

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 проактивного
моніторингу мережевого електронного документообігу”

Виконав здобувач вищої освіти
II курсу, групи КІ-24М
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Батрак С.С.
« ____ » _____ 2025 р.

Керівник проекту
доктор технічних наук, професор
_____ Коваленко О.В.
« ____ » _____ 2025 р.
Рецензент _____

АНОТАЦІЯ

Батрак С.С. Дослідження та програмна реалізація системи проактивного моніторингу мережевого електронного документообігу. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2025.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи проактивного моніторингу мережевого електронного документообігу.

Метою розробки є дослідження та програмна реалізація системи проактивного моніторингу мережевого електронного документообігу.

Об'єктом дослідження є процес проактивного моніторингу мережевого електронного документообігу.

Предметом дослідження є методи проактивного моніторингу мережевого електронного документообігу.

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

Результат роботи – програмна реалізація системи проактивного моніторингу мережевого електронного документообігу.

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 Visual C++.

Ключові слова: комп'ютерна інженерія, проактивний моніторинг, мережевий електронний документообіг

ABSTRACT

Batrak S.S. Research and software implementation of a proactive monitoring system for network electronic document flow. 123 Computer Engineering. 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 a proactive monitoring system for network electronic document flow.

The purpose of the development is the research and software implementation of a proactive monitoring system for network electronic document flow.

The object of the research is the process of proactive monitoring of network electronic document flow.

The subject of the research is the methods of proactive monitoring of network electronic document flow.

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

The result of the work is a software implementation of a proactive monitoring system for network electronic document flow.

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 the software are provided.

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

The program was developed in the Visual C++ environment.

Keywords: computer engineering, proactive monitoring, network electronic document management

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.....	17
2.3 Розгорнута постановка завдання	19
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	21
3.1 Опис функціонування системи	21
3.2 Розробка структурної схеми.....	29
3.3 Розробка функціональної схеми	31
3.4 Розробка діаграми процесів.....	35
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	37
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	37
4.2 Захист розробленого програмного забезпечення.....	55
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	59
6 НАУКОВА НОВИЗНА	64

					ВКРМ-123.25.0028.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	Дослідження та програмна реалізація системи проактивного моніторингу мережевого електронного документообігу	Літ.	Аркуш	Аркушів
Розроб.	Батрак С.С.					М	1	90
Перев.	Коваленко О.В.							
Н.контр.	Коваленко А.С.					ЦНТУ КІ-24М		
Затв.	Смірнов О.А.							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ	65
7.1	Визначення цільової аудиторії кінцевого готового продукту	65
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	66
7.3	Вибір методу оцінки вартості ПЗ	67
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	68
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ	70
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ	70
7.7	Визначення ключових факторів успіху конкретного проєкту.....	71
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	73
8.1	Вступ.....	73
8.2	Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	74
8.3	Розробка заходів з умов поліпшення охорони праці.....	76
8.4	Техніка безпеки та протипожежна профілактика	77
8.5	Розрахункова частина	79
9	ОСНОВНІ ВИСНОВКИ.....	82
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	84

КБПЗ-2025

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ЕА	–	електронний архів
ЕДО	–	електронний документообіг
САДД	–	система автоматизації електронного документообігу й діловодства
СЕА	–	системи електронних архівів
СДЕ	–	системи діловодства електронні
СЕД	–	система електронного документообігу
ТЕ	–	типові елементи
ФФО	–	формат файлових об'єктів
EDMS	–	Electronic Document Management Systems

КБПЗ-2025

ВСТУП

Актуальність теми. Багато глобальних організацій узгодили свою стратегію та діяльність за допомогою структури інтегрованої системи управління (ІСУ) на основі ISO, яка дозволяє їм об'єднувати системи управління якістю, навколишнім середовищем, охороною здоров'я та безпекою. У такому контексті наявність надійної системи електронного управління документами (СЕД) є важливою, особливо на глобальних підприємствах, де велика кількість документів, що генеруються процесами, проходить через різні робочі культури. Однак, не існує універсального рішення для СЕД, оскільки воно залежить від потреб, розміру та розподілу ресурсів організацій.

У цій роботі розглядається взаємозв'язок між СЕД та ІСУ, щоб запропонувати найкращу практику. Дана робота методологічно базується на якісному, інтерпретативному, поздовжньому емпіричному дослідженні на заводі вітрових турбін.

Удосконалення та ефективність ІСУ ігнорують СЕД як ключовий фактор у встановленні належної технологічної підтримки процесів ІСУ. Правильне застосування СЕД може додатково сприяти організаційному навчанню, точності документації та міжорганізаційній співпраці.

Теоретизація щодо ІСУ потребує глибшого розуміння технологічних обмежень та потенціалу базування ІСУ на СЕД.

ІСУ – це складні системи, що включають велику кількість адміністративних функцій. СЕД забезпечує формальне представлення з потенціалом автоматизації, що підвищує та забезпечує достовірність документів. ІСУ, як правило, залишається у професіоналів, наприклад, лінійних менеджерів та фахівців з контролю якості/якості/якісного менеджменту (QA/QMS). Напрямок обговорення СЕД передбачає ширше охоплення. Дослідження ІСУ як технологічної реалізації забезпечує кращу платформу для узгодження ІСУ з

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

іншими бізнес-процесами та наближає ІСУ до операційної діяльності в межах підприємства.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи проактивного моніторингу мережевого електронного документообігу.

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

- Огляд існуючих систем проактивного моніторингу мережевого електронного документообігу.
- Дослідження системи проактивного моніторингу мережевого електронного документообігу.
- Програмна реалізація системи проактивного моніторингу мережевого електронного документообігу.

Об'єктом дослідження є процес проактивного моніторингу мережевого електронного документообігу.

Предметом дослідження є методи проактивного моніторингу мережевого електронного документообігу.

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

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

- Удосконалено метод проактивного моніторингу мережевого електронного документообігу.
- Розроблено вітчизняний продукт проактивного моніторингу мережевого електронного документообігу, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі проактивного моніторингу мережевого електронного документообігу.

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

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти LV науково-технічної конференції «Наука в ЦНТУ: основні досягнення та перспективи розвитку» (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 вищеперахованого, дослідження та програмна реалізація системи проактивного моніторингу мережевого електронного документообігу, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Електронне управління документами (СЕД) – це цифрова система, яка централізує зберігання, організацію та пошук корпоративної інформації, замінюючи фізичні документи більш безпечними та доступними електронними записами. Вона спрямована на оптимізацію ефективності, безпеки та відповідності вимогам, одночасно забезпечуючи безперебійну співпрацю команди.

Традиційне управління документами стикається зі значними труднощами, такими як:

- Тривалі ручні процеси.
- Фрагментовані та непов'язані системи.
- Складність обробки великих обсягів даних.
- Неможливість швидко виконати нові нормативні вимоги.
- Відсутність інтеграції, аналізу даних у режимі реального часу та масштабованості.

Щоб вирішити ці проблеми, важливо розробити такий інструмент, який пропонує передові рішення, такі як автоматизація на основі штучного інтелекту та централізовані робочі процеси.

Його інтегровані функції, включаючи електронні підписи, прогнозне оновлення метаданих та інтелектуальне управління контейнерами, розроблені для підвищення відповідності та адаптивності вашої компанії.

Очікується, що в найближчі роки СЕД включатиме розширені функції, такі як:

- Прогнозні аналітичні дані на основі штучного інтелекту.
- Проактивне зменшення ризиків.
- Комплексна автоматизація процесів.

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

1.2 Область застосування

Традиційні методи управління документами стали тягарем для конкурентоспроможності організацій. Йдеться про такі речі, як фізичний простір для зберігання, ручне керування та непов'язані процеси.

Ці елементи більше не належать до ринку, який вимагає гнучкості, надійної безпеки та сталого розвитку. Зрештою, вони призводять до прихованих витрат, операційних вразливостей та перешкод для інновацій.

Експлуатаційні витрати

Однією з цих перешкод є операційні витрати. Традиційні системи покладаються на ручні процеси, такі як пошук фізичних документів, частий друк і зберігання паперу. Результатом є збільшення витрат на фізичний простір та інші ресурси.

Відсутність автоматизації та інтелектуальної категоризації затримує критично важливі операції, що впливає на продуктивність та масштабованість.

Безпека та відповідність вимогам

Фізичні документи вразливі до втрати, несанкціонованого доступу та стихійних лих. Крім того, вони ускладнюють аудит.

Такі нормативні акти, як європейський GDPR та бразильський LGPD, вимагають відстеження та детального контролю доступу – вимог, які неможливо виконати традиційними методами.

Співпраця та мобільність

Традиційні моделі перешкоджають віддаленій роботі та співпраці в режимі реального часу, що призводить до несумісних версій файлів та затримок із затвердженням.

Відсутність інтеграції з цифровими інструментами, такими як електронні підписи та хмарні платформи, обмежує гнучкість, якої вимагають конкурентні світові ринки.

Інновації та сталий розвиток

Застарілі системи мають критичні обмеження:

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

– Вони не підтримують нові технології, такі як штучний інтелект для вилучення даних або блокчейн для автентифікації (обидві необхідні для прогнозного аналізу та розширеної безпеки).

– Вони споживають природні ресурси, такі як папір та енергія, підриваючи цілі ESG.

Тим часом, цифрові рішення зменшують викиди та експлуатаційні витрати. З огляду на ці структурні обмеження, перехід на інтегровану систему електронного документообігу (СЕД) стає вкрай важливим для конкурентоспроможності бізнесу.

У цьому контексті розуміння того, як СЕД працює на практиці – поєднання автоматизації, штучного інтелекту та прогнозного управління – є першим кроком у перетворенні документів з операційних зобов'язань на стратегічні активи.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи проактивного моніторингу мережевого електронного документообігу, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.25.0028.00.00.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

Представляємо вашій увазі дослідження СЕД, що фокусується винятково на технологічних параметрах. За дужками ми залишили ринкові показники постачальників і функціональні можливості продуктів, тому що вони вже досить добре освітлені в інших роботах. Сподіваємося, дослідження буде вам корисно!

Як ключові критерії оцінки СЕД обрані наступні:

- Архітектура.
- Інфраструктура.
- Використання.
- Розвиток.
- Зовнішнє середовище.

Методика дослідження

На основі аналізу загальних тенденцій в архітектурі й розробці прикладних систем для корпоративного використання, а також досвіду глобальних лідерів ЕСМ, був сформований портрет «ідеальної СЕД» – який могла б бути система електронного документообігу, якби вона відповідала всім сучасним вимогам. Деякі параметри ідеальної СЕД узяті із запасом – жодна з діючих систем ще не в змозі їм відповідати. Таким чином, наша ідеальна СЕД не тільки служить інструментом порівняння, але й задає вектор подальшого розвитку.

Зрозуміло, ідеальної СЕД у природі не існує. Та й не може існувати – тому що розвиток технологій сьогодні випереджає цикл розробки прикладних рішень, і будь-яка реальна система апріорі відстає від ідеалу. Якщо подивитися в

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

ретроспективі на визначення Enterprise Content Management від Gartner, то можна побачити, що состав і частка конкретних технологій, що включаються аналітиками в джентльменський набір ЕСМ-рішень, постійно мінявся. Це буде тривати й далі, і, швидше за все, з більшою швидкістю. Тобто саме визначення ідеальної СЕД, використовуване як базис для порівняння, потрібно регулярно переглядати й актуалізувати, не менш чим раз у рік. У рамках цього огляду ми сформуваємо своє бачення ідеальної СЕД.

Поки вендори СЕД будуть дотримуватися традиційної релізної політики й каскадної методології розробки, цей розрив буде тільки збільшуватися й може привести до зникнення СЕД як класу – їхньої функції можуть бути «порозтягані» суміжними системами. Наприклад, хмарні сховища візьмуть на себе роль репозиторія документів, а функції колективної роботи й керування завданнями будуть реалізовані на базі месенджерів.

Порівняння з ідеалом замість рейтингування

У ході дослідження кожна із представлених систем рівнялася на предмет близькості до ідеалу. По кожному пункті була розроблена шкала оцінки, що враховує сучасні тренди й історичну перспективу.

Наприклад, загальноновизнано, що архітектура «клієнт-сервер» на сучасний момент вважається застарілою й не може забезпечити достатньої гнучкості, масштабованості й надійності. З іншого боку, на піку популярності зараз перебуває мікросервісна архітектура, що дозволяє реалізувати концепцію «безперервного впровадження» і прискорити в такий спосіб доставку нових функцій бізнесу. Відповідно, між цими двома полюсами вибудовується шкала оцінок.

Як виняток за деякими критеріями, де важко виявити ідеальні параметри, проводилося пряме рейтингування. Наприклад, по вимогах до ресурсів для забезпечення роботи системи на 1000 користувачів. У цьому випадку шкала оцінок прив'язувалась до зайнятого в рейтингу місця. Аналогічним образом виставлялися оцінки за критеріями «інтеграція» і «експлуатація».

