

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
“ ____ ” _____ 2025 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи інтелектуальної
інтеграції існуючих компонент ІТ-системи з хмарними
службами наступного покоління”

Виконав здобувач вищої освіти
II курсу, групи КН-24М
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Ковтуненко Д.О.
« ____ » _____ 2025 р.

Керівник проекту
кандидат технічних наук
_____ Смірнова Т.В.
« ____ » _____ 2025 р.

Рецензент _____

АНОТАЦІЯ

Ковтуненко Д.О. Дослідження та програмна реалізація системи інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2025.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління.

Метою розробки є дослідження та програмна реалізація системи інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління.

Об'єктом дослідження є процес інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління.

Предметом дослідження є методи інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління.

Методи дослідження базуються на методах хмарних технологій, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління.

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'ютерні науки, хмарні служби

ABSTRACT

Kovtunenکو D.O. Research and software implementation of the system of intellectual integration of existing IT system components with next-generation cloud services. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2025.

In this final qualification work for the second (master's) level of higher education, software has been developed, which is intended for the system of intellectual integration of existing IT system components with next-generation cloud services.

The purpose of the development is the research and software implementation of the system of intellectual integration of existing IT system components with next-generation cloud services.

The object of the research is the process of intellectual integration of existing IT system components with next-generation cloud services.

The subject of the research is the methods of intellectual integration of existing IT system components with next-generation cloud services.

The research methods are based on cloud technology methods, mathematical statistics methods, software development methods.

The result of the work is a software implementation of a system for intelligent integration of existing IT system components with next-generation cloud services.

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 a PC with OS Windows 10/11.

The program was developed in the Python environment.

Keywords: computer science, cloud services

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	8
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	10
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	10
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	20
2.3 Розгорнута постановка завдання	25
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	26
3.1 Опис функціонування системи	26
3.2 Розробка структурної схеми.....	31
3.3 Розробка функціональної схеми	37
3.4 Розробка діаграми процесів.....	43
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	45
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	45
4.2 Захист розробленого програмного забезпечення.....	59
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	61
6 НАУКОВА НОВИЗНА	68

					ВКРМ-122.25.0040.00.00.ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата	<i>Дослідження та програмна реалізація системи інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління</i>	Літ.	Аркуш	Аркушів
<i>Розроб.</i>	<i>Ковтуненко Д.О.</i>					М	1	92
<i>Перев.</i>	<i>Смірнова Т.В.</i>					ЦНТУ КН-24М		
<i>Н.контр.</i>	<i>Коваленко А.С.</i>							
<i>Затв.</i>	<i>Смірнов О.А.</i>							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ	69
7.1	Визначення цільової аудиторії кінцевого готового продукту	69
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	70
7.3	Вибір методу оцінки вартості ПЗ	71
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	72
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ	73
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ	74
7.7	Визначення ключових факторів успіху конкретного проєкту.....	75
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	76
8.1	Вступ.....	76
8.2	Шкідливі і небезпечні фактори при роботі з комп'ютером.....	77
8.3	Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	78
8.4	Розробка заходів з умов поліпшення охорони праці.....	80
8.5	Розрахункова частина	81
9	ОСНОВНІ ВИСНОВКИ.....	84
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	86

КБПЗ-2023

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ІТ	–	інформаційні технології
ПЗ	–	програмне забезпечення
ЦОД	–	центр обробки даних
QoS	–	механізми контролю й керування якістю

КБПЗ_2025

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. Останнім часом значно активізувалися зусилля по реалізації хмарних сервісів, прагнучи відвоювати лідируючі позиції в конкурентів, які раніше прийшли на цей ринок. Cloud Platform надає хмарні інструменти для мобільної розробки, замовленого програмування й модернізації наявних додатків. Хоча Cloud підтримує всі типи хмарних сервісів: IaaS, PaaS і SaaS, – важливе значення надається взаємодії з розроблювачами.

Розходження між інфраструктурою як сервісом і платформою як сервісом можна визначати по-різному. Говорячи із замовниками, я звичайно не акцентую на цьому їхня увага. Однак певна – невелика – різниця зберігається. Ключове слово – автоматизація. У випадку IaaS замовник має можливість більшого контролю, але й залишає за собою більшу відповідальність. Наприклад, він повинен сам установити й зконфігурувати ПЗ, що збирається перенести в хмарну інфраструктуру.

При використанні ж PaaS (у всякому разі, це справедливо у відношенні Cloud Platform) забезпечується високий ступінь автоматизації надання ресурсів, середовища розробки й процесів. Автоматизація розгортання додатків, установки відновлень, резервного копіювання, відновлення й масштабування – все це атрибути платформи як сервісу, які далеко не завжди застосовні до IaaS.

Cloud Platform розрахована на різні категорії клієнтів. Очевидно, що розроблювачі – одна з них, але цільовим сегментом є також ІТ-менеджери, інтегратори додатків і інші користувачі.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління.

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

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

– Огляд існуючих систем інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління.

– Дослідження системи інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління.

– Програмна реалізація системи інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління.

Об'єктом дослідження є процес інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління.

Предметом дослідження є методи інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління.

Методи дослідження базуються на методах хмарних технологій, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління.

– Розроблено вітчизняний продукт інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління.

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.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти LV науково-технічної конференції «Наука в ЦНТУ: основні досягнення та перспективи розвитку» (2025 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 інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ – 2025

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Якщо говорити послідовно, розроблювачам потрібна гнучкість при реалізації проектів за рахунок використання переваг передових рішень, втілених у хмарі. Вони хочуть одержати їх без необхідності встановлювати, конфігурувати й обслуговувати всі необхідні процеси й технології. Керівники IT-підрозділів зацікавлені в переході від моделі власного ЦОДу до моделі ЦОДу, що функціонує в хмарі. При цьому додатки переносяться в хмару для підвищення їхньої динамічності, скорочення витрат і прискорення інновацій. Крім того, багато замовників беруть на озброєння хмарну платформу для інтеграції додатків, а аналітикам даних Cloud пропонує засоби для інтелектуального аналізу даних і технології Великих Даних прямо із хмари.

Ще одна категорія – співробітники служби безпеки, які ніяк до розроблювачів не ставляться. Їм необхідні інструменти для консолідації фрагментованої ідентифікаційної інформації. Кожний застосунок працює зі своїми ідентифікаційними даними: SaaS – з одними, внутрикorporативні додатки – з іншими, а хмарні від сторонніх постачальників – із третіми. Службі безпеки потрібно консолідувати всі ці розрізнені ідентичності в одній уніфікованій моделі, і наша хмара їм буде корисно.

Таким чином, серед потенційних клієнтів Cloud Platform не тільки розроблювачі, але й інші категорії користувачів, які бачать вигоду від застосування хмарних рішень. Я не згадав ще підприємства з різних сфер діяльності, не пов'язаних з IT. Багато компаній хочуть мати інструменти, за допомогою яких вони могли б створювати додатка, навіть якщо в них немає команди програмістів.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

1.2 Область застосування

Мережна інфраструктура дозволяє забезпечити передбачувані продуктивність і мережну затримку. Вона реалізована таким чином, що між будь-якими двома її обчислювальними елементами є тільки два транзитних вузли. У випадку AWS або Azure такий рівень передбачуваності недосяжний, і це позначається на продуктивності.

Чимало інвестовано в забезпечення замовникові можливості вибору варіантів використання інфраструктури. Це можуть бути віртуальна машина, як в AWS або Azure, або виділене середовище, призначене винятково для конкретного клієнта, або «голий» сервер (bare metal) без якого-небудь передвстановленого ПЗ. Інакше кажучи, замовник може встановити будь-яку ОС або скористатися всією обчислювальною потужністю цієї «залізної» інфраструктури.

Інвестовано в побудову зовсім нової інфраструктури зберігання даних. Накопичувачі NVMe, що підключаються прямо до системної плати, забезпечують кращу продуктивність і меншу затримку при доступі до даних. Таким чином, за допомогою передових рішень в області обчислень, зберігання й передачі даних ми вже створили інфраструктуру наступного покоління. У нас дуже агресивна політика збільшення кількості ЦОДів. У цей час їх налічується більше 20, у найближчий рік з'являться ще трохи.

Замовник одержує передбачувану продуктивність. Крім того, замовник може використовувати інфраструктуру за своїм розсудом. Один із цих ступенів волі, що унікально для Cloud, – можливість використовувати інфраструктуру й/або платформу як у публічній хмарі, так і на своїй території. Концепція «хмара в клієнта» (Cloud at Customer) припускає, що всі переваги нашої пропозиції інфраструктури як сервісу доступні в ЦОДі замовника. І він може використовувати ті ж самі сервіси й технології, які доступні в публічній хмарі, але при цьому вони перебувають за міжмережним екраном замовника. Це унікальна пропозиція на ринку, у конкурентів такої реалізації немає. Оскільки в

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

публічному й частці хмарах технології однакові, замовник може вибирати, що він хоче зберегти в себе і які дані перенести в хмару, а потім безперешкодно переміщати їх туди й назад.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ_2025

					VKPM-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

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

Платформи керування хмарою (CMP) допомагають компаніям контролювати та керувати своєю хмарною інфраструктурою. Вони забезпечують централізований інтерфейс для керування ресурсами, відстеження використання та забезпечення дотримання політик у хмарних середовищах. CMP особливо корисні для організацій, які використовують кілька хмар або складні розгортання. Їхня головна мета – покращити ефективність, прозорість та контроль витрат.

Що роблять платформи керування хмарою?

Платформи керування хмарою (CMP) надають низку інтегрованих функцій, що усувають розрив між керівництвом, інженерією та фінансами, забезпечуючи спільне управління та оптимізацію хмарних ресурсів. Коротко окреслимо ключові варіанти використання програмного забезпечення для керування на основі хмари:

Випадки використання CMP у фінансовій сфері:

- Відстеження та оптимізація витрат на хмарні ресурси.
- Бюджетування, встановлення та забезпечення дотримання лімітів витрат у хмарних середовищах.
- Забезпечення прогнозної аналітики для визначення тенденцій витрат на хмарні технології та зниження витрат.
- Сприяння детальному виставленню рахунків та розподілу витрат між відділами для розбиття витрат за значущими категоріями, такими як команда, продукт, функція, клієнт тощо.
- Надання єдиних інформаційних панелей для аналізу витрат у різних

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

постачальників хмарних послуг.

Інженерні варіанти використання СМР:

- Отримання інформації про відповідні витрати – наприклад, менеджер з інженерії може контролювати щоденні витрати своєї команди, або DevOps-інженеру може знадобитися інформація про погодинні витрати на певні ресурси.
- Ефективний доступ, забезпечення та розподіл хмарних ресурсів.
- Автоматизація завдань управління хмарою для економії цінних годин інженерії.
- Підтримка продуктивності хмарних сервісів, робочих навантажень і програм за допомогою автоматичного масштабування та інших стратегій.
- Сприяння дотриманню вимог хмарних технологій у режимі реального часу та управлінню ними.

Варіанти використання СМР у лідерстві:

- Отримання єдиного уявлення про продуктивність та витрати хмарних технологій в організації.
- Прийняття рішень на основі даних щодо хмарної стратегії та інвестицій.
- Забезпечення узгодженості між ІТ-операціями та бізнес-цілями.
- Демонстрація рентабельності інвестицій у хмарні ініціативи.

Які переваги платформ управління хмарними ресурсами?

Ось деякі з найважливіших причин, чому вам може знадобитися платформа керування хмарою:

- Економія часу. Складність робочих навантажень, програм, послуг та допоміжної інфраструктури зростає, а це означає, що максимізація цінності ваших хмарних ресурсів стає більш трудомісткою та складною. Автоматизація може допомогти інженерним командам швидко позбутися ручних завдань, що звільнить їхній час для розробки та впровадження інновацій.
- Економія коштів : Інструменти керування хмарою можуть допомогти з багатьма можливостями економії коштів, такими як керування знижками та зобов'язаннями щодо закупівель, вибір типів екземплярів, оптимізація робочих

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

навантажень, визначення вільних ресурсів, оптимізація розмірів тощо.

– Прозорість : Допомагаючи з тегами, поверненням платежів, поверненням коштів, розподілом витрат тощо, платформи керування хмарою можуть допомогти вам отримати необхідну прозорість, що дозволить вам легше відстежувати аномалії, розуміти центри витрат, прогнозувати, складати бюджет тощо.

– Швидші інновації . Усунення хмарних відходів може допомогти організаціям значно заощадити на хмарних витратах, негайно звільняючи ресурси, які можна було б використовувати для інших цілей.

