

Центральноукраїнський національний технічний університет  
Центр заочної та дистанційної освіти  
Кафедра кібербезпеки та програмного забезпечення

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

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА**  
**за другим (магістерським) рівнем вищої освіти**  
на тему  
**“Дослідження та програмна реалізація системи автоматичного  
прийняття рішення для імітації діяльності людини”**

Виконав здобувач вищої освіти  
II курсу, групи КН-23Мз  
ОПП «Комп’ютерні науки»  
спеціальності 122 «Комп’ютерні науки»  
\_\_\_\_\_ Бурейко Д.Г.  
« \_\_\_\_ » \_\_\_\_\_ 2024 р.

Керівник проекту  
кандидат технічних наук  
\_\_\_\_\_ Смірнова Т.В.  
« \_\_\_\_ » \_\_\_\_\_ 2024 р.  
Рецензент \_\_\_\_\_  
\_\_\_\_\_

**Центральноукраїнський національний технічний університет**

Центр *Заочної та дистанційної освіти*

Кафедра *Кібербезпеки та програмного забезпечення*

Рівень вищої освіти *магістр*

Галузь знань 12 *“Інформаційні технології”*

Спеціальність 122 *“Комп’ютерні науки”*

Освітньо-професійна (освітньо-наукова) програма *“Комп’ютерні науки”*

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2024 року

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

*Бурейку Дмитру Григоровичу*

(прізвище, ім'я, по батькові)

1. Тема роботи

*Дослідження та програмна реалізація системи автоматичного прийняття рішення для імітації діяльності людини*

2. Керівник роботи

*Смірнова Тетяна Віталіївна, канд. техн. наук*

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

затверджені наказом вищого навчального закладу № 20-13 від 07.08.2024 року

3. Строк подання студентом роботи до захисту 2.12.2024 р.

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи автоматичного прийняття рішення для імітації діяльності людини*

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

*1. Призначення та область використання.*

*6. Наукова новизна.*

*2. Перегляд аналогічних існуючих систем.*

*7. Маркетингове та економічне обґрунтування ІТ-проєкту.*

*3. Опис і обґрунтування проєктних рішень.*

*8. Заходи з охорони праці та техніки безпеки.*

*4. Етапи програмування системи.*

*9. Висновки.*

*5. Впровадження системи в промислову експлуатацію*

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

*Наукова новизна*

*1 аркуш*

*Структурна схема системи*

*1 аркуш*

*Функціональна схема системи*

*1 аркуш*

*Діаграма процесів*

*1 аркуш*

*Блок-схема алгоритму роботи додатку*

*2 аркуша*

*Показники економічної ефективності*

*1 аркуш*

## 6. Консультанти розділів роботи

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

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

## КАЛЕНДАРНИЙ ПЛАН

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

Дата видачі завдання  
« 6 » вересня 2024 р.

Підпис керівника

\_\_\_\_\_ (прізвище та ініціали)

Завдання прийнято до виконання  
« 6 » вересня 2024 р.

Підпис здобувача

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

**Бурейко Д.Г. Дослідження та програмна реалізація системи автоматичного прийняття рішення для імітації діяльності людини. 122 Комп'ютерні науки. Центральнoукраїнський національний технічний університет. Кропивницький. 2024.**

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи автоматичного прийняття рішення для імітації діяльності людини.

Метою розробки є дослідження та програмна реалізація системи автоматичного прийняття рішення для імітації діяльності людини.

Об'єктом дослідження є процес автоматичного прийняття рішення для імітації діяльності людини.

Предметом дослідження є методи автоматичного прийняття рішення для імітації діяльності людини.

Методи дослідження базуються на методах машинного навчання, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи автоматичного прийняття рішення для імітації діяльності людини.

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

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

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

Програму розроблено в середовищі Python.

**Ключові слова:** Комп'ютерні науки, автоматичне прийняття рішення, імітація діяльності людини

## ABSTRACT

**Bureyko D.G. Research and software implementation of an automatic decision-making system for simulating human activity. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.**

In this final qualification work for the second (master's) level of higher education, software was developed, which is intended for an automatic decision-making system for simulating human activity.

The goal of development is research and software implementation of an automatic decision-making system for simulating human activity.

The object of research is the process of automatic decision-making to simulate human activity.

The subject of the research is methods of automatic decision-making for imitating human activity.

Research methods are based on machine learning methods, mathematical statistics methods, and software development methods.

The result of the work is a software implementation of an automatic decision-making system for simulating human activity.

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

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

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

The program was developed in the Python environment.

**Keywords:** Computer science, automatic decision-making, simulation of human activity

## ЗМІСТ

|   |    |
|---|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ .....   | 3  |
| ВСТУП.....  | 4  |
| 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ .....   | 6  |
| 1.1 Призначення системи.....  | 6  |
| 1.2 Область застосування.....   | 7  |
| 2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....  | 9  |
| 2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти..... | 9  |
| 2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....  | 21 |
| 2.3 Розгорнута постановка завдання .....  | 23 |
| 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....   | 25 |
| 3.1 Опис функціонування системи .....   | 25 |
| 3.2 Розробка структурної схеми.....   | 28 |
| 3.3 Розробка функціональної схеми .....   | 36 |
| 3.4 Розробка діаграми процесів.....   | 64 |
| 4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....  | 66 |
| 4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....   | 66 |
| 4.2 Захист розробленого програмного забезпечення.....   | 82 |
| 5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....  | 84 |
| 6 НАУКОВА НОВИЗНА .....   | 90 |

|          |                |          |       |      |  |                           |       |         |
|----------|----------------|----------|-------|------|--|---------------------------|-------|---------|
|          |                |          |       |      |  | ВКРМ-122.24.0001.00.00.ПЗ |       |         |
| Вим      | Арк.           | № докум. | Підп. | Дата | Дослідження та програмна реалізація системи автоматичного прийняття рішення для імітації діяльності людини | Літ.                      | Аркуш | Аркушів |
| Розроб.  | Бурейко Д.Г.   |          |       |      |  | М                         | 1     | 118     |
| Перев.   | Смірнова Т.В.  |          |       |      |  |                           |       |         |
| Н.контр. | Коваленко А.С. |          |       |      |  | ЦНТУ КН-23Мз              |       |         |
| Затв.    | Смірнов О.А.   |          |       |      |  |                           |       |         |

|     |  |     |
|-----|--|-----|
| 7   | МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ .....  | 91  |
| 7.1 | Визначення цільової аудиторії кінцевого готового продукту .....  | 91  |
| 7.2 | Оцінка привабливості шляхом застосування методів експертних оцінок ...                                   | 92  |
| 7.3 | Вибір методу оцінки вартості ПЗ .....  | 94  |
| 7.4 | Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості..... | 95  |
| 7.5 | Пропозиція алгоритму просування проєкту розробки ПЗ .....  | 97  |
| 7.6 | Оптимізація каналів збуту та шляхів реалізації ПЗ .....  | 97  |
| 7.7 | Визначення ключових факторів успіху конкретного проєкту.....   | 99  |
| 8   | ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ .....  | 100 |
| 8.1 | Вступ.....   | 100 |
| 8.2 | Пожежна безпека.....   | 101 |
| 8.3 | Характеристика умов праці програміста .....  | 103 |
| 8.4 | Розробка заходів з умов поліпшення охорони праці.....  | 105 |
| 8.5 | Розрахункова частина .....   | 105 |
| 9   | ОСНОВНІ ВИСНОВКИ.....  | 109 |
|     | СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....   | 111 |

КБПЗ-2024

|      |      |          |        |      |                                  |          |
|------|------|----------|--------|------|----------------------------------|----------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк.     |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | <b>2</b> |

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

|      |   |   |
|------|---|---|
| БД   | – | база даних  |
| БЗ   | – | база знань  |
| БП   | – | база правил   |
| ВД   | – | вітрина даних   |
| ВС   | – | віртуальні співрозмовники                                     |
| ГІС  | – | гібридна інтелектуальна система                               |
| ЕС   | – | експертна система   |
| ЗД   | – | зберігання даних  |
| ЗСП  | – | збалансована система показників                               |
| ЖЦ   | – | життєвий цикл   |
| ІАД  | – | інтелектуальний аналіз даних                                  |
| ІС   | – | інтелектуальні інформаційні системи                           |
| ІТ   | – | інтелектуальні інформаційні технології                        |
| ІС   | – | інтелектуальна система  |
| ІСПР | – | інформаційна система підтримки рішень                         |
| ІСУП | – | інтегровані системи управління підприємством                  |
| КВ   | – | коефіцієнти впевненості                                       |
| ЛПР  | – | людина, яка приймає рішення                                   |
| МЛВ  | – | механізм логічного виводу                                     |
| ОПР  | – | особа, яка приймає рішення                                    |
| ПЗ   | – | програмне забезпечення  |
| РП   | – | робоча пам'ять  |
| СЗД  | – | система зберігання даних                                      |
| СПР  | – | система підтримки рішень                                      |
| СППР | – | система підтримки прийняття рішень                            |
| СУБД | – | система управління базами даних                               |
| УР   | – | управлінське рішення  |
| OLAP | – | On-Line Analytical Processing – оперативна аналітична обробка |

## ВСТУП

**Актуальність теми.** Система автоматичного прийняття рішення для імітації діяльності людини (САПІДЛ) призначена для профілювання, яке аналізує аспекти особистості, поведінки, інтересів і звичок людини, щоб робити прогнози або приймати рішення щодо них. Організації отримують особисту інформацію про людей із різних джерел. Дані про пошук в Інтернеті, купівельні звички, спосіб життя та поведінку, зібрані з мобільних телефонів, соціальних мереж, систем відеоспостереження та Інтернету речей, є прикладами типів даних, які організації можуть збирати. Вони аналізують цю інформацію, щоб класифікувати людей на різні групи чи сектори. Цей аналіз визначає кореляції між різними поведінками та характеристиками для створення профілів для окремих осіб. У цьому профілі будуть нові особисті дані про цю особу.

Організації використовують профілювання, щоб:

- дізнатися щось про індивідуальні переваги;
- прогнозувати їхню поведінку; та/або
- приймати рішення щодо них.

**Мета й завдання дослідження.** Метою роботи є дослідження та програмна реалізація системи автоматичного прийняття рішення для імітації діяльності людини.

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

- Огляд існуючих систем автоматичного прийняття рішення для імітації діяльності людини.
- Дослідження системи автоматичного прийняття рішення для імітації діяльності людини.
- Програмна реалізація системи автоматичного прийняття рішення для імітації діяльності людини.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-122.24.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 4    |

*Об'єктом дослідження є процес автоматичного прийняття рішення для імітації діяльності людини.*

*Предметом дослідження є методи автоматичного прийняття рішення для імітації діяльності людини.*

*Методи дослідження базуються на методах машинного навчання, методах математичної статистики, методах розробки програмного забезпечення.*

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

– Удосконалено метод автоматичного прийняття рішення для імітації діяльності людини.

– Розроблено вітчизняний продукт автоматичного прийняття рішення для імітації діяльності людини, який має більш широкі можливості, на відміну від існуючих аналогів.

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі автоматичного прийняття рішення для імітації діяльності людини.

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

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

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи автоматичного прийняття рішення для імітації діяльності людини, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-122.24.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 5    |

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Профільювання може використовувати алгоритми. Алгоритм – це послідовність інструкцій або набір правил, призначених для виконання завдання або вирішення проблеми. Профільювання використовує алгоритми для пошуку кореляції між окремими наборами даних. Потім ці алгоритми можна використовувати для прийняття широкого діапазону рішень, наприклад, для прогнозування поведінки або контролю доступу до служби. Системи штучного інтелекту (AI) і машинне навчання все частіше використовуються для створення та застосування алгоритмів. Більше інформації про алгоритми, ШІ та машинне навчання можна знайти в нашій статті про великі дані, штучний інтелект, машинне навчання та захист даних.

Ви здійснюєте профільювання, якщо:

- збирати та аналізувати персональні дані у великому масштабі, використовуючи алгоритми, штучний інтелект або машинне навчання;
- ідентифікувати асоціації для побудови зв'язків між різними поведінками та атрибутами;
- створювати профілі, які ви застосовуєте до окремих осіб; або
- прогнозувати поведінку людей на основі призначених їм профілів.

Хоча багато людей вважають маркетинг найпоширенішою причиною профільювання, це не єдине застосування.

Профільювання використовується в деяких медичних процедурах шляхом застосування машинного навчання для прогнозування стану здоров'я пацієнтів або ймовірності успішного лікування для конкретного пацієнта на основі певних групових характеристик.

Менш очевидні форми профільювання включають висновки з очевидно не

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 6    |

пов'язаних між собою аспектів поведінки окремих осіб.

Використання дописів у соціальних мережах для аналізу особистостей водіїв автомобіля за допомогою алгоритму для аналізу слів і фраз, які пропонують «безпечне» та «небезпечне» водіння, щоб призначити рівень ризику для особи та відповідно встановити її страхову премію.

## 1.2 Область застосування

Автоматизоване прийняття рішень – це процес прийняття рішення автоматизованими засобами без будь-якої участі людини. Ці рішення можуть ґрунтуватися на фактичних даних, а також на цифрових профілях або припущеннях. Приклади цього:

- онлайн рішення про видачу кредиту;
- тест на здібності, який використовується для набору персоналу і використовує заздалегідь запрограмовані алгоритми та критерії.

Автоматизоване прийняття рішень часто включає профілювання, але це не обов'язково.

Екзаменаційна комісія використовує автоматизовану систему для позначення бланків відповідей на екзамен з кількома варіантами відповідей. Система попередньо запрограмована на кількість правильних відповідей, необхідних для отримання прохідних і відмінних балів. Бали автоматично приписуються кандидатам на основі кількості правильних відповідей, а результати доступні в Інтернеті.

Це автоматизований процес прийняття рішень, який не передбачає профілювання.

Які переваги профілювання та автоматизованого прийняття рішень?

Профілювання та автоматизоване прийняття рішень може бути дуже корисним для організацій, а також принести користь людям у багатьох секторах, включаючи охорону здоров'я, освіту, фінансові послуги та маркетинг. Вони

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-122.24.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 7    |

можуть призвести до швидших і послідовніших рішень, особливо у випадках, коли потрібно проаналізувати дуже великий обсяг даних і прийняти рішення дуже швидко.

Які ризики?

Хоча ці методи можуть бути корисними, існують потенційні ризики:

- Профілювання часто невидиме для окремих осіб.
- Люди можуть не очікувати, що їхня особиста інформація буде використана таким чином.
- Люди можуть не розуміти, як працює процес або як він може на них вплинути.
- Прийняті рішення можуть призвести до значних негативних наслідків для деяких людей.

Те, що аналіз даних знаходить кореляцію, не означає, що це суттєво. Оскільки процес може зробити лише припущення щодо чиєїсь поведінки чи характеристик, завжди буде певна похибка, і для зважування ризиків використання результатів потрібен баланс.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи автоматичного прийняття рішення для імітації діяльності людини, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 8    |

## 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**

### **Що означає автоматизоване прийняття рішень?**

Автоматизоване прийняття рішень (САПІДЛ) стосується процесу, у якому технології, а не люди, приймають рішення на основі аналізу даних і алгоритмів. Ця технологія варіюється від простих автоматизованих виборів у повсякденних гаджетах до складних алгоритмів, що керують основними бізнес-рішеннями. Завдяки інтеграції величезних наборів даних і складних алгоритмів, САПІДЛ пропонує неперевершену ефективність і точність, революціонізуючи процес прийняття рішень.

### **Чому ми повинні турбуватися про САПІДЛ?**

Хоча САПІДЛ відкриває чудові можливості, він також викликає серйозні занепокоєння, зокрема проблеми, пов'язані з упередженістю, відсутністю прозорості та підзвітності. Оскільки ці системи дедалі більше впливають на важливі сфери нашого життя, розуміння їх наслідків має вирішальне значення. Це вимагає збалансованого підходу, у якому переваги САПІДЛ використовуються з одночасним активним усуненням потенційних ризиків, як наголошує Європейська рада із захисту даних.

### **Приклади автоматизованого прийняття рішень у реальному світі**

#### **Персональні послуги**

– Автоматичні платежі за рахунками: повністю автоматизовані після налаштування, що потребують мінімального або повного втручання користувача. Ці системи автоматично оплачують рахунки за графіком.

– Сортування електронної пошти та фільтрація спаму: вони

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 9    |

високоавтоматизовані, безперервно працюють без участі користувача. Він автоматично класифікує електронні листи та фільтрує спам.

– Навігаційні програми: розрахунок маршруту повністю автоматизований, але вибір користувача може впливати на маршрути. Навігаційні програми пропонують автоматичні маршрути до місць призначення.

– Пристрої «Розумний дім»: здебільшого автоматизовані, особливо для рутинних завдань, але можуть вимагати початкового налаштування або час від часу налаштування. Такі пристрої, як розумні термостати, автоматизують домашні функції.

– Стрічки соціальних медіа: автоматичне курування вмісту, але під впливом взаємодії користувачів. Платформи соціальних медіа використовують алгоритми для показу релевантного вмісту.

– Голосові помічники: працюють за командами користувача, надаючи автоматизовані послуги, але залежно від введення користувача. Голосові помічники реагують на голосові команди користувача.

– Веб-сайти електронної комерції та рекомендації: надають автоматичні рекомендації, але залежать від історії перегляду та покупок користувачів. Сайти електронної комерції пропонують продукти на основі поведінки користувачів.

– Персоналізовані рекомендації щодо розваг: пропонуйте вміст на основі історії користувача, але вибір і відгуки можуть покращити пропозиції. Поточкові платформи використовують алгоритми для рекомендацій вмісту.

– Агрегатори новин: автоматичне курування новин, часто включаючи персоналізацію на основі вподобань користувача. Агрегатори новин збирають і представляють новинні статті.

– Фітнес-трекери та додатки для здоров'я: автоматично відстежують та надають статистику, але введення користувачами покращує функціональність. Ці програми відстежують показники здоров'я та фізичної форми.

– Відстеження сну: автоматично відстежує режим сну, але залучення користувачів підвищує точність і корисність. Трекери сну записують особисті

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 10   |

дані для аналізу.

– Фотографія зі смартфона: автоматично регулює параметри для отримання оптимальних фотографій, але може змінюватись вручну. Програми камери автоматично налаштовують параметри фотографій.

– Пропозиції щодо приготування їжі та рецептів: пропонуйте рецепти на основі інгредієнтів і вподобань, але приготування їжі відбувається вручну. Додатки рецептів пропонують страви для приготування.

– Інструменти автоматизованого планування відпустки: пропонує варіанти подорожей на основі вподобань, але вимагає рішень користувача для остаточного планування. Ці інструменти допомагають у плануванні подорожі.

### **Роздрібна торгівля та електронна комерція**

– Динамічні моделі ціноутворення: висока автоматизація, постійний аналіз ринкових даних і поведінки клієнтів для коригування цін у режимі реального часу. Ці моделі автоматично встановлюють ціни на основі даних у реальному часі.

– Управління запасами: загалом автоматизоване з використанням алгоритмів для відстеження та зміни порядку запасів на основі даних про продажі та тенденцій. Системи управління запасами автоматизують відстеження та поповнення запасів.

– Механізми рекомендацій клієнтів: значною мірою автоматизовані, пропонують продукти на основі даних клієнтів, але під впливом зміни поведінки споживачів. Ці механізми використовують дані, щоб пропонувати продукти клієнтам.

– Планування завдань у магазині: це передбачає певний рівень автоматизації планування завдань і розподілу ресурсів, але часто вимагає людського контролю та прийняття рішень. Цей процес автоматизує деякі завдання в магазині, але покладається на керування людьми.

### **Маркетинг і реклама**

– Онлайн-реклама: висока автоматизація, особливо в програмних місцях

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 11   |

розміщення оголошень на основі даних і поведінки користувачів. Інтернет-реклама передбачає автоматичне розміщення реклами на основі персональних даних.

– Системи цільової реклами: автоматично націлюйте користувачів на основі демографічних показників, поведінки та вподобань. Системи цільової реклами використовують автоматизацію для націлювання на користувачів релевантної реклами.

– Персоналізація вмісту: значною мірою автоматизована підготовка вмісту для окремих користувачів, але може включати деякі елементи, які курує людина. Персоналізація вмісту автоматизує налаштування вмісту, але може включати елементи, які курує людина.

– Персоналізований маркетинг: автоматизовано створення індивідуальних маркетингових повідомлень, але часто включає стратегічний внесок маркетологів. Персоналізований маркетинг автоматизує створення індивідуальних маркетингових повідомлень.

– Аналіз ефективності кампанії: він використовує автоматизовані інструменти для аналізу даних, але стратегічна інтерпретація часто вимагає людського розуміння. Аналіз ефективності кампанії використовує автоматизацію для аналізу даних, але покладається на людське розуміння для інтерпретації.

– Аналіз настрою: автоматизована обробка та аналіз великих обсягів тексту, нюанси розуміння можуть отримати користь від людського нагляду. Аналіз настроїв автоматизує аналіз тексту, але має переваги від людського контролю для детального розуміння.

### **Управління людськими ресурсами**

– Інструменти перевірки резюме: висока автоматизація фільтрації резюме на основі ключових слів і критеріїв. Ці інструменти автоматизують початкову перевірку резюме кандидатів на роботу.

– Перевірка претендентів на роботу: схожа на перевірку резюме, часто автоматизована для початкового сортування кандидатів. Відбір кандидатів на

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 12   |

роботу автоматизує початкове сортування кандидатів на роботу за критеріями.

– Аналіз ефективності роботи співробітників: автоматизований збір і аналіз даних, але може знадобитися людська інтерпретація для детального оцінювання. Аналіз продуктивності співробітників автоматизує збір і аналіз даних, але покладається на людську інтерпретацію для нюансованих оцінок.

– Інструменти планування персоналу: це передбачає автоматизацію аналізу потреб у персоналі, але стратегічне планування часто передбачає значний людський внесок. Інструменти планування персоналу автоматизують аналіз потреб у персоналі, але залучають участь людини в стратегічному плануванні.

### **Фінансові послуги**

– Автоматизована торгівля: високоавтоматизована, виконання угод на основі алгоритмів з мінімальним втручанням людини. Автоматизована торгівля передбачає використання алгоритмів для здійснення угод на фінансових ринках.

– Системи виявлення шахрайства: дуже автоматизовані, постійно відстежують транзакції для виявлення шахрайства. Ці системи використовують автоматизацію для виявлення та запобігання шахрайським транзакціям.

– Кредитний скоринг: значною мірою автоматизований, з використанням алгоритмів для оцінки кредитоспроможності на основі особистих даних. Кредитний скоринг автоматизує оцінку кредитного ризику особи.

– Автоматизована обробка позики: значною мірою автоматизована, оцінка заявок на позику за встановленими критеріями. Цей процес автоматизує оцінку кредитних заявок.

– Управління інвестиційним портфелем: залежить від автоматизації; деякі з них високоавтоматизовані (робо-консультанти), а інші вимагають більшого людського контролю. Управління інвестиційним портфелем може варіюватися від повністю автоматизованого до прийняття рішень під керівництвом людини.

– Моделі оцінки ризиків: автоматизовані в аналізі даних, але часто вимагають людської інтерпретації та прийняття рішень. Ці моделі використовують автоматику для аналізу ризиків, але передбачають оцінку

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 13   |

людини.

– Законодавство щодо боротьби з відмиванням грошей: це нормативна база, а не автоматизована система. Він керує проектуванням автоматизованих систем для відповідності, але сам по собі не є автоматизованим. Законодавство щодо боротьби з відмиванням грошей містить вказівки щодо запобігання відмиванню грошей, але не автоматизує процес.

### **Охорона здоров'я**

– Діагностика охорони здоров'я: висока автоматизація, особливо при аналізі медичних зображень або лабораторних результатів. Медична діагностика передбачає автоматизований аналіз медичних даних для встановлення діагнозу.