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Також у ході дослідження виявилася група критеріїв, для яких по різних причинах не вдалося побудувати шкалу оцінки. Зокрема, масштабованість, що по факті визначається архітектурою, і це вже було враховано в оцінці. Також був виключений з оцінки критерій якості й повноти API і за критерієм підтримуваних протоколів. Ці пункти залишені в анкетах як довідкова інформація.

Критерії включення й виключення

На українському ринку представлено більше сотні систем, що декларують свою приналежність до класу СЕД. Навіть за винятком застарілих систем, модулів відомих СЕД, внесених у каталог як самостійні позиції й унікальні замовлені розробки, однаково залишається кілька десятків продуктів. Із цієї причини повне дослідження ринку було б досить затратно й навряд чи б практично корисно.

Тому, щоб звузити область дослідження, насамперед було ухвалене рішення виключити з розгляду західні ECM-системи. У підсумку було вирішено провести дослідження з наступної вибірки:

- DIRECTUM.
- DocsVision.
- EOS for SharePoint.
- TerraLink xDE.
- TESSA.
- Логіка ECM.
- МСВсфера-інфооберт.
- Перша Форма.
- ТЕЗА.

Надалі із цього списку були виключені кілька систем:

- TerraLink xDE – як рішення на основі імпортного продукту Open Text.
- Логіка ECM – тому що по факті це зонтичний бренд, під яким виступає

шість різних продуктів:

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

- МСВ сфера-інфооберт – через орієнтацію продукту переважно на сегмент захищених систем, де діє інша логіка прийняття рішень.
- Перша Форма – оскільки це не зовсім СЕД, а «система керування класу workflow», як повідомляється на сайті.

Додатково в дослідження була включена система СПРАВА як найбільш відомий і масовий продукт класу СЕД. Таким чином, у фінальний перелік увійшло шість систем. У ході дослідження використовувалася інформація, доступна з відкритих джерел.

Результати дослідження

Для відносного порівняння систем у групі були полічені середні значення по п'яти осях – архітектура, інфраструктура, використання, розвиток і зовнішнє середовище.

Сильні сторони й недоліки досліджуваних СЕД

У дослідженні розглядалися наступні системи (приводяться за абеткою):

- Directum, версія 5.4
- DocsVision, версія 5
- EOS for SharePoint, версія 4.3.1
- TESSA, версія 2.5
- СПРАВА, версія 17.1
- ТЕЗА, версія 4.2.2

За підсумками дослідження сформувалося три кластери систем. Перший кластер представлений однією СЕД ТЕЗА, що технологічно відрізняється від всіх інших, це єдина відкрита система на стеці Java. У другий кластер увійшли близькі по технологічних підходах DocsVision, TESSA і EOS for SharePoint – їх поєднує прихильність платформі Microsoft, у тому числі найсучаснішим. Третій кластер утворюють СПРАВА й Directum, що мають під капотом досить багато успадкованих технічних рішень, що створює ризик для подальшого розвитку цих продуктів.

ТЕЗА

СЕД «ТЕЗА» – Java-додаток, реалізований на платформі CUBA. Завдяки властивостям платформи, забезпечується гарна масштабованість і надійність. Властиво, платформа CUBA призначена для швидкої розробки корпоративних додатків і СЕД може розглядатися як окремий випадок. CUBA є відкритою платформою, широко підтримуваної міжнародним комьюніті розроблювачів. В 2015 році компанія Haulmont була обрана Gartner як учасник секції Emerging Technologies Exhibitor на конференції Gartner Symposium, що можна розцінювати як факт визнання досягнень українських розроблювачів на міжнародному рівні. Мова Java залишається одним із самих популярних мов програмування, і це гарантує постійну наявність кадрових ресурсів різного рівня кваліфікації, що дозволяє гнучко підходити до формуванню команди.

Архітектура ТЕЗА допускає розробку додаткових компонентів для розширення функціональності й інтеграції. Особливо варто відзначити, що механізм спадкування Java дозволяє перевизначати навіть базові класи системи без необхідності вносити зміни не посередньо в їхній код.

У ТЕЗУ немає десктоп-клієнта, тільки веб, включаючи всі функції адміністрування. З огляду на загальний тренд у корпоративному ПЗ на використання веб-застосунків, це не є недоліком. Скоріше навпаки: відсутність товстого клієнта значно спрощує розгортання й супровід ТЕЗА, а також заощаджує ресурси розроблювача.

DocsVision

DocsVision представляє зріле рішення з багатою історією, у версії 5 реалізоване на новій технологічній платформі. Це забезпечує наступність наробітків і актуальність технічних рішень. У СЕД DocsVision відмінно пророблена структура й керування сховищем документів, є клієнтські інтерфейси в різних середовищах -Windows, web і mobile.

До недоліків системи можна віднести усе ще існуючу прив'язку до єдиної платформи Microsoft (яку компанія має намір незабаром перебороти й уже

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

анонсувала плани по підтримці ОС Linux і СУБД PostgreSQL). Високий у порівнянні з іншими СЕД апетит до апаратних ресурсів також варто розглядати як негативний фактор.

TESSA

TESSA – сучасна система на основі технологій Microsoft. Відсутність успадкованих рішень дозволяє їй розвиватися швидше й більш динамічно. Web-Клієнт також реалізований з використанням новітніх технологій. TESSA має досить гарний потенціал розвитку.

З недоліків – також зфокусованість на Microsoft. В умовах імпортозаміщення це може стати обмеженням для входу в деякі сегменти ринку. Якщо не взяти до уваги політичні фактори й працювати в комерційному сегменті, це не настільки критично. Однак розроблювач анонсував майбутню міграцію на Linux/PostgreSQL. При безсумнівних плюсах такого рішення є ризик, що це може привести до розмивання технологічного фокуса.

EOS for SharePoint

SharePoint є одним з лідерів ринку ECM (а тепер і CSP), і це дає помітну фору прикладним рішенням на цій платформі. EOS for SharePoint використовує переваги платформи й досвід компанії EOS в електронному документообігу з українською специфікою.

З недоліків необхідно вказати на неабияк високі вимоги до апаратних ресурсів. Крім того, можна було б одержати більше переваг від використання платформи SharePoint, якщо рухатися у бік більшої відкритості архітектури – у магазині додатків Microsoft доступно більше тисячі плагінів для SharePoint, з їхньою допомогою можна було б розширити можливості продукту.

СПРАВА

Інтерфейс системи СПРАВА ретельно пророблена й задовольняє різним категоріям користувачів. Функціональна повнота рішення EOS підтверджується досить непоганою відповідністю вимогам MoReq.

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Щоб повністю відповідати критеріям імпортозаміщення, розроблювач виконує перенос системи на стек Linux/PostgreSQL, але це ще не відбито в прайс аркуші.

Історично розробка СПРАВА ведеться на стеці Microsoft і навряд чи це кардинально зміниться. Архітектура системи СПРАВА своїми коріннями йде в епоху «клієнт-сервер» – перша версія була випущена в 1996 році. Хоча згодом архітектура була розширена й зараз є частково трьохзвенною, це може виявитися перешкодою надалі розвитку. З огляду на великий обсяг успадкованого коду, перехід на нову архітектуру не представляється економічно виправданим.

Система СПРАВА багато років була лідером ринку СЕД, однак в умовах, що змінилися, утримувати лідерство буде усе суужніше – оскільки відсутня можливість трансформації в сервісну архітектуру й повноцінну підтримку бізнес-процесів.

Directum

До числа безсумнівних переваг Directum, що яскраво виділяють його на тлі інших систем, варто віднести відмінно пророблений інтерфейс, використання ідей гейміфікації для залучення й навчання користувачів і самий широкий набір додаткових модулів, галузевих рішень і інтеграцій. Все це робить роботу із системою приємною й зручною.

Але в контексті технологічних критеріїв даного дослідження показники Directum не настільки гарні. Основний і непереборний ризик створює використання власного інструментарію розробки IS Builder і мови ISBL. Це обмежує «кормову базу» Екосистеми Directum тільки партнерами й клієнтами. Концепція мови ISBL сходить до мов 4GL, таким як Delphi, пік популярності яких минув на початку нульових у зв'язку з масовим переходом до розробки веб-застосунків, де Java і C# виявилися набагато ефективніше.

Ще одним істотним обмеженням Directum є його тверда прив'язка до платформи Microsoft. У корпоративному сегменті росте частка Linux і ВПЗ, і це глобальна тенденція, викликана не просто спробою заощаджувати на ліцензіях, а

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

характеристики застосунка й одержати його тексти, постачені великими коментарями. MFC AppWizard дозволяє створювати одновіконні й багатовіконні застосунки, а також застосунки, що не мають головного вікна, – замість нього використовується діалогова панель. Можна також включити підтримку технології OLE, баз даних, довідкової системи. Звичайно, MFC AppWizard не всесильний. Прикладну частину застосунка програмістові прийдеться розробляти самостійно. Вихідний текст застосунка, створений MFC AppWizard, стане тільки основою, до якої потрібно підключити інше. Але працюючий шаблон застосунка – це вже половина всієї роботи. Вихідні тексти застосунків, автоматично отриманих від MFC AppWizard, можуть становити сотні рядків тексту. Набір його вручну був би дуже стомлюючий. Потрібно відзначити, що MFC AppWizard створює тексти застосунків тільки з використанням бібліотеки класів MFC (Microsoft Foundation Class library). Тому тільки вивчивши мову C++ і бібліотеку MFC, можна користуватися засобами автоматизованої розробки й створювати свої застосунки в найкоротший термін. Як уже згадувався, MFC – це базовий набір (бібліотека) класів, написаних мовою C++ і призначених для спрощення й прискорення процесу програмування для Windows. Бібліотека містить багаторівневу ієрархію класів, що нараховує близько 200 членів. Вони дають можливість створювати Windows-застосунки на базі об'єктно-орієнтованого підходу. З погляду програміста, MFC являє собою каркас, на основі якого можна писати програми для Windows. Бібліотека MFC розроблялася для спрощення завдань, що стоять перед програмістом. Як відомо, традиційний метод програмування під Windows вимагає написання досить довгих і складних програм, що мають ряд специфічних особливостей. Зокрема, для створення тільки каркаса програми таким методом знадобиться близько 75 рядків коду. У міру ж збільшення складності програми її код може досягати воістину неймовірних розмірів. Однак та ж сама програма, написана з використанням MFC, буде приблизно в три рази менше, оскільки більшість приватних деталей приховано від програміста.

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

Одною з основних переваг роботи з MFC є можливість багаторазового використання того самого коду. В зв'язку з тим, що бібліотека містить багато елементів, загальних для всіх Windows-застосунків, немає необхідності щораз писати їх заново. Замість цього їх можна просто успадковувати (говорячи мовою об'єктно-орієнтованого програмування). Крім того, інтерфейс, забезпечуваний бібліотекою, практично незалежний від конкретних деталей, його що реалізують. Тому програми, написані на основі MFC, можуть бути легко адаптовані до нових версій Windows (на відміну від більшості програм, написаних звичайними методами). Ще однією істотною перевагою MFC є спрощення взаємодії із прикладним програмним інтерфейсом (API) Windows. Будь-який додаток взаємодіє з Windows через API, що містить кілька сотень функцій. Значний розмір API утрудняє спроби зрозуміти й вивчити його цілком. Найчастіше навіть складно простежити, як окремі частини API зв'язані один з одним! Але оскільки бібліотека MFC поєднує (шляхом інкапсуляції) функції API у логічно організовану безліч класів, інтерфейсом стає значно легше управляти.

Оскільки MFC являє собою набір класів, написаних мовою C++, тому програми, написані з використанням MFC, повинна бути в той же час програмами на C++. Для цього необхідно володіти відповідними знаннями. Для початку необхідно вміти створювати власні класи, розуміти принципи спадкування й вміти перевизначати віртуальні функції. Хоча програми, що використовують бібліотеку MFC, звичайно не містять занадто специфічних елементів з арсеналу C++, для їхнього написання проте потрібні солідні знання в даній області.

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 проактивного моніторингу мережевого електронного документообігу.

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

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-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Сьогодні нам кажуть, що простого моніторингу недостатньо. Щоб досягти справжньої оптимізації продуктивності та надійності, потрібно проводити моніторинг проактивно. Тому багато компаній звертаються до проактивного моніторингу.

На перший погляд, це цілком очевидно. Не потрібно бути досвідченим інженером з надійності сайтів (SRE), щоб знати, що краще застосовувати проактивний підхід до моніторингу, ніж чекати, поки щось піде не так, перш ніж виправляти це.

Але справжнє питання полягає в тому, як насправді здійснювати проактивний моніторинг. Легко говорити про важливість проактивного моніторингу, але важче підійти до нього на практиці.