– Підвищена продуктивність . Програмне забезпечення для керування хмарними ресурсами допомагає виявляти проблеми та вузькі місця, підвищуючи ефективність та скорочуючи час, витрачений на усунення несправностей. Воно також допомагає підвищити доступність програм.

Щоб досягти оптимальної продуктивності та економії коштів, вкрай важливо вибрати відповідне програмне забезпечення для керування хмарою відповідно до ваших потреб. Давайте розглянемо деякі з найкращих платформ для керування хмарою.

nOps

nOps – це універсальна платформа для оптимізації витрат AWS, яка допомагає користувачам автоматично скоротити витрати до 60%. nOps спрощує розподіл витрат на AWS та забезпечує повну видимість витрат. Вона також інтелектуально автоматично керує всіма вашими знижками на обчислення та ціни, щоб ви отримали оптимальну продуктивність та витрати.

nOps було створено, щоб полегшити інженерам вжиття заходів щодо оптимізації хмарних технологій, і нещодавно було названо №1 у категорії управління витратами на хмарні технології від G2.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

сховища тощо.

– AWS MAP Tracker : максимізує фінансування MAP, автоматично позначає ресурси та відстежує кредити для ефективної міграції в хмару.

– Огляд добре проєктованої структури : автоматизує оцінку та звітність WAFR для оптимальної хмарної архітектури

IBM Cloud Orchestrator

IBM Cloud Orchestrator допомагає вам керувати хмарною інфраструктурою, забезпечуючи комплексне розгортання послуг на всіх рівнях інфраструктури та платформи. Крім того, Cloud Orchestrator підтримує швидкий та безпечний моніторинг, керування та резервне копіювання вашого хмарного середовища.

IBM Cloud Orchestrator прагне забезпечити узгоджений, гнучкий та автоматизований спосіб інтеграції хмари з політиками, процесами та інфраструктурами центрів обробки даних клієнтів у різних ІТ-доменах. Він надає набір інструментів для визначення та впровадження бізнес-правил та ІТ-політик, щоб узгоджувати автоматизовані та ручні завдання в складних хмарних середовищах.

Основні характеристики:

- Автоматизація конфігурації хмари.
- Адміністрування хмарних сервісів.
- Статистика використання хмари.
- Панелі для керування витратами керівників.

Apache CloudStack

Apache Cloudstack – це універсальна платформа хмарних обчислень з відкритим кодом. Вона пропонує потужні можливості для керування віртуальними машинами як високодоступна та масштабована платформа IaaS.

Одна з головних переваг CloudStack криється в назві: вона прагне включити весь «стек» функцій, необхідних для інфраструктури як послуги (IaaS), включаючи оркестрацію обчислень, мережу як послугу, управління

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

рекомендації.

Основні характеристики:

- Управління використанням ресурсів та потужностей.
- Виявлення аномалій на основі штучного інтелекту.
- Адміністрування ІТ-операцій.
- Угоди про рівень обслуговування (SLA) щодо продуктивності.

Morpheus Data

Morpheus Data – це платформа для управління та оркестрації хмарних ресурсів. Спочатку розроблена для фахівців з DevOps, вона пропонує функції для побудови інфраструктури додатків, управління гібридними хмарами, автоматизації робочих процесів управління хмарою, оптимізації витрат на хмарні ресурси та багато іншого.

Morpheus має багато потужних функцій для керування віртуальними машинами, контейнерами та життєвим циклом розробки додатків. Він також інтегрується з багатьма поширеними інструментами DevOps, такими як Terraform, Git, Ansible, Jenkins, Puppet, Chef та іншими.

Основні характеристики:

- Автоматичне ведення журналу.
- Моніторинг та сповіщення хмарних сервісів.
- Налаштування одним клацанням миші.
- Інструменти аналітики та звітності FinOps.

CloudHealth від VMware

CloudHealth дозволяє користувачам керувати своїми хмарними витратами, використанням, продуктивністю та безпекою через єдиний інтерфейс. З моменту придбання VMWare, партнерська програма CloudHealth використовує платформу, щоб допомогти партнерам VMWare керувати своїми хмарними витратами, підвищувати ефективність та монетизувати свій бізнес у сфері публічної хмари.

Деякі з переваг CloudHealth включають його комплексний набір інструментів для управління, аналізу та оптимізації хмарної інфраструктури та

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

витрат, а також можливості роботи з кількома хмарами. Однак деякі вважають, що його можливостям звітності бракує налаштування та деталізації.

Основні характеристики:

- Підходить для складної хмарної архітектури.
- Оптимізація інфраструктури для зниження витрат.
- Моніторинг хмарних середовищ AWS, Azure та GCP.
- Функції безпеки та відповідності.

Terraform

Terraform – це інструмент з відкритим вихідним кодом для роботи з інфраструктурою як кодом (IaC), розроблений HashiCorp. Це один з найпопулярніших інструментів для хмарної інфраструктури та DevOps, доступних сьогодні. Terraform дозволяє користувачам безпечно та ефективно створювати, змінювати та версіонувати хмарну інфраструктуру за допомогою автоматизованого забезпечення та управління ресурсами.

Модульність Terraform, велика спільнота та розгалужена екосистема модулів роблять його переконливим вибором для автоматизації забезпечення інфраструктури, IaC та розробки хмарних додатків.

Основні характеристики:

- Кодифікація та забезпечення багатохмарної архітектури.
- Віддалене керування станом.
- Інтеграція контролю версій.
- Керування Kubernetes, мережевою інфраструктурою, віртуальними образами та іншим.

OpenStack

OpenStack – це проект програмного забезпечення з відкритим кодом для хмарних обчислень. Він допомагає керувати вашими обчислювальними, сховищними та мережевими ресурсами за допомогою API або панелі інструментів.

Окрім стандартної функціональності інфраструктури як послуги (IaaS),

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

додаткові компоненти забезпечують оркестрацію, управління відмовами та управління послугами (серед інших послуг) для забезпечення високої доступності користувацьких застосунків.

Основні характеристики:

– Nova: повний інструмент керування та доступу до обчислювальних ресурсів.

– Neutron: Мережа як послуга.

– Swift: зберігання об'єктів.

– Шлак: блокове сховище.

– Keystone: автентифікація.

– Огляд: Управління віртуальними машинами.

CloudBolt

Гібридна платформа керування хмарою CloudBolt дозволяє ІТ-відділам підприємств створювати, розгортати та керувати приватними та публічними хмарами. Її рішення для управління та автоматизації допомагає виділити, керувати та оптимізувати ресурси різних постачальників хмарних послуг.

Він пропонує інструменти управління хмарними ресурсами, що охоплюють низку функцій, таких як оптимізація витрат на хмарні ресурси, автоматизація, панелі інструментів та звітність, безпека та управління, а також виставлення рахунків.

Основні характеристики:

– ІТ-системи самообслуговування.

– Управління кількома хмарами та гіпервізорами.

– Служба розгортання та виявлення об'єктів, що перебувають у занедбаному стані.

– Управління життєвим циклом та оркестрація.

– Прозорість витрат та стан послуг.

CloudCheckr

CloudCheckr – це комплексна платформа для управління хмарними

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

ресурсами. Вона забезпечує прозорість та аналітичні дані, що допомагають вам знизити витрати, підтримувати безпеку та відповідність вимогам, а також оптимізувати хмарні ресурси.

Якщо ви хочете скористатися перевагами AWS Spot, CloudCheck пропонує багато продуктів, орієнтованих на спотові технології, таких як Spot Eco (оптимізація витрат у хмарі), Spot Ocean (контейнери), Spot Elastigroup (інфраструктура) та Spot Portfolio (CloudOps).

Деякі з переваг CloudCheckr включають інструменти для забезпечення видимості та моніторингу, системи сповіщень та комплексні можливості. З іншого боку, деякі вважають, що для початківців крива навчання досить складна, а деякі з його можливостей звітності про витрати можуть містити неточності.

Основні характеристики:

- Управління відповідністю (AWS WAFR).
- Хмарні сервіси та виставлення рахунків.
- Менеджер з використання ресурсів.
- Оптимізація витрат та прозорість.

Flexera

Місія Flexera полягає в покращенні прозорості, розподілу та ефективності витрат на хмарні технології у великих масштабах. Вона надає практичні рекомендації, засоби контролю бюджету та політику витрат, щоб допомогти вашій організації уникнути несподіванок та зменшити непотрібні витрати на хмарні технології.

Однією з його переконливих переваг є інструменти автоматизації, які він надає для реагування на рекомендації, що робить вашу оптимізацію та управління хмарою більш масштабованими та ефективними.

До недоліків Flexera належать відносно прості панелі інструментів, можливості моніторингу та звітності. Деякі вважають, що ці функції потребують подальшої розробки та документування, щоб розкрити свій повний потенціал.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

Основні характеристики:

- Спільний механізм нормалізації у всіх додатках Flexera.
- Сучасний користувацький досвід та навігація.
- Побудовано на сучасній мікросервісній архітектурі, яка забезпечує швидшу обробку.
- Автоматизоване управління на основі політик.
- Уніфікований огляд ресурсів публічної та приватної хмари.
- Відкрита архітектура API для інтеграції з іншими бізнес-системами.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – це об'єктно-орієнтована мова програмування високого рівня загального призначення з відкритим кодом. Це визначення може бути важким для новачків, тому розглянемо кожну характеристику окремо, щоб зрозуміти, що вона означає:

- Відкритий вихідний код: це безкоштовно та доступно для подальших покращень, таких як додавання корисних функцій або виправлення помилок.
- Об'єктно-орієнтована: заснована не на функціях, але в об'єктах з певними атрибутами й методами.
- Високий рівень: зручний для людини, а не для комп'ютера.
- Загальне призначення: можна використовувати для створення будь-яких програм.

Ця мова використовується в будь-якому програмному забезпеченні, про яке ви тільки можете подумати. Ви можете використовувати його для створення веб-сайтів, штучного інтелекту, серверів, програмного забезпечення для бізнесу та багато іншого. Також застосовується в науці про дані, аналізі даних, машинному навчанні, інженерії даних, веб-розробці, розробці програмного забезпечення та інших галузях.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Переваги та недоліки Python

Переваги:

– Її легко читати, вчити та писати. Це мова програмування високого рівня з англійським синтаксисом. Це полегшує читання та розуміння коду. Її дійсно легко зрозуміти і вивчити, тому багато людей рекомендують Python новачкам. Вам потрібно менше рядків коду для виконання того ж завдання в порівнянні з іншими основними мовами, такими як C/C++ та Java.

– Підвищує продуктивність. Це дуже продуктивна мова. Завдяки її простоті розробники можуть зосередитися на розв'язанні проблеми. Їм не потрібно витрачати багато часу на розуміння синтаксису або поведінку мови програмування. Ви пишете менше коду та виконуєте більше завдань.

– Інтерпретована мова. Python мова, що інтерпретується, а це означає, що вона безпосередньо виконує код по рядку. Якщо сталася помилка, вона зупиняє подальше виконання та повідомляє про її виникнення. Вона показує лише одну помилку, навіть якщо у програмі їх кілька. Це спрощує налагодження.

– Динамічно типізована. Python не визначає тип змінної, доки ми не запустимо код. Вона автоматично надає тип даних, коли відбувається процес виконання. Фахівець може не турбуватися про оголошення змінних та типи даних.

– Безкоштовна та з відкритим вихідним кодом. Ця мова постачається під схваленою OSI ліцензією з відкритим вихідним кодом. Це робить його безкоштовним для використання та розповсюдження. Ви можете завантажити вихідний код, змінити його та навіть розповсюджувати свою версію. Це корисно для організацій, які хочуть використати свою версію для розробки.

– Підтримка великих бібліотек. Стандартна бібліотека Python є величезною, ви можете знайти майже всі функції, необхідні для вашого завдання. Таким чином ви не залежите від зовнішніх бібліотек.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

– Для веброзробки. У той час як для створення візуальної частини вебсайту ми переважно будемо використовувати такі мови, як HTML, CSS та JavaScript, для його невидимої частини ми часто вибираємо Python. Серед масштабних вебсайтів та програм, створених за допомогою цієї мови, варто згадати Google, Facebook, Instagram, YouTube, Dropbox та Reddit.

– Для автоматизації задач/скриптингу. Це відмінний інструмент для написання програм для автоматизації різних завдань, що повторюються. Цей процес називається скриптингом. Зокрема, можна робити скрипти для роботи з файлами та папками. Наприклад, можна створювати, перейменовувати, перетворювати, розділяти, об'єднувати або видаляти файли, перевіряти їх наявність помилок. Ви також можете використовувати автоматизацію Python для пошуку та завантаження інформації з Інтернету, заповнення та надсилання онлайн-форм та надсилання регулярних повідомлень або електронних листів.