– Інструменти діагностичної допомоги: зазвичай автоматизовані для надання діагностичних пропозицій, але часто використовуються в поєднанні з людським медичним досвідом. Ці інструменти допомагають у діагностиці, але потребують досвіду людини.

– Віддалений моніторинг пацієнтів: автоматизований збір даних і генерація сповіщень, але відповідь залежить від інтерпретації людини. Віддалений моніторинг пацієнтів автоматизує збір даних, але вимагає участі людини для прийняття рішень.

– Системи сортування пацієнтів: ці системи автоматизують початкову оцінку потреб пацієнтів, але потребують людського контролю для точного визначення пріоритетів. Системи сортування автоматизують оцінку пацієнтів, але покладаються на людське судження для встановлення пріоритетів.

– Моделі розподілу ресурсів: це передбачає автоматизований аналіз розподілу ресурсів, але адміністратори можуть суттєво впливати на прийняття рішень. Моделі розподілу ресурсів використовують автоматизацію для аналізу розподілу ресурсів, але залучають адміністраторів до прийняття рішень.

### **Освіта**

– Персоналізовані навчальні платформи: високоавтоматизовані платформи, які адаптують зміст курсу та рівень складності на основі

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 14   |

індивідуальних результатів учня та стилю навчання. Ці платформи автоматизують налаштування змісту курсу для кожного студента.

– Автоматизовані системи оцінювання: програмне забезпечення, яке автоматично оцінює завдання, тести та тести, надаючи миттєвий зворотний зв'язок студентам. Автоматизовані системи оцінювання автоматизують оцінювання роботи студентів і забезпечують миттєвий зворотний зв'язок.

– Онлайн-нагляд і виявлення списків: використовує AI для моніторингу студентів під час онлайн-іспитів, автоматично виявляючи та позначаючи підозрілу поведінку. Онлайн-нагляд і виявлення шахрайства автоматизують моніторинг онлайн-іспитів.

– Аналітика навчання: автоматизовані інструменти аналізу даних, які відстежують прогрес і залученість студентів, щоб виявити студентів із груп ризику та рекомендувати заходи. Аналітика навчання використовує автоматизацію, щоб аналізувати дані про студентів і рекомендувати заходи.

– Системи рекомендацій для освітніх ресурсів: платформи, які використовують алгоритми, щоб пропонувати навчальні матеріали та ресурси на основі інтересів і успішності студента. Системи рекомендацій автоматизують пропозицію освітніх ресурсів.

– Програми для вивчення мови: програми, які пропонують автоматизовані уроки мови та практичні вправи з відгуками про вимову та словниковий запас. Програми для вивчення мови автоматизують уроки мови та забезпечують зворотний зв'язок.

– Віртуальні лабораторії та симуляції: інтерактивні комп'ютерні симуляції, які дозволяють студентам проводити експерименти та вивчати наукові концепції у віртуальному середовищі. Віртуальні лабораторії та моделювання автоматизують наукове навчання у віртуальному середовищі.

– Адаптивні системи репетиторства: автоматизовані репетитори, які надають допомогу студентам у режимі реального часу, адаптуючись до їхнього індивідуального темпу навчання та потреб. Адаптивні системи навчання

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 15   |

автоматизують персоналізовану допомогу студентам.

– Автоматизоване відстеження відвідуваності: використовує біометричні дані, RFID або інші технології для автоматичного обліку відвідування студентами занять і лекцій. Автоматизований контроль відвідуваності автоматизує облік відвідуваності студентів.

– Автоматизоване планування курсу: системи, які використовують алгоритми для створення оптимального розкладу занять на основі вподобань студентів і обмежень ресурсів. Автоматизоване планування курсів використовує алгоритми для створення розкладу занять.

### **Виробництво та ланцюг постачання**

– Системи контролю якості: високоавтоматизовані, часто використовують камери та датчики для виявлення дефектів без участі людини. Системи контролю якості автоматично виявляють дефекти продукції за допомогою датчиків і камер.

– Прогнозне технічне обслуговування: автоматизовано для прогнозування несправностей обладнання за допомогою даних датчиків, але може включати рішення людини для фактичних робіт з технічного обслуговування. Прогнозне обслуговування використовує автоматизацію для прогнозування несправностей обладнання на основі даних.

– Оптимізація ланцюга постачань: завдяки автоматизованому аналізу та оптимізації логістики ланцюга поставок стратегічні рішення можуть потребувати людського контролю. Оптимізація ланцюга постачання використовує автоматизацію для аналізу та покращення логістики.

– Системи обробки замовлень: значною мірою автоматизовані в обробці замовлень, але можуть передбачати втручання людини в складних випадках. Системи обробки замовлень автоматизують обробку замовлень, але залучення людини до складних ситуацій.

– Автоматизоване обслуговування клієнтів: включає чат-ботів і автоматизовані телефонні системи; високоавтоматизований, але часто передає складні запити людям. Чат-боти в онлайн-магазинах забезпечують миттєве

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 16   |

обслуговування клієнтів, скорочуючи час очікування та покращуючи задоволеність клієнтів завдяки ефективній обробці поширених запитів.

– Система прийняття рішень про повернення та відшкодування: автоматизоване прийняття рішень під час обробки повернень і відшкодувань, але у виняткових випадках може знадобитися перевірка вручну. Системи прийняття рішень щодо повернення та відшкодування автоматизують обробку, але передбачають перевірку вручну для винятків.

### **Управління майном**

– Оптимізація енергоспоживання в будівлях: цей приклад включає високоавтоматизовані системи, які керують використанням енергії в режимі реального часу шляхом аналізу заповнюваності та моделей використання, що веде до ефективного використання енергії. Ці системи автоматизують оптимізацію енергоспоживання в будівлях.

– Моделі оцінки майна: ці моделі є автоматизованими інструментами, які використовуються для оцінки вартості майна за допомогою ринкових даних і алгоритмів, що спрощує процес оцінки майна. Моделі оцінки власності автоматизують оцінку вартості власності.

– Калькулятори орендної плати: ці калькулятори автоматизують процес розрахунку орендної плати на основі вхідних даних, хоча їх точність залежить від цифр, наданих користувачем. Калькулятори орендної плати автоматизують обчислення орендної плати.

– Аналіз ринкових тенденцій: цей приклад передбачає автоматизацію аналізу даних для моніторингу ринкових тенденцій; однак інтерпретація цих тенденцій часто вимагає людського досвіду, щоб отримати значущі ідеї. Аналіз ринкових тенденцій автоматизує аналіз даних для моніторингу ринкових тенденцій.

### **Міський та екологічний менеджмент**

– Прогнозування погоди: високоавтоматизоване, використання складних моделей для прогнозування погодних умов.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 17   |

- Системи управління дорожнім рухом: Автоматизоване керування транспортними потоками та сигналами на основі даних у реальному часі.
- Управління енергією: це включає в себе автоматизовані системи для оптимізації використання енергії, але може потребувати людського контролю для прийняття стратегічних рішень.
- Автономні транспортні засоби: працюють незалежно, але рівень їхньої автономності залежить від моделі та середовища.
- Оптимізація сільського господарства: автоматизоване прийняття рішень у сільськогосподарській практиці потребує управління людьми.
- Прогнозування дорожнього руху та навігація: автоматизовано генерує прогнози та маршрути, але залежить від вибору користувача.
- Планування та управління польотами: це передбачає автоматизацію, але вимагає прийняття рішень людиною під час планування та вирішення конфліктів.
- Прогнозна поліцейська діяльність: для передбачення використовується аналіз даних, але оперативні рішення приймають співробітники.
- Прогнозування та реагування на стихійні лиха: автоматизоване прогнозування, але орієнтоване на людину планування реагування на надзвичайні ситуації.

### **Безпека**

- Виявлення загроз кібербезпеці: високоавтоматизоване безперервне спостереження та аналіз мережевих даних для виявлення загроз.
- Аналіз відеоспостереження: автоматизований у розпізнаванні закономірностей і аномалій, але може вимагати перевірки людиною для складної інтерпретації.

### **Причини для автоматизації**

Автоматизація революціонує процеси прийняття рішень з кількох причин. Насамперед, це підвищує ефективність шляхом прискорення часу прийняття рішень і зменшення помилок, які виникають вручну. Це призводить до значної економії коштів. Крім того, автоматизація дозволяє обробляти складні

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 18   |



прийняття рішень щодо повернення та відшкодування, системи сортування пацієнтів

– Стратегічна спрямованість: Автоматичні платежі за рахунками, засоби відстеження сну, автоматизовані інструменти для планування відпустки, агрегатори новин, інструменти перевірки резюме

### **Рівень автоматизації**

Рівень автоматизації варіюється залежно від прикладу, від повної до меншої автоматизації. Рішення в повністю автоматизованих системах приймаються без втручання людини. Високоавтоматизовані системи навчаються на основі процедур, але можуть вимагати початкових налаштувань, параметрів або ручних виправлень. Помірно автоматизовані приклади вимагають взаємодії з користувачем. Менш автоматизовані системи вимагають рішень або дій користувача; їх також називають системами підтримки прийняття рішень.

Рівень автоматизації:

– Повністю: Автоматизована торгівля, системи виявлення шахрайства, діагностика охорони здоров'я, системи управління трафіком, виявлення кіберзагроз

– Високо: Розумні домашні пристрої, автоматизовані банківські послуги, фільтрація спаму електронною поштою, навігаційні програми, голосові помічники

– Помірно: Персоналізовані рекомендації, фітнес-трекери та програми для здоров'я, рекомендації щодо електронної комерції, стрічки соціальних мереж, агрегатори новин

– Менше: Фотографія зі смартфона, інструменти автоматизованого планування відпустки, кулінарні пропозиції та рецепти, аналіз юридичних документів, перевірка кандидатів на роботу

### **Symbіob і приклади**

Приклади САПДЛ показують, як він використовується в різних сферах для покращення прийняття рішень. Ми зосереджуємось на САПДЛ загалом, а не

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 20   |

на конкретній програмі. Зараз ми виконуємо багато доручень у сферах HRM, маркетингу та державної політики.

### **Висновок**

Цей підсумок автоматизованого прийняття рішень у різних секторах показує, що ця технологія є не лише інструментом підвищення ефективності, але й каталізатором інновацій. 70 прикладів, які ми обговорили, ілюструють глибокий вплив автоматизації на покращення процесу прийняття рішень, оптимізацію операцій і прокладання шляху для майбутніх досягнень. Оскільки технологія продовжує розвиватися, можливості безмежні. Ми стоїмо на порозі нової ери, коли автоматизація не тільки підтримує, але й надихає людей на творчість і прогрес.

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

Як мова програмування обрана Python. Python – високорівнева мова програмування загального призначення з акцентом на продуктивність розроблювача й читаність коду. Синтаксис ядра Python мінімалістичний. У той же час стандартна бібліотека включає великий обсяг корисних функцій.

Python підтримує кілька парадигм програмування, у тому числі структурне, об'єктно-орієнтоване, функціональне, імперативне й аспектно-орієнтоване. Основні архітектурні риси – динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточні обчислень і зручні високорівневі структури даних. Код у Python організовується у функції й класи, які можуть поєднуватися в модулі (які у свою чергу можуть бути об'єднані в пакети).

Еталонною реалізацією Python є інтерпретатор CPython, що підтримує більшість активно використовуваних платформ. Він поширюється вільно під дуже ліберальною ліцензією, що дозволяє використовувати його без обмежень у

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 21   |

будь-яких застосунках, включаючи пропрієтарні. Є реалізації інтерпретаторів для JVM (з можливістю компіляції), MSIL (з можливістю компіляції), LLVM і інших. Проект PyPy пропонує реалізацію Python на самому Python, що зменшує витрати на зміни мови й постановку експериментів над новими можливостями.

Python – мова програмування, що активно розвивається, нові версії (з додаванням/зміною мовних властивостей) виходять приблизно раз у два з половиною року. Внаслідок цього й деяких інших причин на Python відсутні ANSI, ISO або інші офіційні стандарти, їхню роль виконує CPython.

Python портований і працює майже на всіх відомих платформах – від КПК до мейнфреймів. Існують порти під Microsoft Windows, практично всі варіанти UNIX (включаючи FreeBSD і Linux), Plan 9, Mac OS і Mac OS X, iPhone OS 2.0 і вище, Palm OS, OS/2, Amiga, AS/400 і навіть OS/390, Symbian і Android.

При цьому, на відміну від багатьох портуємих систем, для всіх основних платформ Python має підтримку характерних для даної платформи технологій (наприклад, Microsoft COM/DCOM). Більше того, існує спеціальна версія Python для віртуальної машини Java – Jython, що дозволяє інтерпретаторові виконуватися на будь-якій системі, що підтримує Java, при цьому класи Java можуть безпосередньо використовуватися з Python й навіть бути написаними на Python. Також кілька проектів забезпечують інтеграцію із платформою Microsoft .NET, основні з яких – IronPython і Python.Net.

Python підтримує динамічну типізацію, тобто тип змінної визначається тільки під час виконання. Тому замість «присвоювання значення змінної» краще говорити про «зв'язування значення з деяким ім'ям». У Python є убудовані типи: бульові, рядки, Unicode-рядки, цілі числа довільної точності, числа із плаваючою комою, комплексні числа й деякі інші. З колекцій Python підтримує кортежі (*tuples*), списки, словники (асоціативні масиви) і, починаючи з версії 2.4, безлічі. Всі значення в Python є об'єктами, у тому числі функції, методи, модулі, класи.

Додати новий тип можна або написавши клас (*class*), або визначивши новий тип у модулі розширення (наприклад, написаному мовою C). Система

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 22   |

класів підтримує спадкування (одиначне й множинне) і метапрограмування. Можливе спадкування від більшості убудованих типів і типів розширень.

Всі об'єкти діляться на посилальні й атомарні. До атомарного ставляться `int`, `long`, `complex` і деякі інші. При присвоюванні атомарних об'єктів копіюється їхнє значення, у той час як для посилальних копіюється тільки покажчик на об'єкт, таким чином, обидві змінні після присвоювання використовують те саме значення. Посилальні об'єкти бувають змінювані й незмінні. Наприклад, рядки й кортежі є незмінними, а списки, словники й багато інших об'єктів – змінюваними. Кортеж у Python є, по суті, незмінним списком. У багатьох випадках кортежі працюють швидше списків, тому якщо ви не плануєте змінювати послідовність, то краще використовувати саме їх.

Мова має чіткий і послідовний синтаксис, продуману модульність й масштабованість, завдяки чому вихідний код написаних на Python програм легко читаємий.

Python – стабільна й розповсюджена мова. Він використовується в багатьох проектах і в різних якостях: як основна мова програмування або для створення розширень і інтеграції застосунків. На Python реалізоване велика кількість проектів, також він активно використовується для створення прототипів майбутніх програм. Python використовується в багатьох великих компаніях.

### 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 автоматичного прийняття рішення для імітації діяльності людини.

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

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 23   |

негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 24   |

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

Прийняття рішень людиною в циклі (САПДЛ) поєднує досвід людини з алгоритмами ШІ для покращення результатів. САПДЛ має застосування в різних областях, таких як охорона здоров'я, кібербезпека, фінанси тощо.

Оскільки підприємства продовжують використовувати штучний інтелект (ШІ) для автоматизації процесів прийняття рішень, все більш важливо враховувати роль людського впливу в цих системах.

Прийняття рішень людиною в циклі (САПДЛ) безпосередньо включає людський внесок і керівництво в системах прийняття рішень ШІ для підвищення їх точності, прозорості та етичних основ.

Хоча проекти САПДЛ можна знайти як у процесі прийняття рішень за допомогою штучного інтелекту, так і в машинному навчанні, ця стаття зосереджується на першому застосуванні – досліджуючи, що таке САПДЛ, як він працює та його переваги перед іншими системами на основі штучного інтелекту без САПДЛ.

#### **Що таке прийняття рішень людиною в циклі (САПДЛ)?**

Прийняття рішень людиною в циклі передбачає роботу людини, яка приймає рішення, у тандемі з алгоритмами ШІ для покращення результатів прийняття рішень. У системах САПДЛ система штучного інтелекту надає рекомендації або прогнози, які особа, яка приймає рішення, оцінює та схвалює або відхиляє.

Прийняття рішень САПДЛ використовується в широкому діапазоні застосувань – як обговорюватиметься нижче, від діагностики охорони здоров'я до програмного забезпечення для прийняття рішень.

#### **Як працює людина в петлі?**

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-122.24.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 25   |

Системи «людина в циклі» зазвичай включають алгоритм, навчений розпізнавати закономірності в даних і отримувати результат, а також людину, яка приймає рішення, яка працює разом із системою ШІ (на відміну від повністю автоматизованої системи).

Спочатку штучному інтелекту надається вхідна інформація, наприклад дані або текстова підказка, і він приймає попереднє рішення. Потім особа, яка приймає рішення, може виправити або скоригувати рекомендації штучного інтелекту на основі свого досвіду, знань і розуміння контексту.

Цей змішаний підхід до прийняття рішень забезпечує більшу ефективність у процесі прийняття рішень, ніж це дозволило б повністю прийняте рішення людиною, одночасно досягаючи вищої точності та більш бажаних результатів, ніж повністю автоматизований процес прийняття рішень.

### **Переваги систем САПДЛ**

Система САПДЛ має кілька важливих переваг перед іншими підходами до прийняття рішень ШІ.

По-перше, оскільки результати штучного інтелекту іноді можуть бути неточними або безглуздими, дотримання підходу САПДЛ може підвищити точність шляхом включення в процес людського судження.

По-друге, людська участь підвищує прозорість щодо того, як приймаються рішення, і підвищує довіру до процесу прийняття рішень.

По-третє, САПДЛ може забезпечити врахування етичних і моральних міркувань під час прийняття рішень шляхом врахування людського судження.

Загалом система штучного інтелекту дає кращі рекомендації, оскільки вона вчиться на відгуках людей, які вона отримує з часом, і людина, яка приймає рішення, стає більш ефективною, коли вона навчається працювати з системою штучного інтелекту.

### **Приклади систем прийняття рішень штучним інтелектом за допомогою людини в циклі**

Системи людини в циклі (САПДЛ) використовуються для прийняття

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 26   |

рішень у різних областях і програмах. Ось п'ять випадків використання систем САПДЛ.

### **Медична діагностика**

Системи САПДЛ допомагають у медичній діагностиці, аналізуючи дані пацієнтів, такі як медичні записи, результати тестів і сканування зображень. Автоматизовані системи можуть надавати попередні діагнози або рекомендації, які потім перевіряються та затверджуються медичними працівниками перед прийняттям остаточного рішення щодо лікування.

### **Кібербезпека**

Системи САПДЛ відіграють вирішальну роль у кібербезпеці, відстежуючи мережевий трафік, виявляючи аномалії та потенційні порушення безпеки. Автоматизовані системи генерують сповіщення або попередження, які переглядаються аналітиками, які розслідують і реагують на інциденти безпеки.

### **Оцінка ризиків і виявлення шахрайства**

Системи САПДЛ використовуються у фінансових установах для оцінки ризиків, виявлення шахрайства та прийняття рішень щодо транзакцій. Автоматизовані системи аналізують великі обсяги даних і позначають підозрілі транзакції, які потім переглядаються експертами для остаточного затвердження або подальшого розслідування.

### **Підтримка клієнтів**

Системи САПДЛ використовуються в налаштуваннях підтримки клієнтів, таких як чат-боти або віртуальні помічники. Автоматизовані системи обробляють звичайні запити або надають початкові відповіді, а коли виникають складні або спеціалізовані запити, люди беруть на себе роботу, щоб забезпечити точну та персоналізовану допомогу.

### **Програмне забезпечення для прийняття рішень**

Програмне забезпечення для прийняття рішень і дослідження вподобань використовує систему САПДЛ для допомоги в створенні моделі. Помічник ШІ пропонує критерії та приклади альтернатив на основі підказки

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 27   |

користувача: наприклад, «еспресо-машини для офісу». Користувач переглядає та змінює пропозиції штучного інтелекту, а потім відповідає на кілька простих запитань, щоб виявити та застосувати свої переваги, в результаті чого альтернативи ранжуються.

### **Прискоріть процес прийняття рішень сьогодні**

Прийняття рішень САПДЛ може надати підприємствам потужний інструмент для більшої точності, прозорості та більш етичних рішень.

Якщо ви зацікавлені у використанні системи САПДЛ AI, подумайте про підхід до прийняття рішень САПДЛ, який поєднує людський досвід із алгоритмами AI для прийняття більш ефективних та обґрунтованих рішень.

### **3.2 Розробка структурної схеми**

Зі збільшенням використання інструментів автоматизації в державному секторі політики, урядовці та адміністратори повинні розуміти, як автоматизація впливає на контекст прийняття рішень. Незважаючи на те, що нагляд з боку людини зазвичай сприяють як запобіжний захід, експерти застерігають від хибного відчуття безпеки, яке обіцяє нагляд з боку людини, оскільки ризик, який представляє САПДЛ, лежить поза межами розсуду працівників на першому місці.

На даний момент регулювання відображає поверхневе розуміння взаємодії людини і машини. Тому, щоб ефективно мінімізувати шкоду від упередженості та дискримінації під час прийняття рішень, будь то з боку людей чи алгоритмів, політики повинні спочатку зрозуміти ризики та складності, пов'язані з використанням САПДЛ, і те, як людський контроль може відігравати значущу роль.

Ця аналітична записка має на меті поінформувати аудиторію про складнощі, що стоять за людським наглядом, у його визначенні, регулюванні та практиці. З огляду на те, що існує багато дискусій щодо того, як має

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 28   |



автоматизованого інструменту. Хоча, з іншого боку, також слід враховувати продуктивність самої системи, яка може варіюватися від її прозорості, ефективності як інструменту тощо (Ananny and Crawford 2018; Kemper and Kolkman 2019; Zhang et al. 2020; Lee і див. 2004).

З точки зору регулювання, значущим недоглядом є випадки, коли оператори здійснюють свої повноваження, усвідомлюючи упередженість або обмеження системи (і свої власні). Це означало б, що люди-оператори здатні запобігти шкоді, якщо вони зможуть зрозуміти, коли алгоритм помиляється, зрозуміти, чому алгоритм прийняв рішення та врахувати потенційні упередження системи. Тому, теоретично, для того, щоб нагляд з боку людини був ефективним, проект системи також повинен враховувати обмеження та упередження людей-операторів. На рисунку 3.1 представлена структурна схема системи у вигляді узагальненої структури й компонентів інтелектуальної системи автоматизованого ухвалення рішення для імітації діяльності людини, а також її оточення.



Рисунок 3.1 – Структурна схема системи

Алгоритми увійшли в наше життя вже досить давно, і недавній швидкий розвиток Generative Artificial Intelligence (GenAI) і великих мовних моделей викликав ентузіазм і великі очікування, а також занепокоєння. Однією з важливих сфер застосування штучного інтелекту (AI) і машинного навчання (ML) є алгоритмічна підтримка прийняття рішень (САПДЛ), призначена для підтримки осіб, які приймають рішення, або навіть для заміни їх у деяких процесах. Організації можуть розгорнути ці системи, як і попередні типи підтримки прийняття рішень, для підвищення ефективності прийняття рішень (Clark Jr., Jones, & Armstrong, 2007). Їх розробка та інтеграція в діяльність організації вимагають значних ресурсів, що є виправданим, лише якщо організація очікує явних переваг від використання системи. Використання алгоритмів для підтримки рішень може означати, що людські упередження, помилки та обмеження перестають впливати на рішення, якщо САПДЛ є частиною процесу. Проте ця стаття спрямована на те, щоб Meyer продемонструвати, що розробка та розгортання САПДЛ передбачає прийняття людських рішень у багатьох моментах з усіма його примхами. Спільнота людей, які приймають рішення, відіграє важливу роль у визначенні та вивченні властивостей людських рішень, які мають відношення до САПДЛ, і в розробці інструментів для обмеження можливих несприятливих наслідків. У документі буде коротко описано використання САПДЛ та уявлення про його об'єктивність. Потім обговорюватиметься САПДЛ як діяльність людини та організаційний процес. Потім описуються три типи рішень, пов'язаних з САПДЛ: рішення про те, чи розробляти САПДЛ для підтримки процесу, рішення, прийняті під час розробки та розгортання САПДЛ, і рішення щодо результатів САПДЛ. Зроблено деякі висновки щодо важливого внеску дослідження поведінкових рішень в успішне використання САПДЛ.

