ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І. І. МЕЧНИКОВА

(повне найменування закладу вищої освіти)

Факультет математики, фізики та інформаційних технологій

(повне найменування факультету)

Кафедра інформаційних технологій

(повна назва кафедри)

Кваліфікаційна робота

на здобуття ступеня вищої освіти «Магістр»

«Формування метрик проекту та оптимізація проектних показників для системи контролю розповсюдження вірусної інфекції»

(тема кваліфікаційної роботи українською мовою)

«Formation of project metrics and optimization of project indicators for the viral infection control system»

(тема кваліфікаційної роботи англійською мовою)

Виконав: здобувач заочної форми навчання

спеціальності 122 Комп'ютерні науки

(код, назва спеціальності)

Освітня програма Комп'ютерні науки

(назва)

Стафік Руслан Тимофійович

(прізвище, ім'я, по-батькові здобувача)

Керівник д.т.н., доцент Великодний С.С. _____

(науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент д.т.н., доцент, професор кафедри кібербезпеки НУ "Одеська юридична академія", Соколов Артем Валерійович

(науковий ступінь, вчене звання, прізвище, ініціали)

Рекомендовано до захисту:
Протокол засідання кафедри
Інформаційних технологій
№ ___ від ___ _____ 2024 р.

Завідувачка кафедри

_____ _____

(підпис) (прізвище,ім'я)

Захищено на засіданні ЕК № _____
протокол № ___ від _____ 2024 р.

Оцінка _____ / _____ / _____

(за національною шкалою/шкалою ECTS/ бали)

Голова ЕК

_____ _____

(підпис) (прізвище,ім'я)

Одеса 2024

АНОТАЦІЯ

У кваліфікаційній роботі досліджується тема «Формування метрик проекту та оптимізація проектних показників для системи контролю розповсюдження вірусної інфекції».

Мета роботи. Для вирішення проблеми контролю за розповсюдженням вірусної інфекції, в рамках теми роботи була запропонована модель інформаційної системи, в якій основним джерелом інформації є сама людина. Перед реалізацією такої інформаційної системи необхідно сформувати проектні метрики, розрахувати та знайти шляхи оптимізації проектних показників.

Інформаційні системи для контролю інфікування відіграють ключову роль у сучасному світі для забезпечення безпеки населення та ефективної боротьби з пандеміями. Вони надають урядам і міжнародним організаціям потужні інструменти для управління кризовими ситуаціями, дозволяючи забезпечити швидке реагування на загрози, мінімізуючи негативні наслідки як для системи охорони здоров’я, так і для суспільства в цілому.

У магістерській роботі розглядаються завдання пандемічного характеру, їх вплив на людство, а також заходи, що здійснюються для захисту людей і цілих держав. За допомогою сучасних методів розробки, а також загальнодоступних компонентів, створено універсальний інструмент, який у майбутньому зможе допомагати людям аналізувати та отримувати важливу інформацію, рятуючи велику кількість життів.

Технологічний прогрес у сфері інформаційних технологій створює передумови для того, щоб такі системи стали невід’ємною частиною майбутніх стратегій у сфері громадського здоров’я та життя людей у будь-який точці планети.

ABSTRACT

In the qualification work, the topic «Formation of project metrics and optimization of project indicators for the viral infection control system» is investigated.

The purpose of the work. To solve the problem of controlling the spread of a viral infection, a model of an information system was proposed as part of the work topic, in which the main source of information is the person himself. Before implementing such an information system, it is necessary to form project metrics, calculate and find ways to optimize project indicators.

Information systems for infection control play a key role in today's world to ensure the safety of the population and the effective fight against pandemics. They provide governments and international organizations with powerful crisis management tools, allowing for rapid response to threats while minimizing negative impacts on both the health care system and society as a whole.

The master's work examines the tasks of a pandemic nature, their impact on humanity, as well as the measures taken to protect people and entire states. With the help of modern development methods, as well as publicly available components, a universal tool has been created, which in the future will be able to help people analyze and obtain important information, saving a large number of lives.

Technological progress in the field of information technologies creates the prerequisites for such systems to become an integral part of future strategies in the field of public health and people's lives in any part of the planet.

ЗМІСТ

Стор.

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

БД – база даних

ВВ – варіант використання

ВООЗ – Всесвітня організації охорони здоров'я

ГІК – графічний інтерфейс користувача

ГРВІ – гостра респіраторно-вірусна інфекція

ДО – діюча особа

ЖЦ – життєвий цикл

ІКТ – інформаційно-комунікаційні технології

ІС – інформаційна система

КРМ – кваліфікаційна робота магістра

ПП – програмний продукт

ПС – програмна система

API – Application Programming Interfaces

CMS – Content Management Systems

COVID-19 – coronavirus disease 2019 (коронавірусна хвороба 2019 р.)

CRM – Client Relationship Management

DHR – Default Hourly Rate

DRY – Don't Repeat Yourself

EC – ECF Constant

ECF – Environmental Complexity Factor

EWE – Estimated Work Effort

EWF – ECF Weight Factor

FCM – Fuzzy C-Means clustering

MERS – Middle East Respiratory Syndrome Coronavirus (близькосхідний коронавірусний респіраторний синдром)

MVC – Model-View-Controller

ORM – Object-Relational Mapper

RGB – Red Green Blue

RPFCM – Random Projections Fuzzy C-Means

RUP – Rational Unified Process

RUP – Rational Unified Process

SARS – Severe Acute Respiratory Syndrome (тяжкий гострий респіраторний синдром)

SEIR – Susceptible – Contact (Exposed) – Infected – Recovered

SEO – Search Engine Optimization

SIR – Susceptible – Infected – Recovered (Сприйнятливий – Інфікований – Одужуючий)

SIRS – Susceptible – Infected – Recovered – Susceptible

SIS – Susceptible – Infected – Susceptible

TC – TCF Constant

TCF – Technical Complexity Factor

TWF – TCF Weight Factor

UCP – Use Case Point

UEV – Unadjusted ECF Value

UML – Unified Modeling Language

UTV – Unadjusted TCF Value

UUCP – Unadjusted Use Case Points

ВСТУП

У наш час розвиток технологій досяг небувалих висот. За допомогою штучного інтелекту виконується імітація людини і аналізується величезна кількість даних, а за допомогою удосконалення рівня медицини можна лікувати захворювання, які раніше вважалися смертельними. Не залишилися без уваги і соціальні сфери життя. Соціальні мережі, блоги, хмарні технології, все це дозволило стерти кордони країн і регіонів, об'єднавши світ в одну інформаційну мережу і давши при цьому великі можливості.

Завдяки передовій комунікації було створено мережі обміну інформацією, торгівлі та транспорту, наближаючи світ один до одного. Проте людство не було готово до всього.

Актуальність роботи. Кілька років тому людство зіткнулося з глобальною загрозою, яка всього за кілька місяців може змінити життя мільярдів людей по всьому світу. Соціальні обмеження, карантин, профілактичні заходи, все це вплинуло на життя людства. За цей час ми зрозуміли, що важливіше не допустити погіршення ситуації, ніж боротися з її наслідками. Було розроблено плани та стратегії, написано сценарії та створено прогнозні моделі, за допомогою яких ми могли б змінити ситуацію на самому початку її виникнення.

Запроваджені заходи та обмеження допомогли знизити рівень загрози, врятувати багато життів і зупинити хвилю хаосу, яка може завдати непоправної шкоди всьому людству. Але, на жаль, створені методики не враховують один із найважливіших аспектів – саму людину.

Досвід боротьби з COVID-19 показав, що використання інформаційних систем (ІС) дозволяє швидко ідентифікувати «гарячі точки» поширення інфекції, тобто регіони або окремі групи населення, де ризик інфікування є найвищим. Це дозволяє урядам ефективніше розподіляти ресурси охорони здоров’я: спрямовувати медичні персонал, обладнання та ліки до найбільш уражених регіонів, тим самим знижуючи навантаження на лікарні та мінімі-

зуючи втрати людських життів. Окрім цього, ІС використовуватися для відстеження ланцюжків контактів, що є критичним для вчасного виявлення потенційних носіїв вірусу та їх ізоляції.

Не менш важливим аспектом є соціальна роль ІС. Громадяни, за допомогою спеціалізованих мобільних додатків або веб-порталів, можуть отримувати актуальну інформацію про ризики інфікування у своєму регіоні, отримувати сповіщення про можливі контакти з хворими та дізнаватися рекомендації щодо подальших дій. Така взаємодія з громадянами сприяє підвищенню обізнаності населення про заходи безпеки, стимулює відповідальність за дотримання карантинних вимог і сприяє зниженню панічних настроїв у суспільстві. Ефективна комунікація між державою та громадянами є важливою умовою для успішної боротьби з пандемією.

Окрім внутрішньодержавного значення, ІС для контролю інфікування коронавірусом може мати міжнародний вплив. Координація дій між різними країнами через обмін даними дозволяє розширити масштаби боротьби з пандемією. Наприклад, міжнародні організації, такі як Всесвітня організації охорони здоров'я (ВООЗ), можуть використовувати дані з національних систем для створення глобальних рекомендацій щодо протидії вірусу. Це допомагає забезпечити узгодженість заходів, запроваджених на міжнародному рівні, зокрема на кордонах, у сфері туризму та транспорту. ІС, що збирають дані про нові штами вірусу та їх поширення, сприяють швидшому виявленню мутацій та розробці ефективних вакцин або терапій для нових варіантів інфекції.

Значна увага повинна приділятися питанням конфіденційності та безпеки даних, які будуть зберігатися та оброблятися ІС. Використання персональних даних громадян, особливо у такому чутливому контексті, потребує забезпечення високого рівня захисту інформації. Зокрема, це стосується як технологічних рішень для запобігання несанкціонованого доступу до даних, так і дотримання міжнародних стандартів конфіденційності, таких як Загальний регламент про захист даних в Європі. Урядові органи повинні розробити

прозору політику щодо використання даних, а громадяни мають бути впевнені у захищеності своєї приватної інформації.

Мета роботи. Для вирішення проблеми контролю за розповсюдженням вірусної інфекції, в рамках теми роботи була запропонована модель інформаційної системи (ІС), в якій основним джерелом інформації є сама людина. Перед реалізацією такої ІС необхідно сформувати проектні метрики, розрахувати та знайти шляхи оптимізації проектних показників.

Завдання на роботу. У даній магістерській роботі розглядаються завдання пандемічного характеру, їх вплив на людство, а також заходи, що здійснюються для захисту людей і цілих держав. За допомогою сучасних методів розробки, а також загальнодоступних компонентів, необхідно буде створити універсальний інструмент, який у майбутньому зможе допомагати людям аналізувати та отримувати важливу інформацію, рятуючи велику кількість життів, а для цього необхідно:

– проаналізувати обрану предметну галузь;

– визначитись з методами прогнозування та аналізу даних;

– здійснити формування та розрахунок показників проекту, включно із характеристикою проекту, складанням діаграми варіантів використання тощо;

– сформувати проектні метрики за оцінкою складності;

– виконати оптимізацію отриманих результатів.

Провівши дослідження існуючих превентивних заходів та аналогів подібних ІС, а також загроз, з якими зіткнулося людство, та їх наслідків, слід зробити висновки та проаналізувати результати роботи.

1 АНАЛІЗ ОБРАНОЇ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Види загроз

На даний момент кількість вірусних та інфекційних захворювань може досягати десятків і навіть сотень. Деякі з них добре вивчені, а інші все ще становлять серйозну небезпеку, в тому числі і смертність. У минулому людство зіткнулося з великою кількістю загроз, які загрожували знищити його або значно скоротити населення. Деякі хвороби не виходили за межі міст чи країн, а інші могли поширюватися далеко за межі початкового вогнища виникнення, ставлячи під загрозу весь світ. Щоб спростити класифікацію таких загроз, людство ввело кілька основних термінів, за допомогою яких можна точно описати ступінь загрози того чи іншого захворювання. Дві основні категорії – це пандемії та епідемії.

Епідемія – поширення інфекційного захворювання серед людей, яке прогресує в часі та просторі, значно перевищує рівень захворюваності, який зазвичай реєструється на даній території, і здатне стати джерелом надзвичайної ситуації. У повсякденному житті універсальним епідеміологічним порогом є захворювання 5% жителів території, а іноді 5% будь-якої соціальної групи. Однак багато департаментів охорони здоров'я розраховують власні епідемічні пороги для поширених захворювань на основі середнього рівня цієї хвороби за багато років. Такі епідемічні пороги можуть становити, наприклад, 1% для конкретного захворювання [1].

Ще один вид загрози – пандемія. Пандемія – надзвичайно сильна епідемія, що характеризується поширенням інфекційної хвороби по всій країні, у сусідніх країнах, а іноді й у багатьох країнах світу (наприклад, холера, грип). Пандемія – це, як правило, захворювання, яке набуло широкого поширення, вражаючи значну частину населення світу, спочатку майже все населення [1].

За даними Всесвітньої організації охорони здоров'я (ВООЗ), пандемія може бути оголошена, коли зростання захворювання є експоненціальним. Це означає, що темпи зростання стрімко зростають, і кількість випадків зростає щодня більше, ніж напередодні. Коли хворобу оголошують пандемією, це не має нічого спільного з вірусологією, популяційним імунітетом чи тяжкістю хвороби. Це означає, що хвороба охоплює широку територію та вражає кілька країн і груп населення.

ВООЗ визначає пандемії, епідемії та ендемічні захворювання на основі швидкості поширення хвороби. Отже, різниця між епідемією та пандемією полягає не в тяжкості хвороби, а в тому, наскільки вона поширилася [2].

Пандемія, на відміну від регіональних епідемій, виходить за межі міжнародних кордонів. Через таке широке географічне охоплення пандемії призводять до великих соціальних руйнувань, економічних втрат і загальних труднощів. Важливо відзначити, що після оголошення епідемія може прийняти статус пандемії. Незважаючи на масштаби, епідемія загалом стримується або очікується, що вона поширюється, поки пандемія є міжнародною та поза будь-яким можливим контролем.

Існує кілька факторів, які можуть призвести до спалаху інфекційних захворювань. Зараження може відбутися через передачу між людьми, тваринами або навіть навколишнім середовищем. Наприклад:

– погодні (тобто сезонні) умови: коклюш, наприклад, виникає навесні, а кір, зазвичай, з’являється взимку;

– штучний вплив (наприклад, вплив хімікатів або радіоактивних речовин): як приклад – мінамата – хвороба, що виникає після впливу ртуті;

– соціальні наслідки таких катастроф, як шторми, землетруси, посухи тощо, можуть призвести до високого рівня передачі захворювань;

– фактори навколишнього середовища, такі як водопостачання, їжа, якість повітря тощо, а також санітарні умови – все це може стимулювати поширення інфекційних захворювань.

Також важливо пам’ятати, що походження захворювання може бути невідомим. Це ускладнює відповідну реакцію за короткий проміжок часу. Ці захворювання можуть бути викликані багатьма факторами, в тому числі:

– невідомий або новомодифікований / створений збудник;

– токсини природного походження;

– невиявлені хімічні викиди, надмірне опромінення іонізуючої радіації, раніше невідоме;

– тощо;

Під час пандемії важливо знати:

– на якій стадії перебуває хвороба;

– ступінь ризику зараження та рівень небезпеки для країни чи всього людства.

Також, маючи чітку градацію розвитку пандемії, можна розробити заходи реагування, які зменшать ризик, а в деяких випадках навіть усунуть загрозу. Для вирішення цієї проблеми ВООЗ у 1999 р. розробила фази пандемії, які описують основні відмінності та ступінь загрози під час поширення захворювання [3].

Ці етапи є універсальними та пропонують всесвітню основу для допомоги країнам у плануванні відповіді на пандемію. Щоб полегшити включення нових пропозицій і методів у поточні плани готовності країни та реагування, ВООЗ створила процес із шести етапів. Для того, щоб бути зрозумілішими, точнішими та базуватися на спостережуваних подіях, класифікація та опис стадій пандемії були змінені в 2005 році.

У результаті стадії 1 – 3 стосуються готовності, включаючи зусилля з нарощування потенціалу та планування відповідей, тоді як кроки з 4 по 6 безпомилково показують, що необхідні відповідні заходи та заходи пом’якшення. Крім того, час після початкової хвилі пандемії планується для допомоги в операціях з відновлення після пандемії.

У більшості країн із достатнім моніторингом рівень пандемічної хвороби знизиться протягом періоду після піку нижче задокументованих пікових

рівнів. Пандемічна активність, схоже, слабшає протягом епохи після піку. Країни повинні бути готові до другої чи навіть третьої хвилі, навіть якщо неясно, чи будуть ще хвилі.

Раніше для опису пандемій використовували хвилі активності, які тривали місяцями. Вирішальним питанням комунікації буде збалансувати ці знання з ризиком нової хвилі після того, як рівень активності захворювання знизиться. Початкове попередження про «тишу» може бути передчасним, оскільки хвилі пандемії можуть відбуватися з інтервалом у кілька місяців.

Після пандемії активність сезонного грипу знову досягне допандемічного рівня. Очікується, що пандемічний вірус діятиме подібно до вірусу сезонного грипу А. Оновлення стратегій готовності до пандемії та відповіді на неї відповідно до поточної ситуації є надзвичайно важливим на даний момент. Може виникнути необхідність у ретельному періоді відновлення та оцінки.