Яким фахівцям потрібно володіти Python:

- Фахівець з даних.
- Аналітик даних.
- Інженер даних.
- Інженер з машинного навчання.
- Журналіст даних.
- Архітектор даних.
- Повний стек веб-розробника.
- Backend-розробник.
- DevOps-інженер.
- Інженер-програміст.

Можемо зробити висновок, що Python ще довго буде популярною мовою, хоч і має низку недоліків. Цю мову використовують для створення вебсайтів, штучного інтелекту, серверів, програмного забезпечення для бізнесу, аналізу даних, машинного навчання, інженерії даних та для багатьох інших областей. Це перспективна і затребувана навичка, яка необхідна у всіх галузях.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

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 інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління.

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

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.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Замовники можуть споживати інфраструктуру як сервіс і платформу як сервіс або з публічної хмари, або зі свого ЦОДу. В останньому випадку надається в оренду все необхідне встаткування й управляємо сервісами точно так само, як у публічній хмарі, тобто замовник одержує ті ж самі ресурси й функції в себе в ЦОДі.

Портфоліо Cloud at Customer включає кілька реалізацій. Наприклад, Cloud Machine надає інфраструктуру й платформу як сервіс. Якщо замовник хоче одержати Exadata як сервіс, ми пропонуємо йому Exadata Cloud Machine і надаємо повний контроль над цими сервісами. За допомогою захищеного шлюзу інженери можуть установлювати необхідні відновлення, але даних клієнтів їм не видні. Вся інформація в нашій хмарі шифрується (при зберіганні, при передачі або в пам'яті), а ключі шифрування відомі тільки замовникові.

Незважаючи на те що в інженерів немає доступу до даних, деякі підприємства не хочуть, щоб до їхнього середовища можна було підключитися ззовні. Із цієї причини ми пропонуємо підтримку так званого відключеного режиму (disconnected mode). Це означає, що регіональний інженер регулярно відвідує замовника для установки відновлень, тобто замість проведення операцій віддалено наші інженери виконують їх на місці. Очевидно, що така послуга обходиться дорожче.

Все залежить від того, що розуміти під галузевою хмарою. Якщо мова йде про специфічний різновид додатків, то ми дуже багато робимо в цьому напрямку. пропонує саме різне програмне забезпечення у вигляді сервісу, призначене для конкретних галузей: телекомунікаційних компаній, роздрібного й фінансового секторів, а також для ряду інших напрямків.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

Інший підхід – швидше за все, ви мали на увазі саме його – передбачає наявність окремого ЦОДу для конкретної галузі. У нас є, наприклад, урядова хмара, що відповідає вимогам державних організацій. Є фінансова хмара, побудована з урахуванням відповідних стандартів, зокрема PCI DSS.

Вибір технологій залежить від особливостей законодавства тої або іншої країни. Наприклад, у США державний сектор дуже жорстко регулюється такими нормами, як FedRAMP, тому ми повинні були створити окремий ЦОД для відповідності цим вимогам. Так само ми надходимо й в інших регіонах, де урядове регулювання дуже тверде і є критична маса користувачів, достатня для створення окремої хмари.

Від інших технологій Інтернет речей відрізняється насамперед обсягом даних. Їхня обробка в реальному часі вимагає величезних обчислювальних потужностей.

Для Інтернету речей пропонує різні рішення. Так, розроблений хмарний сервіс IoT Cloud Service, за допомогою якого можна реєструвати різні пристрої, здійснювати безпечні комунікації з ними й управляти двосторонньою взаємодією, а крім того, у реальному часі обробляти всю вступник від них інформацію. Можна аналізувати ці дані, витягати з них важливі відомості й управляти з їхньою допомогою бізнесами-процесами.

З обліком генеруємого обсягу даних і подій разом з IoT часто використовуються технології Великих Даних, а виходить, крім обробки в реальному часі, потрібна можливість поміщати інформацію в конкретні озера даних. Відповідне середовище дуже важко правильно масштабувати, тоді як еластичність хмари дозволяє починати з малого й нарощувати ресурси в міру необхідності. Ми пропонуємо цілий спектр хмарних сервісів для Великих Даних. Так, Big Data Preparation Cloud Service дозволяє підготувати дані для переносу в середовище Великих Даних, а Big Data Discovery Cloud Service – витягти всю корисну інформацію.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Інтернет речей висуває високі вимоги до інтеграції із внутрішніми системами. Подій відбувається багато, а реагувати на них треба вчасно. Однак для цього необхідно запуснути бізнес-процес, що з великим ступенем імовірності буде задіяти різні внутрішні системи, тобто IoT-середовище повинна бути тісно взаємозалежна з ними. Ми надаємо сервіси інтеграції в хмару, щоб можна було запускати необхідні бізнес-процеси при настанні події.

Таким чином, IoT не можна розглядати окремо. Інтернет речей породжує безліч супутніх завдань у таких областях, як керування даними, інтеграція із внутрішніми системами, інформаційна безпека й керування IT.

Про прикордонні обчислення говорять усе більше й більше. Однак можливі різні сценарії. Деякі пристрої, маючи достатні інтелектуальні можливості й необхідну обчислювальну потужність, здійснюють обробку подій в автоматичному режимі, але при цьому потрібно контролювати, які дані вони використовують. Цілісне подання про те, що відбувається на таких пристроях, припускає агрегування й централізацію даних, а для цього необхідно хмара.

Маятник розвитку гойдається туди й назад. Спочатку все поєднується, потім розподіляється, потім знову поєднується й т.д. Хмара – це насамперед централізація. Прикордонні ж обчислення припускають значну децентралізацію. Потреба буде зберігатися й у тому і в іншому. Даних і подій стільки, а потреба в їхній обробці росте так стрімко, що перенос частини обчислень на границю досить бажаний, але хмари при цьому не розсіюються, тому що в одних сценаріях краща децентралізація, в інші – централізація.

Контейнери, безумовно, роблять все більший вплив на те, як замовники розробляють і впроваджують нові додатки. Застосування контейнерів підвищує гнучкість рішення, до того ж додатки ізолюються від платформи, на якій вони будуть функціонувати. Контейнери можна розгорнути у власному ЦОДі, хмарі або будь-якому іншому. Гнучкість – одна із причин того, чому контейнери використовуються всі частіше.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

І знову мова йде про волю вибору. У наші дні можливість вибору надзвичайно важлива, оскільки універсального рішення не існує. Я можу виділити три різних підходи до застосування контейнерів. Один з них припускає повний контроль: замовник сам створює контейнери, розгортає їх у хмарі й управляє ними. Ми підтримуємо цю опцію: Docker, Swarm і інші технології, що базуються на Docker, Kubernetes і т.д., можна розгорнути в хмарі й використовувати самостійно. Але такий контроль жадає від замовника більших зусиль.

Інший підхід представлений Container Cloud Service. Замовник сам створює образи контейнерів і розміщає їх у хмарі за допомогою цього сервісу. Хмара відповідає за оркестрацію, розміщення образів, моніторинг і контроль образів Docker. Таким чином, замовникові немає необхідності освоювати всі необхідні технології оркестрації.

Третій розрахований на тих, хто хотів би скористатися всіма перевагами контейнерів, але не знаком з технологією. Для них ми пропонуємо Application Container Cloud Service. Замовник може написати застосунок і розгорнути його в контейнерному середовищі, при цьому йому не прийдеться мати справу з образами Docker. Додатки можуть створюватися за допомогою різних мов програмування й середовищ розробки: node, Java SE, Ruby, Python. Всі середовища працюють на Docker, але розроблювач навіть не знає, що він використовує контейнери.

Таким чином, залежно від того, що замовникові необхідно, він може вибрати кожної із трьох сценаріїв використання. Ми вважаємо, що Docker відмінно підходить для створення нових додатків. Але, крім того, не можна не відзначити потреба в контейнеризації наявних додатків. Всі продукти, які підтримує Oracle, можуть бути розгорнуті в контейнерному середовищі.

Кожний із замовників, з ким мені доводило зустрічатися, займається розробками програмного забезпечення тією чи іншою мірою, оскільки воно лежить у самому серці інновацій і дозволяє компанії виділитися серед

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

конкурентів. Розроблювачів може бути небагато, але тим, що є, необхідно сфокусуватися на інноваціях.

У деяких галузях (роздрібна торгівля, фінанси, телекомунікації) потреба в диференціації вище, ніж в інші. Перераховані галузі перетерплюють серйозну трансформацію, і програмна розробка лежить у її основі. Але перетворення відбуваються всюди. Наприклад, державні інститути розробляють безліч додатків для надання кращих сервісів громадянам з використанням нових технологій і мобільних пристроїв.

Насправді тестування включене в процес розробки – у цьому суть сучасних підходів DevOps, що передбачають автоматизацію тестування й складання програм. Нові стандарти покликані спростити завдання забезпечення безпеки. Наприклад, технологія OAuth одержала широке поширення як спосіб забезпечити керування ідентифікацією в різних застосунках.

З переходом у хмару зростає потреба в консолідації ідентифікаційної інформації. Фрагментація ідентифікаторів породжує серйозну проблему для компаній: користувач може бути представлений в SaaS-застосунках у внутрикorporативних застосунках і в хмарних сервісах третіх сторін. Тому ми створили хмарний сервіс Identity Cloud Service, за допомогою якого ідентифікаційні дані можна консолідувати в хмарі. Крім того, Identity Cloud Service здійснює об'єднання (федерацію) ідентифікаторів з різних додатків і їхнє надання зовнішнім сервісам. Таким чином, досить один раз зареєструватися, щоб одержати доступ до всіх додатків при наявності відповідних прав.

Інший важливий момент, що стосується безпеки в хмарах, – ідея брокера безпеки хмарного доступу (Cloud Access Security Brocker, CASB). Компанії найчастіше не знають, що саме роблять у хмарах їхні співробітники. CASB реєструється у всіх хмарах від вашого ім'я й надає детальні дані про використання хмарних сервісів. При виявленні якого-небудь аномального поведження (наприклад, завантаження понад 20 Мбайт із корпоративної системи CRM) генеруються тривожні повідомлення. Популярність подібних сервісів

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

росте, оскільки використання CASB стає обов'язковою вимогою в багатьох організаціях.

Можна ще довго говорити про те, що робить для реалізації мер захисту на всіх рівнях хмарної платформи. Підкреслю лише, що безпеки приділяється першорядна увага на всіх рівнях – від елементного до рівня керування.

3.2 Розробка структурної схеми

Що таке хмари й коли має сенс будувати хмарні рішення? І якщо будувати, то які платформи використовувати? Чи потрібно надавати клієнтам хмарні сервіси? А може краще використовувати віртуалізацію? І чим відрізняється віртуалізація від хмар? Ці питання задають всі ІТ і «не ІТ» компанії: від великих операторів зв'язку до невеликих стартапів. Давайте спробуємо розібратися! Виникає питання – якщо й те й інше пов'язане з віртуалізацією і мережами, те це вийшли хмари? Адже всі ми прекрасно знаємо, що хмари – це віртуалізація ресурсів десь у мережі. Я спробував знайти формальне визначення хмар, але не знайшов. Визначення були розпливчасті й нечіткі, як справжні хмари. Для початку знайшов стандарт Cloud Computing, що розробляє NIST (National Institute of Standards and Technology). Він виділяє 3 виміру хмарних сервісів:

Хмари

Тобто чіткого визначення хмари немає навіть у стандарті. Розглянемо докладніше всі виміри:

Модель обслуговування або сервісна модель

Наступні поняття напевно всі чули:

– **IaaS (Infrastructure as a Service)** – постачальник надає тільки обчислювальні ресурси (servers, storage, processing, memory, network bandwidth). При цьому може надаватися набір підготовлених образів OS, які розвертаються за кілька хвилин. Також надається мережна зв'язність між віртуальними машинами

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

й додаткові дискові ресурси. Окремою частиною IaaS є MaaS (Metal as a Service) – надання доступу до голого заліза без OS (bare-metal servers). Приклад постачальника IaaS: AWS. Платформи для побудови IaaS: VMware, OpenStack, Azure.

– **PaaS (Platform as a Service)** – постачальник надає операційну систему й певний софт для самостійної реалізації замовником своїх послуг, наприклад, DB, CMS. Приклади: OpenShift, Heroku, Google Cloud Platform, Cloud Foundry.

– **SaaS (Software as a Service)** – постачальник надає софт для використання замовником, наприклад: Microsoft Office 360, Microsoft Exchange і т.п.

Є думка, що справжні хмари це тільки PaaS і SaaS, але стандарт ця думка не підтверджує.