Що таке проактивний моніторинг?

Проактивний моніторинг – це стратегія моніторингу, за якої команди прагнуть виявляти та вирішувати проблеми до того, як вони перетворяться на критичні збої.

Він відрізняється від звичайного реактивного моніторингу тим, що замість того, щоб чекати, поки щось піде не так, перш ніж вживати заходів для її вирішення, проактивний моніторинг зосереджується на використанні інформації з моніторингу для прогнозування проблем, що виникають, та придушення їх у зародку, перш ніж вони перетворяться на реальні проблеми.

Іншими словами, замість того, щоб чекати, поки сервер вийде з ладу або програма буде перевантажена трафіком, перш ніж вживати заходів щодо виправлення ситуації, проактивний моніторинг передбачає вжиття превентивних заходів для запобігання збою сервера або запуску іншого екземпляра програми,

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

щоб уникнути збою.

Проактивний моніторинг – це автоматизований процес у режимі реального часу, який безперервно відстежує продуктивність системи, мережеву інфраструктуру та ключові показники для виявлення та запобігання потенційним проблемам, перш ніж вони спричинять простої або збої. Використовуючи прогнозу діагностику та виявлення аномалій, проактивний моніторинг виявляє проблеми на ранній стадії, сповіщає команди про необхідність негайного вжиття заходів та забезпечує ефективну та надійну роботу системи.

Такий підхід допомагає запобігти збоям серверів, оптимізувати продуктивність та захистити інфраструктуру, забезпечуючи масштабовані, інтегровані рішення для підприємств будь-якого розміру. Завдяки проактивному моніторингу проблеми вирішуються до їх загострення, мінімізуючи перебої та підтримуючи безперебійну роботу.

Знову ж таки, переваги проактивного моніторингу мають бути досить очевидними. Коли ви запобігаєте проблемам проактивно, ваші користувачі стикаються з меншою кількістю збоїв. Ви забезпечуєте набагато кращий клієнтський досвід – і, в свою чергу, збільшуєте бізнес-цінність – коли повністю уникаєте серйозних збоїв.

За допомогою реактивного моніторингу найкраще, на що ви можете сподіватися, це швидке вирішення проблем із збоями після того, як вони вже сталися. Це не ідеально з точки зору клієнтського досвіду чи бізнесу.

Проактивний проти реактивного моніторингу

Проактивний моніторинг передбачає використання загальносистемних процедур моніторингу для виявлення та вирішення основних проблем, *перш ніж* вони призведуть до збоїв. Реактивний моніторинг здійснюється лише *у відповідь на* проблему, яка вже виникла. Процедури реактивного моніторингу, по суті, слугують розслідуванням першопричини проблеми, тоді як проактивний моніторинг служить запобіжним заходом.

Хоча проактивний моніторинг часто передбачає «реагування»

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

адміністраторів на невідповідності в продуктивності, ці дії все ще вважаються превентивними, оскільки кінцевий користувач ще не постраждав.

Чому проактивний моніторинг кращий за синтетичний моніторинг

Синтетичний моніторинг включає моделювання поведінки кінцевого користувача для тестування функціональності та виявлення проблем продуктивності та несправностей. Результати цих тестів використовуються для виявлення та усунення будь-яких збоїв, виявлених на шляхах користувача.

За своєю природою, синтетичний моніторинг можна вважати розширенням реактивного моніторингу, оскільки він спирається на вивчення досвіду кінцевого користувача і тому не може виявляти та діагностувати проблеми проактивно. Крім того, синтетичний моніторинг розглядає дані лише епізодично та не може протестувати весь набір можливих шляхів користувачів, що часто призводить до пропущених збоїв та проблем. Проактивний моніторинг, за визначенням, використовує цілісний підхід до ІТ-моніторингу та випереджає проблеми, шукаючи ранні індикатори.

Приклади проактивного моніторингу + ранні індикатори

Проблеми неможливо діагностувати, оцінюючи окремі ІТ-метрики. Однак, коли збираються історичні дані та встановлюється базовий рівень продуктивності, тенденції цих показників слугують ранніми індикаторами справжньої проблеми. Деякі приклади проактивного моніторингу ранніх індикаторів включають:

Приклад 1: Значна втрата пакетів

Пакет вважається «втраченим», коли він не досягає місця призначення після передачі. Системи покладаються на передачу пакетів для всієї інтернет-діяльності. У разі значної втрати пакетів кінцевий користувач може зіткнутися з повним перебоєм у роботі сервісу або повільним мережевим з'єднанням. Відсоток втрати пакетів необхідно контролювати, щоб переконатися, що на сервіс не впливають збої мережевого обладнання, несправності програмного забезпечення або порушення безпеки.

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

Приклад 2: Аномалії часу відгуку та тенденції до зниження

Забезпечення стабільної продуктивності передбачає встановлення базового рівня часу відгуку та визначення точок даних, які відрізняються від нього щонайменше на 100% або більше. Відхилення такого значного ступеня вимагає розслідування потенційних основних причин. Звідси мережеві адміністратори можуть дослідити розбіжність та вирішити будь-які проблеми, що перешкоджають продуктивності. Аномалія не завжди є показником серйознішої проблеми, проте закономірність низького або погіршеного часу відгуку, виявлена за допомогою проактивного моніторингу мережі, майже завжди є такою.

Приклад 3: Фактори навколишнього середовища

Визначення проактивного моніторингу виходить за рамки базових ІТ-метрик, таких як втрата пакетів та час відгуку. Навіть фактори навколишнього середовища, такі як температура в комп'ютерній кімнаті, підпадають під поняття проактивного моніторингу. Якщо ваша комп'ютерна кімната або центр обробки даних перегріється, ваше обладнання може зазнати величезних пошкоджень та можливого виходу з ладу. Такої події можна уникнути, просто встановивши датчики навколишнього середовища у вашому комп'ютері, які сповіщають вас про порушення певних температурних порогів.

Приклад 4: Проблеми з конфігурацією

Щось таке просте, як проблема з конфігурацією, може призвести до серйозного збою. Наприклад, інтерфейсне з'єднання на вашому маршрутизаторі відповідає за обробку інтернет-трафіку. Якщо це з'єднання перестає реагувати, це призводить до збоїв у роботі таких служб, як перегляд веб-сторінок та зовнішній трафік. Інші поширені помилки конфігурації включають незбереження конфігурації, випадкову зміну конфігурації пристрою або просто неправильне введення команд. Точні та функціональні конфігурації мають вирішальне значення для щоденної роботи вашої компанії, і успішні організації повинні мати можливість проводити проактивний моніторинг програм, щоб виявляти проблеми, пов'язані з неправильними конфігураціями, перш ніж вони вплинуть

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

на кінцевого користувача.

Найкращі практики для проактивного ІТ-моніторингу

Створіть карту своїх ІТ-активів

Щоб контролювати свої пристрої, вам спочатку потрібно знати, де вони знаходяться відносно свого оточення. Збір цієї інформації та її відображення на проактивній карті моніторингу мережі не лише допоможе вам візуалізувати структуру вашої мережі, але й точно визначити, де виникають проблеми. Наявність процедур автоматичного виявлення та зіставлення пристроїв допоможе вам підтримувати актуальну картину вашого середовища.

Встановлення базових показників

Встановлення базових показників продуктивності на основі історичних даних пристроїв є основою для будь-якого проактивного моніторингу. Без базового рівня немає точки відліку для позначення точок даних як аномалій, запуску функцій сповіщень та сповіщень або спостереження за тенденціями. Базові показники слід визначати шляхом поєднання аналізу історичних даних та цільових показників, встановлених найкращими практиками.

Увімкнути сповіщення та сповіщення

Ваше програмне забезпечення для моніторингу повинно мати можливість проактивно попереджати ІТ-менеджерів, коли ресурс знаходиться на межі досягнення критичного порогу. Такі показники, як високий трафік на мережевому комутаторі або зменшення ресурсів на сервері, що підтримує критично важливі для бізнесу програми, необхідно враховувати якомога швидше. Менеджери повинні мати можливість налаштувати проактивні сповіщення, коли пристрій перебуває в стані тривоги, щоб вони могли усунути проблему, перш ніж це вплине на кінцевого користувача.

Майте інтелектуальний план потужностей

Наявність інтелектуального плану управління потужностями необхідна для забезпечення безперебійного взаємодії з кінцевим користувачем. Щоб випередити дефіцит або надлишок обчислювальних ресурсів, потрібен

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

інтелектуальний та адаптивний план, що базується на поглибленому аналізі історичних даних. Організації повинні мати механізми для прогнозування своїх потреб у потужностях та проактивного моніторингу програм та інших активів на предмет будь-яких потенційних проблем із потужностями у найближчому майбутньому.

Оповіщення про тенденції, а не про порогові значення

Підхід до сповіщень за замовчуванням зазвичай полягає в налаштуванні сповіщень, які спрацьовують, коли показники перевищують певний поріг. Ви налаштовуєте свої інструменти так, щоб вони повідомляли вас, коли використання процесора сервера перевищує 80 відсотків, наприклад, або коли середній час відгуку програми перевищує 5 секунд.

Проблема з цими сповіщеннями на основі порогових значень полягає в тому, що вони, як правило, не повідомляють про проблеми, доки вони вже не перетворяться на збої. Якщо ваш сервер вже майже повністю завантажений процесором, може бути занадто пізно вирішувати проблему, перш ніж вона вийде з ладу.

Кращою стратегією є налаштування інструментів моніторингу таким чином, щоб вони сповіщали вас про відповідні тенденції, такі як стабільне збільшення використання процесора або часу відгуку протягом фіксованого періоду часу. Таким чином, ви будете знати раніше, коли починає проявлятися проблемна тенденція, що збільшує ваші шанси на реагування до того, як щось дійсно станеться не так.

Розтин

Розбір інформації після збою, тобто огляди або звіти, які команди готують після збою, щоб оцінити, що пішло не так, є чудовим способом запобігти повторенню подібних проблем у майбутньому.

Вони також корисні для застосування проактивного підходу до моніторингу, оскільки дозволяють вашій команді бути в курсі тенденцій та даних, пов'язаних з перебоями в минулому. Ретельно оцінюючи минулі проблеми, ваші

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

інженери мають кращі можливості для розпізнавання нових проблем у даних моніторингу в режимі реального часу.

Посібники реагування для проактивного моніторингу

Для фахівців з реагування на інциденти та IT-інженерів є поширеною практикою розробляти методичні посібники з реагування на інциденти, які допомагають їм вирішувати певні типи інцидентів. Однак, методичні посібники часто призначені для усунення проблем після того, як вони вже перетворилися на інциденти, а не для проактивного вирішення потенційних проблем на ранніх стадіях.

Ось чому варто інвестувати в посібники, які також враховують потреби раннього, проактивного вирішення проблем. Не обмежуйте свої посібники реагуванням на серйозні інциденти.

Зіставлення показників з впливом на бізнес

Не всі потенційні проблеми, які виявляють ваші інструменти моніторингу, матимуть однаковий вплив на бізнес. Іноді сервер може вийти з ладу або програма може вийти з ладу, фактично не порушуючи роботу користувачів, оскільки є резервні ресурси.

З цієї причини розумно класифікувати сповіщення залежно від рівня їхнього впливу на бізнес. Це допоможе вашій команді зрозуміти, на які проблеми, що виникають, слід звернути найбільшу увагу, а які можна відкласти, доки не збереться більше даних. В іншому випадку вони можуть витратити час на спроби вирішити некритичні проблеми проактивно, що ускладнить вирішення справді проблемних.

Як впровадити проактивний моніторинг

Крок 1: Визначте та під'єднайте своє обладнання

Скільки у вас маршрутизаторів і комутаторів? Скільки кінцевих точок? Окрім загального переліку пристроїв, вам потрібно знати, де розташований кожен з них і як він підключається до решти вашої мережі. Проактивні рішення для моніторингу з можливостями картографування дозволяють візуалізувати весь

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

ваш ІТ-ландшафт, автоматично додавати та видаляти обладнання, а також відстежувати дані про стан пристроїв у режимі реального часу.

Крок 2: Визначте свої найважливіші активи

На цьому етапі ви повинні визначити, яка інформація є найважливішою для моніторингу та збору. ІТ-середовища можуть мати десятки тисяч кінцевих точок та пристроїв, тому важливо точно визначити, які з них мають найбільший вплив на успіх вашої організації. Пристрої в цій категорії зазвичай включають комутатори, маршрутизатори, брандмауери та сервери. Після визначення ви повинні визначити, які показники потрібно відстежувати, щоб забезпечити оптимальну роботу пристрою, і чи можете ви ділитися цими даними зі своїми серверами.