Цей методичний прийом призначений для того, щоб допомогти державам та іншим зацікавленим сторонам передбачити, коли певні обставини вимагатимуть оцінки та вибору відповідного часу для важливих дій. Кожна з фаз після оголошення застосовна на міжнародному рівні, як і в пораді 2005 р. Окремі країни можуть відчувати епідемію в різні періоди, тому на додаток до фази пандемії, яка була офіційно оголошена, країни можуть вибрати додаткові національні відмінності на основі їх унікальних ситуацій. Наприклад, країни можуть забажати взяти до уваги, чи потенційний пандемічний вірус поширює хворобу в межах їхніх власних кордонів, серед сусідів або в сусідніх країнах.

Важливо підкреслити, що віхи були створені як посібник для урядів щодо реалізації ініціатив, а не як епідеміологічний прогноз. Ризик на перших трьох фазах просто невідомий, навіть якщо може існувати невелика кореляція між підвищенням ризику пандемії та останніми фазами. Таким чином, можливі обставини, які підвищують небезпеку пандемії, але насправді її не викликають.

Однак, незважаючи на значне вдосконалення глобальних систем епіднагляду та моніторингу за грипом, усе ще можливо, що початкові спалахи пандемії не будуть виявлені або локалізовані вчасно. Вірус грипу, який, наприклад, може спричинити пандемію, може циркулювати задовго до того, як його виявлять, якщо симптоми слабкі та нечіткі. Тому глобальна фаза може переходити від фази 3 до фаз 5 або 6. Фаза 4 може повернутися до фази 3, якщо зусилля швидкого стримування будуть ефективними.

Епідемія COVID-19 – не єдина хвороба, яка вразила людство у всьому світі. Ось кілька прикладів пандемій у минулому, які назавжди сформували еволюцію епідемій та людського імунітету:

– чума (чорна смерть) (1346 – 1353) ця хвороба спричинила приблизно 25 мільйонів смертей у всьому світі в 14 ст. – це захворювання викликає бактерія під назвою Yersinia pestis (лат.);

– пандемія грипу (1889 – 1890) завдяки новим транспортним маршрутам грип отримав нову можливість поширюватися світом – через кілька місяців після появи грип вилетів далеко за межі США, обійшовши всю земну кулю – навіть без авіаперельотів під час поширення грипу загинуло понад 1 млн людей;

– іспанка (1918 – 1920 рр.) це ще один грип, який спричинив масовий спалах захворювання, створивши нову пандемію, яку в народі називають іспанкою – відразу після Першої світової війни виникла небезпека пандемії, яка спричинила смерть понад 50 млн людей у всьому світі – ця кількість була зареєстрована під час спалаху, причому захворювання тривало лише два роки.

1.2 Поширення COVID-19

Людство пережило багато труднощів, однією з яких стали численні пандемії, які забрали життя мільйонів, якщо не сотень мільйонів людей. З часів перших пандемій медицина пішла далеко вперед і досягла рівня розвитку,

якого раніше не було. Тепер ми можемо не тільки вилікувати чуму, але й відновити органи або навіть створити нові за допомогою клітин людини. Але не до всього наша медицина була готова. На рубежі 2019 – 2020 рр. світ зіткнувся з глобальною загрозою, з'явився новий і раніше невідомий штам коронавірусу під назвою COVID-19.

COVID-19 – це захворювання, яке викликається вірусом SARS-CoV-2. Перше виявлення COVID відбулося в грудні 2019 року в Ухані, Китай. Він належить до сімейства коронавірусів, до якого входять віруси, які спричиняють будь-яке захворювання: від застуди голови чи грудної клітини до більш серйозних, але менш поширених захворювань, у тому числі тяжкого гострого респіраторного синдрому (SARS) і близькосхідного респіраторного синдрому (MERS). Коронавіруси, як і багато інших респіраторних вірусів, швидко поширюються через краплі, які виділяються з рота або носа під час дихання, кашлю, чхання або розмови [1].

SARS-CoV-2, як зазначалося раніше, може поширюватися від людини до людини повітряно-крапельним шляхом, інтимним контактом із хворими людьми, а також, можливо, через фекально-оральний та аерозольний контакт. Нещодавно було продемонстровано, що повітряно-крапельний шлях передачі є дуже вірулентним і є основним способом передачі захворювання.

Цей висновок був зроблений після аналізу тенденції та заходів із пом'якшення наслідків у трьох окремих місцях, які вважаються епіцентрами COVID-19: Ухань, Італія, Китай і Нью-Йорк з 23 січня по 9 травня 2020 року. Важливо, що цей аналіз демонструє що серед використовуваних заходів пом'якшення наслідків, таких як соціальна дистанція та використання масок, різниця з обов'язковим покриттям обличчя та без нього є чинником у визначенні моделей пандемії та розповсюдження хвороби.

Більшість людей, інфікованих SARS-CoV-2 (приблизно 80%), протікають безсимптомно або мають мінімальні симптоми, що, швидше за все, пов'язано з сильною імунною відповіддю, здатною контролювати прогресування захворювання. Є докази того, що ці безсимптомні пацієнти можуть

поширювати SARS-CoV-2 на інших. З іншого боку, в осіб із симптомами можуть прогресувати дедалі сильніші симптоми та, зрештою, смерть. Найкраща стратегія зменшення передачі вірусу та захворювання – уникати контакту з ним. У зв'язку з цим деякі рекомендації включають часте миття рук, уникнення тісного контакту, носіння маски на роті та носі, прикриття під час кашлю та чхання, а також щоденне очищення та дезінфекцію поверхонь, до яких зазвичай торкаються. У цьому аспекті носіння масок у громадських місцях є найефективнішим методом запобігання передачі інфекції між людьми.

Вакцини були створені рекордно швидко. Випробування показали, що 9 листопада ефективність вакцин Pfizer і BioN-Tech була більш ніж на 90%, а вакцина Moderna також була успішною через тиждень, 16 листопада. Через тиждень, 23 листопада, Оксфордський університет і AstraZeneca COVID-19 також продемонстрували успішність. Дельта-варіація була ідентифікована невдовзі в грудні в Індії. Занепокоєння щодо можливої підвищеної передачі цих різновидів, викликане сплеском випадків у деяких країнах, наприклад у Великій Британії, змусило кілька урядів знову посилити карантинні заходи.

Вакцина Pfizer/BioNTech стала першою вакциною проти COVID-19, яка була доступна для використання, коли 31 грудня 2020 року ВООЗ видала першу сертифікацію для екстреного використання. Екстрену валідацію розглядали як крок у правильному напрямку для того, щоб зробити щеплення від COVID-19 широко доступними, що є важливим для припинення епідемії.

З того часу вакцини Moderna та Oxford/AstraZeneca повторно отримали дозвіл на використання, і по всій країні почалися агресивні програми вакцинації. Станом на 27 квітня 2021 року було введено один мільярд доз вакцини проти COVID-19. Щоб стримати пандемію та зупинити подальші спалахи, розповсюдження вакцинації має продовжуватися в усіх країнах (рис. 1.6). Багато спеціалістів вважають, що уроки, отримані з пандемії COVID-19, можуть допомогти нам краще підготуватися до майбутніх спалахів інфекційних захворювань і запобігти можливим пандеміям у майбутньому.

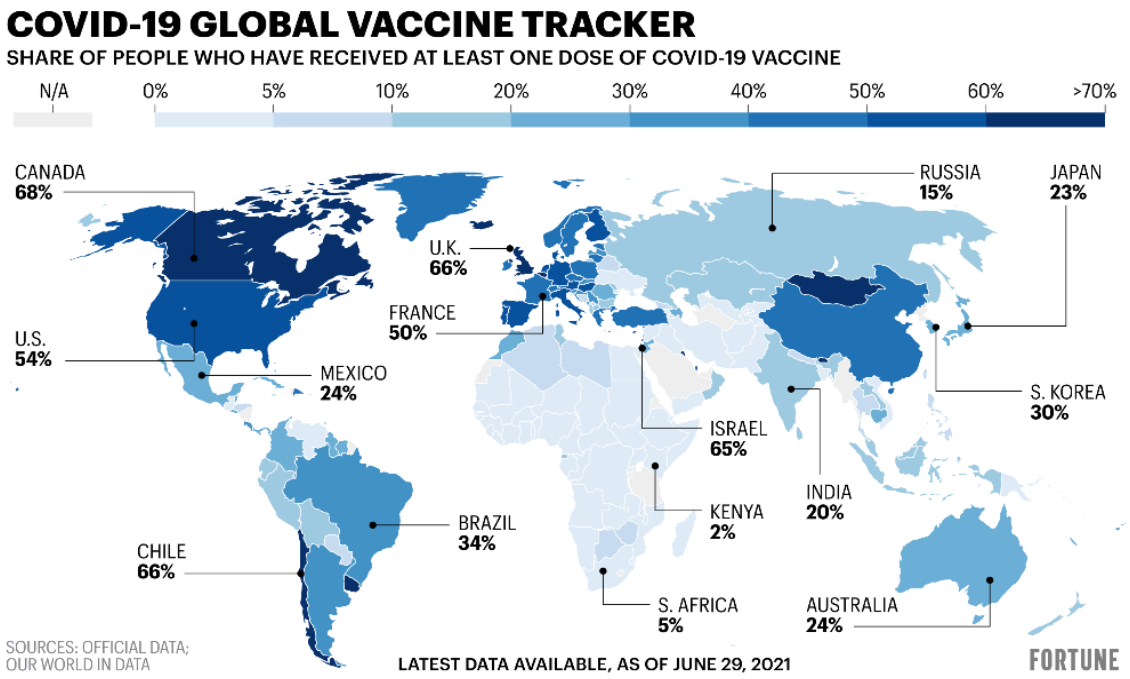

Рисунок 1.6 – Вакцинація людства

Станом на 2024 р. ситуація з Covid-19 стабілізувалася, але вірус залишається відкритим. Кількість інфікованих стрімко падає (рис. 1.7), а кількість смертей активно знижується (рис. 1.8).

Враховуючи динаміку поширення вірусу, можна сказати, що ми успішно подолали пандемічний період і поступово переходимо до рівня епідемії, яка з часом може перерости в ендемію. Проте зменшення загрози від COVID-19 не означає її повну відсутність. Багато вчених впевнені, що ця пандемія не остання і в майбутньому ми можемо зіткнутися не тільки з Covid-19, а й з його різновидами, які вже можуть бути стійкими до вакцин.

За даними ВООЗ, кількість інфікованих людей у світі досягла ~540 мільйонів людей, а кількість смертей зросла до 6 мільйонів. Після вакцинації вдалося суттєво знизити поширення вірусу та знизити кількість випадків захворювання на добу до позначки 300 000 у всьому світі. Країни, які найбільше постраждали від вірусу Covid-19, це Європа, Азія та Америка (табл. 1.1).

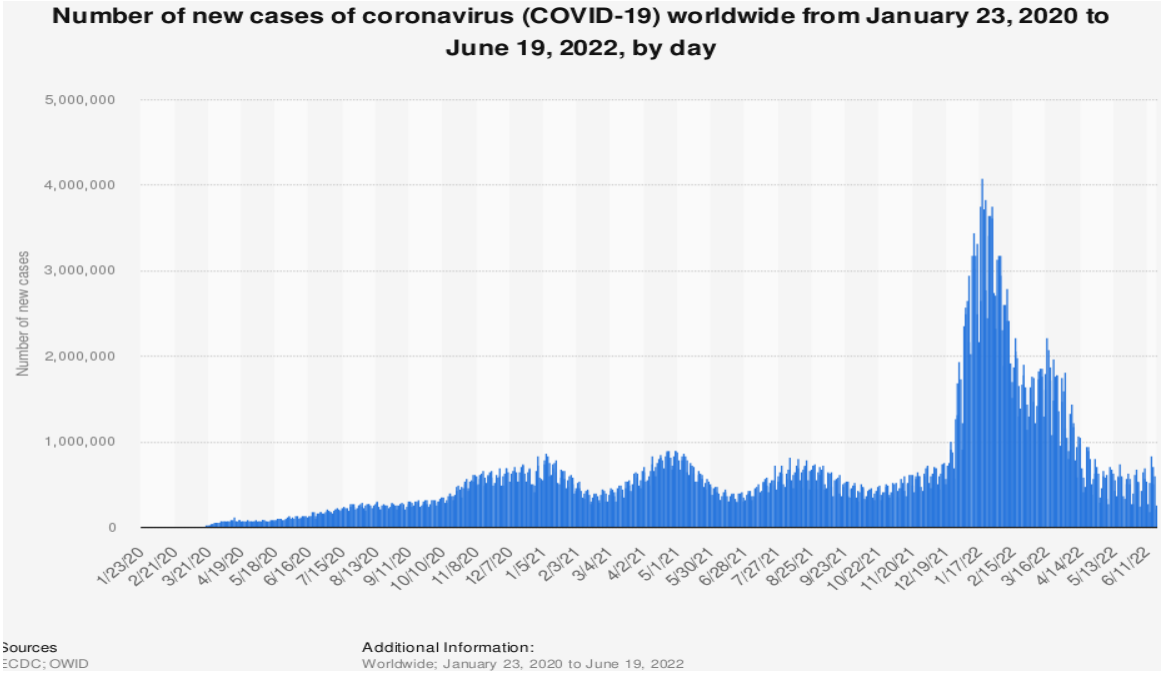

Рисунок 1.7 – Кількість випадків з 2020 по 2022 роки

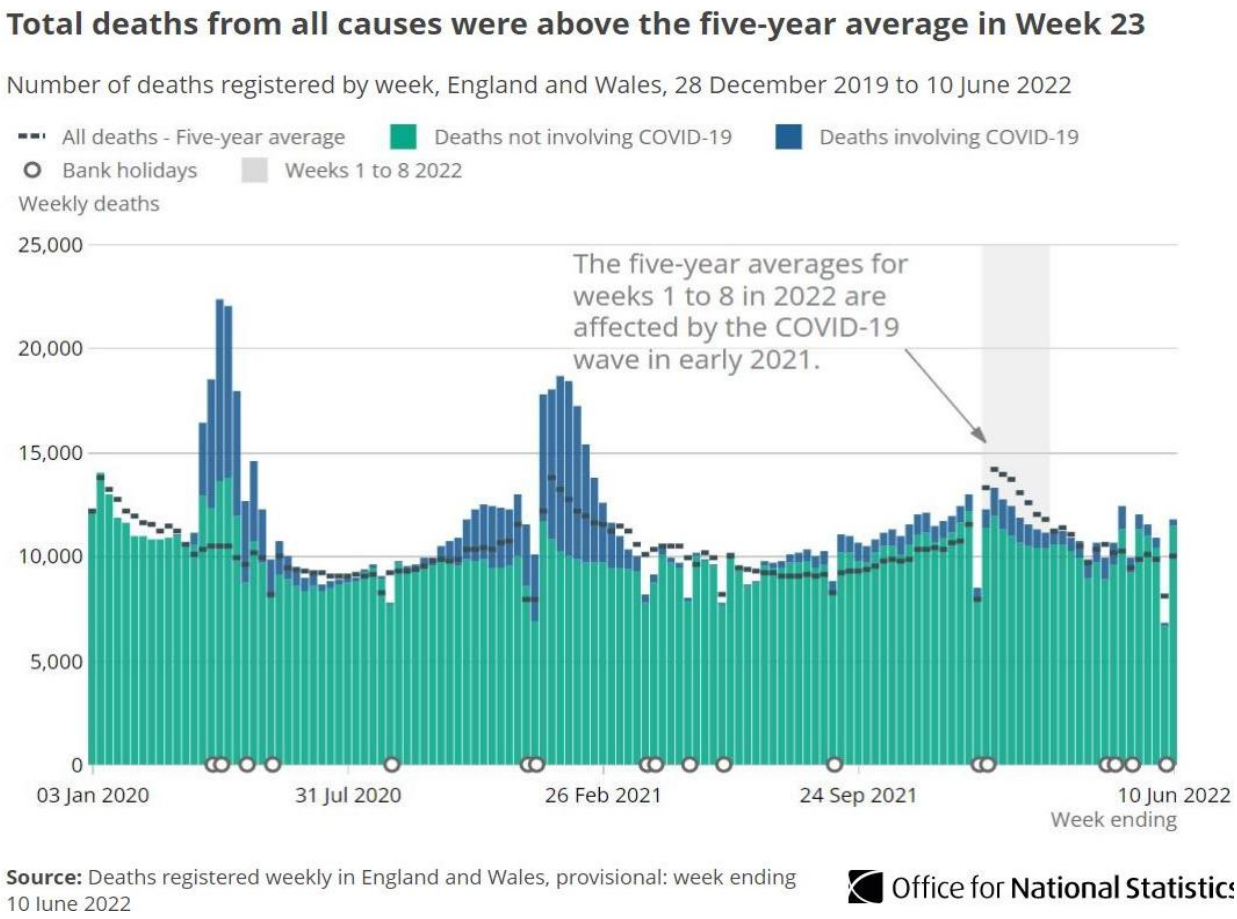

Рисунок 1.8 – Випадки смерті від COVID порівняно з іншими

SARS-Cov-2 зробив те, чого не зміг жоден інший вірус чи бактерія за останні 100 років: замкнув більшість населення світу вдома, паралізував не лише соціальне, а й економічне життя багатьох держав. Це стало науковою загадкою для світової академічної спільноти. Патогенез смертельної хвороби COVID-19, яку вона викликає, досі не розкрито. Але не менш важливо, що SARS-Cov-2 допоміг усім усвідомити, що загроза нової інфекції зараз так само актуальна для людства, як і 100, 200 або 300 років тому.