Модель розгортання

Тепер про моделі розгортання, які теж на слуху:

– **Private cloud (Частка хмара)** – інфраструктура розвертається усередині організації, тим самим забезпечується безпека й продуктивність.

– **Public cloud (Публічна хмара)** – інфраструктура призначена для надання послуг стороннім замовникам. Наприклад, AWS, Google Cloud. Переваги отут очевидні: не потрібний обслуговуючий персонал, швидкорозширяємі ресурси, платиш тільки за те, що спожив (години, гігабайти), можливість зосередитися на ідеї або бізнесі, а не на обслуговуванні інфраструктури. Добре підходять для швидкого розгортання невеликих систем, для організації стартапів.

– **Community cloud (Суспільна хмара)** – інфраструктура розділяється між декількома організаціями зі схожими концепціями споживання ресурсів, політиками в безпеці й т.п. Споживачів менше ніж у публічній хмарі, але більше чим у частці. Приклад: Google Apps for Government.

– **Hybrid cloud (Гібридна хмара)** – це комбінація із двох або більше різних хмарних інфраструктур (приватних, публічних або суспільних). Серед гібридних хмар є цікаві кейси використання, наприклад: усередині частки хмари

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

організація зберігає всю необхідну актуальну інформацію, а резервні копії прозоро відправляються в публічну хмару.

Функціональні характеристики

А от на функціональні характеристики хмар найчастіше не звертають увагу, тоді як для хмар це невід'ємні вимоги:

– Broad Access Network – можливість доступу до ресурсів з Інтернету за допомогою різноманітних пристроїв.

– Rapid Elasticity – еластичність ресурсів. При необхідності швидке виділення додаткових ресурсів, аж до нескінченності.

– Measured Services – автоматичний вимір послуг і виставлення рахунків для оплати.

– On-Demand Self Services, Resource Pooling – виділення обчислювальних ресурсів (resource pooling) за запитом від замовників (on demand).

Виходить, щоб Сервіс мав право називатися хмарним, потрібно вміти відповісти на запитання «Де він перебуває по всім трьох вимірах?».

Зворотна сторона хмар

Але при всіх перевагах хмар існує ряд моментів, які можуть мати негативні наслідки, особливо в публічних хмарах:

– Хмарою порівняно легко управляти, але відновлення софту впливає на тисячі користувачів, що потенційно може привести до серйозних аварій.

– У публічних хмарах більші конфігурації не завжди дешевше, треба вважати ТСО.

Виникає кілька питань до безпеки в публічних хмарах:

– Як обробляються конфіденційні дані?

– Чи дотримані вимоги регулятора до персональним даних?

– Наскільки надійна схоронність даних (дублювання, бекапування)?

– Який захист від хакерських атак?

Це всі, звичайно, не привід відмовитися від використання хмар, а причина більш ретельно розібратися в плюсах і мінусах.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

Віртуалізація

Тепер кілька слів про віртуалізацію, без якої не може бути хмар. У віртуалізації також є кілька вимірів, які умовно можна назвати «Тип» і «Спосіб віртуалізації». Я коротко розсаджую про усі.

Типи віртуалізації

Віртуалізація серверів

Звичайно під віртуалізацією мається на увазі розміщення декількох віртуальних серверів у рамках одного фізичного. Але може ще бути об'єднання декількох фізичних серверів в один логічний для рішення певного завдання:

Розподіленість + віртуалізація = GRID системи

Віртуалізація ресурсів

Ресурси – це оперативна пам'ять, жорсткі диски, процесори. Вони також можуть бути нарізані й роздані вроздріб різним користувачам.

Віртуалізація додатків

Віртуалізація додатків – це те, що ми вже знаємо як PaaS і SaaS.

Способи віртуалізації

Повна віртуалізація й паравіртуалізація

Основні методи віртуалізації – це повна віртуалізація й паравіртуалізація. Схема обох методів дуже схожа. Є гіпервізор і віртуальні машини з гостьовими OS. При повній віртуалізації в гостьову OS не вносяться ніякі зміни. При паравіртуалізації встановлюються оптимізовані образи для певного гіпервізору. Це дозволяє максимально використовувати апаратні ресурси й не вимагає ніяких змін від додатків. Прикладом системи, що реалізує повну віртуалізацію, є VMware, приклад паравіртуалізації – Xen і KVM. В KVM особливістю є те, що гостьова OS може бути тільки Linux.

Ще виділяють кілька способів віртуалізації.

Віртуалізація рівня операційної системи

Особливістю є, що гостьова OS може бути тільки одна. Прикладом віртуалізації на рівні OS є Linux-VServer.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

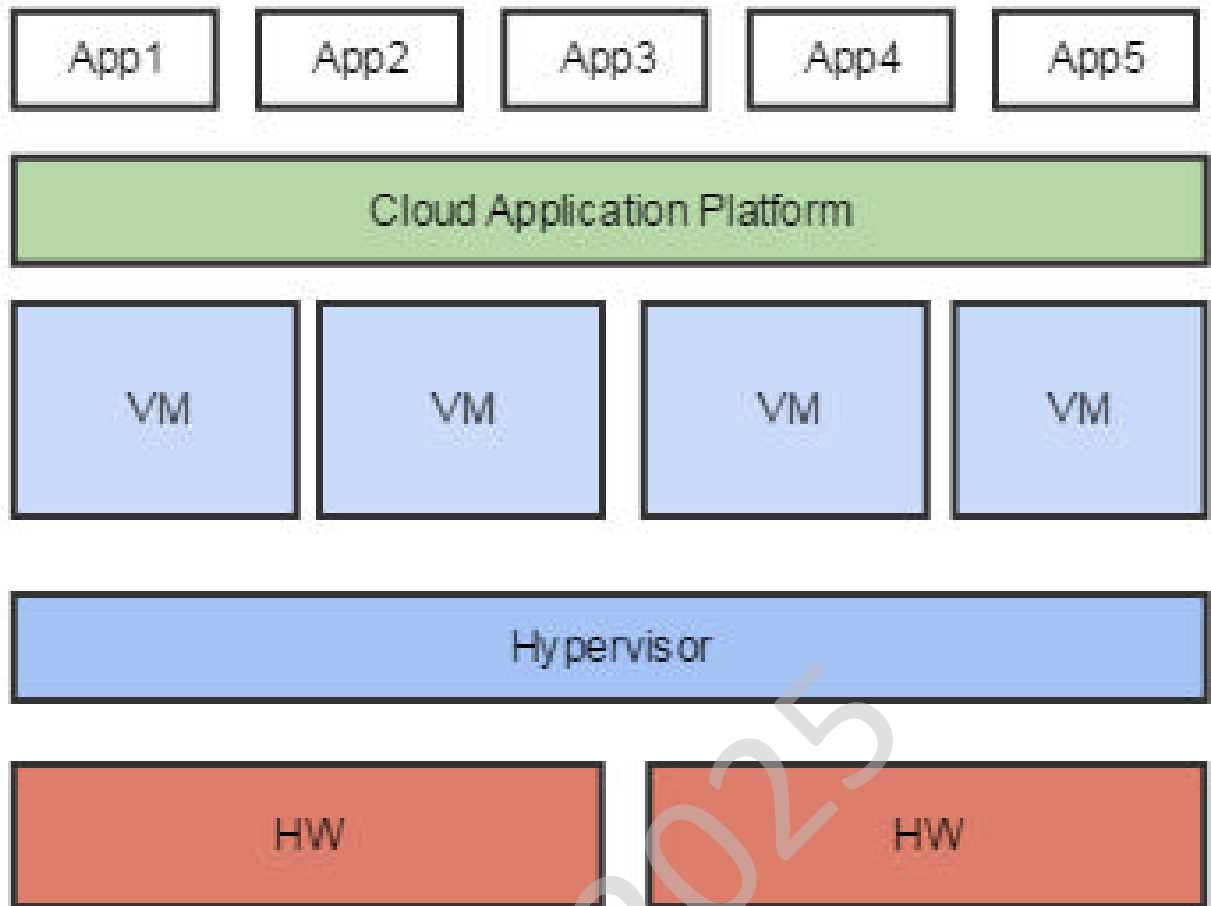


Рисунок 3.1 – Структурна схема системи

Кілька слів про платформи для побудови хмар. Інформації з них маса, тому приведу тільки суху вижимку.

Платформи для побудови хмар

Для побудови хмар зараз є дві основні платформи, інші (Eucalyptus, Cloudstack, Microsoft Azure) займають несуттєву частку.

- VMware – платформа vSphere. VMware зібрала величезну кількість технологій, веде розробку власної версії OpenStack.

- OpenStack – утворився в 2010 році, є комплексом проектів вільного програмного забезпечення, що може бути використаний для створення інфраструктурних хмарних сервісів і хмарних сховищ. Організація OpenStack Foundation (2012) нараховує більш ніж 1000 організацій і більше 30 000 учасників. Основні постачальники стека: Mirantis, RedHat, HP.



Рисунок 3.2 – Функціональна схема системи

При рішенні задачі визначення множини $\mathcal{N}_{\text{баз}}$ шляхів передачі інформації в комп'ютерних мережах ІТ-системи з хмарними службами наступного покоління для ВЗ « i » та « j » з множини \mathcal{R} вузлів зв'язку спочатку необхідно знайти найкоротшу «відстань» (мінімальний час передачі інформаційних пакетів) $T_{i,j \text{ min}}$ від джерела « i » до адресата « j » і множини $S_j^{(i)}$ вузлів, найближчих ВЗ « i » за напрямом руху потоку до

« j » (множина «вузлів-наступників») у порядку рівнів ієрархії дерева допустимих маршрутів множини U .

При рішенні поставленої задачі відомими алгоритмами пошуку найкоротших шляхів в більшості практичних випадків маємо проблему «зациклення» при передачі інформації в знайдених шляхах. Це призводить до збільшення часу передачі інформаційних пакетів, а деколи і до їх втрати. Уникнути «зациклення» при передачі інформації пропонується шляхом додання обмежень (умова постійної відсутності циклів), які надані у вигляді виразів:

$$T_{k,j} \leq T_{i,j \min}; \quad (3.1)$$

$$T_{k,j \min} \leq T_{i,k,j}, \quad k \in R, \quad (3.2)$$

де $T_{k,j \min}$ – найкоротша «відстань» (мінімальний час передачі інформаційних пакетів) від вузла « k » до адресата « j »; $T_{i,k,j}$ – «відстань» (час передачі інформаційних пакетів) від вузла « i » до адресата « j » через вузол « k ».

Рішення задачі пошуку множини шляхів, що виключають «цикли», складається з двох етапів: визначення найкоротшої «відстані» $T_{i,j \min}$ від джерела « i » до адресата « j »; знаходження множини $S_j^{(i)}$ «вузлів-наступників» на маршрутах, що виключають «циклічність», для довільних джерел « i » та адресатів « j » за порядком множини U рівнів ієрархії дерева вибору допустимих маршрутів.

На першому етапі для визначення найкоротшої «відстані» $T_{i,j \min}$ доцільно використати алгоритм розрахунку попередньої топології, який забезпечує вузли зв'язку інформацією про стан зв'язків для обчислення найкоротших шляхів до адресатів. Алгоритм створено на основі відомих алгоритмів стану зв'язків. На відміну від відомих, в цьому алгоритмі враховується ієрархічність побудови комп'ютерної мережі ІТ-системи з хмарними службами наступного покоління (визначаються «відстані» від «вузла-джерела» i до «адресата» j відповідно до існуючих рівнів ієрархії), що надалі дозволить здійснити пропорційний (з урахуванням коефіцієнтів k_s і $k_s^{(c)}$) розподіл інформації по знайдених маршрутах.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

На другому етапі для знаходження множини шляхів передачі інформації, що виключають «циклічність», використовується алгоритм розповсюдження попередньої топології множини шляхів. На відміну від відомих алгоритмів, в яких не враховується поточний стан ВЗ «i» обов'язкова синхронізація обміну службовими повідомленнями про стан зв'язків по всій мережі, в алгоритмі розповсюдження попередньої топології множини шляхів такої синхронізації підлягає тільки один перехід між сусідніми вузлами. Це значно спрощує роботу вузлів зв'язку і скорочує час збіжності алгоритмів. Сукупність алгоритмів розрахунку попередньої топології і розповсюдження попередньої топології множини шляхів є основою способу визначення множини шляхів передачі інформації.

Запропонований спосіб за відсутністю флуктуацій трафіку зіставлений з аналогічними за часом затримки передачі інформації, проте у декілька разів перевершує їх при різких флуктуаціях вхідного трафіку.