Крок 3: Впровадження проактивного рішення для моніторингу

Щойно ви отримаєте уявлення про свою ІТ-систему та чітко визначите структуру пріоритетів, ви зможете впровадити програмне забезпечення для моніторингу, яке автоматично записуватиме та аналізуватиме дані про пристрої та інфраструктуру всієї організації. На цьому етапі вам буде запропоновано налаштувати базові рівні та порогові значення продуктивності.

Після встановлення цих параметрів та запису історичних даних програмне забезпечення, ймовірно, зможе звітувати про показники пристрою, робити прогнози та виявляти будь-які тенденції, які вимагатимуть розслідування потенційних причин. Це також сприятливий час для налаштування сповіщень на основі винятків для елементів та подій, які ви вважаєте критичними.

Крок 4: Підтримуйте свою інфраструктуру в належному стані

Ви повинні постійно переконатися, що основні комутатори, маршрутизатори та брандмауери, на яких базується ваша мережа, підтримують сучасні протоколи моніторингу продуктивності. Якщо ні, то ви будете обмежені в типі даних про продуктивність, які можете збирати, що значно ускладнить вам своєчасний моніторинг та вирішення будь-яких проблем.

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

Крок 5: Додайте кібербезпеку, автоматизацію та аналітику

Після впровадження основи вашої стратегії проактивного моніторингу ви можете додавати рівні для підвищення ефективності та заповнення прогалин у безпеці. Наприклад, автоматизацію можна інтегрувати для розгортання процедур усунення недоліків, щоб у разі перевищення або падіння показників нижче певного порогу заздалегідь запрограмована послідовність вирішувала проблему або передавала її відповідному персоналу.

Функції кібербезпеки, такі як виявлення та запобігання вторгненням, можуть використовувати вашу інфраструктуру моніторингу та зібрані нею дані для виявлення шкідливої поведінки у ваших системах. Автоматизовану аналітичну звітність можна налаштувати для генерування та надання зацікавленим сторонам зворотного зв'язку щодо ключових тенденцій і закономірностей у даних, що дозволить вашій організації постійно підвищувати продуктивність.

3.2 Розробка структурної схеми

Електронне управління документами (СЕД) працює за допомогою інтелектуального захоплення та автоматизованого індексування, перетворюючи неструктуровані дані на стратегічні цифрові активи. У провідних системах СЕД оптичне розпізнавання символів (OCR) працює на базі штучного інтелекту (ШІ) для вилучення даних, метаданих та контексту зі сканованих файлів, електронних листів або форм.

Тим часом, алгоритми машинного навчання можуть класифікувати документи на основі змісту, призначення або нормативних вимог.

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

файли. Це усуває конфлікти версій та пришвидшує такі процеси, як укладення договору чи відповіді на аудит.

Зрештою, розширена аналітика та інтеграція перетворюють документи на стратегічні інструменти: штучний інтелект визначає тенденції або аномалії у використанні документів, а API-інтерфейси з'єднують систему СЕД з платформами ERP, CRM або відповідності. Варіанти перевірки на основі блокчейну можуть забезпечити захист від несанкціонованого доступу, а автоматизовані зашифровані резервні копії можуть захистити документи протягом усього їхнього життєвого циклу.

3.3 Розробка функціональної схеми

На основі структурної схеми моделі середовища проактивного системного моніторингу СЕД була розроблена функціональна схема моделі середовища відкритої СЕД, при цьому визначені наступні об'єкти стандартизації:

- служби засобів забезпечення безпеки;
- інтерфейс користувача;
- служби засобів управління даними;
- служби засобів управління системою;
- служби засобів управління сховищем;
- служби засобів проектування;
- служби засобів управління змістом;
- служби засобів моделювання процесів;
- служби засобів моделювання даних;
- формати файлових об'єктів;
- служба комунікацій.

Отримані дані дозволили синтезувати функціональну схему еталонної моделі середовища відкритої СЕД (рисунок 3.2).

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

Дана модель використана при розробці функціонального стандарту середовища відкритої СЕД. Призначенням даного стандарту є:

- Забезпечення масштабованості СЕД.
- Забезпечення адаптуємості системи.
- Забезпечення мобільності (переміщуваності) СЕД.
- Забезпечення інтероперабельності СЕД.

Визначено, що область застосування функціонального стандарту СЕД установлює загальні положення по створенню й експлуатації відкритої системи управління документообігом на основі окремих модулів, що входять у єдину інформаційну систему організації.

Положення функціонального стандарту підлягають застосуванню при рішенні завдань:

- створення, модернізації СЕД;
- організації доступу користувачів до ресурсів СЕД;
- інтеграції обчислювальних і інформаційних ресурсів із СЕД.

Для заповнення профілю в роботі була розроблена методика вибору стандартів на основі експерименту.

Результатом розробки функціонального стандарту є набір вимог до елементів відкритої СЕД.

Проведемо дослідження особливості функціонування елементів системи управління електронним документообігом і експериментальним дослідженням з визначення ефективного формату файлових об'єктів СЕД, як засобу підвищення техніко-економічних характеристик СЕД.

Виходячи з вимог до функціонала систем управління й аналізу існуючих СЕД, у системах управління виділені типові елементи (ТЕ):

- ТЕ «масив зберігання інформації».
- ТЕ «точка діалогу».
- ТЕ «обробки інформації».
- ТЕ «передачі інформації».

– ТЕ «сполучна лінія».

Дані типові елементи служать для опису інформаційної системи й інформаційних потоків у будь-якій системі, що неможливо зробити через описані вище служби еталонної моделі середовища відкритої СЕД.

Таким чином показано, що через подібний базис типових елементів може бути представлена кожна, як завгодно складна інформаційна система.

Для проведення функціональної стандартизації ТЕ здійснимо перехід від ТЕ до служб СЕД.

Для перевірки експериментальним шляхом залежності ефективності роботи типових елементів СЕД від форматів використовуваних файлових об'єктів і вибору стандартів для відповідного розділу профілю розроблені:

- вимірювальна установка для виміру часу мережної затримки;
- програма статистичного аналізу файлових об'єктів СЕД;
- методики проведення вимірів часу мережної затримки й розмірів файлів у сховище СЕД.

Визначено співвідношення розмірів файлів залежно від типу інформації у файлах і їхніх розмірах. Визначено формати, у яких розмір файлу виходить мінімальним. Визначено, як саме впливає формат файлових об'єктів (ФФО) на роботу служб СЕД. На рисунку 3.2, на моделі СЕД показані служби, на роботу яких впливає формат файлових об'єктів. Ці служби виділені жовтим кольором.

Результати досліджень дозволили визначити залежність розмірів файлових об'єктів від типу їхнього вмісту й залежність часу мережної затримки, що підтвердило, що швидкодія системи залежить від розміру файлових об'єктів СЕД. Також дослідження дозволило підтвердити залежність ефективності роботи СЕД від ФФО й визначити ефективний формат файлових об'єктів СЕД, що дозволяє мінімізувати необхідні для роботи СЕД комунікаційні й обчислювальні ресурси, що дозволило гармонізувати стандарти файлових об'єктів у функціональному стандарті (профілі) середовища відкритої системи управління електронним документообігом і підвищити ефективність роботи ТЕ.

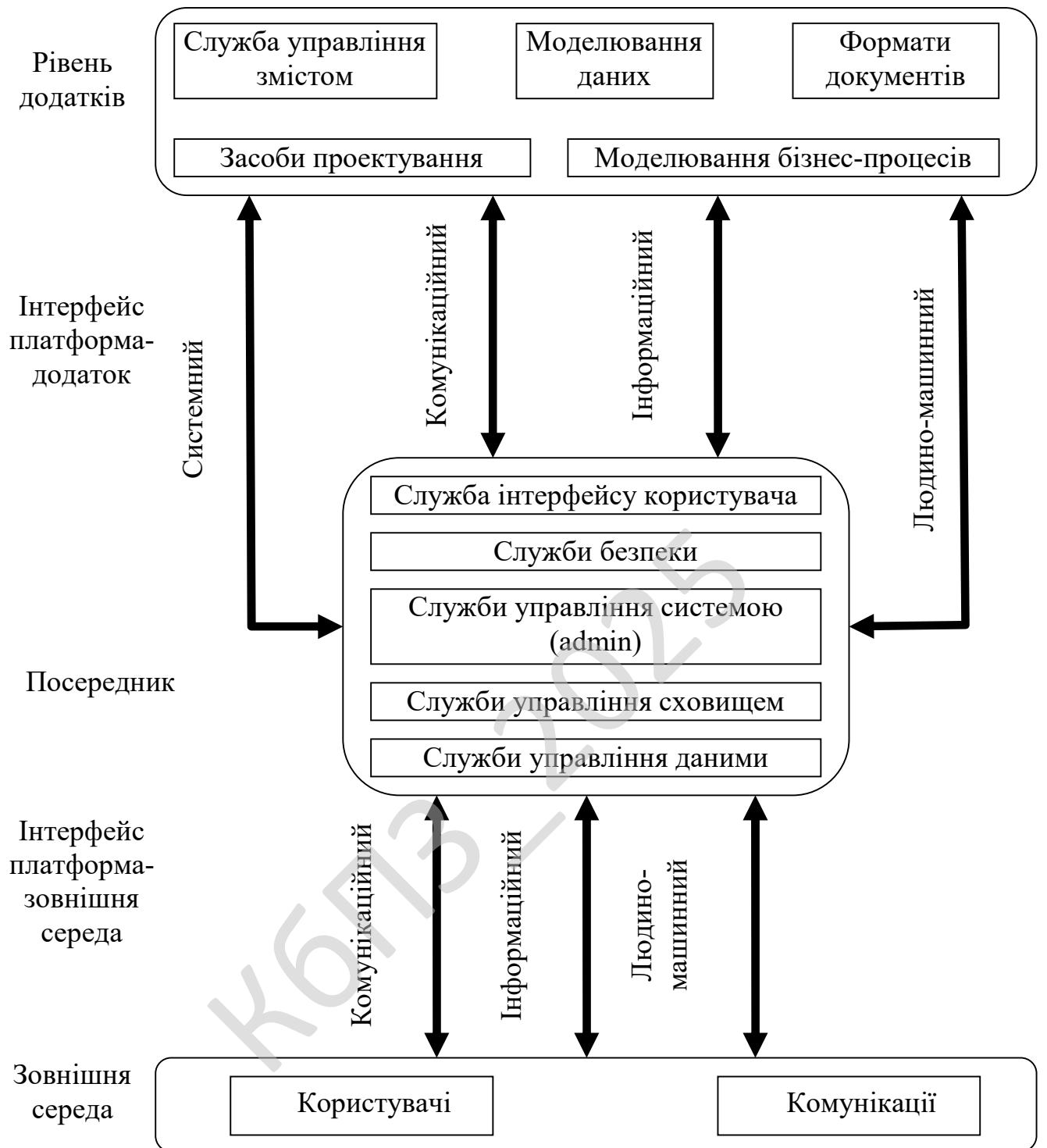


Рисунок 3.2 – Функціональна схема системи

Таким чином показано, що залежно від структури документів є ефективні можливості по поліпшенню роботи СЕД залежно від сфери її застосування.

Діаграми потоків даних містять чотири типи елементів: Процеси які являють собою трансформацію даних в рамках описуваної системи; Сховища даних (репозиторії); Зовнішні по відношенню до системи сутності; Потоки даних між елементами трьох попередніх типів.

КБПЗ_2025

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Незважаючи на те що я працював над ПЗ один в реалізації програми я використовував підходи пришвидшення розробки на основі методологій Agile – Extreme Programming.

Екстремальне програмування (Extreme Programming, далі XP) це методологія розробки програмного забезпечення, найпопулярніша серед так званих гнучких методологій. Має на меті поліпшення якості програмного забезпечення та чутливість до змін у вимогах замовників. Як вид гнучких методологій, XP радить часті "випуски" програми у коротких циклах розробки, що має на меті поліпшити продуктивність праці та покращити можливості виконання вимог замовника що змінюються. Авторами даної методології є Кент Бек, Ворд Каннінгем, Мартін Фаулер та інші.

Інші елементи екстремального програмування включають в собі: парне програмування, проведення обширної перевірки сирцевого коду, модульне тестування всього коду, уникання створення функціональності до того як вона дійсно необхідна, простота та ясність коду, очікування на зміну вимог замовників з плином часу та коли вимоги до продукту стають ясніші, досить часте спілкування із замовником та між самими програмістами.

Назва методології походить від ідеї застосувати корисні методи і практики розробки програмного забезпечення, піднявши їх до "екстремальних" рівнів.

Критики XP зауважують на потенційні недоліки цієї методології – нестабільні вимоги, незадокументовані компроміси конфліктів користувачів, відсутність загального документу дизайну програми.