### **Деякі досягнення та застереження САПДЛ**

САПДЛ вивчали в різних контекстах. Наприклад, у правовому контексті алгоритми можуть підтримувати рішення про звільнення під заставу з можливим

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 31   |

значним зниженням рівня злочинності або ув'язнення (Kleinberg, Lakkaraju, Leskovec, Ludwig, & Mullainathan, 2018). Так само САПДЛ може допомогти прийняти рішення щодо добробуту дітей (Saxena, Badillo-Urquiola, Wisniewski, & Guha, 2020). Спостерігається тенденція до «персоналізованої доказової медицини», яка базується на аналізі електронних медичних записів за допомогою алгоритмічних інструментів (Kent, Steyerberg, & van Klaveren, 2018). Аналіз на основі штучного інтелекту може сприяти прийняттю кращих медичних діагностичних або терапевтичних рішень (Puaschunder, Mantl, & Plank, 2020), таких як рання діагностика раку молочної залози (McKinney et al., 2020). Навіть якщо штучний інтелект не покращує діагностичні рішення, він може знизити робоче навантаження клініциста, який аналізує, наприклад, мамограф (Lång et al., 2023). У деяких сферах, наприклад на фінансових ринках, алгоритмічні консультанти надають очевидні переваги (Tao, Su, Xiao, Dai, & Khalid, 2021), і вони є основою для інвестиційних рішень у високочастотній алгоритмічній торгівлі (Virgilio, 2019). Прийняття рішень людьми у багатьох випадках відрізняється від приписів нормативної теорії прийняття рішень (Такемура, 2021). Навпаки, алгоритмічну підтримку прийняття рішень можна розглядати як математичний, систематичний спосіб Мейєра для прийняття рішень, що усуває людські помилки. Таким чином, він продовжує традицію пошуку методів створення об'єктивних, раціональних процесів прийняття рішень, які ведуть до кращих рішень. Уже в 1960-х роках було багато оптимізму щодо потенціалу використання дослідження операцій і науки управління для прийняття рішень науковими методами, можливо, замінивши людей, які приймають рішення, та їхні обмеження (Dando & Bennett, 1981). Криза 1980-х років послідувала за цим ентузіазмом, коли практики виявляли зменшення інтересу до математичних методів (Ackoff, 1987; Corbett & Van Wassenhove, 1993). Подібним чином на початку розвитку науки про дані існував оптимізм, що наука про дані може замінити наукові теорії (Anderson, 2008), а великі дані вважалися такими, що, можливо, призведуть до «революції в управлінні» (McAfee & Brynjolfsson, 2012).

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 32   |

Тим часом цей оптимізм дещо послабшав, частково через усвідомлення того, що разом із потенційними перевагами САПДЛ існують також різні етичні, правові та соціальні проблеми (Čartolovni, Tomičić, & Lazić Mosler, 2022), частково пов'язані з до можливих упереджень через вибір даних або алгоритмічних процесів (Beil, Proft, van Heerden, Sviri та van Heerden, 2019). Крім того, запровадження САПДЛ може негативно вплинути на систему та створити такі явища, як «миттєві збої» або подібні аномалії на фінансових ринках (Min & Borch, 2022). Слід чітко уникати «зарозумілості великих даних», переконання, що великі дані можуть замінити традиційні аналітичні методи (Lazer, Kennedy, King, & Vespignani, 2014). Проте, незважаючи на зростаюче усвідомлення можливих обмежень, САПДЛ на основі ШІ розробляються для промисловості та бізнесу (Gupta, Modgil, Bhattacharyya, & Bose, 2021), медицини (Antoniadi та ін., 2021), державного сектора (Kuziemski & Misuraca), 2020 та інші сфери.

### **САПДЛ як людська діяльність**

САПДЛ може здатися механічним процесом, у якому дані подаються в машину (алгоритм), що забезпечує обов'язково правильний результат, якому можна довіряти, оскільки він базується на математиці. Однак більш уважний погляд на САПДЛ показує, що для цього потрібна послідовність тісно переплетених людських дій, які пов'язані з трьома аспектами процесу – рішенням про те, чи використовувати штучний інтелект для прийняття рішення Мейєра, процесом генерування результатів ШІ та рішенням про те, як використовувати результат для певної мети (див. рис. 3.1). Три аспекти розробки та використання САПДЛ не є роздільними. Наприклад, плани щодо використання вихідних даних САПДЛ будуть частиною обговорення того, чи розробляти САПДЛ для конкретного рішення. Подібним чином, якість результату штучного інтелекту, яка може залежати від процесу, використаного для його створення, буде важливою при виборі способу його використання. Поки що нам все ще бракує розуміння цих видів діяльності та їх детермінант, хоча є усвідомлення того, що вони можуть бути піддані різним упередженням (Arnott, 2006; Richardson, 2022).

Далі я коротко обговорю три аспекти, стверджуючи, що процес, який веде до САПДЛ, є важливою темою для дослідження прийняття поведінкових рішень. Результати цього дослідження можуть стати цінним внеском для розробників САПДЛ, організацій, які його використовують, і громадськості, яка обслуговується або відчуває результати підтримуваних процесів прийняття рішень. Рисунок 1: Три аспекти розробки алгоритмічної підтримки прийняття рішень – чи варто це робити, як це робити та що з цим робити. Для яких рішень? За допомогою якого процесу? Як використовувати результати?

### **САПДЛ як організаційний процес**

Велика кількість досліджень штучного інтелекту на основі даних і ML може приховати той факт, що впровадження САПДЛ і включення його в операційну діяльність організації вимагає постійних інвестицій значних ресурсів. Розгортання САПДЛ є результатом систематичного процесу, в якому була визначена підтримка Meurer, реалізовано її технічні аспекти, вони протестовані, налаштовані параметри, і, зрештою, її було запущено в певному організаційному середовищі. Поки він використовується, хтось обслуговує його, платить за нього та може (або повинен) постійно його оцінювати, оновлюючи, коли це необхідно або коли з'являються можливості. Таким чином, розгортанню системи САПДЛ передують та супроводжується низкою організаційних рішень, які визначають специфікацію та дизайн системи, обчислення, необхідні для створення її результату, і використання результату для фактичних рішень. Весь процес залежить від різноманітних організаційних факторів, а також людських рішень і участі (Golightly, Kefalidou, & Sharples, 2018). Впровадження САПДЛ в організації – це не просто технічне питання. Натомість це може вимагати культурних змін в організації (Fontaine, McCarthy, & Saleh, 2019). Це створює серйозні проблеми для менеджменту, якому необхідно впоратися зі швидкозмінними можливостями алгоритмів, а також з вимогами та обмеженнями, які виникають, коли ШІ впроваджується в організаційні процеси (Berente, Gu, Recker, & Santhanam, 2021). Швидкий розвиток штучного інтелекту

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 34   |

та машинного навчання та реалізація їх можливого застосування для підтримки прийняття рішень призвели до значного збільшення відповідних досліджень. Наприклад, кількість публікацій із «Підтримкою прийняття рішень» у будь-якій галузі в Clarivate Web of Science подвоїлася з 24 063 у 2014 році до 48 684 у 2021 році. Однак існує порівняно небагато досліджень фактичного впровадження підтримки прийняття рішень в організаціях. В одному огляді лікарняних комп'ютеризованих систем підтримки прийняття рішень виявлено 1836 документів, які стосувалися цієї теми між 2010 і 2021 роками. Лише 9 стосувалися витрат на впровадження такої системи (Donovan, Abell, Fernando, McPhail, & Carter, 2023). Дійсно, незважаючи на велику кількість досліджень щодо САПДЛ, досліджень щодо фактичного впровадження підтримки прийняття рішень на основі машинного навчання в клінічних умовах (або інших сферах) все ще мало (Susanto, Lyell, Widyantoro, Berkovsky, & Magrabi, 2023). Подібним чином огляд економічного впливу систем підтримки прийняття рішень у лікарнях виявив, що з 3113 потенційно релевантних статей, опублікованих між 2010 і 2023 роками, лише 29 стосувалися економічних наслідків використання такої системи (White et al., Meyer 7 2023). Крім того, практиками прийнято підтримку прийняття рішень не є універсальним, і навіть якщо система була розроблена та розгорнута, вона може не використовуватися. Огляд досліджень щодо використання підтримки прийняття рішень у клінічних умовах показав середнє застосування лише в 34,2% випадків, де можна було використовувати систему (Kouri, Yamada, Lam Shin Cheung, Van de Velde та Gupta, 2022). Для більшості заходів фахівці-практики вирішили не використовувати підтримку прийняття рішень. Ці дослідження показують, що ми знаємо відносно мало про умови успішного використання САПДЛ в організаціях.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 35   |

### 3.3 Розробка функціональної схеми

#### Класифікація ПС

ПС можуть розміщатися на якому-небудь сайті, де користувач задає системі питання природною мовою (якщо це питально-відповідна система) або, відповідаючи на питання системи, знаходить необхідну інформацію (якщо це експертна система). Але, як правило, ЕС в інтернеті виконують рекламно-інформаційні функції (інтерактивні баннери), а серйозні системи (такі, як, наприклад, ЕС діагностики встаткування) використовуються локально, тому що виконують конкретні специфічні завдання.

Інтелектуальні розвідувачі відрізняються від віртуальних співрозмовників тим, що вони досить безликі й у відповідь на питання видають деяку вижимку із джерел знань (іноді досить великого обсягу), а співрозмовники володіють «характером», особливою манерою спілкування (можуть використовувати сленг, ненормативну лексику), і їхні відповіді повинні бути гранично лаконічними (іноді навіть просто у формі смайликів, якщо це відповідає контексту).

ПС:

- Експертні системи.
- Властиво експертні системи (ЕС).
- Інтерактивні баннери (web + ЕС).
- Питально-відповідні системи (у деяких джерелах «системи спілкування»).
- Інтелектуальні розвідувачі (наприклад, система Старт).
- Віртуальні співрозмовники.
- Віртуальні цифрові помічники.
- Експертні системи.

**Експертна система (ЕС, expert system)** – комп'ютерна програма, здатна частково замінити фахівця-експерта в дозволі проблемної ситуації. Це обчислювальна система, у яку включені знання фахівців про деяку вузьку

предметну область у формі бази знань. Такі системи можуть використовуватися не експертом для поліпшення їхніх здатностей і можливостей у рішенні завдань певного класу в конкретній предметній області. ЕС можуть бути використані для поширення джерел рідких знань. Ці системи можуть мати значний вплив як на діяльність таких професійних консультантів, як фінансові аналітики, юристи, аудиторы й ін., так і на організації і їхній менеджмент.

У середині експертної системи немає заздалегідь заданого дерева питань, кожне наступне питання вибирається виходячи з відповідей на всі попередні. Це дозволяє виключити зайві питання й не видавати варіанти відповіді, які не приведуть до яких-небудь результатів. Відсутність фіксованого дерева дозволяє користувачеві задавати пріоритет питань, вибираючи найбільш важливі для себе аспекти в процесі пошуку. У будь-який момент можна знову повернутися до питання й вибрати інша відповідь без необхідності знову відповідати на інші питання.

Експертні системи мають одна велика **відмінність від інших ПС**: вони не призначені для рішення якихось універсальних завдань, як наприклад нейронні мережі або генетичні алгоритми. Експертні системи призначені для якісного рішення завдань у певної розроблювачами області, у рідких випадках – областях.

#### **Характерними рисами ЕС є:**

- чітка обмеженість предметної області;
- здатність приймати рішення в умовах невизначеності;
- здатність пояснювати хід і результат рішення зрозумілим для користувача способом;
- чіткий поділ декларативних і процедурних знань (фактів і механізмів виводу);
- здатність поповнювати базу знань, можливість нарощування системи;
- результат видається у вигляді конкретних рекомендацій для дій у сформованій ситуації, що не уступають рішенням кращих фахівців;

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 37   |

- орієнтація на рішення неформалізованих (спосіб формалізації поки невідомий) завдань;
- алгоритм рішення не описується заздалегідь, а будується самою експертною системою;
- відсутність гарантії знаходження оптимального рішення з можливістю вчитися на помилках.

#### Класифікації ЕС:

- Властиво Експертні системи.
- Інтерактивні баннери (web + ЕС).

Інтерактивні мовні баннери – це інфи або експертні системи, призначені для розміщення на зовнішніх ресурсах.

#### Переваги інтерактивних баннеров:

- Підвищена привабливість для споживачів – з незвичайним баннером хочеться поспілкуватися.
- Тривалий контакт із користувачем. Середній час спілкування з баннером може становити близько 3 минут.
- Баннер може вести різних співрозмовників на різні сторінки, відповідно до їх запитів і потребами.

#### Класифікація ЕС по зв'язку з реальним часом:

- Статичні ЕС – це ЕС, що вирішують завдання в умовах не вихідних даних, що змінюються в часі, і знань.
- Квазидинамічні ЕС інтерпретують ситуацію, що міняється з деяким фіксованим інтервалом часу.
- Динамічні ЕС – це ЕС, що вирішують завдання в умовах вихідних даних, що змінюються в часі, і знань.

#### Структура ЕС:

- механізм логічного виводу, називаний також інтерпретатором, вирішувачем;
- робочу пам'ять (РП), називану також робочою базою даних (БД);

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 38   |

- базу знань (БЗ);
- підсистему придбання й поповнення знань;
- підсистему пояснення;
- підсистему діалогу;
- підсистему взаємодії із зовнішнім миром.

**Механізм логічного виводу (МЛВ)** призначений для одержання нових фактів на основі зіставлення вихідних даних з робочої пам'яті й знань із бази знань. Механізм логічного виводу реалізує алгоритми прямого й/або зворотного виводу й формально може бути представлений четвіркою:

Механізм виводу є мозком ЕС, його також називають керуюча структура або інтерпретатор правил (в ЕС, заснованих на правилах).

Цей компонент є в основному комп'ютерною програмою, що забезпечує методологію для міркування про інформацію в БЗ і в робочій області, а також для формулювання висновків. Вона забезпечує вказівки про те, як використовувати знання системи при реалізації оренди (розкладу запланованих дій у робочій області), що організує й управляє кроками, що вживаються для рішення завдання.

Механізм виводу має два головних елементи:

- Інтерпретатор, що виконує обрані позиції оренди, використовуючи відповідні правила БЗ.
- Планувальник, що підтримує керування агендою. Він оцінює результати використовуваних правил виводу у світлі їхніх пріоритетів або інших критеріїв в агенді.

**Робоча пам'ять** призначена для зберігання вихідних і проміжних фактів розв'язуваної в сучасний момент завдання. Як правило, розміщується в оперативній пам'яті ЕОМ і відбиває поточний стан предметної області у вигляді фактів з коефіцієнтами впевненості (КВ) в істинності цих фактів.

Цінність всієї експертної системи як закінченого продукту на 90% визначається якістю створеної **бази знань**. Як правило, БЗ ЕС містить факти (статичні відомості про предметну область) і правила – набір інструкцій,

застосовуючи які до відомих фактів можна одержувати нові факти. У рамках логічної моделі баз даних і бази знань записуються мовою Пролог (мова й система логічного програмування) за допомогою мови предикатів для опису фактів і правил логічного виводу, що виражають правила визначення понять, для опису узагальнених і конкретних відомостей, а також конкретних і узагальнених запитів до баз даних і базам знань.

**Підсистема придбання й поповнення знань** автоматизує процес наповнення експертної системи знаннями, здійснюваний користувачем-експертом, і адаптації бази знань системи до умов її функціонування. Адаптація експертної системи до змін у предметній області реалізується шляхом заміни правил або фактів у базі знань.

**Підсистема пояснення** пояснює, як система одержала рішення завдання (або чому вона не одержала рішення) і які знання вона при цьому використовувала, що полегшує експертові тестування системи й підвищує довіру користувача до отриманого результату. Можливість пояснювати свої дії є одним з найважливіших властивостей експертної системи, тому що:

- підвищується довіра користувачів до отриманих результатів;
- полегшується налагодження системи;
- створюються умови для користувачів по розкриттю нових закономірностей предметної області;
- пояснення отриманих висновків може служити засобом пошуку крапки в парето-оптимальній безлічі рішень.

У цей час на практиці всі системи пояснення реалізуються на тих самих принципах в основному двома способами:

- фіксацією подій і станів за допомогою заготовлених текстів природною мовою;
- трасуванням міркувань, зворотним розгортанням дерева цілей із вказівкою підцілей.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 40   |

При реалізації кожного із цих способів попередньо виділяються ситуації, факти й вузли переходу в нові стани, що вимагають пояснень. Їм ставиться у відповідність деякий текст пояснення.

Структура експертної системи була б неповної без **підсистеми діалогу**. Підсистема діалогу орієнтована на організацію дружнього інтерфейсу з усіма категоріями користувачів як у ході рішення завдань, так і в ході придбання знань і пояснення результатів роботи.

Факти й правила в експертній системі не завжди або щирі, або помилкові. Іноді існує деякий ступінь непевності у вірогідності факту або точності правила. Якщо цей сумнів виражений явно, то воно називається «коефіцієнтом довіри».

**Коефіцієнт довіри** – це число, що означає ймовірність або ступінь упевненості, з якої можна вважати даний факт або правило достовірним або справедливим. Даний коефіцієнт є оцінкою ступеня довіри до рішення, видаваному експертною системою. Така оцінка, наприклад, може проводитися за схемою Шортлиффа.

Режими функціонування ЕС:

– Режим уведення знань – у цьому режимі експерт за допомогою інженера по знаннях за допомогою редактора бази знань вводить відомі йому відомості про предметну область у базу знань ЕС.

– Режим консультації – користувач веде діалог з ЕС, повідомляючи її відомості про поточне завдання й одержуючи рекомендації ЕС. Наприклад, на основі відомостей про фізичний стан хворого ЕС ставить діагноз у вигляді переліку захворювань, найбільш імовірних при даних симптомах.

Системи інтерпретації виявляють описи ситуації зі спостережень. Це категорія включає спостереження, розуміння мови, аналіз образів, інтерпретацію сигналів і багато інших видів інтелектуального аналізу. Система інтерпретації пояснюють спостережувані дані шляхом присвоєння їм символічних значень, що описують ситуацію.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 41   |

Системи пророкування включають прогнозування погоди, демографічні пророкування, економічне прогнозування, оцінки врожайності, а також військове, маркетингове й фінансове прогнозування.

Системи діагностики включають діагностику в медицині, електроніці, механіку й програмному забезпеченні. Діагностуючі системи звичайно співвідносять спостережувані поведінкові відхилення із причинами, що лежать в основі.

Системи проектування розробляють конфігурації об'єктів, які задовольняють певним вимогам завдання проектування. Такі завдання включають конструювання будинків, планування розташування встаткування й ін. Ці системи конструюють різні взаємозв'язки описів об'єктів один з одним і перевіряють, чи задовольняють ці конфігурації встановленим обмеженням і вимогам.

Системи планування спеціалізуються на завданнях планування, наприклад, такий як автоматичне програмування. Вони також працюють із коротко й довгостроковим плануванням у керуванні проектами, маршрутизація, комунікація, розробка продукту а, військові додатки, виробниче й фінансове планування.

Системи моніторингу порівнюють спостереження поведінки системи зі стандартами, які представляються визначальними для досягнення мети. Ці вирішальні виявлення відповідають потенційним недолікам на підприємстві. Існує багато комп'ютерних систем моніторингу: від контролю руху повітряних потоків до завдань керування збором податків.

Системи керування й контролю адаптивно управляють загальним поведінням системи. Для здійснення цього система керування повинна періодично інтерпретувати поточну ситуацію, пророкувати майбутнє, діагностувати причини очікуваних проблем, формулювати план усунення цих проблем і здійснювати моніторинг його виконання для забезпечення успіху.

Найбільш відомі/розповсюджені ЕС:

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 42   |

– CLIPS – популярна ЕС (public domain).

– OpenCyc – потужна динамічна ЕС із глобальною онтологічною моделлю й підтримкою незалежних контекстів.

– MYCIN – найбільш відома діагностична система, що призначена для діагностики й спостереження за станом хворого при менінгіті й бактеріальних інфекціях.

– HASP/SIAP – інтерпретуюча система, що визначає місце розташування й типи судів у Тихому океані по даним акустичних систем спостереження.

Першу ЕС за назвою Dendral розробили в Стенфорді наприкінці 1960-х рр. Вона визначала будову органічних молекул по хімічних формулах і спектрографічних даних про хімічні зв'язки в молекулах. Цінність Dendral полягала в наступному: органічні молекули, як правило, дуже великі й тому число можливих структур цих молекул також велико; завдяки евристичним знанням експертів-хіміків, закладених в ЕС, правильне рішення з мільйона можливих перебувало всього за кілька спроб. Принципи й ідеї, закладені в Dendral виявилися настільки ефективними, що вони дотепер застосовуються в хімічних і фармацевтичних лабораторіях по усьому світі. ЕС Dendral однієї з перших використовувала евристичні знання фахівців для досягнення рівня експерта в рішенні завдань, однак методика сучасних експертних систем пов'язана з іншою розробкою – Mycin. У ній використовувалися знання експертів медицини для діагностики й лікування спеціального менінгіту й бактеріальних інфекцій крові. ЕС Mycin, також розроблена в Стенфорді в середині 1970-х рр., однієї з перших звернулася до проблеми прийняття рішень на основі ненадійної або недостатньої інформації. Всі міркування експертної системи Mycin були засновані на принципах керуючої логіки, що відповідають специфіці предметної області. Багато методик розробки експертних систем, що використовуються сьогодні, були вперше розроблені в рамках проекту Mycin. MYCIN була ранньою експертною системою розробленої за 5 або 6 років на початку 1970х років у Стендфордському університеті. Вона була написана на Лиспі як докторська

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 43   |

дисертація Edward Shortliffe під керівництвом Bruce Buchanan, Stanley N. Cohen і інших. У цій же лабораторії була раніше створена експертна система Dendral, але цього разу увага була акцентована на використанні вирішальних правил з елементами невизначеності. MYCIN був спроектований для діагностування бактерій, що викликають важкі інфекції, такі як бактеріємія й менінгіт, а також для рекомендації необхідної кількості антибіотиків залежно від маси тіла пацієнта. Назва системи походить від суфікса «-міцин», що часто зустрічається в назвах антибіотиків. Також Мусін використовувалася для діагностики захворювань згортаємості крові.

#### Переваги ЕС:

1. Сталість. Людська компетенція слабшає згодом. Перерва в діяльності людини-експерта може серйозно відбитися на його професійних якостях.

2. Легкість передачі. Передача знань від однієї людини іншій – довгий і дорогий процес. Передача штучної інформації – це простий процес копіювання програми або файлу даних.

3. Стійкість і відтворюваність результатів. Експертні системи стійкі до «перешкод». Людина ж легко піддається впливу зовнішніх факторів, які безпосередньо не пов'язані з розв'язуванням завданням. Експерт-Людина може приймати в тотожних ситуаціях різні рішення через емоційних факторів. Результати експертної системи – стабільні.

4. Вартість. Експерти, особливо висококваліфіковані обходяться дуже дорого. Експертні системи, навпаки, порівняно недорогі. Їхня розробка дорога, але вони дешеві в експлуатації. Крім того, експерт-людина може приймати різні рішення в тотожних ситуаціях через емоційних факторів (вплив дефіциту часу, вплив стресу).