Таблиця 1.1 – Статистика COVID-19 ВООЗ

Регіон	Всього випадків	Випадків / тиждень	Всього смертей	Смертей / тиждень
глобально	537,591,764	3,493,650	6,319,395	7,685
США	82,263,864	624,815	1,003,740	1,750
Індія	43,319,396	82,701	524,890	113
Бразилія	31,704,193	247,328	669,065	955
Франція	29,165,005	221,196	145,763	254
Німеччина	27,334,993	381,612	140,358	39
Великобританія	22,509,380	81,040	179,601	157
росія	18,403,417	21,047	380,577	440
Республіка Корея	18,289,373	50,329	24,463	73
Італія	17,896,065	232,022	167,780	348
Туреччина	15,085,742	-	98,996	-
Іспанія	12,563,399	12,257	107,482	210
В'єтнам	10,738,161	5,732	43,083	-
Аргентина	9,341,492	28,039	129,016	22
Японія	9,159,940	98,004	31,045	135
Нідерланди	8,132,482	25,549	22,345	9
Україна	5,015,994	-	108,622	-

Більшість найстрашніших епідемій спалахують не через хвороби людей, а через хвороби тварин. Для них існує спеціальний термін – зоонози. Ці хвороби передаються або напряму, тобто при безпосередньому контакті людини з тваринами, або через так звані переносники, тобто якусь додаткову ланку – тварину, яка не хворіє, але є носієм хвороби. Наприклад, для чуми це були блохи, а для коронавірусу – кажани.

Більшість хвороб людського світу вже відомо та навчилися їм протистояти, а наш імунітет здатний з ними боротися. Тваринний світ живе своїм життям, і в нього є свої хвороби, якими тварини хворіють або переносять їх. Існує понад 1,5 мільйона різних вірусів. Людство вивчило лише 4 тисячі.

1.3 Висновки до розділу

Розглянуті загрози, а також запобіжні заходи, які використовуються для їх усунення, показали, що найсильнішою стороною є соціалізація та соціальна відповідальність людини. Застосування профілактичних заходів у вигляді маскового режиму, дистанціювання та карантину хоч і дає хороший результат, але не може повністю виключити людський фактор із факторів впливу. Причиною цього може бути відсутність повного контролю за виконанням таких заходів.

Багато хто, на жаль, не усвідомлюють ступеня важливості та відповідальності, яку вони можуть нести, порушуючи встановлені запобіжні заходи та наражаючи на небезпеку інших. Запровадження автоматичних систем контролю, сертифікації тощо хоч і дають бажаний ефект і допомагають досягти бажаного результату, але не змінюють ставлення людей до таких заходів. Багато хто шукає способи обійти такі обмеження та обдурити систему. Крім того, такі системи не позбавлені багів і недоліків, які можна помітити при уважному розгляді.

2 МЕТОДИ ПРОГНОЗУВАННЯ ТА АНАЛІЗУ ДАНИХ

2.1 Метод прогнозування SIR

Вивчення механізмів розвитку та поширення епідемій є важливим напрямком боротьби з хворобами, поряд з пошуком нових ліків, вакцинацією та профілактичними засобами. На допомогу лікарям прийшли математики – для цього їм довелося поєднати диференціальні рівняння і теорію ймовірностей.

Першу спробу використати математичний апарат для вивчення механізмів поширення хвороб зробив Даніель Бернуллі, який раніше відкрив перші закони гідродинаміки. Наступний крок зробив Вільям Фарр, який у 1840 році застосував нормальний розподіл до аналізу смертей від віспи.

Нарешті, спираючись на роботу великої кількості попередників, британські вчені Андерсон Кермак і Вільям Маккендрік розробили модель SIR, яка широко використовується сьогодні [4]. Ця абревіатура походить від англійських слів Susceptible – Infected – Recovered, що буквально означає «Сприйнятливий – Інфікований – Одужуючий». Під «Сприйнятливими» мають на увазі ще не заражені організми. У рамках цієї моделі за допомогою систем диференціальних рівнянь (за умови безперервності часу та великої популяції) або різницевих рівнянь (за умов дискретного часу та обмеженої популяції) описується динаміка поширення захворювання.

Модель SIR здобула заслужену популярність завдяки простоті конструкції та експлуатації [5]. Її застосування дозволяє точно моделювати епідемії грипу та інших захворювань у великих містах, вводити нові параметри та аналізувати різні сценарії.

Система рівнянь SIR виглядає так:

$$\frac{dS(t)}{dt} = -\frac{\beta IS}{N},$$
$$\frac{dI(t)}{dt} = \frac{\beta IS}{N} - \gamma I,$$

$$\frac{dR(t)}{dt} = \gamma I,$$

де *S(t)* – кількість сприйнятливих осіб у момент часу *t*;

I(t) – кількість інфікованих осіб у момент часу *t*;

R(t) – кількість одужавших осіб у момент часу *t*;

β – коефіцієнт інтенсивності контактів осіб з наступним зараженням;

γ – швидкість одужання інфікованих осіб;

N – сума трьох чисел (чисельність населення).

Перше рівняння системи означає, що зміна кількості здорових (і водночас сприйнятливих до захворювання) осіб зменшується з часом пропорційно кількості контактів із зараженими. Після контакту відбувається зараження, сприйнятливий переходить у стан інфікованого.

Друге рівняння показує, що швидкість збільшення числа інфікованих зростає пропорційно кількості контактів здорових і інфікованих людей і зменшується в міру одужання останніх, а третє рівняння показує, що кількість видужали в одиницю часу пропорційна до кількості інфікованих. Іншими словами, кожна хвора людина через деякий час повинна одужати.

Таким чином, ми бачимо, що хвороба в моделі SIR розвивається за схемою «сприйнятливі заражаються, потім одужують».

Наступне рівняння описує незмінність чисельності популяції (і не враховує смертність від хвороби):

$$\frac{dS}{dt} + \frac{dI}{dt} + \frac{dR}{dt} = 0 \qquad (2.1)$$

Виконаємо побудову графіку за формулою (2.1) для наступних величин (рис. 2.1): *S* = 80 осіб; *I* = 5 осіб; *R* = 0 осіб; β = 0,5; γ = 0,25.

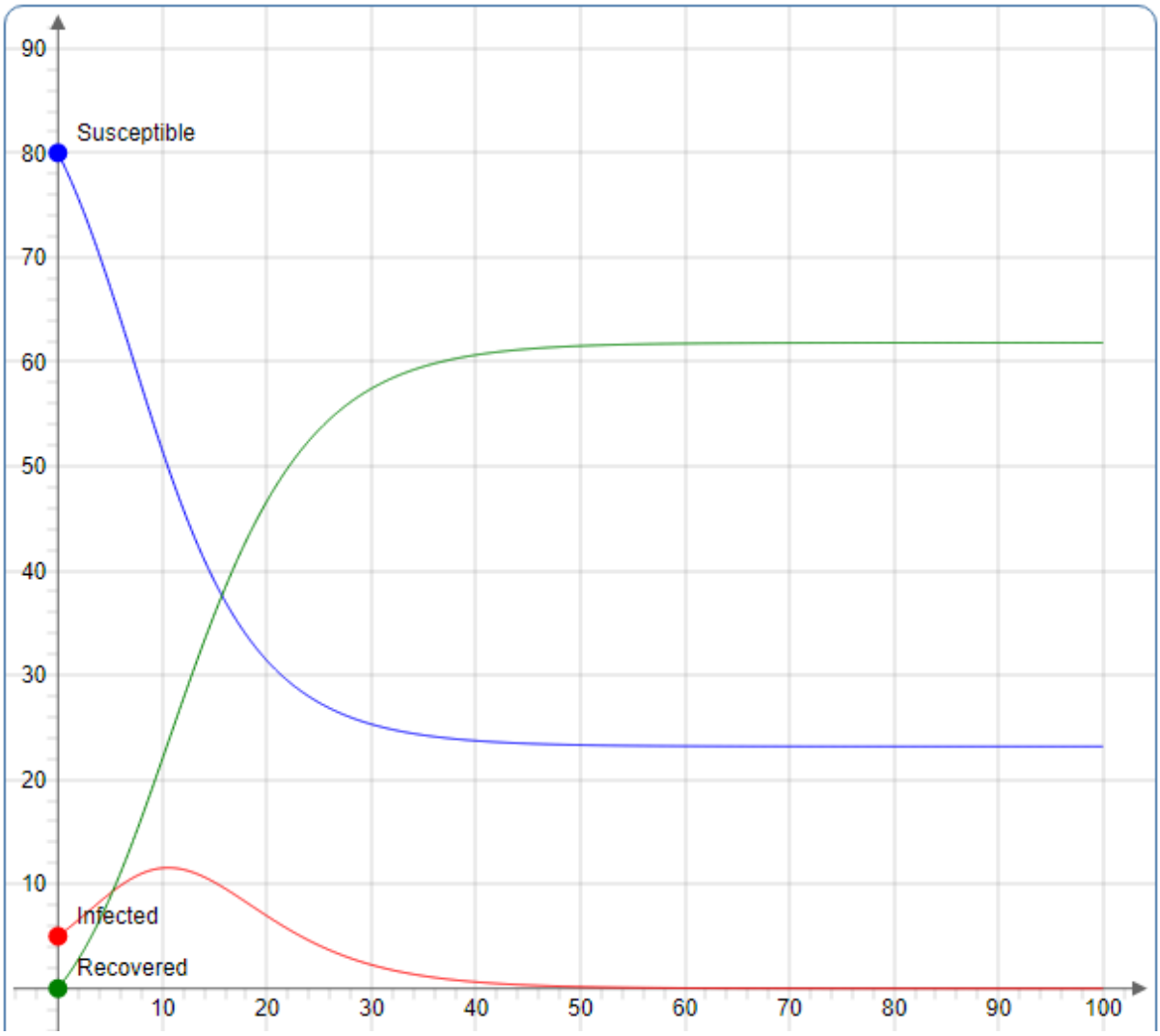

Рисунок 2.1 – Побудований графік для моделі SIR

Тут синя лінія – кількість сприйнятливих осіб, червона лінія – інфікованих, а зелена – одужання. Як ми бачимо на цьому графіку, кількість тих, хто одужав, зростає, знижується кількість сприйнятливих і водночас знижується рівень інфікування. Червоний графік інтенсивності епідемії, що показує кількість одночасно хворих осіб, визначається параметром, який називається «базовим коефіцієнтом відтворення»:

$$R_0 = \frac{\beta}{\gamma}$$

Модель SIR не виправдовує себе, якщо необхідно врахувати неоднорідність популяції (наприклад: різну щільність популяції на різних територіях), різні шляхи передачі інфекції та випадкові фактори, істотні в невеликих популяціях і в початкова фаза поширення захворювання.

Розвитком моделі SIR стали, зокрема, такі моделі [6]:

– SIRS – «Susceptible – Infected – Recovered – Susceptible»: модель для опису динаміки захворювань з тимчасовим імунітетом (одужалі з часом знову стають сприйнятливими);

– SEIR – «Susceptible – Contact (Exposed) – Infected – Recovered»: модель для опису поширення захворювань з інкубаційним періодом;

– SIS – «Susceptible – Infected – Susceptible»: модель поширення хвороби, до якої не вироблено імунітет;

– MSEIR – імунітет матері: модель, яка враховує імунітет дітей, отриманий внутрішньоутробно.

У моделі SEIR (2.4) розвиваються дійсно небезпечні епідемії, оскільки тривалий інкубаційний період може перешкодити своєчасному виявленню захворювання [7]. У цьому випадку існує ризик, що хвороба охопить значну кількість особин популяції. Інфекція розвивається за схемою «Сприйнятливий» – «Контактний» – «Заражений» – «Одужав» і описується системою рівнянь:

$$\frac{dy}{dx} = \mu N - \mu S - \beta \frac{I}{N} S,$$
$$\frac{dE}{dt} = \beta \frac{I}{N} S - (\mu + \alpha) E,$$
$$\frac{dI}{dt} = \alpha E - (\gamma + \mu) I,$$
$$\frac{dy}{dx} = \gamma I - \mu R,$$
$$\frac{dR}{dT} = \gamma I - \mu R$$

де μ – коефіцієнт смертності;

α – величина, зворотна середньому інкубаційному періоду хвороби;

E(t) – кількість особин-носіїв хвороби в момент часу *t*.

Як і в моделі SIR, перше рівняння системи означає, що зміна кількості здорових (і водночас сприйнятливих до захворювання) осіб зменшується з часом пропорційно кількості контактів із інфікованими. Після зараження здорова особина стає контактним захворювачем, або носієм інфекції.

Друге рівняння вводить часову затримку при переході від стану контакту до зараженого стану. Це відбувається через час, рівний інкубаційному періоду захворювання.

Третє рівняння описує перехід від стану «контакт» до стану «заражений», а четверте рівняння показує, що кількість одужалих пацієнтів за одиницю часу пропорційна кількості інфікованих. У цьому випадку в кожному стані індивід може померти, що враховує коефіцієнт μ у кожному рівнянні.

Іншими словами, в кожен момент часу кожна особа з певною ймовірністю може заразитися, через деякий час - захворіти, а потім одужати або померти. Чисельність популяції: $N = S + E + I + R$ не є постійною в часі, а інтенсивність епідемії описує основний рівень відтворення :

$$R_0 = \frac{\alpha}{\mu + \alpha} * \frac{\beta}{\mu + \gamma}$$

Модель SIS або іншими словами «чутливий-інфікований-чутливий» застосовна в аналізі поширення захворювань, до яких не вироблено імунітет, наприклад грипу та гострої респіраторно-вірусної інфекції (ГРВІ). Вона описується такою системою рівнянь:

$$\frac{dS}{dt} = -\frac{\beta SI}{N} + \gamma I,$$
$$\frac{dI}{dt} = \frac{\beta SI}{N} - \gamma I$$

Разом перше і друге рівняння означають, що кількість здорових і хворих людей сумарно не змінюється, а кількість заражень пропорційна кількості контактів між здоровими і хворими людьми.

Друге рівняння описує зміну кількості випадків за одиницю часу, яка пропорційна кількості заражень (кількості контактів здорових та інфікованих осіб) мінус кількість одужань.

Виконаємо побудову графіку для наступних величин (рис. 2.2): S = 80 осіб; I = 5 осіб; R = 0 осіб; β = 0,5; γ = 0,25.

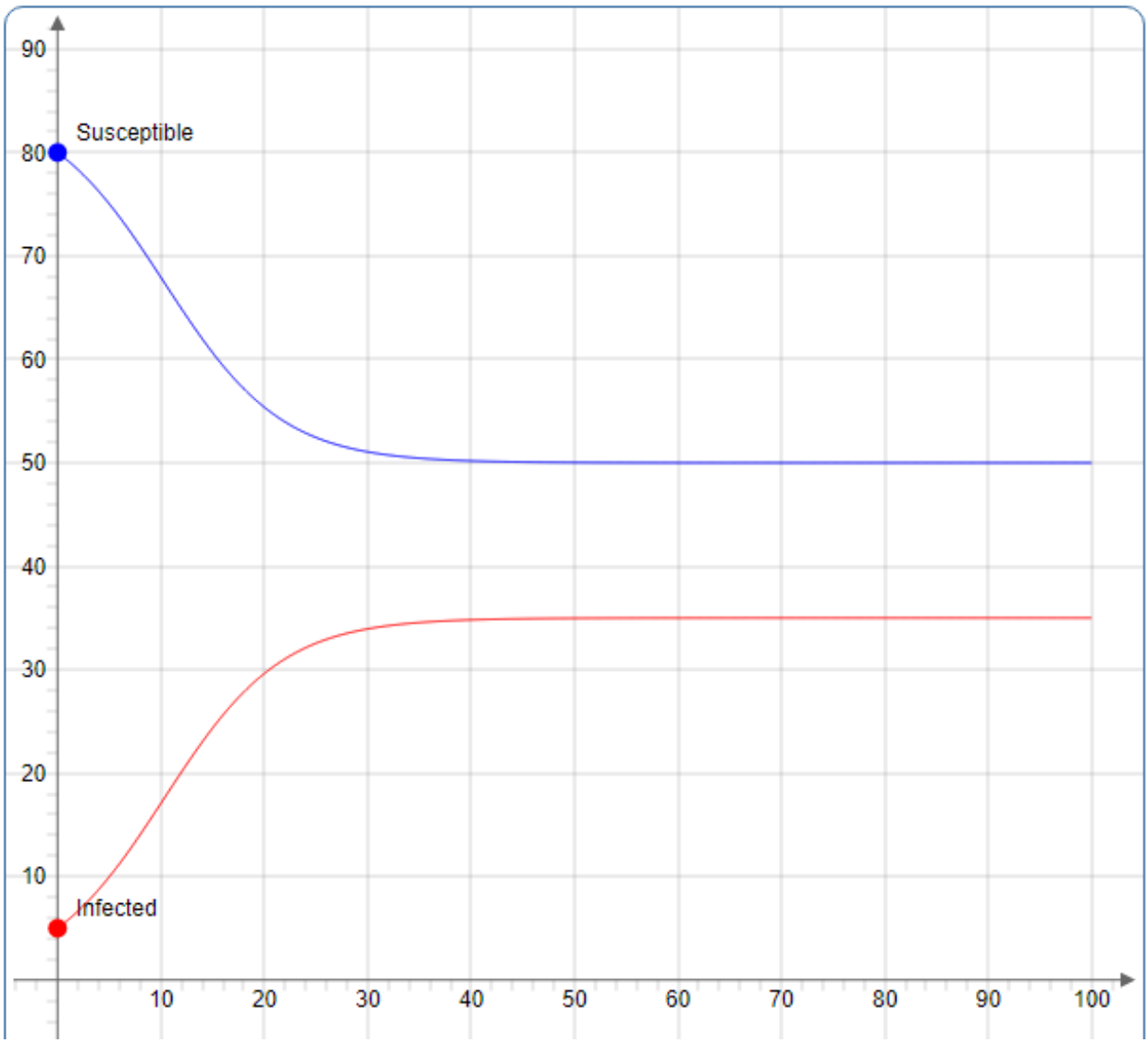

Рисунок 2.2 – Побудований графік для моделі SIS

На цьому графіку блакитна лінія – кількість сприйнятливих осіб, червона лінія зараз інфікована. Як ми бачимо, і сприйнятливі, і заражені збігаються, і якщо деякі з коефіцієнтів будуть рости, то в якийсь момент вони зустрінуться.

