

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЧЕРНІВЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ЮРІЯ ФЕДЬКОВИЧА**

**Факультет математики та інформатики  
Кафедра математичного моделювання**

**УПРАВЛІННЯ РОЗРОБКОЮ ТА РИЗИКАМИ ПРОЄКТУ ДЛЯ ЛЮДЕЙ  
ІЗ ОСОБЛИВИМИ ПОТРЕБАМИ**

**Кваліфікаційна робота**

**Рівень вищої освіти – перший (бакалаврський)**

***Виконала:***

студентка 4 курсу, 407 групи

**Гасюк Марія Ігорівна**

***Керівник:***

кандидат фізико-математичних наук,  
доцент Піддубна Л.А.

*До захисту допущено  
на засіданні кафедри математичного моделювання  
протокол № 16 від 30 травня 2023 р.  
Зав. кафедрою \_\_\_\_\_ проф. Черевко І.М.*

## *Анотація*

Дана робота присвячена процесу організації управління розробкою та ризиками в проєкті для людей із особливими потребами. Було досліджено ризики для проєкту, що пов'язаний із розробкою програмного забезпечення для Буковинського реабілітаційного центру. Основним завданням даної кваліфікаційної роботи є ідентифікація та аналіз ризиків для проєкту, з метою подальшого коригування планів розробки та структури команди. Для цього було застосовано метод Дельфі, та залучено ряд експертів, які взяли участь у анонімному опитуванні із визначення найбільш ймовірних негативних факторів. За результатами даного опитування було зведено загальну статистику, розроблено шляхи реагування на вибрані ризики та проведено факторний та кореляційний аналізи для ризиків.

**Ключові слова:** проєктний менеджмент, управління ризиками, метод Дельфі, факторний аналіз, кореляційний аналіз.

## *Annotation*

This work is devoted to the process of organizing of the development and risks management in the project for people with special needs. The risks for the project related to the development of software for the Bukovinian rehabilitation center were investigated. The main objective of this qualification work is to identify and analyze the risks for the project, in order to further adjustment of the development plans and team structure. To do this, Delphi's method was applied, and a number of experts were involved who took part in an anonymous survey to determine the most likely negative factors. Based on the results of this survey, general statistics were compiled, ways of reacting on selected risks were developed, and factor and correlation analyses for risks were conducted.

**Keywords:** project management, risk management, Delphi method, factor analysis, correlation analysis.

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів наукових досліджень інших авторів мають посилання на відповідне джерело.

\_\_\_\_\_ М.І. Гасюк  
(підпис)

## ЗМІСТ

ВСТУП .....	3
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ УПРАВЛІННЯ ПРОЄКТАМИ .....	4
1.1. Управління проєктами як системна діяльність: поняття, структура, життєвий цикл, основні артефакти .....	4
1.2. Основні парадигми менеджменту розробки програмного забезпечення ..	12
РОЗДІЛ 2 РИЗИКИ В ІТ ПРОЄКТАХ ТА ЇХ ПЛАНУВАННЯ .....	20
2.1 Управління ризиками в проєкті: поняття, ідентифікація, стратегії подолання .....	20
2.2 Методи оцінювання ризиків: Методи Дельфі та матриці ймовірності .....	26
РОЗДІЛ 3 ОЦІНКА РИЗИКІВ В ПРОЄКТІ ДЛЯ ЛЮДЕЙ З ОСОБЛИВИМИ ПОТРЕБАМИ .....	30
3.1 Постановка задачі .....	30
3.2 Організація та результати опитування експертів щодо оцінки ризиків в проєкті для людей з особливими потребами .....	31
3.3 Виявлення ймовірності настання ризику методом Дельфі .....	34
3.4 Аналіз ризиків, їх взаємозв'язок та фактори впливу на проєкт .....	35
ВИСНОВКИ .....	40
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....	41

## ВСТУП

Управління розробкою та ризиками проєктів є визначальними факторами успіху в будь-якій сфері людської діяльності. Управління проєктами та ризиками у ІТ-проєктах набувають особливого значення в умовах жорсткої конкуренції у сфері ІТ-послуг, швидкої зміни користувацьких уподобань, стрімкого розвитку технологій і відповідно старіння кінцевого продукту. Окрім об'єктивних змін є чинники і суб'єктивного характеру, які значно впливають на конкурентоспроможність продуктової ІТ-компанії – якість процесів управління в компанії, організаційна культура, людський фактор, кваліфікація розробників, злагодженість командної роботи та безліч інших. Саме усі вони виступають джерелом ризиків, які потенційно, за сприятливих умов, можуть чинити негативний вплив на кінцевий результат проєкту, або взагалі створити умови, в яких його завершення буде неможливим. Це обумовлює необхідність приділення окремої уваги ризикам, адже вони визначають міру невпевненості в одержанні очікуваного результату, прибутку компанії і найголовніше - якості і витребуваності кінцевого продукту.

Метою мого дослідження є визначення змісту та оцінки ймовірності настання ризиків під час реалізації веб-додатку для людей із особливими потребами. Досягнення мети вбачається можливим шляхом реалізації низки завдань: визначення сутності процесу управління проєктами з точки зору системного підходу: поняття, структура, життєвий цикл, основні артефакти; опрацювання і узагальнення змісту основних парадигм менеджменту розробки програмного забезпечення; дослідження сутності змісту процесів управління ризиками. Узагальнення теоретичних напрацювань дозволить реалізувати практичну частину роботи, завданням якої є проведення комплексної оцінки ризиків в проєкті - розробки веб-додатку для людей з особливими потребами, розробці відповідних шляхів реагування на них.

## РОЗДІЛ 1

### ТЕОРЕТИЧНІ ОСНОВИ УПРАВЛІННЯ ПРОЄКТАМИ ТА РИЗИКАМИ

#### 1.1. Управління проєктами як системна діяльність: поняття, структура, життєвий цикл, основні артефакти

Поняття “проєктного менеджменту” або ж “управління проєктами” в ІТ-індустрії зазнало суттєвих змін за останні 30 років, що були періодом активного розвитку інформаційних технологій. Із появою нових технологій у проєктних менеджерів з’явилися нові можливості, аби мати змогу більш якісно планувати, виконувати та відслідковувати прогрес розробки програмних продуктів. Найбільш вагомими змінами можна відзначити:

#### 1. Поява Agile (гнучкої) методології. Agile Manifesto був опублікований у лютому 2001 року, і є документом, який визначає 4 ключові цінності:

- а. Люди та співпраця важливіші за процеси та інструменти;
- б. Працюючий продукт важливіший за вичерпну документацію;
- в. Співпраця із замовником важливіша за обговорення умов контракту;
- г. Готовність до змін важливіша за дотримання плану [7];

та 12 принципів філософії гнучких методологій:

- а. Найвищим пріоритетом для нас є задоволення потреб замовника, шляхом завчасного та регулярного постачання програмного забезпечення;
- б. Схвальне ставлення до змін, навіть на заключних стадіях розробки. Agile-процеси надають можливість використовувати зміни задля забезпечення конкурентоспроможності замовника;
- в. Працюючий продукт слід випускати якомога частіше, з періодичністю від пари тижнів до пари місяців;
- г. Впродовж усього проєкту розробники і представники бізнесу повинні працювати разом щодня;

- д. Над проектом повинні працювати вмотивовані професіонали. Щоб робота була виконана, створіть їм умови, надайте підтримку і повністю на них покладіться;
  - е. Особиста комунікація – найефективніший та найпрактичніший метод як донести інформацію до команди, так і поширити її всередині;
  - ж. Працюючий продукт – головний показник прогресу;
  - з. Інвестори, розробники і користувачі повинні мати можливість підтримувати постійний ритм як завгодно довго. Agile допомагає налагодити такий сталий процес розробки;
  - и. Постійна увага до технічної досконалості і якості проектування підвищує гнучкість проекту;
  - к. Простота – мистецтво мінімізації зайвої роботи – вкрай необхідна.
  - л. Найкращі вимоги, архітектурні та технічні рішення виникають у командах, що здатні самоорганізовуватись;
  - м. Команда регулярно намагається знайти способи підвищення ефективності та відповідно корегує свою роботу [4].
2. Поява хмарних технологій. Дані технології підвищили якість доступності та ефективності зберігання даних та забезпечили вищий рівень безпеки для інформації. Хмарні технології дали можливість командам розробників використовувати певні готові рішення у розробці проєктів та зосереджувати свою увагу лише на розробці продукту, а не на інфраструктурі.
  3. Поява штучного інтелекту. Штучний інтелект це чудова можливість для проєктного менеджменту, аби покращити якість свого функціонування за допомогою використання різноманітних інструментів для прогнозування, аналізу даних, автоматизації тощо.
  4. Ресурси для віддаленої роботи. Із активним поширенням та популяризацією мережі-інтернет, працівники більшості ІТ-компаній отримали можливість працювати у віддаленому режимі із будь-якої точки

світу. Ця нагода створила нові виклики для проєктних менеджерів та розробників, оскільки їм доводиться організовувати комунікації та співпрацю, а також координувати роботу команди на новому рівні.

Отже, бачимо, що у результаті таких вагомих змін, процес розробки та менеджменту програмного забезпечення став більш гнучким та швидким, завдяки використанню нових методологій і зосередженості на якості та ефективності програмного продукту [17].

Варто зазначити, що згідно із дослідженнями Project Management Institute (PMI), із подальшим розвитком інформаційних технологій та штучного інтелекту суттєво зросте потреба у робочій силі, орієнтованій на сферу проєктного менеджменту. Згідно із статистикою, до 2027 року роботодавці потребуватимуть більше 88 мільйонів проєктних менеджерів у різних сферах діяльності [19].

Проєкт – тимчасова, унікальна та прогресивна спроба отримання матеріального або нематеріального результату у вигляді продукту, послуги, конкурентної переваги тощо. Він включає в себе групу взаємопов'язаних завдань, які є запланованими для виконання протягом визначеного періоду в межах певних вимог та обмежень [31].

Основними обмеженнями проєкту є:

- Час – графік виконання проєкту;
- Вартість – бюджет, виділений на виконання усіх поставлених задач;
- Обсяг робіт – конкретно окреслені завдання для реалізації;
- Якість – стандарти для результатів проєкту [38].



Рисунок 1.1.1 - Проектні обмеження

Процес управління будь-яким проєктом повинен вестися на основі системного підходу.