#### Недоліки ЕС:

На сьогоднішній день створена вже велика кількість експертних систем. За допомогою їх вирішується широке коло завдань, але винятково в вузькоспеціалізованих предметних областях. Як правило, ці області добре

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-122.24.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 44   |

вивчені й розташовують більше менш чіткими стратегіями прийняття рішень.

Зараз розвиток експертних систем трохи призупинилося, і цьому є ряд причин:

– Передача експертним системам «глибоких» знань про предметну область є великою проблемою. Як правило, це є наслідком складності формалізації евристичних знань експертів.

– Експертні системи нездатні надати осмислені пояснення своїх міркувань, як це робить людина. Як правило, експертні системи всього лише описують послідовність кроків, початих у процесі пошуку рішення.

– Налагодження й тестування будь-якої комп'ютерної програми є досить трудомісткою справою, але перевіряти експертні системи особливо важко. Це є серйозною проблемою, оскільки експертні системи застосовуються в таких критичних областях, як керування повітряним і залізничним рухом, системами зброї й у ядерній промисловості.

– Експертні системи володіють ще одним більшим недоліком: вони нездатні до самонавчання. Для того, щоб підтримувати експертні системи в актуальному стані необхідно постійне втручання в базу знань інженерів по знаннях. Експертні системи, позбавлені підтримки з боку розроблювачів, швидко втрачають свою затребуваність.

– Експерти можуть безпосередньо сприймати комплекс вхідної сенсорної інформації (візуальної, звукової, дотикальної, нюхової й тактильної). ЕС – тільки символи. Хоча в окремих напрямках розробки інженерних і виробничих інтелектуальних систем автоматизованого ухвалення рішення для імітації діяльності людини отримані реальні результати певної обробки сенсорної інформації.

– Експерти – люди можуть охопити картину в цілому, всі аспекти проблеми й зрозуміти, як вони співвідносяться з основним завданням. ЕС прагне зосередити на самому завданні, хоча суміжні завдання можуть вплинути на рішення основної.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 45   |

– Люди, експерти й не експерти, мають те, що ми називаємо здоровим глуздом, або загальнодоступними знаннями. Це широкий спектр загальних знань про світ, про те, які закони в ньому діють, тобто знання, якими кожний з нас володіє, здобуває з досвіду і якими постійно користується. Через величезний обсяг знань, що утворюють здоровий глузд, не існує легкого способу вмонтувати їх в інтелектуальну програму. Знання здорового глузду включають знання про те, що ви знаєте й чого не знаєте.

Тому ЕС найбільше часто використовуються як порадики, як консультантів або помічників ЛПР.

### **Питально-відповідні системи**

Класифікація питально-відповідних систем:

- Інтелектуальні розвідувачі (наприклад, система Старт).
- Віртуальні цифрові помічники.
- Віртуальні співрозмовники (ВС).

Віртуальні співрозмовники встановлюються на сайт і спілкуються з його користувачами за допомогою текстового чата. У кожного інфа є свій візуальний образ, що здатний передавати емоції інфа й робить спілкування зі співрозмовником більше особистим і довірчим.

### **Структура віртуальних співрозмовників**

Перший компонент ВС – це користувальницький інтерфейс, за допомогою якого користувач розмовляє із ВС. Користувальницький інтерфейс являє собою віконце з рядком уведення тексту, репліками інфа і його візуальним образом. По суті, це Flash-Додаток, що легко й швидко встановлюється на будь-який сайт.

Другий компонент – це комплексна платформа, що визначає поведження й словниковий запас ВС. Крім іншого, у комплексну платформу входить база знань інфа – набір гнучких сценаріїв із заданими варіантами питань і відповідей на них. Додатково до бази знань може бути підключена клієнтська база даних з користувальницькою інформацією, звідки інф буде брати конкретні дані про

товари й послуги. Зокрема, це широко застосовується при розробці інфів-продавців.

### **Розв'язувані завдання**

ВС легко піддаються навчанню й допомагають вирішити безліч завдань, що коштують перед замовником. Вони можуть бути:

- консультантами, що відповідають на питання користувачів про представлені товари й послуги;
- продавцями, що допомагають підібрати потрібний товар, послугу, тариф і т.п.;
- співробітниками технічної підтримки, що допомагають користувачеві вирішити виниклі технічні проблеми;
- промоутерами, що просувають нові товари й послуги;
- цікавими співрозмовниками, що викликають інтерес, що підвищують настрої і лояльність відвідувачів.

### **Сфери застосування:**

- Банки й страхові компанії, яким важливо мати на сайті грамотного консультанта, здатного оперативно розповісти всі подробиці про надавані послуги;
- Інтернет-Магазини, яким важливо допомагати клієнтам у виборі товарів, а також просувати акції й розпродажі;
- Інтернет-Портали, яким необхідно привертати увагу користувачів до їхніх внутрішніх проектів;
- Організатори заходів, яким важливо інформувати відвідувачів сайту про всі новини й подробиці;
- Компанії, що роблять технічні послуги, яким важливо забезпечити цілодобову технічну підтримку користувачів.

### **Переваги віртуальних співрозмовників:**

- Працездатність: інф працює 24 години на добу 7 днів у тиждень і може одночасно спілкуватися з необмеженою кількістю користувачів. Інф дозволяє

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 47   |

знизити навантаження й витрати на call-центр, консультантів і фахівців технічної підтримки.

– Доступність: інф знімає психологічний бар'єр, що коштує перед користувачем при обігу за допомогою; досить увести фразу – і інф моментально дасть грамотну пораду. При цьому користувачі ставляться до інфа з довірою, оскільки він уміє підтримувати живий, невимушений діалог і навіть виражати емоції у відповідь на репліки користувача.

– Простота роботи: інф не жадає від користувача використання ніяких додаткових програм. У той же час інф не створює проблем і в замовника: для установки інфа на сайт досить розмістити на сторінках спеціальний короткий код.

– Компетентність: інф легко піддається навчанню, що дозволяє закласти в нього всі важливі питання, які цікавлять користувачів. Інф здатний допомагати користувачеві в навігації по сайті, автоматично відкриваючи необхідні сторінки. При необхідності інф може сам ініціювати діалоги на потрібні теми.

– Уважність: Інф записує всі розмови з користувачами, і замовник має до них повний доступ. Запису розмов корисні як з погляду подальшого навчання інфа, так і з погляду збору коштовної інформації про користувачів і їхні інтереси.

Використання ВС дозволяє:

– Збільшити конверсію відвідувачів у клієнтів: інф знімає мотиваційний бар'єр між користувачем і сайтом, оскільки відразу викликає довіра в користувача й дає йому саме ту інформацію, що його цікавить.

– Підвищити лояльність відвідувачів: яскравий, позитивний інф підтримує живе спілкування з користувачем і викликає в нього самі позитивні емоції. Що важливо, у свідомості користувача ці емоції будуть прямо пов'язані з образом компанії – замовника інфа.

– Поліпшити ефект від рекламної кампанії й маркетингових акцій: інф залучає до себе увага користувачів і надає їм саму повну інформацію про рекламований предмет.

– Знизити навантаження на штатних консультантів, продавців і співробітників техпідтримки: відповідаючи на часто виникаючі й легко розв'язувані питання, інф заощаджує час і сили штатних фахівців, дозволяючи їм сконцентруватися на дійсно важливих проблемах.

– Підвищити рівень обслуговування клієнтів: інф дозволяє з'ясувати, що цікавить конкретного клієнта, і надати йому те, що потрібно!

### **Перспективи розвитку ІС у керуванні знаннями**

Розглядаючи тенденції розвитку Інтелектуальних інформаційних систем у керуванні знаннями, слід зазначити наступні основні напрямки, пов'язані з розробкою моделей і методів реалізації окремих аспектів одержання й перетворення знань:

**1. Технології добування й подання знань.** У першому випадку основним завданням є розробка методів: формального опису "ознак знань" (пошукових образів); формалізації Про; розпізнавання й порівняння образів; добування знань із експертів, статистики, текстів, "досвіду" і т.п. У другому – вирішуються завдання, пов'язані з формалізацією знань для їхнього подання в пам'яті інтелектуальних систем автоматизованого ухвалення рішення для імітації діяльності людини (ІС). Рішення цих завдань дозволяє розроблювачам комплексних технологій одержати відповіді на три принципово важливих питання: які знання необхідно представляти в ІС, хто (що) є джерелом цих знань, які методи й моделі забезпечують адекватне подання цих знань в ІС.

**2. Технології маніпулювання знаннями, рішення інтелектуальних завдань** припускає не тільки подання знань в ІС, але і їхню обробку, тобто необхідно навчити ІС оперувати ними. Тому тут вивчаються питання поповнення знань на основі їхніх неповних описів, класифікації знань в ІС, розробляються процедури й методи узагальнення знань, достовірного виводу й ін.

**3. Технологія спілкування.** Перехід до ІС знаменує нову технологію спілкування кінцевих користувачів з ЕОМ і вимагає рішення таких проблем, як розуміння зв'язних текстів на обмеженій і необмеженій природній мові,

розуміння мови і її синтез, розробка комунікативних моделей " користувач-ЕОМ", формування пояснень і т.п. Головна мета даних досліджень – забезпечення комфортних умов для спілкування людини й ІС.

**4. Технології сприйняття.** Розробка цих технологій припускає створення методів: аналізу тривимірних сцен, подання інформації про зорові образи в базі знань ІС, трансформації зорових сцен у текстові описи й назад, а також розробку процедур когнітивної графіки й ін.

**5. Технології навчання.** Відмінною рисою ІС повинна стати їхня здатність вирішувати завдання, у явному виді не представлені в БЗ, що вимагає наділення ІС здатністю до навчання. Для цих цілей необхідно: створити методи формування умов завдання по описі проблемної ситуації або за спостереженням за цією ситуацією, забезпечити перехід від відомого рішення приватних завдань до рішення загального завдання, наділити ІС здатністю декомпонувати вихідне завдання на більше дрібні, рішення яких відомо, розробити нормативні й декларативні моделі самого процесу навчання, створити теорію наслідувального поведіння й ін.

**6. Технології поведіння.** Взаємодія ІС із середовищем вимагає розробки спеціальних поведінкових процедур, які б дозволили їм адекватно реагувати на ті або інші зміни в середовищі. Така взаємодія припускає створення моделей доцільного, нормативного й ситуативного поведіння, а також розробку методів багаторівневого планування й корекції планів у динамічних ситуаціях.

Області застосування існуючих на сьогоднішній день систем ІС охоплює безліч сфер: медичну діагностику, інтерпретацію геологічних даних, наукові дослідження в хімії й біології, військову справу, виробництво, фінанси й інші області. Однак, незважаючи на значні успіхи в області ІС, поки ще існує певний розрив між технічними розробками, програмними засобами ІС й можливостями їх більш широко практичного застосування зокрема, в економіці.

Найбільш показовим сектором, що акумулює різні проблемні напрямки економічної області, є керування промисловим підприємством. На його прикладі особливо добре видні переваги використання систем ІС для рішення як різних

|      |      |          |        |      |                                  |           |
|------|------|----------|--------|------|----------------------------------|-----------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк.      |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | <b>50</b> |

предметних завдань, так і для керування інтегрованою системою підприємства в цілому.

Існує безліч доводів на користь того, що системи штучного інтелекту можуть і повинні стати найважливішою складовою частиною в технології сучасних виробництв. Основними з них є:

- подолання складності (складності керування виникають тоді, коли доводиться робити вибір з безлічі можливих рішень);
- керування підприємством вимагає організації більших обсягів інформації;
- як зменшити інформацію до того рівня, що необхідний для ухвалення рішення (втрата інформації, що надходить від об'єктів, що працюють у реальному режимі часу, може істотно позначитися на результаті);
- нестача часу на ухвалення рішення (проявляється в міру ускладнення виробництва);
- проблема координації (рішення необхідно координувати з іншими ланками процесу або об'єкта);
- необхідність збереження й поширення знань дуже досвідчених експертів, отриманих ними в процесі багаторічної роботи й великого практичного досвіду.

Проблема добування знань і їхнє збереження й розподіли – сьогодні одна з головних проблем організацій.

Таким чином, інтелектуалізація інформаційних систем керування й трансформація їх в інтелектуальні інформаційні системи керування знаннями, підтримки прийняття рішень є найбільш значимим і важливим для економіки й бізнесу напрямком.

Управлінське рішення (УР) – це результат аналізу, прогнозування, оптимізації, економічного обґрунтування й вибору альтернативи з безлічі варіантів для досягнення конкретної мети менеджменту. Якість УР – це ступінь відповідності параметрів обраної альтернативи рішення певній системі

|      |      |          |        |      |                                  |           |
|------|------|----------|--------|------|----------------------------------|-----------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк.      |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | <b>51</b> |

характеристик, що задовольняє його розроблювачів і споживачів і можливість, що забезпечує, ефективної реалізації. До таких характеристик відносять: наукову обґрунтованість, своєчасність, несуперечність, адаптивність, реальність. Роль СППР не у тому, щоб замінити керівника, а у тому, щоб підвищити його ефективність.

У вітчизняній літературі спостерігається термінологічна плутанина, що виникла у зв'язку зі зміною розуміння терміна "система підтримки прийняття рішень". У сучасній англійській літературі еквівалент цього поняття – Decision Support System (DSS). Раніше в період, приблизно, з початку 70-х до початку 90-х років публікувалися оригінальні й перевідні статті, у яких застосовувався інший англійський еквівалент – " Decision-Making Support System (DMSS)". Головне розходження в наступному: раніше під підтримкою прийняття рішень розумівся *інструментарій вироблення рекомендацій для особи, що приймає рішення (ОПР)*, зараз те ж поняття означає *інструментарій підготовки даних для ОПР*.

У сучасній системі автоматичного прийняття рішення для імітації діяльності людини яка у даній роботі реалізується у вигляді системи підтримки прийняття рішень (СППР) повинні бути реалізовані й перший і другий інструментарій, причому вони повинні бути взаємозалежні. Тобто інструментарій підготовки даних повинен бути реалізований з обліком того, що ці дані, можливо, будуть потім використані інструментарієм вироблення рекомендацій. Об'єднання двох інструментаріїв у рамках однієї системи дає більше зручний спосіб взаємодії користувача зі СППР. У процедурі прийняття рішень менеджер може одночасно використовувати інформацію, отриману різними шляхами: будь то звичайні або більше складні агреговані запити, отримані за допомогою OLAP-засобів, або «нові» знання, отримані в ході застосування методів і технологій ІАД.

### **OLAP (Online Analytical Processing – оперативна аналітична обробка)**

Поряд з іншими засобами аналізу даних під час бурхливого розвитку комп'ютерних засобів з'явилась технологія багатомірного аналізу даних OLAP (Online Analytical Processing – оперативна аналітична обробка). Це

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 52   |

спосіб управління і представлення даних у простий і зрозумілий для кінцевого користувача спосіб.

Призначення систем класу OLAP – забезпечити користувача гнучким інтуїтивно зрозумілим і простим доступом до даних. Наявність такого доступу дає змогу відмовитися від використання наперед визначених звітів, робить користувачів самодостатніми, незалежними від адміністраторів БД і програмістів. В основі концепції OLAP дані представлені у вигляді багатовимірного куба, причому користувач має змогу швидко згорнути або розгорнути дані забудь-яким виміром. Ця технологія не замінює, а доповнює традиційні реляційні бази даних з первинною інформацією[1].

OLAP на відміну від інших способів автоматизації бізнес-діяльності дає можливість отримати користувачу «на виході» не готове чітко структуроване рішення, що видається після включення раніше налаштованого майстра обробки форм, а своєрідний матеріал для творчої оцінки існуючої ситуації. Тому сфера застосування OLAP-аналізу звичайно обмежується менеджерським складом підприємств різних розмірів, якому доводиться часто займатися тактичними і стратегічними завданнями на зразок аналізу ключових показників діяльності та сценаріїв розвитку, маркетингового та фінансово-економічного аналізу груп товарів або послуг, а також є довгостроковим прогнозуванням роботи підприємства чи його підрозділів [5].

Всі OLAP-продукти характеризуються загальними принципами побудови[3]:

1. В якості зовнішнього інтерфейсу вони надають керовану динамічну таблицю. На вхід динамічної таблиці подається багатовимірний масив. Масив складається з даних двох типів: вимірювань і фактів. Вимірювання стають колонками і рядками динамічної таблиці. У них відображаються члени вимірювань. На перетині колонок і рядків розміщені факти.

2. Колонки та рядки є основними інструментами управління таблицею. З їх допомогою користувач може маніпулювати вихідними даними: міняти місцями

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 53   |

рядки і колонки, встановлювати фільтри з вимірювань, деталізувати інформацію або навпаки узагальнювати її. При цьому проміжні і остаточні підсумки за фактами автоматично перераховуються. Виконання цих операцій забезпечується OLAP-машиною (або машиною OLAP-обчислень). Самі маніпуляції з даними носять назву OLAP-операцій.

3. Ще однією важливою стороною OLAP-аналізу є графічне відображення даних. Графік синхронізований з динамічною таблицею. Після виконання будь-якої OLAP-операції дані перераховуються, а графік перемальовується.

### **Інтегровані системи управління підприємством**

Елементи автоматизованої підтримки прийняття рішень присутні в інтегрованих системах управління підприємством (ІСУП). Однак для промислових підприємств впровадження ІСУП є однією з найбільш трудомістких, дорогих і тривалих програм розвитку з величезним ризиком невдалого закінчення проекту впровадження. Широко відомі результати дослідження фірми Standish Group, які свідчать, що 84% ІТ-проектів кінчаються зірваними строками, перевищенням бюджету або зовсім нічим. У той же час автоматизована підтримка рішень необхідна керівникам підприємств уже зараз, і тому зроблено вивід про необхідність розробки незалежних СППР, які можуть одержувати дані з існуючих на підприємстві автоматизованих систем.

Виявимо фактори, що стримують побудову сучасних СППР, що дозволяють автоматизувати підготовку даних для прийняття управлінських рішень. Самі головні з них:

- невірогідність даних;
- низька продуктивність при нестандартних запитах;
- неможливість перетворення різнорідних даних у єдину інформацію.

У роботі ухвалене рішення, знайдене Б. Інмоном і сформульоване у вигляді концепції СЗД. По його визначенню, сховище даних – це предметно-предметно-орієнтована, інтегрована, некоректуєма, залежна від часу колекція даних, призначена для підтримки прийняття управлінських рішень.

|      |      |          |        |      |                                  |           |
|------|------|----------|--------|------|----------------------------------|-----------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк.      |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | <b>54</b> |

Розглянемо сховища даних двох типів:

- корпоративні СЗД;
- вітрини даних (ВД).

СЗД і ВД будуються по подібних принципах і використовують практично ті самі технології. ВД мають менший розмір і обслуговують деяке один напрямок або аспект діяльності підприємства.

На основі проведених досліджень зроблений вивід про те, що сучасна СППР повинна складатися з наступних компонентів:

1. Оперативні джерела даних (можуть являти собою OLTP, корпоративні БД, зовнішні джерела).
2. Засоби переносу й трансформації даних (виконують збір, очищення й узгодження даних із джерел).
3. СУБД (високошвидкісна серверна СУБД), що дозволяє підтримувати багаторівневу систему зберігання даних, що складає із СЗД і безлічі ВД.
4. Засоби доступу й аналізу даних (дозволяють одержувати деталізовані дані, агреговані показники й закономірності).
5. Допоміжні компоненти: засобу проектування/розробки й засобу адміністрування.

Питання технологій розробки ІС, і СППР зокрема, багато в чому стали вже традиційними й увійшли в класичні підручники. У даній роботі особлива увага приділяється технологіям створення СЗД/ВД, які ще не досить досліджені.

Аналітичний програмний комплекс являє собою розподілену систему, у якій можна умовно виділити три частини:

- підсистему збору оперативних даних;
- підсистему зберігання даних;
- підсистему аналізу даних.

У підсистему збору даних включені будь-які джерела даних:

- оперативні системи підприємства;
- автоматизовані системи управління технологічними процесами;

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 55   |

– зовнішні джерела.

У зв'язку з низьким рівнем автоматизації найважливіших для оперативного управління виробництвом завдань, розробка аналітичного комплексу була почата зі створення системи обліку виробничих бізнес-процесів. Система має клієнт-серверну архітектуру. Специфіка операційних даних, що заносяться майстрами змін за допомогою системи – їх суворі періодичність. Облікові дані заносяться за підсумками зміни. Таким чином, дані мають строгий часовий прив'язку, що характерно для СЗД. Наступною оперативною системою, розробленою для підприємства, стала система, що реалізує функції обліку даних по обслуговуванню, службою експлуатації заводу. У перспективі планується підключення інших оперативних БД інших підрозділів підприємства (склад, служба якості, бухгалтерія).

Підсистема зберігання даних виконує наступні функції:

– збір інформації з різних джерел, насамперед з оперативних інформаційних систем підприємства, що входять у підсистему збору даних, а також від зовнішніх джерел у сховище даних;

– інтеграція даних у логічні моделі по певних предметних областях на OLAP-сервері;

– зберігання інформації в OLAP-кубах таким чином, щоб вона була легко доступна й зрозуміла різним категоріям користувачів;

– надання даних різноманітним додаткам підсистеми аналізу даних.

У запропонованій структурі СЗД має дворівневу структуру:

– стандартний архів;

– вітрини даних (ВД).

Дані з оперативних джерел за допомогою розроблених програм завантажуються в стандартний архів даних, з якого вони вивантажуються у ВД, організовані відповідно до їхньої предметної спрямованості.

Із ВД формуються OLAP-куби, які надають зручні швидкодіючі засоби доступу, перегляду й аналізу ділової інформації. Користувач одержує природну, інтуїтивно зрозумілу модель даних, організувати їх у вигляді багатомірних кубів.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 56   |

Осями багатомірної системи координат служать основні атрибути аналізованого бізнес-процесу: час роботи зміни, номер зміни, характеристики встаткування й т.д. На перетинаннях осей – у термінології OLAP, названих «вимірами» (Dimensions) – перебувають дані, що кількісно характеризують процес, – «міри» (Measures). Це можуть бути обсяги випущеної продукції, надходження, витрати й залишки матеріалів і ін.. Користувач може «розрізати» куб по різних вимірах, одержувати зведені (наприклад, по роках, кварталах, тижнях) або, навпаки, детальні (по змінах) відомості й здійснювати інші маніпуляції, необхідні в процесі аналізу.

За допомогою засобів OLAP-клієнта керівництво підприємства має можливість оперативно одержувати аналітичні звіти про випуск продукції й простої встаткування в характерні для даної інформації розрізах у різних одиницях виміру (кількість у штуках, коробках, кілограмах, відсотках і т.д.). На кнопках зі списками, що розкриваються, можна задавати умови вибору даних, а шляхом простого перетаскування полів задавати виміру. Дані можуть бути представлені в табличному й у графічному виді.

Візуалізація й агрегування даних, звичайно, полегшують прийняття управлінських рішень, однак засоби OLAP-аналізу не виробляють рекомендації ОПР. При прийнятті рішень корисне виявлення якісне нової інформації в даних, що втримуються в сховище, у вигляді схованих закономірностей, залежностей і взаємозв'язків, що забезпечується методами й засобами ІАД. Ілюстрація роботи інструментарію вироблення рекомендацій ОПР, як елемента нової технології, що реалізують її алгоритмів і програмних засобів була здійснена на конкретному завданні визначення складу ремонтних робіт методами імітаційного моделювання, первинної статистичної обробки даних і статистичної перевірки гіпотез.

Одна з найважливіших проблем – це позапланові простої встаткування. У зв'язку з безперервним технологічним процесом, зупинка встаткування спричиняє зниження якості продукції й утворення дорогих відходів. Крім того, своєчасна заміна деяких деталей запобігає дорогому ремонту цілих вузлів. Вище сказане

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 57   |

спричиняється актуальність завдання моделювання (прогнозування) позапланових простоїв устаткування.

