

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ

до виконання лабораторних робіт

з дисципліни «Гнучкі методології управління ІТ-стартапами та ІТ-проєктами»

для здобувачів освітнього ступеня «магістр» спеціальності

Ф3 – Комп'ютерні науки (освітня програма «Управління стартапами

і проєктами в галузі інформаційних технологій»)

усіх форм навчання

Черкаси
2025

УДК 004:005.8](07)
М54

*Затверджено вченою радою ФІТІС
протокол № 4 від 16.10.2025 р.
згідно з рішенням кафедри комп'ютерних
наук та системного аналізу,
протокол № 4 від 01.10.2025 р.*

Упорядник: Колеснікова К. В., *докт. техн. наук, проф.*,

Рецензент: Прокопенко Т. О., *докт. техн. наук, проф.*

М54 Методичні рекомендації до виконання лабораторних робіт з дисципліни «Гнучкі методології управління ІТ-стартапами та ІТ-проєктами» для здобувачів освітнього ступеня «магістр» спеціальностей ФЗ – Комп'ютерні науки (освітня програма «Управління стартапами і проєктами в галузі інформаційних технологій») усіх форм навчання [Електронний ресурс] / [упоряд. Колеснікова К.В.]; М-во освіти і науки України, Черкас. держ. технол. ун-т. Черкаси: ЧДТУ, 2025. 58 с.

Методичні рекомендації до виконання лабораторних робіт освітнього ступеня «магістр» усіх спеціальностей спрямовані на формування у здобувачів стратегії до самостійної роботи з формування комплексного уявлення про сучасні гнучкі управлінські практики, їх особливості та умови застосування.

УДК 004:005.8](07)

Виробничо-практичне
електронне видання
комбінованого використання

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ
до виконання лабораторних робіт
з дисципліни «Гнучкі методології управління ІТ-стартапами та ІТ-проєктами»
для здобувачів освітнього ступеня «магістр»
усіх спеціальностей
усіх форм навчання

Упорядник: **Колеснікова Катерина Вікторівна**
В авторській редакції.

ВСТУП

Предметом вивчення навчальної дисципліни «Гнучкі методології управління ІТ-стартапами та ІТ-проєктами» є вивчення концепцій, принципів та інструментів гнучких методологій управління стартапами та проєктами та їх застосування у високотехнологічних сферах, з метою швидкого масштабування та виходу ІТ-проєктів на ринок.

Метою вивчення навчальної дисципліни «Гнучкі методології управління ІТ-стартапами та ІТ-проєктами» є: формування комплексного уявлення про сучасні гнучкі управлінські практики, їх особливості, передумови і умови застосування, відмінності від класичних підходів в управлінні проєктами та продуктами. Освоїти практичні навички ефективної роботи в командах, які застосовують гнучкі методи роботи.

Основними завданнями вивчення дисципліни «Гнучкі методології управління ІТ-стартапами та ІТ-проєктами» є:

- сформувані у здобувачів вищої освіти систематизовані знання з ключових принципів Agile, Scrum, Kanban, Lean та інших підходів;
- ознайомити здобувачів вищої освіти з відмінностями між традиційними та гнучкими методами управління проєктами;
- навчити здобувачів вищої освіти застосовувати ефективні стратегії управління стартапами та проєктами на основі гнучких методологій;
- ознайомити здобувачів вищої освіти з інструментами виявлення бар'єрів впровадження гнучких методів та шляхів їх подолання.

Дані лабораторні роботи спрямовані на:

- ознайомлення з класичними та гнучкими методологіями управління проєктами;
- формулювання проблеми, місії, бачення, цінності й SMART-цілей ІТ-проєктів, розробку user stories та обґрунтування вибору методології;
- визначення оточення й учасників проєкту, роботу з реєстром та матрицею зацікавлених сторін, формування стратегії взаємодії;
- планування та ідентифікацію ризиків ІТ-проєкту, розробку плану управління ризиками;
- застосування практик Scrum і Kanban: формування backlog та проведення спринтів, створення burn-down chart і cumulative flow diagram, роботу з Kanban-дошкою та WIP-лімітами, обчислювання cycle time, lead time і velocity, вміння аналізувати інші Agile-метрики, проводити ретроспективу та формувати план покращень проєкту.