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

- 1.1. Розробка через тестування (Test driven development).
- 1.2 Гра в планування (Planning game).
- 1.3. Замовник завжди поруч (Whole team, Onsite customer).
- 1.4 Парне програмування (Pair programming).
- 2. Безперервний, а не пакетний процес.
 - 2.1 Безперервна інтеграція (Continuous Integration).
 - 2.2 Рефакторинг (Design Improvement, Refactor).
 - 2.3 Часті невеликі релізи (Small Releases).
- 3. Розуміння, що поділяється всіма учасниками.
 - 3.1 Простота (Simple design).
 - 3.2 Метафора системи (System metaphor).
 - 3.3 Колективне володіння кодом (Collective code ownership) або обраними шаблонами проектування (Collective patterns ownership).
 - 3.4 Стандарт кодування (Coding standard or Coding conventions).
- 4. Соціальна захищеність програміста (Programmer welfare), а саме 40 годинний робочий тиждень (Sustainable pace, Forty hour week).

Під час роботи над магістрською 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.

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

У другому варіанті схема відображається більш детально, що полегшує її читання та ілюструє принцип роботи.

Основні елементи схем алгоритму це термінатор, процес, рішення, зумовлений процес (підпрограма), дані та з'єднувач.

Термінатор це елемент відображає вхід із зовнішнього середовища або вихід з неї (найчастіше застосування – початок і кінець програми). Всередині фігури записується відповідна дія.

Процес це виконання однієї або кількох операцій, обробка даних будь-якого виду (зміна значення даних, форми подання, розташування). Всередині фігури записують безпосередньо самі операції.

Рішення це показує рішення або функцію перемикального типу з одним входом і двома або більше альтернативними виходами, з яких тільки один може бути обраний після обчислення умов, визначених всередині цього елемента. Вхід в елемент позначається лінією, що входить зазвичай у верхню вершину елемента. Якщо виходів два чи три то зазвичай кожен вихід позначається лінією, що виходить з решти вершин (бічних і нижній). Якщо виходів більше трьох, то їх слід показувати однією лінією, що виходить з вершини (частіше нижній) елемента, яка потім розгалужується. Відповідні результати обчислень можуть записуватися поруч з лініями, що відображають ці шляхи.

Зумовлений процес (підпрограма) це символ відображає виконання процесу, що складається з однієї або кількох операцій, що визначені в іншому місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і передані в нього дані.

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

Дані це перетворення у форму, придатну для обробки (введення) або відображення результатів обробки (виведення). Цей символ не визначає носія даних (для вказівки типу носія даних використовуються специфічні символи).

З'єднувач це символ відображає вихід в частину схеми і вхід з іншої частини цієї схеми. Використовується для обриву лінії та продовження її в іншому місці (приклад: поділ блок-схеми, що не поміщається на листі). Відповідні сполучні символи повинні мати одне (при тому унікальне) позначення.

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 проактивного системного моніторингу СЕД, 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.

Архітектура клієнт-сервер є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережних програм і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

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

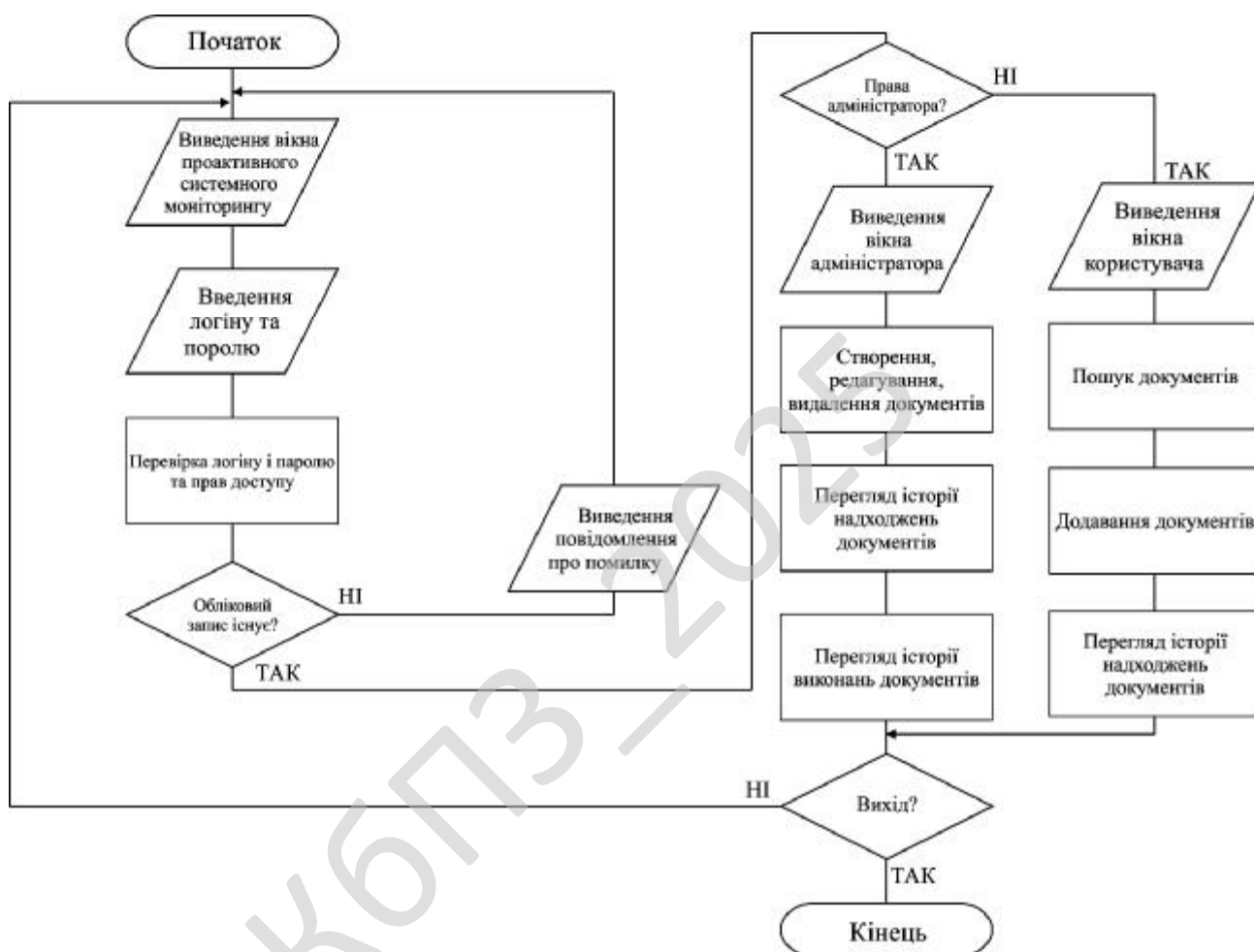


Рисунок 4.1 – Блок-схема основної програми

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

Дуже важливо ясно уявляти, хто або що розглядається як «клієнт». Можна говорити про клієнтський комп'ютер, з якого відбувається звернення до інших комп'ютерів. Можна говорити про клієнтське та серверне програмне забезпечення.

Нарешті, можна говорити про людей, які бажають за допомогою відповідного програмного та апаратного забезпечення отримати доступ до тієї чи іншої інформації.

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

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

- рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд;
- прикладний рівень, який реалізує основну логіку ПЗ і на якому здійснюється необхідна обробка інформації;
- рівень управління даними, який забезпечує зберігання даних та доступ до них.

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

- модель тонкого клієнта, в рамках якої вся логіка ПЗ та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;

– модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта. Товстими клієнтами часто також називають пристрої з обмеженою потужністю: кишенькові комп'ютери, мобільні телефони та ін.

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

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

Але, крім браузерів, до серверів можуть звертатися і інші клієнти, а саме – автономні програми. Вони можуть передбачати взаємодію з людиною, а можуть працювати в цілком автоматичному режимі. Типовим класом таких програм є роботи, призначені для автоматичного перегляду веб-ресурсів. Зокрема, роботи є важливим елементом пошукових систем і використовуються ними для перегляду сторінок і збору інформації про них.

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

Трирівнева клієнт-серверна архітектура, яка почала розвиватися з середини 90-х років, передбачає відділення прикладного рівня від управління даними. Відокремлюється окремий програмний рівень, на якому зосереджується

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

прикладна логіка ПЗ. Програми проміжного рівня можуть функціонувати під управлінням спеціальних серверів ПЗ, але запуск таких програм може здійснюватися і під управлінням звичайного веб-сервера. Нарешті, управління даними здійснюється сервером даних.

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

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

Веб-оглядач формує запит та пересилає його до сервера, який здійснює обробку. При необхідності сервер викликає серверні програмні модулі, які забезпечують обробку запиту і в разі потреби звертаються до сервера даних. Сервер даних здійснює операції з даними, що зберігаються в системі та складають її інформаційну основу. Зокрема, він може здійснити вибірку з інформаційної бази відповідно до запиту та передати її модулю проміжного рівня для подальшої обробки. Дані, з якими працює сервер даних, найчастіше організовані як реляційна база даних.

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

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

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

засоби можуть бути найрізноманітнішими. Так, для створення серверних програм, крім PHP, широко застосовуються Java, Perl, Python, Delphi.

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

Приведемо частину програмного коду, яка реалізує функції головного вікна.

```
/*-----*/
// NAME:      CMyA7
// DESCRIPTION:  Конструктор класу
// INPUT:      N/D
/*-----*/
CMyA7::CMyA7(void)
{
    listDoc = gnew ArrayList();
    logInput = gnew CLogger();
    logOutput = gnew CLogger();
}
/*-----*/
// NAME:      GetIndexByISBNHash
/*-----*/
Int32 CMyA7::GetIndexByISBNHash(Int32 HashValue)
{
    // Цикл пошуку документа із заданим вхідним номером документа
    for(Int32 i = 0; i < listDoc->Count; i++)
    {
        // Хеш-значення вхідного номера документа заданого документа збігається з
        // хеш- значенням вхідного номера документа знайденого документа?
        if(HashValue == ((CDoc^)listDoc[i])->GetISBNHash())
        {
            // Документ знайдений, повернути індекс
            return i;
        }
    }
    // Документ не знайдений, повернути -1
    return -1;
}
```

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

```

}
/*****/
// NAME:      AddDoc
// DESCRIPTION:  Функція додавання нового документа
/*****/
Boolean CMyA7::AddDoc(CDoc^ SourceDoc)
{
    // Документи з таким вхідним номером документа не існує?
    if(this->GetIndexByISBNHash(SourceDoc->GetISBNHash()) == -1)
    {
        // Додати заданній документ у загальний список
        listDoc->Add(SourceDoc);
        // Записати подія в лог
        logInput->WriteEvent(
            "Новий документ '" + ((CDoc^)SourceDoc)->GetName() +
            "', Автор '" + ((CDoc^)SourceDoc)->GetAuthor() +
            "', Вхідний номер документа " + ((CDoc^)SourceDoc)->GetISBN());
        // Функція відробила успішно
        return true;
    }
    else
    {
        // Документ із таким же вхідним номером документа існує, повернути помилку
        return false;
    }
}
/*****/
// NAME:      RemoveDoc
// DESCRIPTION:  Функція видалення документів із системи
/*****/
Int32 CMyA7::RemoveDoc(Int32 ddISBNHash)
{
    // Знайти документ по вхідному номеру документа
    Int32 ddIndex = this->GetIndexByISBNHash(ddISBNHash);
    // Документ із таким вхідним номером документа існує?
    if(ddIndex != -1)
    {
        // Перевірити, що жодного документа немає в користувача
        if(((CDoc^)listDoc[ddIndex])->GetTotalNumber() ==
            ((CDoc^)listDoc[ddIndex])->GetFreeNumber())
        {
            // Записати подія в лог

```

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

```

logOutput->WriteEvent(
    "Виконаний документ '" + ((CDoc^)listDoc[ddIndex])->GetName() +
    "', Автор '" + ((CDoc^)listDoc[ddIndex])->GetAuthor() +
    "', Вхідний номер документа '" + ((CDoc^)listDoc[ddIndex])
    ->GetISBN());
// Видалити знайдений документ зі списку
listDoc->RemoveAt(ddIndex);
// Функція відновила успішно
return 0;
}
else
{
    // Повернути помилку, не всі документи перебувають у системі
    return 1;
}
}
else
{
// Повернути помилку, документ із таким вхідним номером документа не був знайдений
return 2;
}
}
/*****/
// NAME:      GiveDoc
// DESCRIPTION:  Функція видачі документів користувачеві
/*****/
Boolean CMyA7::GiveDoc(ArrayList^ listReader, Int32 ddISBNHash)
{
    // Визначити індекс по вхідному номеру документа
    Int32 ddIndex = this->GetIndexByISBNHash(ddISBNHash);
    // Зменшити кількість вільних документів
    if((ddIndex != -1) && ((CDoc^)listDoc[ddIndex])->DecFreeNumber())
    {
        // Додати документ у список користувача
        listReader->Add(listDoc[ddIndex]);
        // Функція відновила успішно
        return true;
    }
    else
    {
        // Такого документа не є в наявності
        return false;
    }
}