Виділено основні **етапи рішення** поставленого завдання:

1. Оцінка статистичних характеристик досліджуваних вибіркового даних (середнє, мінімальне, максимальне, найбільш імовірне (модальне) час безвідмовної роботи й простоїв машин; середньоквадратичне відхилення, розкид часів і т.д.).

2. Дослідження законів розподілу часу безвідмовної роботи й часів простоїв машин по кожній технічній причині й підбор теоретичних розподілів, що адекватно описує вибіркoві дані, на основі використання статистичних критеріїв згоди.

3. Імітаційне моделювання циклів роботи машин за допомогою календарного методу, коли і-цикл складається з ділянки робочого стану (інтервал часу роботи), за яким треба ділянка поломки (простою) внаслідок виникнення певної технічної причини.

4. Прогнозування моментів зупинки машини по певній технічній причині.

Впровадження розробленої системи дозволило скоротити трудомісткість стандартних операцій по обліку даних виробничих бізнес-процесів і формуванню стандартних аналітичних звітів. Середня кількість помилок, що виявляються в момент формування місячного звіту по випуску, скоротилося з п'яти до однієї на місяць.

Розглянемо функціональну модель системи, відображену на рисунку 3.2. З різних моделей життєвого циклу інформаційних систем при створенні СЗД була обрана спіральна модель. Це пов'язане з необхідністю виділення всіх необхідних даних для довільних запитів, для чого варто скласти вичерпний перелік необхідних даних і побудувати схему їхніх зв'язків. При цьому із загального масиву виділяється значима інформація й з'ясовується потреба в додаткових джерелах даних для прийняття рішень. Наша практика показала, що оптимальна тривалість ітерації становить 2-3 тижні при 4-5 ітераціях. На етапі експлуатації

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 58   |



– склад і форма результатів, які необхідно одержати по завершенні етапу.

Для підтримки вибору технічних рішень в умовах безлічі критеріїв і альтернатив був розроблений програмний пакет, що реалізує процедуру експертного оцінювання груповим ОПР на основі методів ранжирування й попарного порівняння.

Сам неформалізованим і складним є **перший етап** – бізнес-аналіз процесів і даних підприємств. Розробка аналітичної системи на основі СЗД без подібного аналізу заздалегідь приречена на невдачу – без ясного розуміння розроблювачами цілей бізнесу, способів їхнього досягнення, що виникають при цьому проблем і методів їхнього рішення, ресурси, необхідні для розробки СЗД, будуть витрачені зрячи.

Тому розглянуті джерела визначення інформаційних потреб, причому не тільки поточних, але й потенційно можливих. Ними є:

– інтерв'ю з керівниками й безпосередніми учасниками виділених бізнес-процесів;

– аналітичні звіти, формовані на підприємстві;

– документація по сертифікації підприємства по стандарті ІСО 9000 (опис цілей і бізнес-процесів компанії, контрольних точок і характеристик, що знімаються, бізнес-процесу в цих точках, заходів щодо забезпечення якості продукції).

Традиційні підходи до побудови інформаційно-аналітичних систем виходять із того, що на початку проекту складно сказати, що повинне бути поміщене в СЗД і які аналітичні завдання будуть вирішуватися кінцевими користувачами. Пропонується використовувати підхід, заснований на збалансованій системі показників (ЗСП). Підхід ЗСП дозволяє проектувати інформаційно-аналітичну систему зверху-долілиць і формувати в СЗД показники, що дають аналітикам цілісну картину розвитку підприємства по виділених напрямках.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 60   |

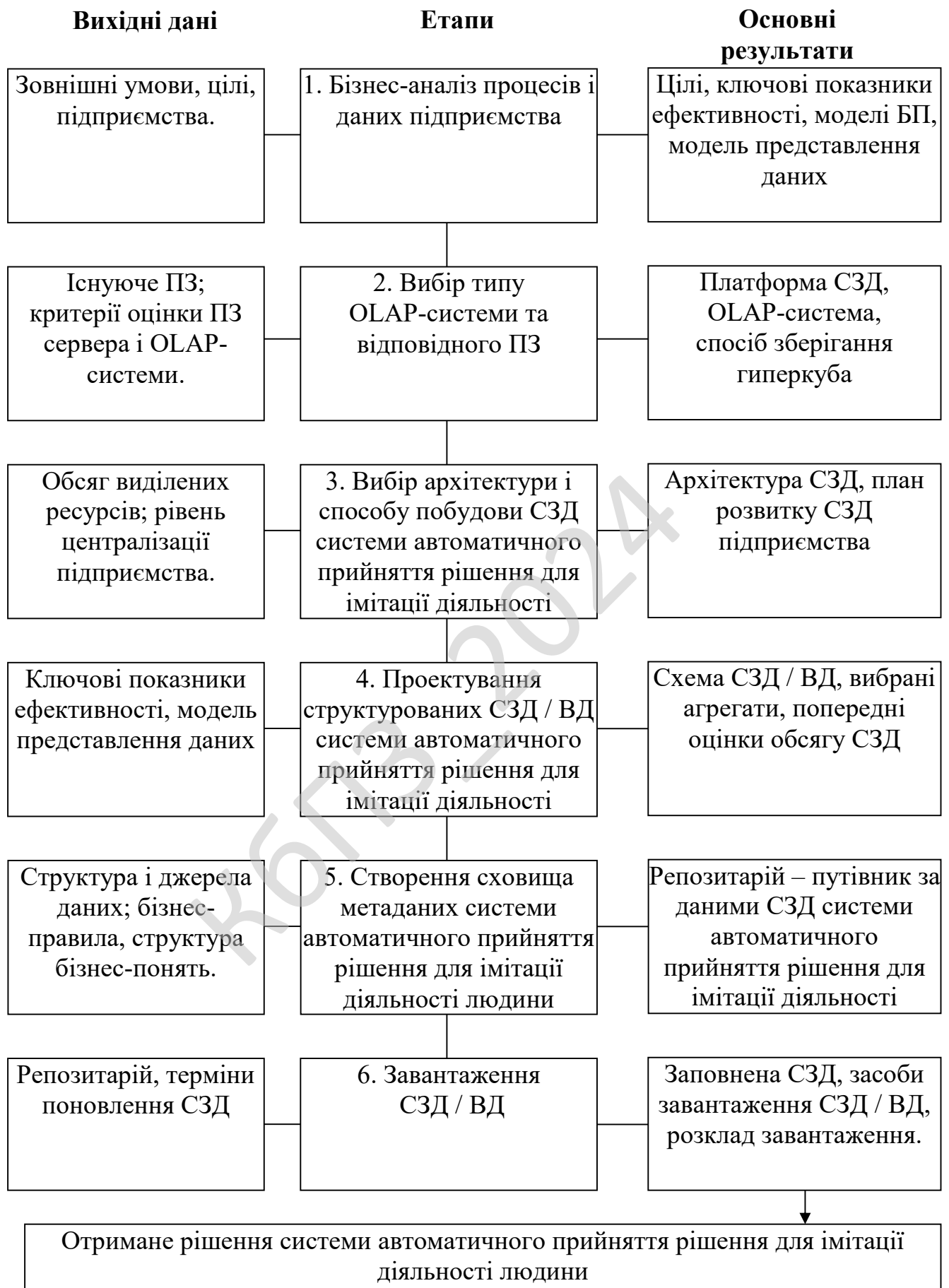


Рисунок 3.2 – Функціональна схема системи

**Другий етап** проекту пов'язаний з розумінням того, у якому виді й на яких апаратних і програмних платформах розміщати структуру даних СППР на основі СЗД. Запропоновано основні критерії вибору OLAP-системи:

- зручність і багатство можливостей засобів адміністрування;
- гнучкість налаштування й наочність форм демонстрації результатів OLAP-систем;
- спектр методів постобробки даних, доступність засобів інтелектуального аналізу;
- можливість обробки більших сховищ даних із прийнятною продуктивністю;
- можливість ув'язування OLAP-Інструментарію з усіма СУБД, використовуваними в організації;
- ціна.

В основі OLAP лежить поняття гіперкуба, або багатомірного куба даних. Залежно від відповіді на питання, чи існує гіперкуб як окрема фізична структура або лише як віртуальна модель даних, виділяють:

- системи MOLAP (Multidimensional OLAP);
- системи ROLAP (Relational OLAP);
- системи HOLAP (Hybrid OLAP).

Показано, що системи третього типу дозволяють сполучити компактне зберігання й підтримку дуже більших БД, забезпечувані ROLAP– системами із простотою налаштування, і гарні часи відгуку при роботі з агрегованими даними, забезпечуваними MOLAP– системами. Саме вони й рекомендуються для побудови СППР.

На **третьому етапі** вибирається архітектура й спосіб побудови СЗД системи автоматичного прийняття рішення для імітації діяльності людини. Існує два основних способи побудови СЗД підприємства: "зверху долілиць" (*top down*) і "знизу нагору" (*bottom up*). Виявлено переваги й недоліки кожного. При підході "зверху долілиць" СЗД розробляється, проектується й будується ітераційним способом. Для підходу «зверху долілиць» характерні великі ресурси (для

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 62   |





Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

Діаграма процесів розробленої системи зображена на рисунку 3.3. При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі.

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

Діаграми потоків даних містять чотири типи елементів:

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

– Сховища даних (репозиторії).

– Зовнішні по відношенню до системи сутності.

– Потоки даних між елементами трьох попередніх типів.

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

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | VKPM-122.24.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 65   |

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

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

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

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

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

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

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

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

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 66   |

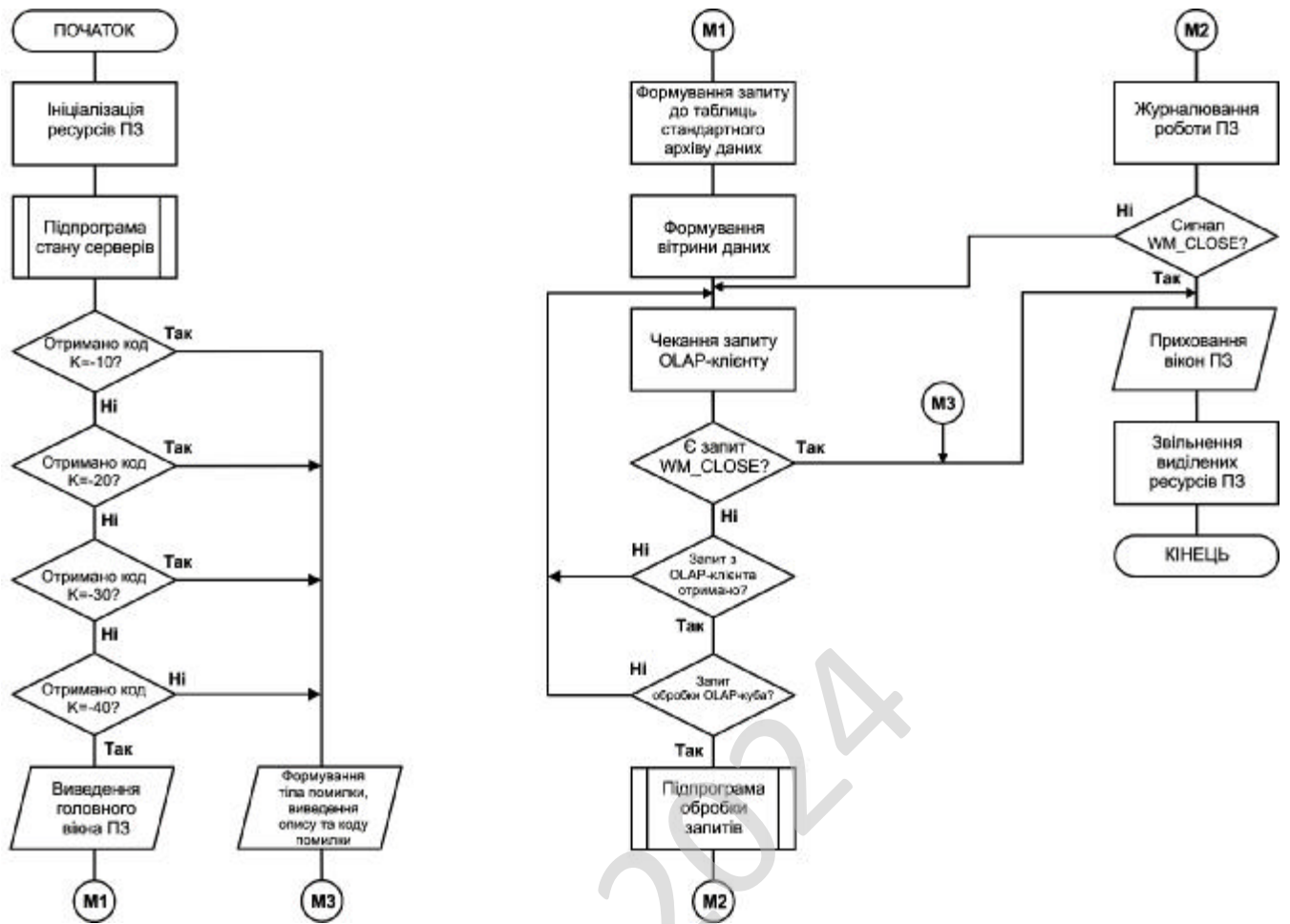


Рисунок 4.1 – Блок-схема основної програми

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

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

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

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

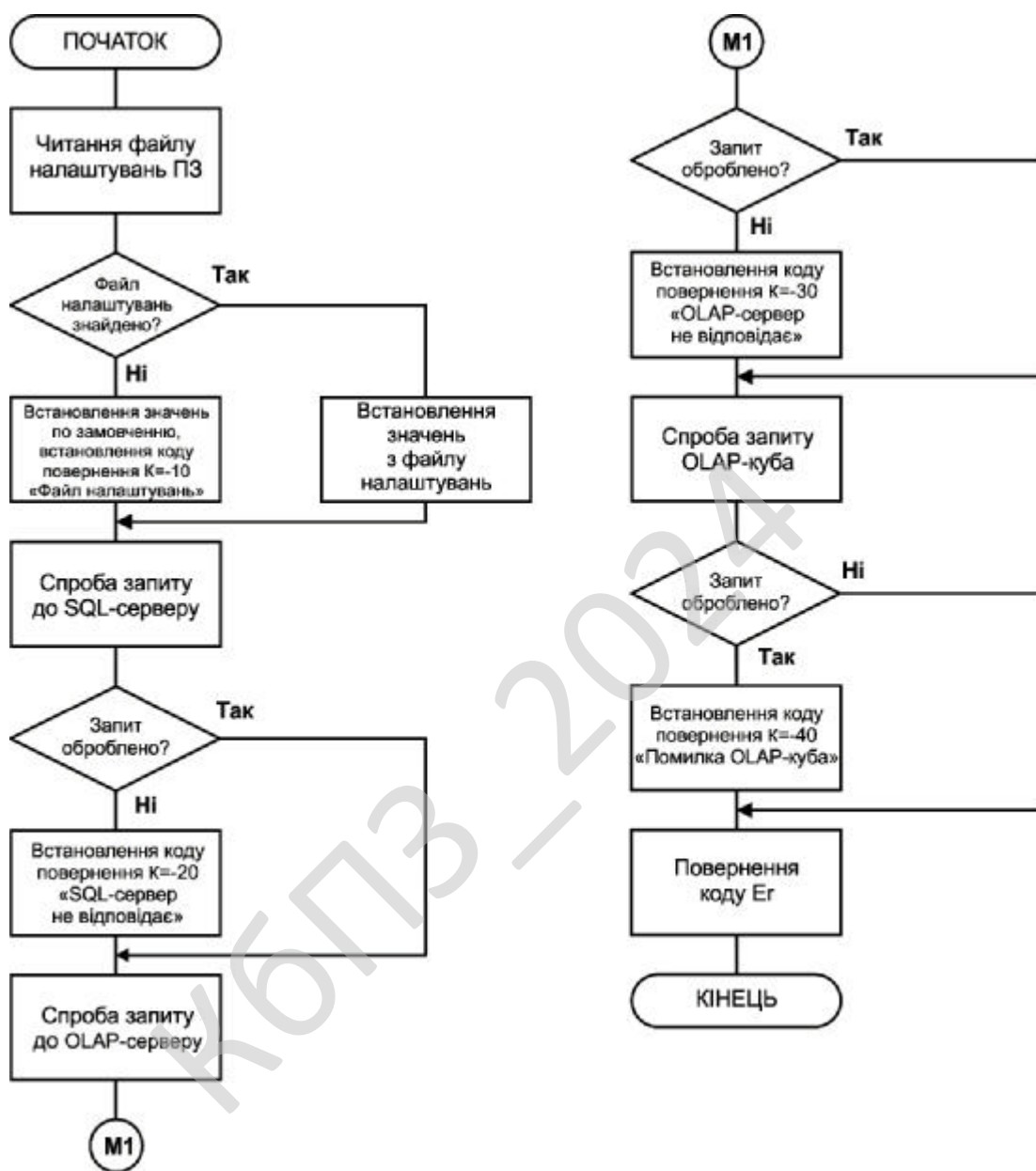


Рисунок 4.2 – Блок-схема роботи підпрограми

Розглянемо використані технології та їх основні компоненти що підтверджують правильність використаних проектних рішень.

Redmine – вільне серверне ПЗ для управління проектами та відстежування помилок. До системи входить календар-планувальник та діаграми Ганта для



Діаграма Ганта являє собою відрізки (графічні плашки), розміщені на горизонтальній шкалі часу. Кожен відрізок відповідає окремому завданню або підзадачі. Завдання і підзадачі, складові плану, розміщуються по вертикалі. Початок, кінець і довжина відрізка на шкалі часу відповідають початку, кінцю і тривалості завдання. На деяких діаграмах Ганта також показується залежність між завданнями.

Діаграма може використовуватися для представлення поточного стану виконання робіт: частина прямокутника, що відповідає завданню, заштриховується, відзначаючи відсоток виконання завдання; показується вертикальна лінія, що відповідає моменту «сьогодні».

Часто діаграма Ганта використовується спільно з таблицею зі списком робіт, рядки якої відповідають окремо взятій задачі, зображеній на діаграмі, а стовпці містять додаткову інформацію про задачу.

Система відстеження помилок Багтрекер – прикладна програма для допомоги розробникам програмного забезпечення (програмістам, тестувальникам тощо) враховувати і контролювати помилки, знайдені у програмах, питання щодо функціональності, рішення та оновлення, побажання користувачів, а також стежити за процесом їх виконання.

Кожному, хто розробляв програмні продукти, добре знайоме співвідношення «20/80» – останні 20 % роботи тривають 80 % часу.

Як це не парадоксально, але нічого дивного в цій пропорції немає, адже саме на завершальній стадії починається тестування проекту, коли виявляються помилки, і що більший проект, то більше буде знайдено помилок.

Водночас досить часто виявляється, що більшість цих помилок були відомі та могли бути виправлені з меншими витратами на попередніх стадіях роботи, але не були вчасно описані, а потім загубилися серед інших важливих завдань.

Отже, система відстеження помилок у найпростішому варіанті – це процес, що включає в себе виявлення помилки, її опис, виправлення і перевірку цього

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 70   |

виправлення, тобто процес «стеження» за багом протягом всього як його життєвого циклу, так і життєвого циклу розробки в цілому.

Сукупність інформації про дефект. Головний компонент такої системи – база даних, що містить відомості про виявлені дефекти. Ці відомості можуть включати в себе:

- номер (ідентифікатор) дефекту;
- хто повідомив про дефект;
- дата і час виявлення дефекту;
- версія продукту, в якій виявлено дефект;
- серйозність (критичність) дефекту та пріоритет рішення;
- опис кроків для відтворення дефекту (неправильної поведінки програми);
- відповідальний за усунення дефекту;
- обговорення можливих рішень та їх наслідків;
- поточний стан виправлення дефекту;
- версії продукту, в якій дефект виправлений.

Крім того, розвинені системи надають можливість прикріплювати файли, які допомагають описати проблему, наприклад, дамп пам'яті або скріншот.

Використання. Основна перевага систем відстеження помилок полягає в забезпеченні чітких централізованих оглядів, запитів на розробку (включаючи помилки і виправлення) та їх стан. У корпоративному середовищі, системи відстеження помилок можуть бути використані для генерації звітів по продуктивності програмістів виправлення помилок. Однак, це може іноді приводити до неточних результатів, тому що різні помилки можуть мати різні ступені пріоритету та серйозності, що пов'язано з складністю їх фіксації.

Життєвий цикл дефекту. Як правило, система відстеження помилок використовує той чи інший варіант «життєвого циклу» помилки, стадія якого визначається поточним станом помилки.

Типовий життєвий цикл дефекту:

1. Новий – дефект зареєстрований тестувальником.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 71   |

2. Призначений – призначений відповідальний за виправлення дефекту.

3. Дозволений – дефект переходить назад у сферу відповідальності тестувальника. Як правило, супроводжується резолюцією, наприклад:

– Виправлено (виправлення включені у версію таку-то).

– Дубль (повторює дефект, що вже знаходиться в роботі).

– Не виправлено (працює відповідно до специфікації, має занадто низький пріоритет, виправлення відкладено до наступної версії тощо).

– «В мене все працює» (запит додаткової інформації про умови, в яких дефект проявляється).

4. Далі тестувальник проводить перевірку виправлення, залежно від чого дефект або знову переходить у стан «Призначений» (якщо він описаний як виправлений, але не виправлений), або у стан «Закрито».

5. Відкрито повторно – дефект знайдено знову в іншій версії.

Система може надавати адміністраторові можливість налаштування користувачі, які можуть переглядати і редагувати помилки залежно від їх стану, переводити їх в інший стан або видаляти.

У корпоративному середовищі, система відстеження помилок може використовуватися для отримання звітів, що показують продуктивність програмістів при виправленні помилок. Однак, часто такий підхід не дає достатньо точних результатів через те, що різні помилки мають різну ступінь серйозності та складності. При цьому серйозність проблеми прямо не стосується складності її усунення.

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

– систему управління;

– систему ідентифікації і вимірювання;

– систему супроводження (моніторингу та контролю).

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 72   |

Сучасна наука представляє ризик як вірогідну подію, в результаті настання якої можуть відбутися позитивні, нейтральні або негативні наслідки. Якщо ризик припускає наявність як позитивних, так і негативних результатів, він відноситься до спекулятивних ризиків. Якщо ж наслідки негативні, або відсутні взагалі, такий ризик іменується чистим.

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

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

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

В кінці ХІХ століття англійський дослідник Ф. Гальтон запропонував вважати регресію або повернення до середнього значення універсальною статистичною закономірністю.

Суть регресії трактувалася ним як повернення явищ до норми з часом. Згодом було доведено, що правило регресії діє в найрізноманітніших ситуаціях, починаючи з азартних ігор та розрахунку вірогідності виникнення нещасних випадків, і закінчуючи прогнозуванням коливань економічних циклів.

У 1952 аспірант Університету Чикаго Гарі Марковіц в статті «Диверсифікація вкладень» («Portfolio Selection») математично обґрунтував стратегію диверсифікації інвестиційного портфеля, зокрема, він показав, як шляхом продуманого розподілу вкладень мінімізувати відхилення прибутковості від очікуваного показника.

У 1990 Г. Марковіцу присуджена Нобелівська премія за розробку теорії і практики оптимізації портфеля фондових активів.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 73   |

## Етапи ризик-менеджменту

У ризик-менеджменті прийнято виділяти декілька ключових етапів:

- на першому етапі відбувається виявлення ризику з супутньою оцінкою вірогідності його реалізації і масштабу наслідків;
- на другому етапі здійснюється розробка ризик-стратегії з метою зниження вірогідності реалізації ризику і мінімізації можливих негативних наслідків;
- на третьому етапі вибираються методи і інструменти управління виявленим ризиком;
- на четвертому етапі проводиться безпосереднє управління ризиком;
- на завершальному етапі оцінюються досягнуті результати і коректується ризик-стратегія.