Метою лабораторних робіт є формування у студентів практичних компетенцій з використання гнучких методологій та інструментів управління ІТ-проєктами для ефективного планування, реалізації й оцінювання результатів командної роботи.

ЛАБОРАТОРНА РОБОТА №1

Аналіз методологій управління проектами

Мета роботи: знайомство з гнучкими методологіями управління проектами.

Порядок виконання роботи

1. Вивчити теоретичні відомості.
2. Обрати варіант завдання.
3. Виконати завдання до лабораторної роботи згідно свого варіанту.
4. Скласти та оформити звіт.

Теоретичні відомості

Управління проектами, як самостійна галузь професійної діяльності, має свою методологію і ґрунтується на певних стандартах. В основі таких методів лежать методики мережного планування, які з'явилися США в кінці 1950-х рр. При цьому методики управління проектами поширилися не тільки в країнах з ринковою економікою, а й в країнах з так званою «плановою» економікою. Вони почали використовуватися в будівництві, що послужило основою їх поширення в інших галузях і появи методів проектного управління.

Прикладом може служити об'єднання зусиль фірми «Дюпон» та фірми «Ремінгтон Ренд» для створення графіка комплексних робіт з модернізації заводів «Дюпон» за допомогою обчислювальної машини Univac в 1956р. Результатом стало створення раціонального і простого методу опису проекту на ЕОМ – методу критичного шляху (CPM – Critical Path Method).

Практично паралельно і незалежно було створено метод аналізу і оцінки PERT (Program Evaluation and Review Technique) у військово-морських силах США. Даний метод розроблявся корпорацією «Локхід» та консалтинговою фірмою «Буз, Аллен енд Гамільтон» для ракетної системи «Поларіс». Проект включав 3800 основних підрядників і складався з 60 000 операцій. Завдяки використанню методики PERT проект став успішним і завершився на два роки раніше встановленого терміну. Після наочного успіху метод PERT став використовуватися в збройних силах США для планування проектів.

Система управління проектами постійно розвивається. Останнім часом було створено декілька уніфікованих методологій, інструментів, механізмів та стандартів, які можуть бути використані також для проектів в ІТ-сфері.

З найбільш поширених можна відзначити процесну модель, яка докладно розглянута в Project Management Body of Knowledge (PMBOK) Американського інституту управління проектами (PMI). Цей документ де-факто є міжнародним стандартом з управління проектами .

Дані Міжнародної асоціації управління проектами (IPMA) говорять про те, що використання сучасних методологій управління проектами економить близько 20-30% часу і близько 15-20% коштів на здійснення проектів.

Все методології (ще їх називають моделями, методиками) розробки програмного забезпечення класифікують за «вагою», тобто за кількістю формалізованих процесів та детальністю їх регламентації. Отже, чим більше

процесів документовано, чим більш детально описана методологія, тим більше буде її «вага».

1. Важковагові методології (деякі з них):

- Capability Maturity Model for Software (SW-CMM) визначає п'ять рівнів «зрілості проекту»;

- Rational Unified Process (RUP) – ітеративна модель розробки;

- Microsoft Solutions Framework (MSF) – база знань компанії Microsoft по розробки програм;

- Personal Software Process – модель визначає вимоги до компетенцій розробника. Різновид методології Team Software Process – модель орієнтує на самоврядні команди від 3 до 20 розробників і ін.

2. Легковагові або agile-методики:

- eXtreme Programming або XP – екстремальне програмування;

- Crystal Clear – сімейство методологій, визначають необхідний ступінь формалізації процесу розробки в залежності від кількості учасників і критичності завдань;

- Feature Driven Development (FDD) – функціонально-орієнтована розробка;

- OpenUP – ітеративно-інкрементальний метод розробки програм, позиціонується як легкий і гнучкий варіант великовагової методології RUP;

- Scrum – управління розробкою ІТ-проектів з високим ступенем невизначеності;

- Kanban – методологія «бережливого виробництва»; та ін.

Результати дослідження Agile Survey про популярність гнучких методологій представлені на рис. 1.

Легковагові методології (або Agile) з'явилися порівняно недавно. У 2001р 17 фахівців (консультантів і практиків) провели семінар, на якому сформулювали основні принципи гнучкої розробки ПО – Agile Manifesto (маніфест гнучкої розробки). Він перекладений багатьма мовами світу і доступний за посиланням <http://agilemanifesto.org/>.