2.2 Методи кластеризації

При прогнозуванні результатів важливо не тільки знати певну кількість параметрів даних, а й розуміти, які між ними існують залежності. Наприклад, у разі поширення вірусної інфекції, аналізуючи дані, ми можемо виявити, що інфекція поширюється швидше в густонаселених містах або в місцях з поганою екологією та підвищеним рівнем CO_2. Щоб знайти залежності між даними, ми можемо застосувати методи кластеризації і з їх допомогою не тільки автоматизувати процес, але й отримати більш точний аналіз з огляду на те, що людина може не помітити зв'язку деяких параметрів, які може прийняти алгоритм до уваги. Для аналізу основними методами кластеризації можуть бути методи K-Means і C-Means [8].

2.2.1 Метод K-Means

Метод K-Means упорядковує дані, намагаючись розділити вибірки на *n* груп з рівною дисперсією, одночасно намагаючись мінімізувати показник, який називається інерцією або внутрішньокластерною сумою квадратів. Для цього алгоритму необхідно вказати кількість кластерів. Він застосовувався в багатьох сферах застосування в кількох різних секторах і добре масштабується до величезної кількості зразків (рис. 2.3).

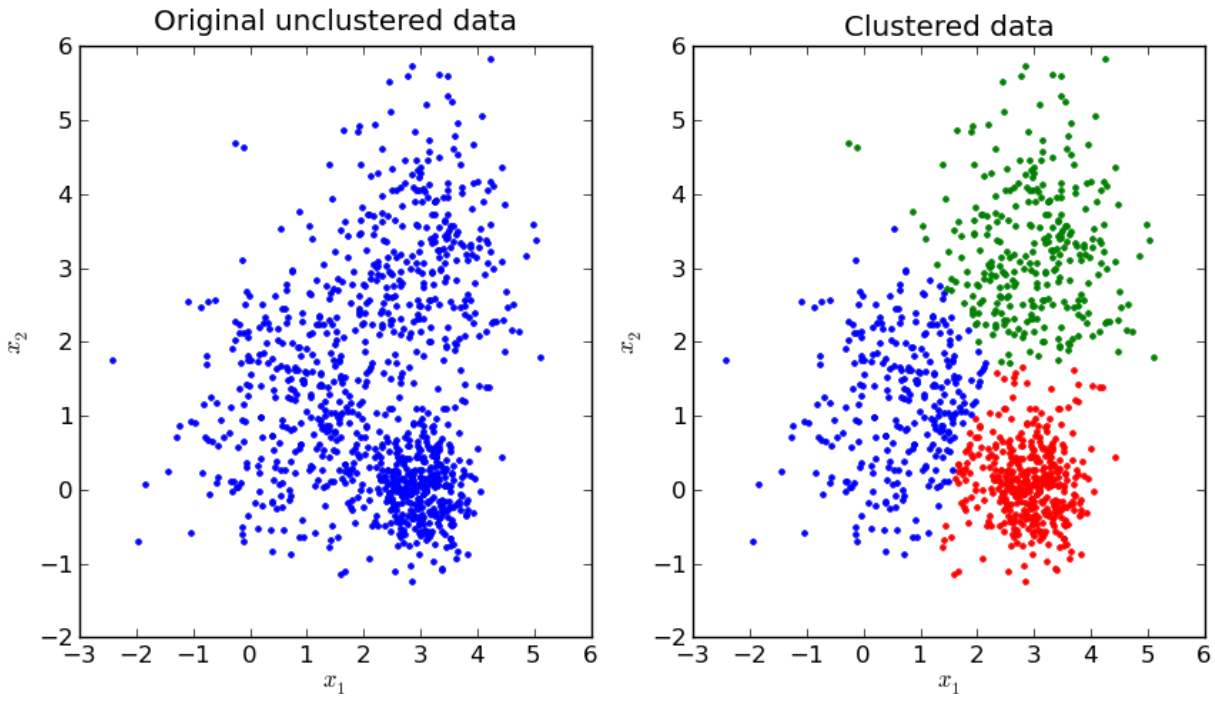

Рисунок 2.3 – Приклади застосування K-Means методу

Набір *N* зразків *X* розділено на *K* непересічних кластерів *C* методом K-Means, і середнє значення кожного кластера служить власним описом μ_j. Середні іноді називають «центроїдами» кластера; зверніть увагу, що, хоча вони знаходяться в тій самій зоні *X*, вони не часто є точками.

Внутрішньокластерний критерій суми квадратів, або інерція, є метою алгоритму K-середніх, який вибирає центроїди:

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_i - \mu_j||^2)$$

Один із способів оцінити, наскільки кластер є внутрішньо когерентним – це його інерція. Він має низку проблем:

– інерція припускає, що кластери завжди опуклі та ізотропні, однак це не обов’язково вірно – він несприятливо реагує на колектори неправильної форми або розширені кластери;

– достатньо розуміння, що кращими є менші значення інфікування, оскільки нуль є ідеальним значенням для інерції, яка не є нормалізованою метрикою. Однак евклідові відстані мають тенденцію збільшуватися в середовищах з надзвичайно високою розмірністю (це приклад «прокляття розмірності»). Цю проблему можна вирішити, а обчислення пришвидшити за допомогою підходу зменшення розмірності, наприклад аналізу головних компонентів перед кластеризацією k-середніх.

Загальна назва K-Means – алгоритм Ллойда [9]. Алгоритм в основному складається з трьох етапів. Перший етап передбачає вибір початкових центроїдів, а найпростішим підходом є вибір зразків із набору даних.

K-means передбачає цикл між двома додатковими фазами після ініціалізації. На першому етапі кожному зразку надається найближчий центроїд. Середнє значення всіх вибірок, призначених кожному попередньому центроїду, використовується на другому етапі для створення нових центроїдів. Алгоритм продовжує останні дві фази, доки різниця між старим і новим центроїдами не стане меншою за заздалегідь визначене порогове значення. Іншими словами, він продовжує рухатися, поки центроїди майже не рухаються.

З невеликою рівною діагональною коваріаційною матрицею K-means ідентичні методу очікування-максимізації. Ідея діаграм Вороного також може бути використана для розуміння алгоритму. Використовуючи наявні центроїди, спочатку обчислюються точки діаграми Вороного.

Діаграма Вороного розбивається на окремі кластери для кожного розділу [10]. Потім центроїди змінюються, щоб відобразити середнє значення для кожної секції. Після цього алгоритм продовжує робити це, доки не буде виконано вимогу зупинки. Коли відносне падіння цільової функції між ітераціями менше, ніж заданий поріг допуску, алгоритм часто припиняє роботу. У цьому підході це не так: ітерація закінчується, коли центроїди відхиляються менше, ніж допуск.

K-means завжди сходитимуться за достатньо часу, хоча це може бути локальний мінімум. На це істотно впливає ініціалізація центроїдів. Як ре-

зультат, алгоритм часто повторюється кілька разів з різними ініціалізаціями центроїдів.

2.2.2 Метод C-Means

Нечіткий метод C-Means – це неконтрольований алгоритм кластеризації, який дозволяє побудувати нечіткий розділ із даних (рис. 2.4). Підхід залежить від параметра *m*, який представляє, наскільки нечітким є результат. Класи будуть заплутані для високих значень *m*, і всі елементи, як правило, належатимуть до всіх кластерів. Параметр *m* впливає на те, як вирішується проблема оптимізації. Іншими словами, різні *m* варіантів, ймовірно, призведуть до різних розділів.

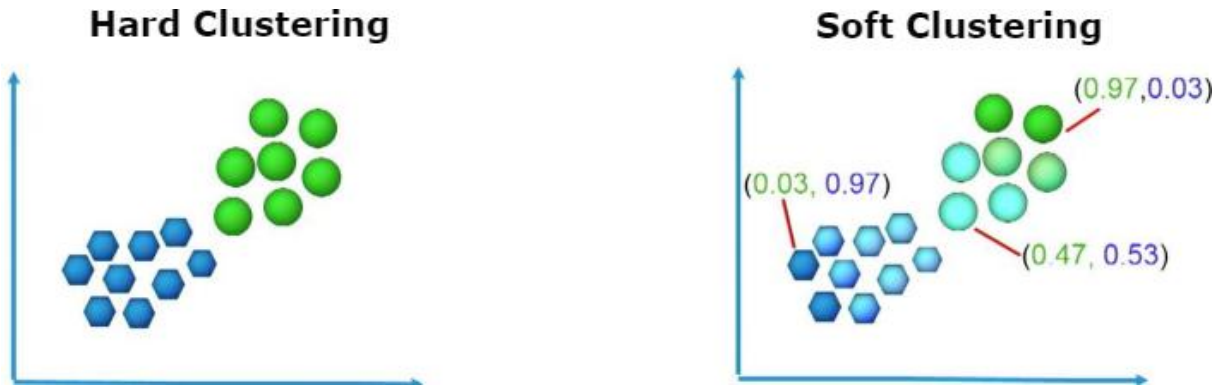

Рисунок 2.4 – Різниця між кластеризаціями за методами K-Means і C-Means

Багатовимірні дані можна кластеризувати за принципами нечіткої логіки, які оцінюють членство кожної точки в кожному центрі кластера від 0 до 10%. Порівняння цього зі звичайним кластеризуванням із жорстким порогом, де кожній точці дається чітка, точна назва, може бути досить ефективним [11]. Відповідно до відстані між центром кластера та точкою даних, цей метод призначає членство кожній точці даних, що відповідає кожному центру кластера. Дані, швидше за все, належать до певного центру кластера, чим

ближче вони до центру кластера. Має бути очевидно, що додавання членства кожної точки даних має дорівнювати одиниці.

На відміну від класичних *k*-середніх, де кожна точка повністю належить лише одному кластеру, нечітка кластеризація призначає кожній точці ймовірність належності до кожного кластера. У кластеризації за методом C-Means кожна точка має вагу, пов'язану з певним кластером; в результаті точка не стільки «в кластері», скільки «пов'язана» з кластером, при цьому ступінь зв'язку визначається зворотною відстанню до центру кластера.

Враховуючи те, що він справді виконує більше роботи, ніж *K*-осіб, нечіткі C-осередки часто працюватимуть повільніше. З кожним кластером переглядається кожна точка, і кожна оцінка передбачає додаткові процеси. У той час як нечіткі C-Means потребують повного оберненого зважування відстані, C-Means потребують лише обчислення відстані. З точки зору витягнутих кластерів, Hard-K-Means є має менше помилок, ніж FCM/Soft-K-Means (FCM – Fuzzy C-Means clustering).

В обробці зображень нечіткі C-Means були ключовим інструментом для групування об'єктів на зображенні. У 1970-х р. математики підвищили точність алгоритму FCM для кластеризації під шумом [12], включивши просторовий термін. Алгоритми FCM також використовувалися для розрізнення різних видів діяльності з використанням характеристик на основі зображення, таких як моменти Ху та Церніке [13]. Крім того, модель нечіткої логіки може бути задана на нечітких наборах, які визначені на трьох компонентах колірного простору HSL: HSL, HSV і LWV.

Функції належності намагаються представити кольори таким чином, щоб імітувати людське сприйняття кольору. Клієнтів можна розділити на більш розмиті кластери в маркетингу залежно від їхніх вимог, уподобань бренду, психографічних профілів або інших критеріїв, пов'язаних з маркетингом.

Розпізнавання образів, виявлення об'єктів і медична візуалізація вже давно використовують сегментацію зображення за допомогою методів клас-

теризації *k*-середніх. Однак традиційна жорстка кластеризація часто не вдається успішно виконати вищезазначені завдання обробки зображень через реальні обмеження, включаючи шум, тіні та зміни в камерах. Як більш ефективний метод для цих робіт запропоновано нечітку кластеризацію. Надається зображення в градаціях сірого, яке було піддано нечіткій кластеризації в Matlab. Зображення в кластері відображається поруч з вихідним зображенням. Три окремі кластери, які використовуються для визначення приналежності кожного пікселя, представлені візуально кольорами.

Різні методи попередньої обробки можуть бути застосовані до фотографій системи RGB (Red Green Blue) залежно від мети, з якою мають використовуватися коефіцієнти нечіткої кластеризації. Перетворення з RGB на HCL є звичайною практикою.

Для нечітких C-Means застосовні більшість показників, які використовуються для оцінки кластерів за іншими методами кластеризації. Незважаючи на те, що ці методи залежать від сфери дослідження експерта, я включив деякі загальні показники для оцінки отриманих кластерів нижче:

– вивчення однорідності кластерів після формування;

– для створення кластерів, які повинні бути однорідними та відрізнятися від інших кластерів, використовувалися нечіткі C-Means;

– для кожного кластера аналіз коефіцієнта варіації;

– якість персонального кластера може бути підтверджена за допомогою кореляції;

– точність, запам'ятовування та f-оцінка також можуть бути враховані, якщо ми знаємо базові значення кластера істинності;

– іншими статистичними інструментами для оцінки ваших кластерів є метод Elbow і Sil-houette;

– техніки, засновані на ентропії.

Нечіткі C-Means, як і самоадаптивні, зазнали кількох змін [14]. Випадкові прогнози нечітких C-Means (RPFCM – Random Projections Fuzzy C-Means) для кластеризації великих даних, алгоритм нечітких C-Means для ви-

значення оптимальної кількості кластерів. Цей підхід посилюється розширеними нечіткими C-Means з методами випадкової вибірки для кластеризації великих даних, нечіткими C-Means++ та нечіткими C-Means з ефективною ініціалізацією заповнення.

2.3 Висновки до розділу

При загрозі пандемії важливо знати не тільки поточну ситуацію в світі чи конкретній країні, а й можливий її розвиток у майбутньому, щоб вжити превентивних заходів і зменшити збитки.

За допомогою методів машинного навчання ми можемо не тільки проаналізувати дані та виділити основні критерії, за якими можна точніше визначити можливі точки зараження та причини їх виникнення, а й спрогнозувати зростання та падіння рівня зараження. поширення інфекції.

За допомогою моделей SIR ми можемо спробувати передбачити, як інфекція поширюватиметься з часом, використовуючи доступні дані про хворих і видужали людей. Використовуючи алгоритми кластеризації, ми можемо ідентифікувати основні ознаки, за якими розрізняємо інфекцію та показуємо її характеристики. За допомогою моделі ARIMA ми можемо використовувати часові ряди, щоб робити більш точні прогнози в часовому масштабі.

Усі ці методи можуть значно допомогти під час розвитку загрози пандемії. Але для їх коректної роботи потрібно зібрати велику кількість даних, частину з яких можуть надати лише медичні установи. Крім того, обчислювальна продуктивність для таких алгоритмів повинна бути досить високою, особливо враховуючи роботу з великими масивами даних. Виходячи з вищесказаного, найкращим рішенням буде використання цих моделей і алгоритмів прогнозування поза інформаційною системою, щоб зменшити навантаження на саму систему і спростити роботу з нею.

Збір даних для аналізу слід передати переважно медичним установам. Звіти користувачів слід використовувати як сукупні загальні дані, які вико-

ристовуються для прогнозування розподілу разом із коригуючими медичними параметрами для визначення ступеня зараження певної області чи міста, іншими словами, дані користувача, зібрані системою, слід використовувати разом із даними з медичних установ, розширюючи і доповнюючи відсутні параметри з обох сторін.

3 РОЗРАХУНОК ПОКАЗНИКІВ ПРОЕКТУ

3.1 Характеристика проекту

Уявлення про складність проекту необхідні для подальшого оцінювання та розрахунку приблизного терміну його реалізації, до якої включаються час на: проектування, тестування, налагодження та деякі інші стадії життєвого циклу (ЖЦ) програмного продукту (ПП). Відповідь на це запитання дає метод Карнера (Karner's Use Case Points Method).

Проект, що розроблюється, являє собою веб-додаток, що буде містити:

– інтерактивну карту місцевості;

– систему авторизації та профілювання;

– систему прогнозування можливого радіусу зараження.

