тому, у користувача, котрий його отримує, повинні бути права на вивчення, модифікацію та поширення сирцевих кодів одержаної програми.

КБПЗ-2024

					VKPM-123.24.0027.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

## 6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи планування модернізації ЦОД.

*Метою розробки є дослідження та програмна реалізація системи планування модернізації ЦОД.*

*Об'єктом дослідження є процес планування модернізації ЦОД.*

*Предметом дослідження є методи планування модернізації ЦОД.*

*Методи дослідження базуються на методах комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.*

**Наукова новизна отриманих результатів.** У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- Удосконалено метод планування модернізації ЦОД.
- Розроблено вітчизняний продукт планування модернізації ЦОД, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-123.24.0027.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

## 7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

### 7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та програмної реалізації системи планування модернізації центрів обробки даних (ЦОД) можуть бути цікаві представленим на рисунку 7.1 споживачам.

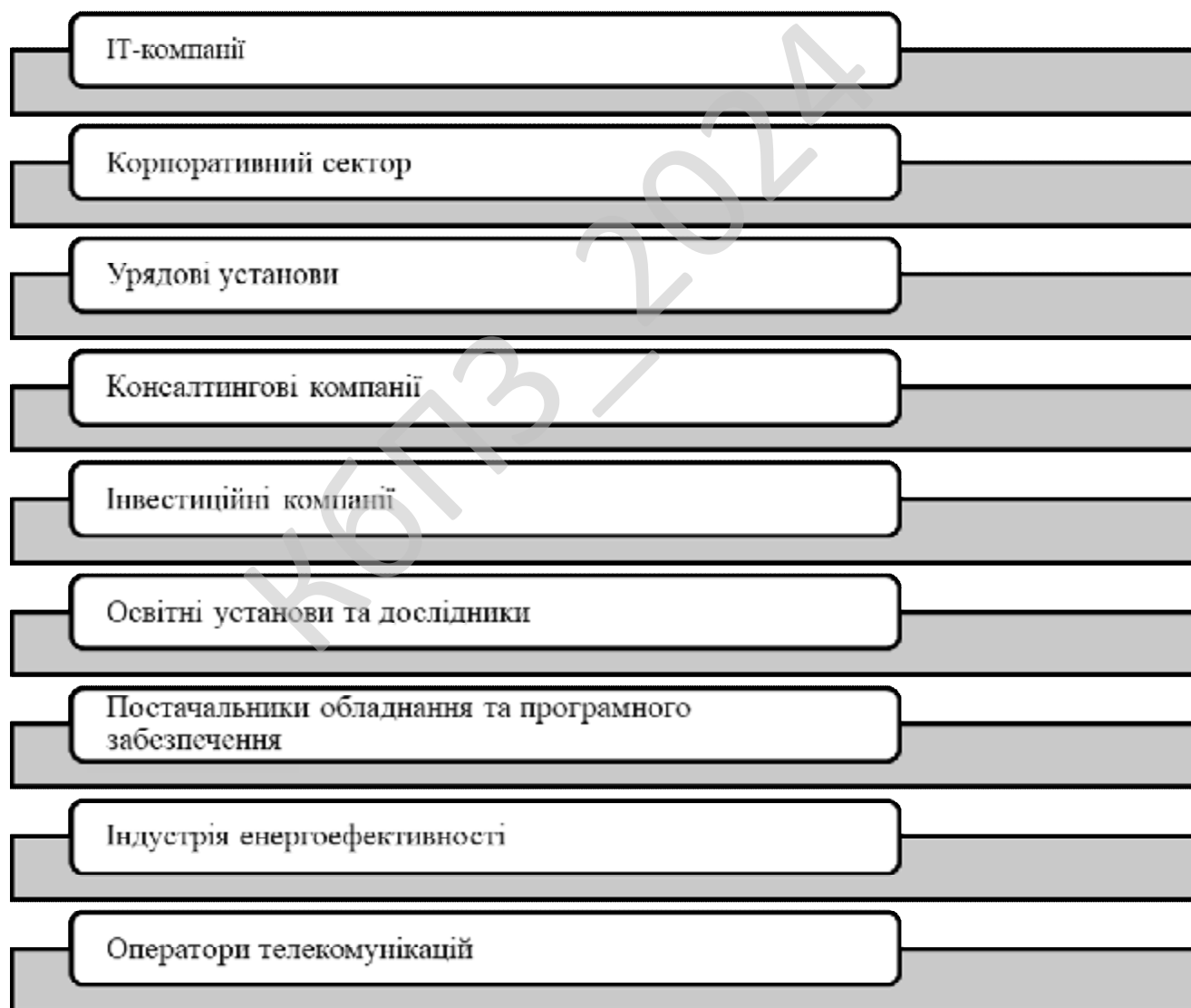


Рисунок 7.1 – Цільова аудиторія

Модернізація ЦОД є актуальним питанням через постійне зростання потреб у зберіганні та обробці даних, що робить ваші результати затребуваними для широкого кола організацій.

## 7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Оцінка привабливості програмної реалізації системи планування модернізації ЦОД за допомогою методів експертних оцінок має виглядати наступним чином.

Обираються ключові критерії для оцінки привабливості проекту, (рисунок 7.2).

<b>Технологічна ефективність</b> (наскільки система полегшує процес модернізації).
<b>Економічна доцільність</b> (зменшення витрат на обслуговування).
<b>Інтеграційний потенціал</b> (сумісність із сучасними ІТ-системами).
<b>Гнучкість та масштабованість</b> (адаптація до змін у вимогах ЦОД).
<b>Енергоефективність</b> (зменшення енергоспоживання).
<b>Рівень ризиків</b> (ймовірність помилок, складність впровадження).

Рисунок 7.2 – Критерії експертної оцінки

Формується група експертів (5–10 осіб), які мають досвід у галузях ІТ, управління ЦОД, консалтингу або впровадження інновацій.

Використовується метод бальної оцінки (наприклад, шкала від 1 до 10, де 10 – найвища оцінка).

Експерти оцінюють систему за кожним критерієм. Результати зводяться в таблицю 7.2.

Таблиця 7.1 – Результати експертних оцінок

Критерій	Експерт 1	Експерт 2	Експерт 3	Середній бал
Технологічна ефективність	8	9	7	8.0
Економічна доцільність	7	8	6	7.0
Інтеграційний потенціал	9	8	9	8.7
Гнучкість та масштабованість	8	9	8	8.3
Енергоефективність	6	7	5	6.0
Рівень ризиків	8	8	7	7.7

Показник обчислюється як середньозважена оцінка, де кожен критерій має свою вагу (визначену за допомогою експертів) – таблиця 7.2.