Безпосереднє використання всієї знайденої множини $\aleph_{\text{баз}}$ шляхів передачі інформації розробленим способом не завжди є виправданим, особливо у разі високої пропускної спроможності декількох з наявних каналів зв'язку, здатних забезпечити виконання вимог при передачі інформації про повітряну обстановку. Розширення такої множини призводить до збільшення таблиць маршрутизації вузлів зв'язку, ускладнення процесу розподілу інформації і, як наслідок, до зниження достовірності передачі цифрової інформації. Тому виникає необхідність в знаходженні такої топології підмережі, тобто у виборі зі всієї знайденої множини $\aleph_{\text{баз}}$ шляхів деякої (оптимальної) сукупності $\aleph_{\text{об}}$ маршрутів, використання якої в умовах обмежень, що накладаються, дозволить забезпечити максимально можливу достовірність передачі інформації.

Збільшення числа M використаних маршрутів, призводить до істотного (у 2,3...3,4 разів) збільшення ймовірності $q(\aleph_{\text{баз}}, \Delta t)$ спотворення інформації в процесі її передачі. Слід особливо відзначити, що для відомих методів розподілу характерна відсутність моніторингу поточного завантаження маршрутів, змін вхідного потоку інформації, а інколи і технічного стану каналів зв'язку. Принцип рівномірного

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

процедура розподілу інформаційних пакетів за оптимальною множиною маршрутів в комп'ютерних мережах ІТ-системи з хмарними службами наступного покоління.

Проведемо оцінки ефективності методу забезпечення інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління при передачі інформації про повітряну обстановку і вірогідності отриманих результатів. В якості рекомендацій щодо практичного застосування методу забезпечення інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління при управлінні повітряним рухом розроблена структура процесу передачі інформації про повітряну обстановку в режимі реального часу. Врахована необхідність здійснювати комплекс заходів (адаптивне кодування, зменшення статистичної і психовізуальної надмірності, адаптивна маршрутизація та ін.), направлених на управління швидкістю передачі інформації про повітряну обстановку і зниження її інтенсивності. Для компенсації втрачених при передачі в каналах зв'язку інформаційних пакетів і забезпечення безперервності відтворення інформації про повітряну обстановку розроблено алгоритм компенсації втрачених інформаційних пакетів. Це дозволить використовувати для управління передачею цифрової інформації про повітряну обстановку протоколи транспортного рівня (RTP і UDP).

Оцінка ефективності роботи методу забезпечення інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління по відношенню до відомих методів показала ряд його переваг (3-ймовірність доведення інформації до одержувача за час, що не перевищує допустиме значення, вище за аналогічну ймовірність відомих методів до 3 разів, середній час доставки інформаційних пакетів менше аналогічної характеристики відомих методів до 15 разів).

Для обґрунтування достовірності отриманих результатів математичного моделювання проведено імітаційне моделювання передачі інформації в повнозв'язній комп'ютерній мережі в умовах застосування бездротових каналів зв'язку (середня пропускна спроможність $\rho_z = 19 \text{ Кбіт/с}$) при різних інтенсивностях вхідного потоку (у діапазоні $\lambda = [5, \dots, 30] \text{ Кбіт/с}$).

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Під час роботи над магістерською дипломною роботою було створено блок-схеми. Перед їх розглядом необхідно провести роз'яснення який саме тип блок-схем використовується.

Блок-схема це представлення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних образів), які позначають такі елементи, як операції, потік, дані тощо. Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

У інформаційних технологіях функціональна схема складається з функціональних блоків, які являють собою конструктивно відособлені частини (елементи або пристрої) автоматичних систем, які виконують певні функції. 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.

Розглянемо використані підходи управління вимогами це процес запису, аналізу, трасування, пріоритезації і узгодження вимог та контролю змін і доведення до їх зацікавлених сторін. Це безперервний процес протягом всього життя проекту. Вимога – якість, якій мають відповідати результати проекту (продукту або послуги).

Мета управління вимогами полягає в тому, щоб переконатися, що організація відповідає потребам і очікуванням своїх клієнтів, внутрішніх або

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

зовнішніх зацікавлених сторін. Управління вимогами починається з аналізу і виявлення цілей і обмежень організації. Управління вимогами додатково включає в себе підтримку планування вимог, інтеграції вимог і організації роботи з ними (атрибути для вимог).

Управління вимогами передбачає спілкування між членами проектної групи і зацікавленими сторонами, і адаптацію до змін у вимогах протягом всього проекту. Щоб запобігти перетину поля одного класу вимог з іншим, постійні зв'язки між членами команди розробників є критичними. Наприклад, при розробці програмного забезпечення для внутрішнього використання у бізнесу можуть бути настільки сильні потреби, що він може проігнорувати вимоги користувачів, або вважати, що створені сценарії використання покриють також і користувальницькі вимоги.

Відслідковування вимоги фактично означає документування всього життєвого циклу вимоги. Часто необхідно дізнатися першоджерело кожної вимоги. Для цього всі зміни вимог повинні бути задокументовані, щоб досягти стану повного відстеження. Відстежувати треба бути навіть використання реалізованих вимог.

Вимоги мають різні джерела, такі як ділова людина, що замовляє продукт, менеджер зі збуту і фактичний користувач. У всіх цих людей є різні вимоги до продукту. Використовуючи відслідковування вимог, реалізована в системі функція може бути простежена назад до людини або групі, яка замовляла її під час збору вимог. Ця особливість може, наприклад, використовуватися в процесі розробки для пріоритезації вимог, визначаючи, наскільки цінною є дана вимога для певного користувача.

Відслідковування може також використовуватися після розгортання продукту. Наприклад, коли вивчення використання системи показує, що якась функція не використовується, можна визначити навіщо вона була потрібна спочатку.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

Завдання управління вимогами

На кожному етапі процесу розробки існують ключові методи і задачі пов'язані з управлінням вимогами. Для ілюстрації, розглянемо наприклад стандартний процес розробки з п'ятьма фазами: дослідженням, аналізом здійсненності, дизайном, розробкою та тестуванням і випуском.

Дослідження. Під час фази дослідження збираються перші три класи вимог від користувачів, бізнесу і команди розробників. У кожній області задають однакові питання: які цілі, які обмеження, які використовуються процеси та інструменти і так далі. Тільки коли ці вимоги добре зрозумілі, можна приступати до розробки функціональних вимог.

Тут необхідне застереження: незалежно від того, як сильно група намагається це зробити, вимоги не можуть бути повністю визначені на початку проекту. Деякі вимоги змінюються, або тому що вони просто не були знайдені спочатку, або тому що внутрішні чи зовнішні сили торкаються проекту в середині циклу. Таким чином, учасники групи повинні спочатку погодитися, що головна умова успіху – гнучкість у мисленні та діях.

Результатом стадії дослідження є документ – специфікація вимог, схвалений усіма членами проекту. Пізніше, в процесі розробки, цей документ буде важливий для запобігання розповзанню меж проекту або непотрібних змін. Оскільки система розвивається, кожна нова функція відкриває світ нових можливостей, таким чином специфікація вимог прив'язує команду до оригінального бачення системи і дозволяє контрольоване обговорення змін.

У той час як багато організацій все ще використовують звичайні документи для керування вимогами, інші управляють своїми базовими вимогами, використовуючи програмні інструменти.

Ці інструменти керують вимогами використовуючи базу даних, і зазвичай мають функції автоматизації відстеження (наприклад, дозволяючи створювати зв'язки між батьківськими і дочірніми вимогами, або між тестами і вимогами), управління версіями, і управління змінами. Зазвичай такі інструментальні засоби

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

містять функцію експорту, яка дозволяє створювати звичайний документ, експортуючи дані вимог.

Аналіз здійсненності

На стадії аналізу здійсненності визначається вартість вимог. Для користувальницьких вимог поточна вартість роботи порівнюється з майбутньою вартістю встановленої системи. Задаються питання такі як: «Скільки нам зараз варті помилки введення даних?» Або, «Яка вартість втрати даних через помилки оператора пов'язаної з використанням інтерфейсом?». Фактично, потреба в новому інструменті часто розпізнається, коли подібні питання потрапляють до уваги людей, що займаються в організації фінансами.

Ділова вартість включає відповіді на такі питання як: «У якого відділу є бюджет на це?» «Який рівень повернення коштів від нового продукту на ринку?» «Який рівень скорочення внутрішніх витрат на навчання і підтримку, якщо ми зробимо нову, більш просту в використанні систему?»

Технічна вартість пов'язана з вартістю розробки програмного забезпечення та апаратною вартістю. «Чи є у нас потрібні люди, щоб створити інструмент?» «Чи потребуємо ми нове устаткування для підтримки нової системи?»

Подібні питання дуже важливі. Група повинна з'ясувати, чи буде новий автоматизований інструмент мати достатню ефективність аби перенести частину тягара користувачів на систему і зекономити час людей.

Ці питання також вказують на основну суть управління вимогами. Людина і інструмент формують систему, і це розуміння особливо важливе, якщо інструмент – комп'ютер або новий додаток на комп'ютері. Людський розум вкрай ефективний у паралельній обробці та інтерпретації тенденцій з недостатніми даними. Комп'ютерний процесор ефективний у послідовній обробці і точному математичному обчисленні. Основна мета управління вимогами для програмного проекту полягала б у тому, щоб гарантувати, що автоматизована робота призначена «правильному» процесору.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

Наприклад, «не змушуйте людину пам'ятати, де вона знаходиться в системі. Примусьте інтерфейс завжди повідомляти про місцезнаходження людини в системі». Або «не змушуйте людини вводити ті ж самі дані в два екрани. Примусьте систему зберігати дані і заповнювати їх де необхідно автоматично». Результатом стадії аналізу здійсненності є бюджет і графік проекту.

Дизайн

Припускаючи, що вартість точно визначена і переваги, які будуть отримані, є досить великими, проект може перейти до стадії проектування.

На стадії дизайну основна діяльність управління вимогами полягає в тому, щоб перевіряти чи відповідають результати дизайну документу вимог, щоб упевнитися, що робота залишається в межах проекту.

І знову, гнучкість є ключем до успіху. Ось класичний приклад змін проекту, які відмінно працювали. Проектувальники Форда на початку 1980-х очікували, що ціни на бензин піднімуться до 3,18 дол за галон до кінця десятиліття. На середині процесу дизайну автомобіля Ford Taurus, ціни встановилися приблизно на рівні 1,50 дол за галон. Колектив дизайнерів вирішив, що вони могли б створити більший, більш зручний, і більш потужний автомобіль, якщо б ціни на бензин залишилися низькими. Таким чином, вони перепроєктували автомобіль. Коли новий автомобіль вийшов, він встановив загальнонаціональні рекорди продажів.

У більшості випадків, однак, відступ від оригінальних вимог до такої міри не працює. Таким чином документ вимог стає ключовим інструментом, який допомагає команді приймати рішення про зміни дизайну.

Розробка та тестування. На стадії розробки і тестування, основна діяльність управління вимогами – це гарантувати, що робота і ціна залишаються в межах графіка і бюджету, і що створюваний інструмент дійсно відповідає вимогам. Основним інструментом, використовуваним на цій стадії, є створення прототипу і ітераційне тестування. Для програмного додатка користувацький

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

інтерфейс може бути створений на папері і перевірений з потенційними користувачами, в той час як створюється основа програми. Результати цих тестів записуються в керівництві по дизайну користувацького інтерфейсу і передаються колективу дизайнерів. Це економить їх час і робить їх завдання набагато простіше.

Блок-схеми є першоджерелами стратегії розвитку ПЗ. Тому від точності і детальної 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 програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулю системи інтеграції існуючих компонент ІТ-системи, модулю обробки помилок програми і основному модулю.

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 – перевірка поточного стану з завершенням роботи розробленого ПЗ. При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

При розробці ПЗ було використано V-Model (або VEE модель) є моделлю розробки інформаційних систем (ІС), спрямованої на спрощення розуміння складнощів, пов'язаних з розробкою систем. Вона використовується для визначення єдиної процедури розробки програмного забезпечення, апаратного забезпечення та людино-машинного інтерфейсу.

Концепція V-подібної моделі була розроблена Німеччиною та США в кінці 1980-х років незалежно один від одного:

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

Німецька V-модель була розроблена аерокосмічної компанією IAVG в Оттобрунні поряд з Мюнхеном у співпраці з Федеральним департаментом з закупівлі озброєнь в Кобленці, для Міністерства оборони Німеччини. Модель була прийнята німецькою федеральною адміністрацією для цивільних потреб влітку 1992.