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

## Методи і інструментарій ризик-менеджменту

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

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

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

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

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 74   |

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

Мета управління вимогами полягає в тому, щоб переконатися, що організація відповідає потребам і очікуванням своїх клієнтів, внутрішніх або зовнішніх зацікавлених сторін. Управління вимогами починається з аналізу і виявлення цілей і обмежень організації. Управління вимогами додатково включає в себе підтримку планування вимог, інтеграції вимог і організації роботи з ними (атрибути для вимог).

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

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

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

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 75   |

розробки для пріоритезації вимог, визначаючи, наскільки цінною є дана вимога для певного користувача.

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

### **Завдання управління вимогами**

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

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

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

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

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 76   |

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

Ці інструменти керують вимогами використовуючи базу даних, і зазвичай мають функції автоматизації відстеження (наприклад, дозволяючи створювати зв'язки між батьківськими і дочірніми вимогами, або між тестами і вимогами), управління версіями, і управління змінами. Зазвичай такі інструментальні засоби містять функцію експорту, яка дозволяє створювати звичайний документ, екпортуючи дані вимог.

### **Аналіз здійсненості**

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

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

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

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

Ці питання також вказують на основну суть управління вимогами. Людина і інструмент формують систему, і це розуміння особливо важливе, якщо

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 77   |

інструмент – комп'ютер або новий додаток на комп'ютері. Людський розум вкрай ефективний у паралельній обробці та інтерпретації тенденцій з недостатніми даними. Комп'ютерний процесор ефективний у послідовній обробці і точному математичному обчисленні. Основна мета управління вимогами для програмного проекту полягала б у тому, щоб гарантувати, що автоматизована робота призначена «правильному» процесору.

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

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

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

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

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

Розробка та тестування. На стадії розробки і тестування, основна

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-122.24.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 78   |



стандартів у галузі проектів, що стосуються розробки нових продуктів. Ця модель багато в чому схожа з Prince2 і описує методи як для проектного управління, так і для системного розвитку.

### **Основні принципи**

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

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

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

### **Цілі**

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

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

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

|      |      |          |        |      |                                  |           |
|------|------|----------|--------|------|----------------------------------|-----------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк.      |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | <b>80</b> |

3. Зменшення загальної вартості проекту: Ресурси на розробку, виробництво, управління і підтримку можуть бути заздалегідь прораховані та проконтрольовані. Отримувані результати також універсальні і легко прогножуються. Це зменшує витрати на подальші стадії та проекти.

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

**Переваги:**

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

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

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

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

– Не регулюється розміщення контрактів на обслуговування.

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

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

**Переваги:**

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

проектування. Фази інтеграції та тестування реалізують архітектурне проектування або проектування на вищому рівні. Фаза тестування системи підтверджує правильність виконання етапу вимог до продукту і його специфікації.

– У моделі передбачені атестація та верифікація всіх зовнішніх і внутрішніх отриманих даних, а не тільки самого програмного продукту.

– У V-подібної моделі визначення вимог виконується перед розробкою проекту системи, а проектування ПО – перед розробкою компонентів.

– Модель визначає продукти, які повинні бути отримані в результаті процесу розробки, причому кожен отриманий дані повинні піддаватися тестуванню.

– Завдяки моделі менеджери проекту можуть відслідковувати хід процесу розробки, так як в даному випадку цілком можливо скористатися тимчасовою шкалою, а завершення кожної фази є контрольною точкою.

Недоліки:

– Модель не передбачає роботу з паралельними подіями.

– У моделі не передбачено внесення вимоги динамічних змін на різних етапах життєвого циклу.

– Тестування вимог в життєвому циклі відбувається занадто пізно, внаслідок чого неможливо внести змін, не вплинувши при цьому на графік виконання проекту.

– У модель не входять дії, спрямовані на аналіз ризиків.

– Деякий результат можна отримати тільки при досягненні низу букви V.

#### 4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою Sinople – симетричний блоковий криптоалгоритм, побудований на основі незбалансованої «мережі Фейстеля». Алгоритм розроблено у 2003 році.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-122.24.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 82   |

Основні вимоги до алгоритму при його розробці:

- Можливість програмної і апаратної реалізації.
- Висока швидкість.
- Простота.
- Низькі вимоги до пам'яті.
- Високий рівень безпеки.

Алгоритм заснований на 32-розрядних операціях і має 64 раунду, серед яких два типи – С і D. D раунди спроектовані для досягнення максимальної дифузії, С раунди – для досягнення перемішування. F-функція D раунду використовує один з елементів блоку даних ( $D[3]$ ) та поточного з'єднання ( $K[r]$ ) для трансформації 3-х елементів блоку даних. F-функція С раунду, навпаки, використовує перші три елемента блоку даних і поточний з'єднання ( $K[r]$ ) для трансформації останнього елемента блоку даних ( $D[3]$ ). Раунди D-типу виконуються до раундів С-типу. Додавання ключів з даними проводиться тільки через таблиці замін. Операції XOR (додавання по модулю 2) обов'язково поєднуються з операціями ADD (додавання по модулю  $2^{32}$ ).

Таблиці замін спочатку запозичені з алгоритму MARS і містять 512 32-розрядних елементів, проте були жорстко проаналізовані на предмет посилення.

Ключове розклад було спроектовано з урахуванням вимог:

- Простота
- Використовується та ж процедура, що і при шифруванні та розшифрування
- Установка ключа займає менше часу, ніж зашифрування
- Виключення еквівалентних ключів
- Виключення слабких ключів

Алгоритм, згідно із заявою авторів, стійкий до лінійного і диференціального аналізу.

## 5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ системи автоматичного прийняття рішення для імітації діяльності людини яке зображено на рисунку 5.1. Обчислення виконується через консольний інструмент з подальшою передачею результатів до інтерфейсу. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

- Навігаційне меню: OLAP-сервер; SQL-сервер; Налаштування; Довідка.
- Розділу виведення результату роботи системи.
- Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші, функціональних кнопок ПЗ.

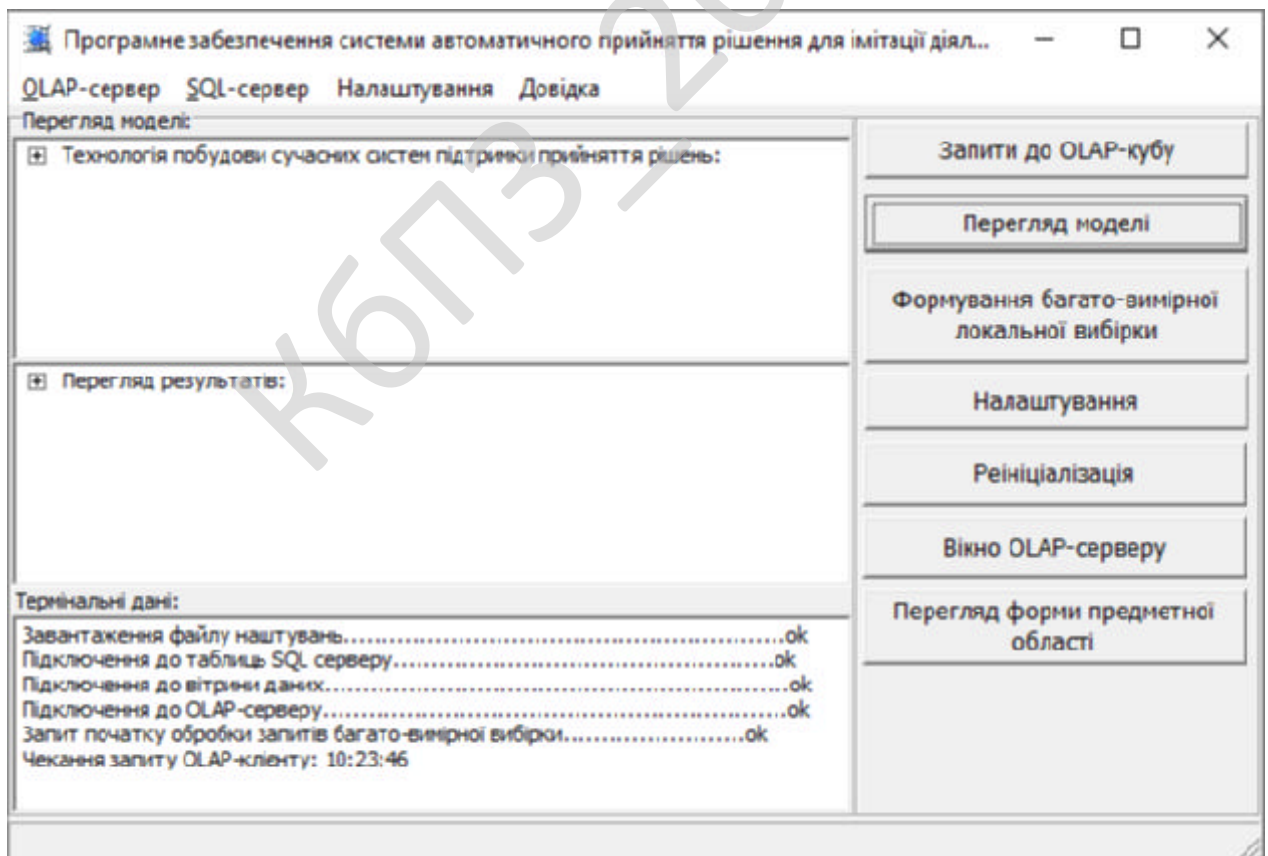


Рисунок 5.1 – Головне вікно ПЗ

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

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

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

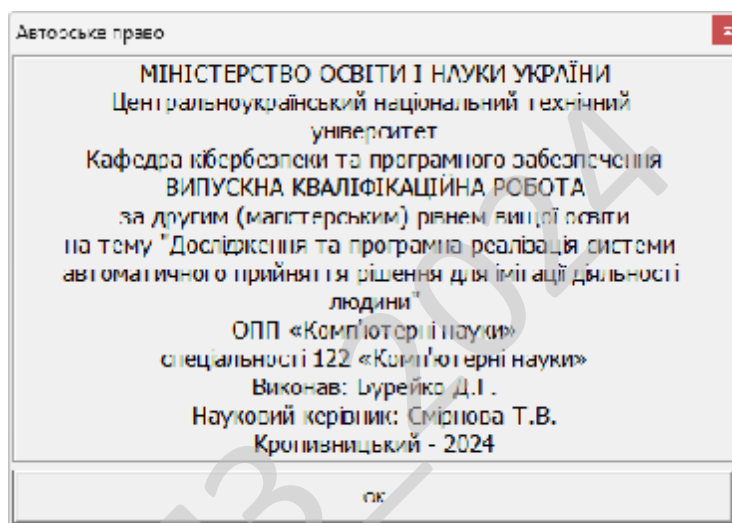


Рисунок 5.2 – Авторське право

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

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 85   |

унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, "розгортання" можна трактувати як загальний процес відповідно до певних вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

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

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

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

– 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

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 86   |

в IT рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

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

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

Проводилась оцінка:

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

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

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

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

У цьому випадку формуються тестові варіанти, в яких:

- Гарантується перевірка всіх незалежних маршрутів програми.
- Знаходяться гілки True, False для всіх логічних рішень.
- Виконуються всі цикли (у межах їхніх кордонів та діапазонів).
- Аналізується правильність внутрішніх структур даних.

Недоліки тестування "білої скриньки":

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 87   |

- Кількість незалежних маршрутів може бути дуже велика.
- Повне тестування маршрутів не гарантує відповідності програми вихідним вимогам до неї.
- У програмі можуть бути пропущені деякі маршрути.
- Не можна виявити помилки, поява яких залежить від даних.

Переваги тестування "білої скриньки" пов'язані з тим, що принцип «білої скриньки» дозволяє врахувати особливості програмних помилок:

- Кількість помилок мінімально в «центрі» і максимально на «периферії» програми.
- Попередні припущення про ймовірність потоку керування або даних у програмі часто бувають некоректними. У результаті типовим може стати маршрут, модель обчислень за яким опрацьована слабо.
- При записі алгоритму програмного забезпечення у вигляді тексту на мові програмування можливе внесення типових помилок трансляції (синтаксичних та семантичних).

– Деякі результати в програмі залежать не від вихідних даних, а від внутрішніх станів програми.

Обрано умови розповсюдження – Freeware.

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

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

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

Безплатне програмне забезпечення можна безоплатно встановлювати та використовувати (іноді з певними обмеженнями, як, наприклад, «безплатне для

домашнього або некомерційного вжитку»), в той час як вільне програмне забезпечення можна продавати за будь-яку суму, але при тому, у користувача, котрий його отримує, повинні бути права на вивчення, модифікацію та поширення сирцевих кодів одержаної програми.

КБПЗ\_2024

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | VKPM-122.24.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 89   |

## 6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи автоматичного прийняття рішення для імітації діяльності людини.

*Метою розробки є дослідження та програмна реалізація системи автоматичного прийняття рішення для імітації діяльності людини.*

*Об'єктом дослідження є процес автоматичного прийняття рішення для імітації діяльності людини.*

*Предметом дослідження є методи автоматичного прийняття рішення для імітації діяльності людини.*

*Методи дослідження базуються на методах машинного навчання, методах математичної статистики, методах розробки програмного забезпечення.*

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

- Удосконалено метод автоматичного прийняття рішення для імітації діяльності людини.
- Розроблено вітчизняний продукт автоматичного прийняття рішення для імітації діяльності людини, який має більш широкі можливості, на відміну від існуючих аналогів.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-122.24.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 90   |

## 7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

### 7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та програмної реалізації системи автоматичного прийняття рішення для імітації діяльності людини можуть бути цікавими для різних груп та секторів. Ось кілька прикладів:

|   |
|---|
| 1. Компанії, що працюють в галузі штучного інтелекту (ШІ) та машинного навчання |
| 2. Підприємства, що займаються автоматизацією бізнес-процесів                   |
| 3. Розробники програмного забезпечення і стартапи                               |
| 4. Організації у сфері охорони здоров'я   |
| 5. Урядові органи та державні установи  |
| 6. Освітні установи та дослідницькі організації                                 |
| 7. Фінансові організації та інвестори   |
| 8. Підприємства в сфері робототехніки   |
| 9. Маркетингові агенції та аналітики  |
| 10. Інтернет-платформи та соціальні мережі                                      |
| 11. Підприємства у сфері безпеки та моніторингу                                 |
| 12. Клієнти у сфері розваг і медіа  |

Рисунок 7.1 – Цільова аудиторія

Всі ці групи можуть отримати значну вигоду від дослідження та впровадження програмних систем, що імітують людське прийняття рішень, оскільки вони забезпечують більш ефективні, точні та швидкі рішення в різних сферах діяльності.

## 7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Оцінка привабливості програмної реалізації системи автоматичного прийняття рішень для імітації діяльності людини через методи експертних оцінок. Для оцінки привабливості проекту доцільно залучити експертів з різних галузей: ІТ та розробка програмного забезпечення (експерти, що мають досвід у створенні систем автоматичного прийняття рішень), бізнес-аналітики та економісти (для оцінки економічної ефективності та вигод від впровадження), фахівці з штучного інтелекту та машинного навчання (для оцінки технічних можливостей системи), представники потенційних користувачів (це можуть бути бізнес-користувачі або клієнти, які використовуватимуть систему).

Далі визначаємо ключові критерії, за якими експерти будуть оцінювати привабливість програмної реалізації системи (рисунок 7.2).

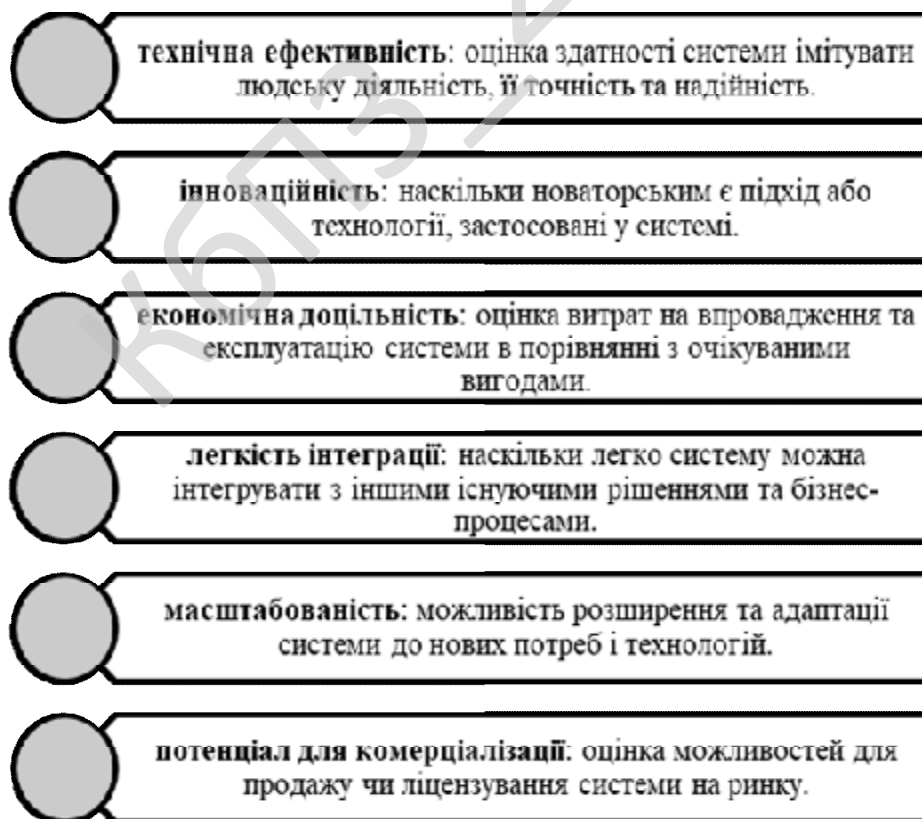


Рисунок 7.2 – Критерії оцінювання проекту експертами

Оцінку можна проводити за допомогою шкали, наприклад, від 1 до 5 або 1 до 10, де: 1 – дуже низький рівень, 5 або 10 – високий рівень привабливості для кожного критерію.

Кожен експерт оцінює кожен критерій, після чого отримуються середні бали для кожного критерію, які зводимо до таблиці 7.1.

Таблиця 7.1 – Зведені результати оцінювання

| Критерій                      | Експерт 1 | Експерт 2 | Експерт 3 | Середній бал |
|-------------------------------|-----------|-----------|-----------|--------------|
| Технічна ефективність         | 8         | 9         | 8         | 8.33         |
| Інноваційність                | 7         | 8         | 9         | 8.00         |
| Економічна доцільність        | 6         | 7         | 6         | 6.33         |
| Легкість інтеграції           | 7         | 7         | 8         | 7.33         |
| Масштабованість               | 8         | 8         | 9         | 8.33         |
| Потенціал для комерціалізації | 7         | 8         | 7         | 7.33         |

Загальний середній бал для кожного критерію допомагає зрозуміти, як експерти оцінюють привабливість проекту. Вищий бал у категоріях, таких як технічна ефективність, масштабованість та інноваційність, свідчить про те, що система має потенціал для успішного впровадження та розвитку. Якщо виявляються слабкі місця, наприклад, в економічній доцільності або легкості інтеграції, це може бути сигналом для коригування проекту.

Кількісні та якісні оцінки, отримані від експертів, дозволяють зробити висновки щодо того, наскільки перспективним і привабливим є проект з погляду технічного та економічного потенціалу. Цей підхід допомагає приймати обґрунтовані рішення щодо подальшого розвитку, фінансування та впровадження програмної реалізації системи.

Цей метод експертних оцінок надає структурований підхід до оцінки привабливості проекту, що дозволяє мінімізувати суб'єктивність та забезпечити прийняття обґрунтованих рішень.

### 7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості програмної реалізації системи автоматичного прийняття рішень для імітації діяльності людини доцільно застосувати комплексний підхід. Пропонуємо розглядати варіант використання методу аналогів та експертних оцінок.

| 1. Метод порівняння з аналогами (Market-based method)  | 2. Метод експертних оцінок (Expert judgment method)   |
|--|---|
| <ul style="list-style-type: none"> <li>• Цей метод полягає в оцінці вартості на основі порівняння з іншими подібними проектами або продуктами на ринку. Для цього проводиться аналіз аналогічних систем автоматичного прийняття рішень, зокрема в галузях штучного інтелекту та машинного навчання.</li> <li>• <b>Переваги:</b> оцінка вартості на основі реальних даних з ринку; допомагає зрозуміти рівень конкурентоспроможності та можливості для монетизації.</li> <li>• <b>Недоліки:</b> може бути важко знайти точні аналоги для специфічних або інноваційних проектів; зміни на ринку можуть вплинути на точність порівняння.</li> </ul> | <ul style="list-style-type: none"> <li>• Цей метод полягає в тому, щоб залучити експертів для оцінки вартості проекту на основі їх досвіду, знань і попередніх проектів. Експерти можуть використовувати кілька критеріїв, таких як складність задачі, рівень технологій, доступність ресурсів, і прогнозовані витрати.</li> <li>• <b>Переваги:</b> здатність оцінити складні чи нові проекти, де відсутні чіткі ринкові аналоги; може враховувати не тільки фінансові аспекти, але й інші чинники, такі як технічний ризик.</li> <li>• <b>Недоліки:</b> суб'єктивність оцінки, що може призвести до помилок, потребує залучення висококваліфікованих експертів.</li> </ul> |

Рисунок 7.3 – методи оцінки вартості ПЗ

Кількість методів та їх вид остаточно залежить від етапу розвитку проекту, характеру витрат і типу інвестицій.

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

Економічна ефективність від впровадження системи автоматичного прийняття рішення для імітації діяльності людини може бути оцінена на основі різних факторів, таких як зниження витрат, підвищення продуктивності, скорочення часу на прийняття рішень та підвищення точності результатів. Ось приклад того, як можна оцінити економічну ефективність цього проєкту.

Крім прямих фінансових вигод, є і додаткові переваги:

- покращення якості обслуговування клієнтів завдяки швидшим та точнішим рішенням;
- підвищення конкурентоспроможності на ринку;
- зниження ризику людських помилок у критичних ситуаціях.

Впровадження системи автоматичного прийняття рішень для імітації діяльності людини може призвести до значних економічних вигод, зокрема за рахунок зменшення витрат на персонал, збільшення продуктивності та зменшення кількості помилок. Це дозволяє суттєво підвищити економічну ефективність проєкту.

Таблиця 7.2 – Основні показники впровадження проєкту

### 1. Передумови для впровадження системи:

Проєкт стосується автоматизації прийняття рішень в компанії, яка працює в галузі фінансових послуг.

Раніше рішення приймалися вручну, що вимагало залучення значної кількості співробітників та часу.

Система автоматичного прийняття рішень повинна імітувати людське прийняття рішень, базуючись на аналізі великих обсягів даних і виявленні патернів.



## 7.5 Пропозиція алгоритму просування проекту розробки ПЗ

Для ефективного просування проекту програмної реалізації системи автоматичного прийняття рішення для імітації діяльності людини важливо розробити детальний алгоритм, який охоплюватиме як технічні, так і маркетингові аспекти (рисунок 7.4).