Таблиця 7.2 – Зведені результати експертних оцінок

Критерій	Середній бал	Вага	Внесок у загальну оцінку
Технологічна ефективність	8.0	0.25	2.0
Економічна доцільність	7.0	0.20	1.4
Інтеграційний потенціал	8.7	0.15	1.31
Гнучкість та масштабованість	8.3	0.20	1.66
Енергоефективність	6.0	0.10	0.6
Рівень ризиків	7.7	0.10	0.77

Загальний показник привабливості = 2.0 + 1.4 + 1.31 + 1.66 + 0.6 + 0.77 = 7.74. Оцінка 7.74 із 10 свідчить про високу привабливість проекту для

реалізації, хоча є області для вдосконалення, наприклад, в аспекті енергоефективності.

Метод експертних оцінок дозволяє отримати об'єктивну та багатокритеріальну оцінку, враховуючи різні аспекти впровадження програмної системи.

### 7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості програмної реалізації системи планування модернізації ЦОД доцільно використовувати метод, який враховує специфіку проекту, масштаб, складність та ризики. Нижче наведені методи (рисунок 7.3), які можна застосувати.

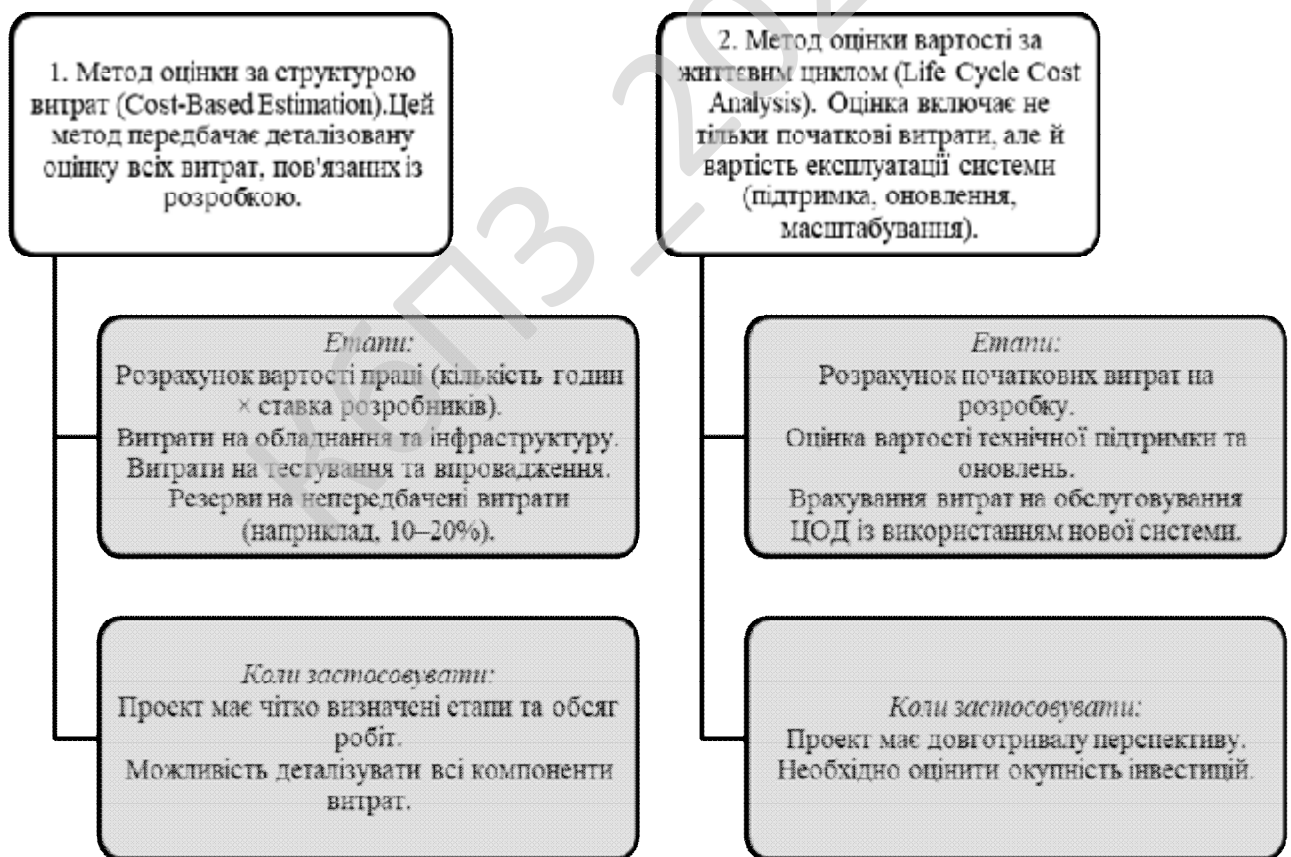


Рисунок 7.3 – Методи оцінки вартості

Для програмної реалізації системи планування модернізації ЦОД найкращим буде комбінований підхід: оцінка за структурою витрат дозволить точно оцінити початкові витрати та метод життєвого циклу допоможе врахувати довгострокову перспективу, окупність і загальні витрати на підтримку та масштабування.

Це забезпечить повну картину витрат, яка буде корисною для прийняття рішень як для замовника, так і для розробника.

#### **7.4 Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості**

Розрахунок економічної ефективності від впровадження системи планування модернізації ЦОД зведено в таблицю 7.3.

Таблиця 7.3 – Економічна ефективність від реалізації проекту для клієнта

<b>Вхідні дані</b>
<b>Поточні витрати на обслуговування ЦОД:</b>
Енергоспоживання: \$500,000 на рік.
Витрати на підтримку обладнання: \$200,000 на рік.
Неефективність (простій через застаріле обладнання): \$100,000 на рік.
<b>Очікувані зміни після впровадження системи:</b>
Зниження енергоспоживання на 20%.
Скорочення витрат на підтримку обладнання на 15%.
Зменшення простоїв на 30%.
<b>Вартість впровадження системи:</b>
Одноразова вартість розробки та впровадження: \$300,000.
Щорічна підтримка системи: \$50,000.

Продовження таблиці 7.3

**Розрахунок економічної ефективності**

**Зменшення витрат на енергоспоживання**

Поточні витрати:  $\$500,000 \times 20\% = \$100,000$  економії на рік.

**Зменшення витрат на підтримку обладнання**

Поточні витрати:  $\$200,000 \times 15\% = \$30,000$  економії на рік.

**Скорочення втрат через простій**

Поточні витрати:  $\$100,000 \times 30\% = \$30,000$  економії на рік.

**Загальна економія за рік**

$\$100,000 + \$30,000 + \$30,000 = \$160,000$ .

**Врахування витрат на підтримку системи**

Чиста річна економія:  $\$160,000 - \$50,000 = \$110,000$ .

**Окупність інвестицій**

Вартість впровадження:  $\$300,000$ .

Період окупності:  $\$300,000 \div \$110,000 \approx 2.73$  роки.

**Результат:**

Впровадження системи дозволяє щорічно економити  $\$110,000$  після врахування витрат на її підтримку.

Система окупиться через **2 роки та 9 місяців**.

За 5 років економія складе:  $\$110,000 \times 5 = \$550,000$ , що перевищує початкові інвестиції майже вдвічі.