```

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

```

}
}
/*****/
// NAME:      TakeDoc
// DESCRIPTION:  Функція прийняття документів від користувача назад у систему
/*****/
void CMyA7::TakeDoc(ArrayList^ listReader, Int32 ddISBNHash)
{
// Визначити індекс по вхідному номеру документа (документ існує,
// тому що його вхідний номер документа був повідомлений користувачеві)
Int32 ddIndex = this->GetIndexByISBNHash(ddISBNHash);
// Видалити документ зі списку користувача
// listReader->RemoveAt(ddIndex);
// Збільшити кількість вільних документів
((CDoc^)listDoc[ddIndex])->IncFreeNumber();
}
/*****/
// NAME:      LoadDocList
// DESCRIPTION:  Функція завантаження документів системи з файлу
/*****/
void CMyA7::LoadDocList()
{
// Перевірити існування файлу
if(!File::Exists(DOCS_FILE))
{
// Файл не знайдений, закінчити виконання функції
return;
}
// Відкрити файл
StreamReader^ srDoc = gcnew StreamReader(DOCS_FILE);
// Продовжувати, поки не буде знайдений кінець файлу
while(srDoc->EndOfStream == false)
{
// Створити новий об'єкт "документ"
CDoc^ NewDoc = gcnew CDoc();
// Завантажити з файлу й установити параметри документа
NewDoc->SetName(srDoc->ReadLine());
NewDoc->SetAuthor(srDoc->ReadLine());
NewDoc->SetISBN(srDoc->ReadLine());
NewDoc->SetTheme(srDoc->ReadLine());
NewDoc->SetPages(Convert::ToInt32(srDoc->ReadLine()));
NewDoc->SetTotalNumber(Convert::ToInt32(srDoc->ReadLine()));
}
}
}

```

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

```

NewDoc->SetFreeNumber(Convert::ToInt32(srDoc->ReadLine()));
// Включити документ у загальний список документів
listDoc->Add(NewDoc);
}
// Закрити файл
srDoc->Close();}
/*****/
// NAME:      SaveDocList
// DESCRIPTION:  Функція збереження документів системи у файл
/*****/
void CMyA7::SaveDocList()
{
// Перевірити існування файлу
if(File::Exists(DOCS_FILE))
{
// Видалити файл
File::Delete(DOCS_FILE);
}
// Створити й відкрити файл
StreamWriter^ swDoc = gcnew StreamWriter(DOCS_FILE);
// Цикл запису по одному документу
for(Int32 i = 0; i < listDoc->Count; i++)
{
// Одержати параметри документа й записати їх у файл
swDoc->WriteLine(((CDoc^)listDoc[i])->GetName());
swDoc->WriteLine(((CDoc^)listDoc[i])->GetAuthor());
swDoc->WriteLine(((CDoc^)listDoc[i])->GetISBN());
swDoc->WriteLine(((CDoc^)listDoc[i])->GetTheme());
swDoc->WriteLine(Convert::ToString(((CDoc^)listDoc[i])->GetPages()));
swDoc->WriteLine(Convert::ToString(((CDoc^)listDoc[i])->GetTotalNumber()));
swDoc->WriteLine(Convert::ToString(((CDoc^)listDoc[i])->GetFreeNumber()));
}
// Закрити файл
swDoc->Close();
}
/*****/
// NAME:      FindDoc
// DESCRIPTION:  Функція пошуку документа по заданих параметрах
/*****/
ArrayList^ CMyA7::FindDoc(String^ strFindValue)
{
// Список для результату

```

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

```

ArrayList^ listResult = gnew ArrayList();
// Шукане значення без обліку реєстра
String^ strValue = strFindValue->ToLower();
// Задана порожній рядок?
if(String::IsNullOrEmpty(strValue->Trim()))
{
    //Вивести весь список
    for(Int32 i = 0; i < listDoc->Count; i++)
    {
        // Додати документ у результуючий список
        listResult->Add(listDoc[i]);
    }
}
else
{
    // Вибрати з умовою
    for(Int32 i = 0; i < listDoc->Count; i++)
    {
        // Пошук рядка в кожному полі без обліку реєстра
        if((((CDoc^)listDoc[i])->GetName())->ToLower())->IndexOf(strValue) != -1 ||
            (((CDoc^)listDoc[i])->GetAuthor())->ToLower())->IndexOf(strValue) != -1 ||
            (((CDoc^)listDoc[i])->GetISBN())->ToLower())->IndexOf(strValue) != -1 ||
            (((CDoc^)listDoc[i])->GetTheme())->ToLower())->IndexOf(strValue) != -1 ||
            Convert::ToString((((CDoc^)listDoc[i])->GetPages()))->IndexOf(strValue) != -1)
        {
            // Додати документ у результуючий список
            listResult->Add(listDoc[i]);
        }
    }
}
// Повернути список збігів
return listResult;
}
/*****/
// NAME: ViewDoc
// DESCRIPTION: Функція перегляду інформації про заданий документ
/*****/
CDoc^ CMyA7::ViewDoc(Int32 ddISBNHash)
{
    // Визначити індекс по вхідному номеру документа (документ існує, тому що
    // його вхідний номер документа був повідомлений користувачеві)
    Int32 ddIndex = this->GetIndexByISBNHash(ddISBNHash);
}

```

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

```

// Повернути елемент "документ" із загальне списку
return (CDoc^)listDoc[ddIndex];
}
/*****/
// NAME:      ReadLogs
// DESCRIPTION:  Функція зчитування історії з файлу
/*****/
void CMyA7::ReadLogs()
{
    // Файл із історією надходжень існує?
    if(File::Exists(LOG_INPUT))
    {
        // Зчитування історії надходжень
        logInput->strLog = File::ReadAllText(LOG_INPUT);
    }
    else
    {
        logInput->strLog = "";
    }
    // Файл із історією списань існує?
    if(File::Exists(LOG_OUTPUT))
    {
        // Зчитування історії списань
        logOutput->strLog = File::ReadAllText(LOG_OUTPUT);
    }
    else
    {
        logOutput->strLog = "";
    }
}
/*****/
// NAME:      WriteLogs
// DESCRIPTION:  Функція збереження історії у файл
/*****/
void CMyA7::WriteLogs()
{
    // Запис історії
    File::WriteAllText(LOG_INPUT, logInput->strLog);
    File::WriteAllText(LOG_OUTPUT, logOutput->strLog);
}
/*****/
// NAME:      ClearLogs

```

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

```

// DESCRIPTION:      Функція очищення історії
/*****/
void CMyA7::ClearLogs()
{
    logInput->strLog = "";
    logOutput->strLog = "";
}
/*****/
// NAME:            ViewInputLog
// DESCRIPTION:     Функція перегляду історії надходжень
/*****/
String^ CMyA7::ViewInputLog()
{
    return logInput->strLog;
}
/*****/
// NAME:            ViewOutputLog
// DESCRIPTION:     Функція перегляду історії списань
/*****/
String^ CMyA7::ViewOutputLog()
{
    return logOutput->strLog;
}
/*****/

```

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою алгоритму Camellia – блоковий шифр на основі мережі Фейстеля. У криптографії, Camellia – це симетричний ключ блоковий шифр із розміром блоку 128 біт і розмірами ключа 128, 192 і 256 біт. Він був розроблений спільно Mitsubishi Electric і NTT з Японії. Шифр був схвалений для використання ISO / IEC, проектом Європейського Союзу NESSIE і Японським CRYPTREC проект. шифр має рівні безпеки й можливості обробки, порівнянні з Advanced Encryption Standard.

Шифр був розроблений, щоб підходити як для програмних, так і для апаратних реалізацій, від недорогих смарт-карти для високошвидкісних

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

мережних систем. Він є частиною криптографічного протоколу Transport Layer Security (TLS), призначеного для забезпечення безпеки зв'язки в комп'ютерній мережі, такий як Інтернет

Camellia – це шифр Фейстеля з 18 раундами (при використанні 128-бітних ключів) або 24 раундами (при використанні 192- або 256-бітних ключів). Кожні шість раундів застосовується шар логічного перетворення: так звана «FL-функція» або її зворотна. Camellia використовує чотири 8×8 -бітних S-блоку із вхідними й вихідними афіними перетвореннями й логічними операціями. Шифр також використовує введення й вивід відбілювання клавiш. Шар дифузiя використовує лінійне перетворення на основі матриці з номером галузей 5.

Аналіз безпеки

Камелія вважається сучасним надійним шифром. Навіть при використанні параметра меншого розміру ключа (128 біт) вважається неможливим зламати його за допомогою атаки грубої сили на ключі за допомогою сучасних технологій. Немає відомих успішних атак, що значно послабляють шифр. Шифр був схвалений для використання ISO / IEC, проектом Європейського Союзу NESSIE і Японським CRYPTREC проект. Японський шифр має рівні безпеки й можливості обробки, порівнянні із шифром AES/Rijndael.

Camellia – це блоковий шифр, який може бути повністю визначені мінімальними системами багатомірних багаточленів:

- Камелія (а також AES) S-блоки можуть бути описані системою 23 квадратних рівнянь в 80 членах.
- Розклад ключів можна описати 1120 рівняннями в 768 змінні з використанням 3328 лінійних і квадратичних членів.
- Увесь блоковий шифр можна описати 5104 рівняннями в 2816 змінні з використанням 14 592 лінійних і квадратичних членів.
- Усього потрібно 6224 рівняння з 3584 змінними з використанням 17 920 лінійних і квадратичних членів.

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

– Кількість вільних членів становить 11 696, що приблизно таке ж число, що й для AES.

Теоретично, такі властивості можуть дозволити зламати Camellia (і AES) за допомогою алгебраїчної атаки, такий як розширена розріджена лінеаризація, у т Майбутнє за умови, що атака стане можливою.

Хоча Camellia запатентована, вона доступна за безоплатною ліцензією. Це дозволило шифру Camellia стати частиною проекту OpenSSL під ліцензією з відкритим вихідним кодом з листопада 2006 року. Це також дозволило йому стати частиною Mozilla Модуль NSS (Служби мережної безпеки).

Підтримка Camellia була додана в остаточний випуск Mozilla Firefox 3 в 2008 році (за замовчуванням відключене починаючи з Firefox 33 в 2014 році в дусі «Пропозиції по зміні стандартних наборів шифрів TLS, пропонувані браузерами», який був виключено з версії 37 в 2015 році). Pale Moon, відгалуження Mozilla / Firefox, продовжує пропонувати Camellia і розширив свою підтримку, включивши в нього набори Galois / Counter mode (GCM) із шифром, але вилучив GCM знову у випуску 27.2.0, пославшись на очевидну відсутність інтересу до них.

Пізніше, в 2008 році, група розробки релізу FreeBSD оголосила, що цей шифр також був включений в FreeBSD 6.4. Крім того, Йошисато Янагисава додав підтримку шифру Camellia у дисковий клас зберігання geli FreeBSD.

У вересні 2009 року GNU Privacy Guard додала підтримку Camellia у версії 1.4.10.

Veracrypt (відгалуження Truecrypt) включав Camellia як один з підтримуваних алгоритмів шифрування.

Крім того, різні популярні бібліотеки безпеки, такі як Crypto ++, Gnutls, mbed TLS і Openssl також включають підтримку Camellia.

26 березня 2013 р. було оголошено, що Camellia була знову обрана для включення в новий список рекомендованих шифрів для електронного уряду Японії як єдиний 128-бітний алгоритм блокового шифрування, розроблений у

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

Японії. Це збігається з тим, що список CRYPTREC обновляється вперше за 10 років. Вибір був заснований на високій репутації Camellia у плані простоти придбання, а також характеристик безпеки й продуктивності, порівнянних з такими з Advanced Encryption Standard (AES). Камелія залишається незмінною у своєму повному втіленні. Неможлива диференціальна атака на Camellia з 12 раундами без шарів FL / FL дійсно існує.

Продуктивність

S-блоки, використовувані Camellia, мають структуру, аналогічну S-блоку AES. У результаті можна прискорити реалізацію програмного забезпечення Camellia за допомогою наборів команд ЦП, розроблених для AES, таких як x86 AES-NI.

КБПЗ_2025

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

- підвищення ефективності керування шляхом автоматизації контролю виконання, більшої прозорості діяльності підрозділів і окремих співробітників;
- автоматизація бізнес-процесів з їхньою одночасною оптимізацією;
- забезпечення підтримки нагромадження, керування й організації доступу до корпоративної інформації й знань;
- протоколювання діяльності організації в цілому, її окремих підрозділів, робочих груп, співробітників з використанням цієї інформації для підтримки прийняття рішень і т.д.;
- скорочення обертів паперових документів (з метою зниження витрат);
- спрощення й здешевлення зберігання документів, що використовуються в поточній діяльності, за рахунок створення оперативного електронного архіву.