Системний підхід передбачає розгляд проєкту як системи, що створена із



конкретною ціллю. Для того, аби краще пояснити сутність системи у даному випадку, можна провести аналогію системи із автомобілем. Машина має певні функції та основне призначення – переміщення із одної точки в іншу, використовуючи сети складних компонентів та синаптичних механізмів: запуск системи запалювання, натискання на педаль зчеплення, встановлення першої передачі, натискання на педаль акселератора та, як наслідок, приведення автомобіля в рух. Ідеєю такої аналогії є те, що усі елементи системи працюють разом задля досягнення єдиного результату, при цьому кожен із них має свої унікальні функції та завдання, але усі роблять свій внесок у загальний план [5].

Гнучкість також є ключовим фактором у системному управлінні проєктами. Вміння швидко адаптовуватись до нового середовища, умов та інших змін допомагає залишатись конкурентоспроможним, а також ефективно та швидко приймати рішення [16].

Системний аналіз – дисципліна, що вивчає складні системи та їх взаємодію із навколишнім середовищем. Системний аналіз дає змогу визначати вимоги та потреби користувачів, створювати якісні системи та забезпечувати їх безперебійну роботу. Також, даний підхід дозволяє оцінювати зміни та їх вплив загалом на систему, на окремі її складові та на інші системи, з якими відбувається взаємодія [25].

Дамо визначення деяким загальним поняттям, які необхідні для повного розуміння понять “системного підходу”, “системного аналізу” та “управління проєктами”.

Система – набір взаємопов’язаних між собою елементів, які функціонують разом, аби досягнути певного результату. Системи можуть бути фізичними чи абстрактними, складними і простими. В ІТ-сфері система часто являє собою певне програмне забезпечення, яке виконує конкретну функцію: автоматизація процесів, збереження даних, менеджмент проєктів тощо [24].

Процес - послідовність дій або операцій, які виконуються для того, щоб досягти конкретного результату. Процеси можуть мати різну складність та тривалість в залежності від характеру поставленої мети [15].

Управління – процес планування, координації, контролю та керування ресурсами задля досягнення певного результату. Процес управління включає встановлення цілей та завдань, вибір стратегій і розробку планів, вирішення проблем, що виникають в ході ведення проєкту, контроль виконання плану та моніторинг використаних та запланованих ресурсів тощо [36].

Система управління проєктом – комбінація методологій та технологій управління проєктами, які допомагають в плануванні, організації, контролі проєктів тощо. Система управління проєктами передбачає взаємодію усіх елементів та процесів проєкту, що пов'язані між собою, задля того, щоб досягти очікуваного результату в раніше встановлених часових та фінансових межах [32].

Управління процесом – планування, виконання, контроль та аналіз процесу із метою досягнення конкретного результату. Поняття “управління процесом” також включає в себе встановлення мети та показників ефективності для процесу, характеристику його основних фаз, встановлення відповідальних за виконання та моніторинг процесу осіб тощо [12].

Розглянемо поняття системи управління проєктами більш детально.

Система управління проєктами складається із декількох підсистем, яким відведено конкретну роль, а саме: Фасилітативна організаційна підсистема. Підсистема планування. Підсистема контролю. Інформаційна підсистема. Підсистема керування ризиками. Підсистема управління змінами.

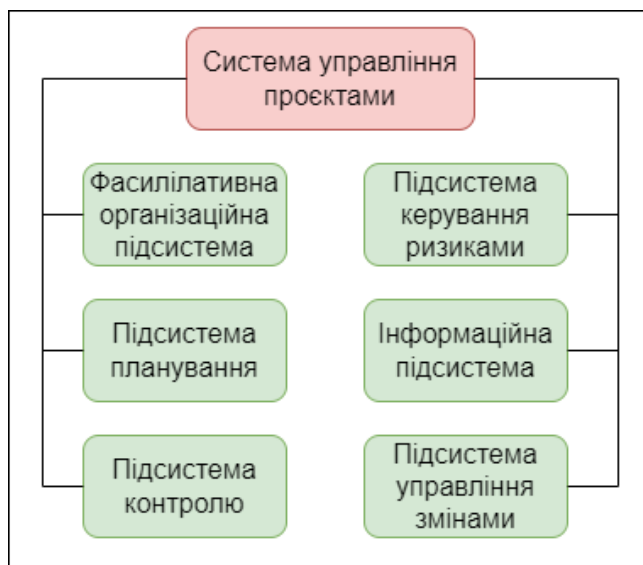


Рисунок 1.1.2 - Структура управління проєктами

Кожна підсистема має свою особливу, унікальну та надзвичайно важливу роль, структуру та функції, що є рушієм для досягнення мети та відповідних кінцевих результатів проєкту [9].

Розглянемо поняття “життєвого циклу управління проєктами” та охарактеризуємо кожен із фаз життєвого циклу управління проєктами.

Життєвий цикл управління проєктами – це по черговість фаз управління проєктами, які є необхідними до виконання для того, аби мати можливість успішно завершити роботу над проєктом. Згідно із затвердженими стандартами проектного менеджменту, які були розроблені Інститутом Управління Проєктами (PMI), існує 5 основних етапів управління проєктами: Ініціація. Планування. Виконання. Моніторинг та контроль. Завершення проєкту [38, 11].



Рисунок 1.1.3 – Життєвий цикл управління проєктами

Розглянемо поняття артефактів в управлінні проєктами та основні необхідні для роботи над проєктом артефакти.

Артефакт – документ, який призначений для того, щоб зберігати, організувати та ділитися інформацією в процесі управління проєктом. Згідно із PMBOK Guide 7<sup>th</sup> edition, що містить стандартну термінологію та керівні принципи проектного менеджменту, артефакт є шаблоном, документом або результатом проєкту [8].

Кожному етапу життєвого циклу управління проєктами характерне створення відповідних артефактів. Сформуємо такі відповідності за допомогою таблиці:

Таблиця №1 – Артефакти управління проєктами

Етап	Назва артефакту	Тип	Вміст
Ініціація	Бізнес-план (Business case)	Стратегія	Цінність запропонованого проєкту, що включає фінансову та нефінансову вигоду

Ініціація	Короткий опис проекту (Project brief)	Стратегія	Опис загальної мети, очікуваних результатів та процесів проекту
	Статут проекту (Project charter)	Стратегія	Містить загальну інформацію про проект і підтверджує початок проекту
	Реєстр ризиків (Risk register)	Журнали та реєстри	Оцінка ризиків, ймовірність їх настання, відповідальні особи
	Реєстр зацікавлених сторін (Stakeholder register)	Журнали та реєстри	Інформація про стейкхолдерів, оцінка їх впливу на хід проекту та класифікація за їх функціями
	Статут команди проекту (Project team charter)	Інші	Цінності команди проекту, структуризація команди, домовленості, і керівні принципи роботи
Планування	Roadmap	Стратегія	Деталізована часова лінія, що містить заплановані фази розробки (milestone), вагомі події, точки прийняття рішень тощо
	Backlog	Журнали та реєстри	Впорядкований список робіт, що повинні бути виконаними
Планування	Журнал змін (Change log)	Журнали та реєстри	Повний список змін, що були подані під час розробки проекту та дані про їх статус
	Журнал проблем і помилок (Issue log)	Журнали та реєстри	Інформація про актуальні проблеми та помилки.
	План контролю змін (Change control plan)	План	Опис процесу, що пов'язаний із способом внесення змін та їх контролю
	План управління комунікацією (Communication management plan)	План	Інформація про те, як, коли і ким буде поширюватись та адмініструватись відповідна інформація
Планування	План управління витратами (Cost management plan)	План	Інформація щодо того, як витрати плануватимуться, будуть структуровані та контрольовані
	План управління закупівлями (Procurement management plan)	План	Опис того, які сервіси, послуги чи товари необхідні для роботи над проектом
	План управління проектом (Project management plan)	План	Характеристика того, як проект виконуватиметься,

			контролюватиметься та закриватиметься
	План управління якістю (Quality management plan)	План	Міра застосування керівних принципів, процедур та політик, що спрямовані на досягнення відповідних показників якості
	План управління вимогами (Requirements management plan)	План	Документація, аналіз та управління вимогами
	План управління ресурсами (Resource management plan)	План	Опис того, як проектні ресурси набуваються, виділяються на певний етап проекту, моніторяться і контролюються
	План управління ризиками (Risk management plan)	План	Опис того, як буде управління ризиками буде структуроване та здійснене
	Розклад структури продукту (Product breakdown structure)	Ієрархічні діаграми	Ієрархічна структура, що є репрезентацією продуктних компонент та очікуваних результатів
Планування	Розклад структури робіт (Work breakdown structure)	Ієрархічні діаграми	Ієрархічна декомпозиція загального обсягу робіт, який має бути виконаний командою для досягнення цілей проекту та створення необхідних результатів
	Бюджет (Budget)	Базові	Затверджений кошторис проекту на основі WBS
Планування	Графік за фазами (Milestone plan)	Базові	Графік етапів розробки проекту та планові дати
	Графік проекту (Project schedule)	Базові	Опис заходів, що відбудуться в заплановані дати, тривалості етапів і процесів тощо
	Діаграма Ганта (Gantt chart)	Візуалізація даних	Графік, що відображає тривалість і графіки робіт. Зручний для того, аби легко слідкувати за відповідністю запланованих часових та бюджетних меж до актуальних
	Звіт про якість	Звіт	Висновки щодо відповідності якості, рекомендації для коригування відхилень та вдосконалення проекту
	Звіт про ризики	Звіт	Підсумки про індивідуальні проектні ризики, оцінка

			загального рівня проектного ризику
	Звіт про статус	Звіт	Підсумки щодо прогресу з моменту останнього аналогічного звіту щодо бюджету та часових ресурсів

Отже, бачимо, що існує велика кількість артефактів, які різняться за своїм призначенням та вмістом, але кожен із них є важливим для успішного управління проектом. Артефакти є необхідними для якісного збереження інформації, забезпечення зрозумілості, прозорості, відповідної якості та високої ефективності управління проектом [30].

## 1.2. Основні парадигми менеджменту розробки програмного забезпечення

Програмне забезпечення (ПЗ) – набір інструкцій, програм та даних, які є призначеними для того, щоб виконувати певні конкретні функції на пристроях, для яких вони створені. Даний термін був визначений для диференціації програмного забезпечення від апаратного забезпечення – фізичних компонент електронних пристроїв. Саме програмне забезпечення керує апаратним для того, щоб електронний пристрій виконував свої необхідні користувачу завдання.