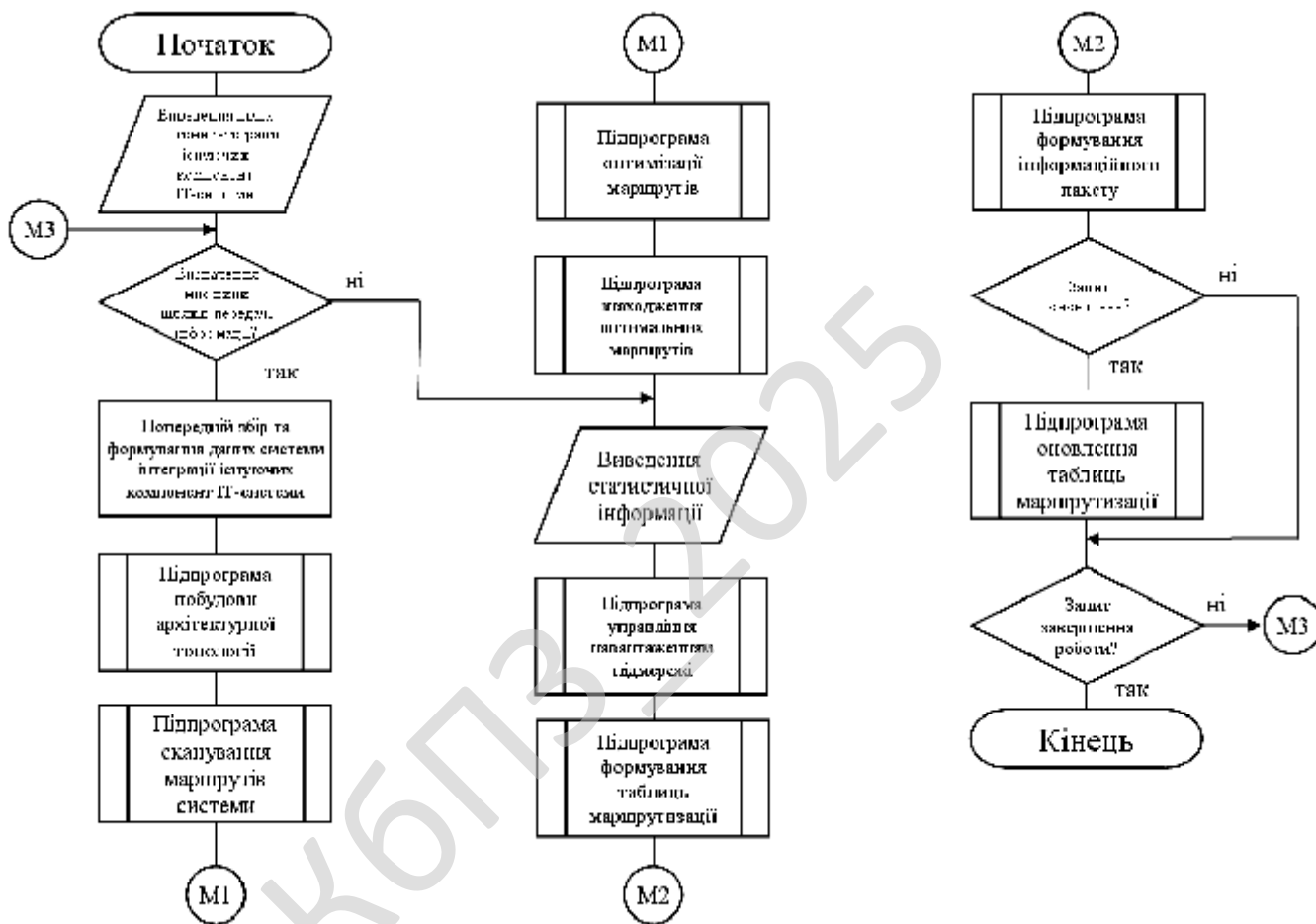


Рисунок 4.1 – Блок-схема основної програми

– Американська V-Model (VEE) була розроблена національною радою з системної інженерії (міжнародна – з 1995 року) для супутникових систем, включаючи обладнання, програмне забезпечення та взаємодію з користувачами.

Сучасною версією V-Model є V-Model XT, яка була затверджена в лютому 2005 року.

Основні принципи

Основний принцип V-подібної моделі полягає в тому, що деталізація проекту зростає при русі зліва направо, одночасно з плином часу, і ні те, ні інше не може повернути назад. Ітерації в проекті виробляються по горизонталі, між лівою і правою сторонами літери.

Стосовно до розробки інформаційних систем V-Model – варіація каскадної моделі, в якій завдання розробки йдуть зверху вниз по лівій стороні букви V, а завдання тестування – вгору по правій стороні букви V. Усередині V проводяться горизонтальні лінії, що показують, як результати кожної з фаз розробки впливають на розвиток системи тестування на кожній із фаз тестування.

Модель базується на тому, що прийнятно-здавальні випробування ґрунтуються, насамперед, на вимогах, системне тестування – на вимогах та архітектурі, комплексне тестування – на вимогах, архітектурі та інтерфейсах, а компонентне тестування – на вимогах, архітектурі, інтерфейсах та алгоритмах

Цілі

V-модель забезпечує підтримку у плануванні та реалізації проекту. В ході проекту ставляться такі завдання:

1. Мінімізація ризиків: V-подібна модель робить проект більш прозорим і підвищує якість контролю проекту шляхом стандартизації проміжних цілей і опису відповідних їм результатів та відповідальних осіб. Це дозволяє виявляти відхилення в проекті і ризики на ранніх стадіях і покращує якість управління проектом.

2. Підвищення та гарантії якості: V-Model – стандартизована модель розробки, що дозволяє домогтися від проекту результатів бажаної якості. Проміжні результати можуть бути перевірені на ранніх стадіях. Універсальне документування полегшує читаність, зрозумілість та контрольованість.

3. Зменшення загальної вартості проекту: Ресурси на розробку, виробництво, управління і підтримку можуть бути заздалегідь прораховані та

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

проконтрольовані. Отримувані результати також універсальні і легко прогножуються. Це зменшує витрати на подальші стадії та проекти.

4. Підвищення якості комунікації між учасниками проекту: Універсальний опис усіх елементів та умов полегшує взаєморозуміння всіх учасників проекту. Таким чином, зменшуються неточності у розумінні між користувачем, покупцем, постачальником і розробником.

Переваги:

– Користувачі V-Model беруть участь у розробці та підтримці V-моделі. Комітет з контролю за змінами підтримує проект і збирається раз на рік для обробки всіх отриманих запитів на внесення змін до V-Model.

– На старті будь-якого проекту V-подібна модель може бути адаптована під цей проект, так як ця модель не залежить від типів організацій та проектів.

– V-model дозволяє розбити діяльність на окремі кроки, кожен з яких буде включати в себе необхідні для нього дії, інструкції до них, рекомендації та докладне пояснення діяльності.

Обмеження

Наступні моменти не враховуються в V-моделі, але можуть бути розглянуті окремо, або можливо адаптувати модель під них:

– Не регулюється розміщення контрактів на обслуговування.

– Організація і виконання управління, обслуговування, ремонту та утилізації системи не враховуються в V-моделі. Однак, планування і підготовка до цих операцій моделлю розглядаються.

– V-подібна модель більше стосується розробки програмного забезпечення в проекті, ніж всієї організації процесу.

Переваги:

– У моделі особливе значення надається плануванню, спрямованому на верифікацію та атестацію розроблювального продукту на ранніх стадіях його розробки. Фаза модульного тестування підтверджує правильність деталізованого проектування. Фази інтеграції та тестування реалізують архітектурне

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

UML може бути застосовано на всіх етапах життєвого циклу аналізу бізнес-систем і розробки прикладних програм. Різні види діаграм які підтримуються UML, і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, так і ділових систем.

Діаграми дають можливість представити систему (як ділову, так і програмну) у такому вигляді, щоб її можна було легко перевести в програмний код. Основною причиною використання мови UML є спілкування розробників між собою.

Крім того, UML спеціально створювалася для оптимізації процесу розробки програмних систем, що дозволяє збільшити ефективність їх реалізації у кілька разів і помітно поліпшити якість кінцевого продукту.

UML прекрасно зарекомендувала себе в багатьох успішних програмних проектах. Засоби автоматичної генерації кодів дозволяють перетворювати моделі мовою UML у вихідний код об'єктно-орієнтованих мов програмування, що ще більш прискорює процес розробки. Практично усі CASE-засоби (програми автоматизації процесу аналізу і проектування) мають підтримку UML. Моделі розроблені в UML, дозволяють значно спростити процес кодування і направити зусилля програмістів безпосередньо на реалізацію системи.

Діаграми підвищують супроводжуваність проекту і полегшують розробку документації.

UML необхідний:

– Керівникам проектів, які керують розподілом завдань і контролем за проектом.

– Проектувальникам інформаційних систем які розробляють технічні завдання для програмістів.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

Агрегація є окремим випадком асоціації і зображується у вигляді простої асоціації з незафарбованим ромбом з боку «цілого». Графічно агрегація представляється порожнім ромбом на блоці класу, і лінією, яка від цього ромба до міститься класу.

Композиція це більш суворий варіант агрегації. Відома також як агрегація за значенням.

Композиція має жорстку залежність часу існування екземплярів класу контейнера та примірників містяться класів. Якщо контейнер буде знищений, то весь його вміст буде також знищено. Графічно представляється як і агрегація, але з зафарбовани ромбиком.

4.2 Захист розробленого програмного забезпечення

Розроблене програмне забезпечення захистимо за допомогою алгоритму захисту інформації RSA. Спочатку необхідно обчислити пару ключів (секретний ключ і відкритий ключ). Для цього відправник електронних документів обчислює два більших простих числа P і Q , потім знаходить їхній добуток $N = P * Q$ і значення функції $\varphi(N) = (P-1)(Q-1)$. Далі відправник обчислює число E з умов $E < \varphi(N)$, НЗД($E, \varphi(N)$) = 1 і число D з умов $D < N$, $E * D \equiv 1 \pmod{\varphi(N)}$.

Пари чисел (E, N) є відкритим ключем. Цю пару чисел автор передає партнерам по переписці для перевірки його цифрових підписів. Число D зберігається автором як секретний ключ для підписування.

Допустимо, що відправник хоче підписати повідомлення M перед його відправленням. Спочатку повідомлення M (блок інформації, файл, таблиця) стискають за допомогою геш-функції $h(-)$ у ціле число m : $m = h(M)$.

Потім обчислюють цифровий підпис S під електронним документом M , використовуючи геш-значення m і секретний ключ D : $S = m \pmod{N}$.

Пари (M, S) передається партнерові-одержувачеві як електронний документ M , підписаний цифровим підписом S , причому підпис S сформований

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

власником секретного ключа D .

Після прийому пари (M, S) одержувач обчислює геш-значення повідомлення M двома різними способами. Насамперед, він відновлює геш-значення m' , застосовуючи криптографічне перетворення підпису S з використанням відкритого ключа E : $m' = S^E \pmod{N}$.

Крім того, він знаходить результат гешування прийнятого повідомлення M з допомогою такої ж геш-функції $h(-)$: $m = h(M)$.

Якщо дотримується рівність обчислених значень, тобто $S^E \pmod{N} = h(M)$, то одержувач визнає пару (M, S) справжньою. Доведено, що тільки власник секретного ключа D може сформулювати цифровий підпис S по документі M , а визначити секретне число D по відкритому числу E не легше, ніж розкласти модуль N на множники. Крім того, можна строго математично довести, що результат перевірки цифрового підпису S буде позитивним тільки в тому випадку, якщо при обчисленні S був використаний секретний ключ D , що відповідає відкритому ключу E . Тому відкритий ключ E іноді називають "ідентифікатором" того, хто підписав.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ системи інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління яке зображено на рисунку 5.1. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

- Навігаційне меню: Файл; Налаштування; Довідка.
- Розділу виведення результату роботи системи – топологія мережі .
- Навігаційного меню яке викликається натисканням правої клавіші манипулятора миші.
- Функціональних кнопок ПЗ.