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.

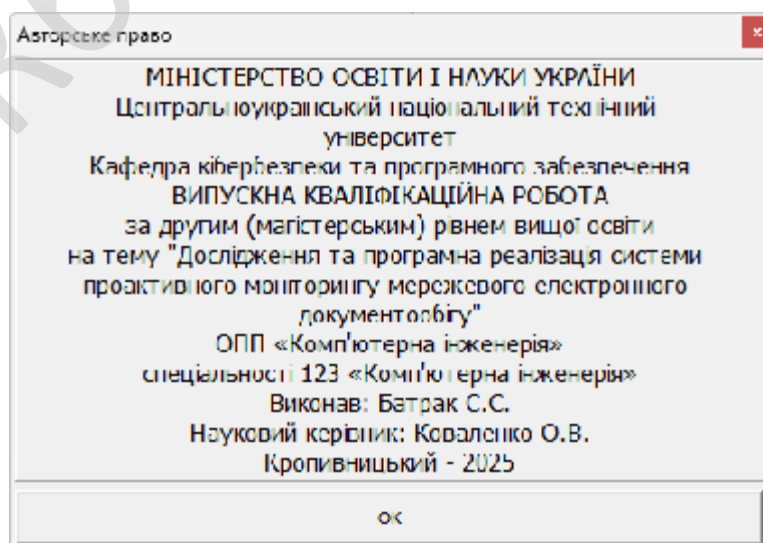


Рисунок 5.2 – Авторське право

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

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

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

процедур. Таким чином, розроблене ІТ рішення автоматично стає життєво важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження.

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

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

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.

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

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

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

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

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

КБПЗ – 2023

					VKPM-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи проактивного моніторингу мережевого електронного документообігу.

Метою розробки є дослідження та програмна реалізація системи проактивного моніторингу мережевого електронного документообігу.

Об'єктом дослідження є процес проактивного моніторингу мережевого електронного документообігу.

Предметом дослідження є методи проактивного моніторингу мережевого електронного документообігу.

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

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

- Удосконалено метод проактивного моніторингу мережевого електронного документообігу.
- Розроблено вітчизняний продукт проактивного моніторингу мережевого електронного документообігу, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження системи проактивного моніторингу мережевого електронного документообігу можуть бути цікавими передусім для компаній, які мають великий обсяг внутрішньої документації та активно користуються електронними сервісами обміну даними. Для таких організацій важливо не просто зберігати документи, а й мати інструмент для відстеження їхнього руху, контролю термінів і своєчасного реагування на проблеми.

Великий інтерес можуть проявити ІТ-відділи підприємств, що займаються цифровізацією внутрішніх процесів. Для них впровадження подібної системи дає можливість підвищити надійність документообігу, уникнути простоїв і технічних збоїв, а також вчасно попереджати проблеми, замість того щоб реагувати на них постфактум.

Державні установи також можуть бути зацікавлені у використанні системи такого типу, адже для них актуальним є питання прозорості документообігу, зниження ризику втрати важливих файлів та покращення контролю за виконанням управлінських рішень.

Для розробників програмного забезпечення результати дослідження можуть стати цінним джерелом ідей щодо побудови ефективних моніторингових систем, які поєднують зручність, надійність і автоматизацію.

І, звісно, така розробка буде цікавою для студентів і науковців, які досліджують питання цифрової трансформації, оптимізації бізнес-процесів та інформаційної безпеки, адже вона демонструє реальний приклад інтеграції сучасних ІТ-рішень у практичну діяльність організацій.

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Для визначення привабливості системи проактивного моніторингу мережевого документообігу можна організувати опитування групи експертів, до якої увійдуть IT-фахівці, керівники відділів документообігу, менеджери проєктів та представники адміністрацій установ. Кожен експерт оцінює систему за кількома критеріями: зручність використання, потенціал економії часу, надійність, рівень автоматизації, вартість впровадження та очікуваний ефект для організації.

Результати оцінюються за десятибальною шкалою, після чого обчислюється середнє арифметичне значення по кожному критерію. Наприклад, якщо середній бал за технічну надійність становить 9, а за економічну доцільність – 8, це свідчить про високу перспективність рішення.

У процесі опитування експерти можуть також залишати свої коментарі, що дозволяє розробникам побачити не лише кількісні, а й якісні сторони оцінки. Часто саме коментарі допомагають зрозуміти, які функції варто доопрацювати, або де користувачам не вистачає простоти у використанні.

Після узагальнення результатів формується рейтинг привабливості, який можна порівняти з іншими подібними продуктами. Якщо інтегральний показник перевищує, наприклад, 8 балів, це означає, що система є конкурентоспроможною і має ринковий потенціал.

Таким чином, експертна оцінка дозволяє отримати об'єктивне уявлення про сильні та слабкі сторони проєкту, а також допомагає сформулювати аргументи для подальшого просування системи на ринку.

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості програмної реалізації системи проактивного моніторингу мережевого електронного документообігу доцільно використати витратний метод. Він полягає у визначенні всіх витрат, необхідних для створення, впровадження та подальшого обслуговування системи, і є найбільш об'єктивним на етапі дослідно-конструкторських робіт.

Такий підхід дозволяє врахувати як прямі витрати (зарплата розробників, ліцензії на програмне забезпечення, вартість серверного обладнання), так і непрямі – витрати на навчання персоналу, тестування, технічну підтримку та оновлення системи.

Перевага витратного методу полягає в тому, що він не потребує наявності ринкових аналогів, адже розрахунок базується на реальних внутрішніх витратах. Це особливо важливо для інноваційних продуктів, які тільки з'являються на ринку.

Водночас доцільно доповнити цей метод елементами прибуткового підходу, розрахувавши можливу економію для підприємства від впровадження системи. Наприклад, зниження витрат на паперовий документообіг або скорочення часу погодження документів можна розглядати як частину фінансового ефекту.

У підсумку витратний метод дозволяє не лише оцінити реальну ціну проекту, а й чітко аргументувати її для потенційних інвесторів або керівництва компанії, що робить його найбільш практичним у цьому випадку.

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

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

Вихідні дані для розрахунку економічної ефективності вносимо в таблицю 7.1.

Економія на заробітній платі через зменшення персоналу - 900 000 грн/рік, економія на матеріальних витратах - 135 000 грн/рік, зниження втрат від затримок документів - 90 000 грн/рік, разом економічний ефект 1 125 000 грн/рік, термін окупності - 0,76 року (~9 місяців), коефіцієнт економічної ефективності (Е) $-(1\ 125\ 000 / 850\ 000) \times 100\% = 132\%$.

Додаткові (немонетарні) ефекти: підвищення дисципліни виконання завдань і прозорості внутрішніх процесів; зменшення кількості помилок при погодженні документів; полегшення аудиту та контролю управлінських процесів; зниження ризиків втрати або дублювання даних; підвищення задоволеності співробітників і швидкості прийняття рішень.

Впровадження системи проактивного моніторингу мережевого електронного документообігу є економічно доцільним, оскільки забезпечує окупність інвестицій менш ніж за рік і коефіцієнт ефективності понад 130%. Окрім фінансової вигоди, система створює значні організаційні переваги – підвищення продуктивності, контроль процесів у реальному часі та зміцнення інформаційної безпеки, що в сукупності формує основу для довгострокового підвищення ефективності управління підприємством.

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

Таблиця 7.1 – Вихідні дані для розрахунків

Показник	До впровадження	Після впровадження	Ефект / Економія
Середня кількість документів на місяць	5 000	5 000	—
Середній час обробки одного документа	20 хв	8 хв	-12 хв (60%)
Кількість працівників, залучених до документообігу	10	7	-3 особи
Середня заробітна плата одного працівника	25 000 грн/міс	25 000 грн/міс	—
Витрати на папір, друк і архівування	180 000 грн/рік	45 000 грн/рік	-135 000 грн
Витрати на втрати через затримки документів	120 000 грн/рік	30 000 грн/рік	-90 000 грн
Вартість розробки і впровадження системи	—	—	850 000 грн

7.5 Пропозиція алгоритму просування проєкту розробки ПЗ

Просування проєкту системи проактивного моніторингу слід розпочати з аналізу ринку та виявлення цільових аудиторій. Це можуть бути великі корпорації, державні структури або освітні установи, де документообіг відіграє важливу роль. Головна мета на цьому етапі – зрозуміти, які проблеми користувачів система може вирішити.

Далі варто створити демонстраційну версію або прототип системи, щоб потенційні клієнти могли оцінити її зручність та ефективність. Така демонстрація часто допомагає переконати керівників у необхідності впровадження рішення, адже вони бачать конкретну економію часу та ресурсів.

Наступним кроком буде інформаційна кампанія: презентації, вебінари, участь у конференціях, публікації в галузевих ЗМІ. Головне – донести, що система не просто автоматизує документообіг, а забезпечує його безперервність і надійність через проактивний моніторинг.

Важливо також вибудувати партнерські відносини з компаніями, які вже займаються цифровими рішеннями для бізнесу. Вони можуть пропонувати систему своїм клієнтам як частину комплексних ІТ-рішень.

Завершальним етапом буде зворотний зв'язок і постійне вдосконалення продукту на основі відгуків користувачів. Це дозволить не лише утримати клієнтів, а й сформувати репутацію надійного та сучасного рішення.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Оптимізація каналів збуту системи проактивного моніторингу передбачає використання як прямих, так і партнерських механізмів. Найпростіший шлях – це прямі продажі великим організаціям через комерційні пропозиції або тендери. Такі клієнти одразу забезпечують стабільний обсяг реалізації.

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

Паралельно можна розвивати онлайн-просування – створити офіційний сайт, де будуть описані можливості системи, приклади успішних впроваджень, а також форма для зворотного зв'язку чи замовлення демо-версії. Це допоможе привернути увагу середніх і малих підприємств.

Варто також налагодити співпрацю з ІТ-компаніями, які займаються автоматизацією бізнес-процесів, адже вони можуть виступати посередниками у продажах і рекомендувати систему своїм клієнтам як перевірене рішення.

Ще одним дієвим способом може бути пропозиція системи за моделлю SaaS (Software as a Service) – тобто на умовах підписки. Це зменшує бар'єр входу для клієнтів і дозволяє поступово розширювати базу користувачів.

У результаті, оптимізація каналів збуту має спиратися на гнучкість, партнерство та прозорість комунікацій, щоб зробити продукт доступним для максимально широкої аудиторії.

7.7 Визначення ключових факторів успіху конкретного проєкту

Першим і найважливішим фактором успіху системи проактивного моніторингу є її надійність. Користувачі повинні бути впевнені, що система не лише фіксує помилки, а й своєчасно їх попереджає, забезпечуючи безперервність документообігу.

Не менш важливим є зручність інтерфейсу. Якщо користувачам легко працювати з системою, вони швидше приймають її і починають активно використовувати, що збільшує цінність продукту.

Велике значення має також адаптивність і масштабованість. Система має легко інтегруватися з іншими інформаційними рішеннями підприємства, а також підтримувати можливість розширення функцій у майбутньому.

Довіра до продукту формується через стабільну технічну підтримку, відкритість комунікацій і позитивні відгуки клієнтів. Саме ці чинники забезпечують репутацію, яка стає найкращою рекламою.

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

І нарешті, ключовим є професіоналізм команди розробників і їхня готовність удосконалювати систему відповідно до потреб ринку. Без цього навіть найкраща технологія залишиться лише прототипом, тоді як жива команда перетворює її на реальний, успішний продукт.

КБПЗ_2025

					VKPM-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Законом України “Про охорону праці” [1] регламентуються загальні положення державної політики в галузі охорони праці, а реалізуються 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» [3], 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.

У розділі даної магістерської роботи висвітлюються основні питання охорони праці працівників, робота яких пов’язана з роботою за комп’ютером, планування робочого приміщення, де працюють користувачі ПК; параметри мікроклімату, освітленість робочих місць та виробничих приміщень; шумові завади.

Для того, щоб об’єктивно проаналізувати відповідність умов праці діючим нормативно-правовим актам, необхідно здійснити санітарно-гігієнічну характеристику умов праці відділу, в якому працює програміст, над розробкою даного програмного продукту.

В зв’язку з цим необхідно сконцентрувати увагу на небезпечних і шкідливих чинниках пов’язаних з постійною роботою за комп’ютером.

Робота з комп’ютером характеризується значною розумовою напругою і нервово-емоційним навантаженням операторів, високою напруженістю зорової

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

роботи і достатньо великим навантаженням на м'язи рук при роботі з клавіатурою ЕОМ.

Електробезпека є одним із критичних питань для співробітників, що працюють із технікою, яка одержує живлення з електричної мережі. При невиконанні норм електробезпеки можлива поразка електричним струмом.