**Додаткові переваги:**

**Підвищення надійності:** скорочення простоїв дозволяє уникнути втрат клієнтів.

**Екологічний ефект:** зменшення енергоспоживання сприяє сталому розвитку.

**Конкурентна перевага:** ефективний ЦОД може залучити більше клієнтів.

Цей розрахунок демонструє, як впровадження системи може не лише знизити витрати, але й забезпечити довгострокову вигоду для бізнесу.

## 7.5 Пропозиція алгоритму просування проекту розробки ПЗ

Алгоритм просування проекту програмної реалізації системи планування модернізації ЦОД представлено на рисунку 7.4.



Рисунок 7.4 – Алгоритм просування проекту

Такий алгоритм допоможе ефективно просувати проєкт і забезпечити його успішне впровадження на ринку.

## 7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Пропозиції для оптимізації каналів збуту та шляхів реалізації проекту програмної реалізації системи планування модернізації ЦОД варто розділити окремо на оптимізацію каналів збуту та окремо на шляхи реалізації і схематично подати на рисунках 7.5 та 7.6 відповідно.

<p><b>Прямий продаж (Direct Sales)</b></p> <ul style="list-style-type: none"> <li>• залучення професійних продавців із технічними знаннями.</li> <li>• Створення спеціалізованої команди з фокусом на B2B (компанії з великими ЦОД, банки, IT-холдинги).</li> <li>• Використання CRM-системи для управління лідами та автоматизації процесу продажу.</li> </ul>	<p><b>Інтегратори та партнери</b></p> <ul style="list-style-type: none"> <li>• Співпраця з IT-інтеграторами, які займаються модернізацією або управлінням ЦОД.</li> <li>• Пропонування партнерських програм: <ul style="list-style-type: none"> <li>• Знижки для інтеграторів.</li> <li>• Спільний маркетинг (спонсорування заходів, конференцій).</li> </ul> </li> </ul>	<p><b>Онлайн-канали</b></p> <ul style="list-style-type: none"> <li>• Запуск власної платформи для продажу (веб-сайт із інтерактивними демонстраціями).</li> <li>• Маркетплейси для B2B-рішень (наприклад, AWS Marketplace, Microsoft Azure Marketplace).</li> <li>• Використання SEO та контекстної реклами для залучення потенційних клієнтів.</li> </ul>	<p><b>Франчайзинг</b></p> <ul style="list-style-type: none"> <li>• Розробка франшизи, яка дозволить регіональним компаніям продавати продукт від вашого імені.</li> <li>• Надання навчання та технічної підтримки партнерам-франчайзі.</li> </ul>	<p><b>Конкурси та державні тендери</b></p> <ul style="list-style-type: none"> <li>• Участь у тендерах, особливо для державних організацій із великими ЦОД.</li> <li>• Підготовка спеціальних пропозицій із врахуванням вимог тендерів.</li> </ul>
---	---	--	---	---

Рисунок 7.5 – Оптимізація каналів збуту

<p><b>Модульна реалізація</b></p> <ul style="list-style-type: none"> <li>• Запропонувати систему, яка впроваджується поетапно (модуль за модулем).</li> <li>• Клієнти можуть оновлювати свій ЦОД, знизуючи одноразові витрати.</li> </ul>	<p><b>Модель SaaS (Software as a Service)</b></p> <ul style="list-style-type: none"> <li>• Надавати систему у вигляді підписки з можливістю оплати за використання.</li> <li>• Переваги: <ul style="list-style-type: none"> <li>• Зниження порігу входу для клієнтів.</li> <li>• Постійний потік доходів від підписок.</li> </ul> </li> </ul>	<p><b>Підтримка кастомізації</b></p> <ul style="list-style-type: none"> <li>• Надання можливості адаптувати систему під індивідуальні потреби клієнтів.</li> <li>• Розробка інтеграцій з існуючими системами управління.</li> </ul>	<p><b>Пілотні проекти</b></p> <ul style="list-style-type: none"> <li>• Запропонувати клієнтам безкоштовне або пільгове впровадження для тестування на окремих модулях ЦОД.</li> <li>• Використовувати успішні кейси як доказ ефективності системи для інших клієнтів.</li> </ul>	<p><b>Створення демо-версії</b></p> <ul style="list-style-type: none"> <li>• Розробити онлайн-демо або тестову версію системи, щоб клієнти могли ознайомитися з функціоналом.</li> </ul>	<p><b>Технічна підтримка та навчання</b></p> <ul style="list-style-type: none"> <li>• Організувати тренінги для персоналу клієнта.</li> <li>• Забезпечити швидку технічну підтримку, що підвищить довіру до продукту.</li> </ul>
---	---	---	--	--	--

Рисунок 7.6 – Шляхи реалізації проєкту

Це має дати наступні результати: розширення охоплення ринку через багатоканальний підхід, зростання продажів завдяки партнерствам і прямому контакту з клієнтами, скорочення термінів прийняття рішень клієнтами через демонстрацію швидкої вигоди.

Ці заходи дозволять ефективно масштабувати проєкт і зробити його конкурентоспроможним на ринку.

## 7.7 Визначення ключових факторів успіху конкретного проєкту

Ключові фактори успіху проєкту програмної реалізації системи планування модернізації ЦОД представлено на рисунку 7.7.