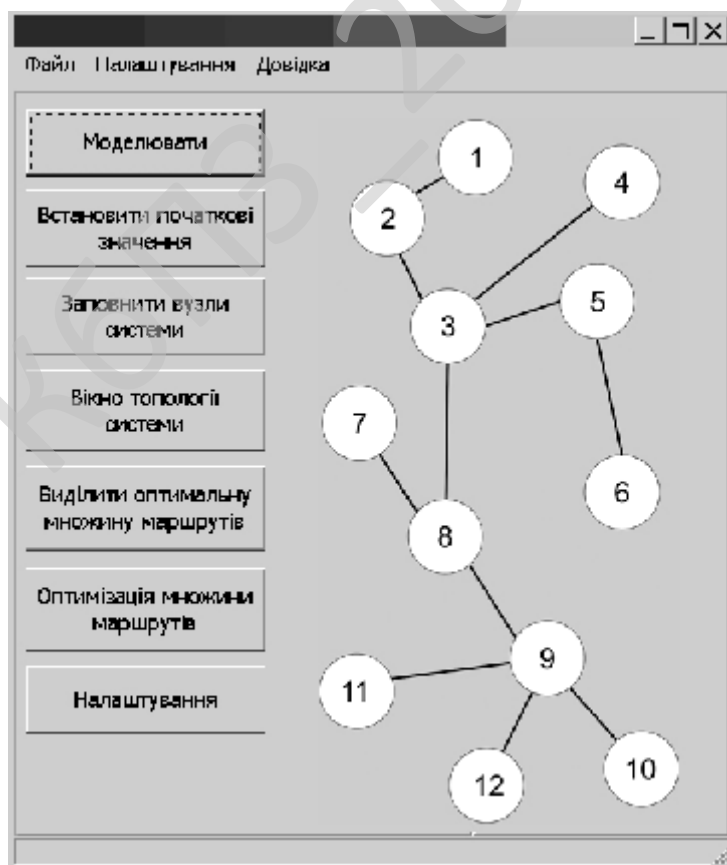


Рисунок 5.1 – Головне вікно ПЗ

Мережна інфраструктура дозволяє забезпечити передбачувані продуктивність і мережну затримку. Вона реалізована таким чином, що між будь-якими двома її обчислювальними елементами є тільки два транзитних вузли. У випадку AWS або Azure такий рівень передбачуваності недосяжний, і це позначається на продуктивності.

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.

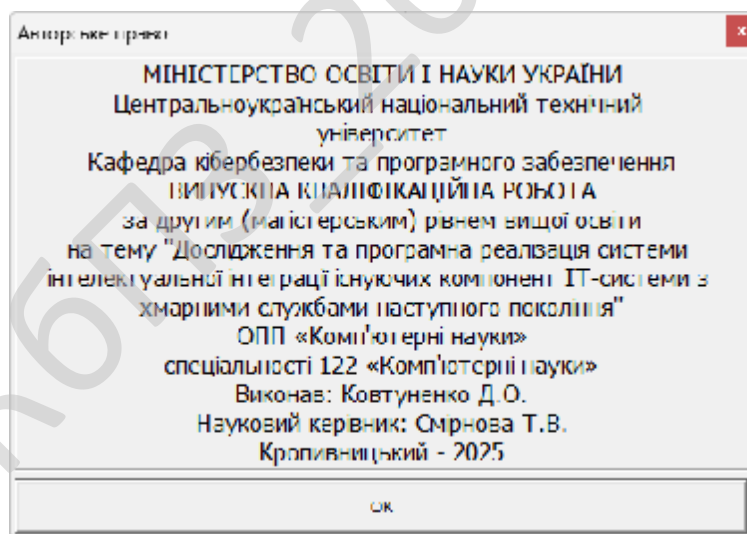


Рисунок 5.2 – Авторське право

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

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

готовою до використання. Даний процес є частинною 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.

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

- Випуск.
- Встановлення та активація.
- Деактивація.
- Адаптація.
- Обновлення.
- 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;

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

– 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 IT 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.

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 форматом білої скриньки засноване на аналізі керуючої структури програми. Програма вважається повністю перевіреною, якщо проведено вичерпне тестування маршрутів (шляхів) її графа управління.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

- 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 імовірністю 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.
- помилок 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.

Обрано умови розповсюдження – Shareware.

Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження комерційного ПЗ на ринку (тобто на шляху до кінцевого користувача), при якому випробувачеві пропонується обмежена за можливостями (не повнофункціональна або демонстраційна версія), терміном дії (тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми.

В угоді про використання (ліцензії для кінцевого користувача, EULA) також може бути обумовлена заборона на комерційне або професійне (не тестове) її використання.

Основний принцип умовно-безплатного ПЗ – «спробуй, перш ніж купити» (try before you buy). ПЗ що поширюється як умовно-безплатний, надається користувачам безоплатно. Звичайно користувач платить тільки за час завантаження файлів через Інтернет або за носій (CD диск, флешку, ключ). Протягом певного терміну, що становить зазвичай тридцять днів, він може користуватися програмою, тестувати її, освоювати її можливості.

Якщо після закінчення цього терміну користувач вирішить продовжити використання ПЗ, він зобов'язаний купити його (zareєstrуватися), заплативши авторові певну суму.

В іншому випадку користувач повинен припинити використання ПЗ та видалити його зі свого комп'ютера.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління.

Метою розробки є дослідження та програмна реалізація системи інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління.

Об'єктом дослідження є процес інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління.

Предметом дослідження є методи інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління.

Методи дослідження базуються на методах хмарних технологій, методах математичної статистики, методах розробки програмного забезпечення.

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

- Удосконалено метод інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління.
- Розроблено вітчизняний продукт інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та розробки системи інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління можуть бути насамперед корисними для підприємств, які мають розгалужену ІТ-інфраструктуру й використовують сервери, мережеве обладнання та корпоративні сервіси для підтримки своєї діяльності. Для таких компаній стабільність і безперебійність роботи інформаційних систем є критично важливими, тому можливість своєчасного виявлення несправностей або перевантажень стає суттєвою конкурентною перевагою. Саме система моніторингу допомагає контролювати роботу мережевих пристроїв у режимі реального часу, виявляючи проблеми ще до того, як вони вплинуть на користувачів. Особливий інтерес до таких систем можуть проявити ІТ-компанії, які займаються наданням послуг хостингу, розробкою програмного забезпечення або підтримкою клієнтів. Для них швидкість реагування на інциденти та якість технічного обслуговування є показниками репутації, а отже, від роботи системи моніторингу залежить рівень довіри клієнтів і лояльність користувачів. Такі підприємства часто працюють у середовищі, де навіть хвилинна затримка чи зупинка сервера призводить до фінансових збитків, тому автоматизація контролю за станом мережі – це не розкіш, а необхідність. Крім комерційних компаній, результати дослідження будуть актуальними для державних структур, освітніх установ і організацій, які мають внутрішні мережі та зберігають великі обсяги інформації. У таких установах впровадження системи моніторингу підвищує ефективність роботи ІТ-відділів, зменшує ризик втрати даних і допомагає раціонально використовувати наявні ресурси.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

Не менш важливим є значення цієї розробки для навчальних і наукових закладів. Вони можуть використовувати систему як навчальну платформу для підготовки фахівців у сфері інформаційних технологій. Студенти отримують можливість не лише спостерігати за реальною роботою системи моніторингу, а й аналізувати дані, моделювати різні ситуації та вчитися реагувати на інциденти. Таким чином, результати дослідження мають універсальний характер і можуть бути впроваджені як у бізнесі, так і в освіті.

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Для оцінки привабливості програмного продукту було проведено експертне опитування серед фахівців у галузі IT-інфраструктури, адміністраторів систем і представників компаній, що мають досвід використання схожих рішень. Експертам було запропоновано оцінити систему за основними критеріями – функціональні можливості, надійність, простота впровадження, масштабованість, вартість експлуатації та потенційна економічна ефективність.

Більшість експертів високо оцінили саме інтелектуальну частину системи – можливість автоматичного сповіщення про інциденти, генерацію аналітичних звітів і прогнозування потенційних відмов обладнання. Особливо було відзначено, що система працює стабільно навіть при великому навантаженні й може адаптуватися до різних типів мережевої інфраструктури, що робить її універсальною.

За результатами оцінки середній рівень привабливості продукту склав 8,7 бала з 10 можливих. Експерти зазначили, що така система може мати великий попит серед середніх і великих підприємств, особливо якщо її вартість залишатиметься конкурентною. Також було підкреслено, що простота інтерфейсу та можливість кастомізації під конкретного користувача є суттєвими перевагами, які підвищують комерційний потенціал рішення.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

Таким чином, метод експертних оцінок показав, що система має високу ринкову привабливість, відповідає актуальним потребам бізнесу та може стати успішним продуктом за умови належного маркетингового просування та підтримки користувачів.

7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості розробки системи інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління доцільно використовувати витратний метод. Він передбачає визначення всіх фактичних витрат, які були понесені під час створення програмного продукту, включаючи оплату праці розробників, витрати на апаратне забезпечення, ліцензії, тестування та впровадження. Такий підхід дозволяє точно визначити базову собівартість проєкту, що є особливо важливим для невеликих команд і стартапів.

Однак, у випадку комерційного впровадження, доцільно поєднати цей підхід із дохідним методом. Дохідний метод дає змогу оцінити майбутні вигоди, які підприємство отримає після впровадження системи. Наприклад, скорочення простоїв серверів, підвищення ефективності роботи персоналу та зменшення витрат на ручну діагностику мережі є прямими джерелами економічної вигоди.

Такий комбінований підхід дозволяє не лише визначити початкову вартість розробки, а й обґрунтувати економічну доцільність проєкту. Він допомагає потенційним інвесторам побачити не просто витрати, а реальні фінансові перспективи, які відкриває впровадження системи.

У результаті використання комбінованої моделі оцінки можна отримати повну картину вартості та окупності проєкту, що стане основою для прийняття управлінських рішень щодо його реалізації чи масштабування.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

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

Компанія має розгалужену ІТ-інфраструктуру, яка включає сервери, мережеве обладнання, робочі станції, системи зберігання даних і корпоративні сервіси. До впровадження системи моніторингу контроль за станом мережі здійснювався вручну: адміністратори виявляли проблеми лише після звернень користувачів або повного виходу сервісів із ладу. Це призводило до простоїв, затримок у роботі та фінансових втрат. Основна мета впровадження системи мережевого моніторингу – забезпечити цілодобове автоматичне відстеження стану обладнання, серверів і додатків, оперативне реагування на інциденти, зниження кількості простоїв і запобігання критичним збоєм у роботі ІТ-інфраструктури. Вхідні дані зафіксовано в таблиці 7.1.

Таблиця 7.1 – Вихідні дані для розрахунку

Показник	До впровадження	Після впровадження	Економічний ефект
Кількість простоїв серверів на рік	20 випадків	5 випадків	-15
Середня тривалість простою одного сервера	4 години	1 година	-3 години
Середні втрати підприємства за 1 годину простою	25 000 грн	5 000 грн	-20 000 грн
Витрати на ручну діагностику й усунення збоїв	300 000 грн/рік	150 000 грн/рік	-150 000 грн
Вартість впровадження системи моніторингу	—	—	450 000 грн
Річні витрати на підтримку системи	—	—	100 000 грн

Наступним етапом є розширення партнерських зв'язків. Доцільно співпрацювати з ІТ-компаніями, які займаються інтеграцією корпоративних систем, адже вони можуть пропонувати продукт своїм клієнтам як частину комплексного рішення. Водночас слід розробити гнучку цінову політику – наприклад, ліцензування за кількістю пристроїв або модель передплати, що зробить продукт доступнішим для малого та середнього бізнесу.

Просування має супроводжуватися технічною підтримкою користувачів, оновленнями та навчанням персоналу. Це створює позитивний досвід використання продукту та сприяє формуванню довгострокових відносин із клієнтами. У підсумку правильна стратегія просування допоможе не лише збільшити продажі, а й побудувати впізнаваний бренд на ринку ІТ-рішень.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Для оптимізації каналів збуту варто поєднати прямі продажі з цифровими платформами розповсюдження програмного забезпечення. Власний сайт компанії може стати не лише вітриною продукту, а й каналом комунікації з клієнтами, де вони зможуть отримати демо-версію, консультацію або підтримку. Це сприятиме зниженню витрат на маркетинг і збільшенню довіри. Додатково ефективним буде впровадження партнерської програми для системних інтеграторів і реселерів, які вже мають доступ до корпоративних клієнтів. Така модель дозволяє розширити охоплення ринку без суттєвих додаткових інвестицій. Також можна запропонувати гібридну форму реалізації: ліцензування для великих компаній і модель SaaS (Software as a Service) для малого бізнесу. Це підвищить доступність системи та дозволить гнучко реагувати на потреби різних сегментів ринку. Ключовим напрямом оптимізації збуту є створення якісного сервісу після продажу – технічна підтримка, регулярні оновлення, аналітичні звіти. Усе це забезпечує стабільність роботи клієнта й стимулює його до подальшої співпраці.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

7.7 Визначення ключових факторів успіху конкретного проєкту

Основним фактором успіху є стабільність і надійність системи. Якщо система моніторингу працює без збоїв і забезпечує реальну користь, вона швидко здобуває довіру користувачів. Технологічна якість продукту, його здатність масштабуватися й інтегруватися з іншими ІТ-рішеннями відіграють ключову роль у його життєздатності.