За допомогою інтерактивної карти місцевості користувачі можуть візуально спостерігати за точками зараження, які представляють собою мітки з адресами чи установами, де були виявлені та підтверджені випадки захворювання, а за допомогою округлих зон побачити ймовірний радіус поширення інфекції навколо наявної точки.

При необхідності користувачі можуть залишити свої позначки на карті, які будуть перевірені та підтверджені. Якщо інформація про наявність хворих на точці користувача не підтверджується, ця точка буде видалена з карти та бази даних. У разі виявлення хворих точка буде збережена на карті і їй буде присвоєно статус активної (зміна кольору мітки), в залежності від ступеня загрози конкретної точки, а також джерело отриманих даних.

Мітки на карті можна ранжувати за кольором залежно від її типу, наприклад, жовтий – мітки, залишені користувачами, синій – мітки, залишені установами / медичними установами, червоний – підтверджені випадки зараження. Червоні мітки встановлюються лише для підтверджених і перевірених загроз високого рівня, а також загроз від медичних установ, підключених

до системи. Їх дані не потребують перевірки та додаються на картку без черги.

У системі також реалізовані різні категорії доступу, в залежності від яких надається різний функціонал для роботи з системою:

– гість;

– зареєстрований користувач;

– системний адміністратор.

Гість може увійти у веб-додаток і переглянути активну карту з позначеними точками та їх радіусами. З огляду на те, що користувач не має профілю в системі, єдиною доступною для нього можливістю, крім адресного перегляду тегів, є реєстрація, після якої він стане зареєстрованим користувачем і зможе повноцінно користуватися системою.

Зареєстрований користувач може не тільки відвідувати веб-додаток і переглядати гарячі точки (як у випадку гостя), але й увійти, використовуючи свої облікові дані. Крім того, зареєстрований користувач може поставити на розгляд свої бали в кількості не більше чотирьох за один раз, після чого, у разі підтвердження статусу, вони стануть активними, а обмеження на кількість балів для користувача буде обмежено. У випадку, якщо користувач надасть неправдиву інформацію три (або більше) рази, його профіль буде заблоковано за спробу порушення системи, крім того, він може бути видалений, якщо потрібно. Якщо ця ситуація була випадковою і жодним чином не має причин для навмисної дестабілізації системи – профіль користувача можна повернути назад у доступ до системи.

Системний адміністратор, як і інші користувачі, має доступ до загальної інформації на карті та може її переглядати. Окрім основних функцій, адміністратор має права виправляти, видаляти та додавати точки на карті. Функція виправлення необхідна для зміни адреси точки та її даних, переданих від користувача або якоїсь установи, якщо її адреса або передані дані були введені неправильно. Функції видалення та додавання точок використовуються для ручного керування механізмом роботи з точками на карті та потрібні у

випадках, коли система не може обробити дані про точки в автоматичному режимі (наприклад, у разі помилки в адресі). Це ж стосується випадку, коли поточна точка хибна (або більше не є небезпечною) і її потрібно видалити.

Крім того, адміністратору надається право видаляти та додавати профілі користувачів у системі, якщо потрібно ручне втручання в механізм керування системними профілями (це також може бути корисним, якщо додається новий адміністратор – у цьому випадку лише один із системних адміністраторів матиме права на цю дію).

Функціонал додавання та видалення точок на карті, а також видалення користувачів (у разі спроб порушити роботу системи, шляхом передачі неправдивих даних тощо), а також виправлення точок буде автоматизовано. Однак на поточному етапі проекту (і при першому запуску) цей функціонал буде керуватися вручну, щоб уникнути ситуацій некоректної роботи механізмів системи. Після завершення налаштування та усунення всіх виявлених проблем цей функціонал буде передано під автоматичний контроль модулям системи.

3.2 Складання діаграми варіантів використання

На основі попереднього опису проекту та аналізу літературних джерел за темою роботи [15], [16] було створено наступну діаграму варіантів використання (ДВВ), яка показана на рис. 3.1. Використовуючи цю діаграму, ми можемо візуалізувати основні частини системи, що проєктується, та їх зв’язок один з одним [17].

На основі створеної ДВВ ми можемо надати опис сутностей для забезпечення чіткого розуміння частин системи (табл. 3.1).

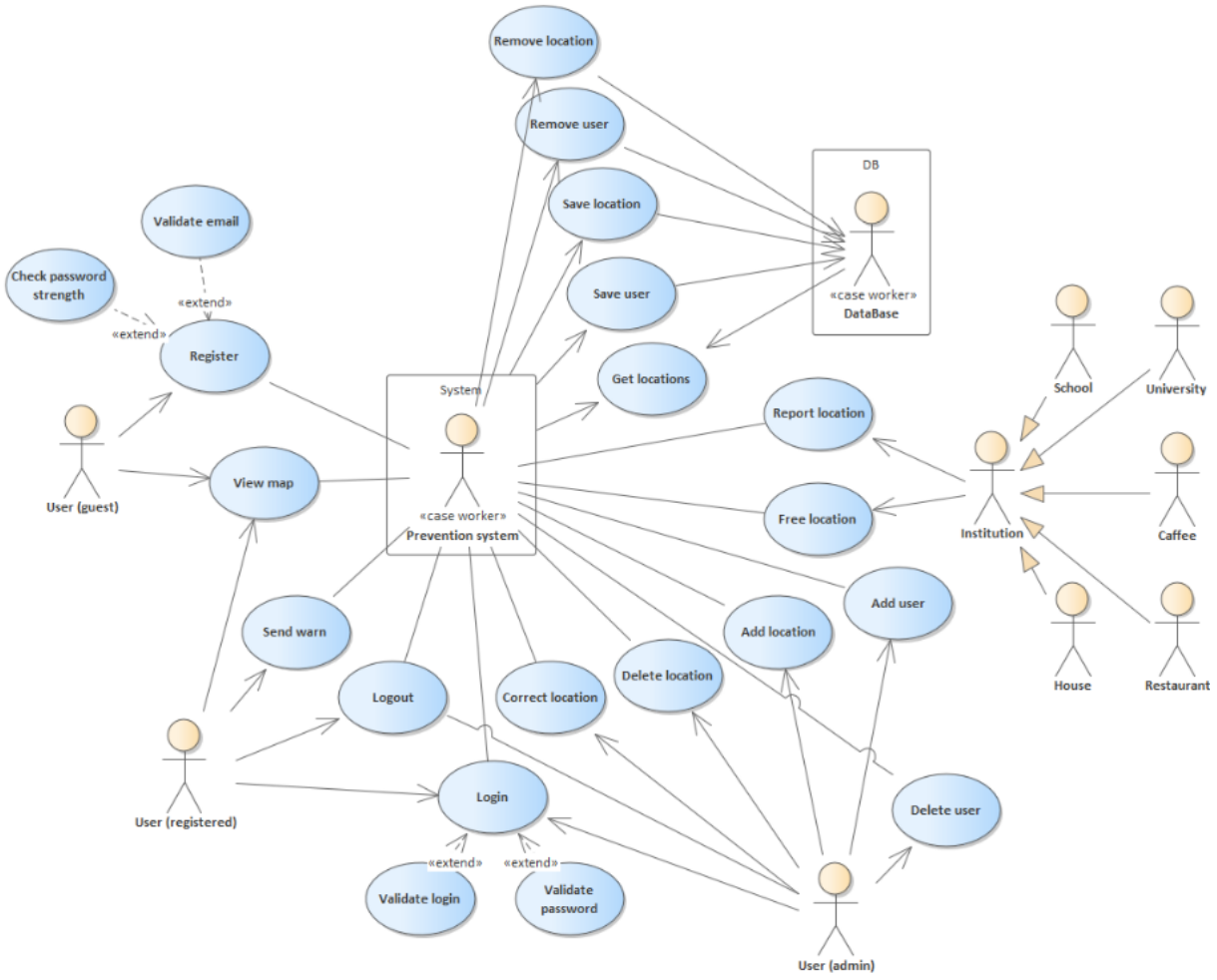

Рисунок 3.1 – Діаграма варіантів використання системи

Таблиця 3.1 – Опис сутностей ДВВ

Сутність	Складність	Пов’язано із:	Опис. Ця функція представляє:
System	Difficult	Many	основні механізми системи, тобто бізнес-логіку програми
Data Base	Difficult	Remove/Set/Get locations, Remove/Save user	механізми роботи бази даних, тобто рівень даних
User (guest)	Easy	Register, View map	незареєстрованого користувача з базовим функціоналом

Продовження таблиці 3.1

Сутність	Складність	Пов’язано із:	Опис. Ця функція представляє:
User (registered)	Easy	View map, Send warn, Logout, Login	зареєстрованого користувача з повним функціоналом системи
User (admin)	Easy	Delete/Add user, Correct/Delete/Add location, Login, Logout	адміністратора системи, що володіє спеціальними функціями на додаток до основних
Institution	Easy	Report/Free location	установи, які нададуть нашій системі дані на додаток до звітів користувачів
Validate email	Easy	Register (extend)	механізм перевірки електронної пошти, який перевіряє її правильність перед реєстрацією
Check password strength	Easy	Register (extend)	механізм перевірки надійності пароля, який можна виконати за допомогою наявних інструментів
Register	Easy	System	механізм реєстрації, можна зробити за допомогою існуючих інструментів профілювання

Продовження таблиці 3.1

Сутність	Складність	Пов’язано із:	Опис. Ця функція представляє:
View map	Medium	System	механізм відображення карти, може мати величезну кількість точок
Send warn	Easy	System	механізм звітності
Logout	Easy	System	механізм виходу із системи
System	Difficult	Many	основні механізми системи, тобто бізнес-логіку програми
Data Base	Difficult	Remove/Set/Get locations, Remove/Save user	механізми роботи бази даних, тобто рівень даних
User (guest)	Easy	Register, View map	незареєстрованого користувача з базовим функціоналом
User (registered)	Easy	View map, Send warn, Logout, Login	зареєстрованого користувача з повним функціоналом системи

3.3 Формування проектних метрик за оцінкою складності

Метод, що орієнтовано на застосування показників Use Case Point (UCP), належить Густаву Карнеру [18]. Багато спеціалістів вважають, що облік видатків [19] на основі методу Карнера дає змогу отримати оцінку із похибкою у 20%, відносно реальних видатків [20], [21], тому саме цей метод візьмемо за основу при виконанні проектної оцінки.

Згідно з пропозиціями методу Карнера, пункти ВВ є функціями наступних аргументів:

а) кількість та складність ВВ в програмній системі (ПС);

б) кількість та складність акторів у ПС;

в) різні нефункціональні вимоги (такі як продуктивність, переносимість тощо, які були не описані у ВВ);

г) середовище розробки (МП, мотивація учасників тощо).

Згідно з [22] методика Карнера пропонує загальну оцінку витрат на розробку проекту, але вона не дозволяє виділити який-небудь його етап. Більш того, методика не може бути використана, доти, поки всі діаграми ВВ не будуть спроектовані [23]. Чинник технічної складності проекту розраховується на підставі відповідних показників технічної складності проекту.

Перед тим, як оцінювати обсяги проекту, необхідно налаштувати управління технічними чинниками та чинниками середовища. Для визначення технічного чинника складності (англ. – Technical Complexity Factor (TCF)) та чинника складності оточуючого середовища (англ. – Environment Complexity Factor (ECF)) треба заповнити перелік чинників, що вплинуть на показники проекту.

3.3.1 Метод проектних точок Карнера та показники нескорегованих варіантів використання

Першим з таких є чинник ваги, що визначається методом використання проектних точок Карнера, проте вони можуть корегуватись відповідно до конкретних вимог проекту.

Наступний чинник – значення, що вказує на ступінь впливу визначеного чинника на проект. Індикацією значення є варіювання від «0» до «5», що означає діапазон управління від «відсутності» до «сильного» із можливими проміжними станами.

Перед тим, як оцінювати ПС за допомогою показників використання функцій, необхідно призначити вагу для кожного з ВВ майбутньої ПС, спираючись на проектні чинники.

Після опису основних елементів системи ми можемо розрахувати значення UUCP (Unadjusted Use Case Points) для цього проекту [24]. Для цього позначимо значення ваги для кожної категорії сутностей, а саме необхідну кількість класів для реалізації, наявність сутностей бази даних і кількість кроків дії (табл. 3.2).

Таблиця 3.2 – Оцінка вагових складності

Рейтинг	Індикатор складності	Інтерфейс користувача	Кількість екз. БД	Кількість класів реалізації	Кількість класів реалізації	Вага, W
1	Easy	Пропонується з шаблону	1	<= 3	< 5	5
2	Medium	З елементами дизайну	>= 2	4 – 7	5 – 10	10
3	Difficult	Індивідуальний дизайн	>= 3	> 7	> 10	15

Виходячи із табл. 3.2, слід призначати відповідну вагу ВВ у тому випадку, коли присутній хоча б один із чинників з максимальним показником для відповідного рангу. Сума показників складності ВВ знаходить своє відображення у показнику не скорегованих точок ВВ (англ. – Unadjusted Use Case Points (UUCP)).

У сценарії використання актором може бути людина, інша програма тощо. Деякі суб'єкти, наприклад система з певним прикладним програмним інтерфейсом (API – Application Programming Interface), мають відносно мінімальні вимоги та лише незначно підвищують складність використання. Деякі сутності такої системи, що працюють через протокол, мають більше вимог та підвищують складність сценарію використання. Інші учасники, такі як користувач, який взаємодіє через графічний інтерфейс користувача (ГІК),

значно впливають на складність сценарію використання. Діючих осіб можна класифікувати як легких, середніх або важких на основі цих відмінностей.

Точки використання, часто відомі як UCP, є методом оцінки програмного забезпечення для оцінки складності розробки програмного забезпечення для проектів розробки програмного забезпечення [25].

Коли проектування та розробка програмного забезпечення проводяться з використанням підходів уніфікованої мови моделювання (UML – Unified Modeling Language) та раціонального уніфікованого процесу, використовується UCP. Варіанти використання, підхід до моделювання з набору UML, є основою ідеї UCP, яка базується на вимогах до системи, що створюється.

Розмір програмного забезпечення оцінюється за допомогою компонентів ДВВ системи з урахуванням чинників середовища та технологічних факторів [26]. Очікувані зусилля для проекту потім можна визначити за допомогою UCP для цього проекту [17]. Значення UUCP має те саме значення, за винятком того, що цей показник не нормалізовано, і його можна назвати дещо грубою оцінкою.

На основі таблиці складності ми можемо призначити значення ваги для елементів системи та знайти значення UUCP.

Таблиця 3.2 – Оцінка ваги за складністю

Назва	Тип сутності	Складність
Remove location	UseCase	5
Remove user	UseCase	5
Add user	UseCase	5
Delete user	UseCase	5
Validate password	UseCase	5
Validate login	UseCase	5
Check password strength	UseCase	5
Free location	UseCase	5

Продовження таблиці 3.2

Назва	Тип сутності	Складність
Report location	UseCase	5
Add location	UseCase	5
Delete location	UseCase	5
Correct location	UseCase	5
Save user	UseCase	5
Logout	UseCase	5
Validate email	UseCase	5
Login	UseCase	5
View map	UseCase	10
Send warn	UseCase	5
Register	UseCase	5
Get locations	UseCase	5
Save location	UseCase	5

На основі отриманих даних можна визначити, що значення метрики UUCP для цього проекту матиме таке значення: загальна складність (UUCP) = 110.