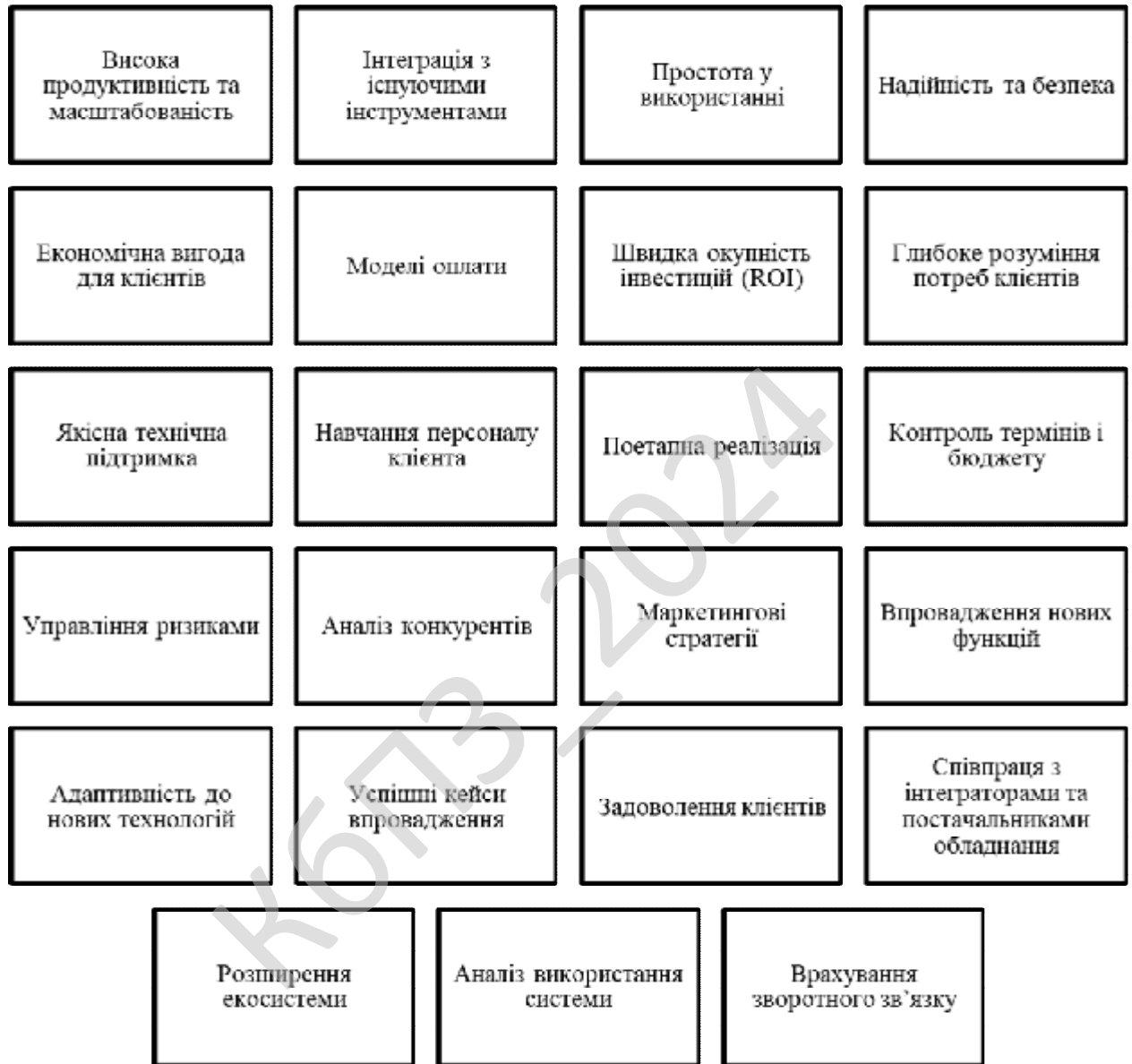


Рисунок 7.7 – Ключові фактори успіху проєкту

Ключові фактори успіху – це поєднання технічної якості продукту, орієнтованості на клієнта, ефективного управління проєктом і постійного вдосконалення. Це забезпечить конкурентоспроможність рішення та його популярність серед цільової аудиторії.

## 8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

### 8.1 Вступ

В кожній ІТ-компанії є трудові відносини з працівниками. Згідно закону України “Про охорону праці” [3] кожна компанія впроваджує заходи з охорони праці. Реалізується трудові відносини з вживанням необхідних засобів з охорони праці та розробки відповідних документів:

- Інструкцій з охорони праці по кожній професії і загальні.
- Положення про охорону праці.
- Накази з охорони праці.
- Журнали реєстрації та інструктажу.

Роботодавець створює відділ який працює відповідно до типового положення, яке затверджується центральним органом виконавчої влади і забезпечує виконання вимог державної політики у сфері охорони праці.

За недотримання вимог, керівники ІТ-компаній можуть бути притягнуті до відповідальності, яка тягне накладання штрафу. Якщо в результаті порушення умов охорони праці є постраждалі працівники, то керівні особи ІТ-компаній притягуються до кримінальної відповідальності.

Законом України “Про охорону праці” [3] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» [5], яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми

					ВКРМ-123.24.0027.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98.

Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругою і нервово-емоційне навантаження. Руки (суглоби пальців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаженням. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій (ІТ) спеціалісти відносять високочастотні електромагнітні коливання (випромінювання) роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

Розглянемо шкідливі чинники роботи програмістів керуючись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно- обчислювальних машин» ДСанПіН 3.3.2-007-98 [5], та «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» НПАОП 0.00-7.15-18.

Умови праці програміста включають наступні фактори:

- параметри повітряного середовища в приміщенні;
- вентиляція приміщення;
- освітлення приміщення;
- параметри повітряного середовища в приміщенні, тощо.

Щоб запропонувати заходи щодо зменшення негативного впливу комп'ютера на організм людини визначимо фактори, які можуть викликати професійне захворювання і впливають на працездатність програміста.

## **8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером**

Електронно-обчислювальна машин (ЕОМ) та інше обладнання є джерелами безпеки ураження електричним струмом. Оскільки робота програміста характеризується істотним зоровим навантаженням, то вимагає належного

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>103</b>

освітлення. У приміщенні, в якому працюють програмісти, необхідно створити належний мікроклімат, параметри якого регламентуються Державними санітарними правилами і нормами, зокрема ДСанПіН 3.3.2.007-98.

При роботі з використанням ЕОМ відзначають наступні небезпечні та шкідливі фактори:

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.
- ризик виникнення пожежі;
- негативний вплив на органи зору людини;
- ризики ураження електричним струмом;
- недостатня, або надмірна освітленість робочого місця;
- електромагнітні (у тому числі високочастотні) випромінювання (коливання);
- несприятливі мікрокліматичні умови;
- нервово-емоційна напруженість праці;
- інтелектуальні навантаження;
- монотонність праці;
- невідповідність ергономічних показників робочого місця діючим вимогам;
- шум;
- статичні навантаження на кістково-м'язовий апарат.

Працю користувачів ЕОМ відносять до психічних форм праці з високим ступенем навантаження. Ця діяльність пов'язана зі сприйняттям зображення на екрані, постійним стеженням за його динамікою, розрізненням картин, схем, читанням тексту рукописних та друківаних матеріалів, введенням інформації з клавіатури, необхідністю підтримувати активну увагу високою ціною помилки. Будь-яка діяльність із застосуванням ЕОМ супроводжується необхідністю активації уваги та інших вищих психічних функцій, а організм людини, крім того, піддається впливу кількох десятків різноманітних факторів.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

### 8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Розглянемо умови праці у приміщенні, в якому працюють програмісти. Геометричні розміри приміщення наведено у таблиці 8.1.

Таблиця 8.1 – Розміри приміщення

Найменування	Значення, м
Ширина	5
Довжина	6,2
Висота	3,4

Таблиця 8.2 – Площа та обсяг приміщення, на одного працюючого\*

Геометрична характеристика	Одиниця виміру	Нормативне значення*	Фактичне значення
Площа, S	м <sup>2</sup>	не менше 6.0	6,2
Об'єм, V	м <sup>3</sup>	не менше 20.0	21,8

\* Згідно ДСанПіН 3.3.2.007-98 (Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин).

У зазначеному приміщенні працюють двоє людей. За даними, які наведено у таблицях 8.1-8.2, можна зробити висновок, що площа та об'єм приміщення у розрахунку на одне робоче місце програміста не відповідають нормативним вимогам ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно- обчислювальних машин» [5], але відповідають нормативним вимогам Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [5] та НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-

обчислювальних машин»). Таким чином, можна зробити висновок, що санітарно-гігієнічні умови праці на робочому місці програміста відповідають вимогам.