Другим важливим чинником є професійна команда розробників і технічної підтримки. Клієнти цінують не лише продукт, а й можливість отримати швидко допомогу у випадку проблем або питань. Від рівня компетенції фахівців залежить не лише якість обслуговування, а й довгострокові відносини з партнерами.

Не менш значущим є гнучкість системи – можливість адаптувати її під специфіку кожного клієнта. Різні компанії мають різну інфраструктуру, тому універсальне, але налаштоване рішення стає перевагою.

І, нарешті, успіх будь-якого ІТ-проєкту визначається здатністю постійно вдосконалюватися. Регулярні оновлення, впровадження нових технологій і зворотний зв'язок із користувачами формують довіру й підтримують актуальність продукту на ринку. Саме ці чинники разом створюють основу для стабільного розвитку та комерційного успіху системи моніторингу.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Електронно-обчислювальна машина (ЕОМ) відіграє важливу роль у житті сучасної людини. Кожного дня мільйони людей використовують ЕОМ для пошуку необхідної інформації, спілкуванні у соціальних мережах, перегляду новин, роботи тощо. Багато людей користуються ЕОМ у професійних цілях, оскільки завдяки ЕОМ з'явилося багато нових професій. Тому для розробника хмарних сервісів так важливо розробити зручний інтерфейс для зручного сприйняття інформації, та необхідний функціонал, який буде відповідати необхідним вимогам та навантаженням. Все це вимагає багато часу та великого навантаження з боку розробників. Тому так важливо слідкувати за умовами праці, в яких відбувається робочий процес. Оскільки захворювання можуть бути спричинені надмірним фізичним або розумовим навантаженням, через велику нервово-емоційну напругу, або через виробниче середовище. В даному розділі магістерської роботи проведемо аналіз основних чинників при роботі програміста.

Законом України “Про охорону праці” 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», 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 роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Електронно-обчислювальна машина (ЕОМ) та інше обладнання є джерелами небезпеки ураження електричним струмом. Так як робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. У приміщенні, в якому працюють люди (у т.ч. програмісти) необхідно створити належний мікроклімат, параметри якого регламентуються, Державними санітарними правилами і нормами, зокрема ДСанПіН 3.3.2.007-98.

На роботу програміста впливають наступні фактори: невідповідний мікроклімат приміщення (температура, вологість), недостатня освітленість робочої зони, підвищений рівень шуму та електромагнітного випромінювання, порушення іонного складу повітря, неправильна ергономічна організація робочого місця, ризики, пов'язані із погіршенням зору, порушенням фізичного стану, стресом тощо.

Шкідливими факторами при роботі з персональним комп'ютером є неіонізуюче випромінювання промислової частоти, збільшене нервово-емоційне навантаження на оператора, збільшення навантаження на органи зору та дрібні стереостатичні рухи кінцівок. Ці фактори можуть викликати у працівника певні розлади здоров'я, зокрема підвищення артеріального тиску, кон'юктивіти, тендовагініти та інші захворювання.

Комп'ютер, як і будь-який електричний прилад, особливо при його неправильному підключенні, може бути джерелом ураження оператора електричним струмом. Саме тому всі працівники, які працюють з персональним комп'ютером, повинні мати першу (або другу) групу допуску з електробезпеки.

Через наявність зазначених факторів працівники, які працюють з персональними комп'ютерами, підлягають попередньому та періодичному медичному огляду згідно з пунктом 6.2.3 додатку 4 до наказу Міністерства охорони здоров'я України «Про затвердження Порядку проведення медичних оглядів працівників певних категорій» від 21 травня 2007 року № 246.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Оптимальна температура в приміщенні для праці має становити 20-24°C, відносна вологість – 40-60 %, атмосферний тиск – 750 мм. рт. ст., запиленість не повинна перевищувати 10 мг/м³, швидкість руху повітря – 0,1 м/с.

Через те, що обчислювальна техніка є джерелом тепловиділення, організація мікроклімату потребує додаткових зусиль: кондиціонування, провітрювання, використання систем опалення тощо. Об'єм приміщень повинен передбачатися з урахуванням як мінімум 20 м³ /на особу [4].

Монітори комп'ютерів є джерелом випромінювання, яке може зашкодити здоров'ю людини. Для забезпечення роботи з комп'ютером відстань від монітора повинна становити не менше 50 см, бажано використовувати монітори зі зниженим рівнем, скорочувати час безперервної роботи за комп'ютером (робити п'ятнадцяти хвилинні перерви після кожних півтори години праці). Також в приміщенні необхідно встановлювати іонізатори повітря, використовувати нейтралізатори та зволожувачі.

Комп'ютери та периферійні пристрої є джерелами шуму, висока інтенсивність якого може призвести до проблем з органами слуху та негативно впливати на психологічний стан. Рівень шуму на робочому місці не повинен перевищувати 50 дБА [5]. Для зменшення рівня шуму можна використовувати звукопоглинальні пристрої, а стіни приміщень з комп'ютерами можуть бути покриті звукопоглинальними матеріалами. Поряд із шумом часто виникає вібрація. Для зменшення рівня вібрації в приміщенні на поверхні необхідно встановлювати віброізолятори.

Ергономічні показники робочого місця програміста мають бути наступними: висота робочої поверхні повинна складати 720 мм, розмір поверхні має становити 1600 x 1000 мм; під столом повинен бути простір з розмірами по глибині 650 мм; стіл повинен мати підставку для ніг, розташовану під кутом

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

15° до поверхні; відстань клавіатури від краю столу має бути не більше 300 мм; відстань між очима й екраном повинна складати 40 – 80 см; стілець повинен мати підйомно-поворотний механізм; висота сидіння має регулюватися в межах 400 – 500 мм, глибина – не менше 380 мм, а ширина – не менше 400 мм, висота опорної поверхні спинки має бути не менше 300 мм, ширина – не менше 380 мм. Кут нахилу спинки стільця до площини сидіння повинен змінюватися в межах 90 – 110° [6].

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.

У приміщенні знаходяться наступні джерела шуму: принтер Prinics PicKit M1 Smartphone Photo Printer White, електродвигуни вентиляторів ЕОМ.

Робота програміста передбачає постійний візуальний контакт з моніторами комп'ютерів, та, як наслідок, значне навантаження на зір. Традиційно, це зорова робота високої або середньої точності. Для зорової роботи високої точності загальне освітлення (розподіл світла у всьому об'ємі приміщення) має становити 300 лк, комбіноване освітлення (поєднання загального і місцевого освітлення) – 750 лк. Штучне освітлення повинно бути рівномірним та використовуватися в світлий і темний час доби. Джерелами штучного освітлення можуть слугувати люмінесцентні лампи. Правильне освітлення передбачає уникнення відблисків на екранах.

З 2019 року діють Державні будівельні норми України “Природне і штучне освітлення” – ДБН В.2.5-28:2018 [4], у яких прописані вимоги до використання всіх освітлювальних приладів, у т.ч. світлодіодних.

Працю працівника, який постійно працює за комп'ютером, згідно ДБН В.2.5-28:2018 [4], можна віднести до роботи з малою точністю (найменший

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи B). Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 Лк. [4], Крім того все поле зору повинно бути освітлено достатньо рівномірно – це основна гігієнічна вимога. Оскільки яскраве світло на ділянці периферійного зору значно 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

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язковою 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 напрузі вище 36 В.

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 6 м, довжина – 7 м, висота – 2,9 м.

У зазначеному приміщенні працює 4 людей.

Для того, щоб визначити потрібну кількість світильників, які повинні забезпечити нормований рівень освітленості, визначимо світловий потік, що падає на робочу поверхню за формулою [1]:

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

де:

F – світловий потік, що розраховується, Лм;

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

S – площа освітлюваного приміщення (у нашому випадку $S = 6 \times 7 = 42$ м²);

K – коефіцієнт запасу, що враховує зменшення світлового потоку лампи в результаті забруднення світильників у процесі експлуатації (його значення

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

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, у нашому випадку $Z = 1,1$);

n – коефіцієнт використання світлового потоку, (відношення світлового потоку, що падає на розрахункову поверхню, до сумарного потоку від усіх ламп і обчислюється в долях одиниці [8]); залежить від характеристик світильника, розмірів приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін ($\rho_{стін.}$) і стелі ($\rho_{стелі}$), значення коефіцієнтів дорівнюють $\rho_{стін} = 50\%$ і $\rho_{стелі} = 50\%$.

Обчислимо індекс приміщення за формулою:

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

де:

S – площа приміщення, $S = 42$ м²;

h – розрахункова висота підвісу, $h = 2,9$ м (співпадає з висотою стелі, оскільки лампи освітлення закріплюються на стелі);

A – ширина приміщення, $A = 6$ м;

B – довжина приміщення, $B = 7$ м.

Підставимо всі значення у формулу та визначимо індекс приміщення:

$$i=1,4.$$

Знаючи індекс приміщення, за знаходимо $n = 0,29$ (з табличних даних коефіцієнтів використання світлового потоку (n) світильників з відповідним типом лампам) [8]. Підставимо всі значення у формулу, визначимо світловий потік: $F=71689$ Лм.

Для розрахунку будемо використовувати світлодіодні стельові панелі Delux LED Panel 41 44 Вт, світловий потік яких $F_{л} = 3600$ Лм.

Число ламп визначається за формулою:

$$N=F/F_{л}$$

де:

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

F – світловий потік,

F_л – світловий потік однієї лампи.

Підставимо всі значення у формулу та визначимо індекс приміщення:

$$N = 71689 / 3600 = 19,9 \text{ шт.}$$

Приймаємо необхідну кількість світлодіодних світильників 20 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 умов поліпшення охорони праці.

КБПЗ – 2025

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління.

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 інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління.
- Досліджена система інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління.
- На основі отриманих результатів досліджень створена програмна реалізація системи інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління.

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.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

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.

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 RSA.

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.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ковтуненко Д.О. Дослідження та програмна реалізація системи інтелектуальної інтеграції існуючих компонент ІТ-системи з хмарними службами наступного покоління // Збірник праць молодих науковців ЦНТУ. – Вип. 15. – Кропивницький: ЦНТУ, 2025.

2. Wendell Odom. «CCNA 200-301 Official Cert Guide, Volume 1». Cisco Press. 2020. – 848 p.

3. Wendell Odom. «CCNA 200-301 Official Cert Guide, Volume 2 Premium Edition eBook and Practice Test». Cisco Press. 2020. – 624 p.

4. Scott Jernigan «CompTIA Network+ Certification All-in-One Exam Guide, Eighth Edition». 2022. – 976 p.

5. Doug Lowe «Networking For Dummies 12th Edition». 2020. – 480 p.

6. Ramon Nastase «Computer Networking: The Beginner's guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.

7. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.

8. Вінтенко Б., Смірнов О., Миронець І., Смірнова Т., Смірнов С. «Імітаційна модель шляхів вхідних даних комп'ютерної інтелектуальної системи підтримки оператора енергоблоку АЕС». *Комбінаторні конфігурації та їхні застосування: Матеріали XXVII Міжнародного науково-практичного семінару, присвяченого 125-річчю Національного університету «Запорізька політехніка» (Запоріжжя-Кропивницький-Київ, 4-6 червня 2025 р.)*. Запоріжжя: НУ «Запорізька політехніка», 2025. С.82-91.

9. Al-Azzeh, J., Ayyoub, B., Mesleh, A., Smirnova, T., Gnatyuk, S., Drieiev, O., Smirnov, O., Dorenskyi, O. «Cloud-Based Information System for Evaluating Caverns in the Process of Blasting Metal Surfaces of Details». *International Review on Modelling and Simulations* 18 (1), 2025. pp. 32-42.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

10. 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.
11. 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.
12. 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.
13. 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.
14. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yanchev, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.
15. 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.
16. 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.
17. 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-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

18. 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.

19. 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.

20. 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.

21. 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.

22. 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.

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

24. 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.

25. 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.

26. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhienko 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.

27. 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.

28. 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.

29. 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.

30. 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.

31. 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.

32. 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.

33. 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.

34. 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.

35. 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.

36. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 353-358.

37. 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.

38. 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.

39. 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.

40. 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

41. 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.

42. 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.

43. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Поліщук Л.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2020. – 294 с.

44. 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.

45. 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.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

46. 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.

47. 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).

48. 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

49. 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.

50. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та зміцнення поверхонь деталей. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.

51. 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.

52. 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.

					ВКРМ-122.25.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92