Для кращого розуміння причини призначення відповідної ваги сутностям системи опишемо причини, чому було вирішено встановити значення ваги саме з такими значеннями:

– видалення розташування: складність проста, оскільки кількість класів, які використовуються для цієї функції, дорівнює одному (клас керування розташуваннями), а сценаріїв, які використовуються для цього, дорівнює двом (перевірити наявність + видалити розташування з БД (бази даних));

– видалення користувача: складність проста, тому що кількість класів, які використовуються для цієї функції, дорівнює одному (клас керування користувачами), а сценарії, що використовуються для цього, дорівнюють двом (перевірити існування + видалити користувача з БД);

– збереження розташування: складність проста, тому що кількість класів, які використовуються для цієї функції, дорівнює одному (клас керування розташуваннями), а сценаріїв, які використовуються для цього, дорівнює двом (перевірити наявність + зберегти розташування в БД);

– збереження користувача: складність проста, тому що кількість класів, які використовуються для цієї функції, дорівнює одному (клас керування користувачами), а сценарії, що використовуються для цього, дорівнюють двом (перевірити існування + зберегти користувача в БД);

– отримати розташування: складність проста, оскільки кількість класів, які використовуються для цієї функції, дорівнює одному (клас керування розташуваннями), а сценарії, що використовуються для цього, дорівнює одному (отримати всі розташування з БД);

– звіт про місцезнаходження: складність проста, оскільки кількість класів, які використовуються для цієї функції, дорівнює одному (клас керування місцеположеннями), а сценарії, що використовуються для цього, дорівнює одному (надсилати звіт про місцезнаходження);

– вільне розташування: складність проста, тому що кількість класів, які використовуються для цієї функції, дорівнює одному (клас керування розташуваннями), а сценарії, які використовуються для цього, дорівнюють одному (надсилати запит на звільнення);

– додати користувача: складність проста, тому що кількість класів, які використовуються для цієї функції, дорівнює одному (клас адміністрування), а сценарії, що використовуються для цього, дорівнюють одному (надсилати запит на додавання користувача);

– видалити користувача: складність проста, тому що кількість класів, які використовуються для цієї функції, дорівнює одному (клас адміністрування), а сценарії, що використовуються для цього, дорівнюють одному (надсилати запит на видалення користувача);

– додати розташування: складність проста, оскільки кількість класів, які використовуються для цієї функції, дорівнює одному (клас адміністру-

вання), а сценарії, що використовуються для нього, дорівнюють одному (надсилати запит на додавання розташування);

– видалити місцезнаходження: складність проста, оскільки кількість класів, які використовуються для цієї функції, дорівнює одному (клас адміністрування), а сценарії, що використовуються для цього, дорівнюють одному (надсилати запит на видалення місцезнаходження);

– правильне розташування: складність проста, тому що кількість класів, які використовуються для цієї функції, дорівнює одному (клас адміністрування), а сценарії, що використовуються для нього, дорівнюють одному (надсилати запит на зміну розташування);

– логін: складність проста, тому що кількість класів, які використовуються для цієї функції, дорівнює одному (клас керування розташуваннями), а сценарії, що використовуються для нього, дорівнюють одному (отримати всі розташування);

– вихід із системи: складність проста, оскільки кількість класів, які використовуються для цієї функції, дорівнює одному (клас користувачів), а сценарії, що використовуються для цього, дорівнюють одному (надсилати запит на вхід до системи);

– перевірити вхід: складність проста, оскільки кількість класів, які використовуються для цієї функції, дорівнює одному (клас профілювання), а сценаріїв, які використовуються для нього, дорівнює двом (отримати логіни користувачів з бази даних + перевірити відповідність);

– перевірка пароля: складність проста, тому що кількість класів, які використовуються для цієї функції, дорівнює одному (клас профілювання), а сценаріїв, які використовуються для нього, дорівнює двом (отримати пароль користувача з бази даних + перевірити відповідність);

– надсилати попередження: складність проста, тому що кількість класів, які використовуються для цієї функції, дорівнює одному (клас користувачів), а сценарії, що використовуються для нього, дорівнюють одному (запит на додавання точки надсилання);

– переглянути карту: складність середня, оскільки кількість класів, які використовуються для цієї функції, дорівнює одному (клас користувачів), а сценаріїв, які використовуються для нього, дорівнює двом (отримати всі місця + завантажити карту + додати точку на карту + додати радіус до кожної точки);

– реєстрація: складність проста, тому що кількість класів, які використовуються для цієї функції, дорівнює одному (клас користувачів), а сценаріїв, які використовуються для нього, дорівнює одному (надсилати запит на реєстрацію);

– перевірка електронної пошти: складність проста, оскільки кількість класів, які використовуються для цієї функції, дорівнює одному (клас профілювання), а сценаріїв, які використовуються для нього, дорівнює двом (отримувати електронні листи користувачів від бази даних + перевіряти відповідність);

– перевірити надійність пароля: складність проста, оскільки кількість класів, що використовуються для цієї функції, дорівнює одному (клас профілювання), а сценаріїв, які використовуються для нього, дорівнює одному (перевірте пароль на відповідність шаблону).

На основі отриманих даних можна побачити, що більшість системних об’єктів мають легку складність і лише деякі окремі об’єкти, оскільки БД і карта мають складність вище легкої. Причиною таких високих балів є складність обробки даних та їх кількість. Перегляд карти передбачає нанесення і додавання на карту великої кількості точок, а якщо врахувати розширення охоплення системи в глобальному масштабі, то їх кількість може досягати десятків і навіть сотень тисяч на одній території.

З огляду на це навантаження буде досить високим, що вимагатиме певної оптимізації даних і процесу їх обробки, вимагаючи збільшення витрат ресурсів на цей функціонал. Сама БД має високу складність з огляду на те, що вона є складним об'єктом системи, який є одним із ключових і відіграє важливу роль у роботі системи в цілому.

Таким чинок надійність та стабільність є основними критеріями, з якими потрібно працювати при розробці та оптимізації структури програмного засобу.

3.3.2 Розрахунок чинника технічної складності

Чинник технічної складності (англ.: Technical Complexity Factor – TCF) базується на технічних проблемах, використовується для обчислення метрик проекту (табл. 3.3). Ці елементи конфігурації працюють із факторами складності середовища, щоб спотворювати загальну складність вгору або вниз залежно від рівня технічної складності та пов'язаного з цим рівня допомоги середовища, а їхні додані значення виробляють нескориговане значення TCF.

Наскільки технічну складність ми надаємо, вказує вага TCF. На відміну від «система має бути сценарієм оболонки», «система має бути створена в ADA» матиме більшу вагу. Вагомість оцінює відповідний фактор, але він не впливає на проект. Ми можемо призначити кожному фактору значення, яке показує як кожен фактор впливає на наш проект, від 0 (не має значення) до 5 для кожного чинника під час оцінки проекту, що означає «дуже впливає».

Таблиця 3.3 – Чинники технічної складності проекту

Показник	Опис	Вага	Значення	TCF
TCF01	Розподілена система	2,00	5,00	10,00
TCF02	Цільові показники відгуку або пропускної здатності	1,00	3,00	3,00
TCF03	Ефективність кінцевого користувача (онлайн)	1,00	3,00	3,00
TCF04	Комплексна внутрішня обробка	1,00	5,00	5,00
TCF05	Код має бути придатним для повторного використання	1,00	1,00	1,00

Продовження табл. 3.3

Показник	Опис	Вага	Значення	TCF
TCF06	Легко встановити	0,50	5,00	2,50
TCF07	Easy to use	0,50	4,00	2,00
TCF08	Портативний	2,00	5,00	10,00
TCF09	Легко змінити	1,00	4,00	4,00
TCF10	Одночасний	1,00	4,00	4,00
TCF11	Включає спеціальні функції безпеки	1,00	4,00	4,00
TCF12	Забезпечте прямий доступ для третіх осіб	1,00	4,00	4,00
TCF13	Потрібні спеціальні засоби навчання користувачів	1,00	2,00	2,00
			Усього (UTV):	54,50

Для переконання, що всі значення були призначені правильно, опишемо кожен чинник нашого проекту.

TCF01 вплив на складність проекту має значення, що дорівнює 5, оскільки кількість серверів БД і серверів бізнес-логіки програми може бути більше одного (наприклад, при розширенні зони функціонування системи від місцевого до національного масштабу). Крім того, важливо зазначити, що кількість робочих станцій, які використовують систему, може досягати десятків і сотень одиниць. З цієї причини розробка розподіленої системи та організація її стабільної роботи займає багато часу і ускладнить як сам процес розробки, так і налагодження.

TCF02 має значення впливу на складність проекту рівне 3, оскільки швидкість оновлення ковід-карти не повинна відбуватися щогодини. Достатній інтервал для оновлення інформації повинен бути встановлений від 3 до 6 годин, тому система не вимагає високої продуктивності. Хоча обробка вхід-

них даних може бути обчислювально-інтенсивною, через скорочений час оновлення стає можливим виконувати необхідні операції повільніше, що, у свою чергу, спростить процес розробки механізмів оновлення даних, залишаючи додатковий простір для оптимізації та розширення у майбутньому.

TCF03 має значення впливу на складність проекту рівне 3, з огляду на те, що користувач системи не повинен мати високу швидкість роботи або вміти інтерпретувати отримані дані. Завдяки цьому її ефективність при роботі з системою не є серйозним фактором, що впливає на її роботу в цілому, змінюючи її стан або процеси. Оскільки багато або майже всі механізми роботи з системою максимально спрощені для кінцевого користувача, єдиним фактором, який може вплинути на загальну складність у цьому випадку, може бути лише кількість ресурсів, які будуть витрачені на спрощення роботи з механізмами системи.

TCF04 значення впливу на складність проекту дорівнює 5, з огляду на те, що обсяг оброблених даних, які будуть надходити в систему, може бути величезним і досягати сотень Гб записів. Тому для скорочення часу обробки такого обсягу даних і загальної оптимізації самих процесів обробки буде витрачено велику кількість ресурсів на етапі розробки самих механізмів, що дозволить тому вимагають врахування великої кількості можливих проблем. Через це складність розробки програмного коду, а також його структури стане досить тривалою і трудомісткою, що в свою чергу також збільшить загальну складність розробки самої системи.

TCF05 вплив на складність проекту дорівнює 1, оскільки всі механізми роботи з системою будуть розроблятися окремо один від одного, що не дозволить використовувати один і той же код в різних частинах системи. Завдяки цьому відпадає необхідність проектування універсального коду, що спрощує процес розробки та дозволяє застосовувати лише ті рішення, які підходять для певних механізмів системи. Одна з переваг такого модульного підходу полягає в тому, що кожен з механізмів, що розробляються, можна налаштовувати і налаштовувати індивідуально, враховуючи всі особливості

завдання для кожного конкретного випадку. Крім того, такий підхід зменшить ризик злому системи, оскільки вразливий код використовуватиметься лише в одній частині системи, що, у свою чергу, створює ефект ізоляції, не дозволяючи зловмиснику використати ті самі методи та отримати доступ до усіх функцій системи.

TCF06 вплив на складність проекту дорівнює 5, оскільки ця система розроблена як веб-додаток і при її розробці необхідно буде створити оптимізований код, не вимогливий до ресурсів користувача, який в поворот ускладнить процес його оформлення. Також при розробці універсального веб-додатку для більшості користувальницьких платформ виникне вимога до правильного поєднання різних технологій і механізмів, що, в свою чергу, призведе до можливої необхідності розробки власних рішень на основі існуючих, для універсалізації наших система для більшості користувачів. Такий підхід створить додаткові труднощі та перешкоди в розробці рішень, але в свою чергу допоможе уникнути залежності від сторонніх розробників, маючи власний набір інструментів для роботи з потрібним функціоналом.

TCF07 має вплив на складність проекту рівний 4, з огляду на те, що кількість елементів інтерфейсу для управління зведена до мінімуму, що дозволяє максимально спростити процес роботи з системою, створюючи високий рівень зручність для користувача при роботі з системою. У той же час, максимізація простоти управління вимагає значної оптимізації та автоматизації процесів у самій системі, що, в свою чергу, ускладнить процес її розробки та додасть складності її конструкції. Однак, завдяки тому, що функціональність системи не потребує складних розрахунків, автоматизація та спрощення роботи в цілому не сильно вплине на складність проекту, залишаючи велику кількість можливостей для розширення та вдосконалення у майбутньому.

TCF08 важливість впливу на складність проекту дорівнює 5, оскільки, а також для простоти установки через портативність системи, необхідно спростити велику кількість функціональних можливостей, що в свою чергу ускладнює процес розробки та проектування необхідних механізмів. Склад-

ність цього моменту можна зменшити, переклавши обробку частини процесів в систему користувача, що знизить навантаження на сервери і прискорить обробку даних. Але через те, що користувацькі системи можуть мати різний рівень продуктивності, ми можемо втратити велику кількість користувачів через те, що їхні системи не можуть обробити необхідні дані.

TCF09 вплив на складність проекту дорівнює 4, з огляду на те, що для зручного і швидкого внесення змін до існуючої системи необхідно буде розробити спеціалізований набір інструментів, який допоможе взаємодіяти з системою на необхідному рівні. Через це складність розробки може істотно зрости, оскільки створюваний нами інструментарій повинен бути не тільки простим і зручним у використанні, але й оптимізованим, щоб дозволяти вносити необхідні зміни в систему якомога швидше і ефективніше, не впливаючи на його функціонування. Хоча враховуючи те, що існує велика кількість готових інструментів для роботи з даними, їх застосування допоможе не збільшити складність проекту до максимуму, залишивши її на середньому рівні.

TCF10 має вплив на складність проекту рівний 4, з огляду на те, що обробка отриманих даних, а також їх демонстрація повинні проводитися паралельно для оптимізації використання ресурсів. Щоб вирішити цю проблему, буде необхідно розробити оптимізований і добре розпаралелений код, який зможе виконувати велику кількість паралельних операцій, не вносячи істотних змін у загальний процес роботи системи, тобто не створюватиме затримок. або простою в його роботі. Процес розробки такого коду може збільшити складність проекту, але завдяки вже наявним інструментам для розпаралелення завдань, рівень складності залишиться на прийнятному рівні, залишаючи можливість подальшого розвитку.

TCF11 має вплив на складність проекту, що дорівнює 4, з огляду на те, що деякі засоби управління системою вимагають розробки спеціалізованих механізмів захисту, які захистять систему від неправильного використання або навмисних спроб дестабілізувати її роботу. З цієї причини загальна складність проекту зростає, але через те, що більшість проблем безпеки (включа-

ючи несанкціонований доступ) можна вирішити за допомогою існуючих інструментів і механізмів, її рівень не досягає свого максимуму. Крім того, рівень безпеки системи, що розробляється, не можна порівняти з рівнем безпеки банківської чи державної системи; з цієї причини не потрібно надто ускладнювати його структуру та алгоритми.

TCF12 вплив на складність проекту дорівнює 4, з огляду на те, що для коректної роботи системи з різними інституціями необхідно розробити способи взаємодії з кожною з них. Особливо щодо даних, які вони передають, через те, що не існує єдиного стандарту ведення записів та організації бази даних. З цієї причини необхідно розробити інструментарій, який би дозволяв гнучко налаштовувати процес обміну інформацією з будь-якою інституцією, не створюючи проблем у роботі всієї системи, що в свою чергу збільшує складність розробки проекту та підвищує його загальна складність вище середнього рівня.

TCF13 вплив на складність проекту дорівнює 2, з огляду на те, що управління системою кінцевим користувачем максимально спрощено та уніфіковано, щоб людина мала можливість застосувати наявні навички та досвід роботи в системі, що розробляється. Наприклад, інтерфейс системи передбачає використання випадаючих списків і полів пошуку, схожих на ті, які ми можемо спостерігати в будь-якому сервісі Google. Завдяки цьому користувачеві не потрібно володіти спеціальними навичками роботи з системою, а всі додаткові функції та можливості управління можна продемонструвати за допомогою підказок і сторінки допомоги.

3.3.3 Методика організації розрахунків

Накопичене значення UTV (Unadjusted TCF Value) розрахуємо як:

$$UTV = \sum_{i=1}^{13} [V_i \times k_i],$$

де V_i – ваговий коефіцієнт кожного з 13-ти показників технічної складності проекту;

k_i – кількісне значення впливу кожного з 13-ти показників технічної складності проекту.

Ваговий коефіцієнт TCF Weight Factor (TWF) – за замовчуванням встановлюється як 0,01; хоча теоретично може корегуватись. Постійна проекту TCF Constant (TC) – гарантований мінімум показника TC = 0,6. Таким чином TCF для кожного конкретного проекту приймає значення:

$$TCF = TC + TWF \times UTV,$$

тобто від теоретичного мінімуму: 0,6 до теоретичного максимуму: 1,25; чим ілюструє як позитивний вплив на витрати (зменшуючи на 40%), так і негативний – збільшуючи на 25%.

Крім того, слід розрахувати значення TCF за допомогою формули, де TWF – це ваговий коефіцієнт TCF (TWF – is a TCF Weight Factor), а UTV – це нескориговане значення TCF (UTV – is Unadjusted TCF Value). Загальне значення для таблиці TCF:

$$TCF = 0{,}6 + (TWF * UTV),$$

якщо підставити значення:

$$TCF = 0{,}6 + 54{,}5 * 0{,}01 = 1{,}145.$$

Отримавши описані вище результати, можна сказати, що технічна складність проекту знаходиться на досить високому рівні. Хоча деякі чинники можуть не сильно вплинути на кінцеві результати, окремі компоненти можуть серйозно обумовити завдання. До цих чинників відносяться розподіл

системи, компактність і простота монтажу. Через високі оцінки впливу проекту загальна складність значно зросла.

Причиною цього є складність розробки таких аспектів системи, а також ступінь їх інтеграції. Наприклад, компактність повинна забезпечувати легкість і невибагливість до ресурсів. Для вирішення такої проблеми необхідно буде скоротити кількість операцій, що обробляються на стороні клієнта, перемістивши їх на сторону сервера. Через це буде необхідно розробити або впровадити інструменти, які дозволять реалізувати це завдання.

Простота установки також передбачає зниження вимог до користувача, що означає збільшення кількості завдань, які вирішує серверна частина системи. У зв’язку з вимогою розподілу, зокрема, для збалансування доступу до даних з будь-якої точки країни (чи світу, якщо вона розширюється), потрібно створити не лише мережу, здатну розподіляти навантаження, а й, знову ж таки, інструменти. для його конфігурації, встановлення та взаємодії.

Виходячи з вищесказаного, можна сказати, що впровадження даної системи поставить високу, але все ж не максимальну планку складності, що, в свою чергу, призведе до зміни інших факторів проекту.

3.3.5 Розрахунок показника складності середовища

Оскільки розвиток ПС – це комплексний проект, що включає також стейкхолдерів, то необхідно при розрахунках показників проекту розглянути проектну команду, тобто включити чинники розробників – діючих осіб (ДО) до оцінки складності ПС.