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно Постанови №42 від 01.12.1999 Головного державного санітарного лікаря України, робота, виконувана в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря в приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

Мікроклімат повітряного середовища в приміщенні характеризується запиленістю та загазованістю повітря. Мікроклімат приміщення визначається діючим на організм людини поєднанням, вологості, температури, швидкості руху повітря та інтенсивності теплового випромінювання. Аналіз мікроклімату складається з визначення зазначених вище факторів і порівняння результатів із встановленими нормами.

У таблиці 8.3 наведено оптимальні та фактичні значення параметрів мікроклімату як для категорії ваги робіт Іа, так і розглянутого приміщення. У приміщеннях, де встановлено ЕОМ, рекомендується дотримування тільки оптимальних значень показників мікроклімату.

Проведений аналіз показує, що показники мікроклімату в приміщенні відповідають установленим нормам. Штучне опалення застосовується у холодний період року.

В літню пору застосовується кондиціонер.

Для боротьби з пилом робляться регулярні провітрювання та вологі прибирання приміщенні.

					ВКРМ-123.24.0027.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		106



збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

#### **8.4 Розробка заходів з умов поліпшення охорони праці**

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

- розмірі приміщення, у розрахунку на одному працюючого, відповідають нормативам;
- мікроклімат відповідає нормативному значенню;
- акустичні умови роботи не перевищують нормативних значень;

Таким чином можна припустити, що основною причиною можливого зниження працездатності програміста є психофізіологічний фактор, тому основна пропозиція буде така: дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці та відпочинку, організація робочого місця з урахуванням ергономічних вимог.

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язково наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга).

Регулярна наочне знайомство персоналу із шляхами для евакуації людей із приміщення відповідно до плану евакуації, забезпечення розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв, які працюють при нарузі вище 36 В.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>108</b>

Так як при ураженні електричним струмом у людини може статися фібриляція шлуночків серця, в організації бажано мати дефібрилятор і підготовлений персонал для роботи з ним.

## 8.5 Розрахункова частина

Проведемо розрахунок штучного освітлення за методом коефіцієнта використання світлового потоку для приміщення ширина якого складає 5 м, довжина – 6,2 м, висота – 3,4 м.

У зазначеному приміщенні працює 5 людей.

Для того, щоб визначити потрібну кількість світильників, які повинні забезпечити нормований рівень освітленості, визначимо світловий потік, що падає на робочу поверхню за формулою:

$$F = E \cdot S \cdot K \cdot Z / n,$$

де  $F$  – світловий потік, що розраховується, Лм;

$E$  – нормована мінімальна освітленість, Лк;  $E = 300$  Лк;

$S$  – площа освітлюваного приміщення (у нашому випадку  $S = 5 \times 6,2 = 31 \text{ м}^2$ );

$K$  – коефіцієнт запасу, що враховує зменшення світлового потоку лампи в результаті забруднення світильників в процесі експлуатації (його значення залежить від типу приміщення і характеру робіт, що проводяться в ньому, в нашому випадку  $K = 1,5$ );

$Z$  – відношення середньої освітленості до мінімальної (зазвичай приймається рівним 1.1... 1.2, в нашому випадку  $Z = 1,1$ );

$n$  – коефіцієнт використання світлового потоку, (відношення світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп, обчислюється в долях одиниці; залежить від характеристик світильника, розмірів приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін ( $\rho_{\text{стін}}$ ) і стелі ( $\rho_{\text{стелі}}$ ), значення коефіцієнтів дорівнюють  $\rho_{\text{стін}} = 50\%$  і  $\rho_{\text{стелі}} = 50\%$ .

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		109

Обчислимо індекс приміщення за формулою:

$$i=S/(h\cdot(A+B)),$$

де:  $S$  – площа приміщення,  $S = 31 \text{ м}^2$ ;

$h$  – розрахункова висота підвісу,  $h = 3 \text{ м}$  (співпадає з висотою стелі, оскільки лампи освітлення закріплюються на стелі);

$A$  – ширина приміщення,  $A = 5 \text{ м}$ ;

$B$  – довжина приміщення,  $B = 6,2 \text{ м}$ .

Підставимо всі значення у формулу та визначимо індекс приміщення:  
 $i=0,43$ .

Знаючи індекс приміщення, за знаходимо  $n = 0,23$  (з табличних даних коефіцієнтів використання світлового потоку ( $n$ ) світильників з відповідним типом ламп). Підставимо всі значення у формулу, визначимо світловий потік:  $F=66717 \text{ Лм}$ . Для розрахунку будемо використовувати світлодіодні панелі LED панель PL PFM 600 30W/3000K, світловий потік яких  $F_{\text{л}} = 3000 \text{ Лм}$ .

Число ламп визначається за формулою:

$$N=F/F_{\text{л}}$$

де  $F$  – світловий потік,  $F_{\text{л}}$  – світловий потік однієї лампи.

Підставимо всі значення у формулу та визначимо потрібну кількість ламп:

$$N= 66717/3000 = 22,18 \text{ шт.}$$

Приймаємо необхідну кількість світлодіодних світильників 23 шт.

### **Висновки до розділу**

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров'я працівників, а також підвищує ефективність виробництва в цілому.

З цих міркувань було здійснено аналіз умов праці, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок штучного освітлення, як одного з ключових факторів впливу на працездатність та здоров'я програміста. Розроблено заходи з охорони праці.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		110

## 9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи планування модернізації ЦОД.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів планування модернізації ЦОД.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем планування модернізації ЦОД.
- Досліджена система планування модернізації ЦОД.
- На основі отриманих результатів досліджень створена програмна реалізація системи планування модернізації ЦОД.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання планування модернізації ЦОД.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

					ВКРМ-123.24.0027.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		111

Програма реалізована на мові високого рівня Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ДСТУ 8845:2019.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

Проведено маркетингове та економічне обґрунтування ІТ-проєкту, що дозволило визначити ключові фактори успіху даного проєкту.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		112

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Найдъонов Є.В. Дослідження та програмна реалізація системи планування модернізації ЦОД // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2024.
2. Ramon Nastase «Computer Networking: The Beginner’s guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.
3. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
4. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.
5. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.
6. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchев, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.
7. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.
8. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». *CEUR Workshop Proceedings Volume 3156*, 2022, Pages 390-399.

					ВКРМ-123.24.0027.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		113

9. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». *Communications in Computer and Information Science*, 2021, vol 1486. Springer, Cham. pp 169-184.

10. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.

11. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.

12. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

13. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) *Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies*, vol 48. Springer, Cham. 2021. pp 557-587.

14. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 125-136.

15. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 366-379.

16. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated

with the Markov chains», *CEUR Workshop Proceedings* Volume 2608, 2020, Pages 633-645.

17. Smirnov O., Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». *International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019*; Odessa; Ukraine; 9-13 September 2019. P.22-28.

18. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». *International Journal of Computing*; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

19. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

20. Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyz, M., «QoE optimization technique for media delivery in 5G networks». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019. P.597-601.

21. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». *CEUR Workshop Proceedings*, Vol 2588, P. 90-106, 2019.

22. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», *2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019*, P. 395-399.

23. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in

					<b>БКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		115

Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 353-358.

24. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 347-352.

25. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», *CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019*, Pages 618-629.

26. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», *Telecommunications and Radio Engineering*. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

27. Смірнова Т.В., Коноплицька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка*. 2024. №4(24), С. 6-27.

28. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології*, 2024, № 13, с. 28-35.

29. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.

30. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління*, № 2(70). 2022. С. 28-37.

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		116

31. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 3(69). С. 93-98.

32. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»*, № 2 (307). С. 46-52. 2022.

33. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89.

34. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». *Сучасні інформаційні системи*. 2021. Т. 5, № 4. С. 79-95

35. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» *Вісник Черкаського державного технологічного університету. Технічні науки*. №4. С. 103-110. 2020.

36. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», *Кібербезпека: освіта, наука, техніка*. № 3(7). С. 43-62. 2020.

37. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Поліщук Л.І. Інформаційна безпека в

					ВКРМ-123.24.0027.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		117

комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2020. – 294 с.

38. О.А. Смірнов, П.С. Усік, «Дослідження перспектив використання технологічних рішень в мережах 5G» у *Кібербезпека та інформаційні технології: монографія*. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.

39. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». *Центральноукраїнський науковий вісник. Технічні науки*. № 2(33). с. 161-172, 2019.

40. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. Проектування комп'ютерних систем та мереж. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.

41. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

42. Смірнов О.А., Дреєва Г.М. Метод генерування фрактального трафіку за допомогою моделі генератора на графі. Монографія: Інформаційна безпека та інформаційні технології : монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139

43. Дреєва Г.М., Смірнов О.А., Дреєв О.М. Метод генерування фрактальноподібної числової послідовності на основі скінченного автомату для моделювання трафіку у мережі. *Центральноукраїнський науковий вісник. Технічні науки*. № 1(32). с. 173-183, 2019.

44. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу

					<b>ВКРМ-123.24.0027.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		118

відновлення та зміцнення поверхонь деталей. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.

45. Смірнов О.А., Смірнов С.А., Поліщук Л.І., Смірнова Т.В., Коноплицька-Слободенюк О.К. Метод формування антивірусного захисту даних з використанням безпечної маршрутизації метаданих. Кібербезпека: освіта, наука, техніка. – Том 3 № 3. – Київ: КУ ім. Бориса Грінченка. – 2019. – С. 63-87.

46. Смірнов О.А., Гнатюк С.О., Кавун С.В., Терейковський І.А., Жмурко Т.О., Смірнов С.А., Коваленко А.С. Основи безпеки в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2018. – 177 с.

47. Смірнов О.А., Котелянець В.В. Стійкі до колізій стохастичні моделі функціонування безпроводових сенсорних мереж. Вісник інженерної академії України, №3, с. 145-152, 2018

48. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Алгоритми формування безлічі маршрутів передачі метаданих у антивірусні хмарні системи. Збірник наукових праць "Системи обробки інформації". – Випуск 5 (142). – Х.: ХУПС – 2016. – С. 148-152.

49. Смірнов О.А., Смірнов С.А. Дідик А.К., Дреєв О.М. Моделі системи нейромережових експертів безпечної маршрутизації у хмарних антивірусних системах. Збірник наукових праць "Системи обробки інформації". – Випуск 3 (140). – Х.: ХУПС – 2016. – С. 36-39.

50. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 2 (47). – Харків: ХУПС. – 2016. – С. 121-127.

51. Смірнов О.А., Смірнов С.А., Дідик А.К. Метод безпечної маршрутизації метаданих у хмарні антивірусні системи. Системи озброєння та військова техніка. – Випуск 2 (46) – Х.: ХУПС – 2016. – С. 146-149.

					ВКРМ-123.24.0027.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		119

Додаток А  
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування.....	2
2 Підстава для розробки.....	2
3 Мета та призначення розробки.....	2
4 Джерела розробки.....	2
5 Технічні вимоги.....	2
5.1 Вміст проекту.....	2
5.2 Показники призначення.....	3
5.3 Вимоги до функціональних характеристик.....	3
5.4 Вимоги до архітектури.....	3
5.5 Вимоги до надійності.....	3
5.6 Умови експлуатації.....	4
5.7 Вимоги до складу та параметрів технічних засобів.....	4
5.8 Вимоги до інформаційної і програмної сумісності.....	4
5.8.1 Обладнання.....	4
5.8.2 Мова програмування.....	4
5.8.3 Вхідні дані.....	5
5.8.4 Вихідні дані.....	5
6 Вимоги до програмної документації.....	5
7 Економічні вимоги.....	5
8 Вимоги щодо охорони праці.....	5
9 Перелік документів, що розробляються.....	6
10 Етапи розробки.....	6
11 Порядок контролю та приймання.....	6

					<b>ВКРМ-123.24.0027.00.00.ТЗ</b>			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Найдьонов Є.В.				<i>Дослідження та програмна реалізація системи планування модернізації ЦОД</i>	Літ.	Аркуш	Аркушів
Перевірів	Усік П.С.					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-23М			
Затв.	Смірнов О.А.							

## 1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи планування модернізації ЦОД.

## 2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 19-13 від 07.08.2024 року).

## 3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи планування модернізації ЦОД.

## 4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

## 5 Технічні вимоги

### 5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРМ-123.24.0027.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;
- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

## 5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи планування модернізації ЦОД;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

## 5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

## 5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

## 5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					ВКРМ-123.24.0027.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

## 5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

## 5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

## 5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

### 5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

### 5.8.2 Мова програмування

Середовище Python.

					ВКРМ-123.24.0027.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

### 5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

### 5.8.4 Вихідні дані

Робоча програма.

## 6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

## 7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати маркетингове та економічне обґрунтування ІТ-проєкту з урахуванням цін на 3 вересня 2024 року.

## 8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинні бути розглянуті шкідливі і небезпечні фактори при роботі з комп'ютером.

					ВКРМ-123.24.0027.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

## 9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 119 аркушів.

## 10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

10.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Маркетингове та економічне обґрунтування ІТ-проєкту.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

## 11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 02.12.2024 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 20.12.2024 р.

					<b>ВКРМ-123.24.0027.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б  
(обов'язковий)

**Міністерство освіти і науки України**  
**Центральноукраїнський національний технічний університет**

**ЗАТВЕРДЖУЮ**

Керівник випускної кваліфікаційної роботи за  
другим (магістерським) рівнем вищої освіти  
\_\_\_\_\_ Усік П.С.