Програмне забезпечення, як правило, зберігається на таких зовнішніх пристроях довготривалої пам'яті, як жорсткий диск, магнітна дискета тощо. В той час, коли програмна активна у використанні, комп'ютер зчитує її із накопичувача, після чого зчитані інструкції тимчасово розміщуються у пам'яті випадкового доступу (RAM). Процес зберігання та виконання інструкцій електронним пристроєм має назву “запущеної (виконуваної) програми” [23].

Програмне забезпечення зазвичай складається із кількох компонент, до яких входить вихідний код, програми, різноманітні бібліотеки та модулі, конфігураційні файли, документація та інші файлові ресурси. Воно має можливість вдосконалення, розширень або ж оновлення, у разі виникнення таких потреб та вимог у користувача.

Існує 2 основні типи програмного забезпечення, а саме: системне та прикладне (апаратне). Основною функцією системного програмного забезпечення є управління та контроль внутрішніми функціями електронного пристрою через операційну систему, контроль периферійних пристроїв, наприклад – моніторів, принтерів, пристроїв зберігання даних тощо. Функцією ж прикладного програмного забезпечення є направлення комп'ютера на виконання заданих користувачем команд [33].

Програмне забезпечення може бути написаним, використовуючи різні мови програмування: Java, Python, C++, PHP, React Native тощо, та різні платформи, такі як: Windows, macOS, Linux, Android, IOS, Web. Існує безліч мов програмування та платформ, які призначені для створення програмних продуктів. Кожна із них має як переваги, так і недоліки, тому важливим фактором є підбір тієї платформи або мови програмування, яка найбільше відповідає запитам та вимогам проєктної мети.

Розробка програмного забезпечення – діяльність, направлена на створення, впровадження та проектування програм для електронних пристроїв. Основною ціллю цього процесу є покращення функціонування певних електронних пристроїв, розширення їх можливостей, покращення якості їх роботи, підвищення ефективності виконання необхідних завдань та багато іншого. Процедура розробки програмного забезпечення має загальну структуру, яка підходить більшості програмних продуктів: визначення потреб, аналіз вимог, створення дизайну, розробка та впровадження, тестування, підтримка [42].

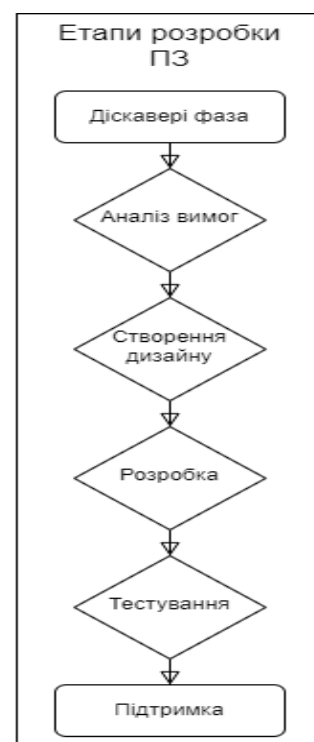


Рисунок 1.2.1 - Структура розробки ПЗ

Розробка програмного забезпечення є надзвичайно важливим процесом у сучасному світі, оскільки програми є широко використовуваними у всіх сферах суспільного життя – освітній, науковій, фінансовій, медичній, розважальній сферах тощо. Розробка

програмного забезпечення активно вдосконалювалась і зазнала ряд важливих змін за останні 30 років, а саме:

- Поява гнучких (Agile) методологій;
- Поява мережі-Інтернет;
- Розвиток штучного інтелекту;
- Розробка відкритого програмного забезпечення [14].

Менеджмент розробки програмного забезпечення – це процес управління розробкою програмного забезпечення, а саме керування плануванням, виконанням, моніторингом та доставкою програмних продуктів [13].

Отже, з огляду на аспекти, згадані вище, можна зробити висновок, що менеджмент розробки програмного забезпечення є необхідною складовою розробки програмного забезпечення, від якої залежить якість кінцевого продукту.

Менеджмент розробки програмного забезпечення може мати різну структуру в залежності від того, яка методологія використовується у даному процесі. Методологія управління проектом – це структурований підхід до планування, виконання, моніторингу і контролю проекту із метою його успішного завершення, із дотриманням усіх обмежень. Вибір методології впливає на ефективність управління проектом, і залежить від таких факторів, як масштабність проекту та рівень його складності, обсяг робіт, часові межі, потреби та вимоги клієнта, рівень кваліфікації та досвіду членів команди проекту тощо. Немає єдиного універсального вірного рішення для вибору тієї чи іншої методології для проекту, тому для здійснення такого вибору необхідним є ретельний аналіз та дослідження усієї отриманої та зібраної інформації на етапі діскавері та фазі ініціації.

В загальному, існує 2 основні категорії методологій управління проектами, а саме традиційна (Waterfall) та гнучка (Agile). Розглянемо детальніше кожен із них:



1. **Методологія Waterfall.** Це методологія управління проєктами, яка полягає у послідовному виконанні етапів проєкту. Вона була представлена у 1970 році Вінстоном В. Ройсом і складається із 6-ти етапів менеджменту проєкту, причому дані етапи мають жорстку залежність один від одного. Жоден із подальших етапів не може початися раніше, ніж закінчиться попередній для нього [44; 43];



Рисунок 1.2.2 - Етапи розробки ПЗ методології Waterfall

2. **Методологія Agile.** Це ітеративна методологія розробки програмного забезпечення, яка передбачає часті ітерації розробки та тестування програмного продукту та співпрацю між клієнтом та проєктною командою. Гнучка методологія була представлена у 2001 році 17-ма технологіями у вигляді “Agile Manifesto”. Принциповим аспектом даної методології є вміння пристосовуватись та швидко вносити зміни і реагувати на них. Важливою складовою гнучкої методології є постійна комунікація із замовником та фідбек від користувачів, що дозволяє робити зміни та вдосконалювати програмний продукт із кожною ітерацією, робити його більш зручним у користуванні та більш конкурентоспроможним і актуальним на ринку. Agile методологія передбачає розгляд проєкту як набору задач, розділених на ітерації, які мають відповідальних осіб у вигляді учасників проєкту, і часові та фінансові обмеження. Дуже зручним та важливим є те, що замовник може побачити і оцінити роботу програмного забезпечення після закінчення кожної ітерації, а також внести свої корективи у плани роботи в наступних ітераціях, якщо це потрібно [22, 27].

Отже, гнучкі методології є правильним вибором для управління проєктами, які мають велику кількість невизначеностей та нестабільність вимог, високий рівень технологічної складності та багато змін. Agile методології є чудовим інструментом менеджменту у проєктах, де передбачена регулярна комунікація

між замовником та проєктною командою, присутня зосередженість на потребах та інтересах користувачів, ринок є відносно нестабільним і зміни є частими та значними.

Узагальнимо основні відмінності Waterfall та Agile методологій у вигляді таблиці:

Таблиця №2 – Основні відмінності Waterfall та Agile методологій

Характеристика	Agile методологія	Waterfall методологія
Обсяг робіт	Може змінюватися кожен ітерацію і не плануватися заздалегідь, а створюватися безпосередньо перед початком роботи над ітерацією	Планується на початку проєкту і не є гнучким, жорстко обмежений у часі та бюджеті
Структура	Ітераційна	Послідовна
Комунікація	Регулярна комунікація із замовником проєкту по ходу роботи над програмним продуктом	Комунікація лише на початку нових етапів розробки програмного забезпечення
Цілі проєкту	Гнучкі, можуть коригуватися в процесі розробки програмного забезпечення	Жорстко визначені
Зміни	Легке та швидке впровадження змін	Складне та дорогівартісне впровадження змін
Тестування	Регулярне тестування, що проводиться по ходу роботи над проєктом	Тестування є окремим етапом, який проводиться після фази реалізації
Частота виконання фаз методології	Фази можуть повторюватись по декілька разів	Робота над фазою відбувається лише 1 раз
Тип контракту	Нефіксована вартість або контракт “Час та вартість (time and materials)”	Фіксована вартість

Отже, кожна із наведених методологій може бути корисною та ефективною в разі її використання в управлінні проєктом, який характеризується ознаками, що притаманні тій чи іншій із методологій, тому вибір повинен бути добре обміркованим та базуватися на ретельному аналізі вимог та завдань конкретного окремого проєкту [6].

Як бачимо, управління розробкою програмного забезпечення більше фокусується на гнучких методологіях, і саме вони користуються найбільшим

попитом у сучасному світі. На це є низка причин, описаних вище. Тому розглянемо тему гнучких методологій та познайомимося більш деталізовано із її видами та їх характеристиками. Отже, до основних найбільш популярних видів Agile методологій належать:

1) Scrum. Основою цієї методології є робота над проєктом невеликими ітераціями, що мають назву “спринт”, і зазвичай тривають від 1-го до 4-х тижнів. Кожен спринт починається із планування і визначення обсягу робіт, встановлення часових меж. Відбуваються регулярні зустрічі між учасниками команди, а також із замовником, аби підтримувати якісну співпрацю та планувати наступні кроки. Даний метод передбачає високий рівень самоорганізації в команді, оскільки учасники самі колективно вирішують яким чином виконувати отримані завдання та як правильно розподілити роботу між собою. Список задач за методологією Scrum має назву “беклог (backlog)” та оновлюється на регулярній основі, зокрема на початку кожного спринта, коли команда приймає рішення щодо того, які завдання будуть виконаними в наступну чергу [41].

2) Kanban. В основі цієї методології лежить принцип “потoku (flow)” та візуалізації роботи. Головним інструментом Kanban-методології є Kanban-дошки, що мають вигляд декількох стовпчиків, що інтерпретують етапи розробки програмного продукту, і картки в них, що відображають ту чи іншу задачу. Таким чином легко відслідковувати прогрес роботи та перевіряти статуси задач. Принцип неперервного потоку передбачає постійний рух карток по Kanban-дошці, згідно із етапом виконання відповідного завдання, що забезпечує високий рівень візуалізації. Також, однією із практик цієї гнучкої методології є обмеження WIP (work in progress), який полягає в тому, що учасники команди не мають можливості брати на себе нові завдання, до закінчення поточних, а також ця практика передбачає роботу учасника не більше ніж над 2-ма завданнями одночасно. Тому, очевидно, методологія Kanban найбільше підходить для проєктів, які мають багато змін, відкриті до процесу постійного вдосконалення та зміни вимог, а також потребують

швидкого процесу доставки та завершення розробки програмного продукту [35].