Ідея всіх гнучких методологій полягає в тому, що процес, який застосовується під час розробки ПЗ повинен бути адаптивним, орієнтованим на людей і їх взаємодію (Agile Software Development). По суті це скоріше набір практик, ніж методологія.

А. Коуберн, один з авторів «Маніфесту», провів аналіз ІТ-проектів за останні 20 років, виконаних на підставі різних моделей управління: від полегшених, гнучких до важких. Даний аналіз показав відсутність кореляції між успіхом або провалом проектів і вибраними моделями розробки, застосовуваними в проектах.

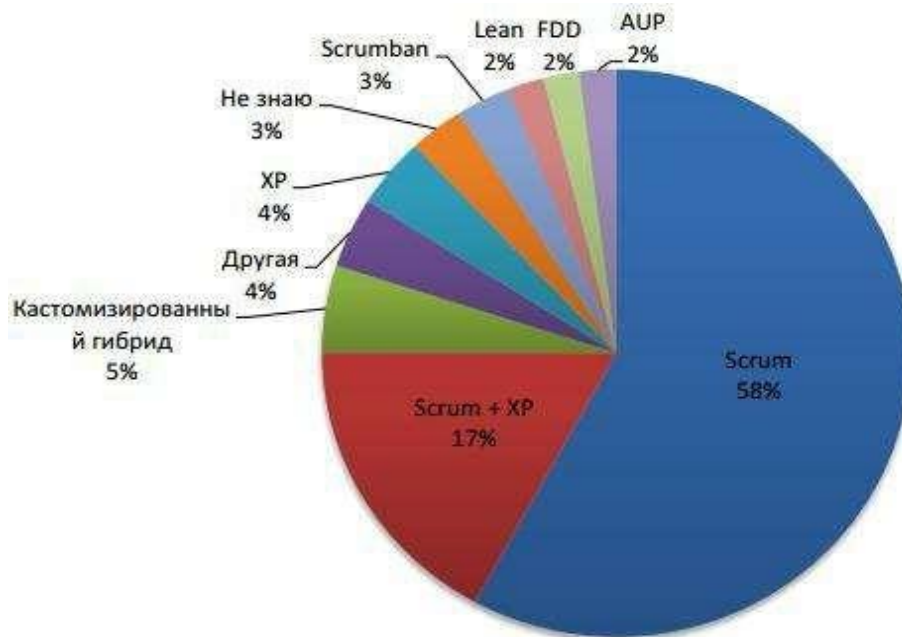


Рис. 1. Популярність гнучких методологій УП

А. Коуберн також зробив висновок про те, що: у кожного проекту повинна бути своя модель процесу розробки, а у кожній моделі – свій час використання.

Для розробників це означає, що не існує єдиного правильного процесу розробки. У кожному новому проекті процес повинен визначатися кожен раз заново по «закону чотирьох П» (рис. 2).

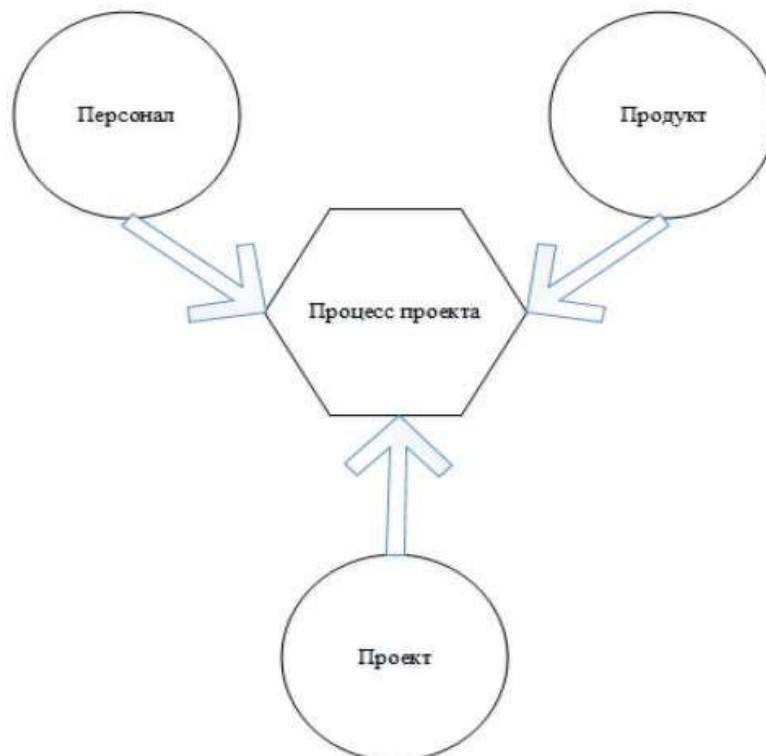


Рис. 2. Закон 4П

Під час вибору моделі управління проектом можна орієнтуватися на табл. 1:

Таблиця 1 – Вибір методології управління проектом

«Вага» моделі	Переваги	Недоліки
Важковагові	1. Процеси розраховано на середню кваліфікацію виконавців; 2. Великий перелік спеціалізацій виконавців; 3. Не високі вимоги до стабільності у проектній команді; 4. Відсутні обмеження за обсягом робіт та складністю проектів	1. Вимагають суттєвої управлінської надбудови; 2. Мають довгі стадії аналізу та проектування; 3. Мають високий ступень формалізації комунікацій
Легковагові (гнучкі)	1. Зниження витрат пов'язаних з управлінням проектом, ризиками, змінами та конфігурацією проекту;	1. Ефективність залежить від індивідуальних особливостей членів команди;
	2. Спрощення стадій аналізу та проектування; 3. Орієнтація на розробку функціоналу та сумісництво ролей; 4. Неформальна система комунікацій	2. Вимагають висококваліфікованої, універсальної та стабільно працюючої команди; 3. Масштаб та складність проектів обмежена

Завдання на лабораторну роботу

1. За допомогою пошуку в мережі Інтернет знайдіть інформацію про сучасні методології управління IT-проектами. Надайте підстави для їх класифікації. Для кожної випадку наведіть приклади методологій.

2. З отриманого списку «важких» методологій управління IT-проектами оберіть одну. Виконайте аналіз методології. Результати надайте у вигляді таблиці (табл. 2).

Таблиця 2 – Особливості методики

Характеристика	Опис
<i>Повна назва методології</i>	
<i>Автори</i>	
<i>Історія виникнення</i>	
<i>Країна виникнення</i>	
<i>Основні принципи, підходи</i>	

<i>Програмні засоби для реалізації, якщо є</i>	
<i>Використовується зараз</i>	

3. З отриманого списку гнучких (agile) методологій виберіть одну. Проведіть її аналіз. Результат надайте у вигляді таблиці (табл. 2).

4. Проведіть порівняльний аналіз методологій (завдання 3, 4). Результати подайте у вигляді презентації.

Зміст звіту

1. Титульний аркуш (додаток А).
2. Тема і мета роботи.
3. Короткі теоретичні відомості.
4. Таблиці з даними.
5. Висновки.

ЛАБОРАТОРНА РОБОТА №2

Розробка проекту створення IT-продукту

Мета роботи: навчитися демонструвати ідею свого проекту стейкхолдерам

Порядок виконання роботи

1. Вивчити теоретичні відомості.
2. Обрати варіант завдання.
3. Виконати завдання до лабораторної роботи згідно свого варіанту.
4. Скласти та оформити звіт.

Теоретичні відомості

Запускаючи нові проекти, ми часто задаємося питанням – З чого почати? Насправді все просто. Існує дієве правило: «Хочеш розібратися зі своїми думками – занотуй їх на папір». Так і з проектами – хочеш запустити новий проект, почни висловлювати свої ідеї про проект у вигляді літер, цифр та графіків, на папері чи комп'ютері – не важливо! Щоб зрозуміти послідовність записи своїх думок «на папері», потрібно розуміти технологію зародження проекту. Все починається з задуму. Задум може бути виражений або не виражений на папері, але його головна відмінність від ідеї в тому, що їм можна ділитися. Задум – це ідея, яка однаково розуміється усіма учасниками проекту. В задумі ми повинні чітко сформулювати проблематику, яка викликала потребу у проекті, і завдання які треба буде розв'язати. Перевірити грамотність формулювання задуму можна питанням «Що і для чого ми робимо?» і, якщо учасники однозначно на нього відповідають – у нас є задум!

На основі задуму формуються цілі проекту. Головна характеристика цілей – вони повинні бути вимірні. Неможливо досягти того, що не можна виміряти! Тому формулюючи кожну мету, потрібно питати себе «Чим будемо вимірювати?» і якщо формулювання мети дає відповідь на це питання, то є ЦІЛІ. Хоча, звичайно, вимірність мети – це не єдина її характеристика. В ідеалі потрібно мати SMART – цілі.