*Дослідження та програмна реалізація  
системи планування модернізації ЦОД*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 14

Літера: РП

Кропивницький – 2024 року

## Основна програма

```
# Оголошуємо клас для автоматичного моніторингу ресурсів у реальному часі
class ResourceMonitor:
    def __init__(self, datacenter):
        self.datacenter = datacenter

    def monitor_resources(self):
        for server in self.datacenter.servers:
            print(f"Monitoring {server.name}: {server.cpu} CPU,
{server.memory}GB Memory")

# Оголошуємо клас для інтеграції з хмарними провайдерами
class CloudIntegration:
    def __init__(self):
        self.cloud_providers = []

    def add_provider(self, provider_name, cost_per_cpu, cost_per_memory):
        self.cloud_providers.append({
            "provider_name": provider_name,
            "cost_per_cpu": cost_per_cpu,
            "cost_per_memory": cost_per_memory
        })

    def analyze_migration(self, cpu_needed, memory_needed):
        migration_options = []
        for provider in self.cloud_providers:
            total_cost = (cpu_needed * provider["cost_per_cpu"] +
                memory_needed * provider["cost_per_memory"])
            migration_options.append({
                "provider_name": provider["provider_name"],
                "total_cost": total_cost
            })
        return migration_options

# Оголошуємо клас для оцінки ризиків при оновленні
class RiskAssessment:
    def __init__(self, datacenter):
        self.datacenter = datacenter

    def assess_risks(self):
        risks = []
        for server in self.datacenter.servers:
            if server.cpu < 4:
                risks.append(f"Server {server.name} has low CPU performance.")
            if server.memory < 8:
```

```

        risks.append(f"Server {server.name} has insufficient memory.")
    return risks

# Оголошуємо клас для моделювання сценаріїв навантаження
class LoadModeling:
    def __init__(self):
        self.scenarios = []

    def add_scenario(self, name, cpu_load, memory_load):
        self.scenarios.append({
            "name": name,
            "cpu_load": cpu_load,
            "memory_load": memory_load
        })

    def simulate(self, datacenter):
        results = []
        for scenario in self.scenarios:
            total_cpu, total_memory = datacenter.total_resources()
            cpu_ok = total_cpu >= scenario["cpu_load"]
            memory_ok = total_memory >= scenario["memory_load"]
            results.append({
                "scenario_name": scenario["name"],
                "cpu_sufficient": cpu_ok,
                "memory_sufficient": memory_ok
            })
        return results

# Ініціалізуємо та використовуємо модулі

# Автоматичний моніторинг ресурсів
monitor = ResourceMonitor(datacenter)
monitor.monitor_resources()

# Інтеграція з хмарними провайдерами
cloud = CloudIntegration()
cloud.add_provider("AWS", 10, 5)
cloud.add_provider("Azure", 12, 4)
cloud_migration_options = cloud.analyze_migration(20, 40)
for option in cloud_migration_options:
    print(f"Provider: {option['provider_name']}, Cost: ${option['total_cost']}")

# Оцінка ризиків при оновленні
risk_assessment = RiskAssessment(datacenter)
risks = risk_assessment.assess_risks()
for risk in risks:

```

```
print(f"Risk: {risk}")

# Моделювання сценаріїв навантаження
load_modeling = LoadModeling()
load_modeling.add_scenario("High Load", 50, 100)
load_modeling.add_scenario("Medium Load", 30, 60)
simulation_results = load_modeling.simulate(datacenter)
for result in simulation_results:
    print(f"Scenario: {result['scenario_name']}, "
          f"CPU Sufficient: {result['cpu_sufficient']}, "
          f"Memory Sufficient: {result['memory_sufficient']}")
```

КБПЗ\_2024

## Файл VirtualMachineManager.py

```
# Оголошуємо клас для управління віртуальними машинами та контейнерами
class VirtualMachineManager:
    def __init__(self):
        self.virtual_machines = []

    def create_vm(self, name, cpu, memory):
        self.virtual_machines.append({
            "name": name,
            "cpu": cpu,
            "memory": memory
        })

    def list_vms(self):
        for vm in self.virtual_machines:
            print(f"VM {vm['name']}: {vm['cpu']} CPU, {vm['memory']}GB Memory")

# Оголошуємо клас для резервного копіювання та відновлення
class BackupManager:
    def __init__(self):
        self.backups = []

    def create_backup(self, server_name):
        self.backups.append(f"Backup for {server_name}")
        print(f"Backup created for {server_name}")

    def list_backups(self):
        for backup in self.backups:
            print(f"Available backup: {backup}")

# Оголошуємо клас для інтеграції з інструментами кібербезпеки
class CyberSecurityIntegration:
    def __init__(self):
        self.security_tools = []

    def add_tool(self, tool_name, description):
        self.security_tools.append({
            "tool_name": tool_name,
            "description": description
        })

    def list_tools(self):
        for tool in self.security_tools:
            print(f"Tool: {tool['tool_name']}, Description: {tool['description']}")

# Оголошуємо клас для підтримки багатомовного інтерфейсу
class MultiLanguageSupport:
    def __init__(self):
        self.languages = {}
```

```
def add_language(self, lang_code, messages):
    self.languages[lang_code] = messages

def get_message(self, lang_code, message_key):
    return self.languages.get(lang_code, {}).get(message_key, "Message not
found")

# Ініціалізація модулів для підключення до основного коду
def initialize_library():
    vm_manager = VirtualMachineManager()
    backup_manager = BackupManager()
    security_integration = CyberSecurityIntegration()
    language_support = MultiLanguageSupport()

    return vm_manager, backup_manager, security_integration, language_support
```

КБПЗ\_2024

```
# Оголошуємо клас для генерації детальних аналітичних звітів
class ReportGenerator:
    def __init__(self):
        self.reports = []

    def create_report(self, title, content):
        report = {
            "title": title,
            "content": content
        }
        self.reports.append(report)
        print(f"Report '{title}' created.")

    def list_reports(self):
        for report in self.reports:
            print(f"Report Title: {report['title']}")

    def show_report(self, title):
        for report in self.reports:
            if report["title"] == title:
                print(f"Title:
{report['title']}\nContent:\n{report['content']}")
                return
        print("Report not found.")

# Оголошуємо клас для використання штучного інтелекту для оптимізації планів
class AIOptimizer:
    def __init__(self):
        self.plans = []

    def add_plan(self, name, efficiency_score):
        self.plans.append({
            "name": name,
            "efficiency_score": efficiency_score
        })

    def optimize(self):
        if not self.plans:
            print("No plans to optimize.")
            return None
        optimal_plan = max(self.plans, key=lambda p: p["efficiency_score"])
        print(f"Optimal Plan: {optimal_plan['name']} with efficiency score
{optimal_plan['efficiency_score']}")
        return optimal_plan

# Оголошуємо клас для розширення фінансового планування
class DynamicBudgeting:
    def __init__(self):
        self.budget = {}
```

```

def set_budget(self, category, amount):
    self.budget[category] = amount
    print(f"Budget for {category} set to ${amount}")

def get_total_budget(self):
    total_budget = sum(self.budget.values())
    print(f"Total Budget: ${total_budget}")
    return total_budget

def adjust_budget(self, category, new_amount):
    if category in self.budget:
        self.budget[category] = new_amount
        print(f"Budget for {category} adjusted to ${new_amount}")
    else:
        print(f"Category {category} not found in the budget.")

# Оголошуємо клас для оцінки впливу модернізації на екологічність
class SustainabilityAnalyzer:
    def __init__(self):
        self.projects = []

    def add_project(self, name, carbon_footprint_reduction):
        self.projects.append({
            "name": name,
            "carbon_footprint_reduction": carbon_footprint_reduction
        })

    def analyze_impact(self):
        total_reduction = sum(project["carbon_footprint_reduction"] for project
in self.projects)
        print(f"Total Carbon Footprint Reduction: {total_reduction} metric tons
CO2")
        return total_reduction

# Ініціалізація модулів для підключення до основного коду
def initialize_library():
    report_generator = ReportGenerator()
    ai_optimizer = AIOptimizer()
    dynamic_budgeting = DynamicBudgeting()
    sustainability_analyzer = SustainabilityAnalyzer()

    return report_generator, ai_optimizer, dynamic_budgeting,
sustainability_analyzer