3) Lean. Основним завданням даної методології є висока ефективність використання ресурсів, зменшення витрат та доставка максимально високої якості програмного продукту замовникові. Основними принципами методології Lean є неперервне вдосконалення якості продукту, ідентифікація потоків роботи для аналізу та виявлення непотрібних дій з метою зменшення фінансових та часових витрат, визначення цінностей клієнта та гарантування максимальної якості програмного забезпечення. Як і інші гнучкі методології управління розробкою ПЗ, дана методологія передбачає високий рівень комунікації між учасниками проєкту, активне внесення та управління змінами, додавання обсягу робіт та функціоналу порціями, аби програмне забезпечення було більш адаптивним і конкурентоспроможним на ринку [29].

4) Extreme Programming (XP). Основним акцентом методології XP є якість та швидкість розробки програмного продукту. Ключовими принципами даної методології є парне програмування, що полягає у спільній роботі двох розробників, причому один пише код, а другий займається його перевіркою, автоматизоване тестування, яке забезпечує швидке виявлення та усунення помилок, а також підтримку високої якості програмного забезпечення, континуальна інтеграція, що передбачає регулярне інтегрування змін у велику базу коду тощо. Методологія XP так само, як інші гнучкі методології має ітеративний характер роботи, і дозволяє вносити зміни до обсягу робіт та оновлювати вимоги до програмного забезпечення [34].

Охарактеризуємо основні властивості найбільш застосовуваних гнучких методологій – Scrum та Kanban у вигляді наступної таблиці [20]:

Таблиця №3 – Властивості гнучких методологій Scrum та Kanban

Властивість	Scrum	Kanban
Ролі	Власник продукту, Scrum майстер, команда розробників	Формальні ролі відсутні

Часові обмеження	Часова лінія поділена на спринти	Можуть змінюватись в ході розробки проекту. Існує єдина часова лінія, яка не ділиться на окремі етапи
Зміни	Можуть бути внесеними і запланованими лише після закінчення поточного спринта	Можуть вноситися негайно
Пріоритезація завдань	Незмінна в ході спринта	Може змінюватися на щоденній основі
Планування та оцінка часових затрат на роботу	Обов'язкові	Необов'язкові
Зустрічі	Фіксовані	Нефіксовані
Власники завдань	Беклог має одного власника у вигляді учасника команди або команди в загальному	Обсяг робіт може бути розподілений між декількома учасниками або командами
Організація роботи	Самоорганізація роботи командою та розподілення задач між ними	Самоорганізація роботи командою та розподілення задач між ними

Отже, як бачимо із порівняльної таблиці, гнучкі методології Scrum та Kanban мають як спільні риси, так і відмінності, але мають однакову мету: швидка розробка та доставка програмного продукту, високий рівень залучення та комунікації команди розробників та клієнта, максимізація якості та постійне вдосконалення програмного забезпечення.

## РОЗДІЛ 2

### РИЗИКИ В ІТ ПРОЄКТАХ ТА ЇХ ПЛАНУВАННЯ

#### 2.1 Управління ризиками в проєкті: поняття, ідентифікація, стратегії подолання

Ризики – непередбачувані обставини або події, які можуть вплинути на проєкт як позитивно, так і негативно. Область впливу ризиків не є обмеженою, тому вони можуть “ставити свій відбиток” на якості та ефективності роботи людських та технологічних ресурсів, процесів тощо. Сфери походження ризиків також можуть відрізнятися, наприклад: технологічна, економічна, соціальна, політична сфери, природні фактори та ін. [39]

Усі ризики характеризуються такими факторами:

- Суть ризику. Інформація про те, яка подія може статися і вплинути на хід розробки проєкту;
- Часові межі. Ймовірні часові проміжки, коли ризик може настати;
- Імовірність. Оцінка того, які шанси настання того чи іншого ризику;
- Вплив. Характеристика наслідків, які можуть бути після настання ризиків;
- Тригери. Інформація про події, які можуть викликати ризики.

Розглянемо типи ризиків, що є важливими для ІТ-проєктів:

1. Технічні ризики. Сюди входять технічні збої, системні помилки та порушення, кібератаки, проблеми із сумісністю та ін.;
2. Фінансові ризики. Характеризуються необхідністю у більшій кількості коштів, аніж було заплановано. Це є результатом неточного збору вимог та планування, що призводить до незадоволення клієнта, або до неможливості успішного завершення проєкту;
3. Ризики, пов’язані із графіком робіт. Неправильне планування призводить не тільки до додаткових фінансових затрат, а й до зсунення графіку робіт, і, як наслідок, пропуску запланованих дедлайнів;

4. Ризики, пов'язані із результатом. Цей тип ризиків є областю відповідальності і сторони виконавця, і сторони замовника. Велику роль для цього типу ризиків грає вміння пояснювати та конкретизувати – з боку клієнта, та збирати і узгоджувати – з боку розробника, вимоги до програмного продукту. Але варто зазначити, що жоден із учасників проєкту не може передбачити чи користуватиметься популярністю кінцевий продукт, і чи буде він актуальним на момент релізу;
5. Організаційні ризики. Тригером для цієї групи ризиків є низька якість організації роботи, неправильний поділ та призначення відповідальних осіб для проєктних задач, погано продумані плани закупівлі та розподілу ресурсів тощо;
6. Комунікаційні ризики. Найбільш поширеними ризиками цієї категорії є поява непорозумінь, відсутність зв'язку між учасниками проєкту та ін., що можуть бути викликаними через недостатню чіткість і ясність вираження ідей та думок членами команди, мовні бар'єри, культурні бар'єри тощо;
7. Ризики, пов'язані із обсягами робіт. Часто у замовників виникають нові ідеї, потреби і бажання по ходу розробки продукту, або ж змінюються потреби бізнесу чи ринку, тому виникає необхідність у тому, аби виходити за межі початково встановленого обсягу робіт. Наслідком цього ризику можуть бути збільшення фінансових затрат, зсув графіку, зниження якості продукту тощо;
8. Ризики, пов'язані із навичками команди розробників. Різноманітні зміни, виникнення технічних проблем та інших факторів, можуть призвести до того, що команда не буде містити в собі достатньо кваліфікованих та досвідчених кадрів для вирішення відповідного питання [1].

Для глибшого розуміння, наведемо приклади ризиків, які можуть настати під час роботи над програмним продуктом:

- Збій або несправність програмно-технічних засобів;

- Шпигунські та шкідливі програми. Це означає захоплення систем за допомогою даних програм, викрадення конфіденційної інформації або її використання проти компанії-власника проєкту;
- Хакерство. Злом системи контролю, а отже отримання хакером доступу до усіх файлів та важливої інформації;
- Стихійні лиха;
- Незадоволені потреби клієнта. Цей ризик може бути зумовленим різними причинами: непрофесіоналізм компанії-розробника, застарілість технічних навичок її робітників, неякісний збір вимог та потреб замовника тощо;
- Втрата даних. Якщо система є вразливою, то наслідком може стати утрата даних через нестабільність системи, або крадіжка даних. Даний ризик несе за собою тяжкі наслідки у вигляді штрафів або ж відкриття юридичних дій різного характеру в залежності від конкретного випадку;
- Втрачені можливості. Причиною цього ризику є відсутність моніторингу змін на ринку та технологічних трендів з боку компанії-виконавця, що призводить до неактуальності програмного продукту або його застарілості [26].

Отже, бачимо, що існує велика кількість ризиків, які можуть настати під час роботи над проєктом. Усі вони різняться за джерелом виникнення, ступенем впливу та методами вирішення, але повинні бути детально розглянуті для успішного ведення та завершення проєкту. Тому виникає необхідність у менеджменті проєктних ризиків, що є наступним предметом розгляду.

Управління ризиками – це процес виявлення, аналізу, попередження та реагування на ризики, що виникають протягом життєвого циклу проєкту. Основним завданням процесу управління ризиками є якісне планування ризиків, аби попередити виникнення проблем, з якими, зазвичай, набагато важче боротися.



Управління ризиками є необхідним і корисним не лише у виявленні та усуненні негативних ризиків, але й в ідентифікації позитивних ризиків, тобто певних можливостей до покращення якості роботи, окремого продукту чи вартості бізнесу в загальному. Менеджмент ризиків передбачає вивчення взаємозв'язків між ризиками та їх каскадним впливом на хід проєкту, тому варто складати та розглядати загальну “картину” системи ризиків та використовувати цілісний підхід при їх аналізі [40].

Як і кожен процес, управління ризиками проєкту має визначені етапи, а саме:

1. Ідентифікація ризиків. Метою цього етапу є визначення усіх можливих ризиків, що можуть виникнути під час життєвого циклу проєкту;
2. Аналіз ризиків та оцінка ймовірності їх настання. Визначення джерел ризиків, значимості їх наслідків, оцінки імовірності їх настання та ступінь впливу на проєкт;
3. Пріоритезація ризиків, базуючись на бізнес-цілях. До даної фази входять планування ризиків, розробка плану управління ними для того аби їм запобігти або зменшити їх вплив;
4. Реагування на ризики. На цьому етапі план управління ризиками починає діяти, тому реалізуються усі заплановані заходи щодо менеджменту ризиків;
5. Моніторинг та адаптація. Хоча цей етап наведений останнім у списку фаз менеджменту ризиків проєкту, він не є завершальним. Моніторинг є обов'язковим та постійним процесом під час управління ризиками. Необхідним є відслідковування та контролювання ризиків протягом усього процесу роботи над програмним продуктом, аби могли вчасно та якісно приймати певні рішення і заходи щодо усунення або запобігання ризиків [2].

Для того, аби могли ефективно вести та управляти усіма перерахованими етапами менеджменту ризиків в проєкті, необхідно вести відповідну проєктну

документацію, що дозволить якісно збирати, аналізувати та зберігати важливу інформацію. Розглянемо артефакти, пов'язані із управлінням ризиками:

- Беклог, із урахуванням ризиків (Risk-adjusted backlog). Містить обсяги робіт із урахуванням роботи та певних дій над усуненням загроз та використанням можливостей;
- Реєстр ризиків (Risk register). Включає в себе загальний опис ризику, його оцінку, міру впливу та відповідальну особу за менеджмент відповідного ризику;
- План управління ризиками. Характеристика дій для запобігання або план усунення того чи іншого ризику, структуризація процесів боротьби із ними;
- Структурне розбиття ризиків (Risk breakdown structure). Даний артефакт має вигляд діаграми, що являє собою ієрархічне представлення потенційних джерел ризиків;
- Звіт про ризики (Risk report). Цей документ ведеться протягом усього життєвого циклу проєкту і підводить підсумки щодо менеджменту ризиків проєкту, а також щодо рівня загального проєктного ризику.