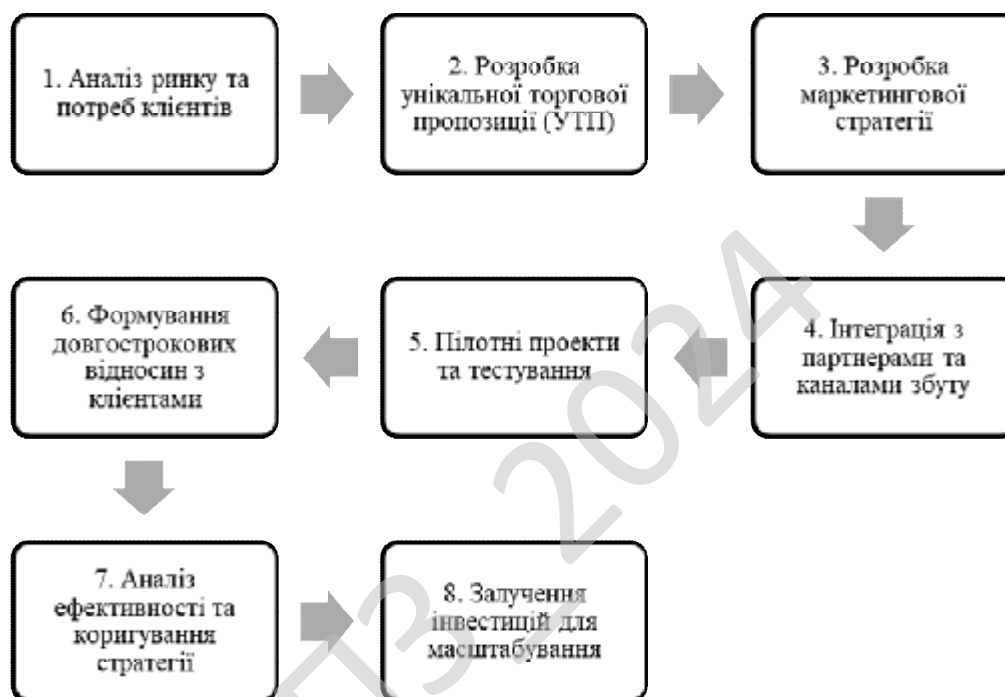


Рисунок 7.4 – Алгоритм просування проекту

Цей алгоритм охоплює всі етапи просування проекту, від розробки до масштабування, і дозволяє ефективно просувати програмну реалізацію системи автоматичного прийняття рішень для імітації діяльності людини на ринку.

## 7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Оптимізація каналів збуту та шляхів реалізації проекту програмної реалізації системи автоматичного прийняття рішень для імітації діяльності

людини вимагає інтегрованого підходу, який включає як технологічні, так і маркетингові стратегії (рисунок 7.5).

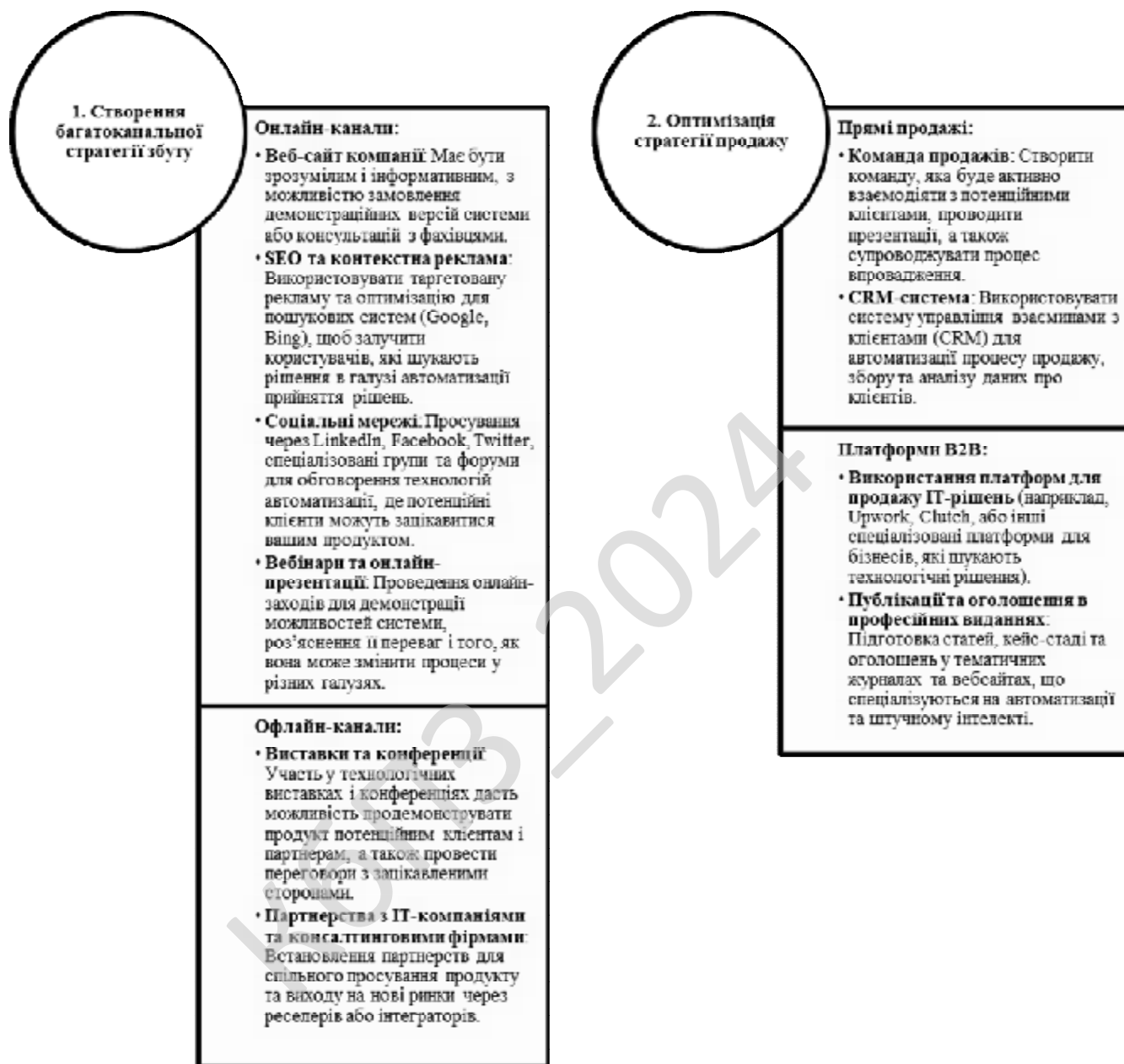


Рисунок 7.5 – Напрями оптимізації каналів збуту

Ці стратегії допоможуть ефективно оптимізувати канали збуту та шляхи реалізації проєкту, забезпечуючи стійке зростання продажів і максимальну вигоду для обох сторін – розробника і клієнтів.

## 7.7 Визначення ключових факторів успіху конкретного проєкту

Ключовими факторами успіху проєкту програмної реалізації системи автоматичного прийняття рішення для імітації діяльності людини є кілька критичних аспектів, які забезпечують ефективне впровадження та стабільну роботу такої системи (рисунок 7.6).

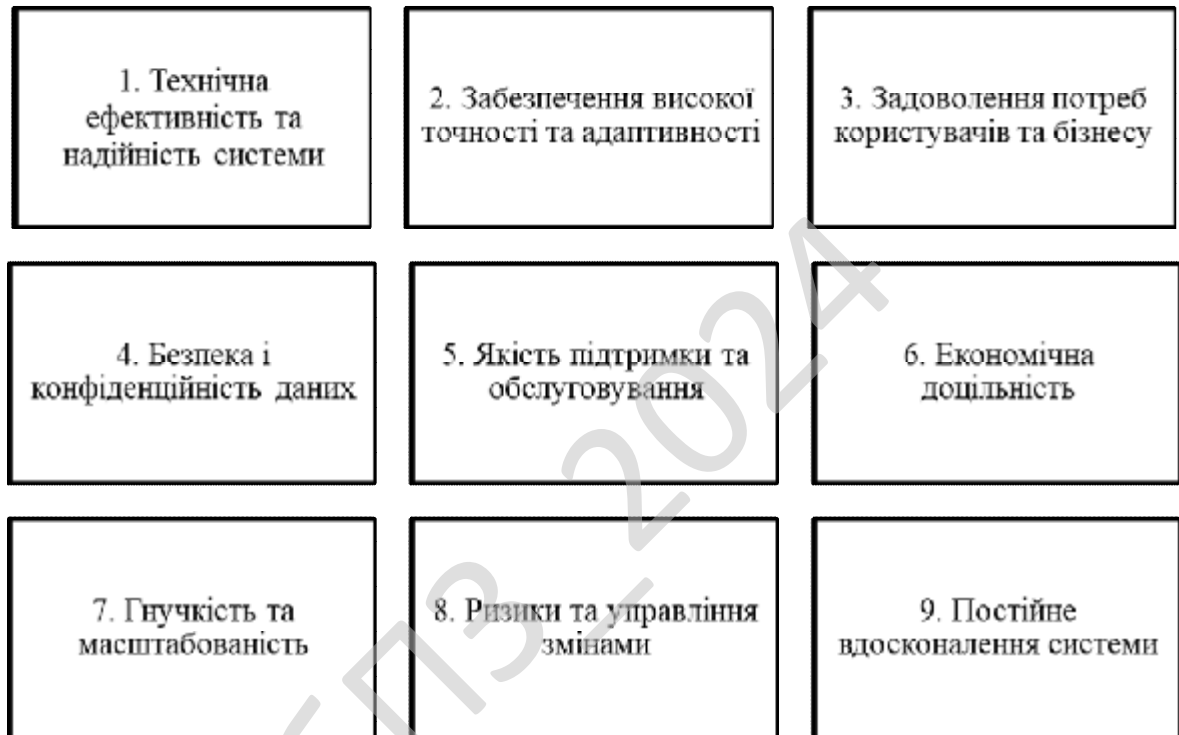


Рисунок 7.6 – Ключові фактори успіху проєкту

Забезпечення успіху проєкту програмної реалізації системи автоматичного прийняття рішень для імітації діяльності людини потребує збалансованого підходу до технічних, економічних і організаційних аспектів. Успішна реалізація такого проєкту забезпечить значні переваги для бізнесу, включаючи підвищення ефективності, швидкості прийняття рішень та зниження витрат.

## 8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

### 8.1 Вступ

Охорона праці – система збереження життя і здоров'я працівників у процесі трудової діяльності, що включає правові, соціально-економічні, організаційні, технічні, санітарно-гігієнічні, лікувально-профілактичні, реабілітаційні та інші заходи.

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

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

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

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

Законом України “Про охорону праці” [3] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-122.24.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 100  |

з екранними пристроями» [5], яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [2].

Постійне перебування за комп'ютером негативно впливає на органи зору та поставу. Тому вже після першого року роботи, багатьом фахівцям у сфері ІТ знайомі такі хвороби, як остеохондроз, синдром сухого ока, безсоння та інше.

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

Ці шкідливі фактори можуть привести до професійних захворювань.

Науково-технічний прогрес вніс серйозні зміни в умови виробничої діяльності робітників розумової діяльності. Їх праця стала більш інтенсивною, напруженою і вимагає значних витрат розумової, емоційної і фізичної енергії. Це призвело до необхідності у знаходженні комплексного рішення проблем ергономіки, гігієни і організації праці, регламентації режимів праці та відпочинку. Охорона здоров'я робітників, забезпечення безпеки умов праці, ліквідація та профілактика професійних захворювань і виробничого травматизму складає одну з головних турбот людського суспільства.

## 8.2 Пожежна безпека

Вимоги до пожежної безпеки на підприємстві неухильно повинен дотримуватися кожен співробітник, а організаційна складова при цьому

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 101  |

покладається на посадових осіб за відповідним рішенням керівництва і прописується в посадових інструкціях і положеннях по структурним підрозділам.

Зокрема, вказуються конкретні території, ділянки, зони, об'єкти, цілі будівлі і їх частини, поверхи, на яких відповідального співробітника повинне проводити такі організаційні роботи.

Відповідальні особи зобов'язуються розробити, впровадити та підтримувати в певному інструкцією і положенням на ввірених їм об'єктах протипожежний режим і інструкції відповідно до вимог, викладених в нормативних актах.

Передбачено також створення підрозділу добровільної пожежної охорони та пожежно-рятувальної команди в його складі.

Встановлений режим включає порядки з описом місць спеціального призначення та правила їх користування та утримання, наприклад:

- евакуаційних шляхів;
- так званих «курилок»;
- місць складування продукції та сировини;
- стоянки транспорту.

Також встановлюється порядок роботи та технічного обслуговування:

- вентиляційного устаткування;
- засобів пожежогасіння і захисту від загорянь;
- нагрівальних приладів;
- електрообладнання.

Розробляються і впроваджуються правила роботи з відкритим вогнем і горючими матеріалами. Створюються графіки проходження інструктажів з пожежної безпеки співробітників, а також порядок і терміни перевірок знань пожежно-технічного мінімуму, в тому числі, тих працівників, які відповідальні за цю ділянку роботи на підприємстві. При цьому можуть передбачатися внутрішні лекції, семінари, тренінги та практичні заняття на підприємстві, а також зовнішні – на базі спеціалізованих навчальних центрів з професійними

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 102  |

викладачами.

Важливою складовою протипожежного режиму на будь-якому об'єкті є розробка і впровадження порядку дій при виникненні пожежі. Неодмінно має бути план евакуації, описано, як повинні відключатися електроустановки, що і в якій послідовності необхідно робити співробітникам.

Відповідно, для кожного об'єкта, кожного приміщення (крім коридорів, санвузлів, басейнів і подібних приміщень), окремих видів робіт складаються інструкції, за якими повинен працювати персонал, залучений на певних ділянках і в виконанні окремих видів робіт. За інструкціями проводиться навчання (інструктаж) персоналу з подальшим контролем знань.

Детально про те, як розробити протипожежний режим, прописати порядки та інструкції, пояснюють на тематичних курсах і семінарах. [4]

### 8.3 Характеристика умов праці програміста

В приміщенні, в якому проводиться розробка і дослідження програмного продукту, відсутні умови, які можуть створювати підвищену або особливо підвищену небезпеку, тому воно відноситься до класу звичайних приміщень згідно Правил улаштування електроустановок (ПУЕ). Джерелом живлення є трифазна мережа напруги 380/220 В з глухо заземленою нейтралі, з частотою 50 Гц згідно За вибухопожежонебезпеку приміщення відноситься до класу В. В таблиці 8.1 наведена загальна характеристика приміщення щодо вибухопожежонебезпеки та важкістю робіт.

Температура повітря в приміщенні визначається температурою зовнішнього повітря і тепловою енергією, що виділяється всередині приміщення. Джерелами теплоти в даному приміщенні є люди, електроустаткування, а також освітлювальні прилади в темний час доби. Зовнішнім джерелом надлишкового тепла є сонячна радіація у світлий час доби. Робота, виконувана в даному приміщенні, відноситься до категорії І-а. Людиною в цьому випадку виділяється

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 103  |

до 120 ккал теплової енергії за годину. Вологість повітря в приміщенні визначається вологістю атмосферного і видихуваного людьми повітря, а також випарами з поверхні шкіри.

Таблиця 8.1 – Загальна характеристика приміщення щодо вибухопожежонебезпеки та важкістю робіт

| Характеристика приміщень за вибухопожежною категорією та класом зони | Загальна характеристика приміщення  | Категорія за важкістю робіт згідно ГН 3.3.5-8.6.6.1-2002  |
|--|---|---|
| В – пожежонебезпечне клас П – П                                      | Звичайне без ознак хімічного забруднення; нормальної вологості і за санітарними нормами | 1а – до 139 Вт/м <sup>2</sup><br>1б – до 140-174 Вт/м <sup>2</sup><br>Клас умов праці – оптимальний |

У приміщенні немає виділення шкідливих газів. Тому що в ньому не проводяться монтажні роботи, як пайка, зварювання чи інші, при яких виділяються шкідливі гази. Для нормалізації параметрів повітряного середовища також періодично здійснюється провітрювання приміщення і вологе прибирання. У всьому будинку діє встановлена загально обмінна витяжна вентиляція.

Раціональне освітлення приміщення сприяє кращому виконанню виробничих завдань і забезпеченню комфорту при роботі. Для забезпечення нормального освітлення застосовуються природне, однобічне, бічне і штучне освітлення, а також комбіноване, нормуються згідно ДБН В.2.5-28-2006 Природне і штучне освітлення. За результатами виміру освітленості величина освітленості від системи загального штучного освітлення дорівнює 310 лк, що відповідає вимогам, які пред'являються до даного приміщення [1].

Основними джерелами шуму на робочих місцях, обладнаних відео дисплейними терміналами, є принтер, сканер факс і обладнання для кондиціонування повітря, в самих відео дисплейних терміналів – вентилятори систем охолодження і трансформатори.

Згідно ДСанПіН 3.3.2.007-98 [2] допустимий еквівалентний рівень шуму для робочого місця програміста складає 50 дБА (акустичних децибела).

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

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

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

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) [4, 10, 11].

#### **8.5 Розрахункова частина**

Для захисного штучного заземлення застосовуються вертикальні електроди: металевий куток 63х63х6 мм, (згідно з ДСТУ 2251-93 «Кутики сталеві гарячекатані рівнополічні. Сортамент») довжиною  $L=2$  м, та горизонтальний електрод – металева полоса з перетином 60х5 мм. Напруга – 220/380 В.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 105  |





Умова  $R \leq R_{3H}$  виконується ( $3,53 \leq 4$ ).

Остаточо отримали: кількість вертикальних електродів дорівнює 8 при  $R = 3,53$  Ом.

### **Висновки до розділу**

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

З цих міркувань було здійснено аналіз приміщення, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи.

Тільки повна усвідомленість працівника про можливі небезпеки, що можуть підстерігати його на робочому місці та дотримання вимог нормативних актів з питань охорони праці та відповідних рекомендацій фахівців, дозволять значною мірою знизити негативний вплив шкідливих та небезпечних факторів при роботі з комп'ютером на організм людини.

Виконано розрахунок захисного штучного заземлення, як одного з ключових факторів безпеки програміста.

|      |      |          |        |      |                                  |            |
|------|------|----------|--------|------|----------------------------------|------------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк.       |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | <b>108</b> |

## 9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи автоматичного прийняття рішення для імітації діяльності людини.

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 автоматичного прийняття рішення для імітації діяльності людини.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем автоматичного прийняття рішення для імітації діяльності людини.
- Досліджена система автоматичного прийняття рішення для імітації діяльності людини.
- На основі отриманих результатів досліджень створена програмна реалізація системи автоматичного прийняття рішення для імітації діяльності людини.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання автоматичного прийняття рішення для імітації діяльності людини.

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.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 109  |

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

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

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

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм Sinople.

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 IT-проєкту, що дозволило визначити ключові фактори успіху даного проєкту.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 110  |

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бурейко Д.Г. Дослідження та програмна реалізація системи автоматичного прийняття рішення для імітації діяльності людини // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2024.
2. Knuth D. The Art of Computer Programming, Vol. 1: Fundamental Algorithms, 3rd Edition 3rd Edition. – Addison-Wesley Professional, 2019. – 672 p.
3. Mattmann C. Machine Learning with TensorFlow, Second Edition. – Manning, 2020. – 1124 p.
4. Mueller J.P., Massaron L. Machine Learning For Dummies. – Wiley, 2016. – 714 p.
5. Teofili T. Deep Learning for Search. – Manning, 2019. – 695 p.
6. Rungta K. TensorFlow in 1 Day: Make your own Neural Network. – Publishdrive, 2019. – 587 p.
7. Weidman S. Deep Learning from Scratch: Building with Python from First Principles. – O'Reilly. 2019. – 252 p.
8. Rajasekaran S., Vijayalakshmi Pai G.A. Neural networks, fuzzy logic, and genetic algorithms: synthesis and applications (with cd-rom) Kindle Edition. – PHI, 2013. – 628 p.
9. Lakhno, V., Malyukov, V., Smirnov, O., Bebeshko, B., Chubaievskiy, V., Zhumadilova, M., Malyukova, I., Smirnov, S. «Multifactorial Model for Targeted Attacks Counteracting Within the Framework of a Multi-Step Quality Game with Fuzzy Information». *ISI 2023. Smart Innovation, Systems and Technologies*, 2024. vol 389. pp 377–389. Springer, Singapore.
10. 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.
11. Smirnov O., Fedorov E., Neskorodieva A., Neskorodieva T.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 111  |

«Intellectual Classification method of Gymnastic Elements Based on Combinations of Descriptive and Generative Approache». *CEUR Workshop Proceedings Volume 3664*, 2024, Pages 11-23.

12. Malyukov V., Bebeshko B., Lakhno V., Smirnov O., Malyukova I., Mohylnyi H. «Managing the Purchase-Sale Process of Digital Currencies Under Fuzzy Conditions». *Lecture Notes in Networks and Systems*, 2023, 729 LNNS, pp. 104–112.

13. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.

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

15. Smirnov, O., Karapetyan, A., Fedorov, E., «Creating Neural Network and Single Solution Human-Based Metaheuristic Methods of Solving the Traveling Salesman Problem». *CEUR Workshop Proceedings*, Volume 3312, 2022, pp. 47-58.

16. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» *CEUR Workshop Proceedings*, Volume 3187, 2022, pp. 1-12.

17. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». *Sensors (Basel, Switzerland)* Volume 22, Issue 16, 6223, 2022.

18. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebeshko, B. «Selection of a Rational Composition of Information Protection Means Using a Genetic Algorithm». In: Rajakumar, G., Du, KL., Vuppapalapati, C., Beligiannis, G.N. (eds) *Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies*, vol 131. 2023. Springer, Singapore. pp. 21-34.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>BKPM-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 112  |

19. Kuznetsov, A., Oleshko, I., Chernov, K., Bagmut, M., Smirnova, T. «Biometric authentication using convolutional neural networks». Lecture Notes in Networks and Systems. Volume 152, 2021, Pages 85-98.

20. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. <https://doi.org/10.1007/s42979-021-00739-w>.

21. Smirnov O., Neskorođieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207.

22. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

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

24. 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.

25. 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.

26. 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.

27. 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.

28. 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.

29. 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.

30. 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.

31. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». CEUR Workshop Proceedings, Vol 2588, P. 215-227, 2019.

32. 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.

33. 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.

34. 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.

35. 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.

36. 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.

37. Smirnov, S., Bulekbaeva, G., Kikvidze, O.G., Lakhno, V., Brzhanov, R., Tabylov, A. «Computer simulation in the MathCAD package of plastic deformation of the deposited layer on the flat surface of the part». Journal of Theoretical and Applied Information Technology Volume 97, Issue 20, 2019, Pages 2467-2484. (Scopus).

38. 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.

39. 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.

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

41. Смірнов, О.А., Усік П.С., Полігенько О.О., Одарченко Р.С.,

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 115  |

Терещенко Л.Ю. «Інформаційна технологія та програмне забезпечення для підвищення ефективності планування підсистеми базових станцій стільникового зв'язку». Проблеми телекомунікацій. № 1(26). С. 83-96. 2020.

42. 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.

43. 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.

44. Смірнов, С.А., Смірнова, Т.В., Минайленко, Р.М., Доренський, О.П., Сисоєнко С.В. «Хмарна автоматизована система інтелектуальної підтримки прийняття рішень для технологічних процесів». Вісник Черкаського державного технологічного університету. Технічні науки. №4, 2020, С. 84-92.

45. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». Центральнуукраїнський науковий вісник. Технічні науки. № 2(33). с. 161-172, 2019.

46. 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).

47. О.А. Смірнов, Т.В. Смірнова, О.М. Дреєв, Є.К. Солових, «Методи оптимізації технологічних процесів відновлення сталевих покриттів», Shipbuilding & marine infrastructure / Суднобудування і морська інфраструктура № 1 (11). с. 48-57, 2019.

48. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 116  |

відновлення та зміцнення поверхонь деталей. Центральнуукраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.

49. Смірнова Т.В., Дреєв О.М., Смірнов О.А. Експертна система оптимізації процесу відновлення та зміцнення поверхонь деталей типу «вал» електродуговим напиленням. Системи управління, навігації та зв'язку, № 2 (54). С. 149-154, 2019.

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., Лисенко І.А. Інформаційна технологія проектування тестових наборів з урахуванням вимог до програмного забезпечення. Системи управління, навігації та зв'язку. – Випуск 4 (44). – Полтава: ПолтНТУ. – 2017. – С. 112-115.