«SMART» (від англійського «розумний») – це аббревіатура слів Specific (конкретний; пояснюється, що саме необхідно досягти), Measurable (вимірний; пояснюється, в яких одиницях буде вимірюватися результат), Attainable (досяжний; пояснюється, за рахунок чого планується досягти мети), Relevant (що відноситься до справи; співвіднесення мети з іншими цілями проекту та визначення, чи дійсно досягнення даної мети дозволить досягти бажаного результату) і Timely (обмежений у часі). Якщо потрудитися і описати цілі в такому контексті, то всім учасникам буде однозначно зрозуміло що, коли і навіщо буде з'являтися в результаті проекту.

Наступним етапом буде визначення засобів реалізації, тобто однозначне розуміння того, чи потрібен нам взагалі цей проект або можна просто купити на стороні те, що нам потрібно.

Отже, якщо проект все таки потрібен, то визначаємо за допомогою чого і як будемо досягати описані цілі і реалізовувати проект.



Рис. 1. Логіка формування проекту: задум – мета – засоби

Необхідно зосередитися на розробці цілей, які є конкретними, значущими, досяжними, актуальними і своєчасними. Цілі, які мають ці характеристики, з більшою ймовірністю будуть реалізовані.

Кожну ціль потрібно розглядати з точки зору цих п'яти параметрів. Для зручності кожен мету записуємо на окремому стікері і потім формуємо єдиний документ.

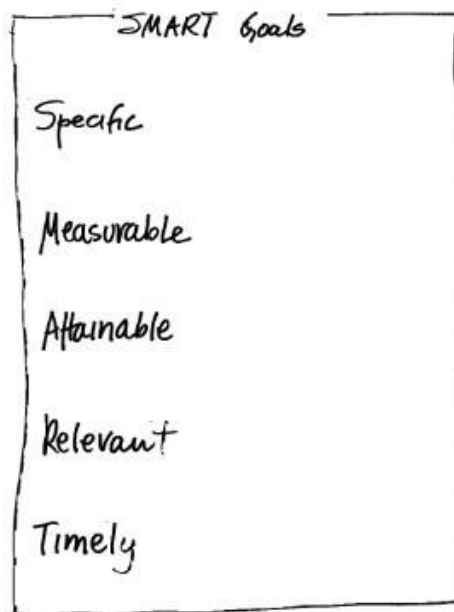


Рис.2. Модель SMART для постановки цілей

Визначити ролі користувачів продукту з їх опис. Написати як мінімум 10 історій Користувачів (user stories) для Вашого продукту.

Визначити ролі користувачів продукту з їх коротким описом. Можливі ролі: An account holder – людина, що має обліковий запис в системі DevHelpCenter. A guest – людина, що не має облікового запису в системі DevHelpCenter.

A developer – людина, яка може переглядати та шукати питання без облікового запису, а також голосувати за, проти, коментувати і ставити питання за допомогою облікового запису в системі DevHelpCenter.

An expert – людина, здатна відповісти на питання.

Формат написання історій користувачів:

«Як _____, я хочу _____, <з такою-то ціллю>».

Історії користувачів повинні відповідати принципам INVEST

Таблиця 1. Принципи INVEST для формування user stories

I	Independent. Це означає, що вимога може існувати поза іншим програмам користувача історій і при цьому залишатися значущим. Це дозволяє при необхідності вільно змінювати порядок вимоги і їх призначені для користувача історії.
N	Negotiable. Призначені для користувача історії також повинні бути досить загальними, щоб команда розробників і клієнт могли працювати над їх реалізацією. Вони не повинні зосереджуватися на конкретних технічних деталях. Замість цього слід зосередитися на найбільш важливих аспектах вимог, пам'ятаючи, що вони можуть змінитися.
V	Valuable. Призначені для користувача історії повинні приносити користь клієнтові.
E	Estimable. Повинна бути можливість оцінити, скільки часу буде потрібно на розробку і реалізацію вимоги в призначеній для користувача історії.
S	Small. Призначена для користувача історія повинна бути невеликою, тому що її потрібно розробити в короткі терміни. Якщо час на розробку і впровадження вимоги не визначено, призначена для користувача історія, ймовірно, занадто велика, тому її слід розбити на більш дрібні, керовані.
T	Testable. Призначені для користувача історії повинні піддаватися перевірці по набору критеріїв, щоб визначити, «виконано» Чи означає це, що означає, що призначена для користувача історія виконала те, що мала намір зробити, і не вимагає подальшої доробки. Зазвичай це досягається за допомогою приймальних випробувань..