Отже, можемо зробити висновок, що документація менеджменту ризиків проєкту є необхідною для якісного та ефективного виконання даного процесу, і грає ключову роль у моніторингу і контролі проєктних ризиків.

Для того, аби мати можливість управляти ризиками, спершу потрібно вміти їх виявляти та аналізувати. Найперший крок, потрібний для ідентифікації ризиків – збір інформації про проєкт. Сюди входить інформація про проєктну ціль, обсяги робіт та обмеження у вигляді часу, бюджету, ресурсів тощо. Також важливо визначити якомога більше зацікавлених сторін – стейкхолдерів, які можуть тим чи іншим чином впливати на хід розвитку проєкту. Дуже корисним фактором у ідентифікації ризиків є використання досвіду із минулих проєктів командою розробників, оскільки деякі проєкти можуть мати схожі ризики або взагалі ідентичні, тому компанія-виконавець

може застосувати вже розроблені техніки усунення чи запобігання ризикам, які забезпечили успішний менеджмент ризиків проєкту в попередній роботі. Не менш важливим є консультаційний процес із спеціалістами конкретної галузі, для якої розробляється програмне забезпечення, а також загальні збори команди для брейнштурмінгу і генерації нових ідей щодо виявлення, запобігання та усунення ризиків.

Для аналізу ризиків в ІТ-проєктах найчастіше використовуються такі моделі аналізу, як: SWOT-аналіз та PESTLE-аналіз. Розглянемо кожну із них більш детально.

SWOT-аналіз - інструмент для виявлення внутрішніх слабких сторін проєкту та оцінки внутрішніх загроз та можливостей.

- S – Strength. Сильні внутрішні сторони, позитивні атрибути проєкту – технології, фінанси, високий рівень навичок команди, знання тощо;
- W – Weakness. Слабкі внутрішні сторони, тобто певні негативні фактори, які потрібно викоринити або вдосконалити для більш ефективної роботи над проєктом;
- O – Opportunity. Можливості – зовнішні позитивні фактори, які можуть посприяти успіху проєкту;
- T – Threat. Загрози – зовнішні фактори, на які не можна вплинути. Потрібно розробити план дій для випадку справдження цих обставин [28].

PESTLE-аналіз - інструмент для виявлення зовнішніх факторів, які можуть вплинути на проєкт та процес його планування.

- P – Political. Політичні фактори, прикладом можуть слугувати податкова політика, фіскальна політика, торговельні штрафи та ін.;
- E – Economical. Економічні фактори, до яких належать інфляція, відсоткові ставки, валютні курси тощо;
- S – Social. Соціальні фактори, предметом уваги яких є культурні тенденції, демографія, аналітика населення тощо;

- T – Technological. Технологічні фактори стосуються появи нових технологій, інноваційної діяльності на ринку, різноманітних досліджень і розробок;
- L – Legal. Юридичні фактори включають закони, які можуть вплинути на проєкт тим чи іншим чином;
- E – Environmental. Екологічні фактори – це усі фактори, що пов’язані із навколишнім середовищем, і зазвичай обмежені такими складовими, як клімат, погода, географічне положення та ін. [37]

Отже, SWOT-аналіз та PESTLE-аналіз є важливими моделями для аналізу ризиків проєкту, і допомагають у виявленні та оцінюванні як внутрішніх, так і зовнішніх факторів, які мають вплив на проєкт.

## 2.2 Методи оцінювання ризиків: Методи Дельфі та матриці ймовірності

Як було зазначено раніше, процес ідентифікації, прогнозування та оцінки ризиків є надзвичайно важливим для того, аби мати змогу якісно ними керувати та ефективно вести розробку проєкту. Розглянемо один із методів, що забезпечує даний процес, а саме метод Дельфі.

Метод Дельфі був створений у США на початку Холодної війни для того, аби дати його користувачам змогу прогнозувати вплив технологій на ведення війни. Тобто початковою ціллю даного методу був збір експертної інформації задля забезпечення процесу прогнозування наукових технологій, застосовуваних в ході війни.

Завдяки даному методу було відкрито нові перспективи оцінки ризиків, оскільки метод полягає у залученні експертів із різних областей, і, як наслідок, отримання загальної об’єктивної оцінки. Метод Дельфі дозволяє ідентифікувати, прогнозувати і оцінювати ризики більш обґрунтовано та об’єктивно, тому у команди проєкту виникає можливість якісного планування та боротьби із ризиками, застосування ефективних заходів для зменшення їх впливу та наслідків. Метод Дельфі складається із таких основних етапів:

1. Визначення проблеми чи аспекту, який потребує оцінки ризиків;

2. Вибір експертів, які будуть брати участь у опитуванні;
3. Складання запитань, на які експертам необхідно дати відповідь;
4. Проведення сету анонімних опитувань за допомогою надсилання учасникам опитувальних форм. Анкетування може проводитись через мережу Інтернет, електронну пошту та інші засоби віртуального зв'язку;
5. Проведення аналізу результатів;
6. Формування фінального звіту. Підведення підсумків щодо проведеного процесу оцінки ризиків складанням звіту, що містить відповіді експертів на запитання, аналіз результатів та рекомендації щодо подальшої роботи.

Як бачимо, жоден із етапів даного методу не передбачає дискусій та обговорень, а тому і виключає можливість необ'єктивного та непрозорого оцінювання, що спричинене впливом різних факторів. Обговорення, у даному випадку, замінені послідовними індивідуальними опитуваннями – анкетуванням. Із проведенням кожного наступного кола анкетування, експерти обґрунтовують та уточнюють свою точку зору, а також ознайомлюються із думками інших учасників опитування і порівнюють свою точку зору із узагальненими висновками групи учасників-експертів. Проведення опитувань проводиться доти, доки експерти не досягнуть консенсусу. Метод Дельфі має такі основні принципи:

- Анонімність. Експерти відповідають на анкетування анонімно, тому можуть вільно висловлювати свої прогнози та оцінки;
- Ітераційність. Проводиться декілька раундів анкетування, в кожному з яких учасники відповідають на питання, а потім отримують фідбек від інших учасників групи і мають змогу скоригувати свої прогнози щодо конкретного питання, і в наступному раунді надають їх знову;
- Статистична обробка результатів анкетування групи експертів;
- Структурований процес. Він забезпечує зосередження на основній темі опитування і проведення деталізованого аналізу ризиків аспекту.

Наведені принципи є також і вагомими перевагами даного методу прогнозування і оцінювання ризиків. Але, є і недоліки, а саме:

- Часові витрати. Процес анкетування може потребувати великих витрат часу, оскільки може виникати необхідність у проведенні додаткових турів анонімних опитувань, а також у вивченні та аналізі інформації та підведенні підсумків;
- Людський фактор. Іноді оцінки можуть бути необ'єктивними, оскільки деякі експерти є більш досвідченими та впевненими у своїй професійній думці, тому інші, менш впевнені в собі експерти, можуть відступати та відмовлятися від своєї точки зору під впливом даного фактору;
- Ймовірність неправильного підбору експертів. Можливість хибного визначення поняття експерта для конкретного предмету анкетування, і, як наслідок, отримання спотворених та недостовірних даних.

Наразі метод Дельфі використовується в різних сферах діяльності, але не є універсальним рішенням і має як позитивні, так і негативні аспекти у застосуванні в окремих проєктах [10].

Метод побудови матриці ймовірності – метод, що застосовується для оцінки ризиків в управлінні проєктами, бізнесом тощо. Він допомагає оцінювати ймовірність настання ризику та наслідки його впливу на проєкт. Для побудови матриці ймовірності потрібно виконати наступні кроки [21]:

1. Визначити ймовірні події та їх джерела;
2. Створити модель залежностей визначених ризиків;
3. Визначити ймовірності настання кожного ризику, попередньо скориставшись обраним методом оцінки, статистичними даними, досвідом із схожих минулих проєктів, експертними оцінками тощо;
4. Поділити події за рівнем ймовірності настання та рівнем впливу на проєкт;
5. Підготувати план із мінімізації настання та рівня наслідків визначених ймовірних подій.

Матриця ймовірності та впливу являє собою список групованих подій за певним рівнем оцінок, що були визначені та вираховані раніше, використовуючи підходящий для конкретного проєкту метод. Імовірності та рівні наслідків окремих ризиків формують категорії за діапазонами значень. Елементами матриці виступають значення добутків оцінок ймовірності настання подій та вагомості їх впливу на проєкт. Дані елементи складають матрицю, що поділена на сегменти в залежності від отриманих результатів. Приклад матриці ймовірності та впливу наведений у посібнику РМВОК [45], та відображений у таблиці, що наведена нижче.

Таблиця №4 – Матриця ймовірності та впливу

Імовірність	Загрози					Можливості				
0.90	0.05	0.09	0.18	0.36	0.72	0.72	0.36	0.18	0.09	0.05
0.70	0.04	0.07	0.14	0.28	0.56	0.56	0.28	0.14	0.07	0.04
0.50	0.03	0.05	0.10	0.20	0.40	0.40	0.20	0.10	0.05	0.03
0.30	0.02	0.03	0.06	0.12	0.24	0.24	0.12	0.06	0.03	0.02
0.10	0.01	0.01	0.02	0.04	0.08	0.08	0.04	0.02	0.01	0.01
	0.05	0.10	0.20	0.40	0.80	0.80	0.40	0.20	0.10	0.05

Із наведеної таблиці можемо спостерігати різні рівні ризиків, а саме відповідно до порогів: 0.05 – дуже низький, 0.10 – низький, 0.20 – середній, 0.40 – високий, 0.80 – дуже високий. Таким чином кожен оцінюється кожен ризик, і з’являється можливість вибору превентивних заходів для окремих ризиків, а також розробки планів управління та боротьби із ними.

## РОЗДІЛ 3

### ОЦІНКА РИЗИКІВ В ПРОЄКТІ ДЛЯ ЛЮДЕЙ З ОСОБЛИВИМИ ПОТРЕБАМИ

#### 3.1 Постановка задачі

Питання дитячої інвалідності – проблема, що потребує особливої уваги та постійного вдосконалення методів вирішення і боротьби. Дане питання є “гострим” не лише для громадян України, а й всесвітньо поширеним. Дітям із особливими потребами необхідне створення відповідних можливостей для їх розвитку та усунення будь-яких уражень для забезпечення вищого рівня життя.