52. Смірнов О.А., Коваленко О.В. Використання псевдобулевих методів бівалентного програмування для управління ризиками розробки програмного забезпечення. Системи управління, навігації та зв'язку. – Випуск 1 (37). – Полтава: ПолтНТУ. – 2016. – С. 98-103

53. Смірнов О.А., Лисенко І.А. Формалізація процесу проектування тестових наборів. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 3 (48). – Харків: ХУПС. – 2016. – С.96-100.

54. Смірнов О.А., Лисенко І.А. Удосконалення методу перевірки коректності таблиць рішень для подання тестових наборів. Збірник наукових праць "Системи обробки інформації". – Випуск 8 (145). – Х.: ХУПС – 2016. – С. 77-80.

55. Смірнов О.А., Лисенко І.А. Розробка впорядкованих каскадних таблиць рішень із використанням матриць слідування. Збірник наукових праць "Системи обробки інформації". – Випуск 6 (143). – Х.: ХУПС – 2016. – С. 216-220.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРМ-122.24.0001.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 117  |

КБПЗ - 2024

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРМ-122.24.0001.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 118  |

Додаток А  
(обов'язковий)

Технічне завдання

Зміст

|   |   |
|---|---|
| 1 Найменування та область застосування.....               | 2 |
| 2 Підстава для розробки.....                              | 2 |
| 3 Мета та призначення розробки.....                       | 2 |
| 4 Джерела розробки.....                                   | 2 |
| 5 Технічні вимоги.....                                    | 2 |
| 5.1 Вміст проекту.....                                    | 2 |
| 5.2 Показники призначення.....                            | 3 |
| 5.3 Вимоги до функціональних характеристик.....           | 3 |
| 5.4 Вимоги до архітектури.....                            | 3 |
| 5.5 Вимоги до надійності.....                             | 3 |
| 5.6 Умови експлуатації.....                               | 4 |
| 5.7 Вимоги до складу та параметрів технічних засобів..... | 4 |
| 5.8 Вимоги до інформаційної і програмної сумісності.....  | 4 |
| 5.8.1 Обладнання.....                                     | 4 |
| 5.8.2 Мова програмування.....                             | 4 |
| 5.8.3 Вхідні дані.....                                    | 5 |
| 5.8.4 Вихідні дані.....                                   | 5 |
| 6 Вимоги до програмної документації.....                  | 5 |
| 7 Економічні вимоги.....                                  | 5 |
| 8 Вимоги щодо охорони праці.....                          | 5 |
| 9 Перелік документів, що розробляються.....               | 6 |
| 10 Етапи розробки.....                                    | 6 |
| 11 Порядок контролю та приймання.....                     | 6 |

|           |                |             |        |      |   |      |       |         |
|-----------|----------------|-------------|--------|------|---|------|-------|---------|
|           |                |             |        |      | <b>ВКРМ-122.24.0001.00.00.ТЗ</b>  |      |       |         |
| Вим.      | Арк.           | № документа | Підпис | Дата |   |      |       |         |
| Розробив  | Бурейко Д.Г.   |             |        |      | <i>Дослідження та програмна<br/>реалізація системи<br/>автоматичного прийняття<br/>рішення для імітації діяльності<br/>людини</i> | Літ. | Аркуш | Аркушів |
| Перевірів | Смірнова Т.В.  |             |        |      |   | М    | 1     | 6       |
| Н. Контр. | Коваленко А.С. |             |        |      | ЦНТУ КН-23Мз  |      |       |         |
| Затв.     | Смірнов О.А.   |             |        |      |   |      |       |         |

## 1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи автоматичного прийняття рішення для імітації діяльності людини.

## 2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 20-13 від 07.08.2024 року).

## 3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи автоматичного прийняття рішення для імітації діяльності людини.

## 4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

## 5 Технічні вимоги

### 5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

|      |      |             |        |      |                           |      |
|------|------|-------------|--------|------|---------------------------|------|
|      |      |             |        |      | ВКРМ-122.24.0001.00.00.ТЗ | Арк. |
| Вим. | Арк. | № документа | Підпис | Дата |                           | 2    |

- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;
- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

## 5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи автоматичного прийняття рішення для імітації діяльності людини;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

## 5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

## 5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

## 5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

|      |      |             |        |      |                                  |      |
|------|------|-------------|--------|------|----------------------------------|------|
|      |      |             |        |      | <b>ВКРМ-122.24.0001.00.00.ТЗ</b> | Арк. |
| Вим. | Арк. | № документа | Підпис | Дата |                                  | 3    |

## 5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

## 5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

## 5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

### 5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

### 5.8.2 Мова програмування

Середовище Python.

|      |      |             |        |      |                           |      |
|------|------|-------------|--------|------|---------------------------|------|
|      |      |             |        |      | ВКРМ-122.24.0001.00.00.ТЗ | Арк. |
| Вим. | Арк. | № документа | Підпис | Дата |                           | 2    |

### 5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

### 5.8.4 Вихідні дані

Робоча програма.

## 6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

## 7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати маркетингове та економічне обґрунтування ІТ-проєкту з урахуванням цін на 3 вересня 2024 року.

## 8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинна бути розглянута характеристика умов праці програміста.

|      |      |             |        |      |                           |      |
|------|------|-------------|--------|------|---------------------------|------|
|      |      |             |        |      | ВКРМ-122.24.0001.00.00.ТЗ | Арк. |
| Вим. | Арк. | № документа | Підпис | Дата |                           | 5    |

## 9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 118 аркушів.

## 10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

10.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Маркетингове та економічне обґрунтування ІТ-проєкту.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

## 11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 02.12.2024 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 18.12.2024 р.

|      |      |             |        |      |                                  |      |
|------|------|-------------|--------|------|----------------------------------|------|
|      |      |             |        |      | <b>ВКРМ-122.24.0001.00.00.ТЗ</b> | Арк. |
| Вим. | Арк. | № документа | Підпис | Дата |                                  | 6    |

Додаток Б  
(обов'язковий)

**Міністерство освіти і науки України**  
**Центральноукраїнський національний технічний університет**

**ЗАТВЕРДЖУЮ**

Керівник випускної кваліфікаційної роботи за  
другим (магістерським) рівнем вищої освіти

\_\_\_\_\_ Смірнова Т.В.

*Дослідження та програмна реалізація  
системи автоматичного прийняття рішення для імітації діяльності  
людини*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 20

Літера: РП

Кропивницький – 2024 року

## Основна програма

```
import random
import time

# Ініціалізація основних параметрів системи
class HumanActivitySimulation:
    def __init__(self):
        self.energy_level = 100
        self.mood = 'neutral'
        self.task_list = ['eat', 'sleep', 'work', 'exercise', 'relax',
'socialize']
        self.completed_tasks = []

# Функція вибору наступної дії на основі рівня енергії та настрою
    def choose_next_action(self):
        if self.energy_level < 30:
            return 'sleep'
        elif self.mood == 'stressed':
            return 'relax'
        else:
            return random.choice(self.task_list)

# Функція імітації виконання завдання
    def perform_task(self, task):
        if task == 'eat':
            self.eat()
        elif task == 'sleep':
            self.sleep()
        elif task == 'work':
            self.work()
        elif task == 'exercise':
            self.exercise()
        elif task == 'relax':
            self.relax()
        elif task == 'socialize':
            self.socialize()

# Оновлення рівня енергії після виконання кожної дії
    def update_energy(self, change):
        self.energy_level += change
        if self.energy_level > 100:
            self.energy_level = 100
        elif self.energy_level < 0:
            self.energy_level = 0

# Функція харчування
    def eat(self):
        self.update_energy(20)
        self.mood = 'satisfied'
        time.sleep(1)
        self.completed_tasks.append('eat')

# Функція сну
    def sleep(self):
        self.update_energy(50)
        self.mood = 'refreshed'
        time.sleep(1)
        self.completed_tasks.append('sleep')

# Функція роботи
    def work(self):
        self.update_energy(-30)
        self.mood = 'focused' if self.energy_level > 30 else 'tired'
        time.sleep(1)
        self.completed_tasks.append('work')
```

```

# Функція фізичних вправ
def exercise(self):
    self.update_energy(-20)
    self.mood = 'energized'
    time.sleep(1)
    self.completed_tasks.append('exercise')

# Функція релаксації
def relax(self):
    self.update_energy(10)
    self.mood = 'calm'
    time.sleep(1)
    self.completed_tasks.append('relax')

# Функція соціалізації
def socialize(self):
    self.update_energy(-10)
    self.mood = 'happy'
    time.sleep(1)
    self.completed_tasks.append('socialize')

# Функція оцінки поточного стану для прийняття рішення
def evaluate_state(self):
    if self.energy_level > 50 and self.mood == 'happy':
        return 'continue working'
    elif self.energy_level < 30:
        return 'need rest'
    else:
        return 'balanced'

# Основний цикл роботи симуляції
def run_simulation(self):
    for _ in range(10):
        next_task = self.choose_next_action()
        self.perform_task(next_task)
        state_evaluation = self.evaluate_state()
        print(f'Energy: {self.energy_level}, Mood: {self.mood}, State:
{state_evaluation}')
        time.sleep(1)

# Запуск симуляції
simulation = HumanActivitySimulation()
simulation.run_simulation()

# Додаткові функції для більш складної поведінки

# Функція для планування завдань на день
class TaskPlanner:
    def __init__(self):
        self.daily_tasks = []

# Додавання нових завдань
def add_task(self, task):
    self.daily_tasks.append(task)

# Видалення виконаного завдання
def complete_task(self, task):
    if task in self.daily_tasks:
        self.daily_tasks.remove(task)

# Перевірка наявних завдань
def check_tasks(self):
    return self.daily_tasks

# Створення об'єкта планувальника завдань
task_planner = TaskPlanner()
task_planner.add_task('work on project')
task_planner.add_task('call friend')
task_planner.add_task('prepare dinner')

```

```

# Оновлення системи для інтеграції планувальника
class HumanActivityWithPlanner(HumanActivitySimulation):
    def __init__(self):
        super().__init__()
        self.planner = TaskPlanner()

# Виконання завдань з планувальника
    def perform_planned_task(self):
        if self.planner.check_tasks():
            next_planned_task = self.planner.check_tasks()[0]
            self.perform_task(next_planned_task)
            self.planner.complete_task(next_planned_task)
        else:
            self.perform_task(self.choose_next_action())

# Оновлений цикл симуляції з планувальником
    def run_simulation_with_planner(self):
        for _ in range(10):
            self.perform_planned_task()
            state_evaluation = self.evaluate_state()
            print(f'Energy: {self.energy_level}, Mood: {self.mood}, Tasks Left:
{self.planner.check_tasks()}, State: {state_evaluation}')
            time.sleep(1)

# Запуск нової симуляції з планувальником
simulation_with_planner = HumanActivityWithPlanner()
simulation_with_planner.planner.add_task('work')
simulation_with_planner.planner.add_task('exercise')
simulation_with_planner.run_simulation_with_planner()

# Функція для більш точного відстеження часу виконання завдань
class TaskTiming:
    def __init__(self):
        self.task_times = {}

# Початок відліку часу для конкретного завдання
    def start_task(self, task):
        self.task_times[task] = time.time()

# Завершення завдання і підрахунок часу
    def end_task(self, task):
        if task in self.task_times:
            elapsed_time = time.time() - self.task_times[task]
            print(f'Task "{task}" completed in {elapsed_time:.2f} seconds')
            del self.task_times[task]

# Інтеграція системи таймінгу в симуляцію
class HumanActivityWithTiming(HumanActivityWithPlanner):
    def __init__(self):
        super().__init__()
        self.timing = TaskTiming()

# Виконання завдань з таймінгом
    def perform_task_with_timing(self, task):
        self.timing.start_task(task)
        self.perform_task(task)
        self.timing.end_task(task)

# Оновлений цикл симуляції з таймінгом
    def run_simulation_with_timing(self):
        for _ in range(5):
            task = self.choose_next_action()
            self.perform_task_with_timing(task)
            state_evaluation = self.evaluate_state()
            print(f'Energy: {self.energy_level}, Mood: {self.mood}, State:
{state_evaluation}')
            time.sleep(1)

# Запуск симуляції з таймінгом
simulation_with_timing = HumanActivityWithTiming()
simulation_with_timing.run_simulation_with_timing()

```

## Файл ai\_decision\_system.py

```
import random

# Моделювання базової AI системи прийняття рішень
class AIDecisionSystem:
    def __init__(self):
        self.knowledge_base = {
            'energy_low': 'rest',
            'mood_stressed': 'relax',
            'task_important': 'work',
        }

# Функція прийняття рішення на основі поточного стану
def make_decision(self, energy, mood, task_priority):
    if energy < 30:
        return self.knowledge_base['energy_low']
    elif mood == 'stressed':
        return self.knowledge_base['mood_stressed']
    elif task_priority == 'high':
        return self.knowledge_base['task_important']
    else:
        return random.choice(['relax', 'work', 'socialize'])

# Використання AI-системи
ai_system = AIDecisionSystem()
decision = ai_system.make_decision(25, 'neutral', 'low')
print(f"AI-based decision: {decision}")
```

Файл `physiological_sensors.py`

```
# Моделювання фізіологічних датчиків
class PhysiologicalSensors:
    def __init__(self):
        self.heart_rate = 70# нормальний пульс
        self.body_temperature = 36.6# нормальна температура

# Оновлення даних з датчиків
    def update_sensors(self, heart_rate, body_temperature):
        self.heart_rate = heart_rate
        self.body_temperature = body_temperature

# Отримання стану на основі показників
    def get_status(self):
        if self.heart_rate > 100:
            return 'stressed'
        elif self.body_temperature > 37:
            return 'unwell'
        else:
            return 'normal'

# Приклад використання датчиків
sensors = PhysiologicalSensors()
sensors.update_sensors(110, 36.5)
status = sensors.get_status()
print(f"Physiological status: {status}")
```

КБПЗ\_2024

```
# Моделювання системи мотивації
class MotivationSystem:
    def __init__(self):
        self.motivation_level = 50

# Оновлення мотивації
    def update_motivation(self, task_importance, mood):
        if task_importance == 'high':
            self.motivation_level += 20
        elif mood == 'stressed':
            self.motivation_level -= 10
        else:
            self.motivation_level += 5

# Отримання рівня мотивації
    def get_motivation_level(self):
        return self.motivation_level

# Приклад використання системи мотивації
motivation_system = MotivationSystem()
motivation_system.update_motivation('high', 'neutral')
print(f"Motivation level: {motivation_system.get_motivation_level()}")
```

КБПЗ\_2024

## Файл agent\_interaction.py

```
# Моделювання взаємодії з іншими агентами
class Agent:
    def __init__(self, name):
        self.name = name
        self.mood = 'neutral'

# Взаємодія з іншим агентом
    def interact(self, other_agent):
        if self.mood == 'happy':
            print(f"{self.name} has a positive interaction with
{other_agent.name}")
        else:
            print(f"{self.name} has a neutral interaction with
{other_agent.name}")

# Створення агентів і їх взаємодія
agent1 = Agent('Agent 1')
agent2 = Agent('Agent 2')
agent1.interact(agent2)
```

КБПЗ\_2024

## Файл task\_scheduling.py

```
# Моделювання системи оптимізації розкладу
class TaskScheduler:
    def __init__(self):
        self.tasks = {'work': 3, 'exercise': 1, 'socialize': 2}

# Оптимізація завдань за рівнем продуктивності
    def optimize_schedule(self):
        optimized_tasks = sorted(self.tasks.items(), key=lambda x: x[1],
reverse=True)
        return optimized_tasks

# Приклад використання оптимізації розкладу
scheduler = TaskScheduler()
optimized_schedule = scheduler.optimize_schedule()
print(f"Optimized Task Schedule: {optimized_schedule}")
```

КБПЗ\_2024

## Файл learning\_system.py

```
# Моделювання системи навчання
class LearningSystem:
    def __init__(self):
        self.experience = 0

# Оновлення досвіду після виконання завдань
    def update_experience(self, task_success):
        if task_success:
            self.experience += 10
        else:
            self.experience += 5

# Отримання рівня досвіду
    def get_experience(self):
        return self.experience

# Приклад використання системи навчання
learning_system = LearningSystem()
learning_system.update_experience(True)
print(f"Experience: {learning_system.get_experience()}")
```

КБПЗ\_2024

## Файл emotional\_variability.py

```
import random

# Моделювання емоційної варіативності
class EmotionalSystem:
    def __init__(self):
        self.mood = 'neutral'

# Оновлення настрою на основі випадкових подій
    def update_mood(self):
        moods = ['happy', 'sad', 'angry', 'stressed', 'calm']
        self.mood = random.choice(moods)

# Отримання поточного настрою
    def get_mood(self):
        return self.mood

# Приклад використання емоційної системи
emotional_system = EmotionalSystem()
emotional_system.update_mood()
print(f"Current mood: {emotional_system.get_mood()}")
```

КБПЗ\_2024

## Файл multitasking\_system.py

```
# Моделювання системи багатозадачності
class MultitaskingSystem:
    def __init__(self):
        self.tasks = []

# Додавання нового завдання
    def add_task(self, task):
        self.tasks.append(task)

# Виконання всіх завдань одночасно
    def execute_tasks(self):
        for task in self.tasks:
            print(f"Executing {task}")

# Приклад використання системи багатозадачності
multitasking_system = MultitaskingSystem()
multitasking_system.add_task('work')
multitasking_system.add_task('exercise')
multitasking_system.execute_tasks()
```

КБПЗ\_2024

Файл `iot_integration.py`

```
# Моделювання інтеграції з пристроями IoT
class IoTDevice:
    def __init__(self, device_name):
        self.device_name = device_name
        self.status = 'off'

# Включення пристрою
    def turn_on(self):
        self.status = 'on'
        print(f"{self.device_name} is now ON")

# Виключення пристрою
    def turn_off(self):
        self.status = 'off'
        print(f"{self.device_name} is now OFF")

# Приклад використання IoT-пристроїв
iot_device = IoTDevice('Smart Light')
iot_device.turn_on()
iot_device.turn_off()
```

КБПЗ\_2024

## Файл behavior\_settings.py

```
# Моделювання гнучких налаштувань сценаріїв поведінки
class BehaviorSettings:
    def __init__(self):
        self.settings = {
            'work_duration': 60,
            'relax_duration': 30,
            'exercise_intensity': 'medium',
        }

# Оновлення налаштувань
    def update_setting(self, setting, value):
        if setting in self.settings:
            self.settings[setting] = value

# Отримання налаштувань
    def get_settings(self):
        return self.settings

# Приклад використання налаштувань поведінки
behavior_settings = BehaviorSettings
```

КБПЗ\_2024

## Файл self\_learning\_system.py

```
# Система самонавчання через зворотний зв'язок
class SelfLearningSystem:
    def __init__(self):
        self.performance_data = []
        self.learning_rate = 0.1

# Оновлення даних після зворотного зв'язку
    def update_performance(self, feedback_score):
        self.performance_data.append(feedback_score)
        self.adjust_behavior()

# Коригування поведінки на основі даних
    def adjust_behavior(self):
        average_performance = sum(self.performance_data) /
len(self.performance_data)
        print(f"Average performance: {average_performance}")
        if average_performance < 5:
            print("Agent needs to improve performance.")
        else:
            print("Agent is performing well.")

# Приклад використання системи самонавчання
learning_system = SelfLearningSystem()
learning_system.update_performance(4)
learning_system.update_performance(7)
```

КБПЗ\_2024

Файл `advanced_task_planner.py`

```
# Розширений планувальник завдань
class AdvancedTaskPlanner:
    def __init__(self):
        self.tasks = []
        self.priority_map = {'high': 3, 'medium': 2, 'low': 1}

# Додавання завдань з пріоритетом
    def add_task(self, task, priority):
        self.tasks.append((task, self.priority_map[priority]))

# Оптимізація за пріоритетом
    def prioritize_tasks(self):
        self.tasks.sort(key=lambda x: x[1], reverse=True)

# Виконання завдань за пріоритетом
    def execute_tasks(self):
        for task, priority in self.tasks:
            print(f"Executing {task} with priority {priority}")

# Приклад використання розширеного планувальника
task_planner = AdvancedTaskPlanner()
task_planner.add_task('Complete report', 'high')
task_planner.add_task('Exercise', 'medium')
task_planner.prioritize_tasks()
task_planner.execute_tasks()
```

## Файл activity\_simulation.py

```
# Симуляція різних типів діяльності
class ActivitySimulation:
    def __init__(self):
        self.activity_types = ['physical', 'mental', 'creative']

# Симуляція активності за типом
def simulate_activity(self, activity_type):
    if activity_type == 'physical':
        print("Simulating physical activity: Running, lifting weights.")
    elif activity_type == 'mental':
        print("Simulating mental activity: Solving puzzles, programming.")
    elif activity_type == 'creative':
        print("Simulating creative activity: Painting, writing.")
    else:
        print("Unknown activity type.")

# Приклад використання симуляції
activity_simulation = ActivitySimulation()
activity_simulation.simulate_activity('mental')
activity_simulation.simulate_activity('physical')
```

КБПЗ\_2024

## Файл database\_integration.py

```
import sqlite3

# Інтеграція бази даних для збереження прогресу
class ProgressDatabase:
    def __init__(self):
        self.conn = sqlite3.connect('progress.db')
        self.create_table()

# Створення таблиці для зберігання прогресу
def create_table(self):
    self.conn.execute('''CREATE TABLE IF NOT EXISTS progress
                        (task TEXT, status TEXT, time_completed REAL)''')

# Збереження виконаного завдання
def save_task(self, task, status, time_completed):
    self.conn.execute('INSERT INTO progress (task, status, time_completed)
VALUES (?, ?, ?)',
                    (task, status, time_completed))
    self.conn.commit()

# Отримання даних про всі завдання
def fetch_tasks(self):
    cursor = self.conn.execute('SELECT task, status, time_completed FROM
progress')
    return cursor.fetchall()

# Приклад використання бази даних
db = ProgressDatabase()
db.save_task('Complete assignment', 'completed', 2.5)
tasks = db.fetch_tasks()
print(f"Saved tasks: {tasks}")
```

## Файл gui\_behavior\_settings.py

```
import tkinter as tk

# Графічний інтерфейс для налаштувань поведінки
class BehaviorGUI:
    def __init__(self):
        self.window = tk.Tk()
        self.window.title("Behavior Settings")

# Налаштування параметрів
    def create_widgets():

# Створення віджетів
    def create_widgets(self):
        self.work_label = tk.Label(self.window, text="Work Duration (minutes):")
        self.work_label.pack()
        self.work_duration = tk.Entry(self.window)
        self.work_duration.pack()

        self.relax_label = tk.Label(self.window, text="Relax Duration
(minutes):")
        self.relax_label.pack()
        self.relax_duration = tk.Entry(self.window)
        self.relax_duration.pack()

        self.submit_button = tk.Button(self.window, text="Save Settings",
command=self.save_settings)
        self.submit_button.pack()

# Збереження налаштувань
    def save_settings(self):
        work_time = self.work_duration.get()
        relax_time = self.relax_duration.get()
        print(f"Work Duration: {work_time} minutes")
        print(f"Relax Duration: {relax_time} minutes")

# Старт графічного інтерфейсу
app = BehaviorGUI()
app.window.mainloop()
```

## Файд environment\_response.py

```
# Моделювання реакції на зміни в середовищі
class EnvironmentResponse:
    def __init__(self):
        self.current_weather = 'sunny'

# Оновлення стану середовища
    def update_environment(self, new_weather):
        self.current_weather = new_weather
        self.respond_to_environment()

# Реакція на нові умови
    def respond_to_environment(self):
        if self.current_weather == 'rainy':
            print("It's rainy. The agent decides to stay inside and relax.")
        elif self.current_weather == 'sunny':
            print("It's sunny. The agent goes outside for exercise.")
        else:
            print(f"Unknown weather: {self.current_weather}")

# Приклад використання
env_response = EnvironmentResponse()
env_response.update_environment('rainy')
```

КБПЗ\_2024