Розрахунковий чинник, що стосуються ДО, носить назву чинника складності оточуючого середовища (ECF) та обчислюється виходячи із декількох показників. Значення Unadjusted ECF Value (UEV), що розраховується спираючись на перелік показників кваліфікації проектної команди, яка задіяна у розробці.

Важливо врахувати сторону співробітників і фактори, які на них впливатимуть. Ступінь впливу на працівників, які беруть участь у розробці проекту, може сильно вплинути не тільки на їхню продуктивність, а й на продукт в цілому. Тому для правильної оцінки можливих факторів впливу, необхідно розглянути найбільш поширені з них і провести розрахунок, за результатами якого зробити висновок про рівень важливості одного або кількох факторів впливу. Для цього використовується метрика ECF (Environmental Complexity Factor) або чинник складності навколишнього середовища.

Згідно з одним визначенням, складність навколишнього середовища відноситься до кількості змінних навколишнього середовища та їх взаємозв'язку. Низька організаційна складність передбачає, що середовище можна описати невеликою кількістю факторів, тоді як висока складність передбачає, що середовище можна описати великою кількістю значущих змінних.

Використовуючи наведені коефіцієнти технічної складності, ми присвоюємо відповідні значення кожному фактору, представляючи їх усі в таблиці 3.4.

Таблиця 3.4 – Таблиця показників ECF

Метрика	Опис	Вага	Значення	ECF
ECF01	Знайомий з RUP (Rational Unified Process)	1,50	4,00	6,00
ECF02	Досвід застосування	0,50	3,00	1,50
ECF03	Об'єктно-орієнтований досвід	1,00	5,00	5,00
ECF04	Здатність провідного аналітика	0,50	3,50	1,75
ECF05	Мотивація	1,00	4,00	4,00

Продовження таблиці 3.4

ECF06	Стабільні вимоги	2,00	5,00	10,00
ECF07	Неповний робочий день	-1,00	3,00	-3,00
ECF08	Складна мова програмування	-1,00	2,00	-2,00
			Усього (UEV):	23.25

Щоб обґрунтувати встановлені значення ECF, детально опишемо причини, з яких ці значення були встановлені.

ECF01 має значення впливу на проект рівне 4, тому що знання UML для розробника є спрощенням розуміння вимог клієнта. Це може пришвидшити процес розробки та полегшити пояснення змін або вдосконалень для клієнта. Крім того, ми зможемо полегшити роботу інженера-програміста, оскільки йому не знадобиться додатковий час на розробку окремих діаграм для розробників, що ще більше скоротить час розробки, але в той же час збільшить вимоги до персонал.

Вплив ECF02 на складність проекту дорівнює 3, оскільки розробка ПЗ на обраній мові програмування, якою є Python, не вимагає використання важких компіляторів, що дозволяє використовувати для роботи практично будь-який текстовий редактор. Однак, щоб полегшити процес розробки та уникнути основних помилок, будуть використані відповідні середовища розробки, які не вимагають великого досвіду чи знань при роботі з ними. Наприклад, такі як PyCharm або Visual Studio Code.

ECF03 має значення впливу на складність проекту, що дорівнює 5, з огляду на те, що знання принципів об'єктно-орієнтованого підходу є дуже важливим при розробці. Крім того, цей параметр може визначити рівень спеціалізації конкретного розробника, дозволяючи нам зрозуміти, наскільки добре він зможе виконувати поставлені завдання під час розробки, а також рівень ефективності та оптимізації його коду. Завдяки цьому ми зможемо ско-

ротити час розробки, не витрачаючи додатковий час на вдосконалення коду пізніше, роблячи це вже в процесі.

ECF04 має значення впливу на складність проекту рівне 3,5, оскільки компетентність головного аналітика хоч і має високе значення, але не є критичною для проекту. Сюди ж входить компетенція головних керівників проекту. При максимальній компетентності вартість і вимоги зростуть, що вплине на проект в цілому і може ускладнити процес розробки.

ECF05 має вплив на складність проекту рівний 4, оскільки рівень мотивації команди розробників, як і інших учасників проекту, безпосередньо впливає на швидкість і процес розробки. При низькому рівні мотивації знижується інтерес і ентузіазм, що може мати негативний ефект у вигляді погіршення загальної якості виконуваної роботи. Гроші можуть бути одним із мотивуючих факторів, але гроші не завжди можуть покрити низький моральний дух команди та утримати його на досить тривалий час, і цей підхід може працювати не для всіх однаково. Тому цей фактор є одним із важливих аспектів при розробці проекту.

ECF06 має вплив на складність проекту рівним 5, оскільки стабільність вимог сильно впливає на процес розробки. Через часту модифікацію завдань ми можемо зіткнутися з проблемою розтягування часу розробки. Через внесення частих змін до готового кодексу це не тільки ускладнить, а й спричинить підвищену витрату ресурсів, адже при появі термінових змін необхідно буде перезалучати працівників з оплатою позаурочний час або доплата за зміну термінів виконання кожного зі змінених завдань.

ECF07 має вплив на складність проекту рівний 3, з огляду на те, що для розробки проекту буде потрібно залучення сторонніх спеціалістів, які не працюють постійно в команді. Основна причина залучення таких спеціалістів – прискорення процесу розробки та зниження витрат ресурсів. Проте, залучаючи аутсорсингових спеціалістів, ми можемо зіткнутися з проблемою доступності, тому що нам доведеться підлаштовуватися під їхній графік і в разі виникнення непередбачених ситуацій ми не зможемо їх швидко вирішити.

ECF08 має значення впливу на складність проекту, що дорівнює 2, оскільки обрана мова програмування вважається досить легкою, складність проекту не буде сильно збільшуватися при його застосуванні. Крім того, поширеність допомагає швидко знаходити рішення проблем завдяки великій спільноті та кількості спеціалістів. Крім того, використовуючи готові рішення для деяких механізмів системи, таких як профілювання та робота з базами даних, ми можемо спростити процес розробки та скоротити витрати часу, що, в свою чергу, зменшить витрати ресурсів.

3.3.3 Методика організації розрахунків

Накопичене значення UEV розраховується як:

$$UEV = \sum_{i=1}^{8} [W_i \times m_i],$$

де W_i – ваговий коефіцієнт кожного з 8-ми показників кваліфікації проектної команди;

m_i – кількісне значення наявності кожного з 8-ми показників проектної команди.

Ваговий коефіцієнт ECF Weight Factor (EWF) – за замовчуванням встановлюється як (-0,03), хоча теоретично може корегуватись. Постійна проекту ECF Constant (EC) – абсолютний максимум показника EC = 1,4. ECF ілюструє як позитивний вплив на витрати (зменшуючи на 60%), так і негативний – збільшуючи на 40%.

Для обчислення значення ECF, ми використаємо формулу, де EWF – це ваговий коефіцієнт ECF (EWF – is a ECF Weight Factor), UEV – це нескориговане значення ECF (UEV – is a Unadjusted ECF Value,) а EC – це константа ECF (EC – is ECF Constant):

$$ECF = EC + (EWF * UEV),$$

якщо підставити значення:

$$ECF = 1,4 + (- 0,03 * 23,25) = 0,702.$$

Виходячи з отриманих результатів, видно, що ECF має досить високе значення, наближаючись до 1. Більшою мірою це пов’язано з наявністю факторів, які сильно впливають на процес роботи над проектом. Одним з найважливіших факторів є мотивація, тому значення впливу цього фактору встановлено на високому рівні. Хороша мотивація робочого колективу може не тільки прискорити процес, а й підвищити ефективність в цілому, оскільки зацікавлення та бажання, більшою мірою, допомагають досягти поставлених цілей, ніж чисто матеріально-фінансова зацікавленість. Ще один важливий фактор – стабільні вимоги.

Структуроване та деталізоване завдання дозволяє не тільки уникнути непорозумінь або неточностей у процесі розробки, але й дозволяє точніше розподілити навантаження, створивши часове планування завдань та рамки для реалізації певних цілей. При наявності постійно мінливих вимог або завдань введення нових структурних або візуальних елементів може істотно ускладнити процес розробки, зробити його нестабільним і неорганізованим, це може створити не тільки проблеми, але і призвести до збільшення термінів до невизначений час.

Також важливо відзначити, що важливим фактором є професіоналізм і знання членів команди. Якщо рівень знань, необхідний для виконання конкретного завдання, недостатній, для його підвищення до необхідного рівня знадобляться додаткові ресурси, що, в свою чергу, зменшить часовий ресурс проекту. Виходячи з вищесказаного, ми можемо сказати, що впровадження інформаційної системи потребує професійної команди, яка має належний рівень знань не лише у розробці програмного забезпечення, але й у розумінні

вимог до завдання (наприклад, знання UML тощо), хороша мотивація та вміння працювати у команді.

3.3.4 Оцінка загального часу та проектних витрат на розробку

Розрахункове значення для BB UCP – це число, що отримаємо для управління простим (див. табл. 3.1) BB. Значення вимірюється у так званих «проектних точках» та обчислюється як:

$$UCP = UUCP \times TCF \times ECF.$$

Далі прогнозується загальний час УП Estimated Work Effort (EWE) у годинах:

$$EWE = UCP \times DR,$$

де DR (Duration) – значення часу тривання розробки однієї «проектної точки» USP, яке краще за все обирати, спираючись на використання досвіду проектів-аналогів.

DR є критичним чинником, і хоча його значення за замовчуванням дорівнює 10 год., воно легко може перевищувати 30 год., якщо значення UEV незбалансоване. При обранні DR необхідно брати до уваги кваліфікацію розробників, наприклад для нової команди слід обрати DR = 20 год.

Таким чином, приходимо до оцінки вартості проекту EC:

$$EC = EWE \times DHR,$$

де DHR (Default Hourly Rate) – усереднена за статистичними даними погодинна ставка персоналу (програміста).

Використовуючи результати, отримані під час розрахунку метрик проекту, розраховуємо загальні показники та робимо висновки на основі отриманих результатів. Для цього використовуємо автоматичний розрахунок у CASE-засобі Enterprise Architecht [27]. За допомогою цього програмного забезпечення ми можемо не тільки візуалізувати систему за допомогою діаграм, таких як Use Case, але також розрахувати її продуктивність і визначити доцільність розробки.

Використовуючи отримані вище результати та показники, проводимо розрахунок (рис. 3.2).

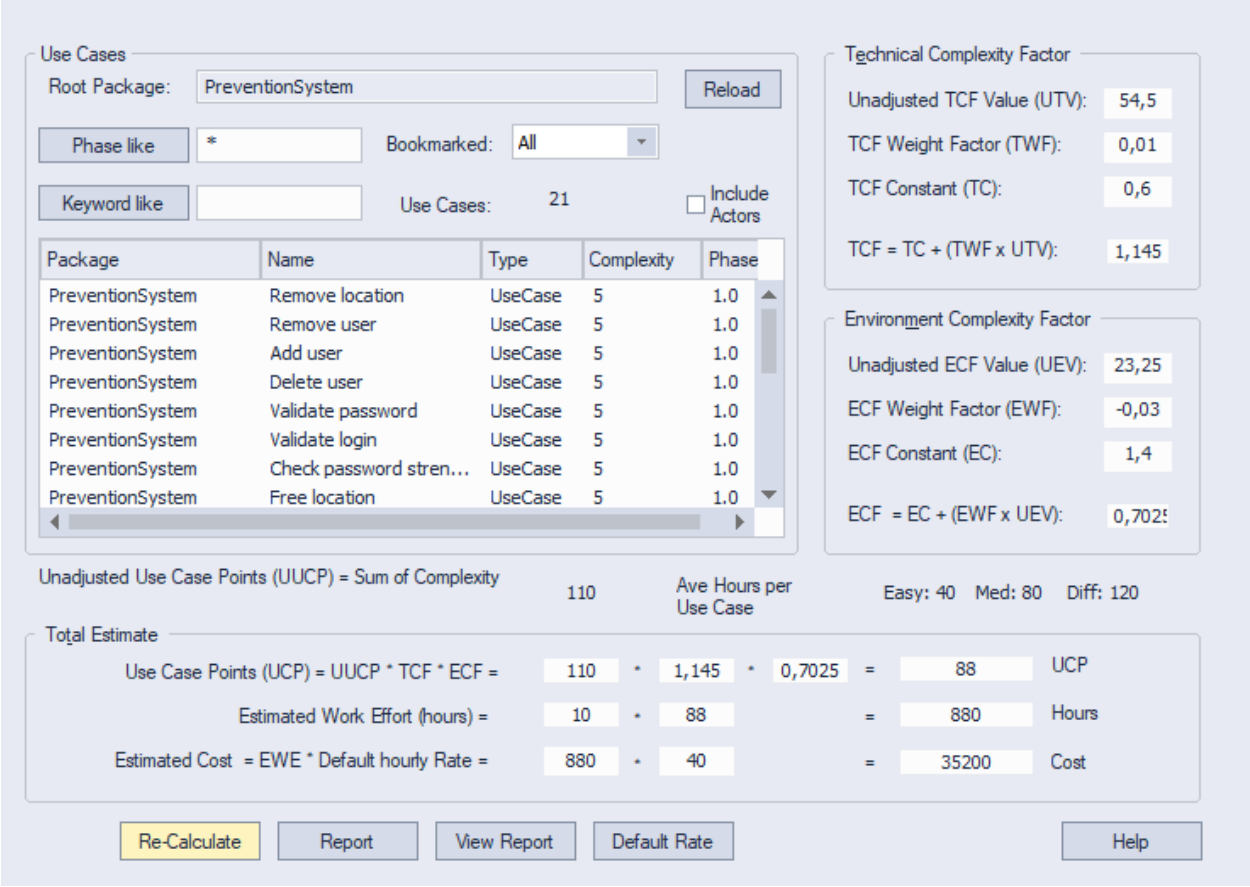

Рисунок 3.2 – Розрахунок кінцевих значень на основі попереднього встановлення результатів та показників

Для більшості розрахованих чинників та показників коментарі щодо їх одержання наведено вичерпно вище.

Пояснимо тільки призначення елементів керування, що подано у нижній частині рис. 3.2. Кнопка «Re-Calculate» – виконує перерахунок проектних чинників, спираючись на змінені показники. Кнопка «Report» – генерує запит на створення та вказівку шляху збереження звіту з оцінки проекту у форматі «RTF». Кнопка «View Report» – виконує перегляд встановлених показників оцінки проекту із викликом ПЗ, асоційованого із переглядом Rich Text Files (RTF-файлів), наприклад: Microsoft Word. Кнопка «Default rate» – викликає діалогове вікно зміни показників оцінки проекту, що завдані у Enterprise Architect 14 за замовчуванням.

3.4 Оптимізація отриманих результатів

Як ми бачимо з отриманих результатів, у нас є кілька варіантів, які можна оптимізувати, щоб зменшити загальну вартість проекту. Першим варіантом зміни буде оплата за годину, оскільки в реальному житті ця вартість значно нижча. Після цього ми можемо повторно перевірити інші фактори проекту та внести деякі покращення, зменшивши не лише вартість проекту, але й його загальну складність.

Використовуючи попередньо заповнені таблиці факторів і параметри проектів, ми можемо максимально оптимізувати всі існуючі фактори і спробувати в цілому зменшити вартість проекту і часові ресурси. Після деяких удосконалень ми можемо написати короткий висновок про загальну вартість і припустити, наскільки вигідно створити цей проект.

Було вирішено відкорегувати показники технічної складності проекту (табл. 3.5).

Спочатку значення TCF04 було змінено з 5 на 4 через можливе зменшення обсягу даних, необхідних для обробки, стиснення даних і використання більшої обчислювальної потужності. Після цього портативність проекту (TCF08) знизилася до 4, тому що портативність реалізована за допомогою вже існуючих інструментів і рішень, дозволяє спростити вимоги до розробки

та знизити загальну складність. Остання можлива оптимізація, зроблена для TCF01 розподіленої системи: змінено значення з 5 до 4,5, зменшивши потребу в розподілених обчисленнях на кількох машинах. Це покращення є наслідком оптимізації «Комплексної внутрішньої обробки». В результаті ми знизили загальну складність UTV з 54,5 до 51,5.

Таблиця 3.5 – Оптимізація факторів TCF

Показник	Опис	Вага	Старе значення	Нове значення	TCF
TCF04	Комплексна внутрішня обробка	1,00	5,00	4,00	4,00
TCF08	Портативність	2,00	5,00	4,00	8,00
TCF01	Розподілена система	2,00	5,00	4,00	9,00

Наступним кроком для оптимізації було покращення факторів складності середовища ECF (табл. 3.6).

Таблиця 3.6 – Оптимізація показників ECF

Метрика	Опис	Вага	Старе значення	Нове значення	ECF
ECF01	Знайомий з RUP (Rational Unified Process)	1,50	4,00	4,50	6,75
ECF04	Здатність провідного аналітика	0,50	3,50	4,50	2,25
ECF07	Неповний робочий день	-1,00	3,00	2,00	-2,00

Одним із можливих факторів покращення був досвід роботи з UML – ECF01. Знання UML є дуже корисною навичкою у процесі розробки, покращуючи комунікацію. Добре знання UML не є обов’язковим, але це стане ва-

жливим під час робочого процесу над складним проектом, тому ми повинні підняти його з 4 до 4,5.