Саме із такою метою із 2007 року працює Буковинський центр комплексної реабілітації для дітей із інвалідністю “Особлива дитина” у місті Чернівці. Центр займається комплексною реабілітацією, що направлена на забезпечення всебічного розвитку дітей із інвалідністю до 18 років, а також дітей віком до 3 років, які належать до групи ризику щодо отримання інвалідності, засвоєння ними знань, умінь, навичок пересування, комунікації, орієнтації в просторі, контролю за своєю поведінкою, самообслуговування, навчання з метою досягнення і збереження їхньої максимальної незалежності та реалізації особистого потенціалу.

У центрі визначено різні реабілітаційні напрями, що направлені на розвиток як комунікативних, так і фізичних навичок, відновлення та компенсацію вмінь тощо. Центр має 4 відділи реабілітації, а саме:

1. Інтерактив;
2. Ерготерапія;
3. Фізична терапія;
4. Медична реабілітація.



У даному реабілітаційному закладі кожен пацієнт має індивідуальну програму реабілітації, що складається комісією відповідних фахівців. Тому існує необхідність у програмному забезпеченні, що дало б можливість:

- Ефективно та легко вести та зберігати документацію;
- Відслідковувати прогрес;
- Вести процес букінгу/резервації тренажерів;
- Записуватися на прийом до лікаря (з інтегрованим календарем).

Із цією метою, студенти Чернівецького навчального університету почали розробку веб-додатку, що наразі містить інформацію про реабілітаційний центр “Особлива дитина”, дозволяє реєстрацію пацієнтів на реабілітацію, календарно відображає записи, а також запланованим є функціонал із побудови розкладу реабілітації пацієнтів та доповнення системи кабінетами лікарів та пацієнтів.

Моїм завданням було визначення та прогнозування ризиків для даного веб-додатку, їх оцінювання за допомогою використання методу Дельфі, визначення взаємозв’язків між ризиками (кореляційний аналіз) та міри їх сукупного впливу на реалізацію проекту (факторний аналіз) із використанням мови R та середовища R-Studio.

### 3.2 Організація та результати опитування експертів щодо оцінки ризиків в проєкті для людей з особливими потребами

Для визначення змісту ризиків нами був проведений попередній відбір та структурування ризиків за їх видами, які відображали загальний процес розробки проєкту на кожному із його етапів. В опитуванні взяли участь 18 спеціалістів компанії “Elogic Commerce”, які мають досвід роботи більше п’яти років, є провідними розробниками і очільниками відповідних технічних напрямів в більшості проєктів, які реалізує компанія. До основного кола мов програмування та технологій, якими володіють експерти належать наступні: Php, Laravel, Symphony, ZendFramework, Magento 1, Magento 2, MySql,

JavaScript, jQuery, Require.js, React, Vue.js, Alpine.js, Vue Storefront, CSS, LESS, SASS, Gulp, Grunt, Tailwind CSS, Hyva, Docker, Varnish, Git, Composer, Varnish.

Опитування проводилось із застосуванням технології Google-Form (<https://forms.gle/8LHZgFhC65bForeZ8>), що дозволило забезпечити максимально комфортні та прийнятні умови для проведення анонімного опитування відповідно до принципів методу Дельфі, зважаючи на високу зайнятість експертів. Експертами на попередньому етапі було відібрано п'ять типів ризиків, які відображали зміст певного етапу реалізації проєкту і оцінювались за п'яти-бальною шкалою, яка в подальшому була переведена в оцінки ймовірності їх настання (див. таблицю 5).

Таблиця №5 - Зміст ризиків, їх розподіл та оцінка за ймовірністю виникнення

№	Ризики	К-ть експертних голосів за відповідну ймовірність настання ризику				
		Дуже висока	Висока	Середня	Низька	Дуже низька
1	<b>Ризик конфіденційності даних:</b> збір та збереження медичних даних пацієнтів може викликати проблеми з конфіденційністю та безпекою даних.	3	4	7	4	0
2	<b>Ризик залежності від технічного персоналу:</b> веб-додаток може бути складним у використанні, що може вимагати підтримки технічного персоналу.	2	2	7	7	0
3	<b>Ризик низької прийнятності користувачами:</b> веб-додаток може бути складним у використанні та не задовольняти потреби користувачів.	3	3	8	4	0
4	<b>Проблеми з підтримкою та розширенням:</b> веб-додаток потребує постійної підтримки та оновлення, щоб забезпечити його безперебійну роботу та відповідність змінюваним потребам користувачів. Недостатня підтримка та оновлення можуть призвести до проблем з роботою системи та втратою користувачів.	2	7	7	2	0
5	<b>Проблеми з тестуванням та контролем якості:</b> веб-додаток повинен проходити тестування та контроль якості, щоб забезпечити його відповідність вимогам користувачів та відповідним стандартам якості.	4	4	4	5	1

Згідно таблиці 4 кожному із ризиків було надане цифрове значення ймовірності його настання, а саме: 0.05 – дуже низький, 0.10 – низький, 0.20 – середній, 0.40 – високий, 0.80 – дуже високий. Для проведення процедури ранжування, загальна кількість оцінок по кожному із видів ризиків була помножена на коефіцієнт ймовірності його настання, що дозволило отримати загальну підсумкову оцінку настання кожного із ризиків із подальшим ранжуванням: 1 – найвищий ризик та 5 – найменший ризик. Формула розрахунків була наступною:  $P = x_1 * 0.8 + x_2 * 0.4 + x_3 * 0.2 + x_4 * 0.1 + x_5 * 0.05$  (1).

Попередня процедура впорядкування ризиків за рангами згідно оцінок ймовірності їх настання свідчить про наступне (див. табл. 6).

Таблиця №6 - Результати процедури ранжування ризиків проекту.

№	Ризики	Значення	Ранг
1	<b>Ризик конфіденційності даних:</b> збір та збереження медичних даних пацієнтів може викликати проблеми з конфіденційністю та безпекою даних.	5,8	3
2	<b>Ризик залежності від технічного персоналу:</b> веб-додаток може бути складним у використанні, що може вимагати підтримки технічного персоналу.	4,5	5
3	<b>Ризик низької прийнятності користувачами:</b> веб-додаток може бути складним у використанні та не задовольняти потреби користувачів.	5,6	4
4	<b>Проблеми з підтримкою та розширенням:</b> веб-додаток потребує постійної підтримки та оновлення, щоб забезпечити його безперебійну роботу та відповідність змінюючимся потребам користувачів. Недостатня підтримка та оновлення можуть призвести до проблем з роботою системи та втратою користувачів.	6	2
5	<b>Проблеми з тестуванням та контролем якості:</b> веб-додаток повинен проходити тестування та контроль якості, щоб забезпечити його відповідність вимогам користувачів та відповідним стандартам якості.	6,15	1

Отже, на думку експертів найбільшим ризиком в проекті є проблема з тестуванням та контролем якості, на другому місці за значущістю є ризик проблеми із підтримкою та розширенням, на третьому – ризик втрати конфіденційності даних. Найменшими ризиками в цьому проекті є ризик залежності від технічного персоналу та ризик низької прийнятності користувачами. Водночас слід відзначити, що ранжування дозволило нам створити початкове уявлення про можливі ризики і їх значення, але в цілому

бальні оцінки та ранги є високими і ймовірно статистично значуще одне від одної не відрізняються, що вимагає застосування підходу оцінки ймовірностей.

### 3.3. Виявлення ймовірності настання ризику методом Дельфі

Для отримання більш точної інформації щодо настання відповідних ризиків на проєкті було розраховано деякі статистичні оцінки досліджуваних показників за їх ймовірнісними значеннями (див. табл. 7).

Таблиця №7 – Результати опитування за описовими статистиками

Назва ризику	Медіана	Проста оцінка	Стандартне відхилення	Середньозважена оцінка	Довірчий інтервал
<b>Ризик конфіденційності даних</b>	0,40	0,45	0,16	0,40	{0,32;0,53}
<b>Ризик залежності від технічного персоналу</b>	0,40	0,39	0,16	0,34	{0,27;0,46}
<b>Ризик низької прийнятності користувачами</b>	0,40	0,44	0,16	0,39	{0,31;0,52}
<b>Проблеми з підтримкою та розширенням</b>	0,48	0,48	0,14	0,44	{0,37;0,54}
<b>Проблеми з тестуванням та контролем якості</b>	0,40	0,44	0,20	0,32	{0,23;0,54}

Отже, як бачимо для усіх ризиків за ймовірністю їх настання за числовими значеннями медіана, проста та гармонічна середня є майже однаковими, а верхня межа довірчого інтервалу для середньої оцінки не перевищує 0,54. Межі довірчих інтервалів за рахунок незначного стандартного відхилення на рівні значущості  $p=0,05$  є вузькими, що може свідчити про достатню точність отриманої середньої оцінки за вибіркою.

Отже, на основі отриманих ймовірностей, визначених експертами, будемо матрицю ймовірності та впливу у таблиці 8, де розташуємо кожен ризик. Введемо пороги для визначення рангу ризику, що є загальноживаними

у багатьох компаніях: – дуже низький ризик – 0,05; – низький ризик – 0,10; – середній ризик – 0,20; – високий ризик – 0,40; – дуже високий ризик – 0,80.

Отже, як ми визначили за процедурою ранжування, міра впливу ризиків на проєкт є різною, а от процедури реагування на них залежать від ймовірності їх настання за результатами узгодженої оцінки експертами (див. табл. 8).

Таблиця №8 - Ризики та можливі рекомендації щодо реагування на них

Назва ризику	Можливе реагування за величиною ризику
<b>Ризик конфіденційності даних</b>	Ризик є високим – 0,40. Необхідно запланувати додаткові заходи захисту даних користувачів з урахуванням останніх досягнень та технологій.
<b>Ризик залежності від технічного персоналу</b>	Ризик є середнім – 0,34. Доцільним є пошук і найм спеціаліста здатного працювати і підтримувати додаток в робочу стані.
<b>Ризик низької прийнятності користувачами</b>	Ризик є середнім – 0,39. Під час планування дизайну додатку необхідно провести додаткові заходи узгодження його вигляду і функціональності із потенційними користувачами.
<b>Проблеми з підтримкою та розширенням</b>	Ризик є високим – 0,44. Вирішення ризику залежить від монетизації проєкту, попит на додаток в користувачів. Можливе рішення в оптимізації маркетингових процедур, додавання нових функціональних можливостей.
<b>Проблеми з тестуванням та контролем якості</b>	Ризик є середнім – 0,32. Звернути увагу на процедури тестування, кваліфікацію тестерів, залучити до тестування бета версій широкого кола користувачів.