8.2 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Розглянемо вимоги до приміщень, в яких буде розміщуватися робоче місце. Площа на одне робоче місце повинна становити мінімум 6,0 м², при цьому об'єм – мінімум 20,0 м³ [2, 4]. Розміщення робочих місць у підвальних приміщеннях, а також на цокольних поверхах заборонено.

У відповідності до НПАОП 0.00-7.15-18, приміщення повинні мати природне та штучне освітлення. Приміщення не повинні межувати з іншими приміщеннями, в яких рівні шуму і вібрації перевищують допустимі значення. Покриття підлоги повинне бути матовим з коефіцієнтом відбиття 0,3-0,5. Для внутрішнього оздоблення приміщень слід використовувати дифузно-відбивні матеріали з коефіцієнтами відбиття для стелі 0,7-0,8, для стін 0,5-0,6 [4].

Саме робоче місце теж повинно відповідати вимогам, що описані в [2, 4]. Конструкція робочого місця повинна забезпечити підтримання оптимальної робочої пози. У відповідності до НПАОП 0.00-7.15-18, обладнання і організація робочого місця працюючих з ЕОМ мають забезпечувати відповідність конструкції всіх елементів робочого місця та їх взаємного, розташування ергономічним вимогам з урахуванням характеру і особливостей трудової діяльності.

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

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

8.3 Розробка заходів з умов поліпшення охорони праці

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

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

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

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

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

Приміщення повинні бути оснащені аптечками першої медичної допомоги, а також обов'язковим є щоденне вологе прибирання приміщень.

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

8.4 Техніка безпеки та протипожежна профілактика

Електронно-обчислювальні машини (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 тому числі програмісти) необхідно створити належний мікроклімат, параметри якого регламентуються Державними санітарними правилами і нормами, зокрема ДСанПіН 3.3.2.007-98.

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

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

Вимоги безпеки при роботі з ПК визначено в НПАОП 0.00-1.28-18. Згідно вимог електробезпеки, ПК повинні підключатися до електромережі тільки за допомогою справних штепсельних з'єднань. Не допускається підключати ПК до

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

звичайної двопровідної електромережі, в тому числі з використанням перехідних пристроїв. Електромережі штепсельних з'єднань та електророзеток для живлення ПК потрібно виконувати за магістральною схемою. Щодо безпеки при роботі з ПК, щодня перед початком роботи необхідно очищати монітор від пилу та інших забруднень. Після закінчення роботи з ПК, він та периферійні пристрої повинні бути відключені від електричної мережі. У разі виникнення певної аварійної ситуації необхідно негайно відключити ПК від електричної мережі. Не допускається виконувати обслуговування, ремонт та налагодження ПК безпосередньо на робочому місці [2]. Згідно з [4], на та під приміщеннями, в яких розміщені ЕОМ, а також у суміжних із ними приміщеннях не дозволяється розташування приміщень категорій А та Б за вибухопожежною небезпекою. Фальшпідлога у приміщеннях з ЕОМ має бути з негорючих матеріалів або матеріалів груп горючості Г1, Г2 з межею вогнестійкості не менше 0,5 години. Простір під нею слід розділяти негорючими діафрагмами на відсіки площею не більше 250 м². Діафрагми повинні мати межу вогнестійкості не менше 0,75 год. Звукопоглинаюче облицювання стін та стель цих приміщень слід виготовляти з негорючих матеріалів або матеріалів груп горючості Г1, Г2. Персональні комп'ютери після закінчення роботи повинні відключатися від мережі. Не рідше одного разу на квартал необхідно очищати від пилу агрегати та вузли, кабельні канали та простір між підлогами [4]. Приміщення повинні бути забезпечені первинними засобами пожежогасіння, а саме вогнегасниками, що використовуються для локалізації і ліквідації пожеж у їх початковій стадії розвитку.

Вогнегасники слід встановлювати у легкодоступних та помітних місцях (коридорах, біля входів або виходів з приміщень тощо), а також у пожежонебезпечних місцях, де найбільш вірогідна поява осередків пожежі. При цьому необхідно забезпечити їх захист від попадання прямих сонячних променів та безпосередньої (без загороджувальних щитків) дії опалювальних та нагрівальних приладів. Вибір типу та необхідна кількість вогнегасників

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

визначається відповідно до Типових норм належності вогнегасників.

Вогнегасники, допущені до введення в експлуатацію, повинні мати:

- облікові (інвентарні) номери за прийнятою на об'єкті системою нумерації;

- пломби на пристроях ручного пуску;

- бирки та маркувальні написи на корпусі, червоне сигнальне пофарбування згідно з державними стандартами [4].

У відповідності до [4] не дозволяється:

- відкрите прокладання електропроводів і кабелів транзитом через пожежонебезпечні і вибухонебезпечні зони будь-якого класу і ближче 1 м і 5 м від них відповідно, а також у сходових клітках;

- експлуатація кабелів і проводів з пошкодженою або такою, що в процесі експлуатації втратила захисні властивості, ізоляцією;

- залишення під напругою кабелів та проводів з неізольованими струмопровідними жилами;

- застосування саморобних подовжувачів, які не відповідають вимогам ПУЕ, що пред'являються до переносних (пересувних) електропроводок;

- заклеювати шпалерами відкрито прокладені електропроводи і кабелі.

8.5 Розрахункова частина

Для забезпечення безпечної роботи необхідне виконання вимог електричної безпеки, оскільки все офісне обладнання заживлюється від електричної мережі. Одним з необхідних засобів електричної безпеки є встановлення захисного заземлення.

Початкові дані, необхідні для розрахунку захисного заземлення:

- допустимий опір розповсюдженню струму в землі від заземлювального пристрою $R_{\text{зн}} = 10 \text{ Ом}$;

- питомий опір ґрунту в місці встановлення заземлювача $\rho_3 = 100 \text{ Ом/м}$;

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

– тип ґрунту – суглинок;
 – тип заземлювача – труба, діаметром $d=0.05$ м і довжиною $l = 2.5$ м; –
 конструкція заземлювача – розташування заземлювачів по контуру. Розрахунок
 проводимо за стандартною методикою [7].

Визначимо розрахунковий опір землі:

$$\rho_{pz} = \phi \cdot \rho_3$$

де ϕ – коефіцієнт сезонності, що враховує коливання питомого опору при зміні
 вологості ґрунту протягом року; при використанні заземлювача довжиною $l = 2.5$
 м при глибині закладання від вершини $h = 0.5$ м $\phi = 1.1$ для четвертої кліматичної
 зони.

Схема розташування заземлювачів показана на рисунку 8.1.

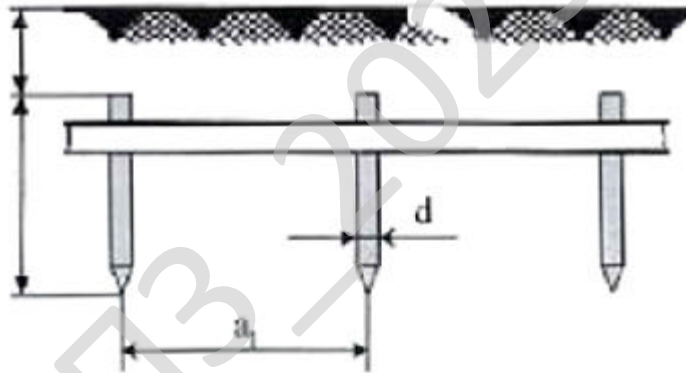


Рисунок 8.1 – Схема розташування заземлювачів

Опір землі:

$$\rho_{pz} = 1,1 \cdot 100 = 110 \text{ Ом}\cdot\text{м}$$

Опір R_B , розповсюдженню струму в землі від одного вертикального
 заземлювача:

$$R_B = \frac{\rho_{pz}}{2\pi \cdot l} \left(\ln \frac{2 \cdot l}{d} + 0.5 \ln \frac{4t+l}{4t-l} \right)$$

де

l – довжина заземлювача ($l = 2.5$ м);

$d = 0.05$ м – діаметр заземлювача при $U < 1$ кВ та при $S < 100$ кВА;

t – відстань від поверхні до середини заземлювача:

$$t = h + l/2 = 0.5 + 2.5/2 = 1.75 \text{ м.}$$

$$R_B = \frac{110}{2 \cdot 3.14 \cdot 2.5} \left(\ln \left(\frac{2 \cdot 2.5}{0.05} \right) + 0.5 \cdot \ln \left(\frac{4 \cdot 1.75 + 2.5}{4 \cdot 1.75 - 2.5} \right) \right) = 34.325 \text{ Ом}$$

Визначаємо потрібну кількість заземлювачів:

$$n' = \frac{R_B}{R_{3Н}} = \frac{34.325}{10} = 3.4 \approx 4 \text{ шт.}$$

Коефіцієнт використання вертикальних заземлювачів враховує ефект екранування. При вибраному значенні $k = a/l$, де a – відстань між вертикальними заземлювачами, м; $k = 1$ при $a = 2$ м. Таким чином, коефіцієнт використання вертикального заземлювача за довідковими даними дорівнює $\eta_B = 0,6$.

Кількість вертикальних заземлювачів з урахуванням коефіцієнту використання η_B приблизно складає

$$n = \frac{R_B}{R_{3Н} \cdot \eta_B} = \frac{34.325}{10 \cdot 0.6} = 5.72 \approx 6 \text{ шт.}$$

Довжина горизонтального заземлювача, необхідна для розміщення вертикальних заземлювачів по контуру

$$L = a \cdot n = 2 \cdot 6 = 12 \text{ м}$$

Опір горизонтального заземлювача R_G , Ом, прокладеного на глибині $h = 0.5$ м від поверхні землі буде

$$R_G = \frac{R_{pz}}{2 \cdot 3.14 \cdot L} \cdot \ln \frac{2 \cdot L^2}{b \cdot h} = \frac{110}{2 \cdot 3.14 \cdot 12} \cdot \ln \frac{2 \cdot 12^2}{0.04 \cdot 0.5} = 13.976 \text{ Ом}$$

де $b = 0.04$ м – ширина сталевієї смуги, з якої виготовлений заземлювач.

Обчислюємо загальний опір:

$$R_3 = \frac{R_B \cdot R_G}{n \cdot R_G \cdot \eta_B + R_B \cdot \eta_B} = \frac{34.325 \cdot 13.976}{6 \cdot 13.976 \cdot 0.6 + 34.325 \cdot 0.34} = 7.65 \text{ Ом}$$

де η_G – коефіцієнт використання горизонтального заземлювача ($\eta_G = 0.34$).

Маємо $7.65 \text{ Ом} < 10 \text{ Ом}$ (за потужності генераторів та трансформаторів 100 кВт і менше), отже нормативне обмеження $R_3 < R_{3,норм}$ виконується.

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи проактивного моніторингу мережевого електронного документообігу.

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 проактивного моніторингу мережевого електронного документообігу.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем проактивного моніторингу мережевого електронного документообігу.
- Досліджена система проактивного моніторингу мережевого електронного документообігу.
- На основі отриманих результатів досліджень створена програмна реалізація системи проактивного моніторингу мережевого електронного документообігу.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання проактивного моніторингу мережевого електронного документообігу.

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-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

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 Visual C++. 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 Camellia.

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-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Батрак С.С. Дослідження та програмна реалізація системи проактивного моніторингу мережевого електронного документообігу // Збірник праць молодих науковців ЦНТУ. – Вип. 15. – Кропивницький: ЦНТУ, 2025.
2. Doug Lowe «Networking For Dummies 12th Edition». 2020. – 480 p.
3. Ramon Nastase «Computer Networking: The Beginner’s guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.
4. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
5. 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.
6. 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.
7. 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.
8. 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.

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

9. 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.

10. 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.

11. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchев, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.

12. 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.

13. 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.

14. 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.

15. 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.

16. 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

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 3(69). С. 93-98.

17. 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.

18. 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.

19. 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.

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

21. 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.

22. 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.

23. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V.,

Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587.

24. 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.

25. 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.

26. 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.

27. 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.

28. 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.

29. 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.

30. 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.

31. 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.*

32. 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.*

33. 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.*

34. 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.*

35. 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.*

36. 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.*

37. 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*

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

38. 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.

39. 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.

40. 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.

41. 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.

42. 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.

43. 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.

44. 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).

45. 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

46. 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.

47. 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.

48. 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.

49. 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.

50. Смірнов О.А., Котелянець В.В. Стійкі до колізій стохастичні моделі функціонування безпроводових сенсорних мереж. Вісник інженерної академії України, №3, с. 145-152, 2018

51. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Алгоритми формування безлічі маршрутів передачі метаданих у антивірусні хмарні системи. Збірник наукових праць "Системи обробки інформації". – Випуск 5 (142). – Х.: ХУПС – 2016. – С. 148-152.

					ВКРМ-123.25.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90