Наступний фактор – це можливості провідного аналітика – ECF04. Цей фактор дуже важливий для якості розробки, тому що тільки хороший аналітик може передбачити можливі проблеми і розробити відповідний план, що зробить розробку проекту набагато простіше і безпечніше від несподіваних ситуацій. Тому потрібно взяти на роботу хорошого фахівця з відповідними знаннями, підвищивши цей коефіцієнт з 3,5 до 4,5.

Останній фактор, який підлягає оптимізації – це ECF07 «Працівники, які працюють неповний робочий день». Через сильний вплив аутсорсингових працівників на процес розробки ми вирішили знизити початкове значення з 3 до 2, що дозволяє нам менше покладатися на графік роботи спеціалістів, які працюють неповний робочий день. В результаті ми підняли загальну складність середовища з 23,25 до 25,50.

З одного боку, таку зміну навряд чи можна назвати «покращенням». Але, з іншого боку, підвищені вимоги до працівників дозволили нам скоротити час, необхідний для розробки, за рахунок більшої кількості професійних спеціалістів.

Остання зроблена оптимізація – це зниження погодинної оплати працівників. Основною причиною такої зміни є різниця між реальністю та теоретичними значеннями. У реальних умовах вартість години коливається від 25$ до 35$, тому оптимальним рішенням тут буде встановити вартість 30$, наближаючи її до реальних ринкових умов, що відповідним чином відображено на рис. 3.3.

Таким чином, завдяки аналізу розрахованих показників оптимізовано ключові фактори, з яких будується майбутня ціна продукту та бюджет часу розробки. Кінцева вартість проекту знижена на 12,1 тис. $: 35200$ – 23100$ = 12100$. Після оптимізації складності загальна кількість годин, необхідних для проекту, зменшилася на: 880 – 770 = 110 (год.).

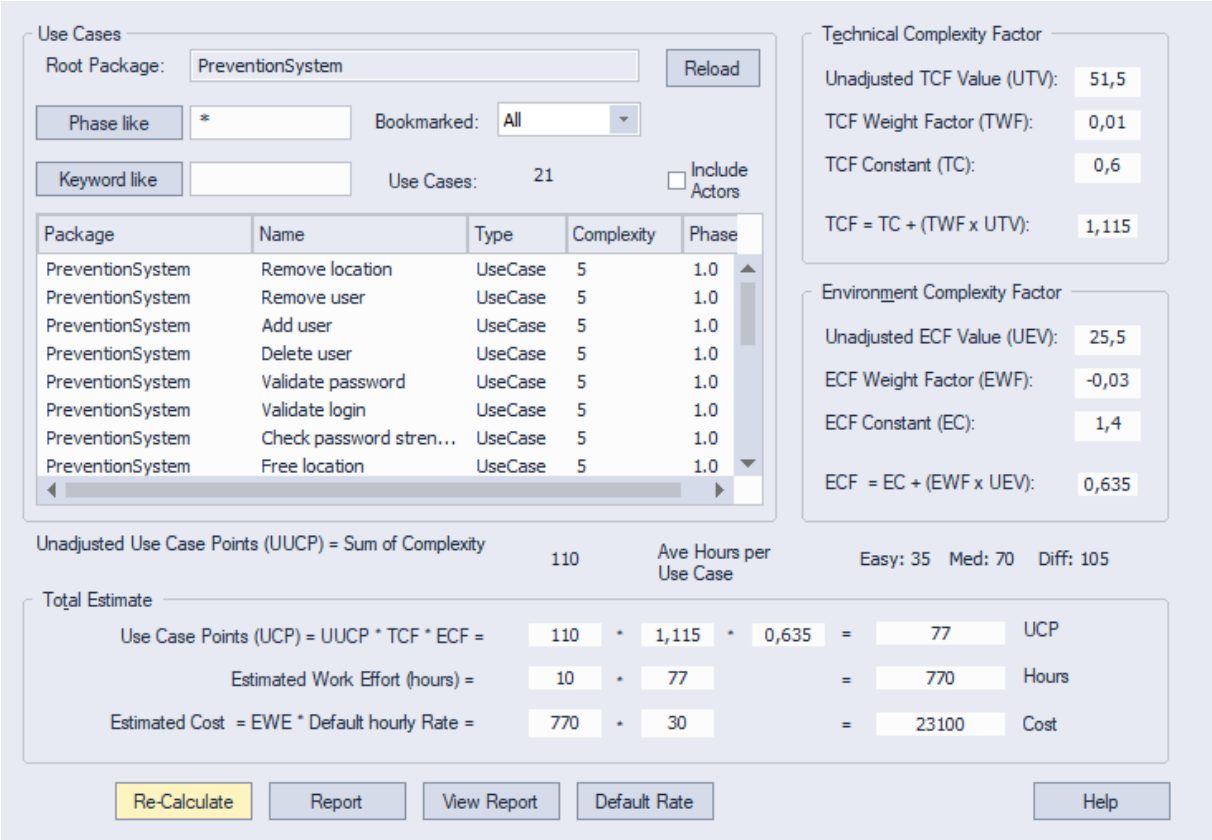

Рисунок 3.3 – Розрахунок кінцевих значень після оптимізації

3.5 Висновки за розділом

Для оцінки проектних показників з розробки ІС необхідно мати набір даних із конкретизованих проектних чинників, а саме:

– 3-х видів ваги ВВ;

– 13 показників технічної складності;

– 8 показників кваліфікації персоналу.

Необхідно отримати розрахункові показники за якими буде відбуватись прогнозований фінансовий перелік витрат. Також треба формалізувати критерії оцінки та визначення складності вже готової ПС, але яка потребує зміни (розвитку) із плином часу.

1) Метод Карнера рекомендує виключати розширені ВВ при розрахунку проектних точок. У [17] рекомендовано розглядати усі ВВ при оцінки

проекту, якщо ці ВВ вимагають перепроектування функціоналу, то існують зусилля на їх переробку, що повинні враховуватися за наведеною методикою.

2) Об'єктивний спосіб точно налаштувати розробку проекту – це розглянути ВВ 3-х – 7-ми вже успішно завершених (у свій час) схожих проектів. Після аналізу завершеного проекту і вивчення звіту про показники, доступні чинники можуть бути точно відкориговані, щоб дати оцінку фактичним годинам розробки. Згодом, можна використовувати ці дані у якості базової траєкторії ЖЦ ІС.

3) Достатня перевірка працездатності полягає в тому, щоб спираючись на показник «Ave Hours per Use Case» (рис. 3.3) проаналізувати: чи можна управляти розвитком простого, середнього чи складного ВВ у відведений час (включаючи усі стадії ЖЦ ІС).

4) Не слід очікувати вичерпної відповіді на питання «скільки коштує розробка ІС?» або «як довго вона триватиме?» – необхідно оцінювати отримані статистичні дані, управляти показниками та аналізувати досвід втілення показників успішних проектів.

Кожна змінна, що використовується для розрахунків у рамках проектної оцінки визначається та обчислюється окремо із використанням:

– вимірювань характерних параметрів;
– технічної складності проекту;
– рівня кваліфікації команди розробників;
– вагових коефіцієнтів;
– обмежуючих констант.

Встановлення вагових коефіцієнтів та обрання значення обмежуючих констант засновані на багато чисельній статистиці найбільш схожих проектів, що виконані за технологією А. Якобсона.

Управління параметрами проводяться керівником проекту, який, спираючись на досвід 3-х – 7-ми схожих проектів, виходить із власних уявлень про технічну складність проекту та можливостях команди, яка буде виконувати розробку.

ВИСНОВКИ

Пандемія COVID-19 показала, що для швидкого реагування на кризи подібного масштабу сучасним державам необхідні потужні інструменти для збору та аналізу інформації. Створення ефективної ІС для контролю інфікування коронавірусом є критичним кроком у напрямку боротьби з поточними та майбутніми викликами. Вона не тільки допоможе врятувати життя, але й мінімізувати економічні та соціальні наслідки пандемії, забезпечивши швидку адаптацію суспільства до нових умов.

Створення ІС для контролю інфікування коронавірусом є важливим кроком у побудові ефективної інфраструктури для управління кризовими ситуаціями в охороні здоров’я. Пандемія COVID-19 оголила низку системних проблем, зокрема недосконалість існуючих механізмів збору, обробки та аналізу інформації про поширення інфекцій. Це, у свою чергу, призвело до того, що багато країн стикнулися з труднощами в управлінні ситуацією та своєчасному прийнятті рішень на всіх рівнях. Погана координація між державними органами, нестача достовірних даних та неефективна комунікація з громадянами значно погіршили ситуацію.

Однією з основних переваг ІС для контролю інфікування є можливість забезпечити своєчасний і точний збір даних у реальному часі. Замість того, щоб покладатися на фрагментовані або застарілі дані, що надходять від окремих медичних закладів або установ, така система може централізовано акумулювати інформацію з різних джерел: від державних структур, медичних закладів, тестових лабораторій, до мобільних додатків громадян. Це дозволяє створити повну картину поточної епідеміологічної ситуації, що є ключовим для ефективного управління не тільки під час активної фази пандемії, але і для прогнозування її подальшого розвитку.

Для досягнення мети роботи в рамках комплексної теми була запропонована модель ІС, в якій основним джерелом інформації є сама людина. Було

сформовано: проектні метрики, розраховано та знайдено шляхи оптимізації проектних показників.

У якості виконаних завдань було:

– проаналізовано обрану предметну галузь;

– визначено методи прогнозування та аналізу даних;

– здійснено формування та розрахунок показників проекту, включно із характеристикою проекту, складанням проектних діаграм;

– сформовано проектні метрики за оцінкою складності;

– виконано оптимізацію отриманих результатів.

У якості перспектив розвитку поданої теми слід окреслити траєкторії розвитку подальших досліджень:

– використання алгоритмів штучного інтелекту;

– використання методів машинного навчання, які здатні аналізувати великі обсяги даних;

– на основі отриманих результатів прогнозування подальшого поширення вірусу;

– оцінювання ризиків нових спалахів;

– рекомендація оптимальних заходів для запобігання розповсюдження інфекції;

– запровадження своєчасних обмежувальних заходів, які підтримуються ІС, таких як: карантини, соціальне дистанціювання або запровадження додаткових тестувань в окремих регіонах.

Впровадження таких систем, незважаючи на їх очевидні переваги, також вимагає ретельного планування та координації на етапі розробки та експлуатації. Необхідно враховувати не лише технологічні аспекти, а й культурні, економічні та соціальні чинники. Наприклад, у країнах із різним рівнем доступу до інтернету або мобільного зв’язку необхідно забезпечити альтернативні канали доступу до системи, щоб інформація була доступною всім верствам населення. Крім того, важливо розробити навчальні програми для

медичних працівників та державних службовців, які будуть працювати з такими системами, що забезпечить їх ефективне використання.

Отже, ІС для контролю інфікування відіграють ключову роль у сучасному світі для забезпечення безпеки населення та ефективної боротьби з пандеміями. Вони надають урядам і міжнародним організаціям потужні інструменти для управління кризовими ситуаціями, дозволяючи забезпечити швидке реагування на загрози, мінімізуючи негативні наслідки як для системи охорони здоров’я, так і для суспільства в цілому.

Технологічний прогрес у сфері інформаційних технологій створює передумови для того, щоб такі системи стали невід’ємною частиною майбутніх стратегій у сфері громадського здоров’я та життя людей у будь-який точці планети.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Epidemic, Endemic, Pandemic: What are the Differences? URL: https://www.publichealth.columbia.edu/public-health-now/news/epidemic-endemic-pandemic-what-are-differences (дата звернення: 06.09.2024 р.).
2. Coronavirus disease (COVID-19) pandemic. URL: https://www.who.int/emergencies/diseases/novel-coronavirus-2019 (дата звернення: 03.09.2024 р.).
3. Pandemic Influenza Preparedness and Response: A WHO Guidance Document. URL: https://www.ncbi.nlm.nih.gov/books/NBK143061/ (дата звернення: 05.09.2024 р.).
4. Nadler P., Arcucci R., Guo Y. A neural sir model for global forecasting. *Machine Learning for Health.* PMLR. Vol. 136. pp. 254-266.
5. Atkeson A., Kopecky K., Zha, T. Estimating and forecasting disease scenarios for COVID-19 with an SIR model (No. w27335). National Bureau of Economic Research. 150 p.
6. Adams P., AZRI N. R., SATHASIVAM S., NAM C. J., HANG T. (2023). Forecast on COVID-19 cases in Malaysia using SIRS model and Adams predictor-corrector method. *Journal of Quality Measurement and Analysis JQMA.* 2023. Vol. 19 (1). pp. 59-73.
7. Salaudeen A., Sathasivam S., Nam C. J., Hang, T. Y. Modelling the Early Outbreak of Covid-19 Disease in Malaysia Using SIRS Model with 4-Step Adams-Bashforth-Moulton Predictor-Corrector Method. *Applied Mathematics and Computational Intelligence (AMCI).* 2024. Vol. 13(2). pp. 121-136.
8. Hamerly G., Drake J. Accelerating Lloyd's algorithm for k-means clustering. *Partitional clustering algorithms*, 2015. pp. 41-78.
9. Tang C., Monteleoni C. On Lloyd's algorithm: new theoretical insights for clustering in practice. *Artificial Intelligence and Statistics.* 2016, May. pp. 1280-1289.

10. Томащук М. М. Застосування діаграм Вороного при геоінформаційному моделюванні спроможних територіальних громад. *Містобудування та територіальне планування.* 2018. № 66. С. 594-601.
11. Nayak J., Naik, B., Behera, H. Fuzzy C-means (FCM) clustering algorithm: a decade review from 2000 to 2014. *Computational Intelligence in Data Mining-Volume 2: Proceedings of the International Conference on CIDM,* 20-21 December 2014. pp. 133-149. Springer India.
12. Askari S. Fuzzy C-Means clustering algorithm for data with unequal cluster sizes and contaminated with noise and outliers: Review and development. *Expert Systems with Applications*, 2021. 165, 113856.
13. Hathaway R. J., Bezdek J. C., Hu, Y. Generalized fuzzy c-means clustering strategies using L/sub p/norm distances. *IEEE transactions on Fuzzy Systems.* 2020. 8(5), 576-582.
14. Jia S., Zhao Q., Wang L., Li Y. Random projection based bias-corrected fuzzy C-means algorithm for hyperspectral remote sensing image segmentation. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.* 2020. Vol. 43. pp. 435-439.
15. Великодний С. С. Моделювання складних процесів та систем (Частина 1): конспект лекцій. Одеса: Одеський державний екологічний університет, 2021. 92 с. ISBN 978-966-186-181-6. (URL: http://eprints.library.odeku.edu.ua/id/eprint/9494/)
16. Velykodniy S. S. Analysis and synthesis of the results of complex experimental research on reengineering of open CAD systems. *Applied Aspects of Information Technology.* 2019. Vol. 2. No 3. P. 186–205. DOI: 10.15276/aait.03.2019.2.
17. Великодний С. С. Моделі та методи проактивного управління проєктами із розвитку програмних систем і продуктів. Монографія. Одеса: Одеський державний екологічний університет, 2021. 322 с. ISBN 978-966-186-182-3. (URL: http://eprints.library.odeku.edu.ua/id/eprint/9595/)

18. Karner G. Resource Estimation for Objectory Projects: project report. Objective Systems. San Francisco: AB, 1993. 9 p.
19. Anda B. Effort Estimation of Use Cases for Incremental Large-Scale Software Development. 27-th International Conference on Software Engineering, St. Louis, MO, 15 – 21 May 2005, P. 303–311.
20. Carroll E. R. Estimating Software Based on Use Case Point. OOPSLA '05: Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, San Diego, 2005. P. 257–265. DOI: 10.1145/1094855.1094960
21. Clemmons R. Project Estimation with Use Case Points.*Cross Talk.*2016. Vol. 2, Iss. February. P. 18–22.
22. Дідковська М. В. Тестування: Критерії та методи. Київ: НТУУ «КПІ», 2010. 96 с.
23. Cohn M. Agile Estimating and Planning. New York City: Prentice Hall, 2005. 368 p.
24. Великодний С. С., Тимофєєва О. С., Зайцева-Великодна С. С. Метод розрахунку показників оцінки проекту при виконанні реінжинірингу програмних систем. *Радіоелектроніка, інформатика, управління.* 2018. № 4. С. 135–142. (Web Of Science) DOI: 10.15588/1607-3274-2018-4-13.
25. Velykodniy S., Burlachenko Zh., Zaitseva-Velykodna S. Modelling the behavioural component of the emergent parallel processes of working with graph databases using Petri net-tools. International Journal of Parallel, Emergent and Distributed Systems. (Scopus) 2021. Vol. 36. Iss. 6. P. 498-515. (Scopus) DOI: https://doi.org/10.1080/17445760.2021.1934836. Taylor & Francis Group, England & Wales. London.
26. Великодний С. С., Бурлаченко Ж. В., Зайцева-Великодна С. С. Реінжиніринг графічних баз даних у середовищі відкритої системи автоматизованого проектування BRL-CAD. Моделювання структурної частини. Вісник Кременчуцького національного університету ім. Михайла

Острогрaдського. 2019. Вип. 3 (116). С. 130–139. DOI: 10.30929/1995-0519.2019.3.130-139.

27. Enterprise Architect documentation [Electronic source]. URL: https://sparxsystems.com/enterprise_architect_user_guide/14.0/guidebooks/tools_ea_documentation.html (дата звернення: 02.09.2024).