### 3.4 Аналіз ризиків, їх взаємозв'язок та фактори впливу на проєкт

З метою визначення наявності взаємозалежності між оцінками експертів щодо ризиків в проєкті, було проведено кореляційний аналіз за розподілом експертних оцінок за ймовірністю настання ризиків. Для цього було сформовано п'ять змінних і завантажено їх у середовище R-Studio. Формуємо вихідну таблицю із даними та дивимось описові статистики розподілу оцінок експертів.

```
risk1 <- c(3,4,7,4,0)
risk2 <- c(2,2,7,7,0)
risk3 <- c(3,3,8,4,0)
risk4 <- c(2,7,7,2,0)
```

```
risk5 <- c(4,4,4,5,1)
data <- data.frame(risk1, risk2, risk3, risk4, risk5)
summary(data)
```

```
##      risk1      risk2      risk3      risk4      risk5
## Min.   :0.0   Min.   :0.0   Min.   :0.0   Min.   :0.0   Min.   :1.0
## 1st Qu.:3.0   1st Qu.:2.0   1st Qu.:3.0   1st Qu.:2.0   1st Qu.:4.0
## Median :4.0   Median :2.0   Median :3.0   Median :2.0   Median :4.0
## Mean   :3.6   Mean   :3.6   Mean   :3.6   Mean   :3.6   Mean   :3.6
## 3rd Qu.:4.0   3rd Qu.:7.0   3rd Qu.:4.0   3rd Qu.:7.0   3rd Qu.:4.0
## Max.   :7.0   Max.   :7.0   Max.   :8.0   Max.   :7.0   Max.   :5.0
```

Побудуємо матрицю кореляцій між досліджуваними показниками. Зважаючи на те, що досліджувані показники виражені у порядковій шкалі, ми не будемо визначати, зважаючи на невелику кількість даних, нормальність їх розподілу і застосуємо непараметричний метод кореляції Спірмена.

```
library(psych)
corr.test(data, use="complete", method = "spearman")

## Call:corr.test(x = data, use = "complete", method = "spearman")
##
##              Correlation
##              risk1  risk2  risk3  risk4  risk5
## risk1          1.00   0.87   0.92   0.87   0.57
## risk2          0.87   1.00   0.97   0.56   0.82
## risk3          0.92   0.97   1.00   0.65   0.69
## risk4          0.87   0.56   0.65   1.00   0.35
## risk5          0.57   0.82   0.69   0.35   1.00
##
##              Sample
##              [1] 5
## Probability values (Entries above the diagonal are adjusted for multiple t
ests.)
##              risk1  risk2  risk3  risk4  risk5
## risk1          0.00   0.47   0.24   0.47   0.99
## risk2          0.06   0.00   0.05   0.99   0.51
## risk3          0.03   0.01   0.00   0.99   0.99
## risk4          0.06   0.33   0.24   0.00   0.99
## risk5          0.31   0.09   0.20   0.56   0.00
##
```

Таким чином, проведений кореляційний аналіз свідчить, що між оцінками експертів ймовірності настання досліджуваних ризиків є сильні і статистично значущі корелятивні зв'язки. Для прикладу розглянемо один із них. Статистично значущим на рівні  $p < 0.01$  є коефіцієнт кореляції, який становить 0,97 (сильний корелятивний зв'язок) між ризиками: залежності від технічного персоналу та ризиком низької прийнятності користувачами. Такий зв'язок може пояснюватись тим, що необхідність доопрацювання функціональної та

візуальної частини додатку тісно пов'язаним із наявним кваліфікованим персоналом, і в разі відсутності якого таке удосконалення стає неможливим. Висновком наявних статистично значущих корелятивних зв'язків є те, що такі залежності необхідно враховувати під час планування дій регульовального характеру, адже впроваджуючи певний механізм для подолання одного ризику ми можемо активно впливати і на подолання ризику, пов'язаного із ним. Наступним завданням даної роботи є визначення впливу обраних ризиків на проєкт в цілому і можливості їх групування в меншу кількість змінних шляхом зменшення вихідного простору вхідних даних. Для цього застосуємо аналіз головних компонент PCA із варімакс обертанням.

```
fa.parallel(data, show.legend = FALSE, main = "Діаграма паралельного аналізу")
```

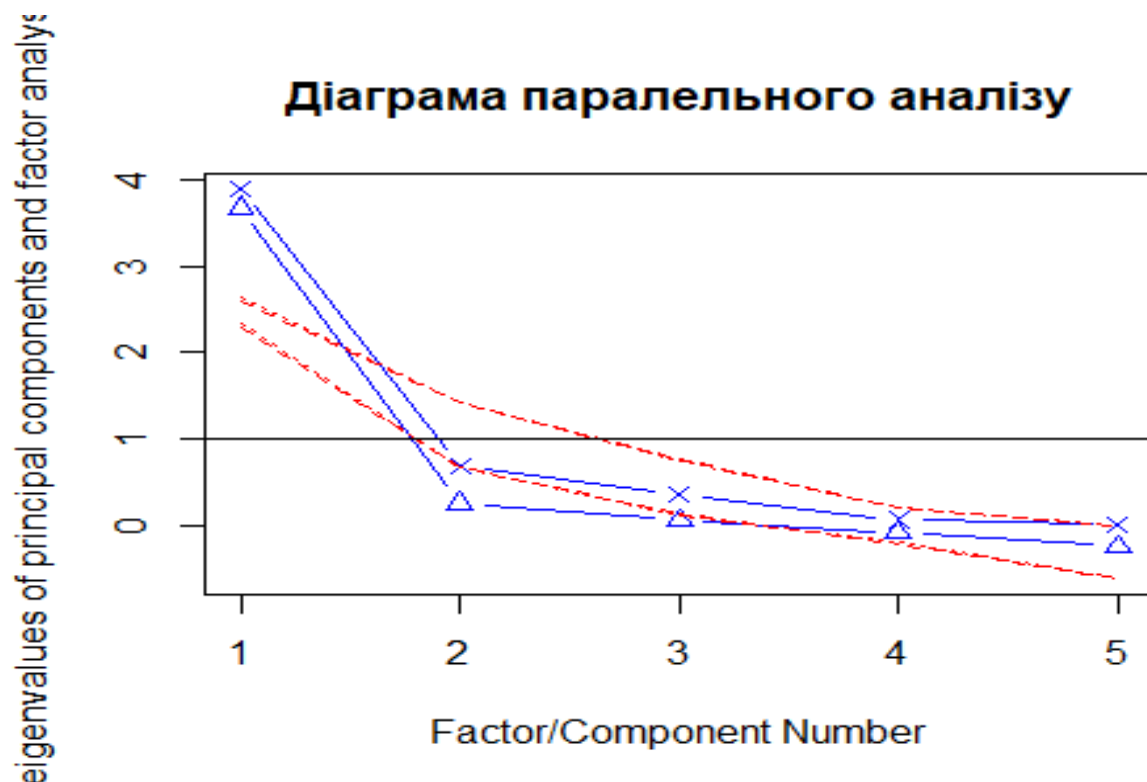


Рисунок 3.1 – Діаграма паралельного аналізу

Як бачимо в результаті проведеного паралельного аналізу із досліджуваних показників можливо виокремити два фактори. З цією метою проведемо факторний аналіз головних компонент.

```
pc <- principal(data, nfactors = 2, rotate = "vaimax")
```

```

## Principal Components Analysis
## Call: principal(r = data, nfactors = 2, rotate = "vaimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##      PC1  PC2  h2    u2 com
## risk1 0.99  0.12 0.99 0.0074 1.0
## risk2 0.87 -0.41 0.92 0.0768 1.4
## risk3 0.96  0.05 0.92 0.0843 1.0
## risk4 0.76  0.62 0.97 0.0328 1.9
## risk5 0.81 -0.34 0.77 0.2293 1.3
##
##
##          PC1  PC2
## SS loadings      3.88 0.69
## Proportion Var    0.78 0.14
## Cumulative Var    0.78 0.91
## Proportion Explained 0.85 0.15
## Cumulative Proportion 0.85 1.00
##
## Mean item complexity = 1.3
## Test of the hypothesis that 2 components are sufficient.
##
## The root mean square of the residuals (RMSR) is 0.06
## with the empirical chi square 0.36 with prob < 0.55
##
## Fit based upon off diagonal values = 0.99
fa.diagram(pc, simple = FALSE)

```

### Components Analysis

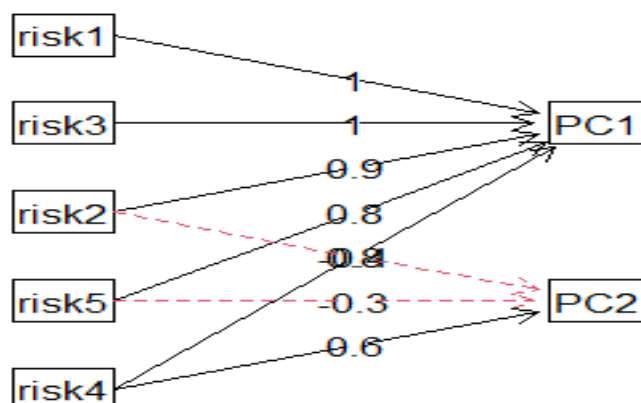


Рисунок 3.2 – Факторний аналіз компонент

Отримані результати факторного аналізу свідчать, що в сукупності обидва фактори пояснюють 91% накопиченої дисперсії. На перший фактор припадає 78%, а на другий - 14%. До складу першого і головного фактору увійшли усі досліджувані ризики, що дозволяє констатувати їх взаємодію і



значний спільний вплив на реалізацію проєкту. Другий фактор має значно менші факторні навантаження і в цілому може бути в подальшому не врахований під час розробки стратегії подолання ризиків.

Підводячи підсумок необхідно відзначити, що наявний сукупний вплив усіх ризиків та наявні корелятивні зв'язки між ними обумовлюють необхідність їх врахування під час планування і передбачення дієвих заходів превентивного реагування на наявні ризики, які дозволять зменшити вірогідність їх виникнення.