```

## Файл APIIntegration.py

```
# Оголошуємо клас для додавання API для інтеграції зі сторонніми сервісами
class APIIntegration:
    def __init__(self):
        self.api_endpoints = []

    def add_endpoint(self, endpoint, description):
        self.api_endpoints.append({
            "endpoint": endpoint,
            "description": description
        })

    def list_endpoints(self):
        for endpoint in self.api_endpoints:
            print(f"Endpoint: {endpoint['endpoint']}, Description: {endpoint['description']}")

    def call_endpoint(self, endpoint, data):
        print(f"Calling API Endpoint: {endpoint} with data: {data}")
        # Симуляція виклику API
        return f"Response from {endpoint}"

# Оголошуємо клас для автоматизації встановлення оновлень обладнання
class HardwareUpdater:
    def __init__(self):
        self.update_logs = []

    def schedule_update(self, server_name, update_description):
        self.update_logs.append({
            "server_name": server_name,
            "update_description": update_description
        })
        print(f"Update scheduled for {server_name}: {update_description}")

    def execute_updates(self):
        for log in self.update_logs:
            print(f"Executing update for {log['server_name']}: {log['update_description']}")
        self.update_logs.clear()
        print("All updates executed.")

# Оголошуємо клас для впровадження графічних дашбордів для візуалізації
class DashboardVisualizer:
    def __init__(self):
        self.data_points = []

    def add_data_point(self, label, value):
        self.data_points.append({
            "label": label,
            "value": value
        })
```

```
def display_dashboard(self):
    print("Dashboard Visualization:")
    for data_point in self.data_points:
        print(f"{data_point['label']}: {data_point['value']}")

# Ініціалізація модулів для підключення до основного коду
def initialize_library():
    api_integration = APIIntegration()
    hardware_updater = HardwareUpdater()
    dashboard_visualizer = DashboardVisualizer()

    return api_integration, hardware_updater, dashboard_visualizer
```

КБПЗ\_2024

## Файл RemoteManagement.py

```
# Оголошуємо клас для віддаленого керування модернізацією
class RemoteManagement:
    def __init__(self):
        self.tasks = []

    def add_task(self, task_name, server_name):
        self.tasks.append({
            "task_name": task_name,
            "server_name": server_name
        })
        print(f"Task '{task_name}' added for server '{server_name}'")

    def execute_tasks(self):
        for task in self.tasks:
            print(f"Executing task '{task['task_name']}' on server '{task['server_name']}'")
        self.tasks.clear()
        print("All remote tasks executed.")

# Оголошуємо клас для збереження історії оновлень та змін
class UpdateHistory:
    def __init__(self):
        self.history = []

    def log_update(self, server_name, update_description):
        self.history.append({
            "server_name": server_name,
            "update_description": update_description
        })
        print(f"Update logged for server '{server_name}': {update_description}")

    def show_history(self):
        print("Update History:")
        for entry in self.history:
            print(f"Server: {entry['server_name']}, Update: {entry['update_description']}")

# Оголошуємо клас для оцінки продуктивності після модернізації
class PerformanceEvaluator:
    def __init__(self):
        self.performance_data = []

    def add_performance_metric(self, server_name, metric_name, value):
        self.performance_data.append({
            "server_name": server_name,
            "metric_name": metric_name,
            "value": value
        })
        print(f"Performance metric '{metric_name}' with value '{value}' added for server '{server_name}'")
```

```
def evaluate_performance(self):
    print("Performance Evaluation:")
    for metric in self.performance_data:
        print(f"Server: {metric['server_name']}, Metric:
{metric['metric_name']}, Value: {metric['value']}")

# Ініціалізація модулів для підключення до основного коду
def initialize_library():
    remote_management = RemoteManagement()
    update_history = UpdateHistory()
    performance_evaluator = PerformanceEvaluator()

    return remote_management, update_history, performance_evaluator
```

КБПЗ\_2024

## Файл MultiDataCenterManager.py

```

# Оголошуємо клас для підтримки роботи з кількома ЦОД одночасно
class MultiDataCenterManager:
    def __init__(self):
        self.datacenters = {}

    def add_datacenter(self, name, datacenter_object):
        self.datacenters[name] = datacenter_object
        print(f>Data center '{name}' added.")

    def list_datacenters(self):
        print("Registered Data Centers:")
        for name in self.datacenters.keys():
            print(f"- {name}")

    def get_resources(self):
        print("Aggregated Resources from All Data Centers:")
        total_cpu = 0
        total_memory = 0
        for name, datacenter in self.datacenters.items():
            dc_cpu, dc_memory = datacenter.total_resources()
            print(f"{name}: {dc_cpu} CPU, {dc_memory}GB Memory")
            total_cpu += dc_cpu
            total_memory += dc_memory
        print(f>Total CPU: {total_cpu}, Total Memory: {total_memory}GB")

# Оголошуємо клас для інтеграції з системами управління проектами
class ProjectManagementIntegration:
    def __init__(self):
        self.projects = []

    def create_project(self, project_name, description, deadline):
        self.projects.append({
            "project_name": project_name,
            "description": description,
            "deadline": deadline
        })
        print(f>Project '{project_name}' created with deadline '{deadline}'.")

    def list_projects(self):
        print("Active Projects:")
        for project in self.projects:
            print(f>Name: {project['project_name']}, Description:
{project['description']}, Deadline: {project['deadline']}")

    def mark_project_complete(self, project_name):
        for project in self.projects:
            if project['project_name'] == project_name:
                self.projects.remove(project)
                print(f>Project '{project_name}' marked as complete.")
        return

```

```
print(f"Project '{project_name}' not found.")

# Ініціалізація модулів для підключення до основного коду
def initialize_library():
    multi_datacenter_manager = MultiDataCenterManager()
    project_management_integration = ProjectManagementIntegration()

    return multi_datacenter_manager, project_management_integration
```

КБПЗ\_2024