## ВИСНОВКИ

Метою даного дослідження було встановлення можливих ефективних шляхів аналізу ризиків при плануванні ІТ-проєкту - веб-додатку для людей с особливими потребами. Згідно поставлених завдань можна констатувати, що теорію управління проєктами доцільно пояснювати через призму системного підходу, основним інструментом якого виступає системний аналіз. Управління ризиками є важливою складовою системи управління проєктами, і дозволяє застосовувати апробовані та перевірені численними практиками методи та технології управління ними: контролю, оцінки, розробки заходів реагування. Саме вони дозволяють здійснювати превентивні впливи з метою запобігання їх виникнення, нівелювати деструктивні впливи на реалізацію проєкту на кожному із його етапів.

Було встановлено, що оцінка ризиків є актуальною проблемою і є предметом численних наукових досліджень. Розглянуті методи оцінки ризиків дозволяють проводити комплексне вивчення їх змісту, розробляти на основі такої оцінки рекомендації по їх подоланню, активно впливати на хід реалізації проєкту, розширювати канал комунікації із зацікавленими сторонами, замовниками та користувачами. Проведена оцінка ризиків проєкту шляхом використання методів Дельфі, ранжування, побудови ймовірнісної матриці, корелятивного та факторного аналізу свідчить, що їх використання дозволяє краще зрозуміти сутність самих ризиків, величину їх впливу на проєкт, взаємодію. Взаємодія між ризиками дозволяє застосовувати однорідні засоби та технології щодо їх нівелювання шляхом врахування особливостей зв'язків між ними. Активно впливаючи на один із ризиків, за умови правильно підібраних технологій реагування, активно впливаємо на пов'язаний із ним. Це в підсумку дозволить зекономити не тільки кошти на розробку і підтримку кінцевого продукту, але й забезпечити необхідний рівень якості продукту та задовольнити вимоги та потреби кінцевого споживача.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *13 Common Project Risks and How to Tackle Them* | Simplilearn. (2020a, Листопад 20). Simplilearn.Com. <https://www.simplilearn.com/common-project-risks-article>
2. *13 Common Project Risks and How to Tackle Them* | Simplilearn. (2020b, Листопад 20). Simplilearn.Com. <https://www.simplilearn.com/common-project-risks-article>
3. *16 Types of Software: What They Are and How To Use Them*. (б. д.). Indeed Career Guide. Вилучено 08, Квітень 2023, із <https://www.indeed.com/career-advice/career-development/types-of-software>
4. *Основні принципи Agile-маніфесту*. (б. д.). Вилучено 05, Квітень 2023, із <https://agilemanifesto.org/iso/uk/principles.html>
5. СИСТЕМНИЙ ПІДХІД У НАУЦІ ТА В ПЕДАГОГІЧНИХ ДОСЛІДЖЕННЯХ. Ч. 2. ОНТОЛОГІЯ СИСТЕМ. (б. д.). *Серія: «Педагогіка, психологія і соціологія»*. Вилучено 05, Квітень 2023, із <https://pedagogy.donntu.edu.ua/2-25-2018/m-p-kostiuchenko/>
6. *Agile vs Waterfall: Difference Between Two Powerful Methodologies*. (2021, Квітень 30). Hygger.Io Guides. <https://hygger.io/guides/agile/agile-vs-waterfall/>
7. *Agile-маніфест розробки програмного забезпечення*. (б. д.). Вилучено 05, Квітень 2023, із <https://agilemanifesto.org/iso/uk/manifesto.html>
8. Bell, M. (2022, Грудень 16). Project Management Artifacts: Definition, Types, and Phases. *Project Management Academy Resources*. <https://projectmanagementacademy.net/resources/blog/types-of-project-management-artifacts/>
9. *Defining a project management system—Systems Approach*. (б. д.). Вилучено 05, Квітень 2023, із <https://www.pmi.org/learning/library/defining-project-management-systems-approach-1752>
10. Delphi method. (2023). В *Wikipedia*. [https://en.wikipedia.org/w/index.php?title=Delphi\\_method&oldid=1144074594](https://en.wikipedia.org/w/index.php?title=Delphi_method&oldid=1144074594)

11. Donato, H. (2022, ЖОВТЕНЬ 17). *5 Phases of Project Management Life Cycle You Need to Know*. Project-Management.Com. <https://project-management.com/project-management-phases/>
12. Feldman, K. (2018, Листопад 7). *Process Management*. *Isixsigma.Com*. <https://www.isixsigma.com/dictionary/process-management/>
13. monday.com, A. of us at. (2021, Червень 11). *Project management for software development: A complete guide*. *Monday.Com Blog*. <https://monday.com/blog/project-management/project-management-for-software-development/>
14. Press, G. (б. д.). *Modern Computing: A Short History, 1945-2022*. Forbes. Вилучено 08, Квітень 2023, із <https://www.forbes.com/sites/gilpress/2022/04/26/modern-computing-a-short-history-1945-2022/>
15. Process. (2023). В *Wikipedia*. <https://en.wikipedia.org/w/index.php?title=Process&oldid=1144445585>
16. *Project Management: A Systematic Approach From Planning To Implementation*. (б. д.). Вилучено 05, Квітень 2023, із <https://www.consultantsreview.com/cxoinsights/project-management-a-systematic-approach-from-planning-to-implementation-vid-617.html>
17. *Project Management Evolution Research Directions*. (б. д.). Вилучено 05, Квітень 2023, із <https://www.pmi.org/learning/library/project-management-evolution-research-directions-8348>
18. *Project Management in Software Development | Trio Developers*. (б. д.). Вилучено 08, Квітень 2023, із <https://www.trio.dev/blog/project-management-software-development>
19. *Project Management Job Growth and Talent Gap Report | PMI*. (б. д.). Вилучено 05, Квітень 2023, із <https://www.pmi.org/learning/careers/job-growth>
20. Prokopets, M. (2019, Вересень 10). *Kanban vs Scrum: 6 Golden Rules to Help You Pick the Right One*. Nira. <https://nira.com/kanban-vs-scrum/>

21. Risk matrix. (2023). В *Wikipedia*.  
[https://en.wikipedia.org/w/index.php?title=Risk\\_matrix&oldid=1151338503](https://en.wikipedia.org/w/index.php?title=Risk_matrix&oldid=1151338503)
22. Sacolick, I. (2022, Квітень 8). *A brief history of the agile methodology*. InfoWorld. <https://www.infoworld.com/article/3655646/a-brief-history-of-the-agile-methodology.html>
23. *Software | Definition, Types, & Facts | Britannica*. (б. д.). Вилучено 08, Квітень 2023, із <https://www.britannica.com/technology/software>
24. System. (2023). В *Wikipedia*.  
<https://en.wikipedia.org/w/index.php?title=System&oldid=1147813535>
25. Systems analysis. (2023). В *Wikipedia*.  
[https://en.wikipedia.org/w/index.php?title=Systems\\_analysis&oldid=1138378625](https://en.wikipedia.org/w/index.php?title=Systems_analysis&oldid=1138378625)
26. Technologies, \_ Zindagi. (2022, Червень 13). *8 Types Of IT Risks & Consequences | Zindagi Technologies*. <https://zindagitech.com/8-types-of-it-risks-and-there-consequences/>
27. *The Stages of the Agile Software Development Life Cycle*. (2017, Грудень 1).  
<https://www.lucidchart.com/blog/agile-software-development-life-cycle>
28. *The Ultimate Guide To SWOT Analysis For Business And Why It Matters*. (2020, Вересень 22). Simplilearn.Com. <https://www.simplilearn.com/swot-analysis-for-business-article>
29. *Understanding Lean Agile and the 5 Lean Principles*. (б. д.). Easy Agile. Вилучено 09, Квітень 2023, із <https://easyagile.com/blog/lean-agile/>
30. *What are Project Artifacts? Definition, Meaning, and Example*. |. (2022, Серпень 17). <https://pmstudycircle.com/project-artifacts/>
31. *What Is a Project? Definition, Key Characteristics Lifecycle*. (б. д.). Вилучено 05, Квітень 2023, із <https://mymanagementguide.com/basics/what-is-a-project/>
32. *What is a project management system?* (б. д.). Вилучено 05, Квітень 2023, із <https://www.inloox.com/project-management-glossary/project-management-system/>

33. *What is Application Software? (With Examples) | Simplilearn.* (2022, Жовтень 21). Simplilearn.Com. <https://www.simplilearn.com/tutorials/programming-tutorial/what-is-application-software>
34. *What Is Extreme Programming (XP)? - Values, Principles, And Practices.* (2022, Грудень 22). <https://www.nimblework.com/agile/extreme-programming-xp/>
35. *What Is Kanban? An Overview Of The Kanban Method.* (2022, Листопад 23). <https://www.nimblework.com/kanban/what-is-kanban/>
36. *What is Management? Objectives, Functions, and Characteristics.* (б. д.). Вилучено 05, Квітень 2023, із <https://www.knowledgehut.com/blog/others/what-is-management>
37. *What is PESTLE Analysis? An Important Business Analysis Tool.* (б. д.). Вилучено 12, Квітень 2023, із <https://pestleanalysis.com/what-is-pestle-analysis/>
38. *What Is Project Management Life Cycle? And Its Phases.* (б. д.). Вилучено 05, Квітень 2023, із <https://www.nimblework.com/project-management/project-management-life-cycle/>
39. *What is Risk in Project Management?* (б. д.). Вилучено 11, Квітень 2023, із <https://www.wrike.com/project-management-guide/faq/what-is-risk-in-project-management/>
40. *What is Risk Management and Why is it Important?* (б. д.). Security. Вилучено 12, Квітень 2023, із <https://www.techtarget.com/searchsecurity/definition/What-is-risk-management-and-why-is-it-important>
41. *What is Scrum?* (б. д.). Software Quality. Вилучено 09, Квітень 2023, із <https://www.techtarget.com/searchsoftwarequality/definition/Scrum>
42. *What is software development and why is it needed?* (б. д.). Indeed Career Guide. Вилучено 08, Квітень 2023, із <https://uk.indeed.com/career-advice/career-development/software-development>
43. *What is Waterfall Methodology? – Forbes Advisor.* (б. д.). Вилучено 09, Квітень 2023, із <https://www.forbes.com/advisor/business/what-is-waterfall-methodology/>

44. Zulqadar, A. (2018, СЕРПЕНЬ 10). SDLC Waterfall Model: The 6 phases you need to know about. *Rezaid*. <https://rezaid.co.uk/sdlc-waterfall-model/>
45. Project Management Institute, *A Guide to the Project Management Body of Knowledge*, Fifth Edition, 2013, p. 331-332