Приклади історій користувачів:

– Як розробник, я хочу мати можливість переглядати питання, щоб знаходити питання, схожі на мої.

– Як розробник, я хочу мати можливість шукати питання за ключовими словами, щоб знайти питання, схоже на моє.

– Як експерт, я хочу мати можливість відповідати на питання, щоб допомогти розробникам з їх проблемами.

– Як власник облікового запису, я хочу мати можливість

завантажити зображення профілю, щоб персоналізувати свій обліковий запис.

– Як розробник, я хочу мати можливість проголосувати за відповіді, щоб показати, що відповідь була корисною.

– Як розробник, я хочу мати можливість впорядкувати відповіді на питання за рейтингом, щоб швидко знайти найбільш корисну відповідь.

– Як розробник, я хочу мати можливість зареєструватися в системі, щоб я міг ставити питання та давати відповіді на питання.

– Як розробник, я хочу мати можливість змінювати свій профіль, щоб це допомагало мені актуалізувати мій обліковий запис.

– Як розробник, я хочу мати можливість переглядати обліковий запис іншого розробника, щоб знати його відповіді.

Завдання на лабораторну роботу

1. Обрати тему проекту: студент повинен самостійно придумати ідею проекту або використати один із проектів, наведених у додатку В.

2. Описати проблему, яку вирішує ваш проект.

3. Сформулювати Мету, Місію, Візію, Цінності та Цілі проекту. Для формування цілей обов'язково використовувати методику SMART.

4. Обґрунтувати вибір Agile методології та інструментів для реалізації проекту (можна використовувати матеріали практичної роботи 1).

5. Визначити ролі користувачів продукту з їх описом. Написати щонайменше 15 історій користувачів (user stories) для вашого продукту.

6. Запропонувати план реалізації проекту відповідно до обраної Agile методології.

7. Підготувати презентацію проекту (1–2 слайди на кожне питання).

Зміст звіту

1. Титульний аркуш (додаток А).

2. Тема і мета роботи.

3. Короткі теоретичні відомості.

4. Результати виконання роботи (тема проекту, проблема, мета, місія та інше).

5. Висновки.

ЛАБОРАТОРНА РОБОТА №3

Оточення та учасники проекту. Реєстр зацікавлених сторін

Мета роботи: Набуття досвіду формування реєстру зацікавлених осіб та розробка стратегії роботи з користувачами проекту.

Порядок виконання роботи

1. Вивчити теоретичні відомості.
2. Виконати завдання до лабораторної роботи згідно проекту з лабораторної роботи № 2.
3. Скласти та оформити звіт.

Теоретичні відомості

Важливим елементом управління проектами є оточення проекту, оскільки важливо визначити середовище, в якому виникає, існує і завершується проєкт.

Оточення проєкту (Project Environment) — сукупність зовнішніх та внутрішніх сил, які сприяють чи заважають досягненню цілей проєкту.

Відносини між проєктом і середовищами не дозволяють провести чітку межу між ними. Дуже важливе значення має керівництво підприємства, оскільки саме воно визначає цілі та основні вимоги щодо проєкту, його реалізації, методів управління.

До факторів ближнього оточення проєкту перш за все відноситься керівництво підприємства, яке визначає цілі та основні вимоги до проєкту. Великий вплив на проєкт можуть надавати основні структурні підрозділи підприємства, інфраструктура підприємства, ставлення громадських організацій і колективу в цілому. Через взаємодію проєкту зі службами та відділами відбувається вплив близького оточення на проєкт. Враховуючи його «близькість», такий вплив є найрегулярнішим і найсуттєвішим.

До факторів далекого оточення належать ті сфери, галузі життєдіяльності суспільства, в яких діє організація. Насамперед це ринки, на яких працює організація. Крім того, великий вплив на проєкт мають такі фактори, як політика держави, стан її економіки та науки, особливості законодавства, культурних та природних чинників.

Враховуючи, що проєкт реалізується зазвичай у конкретному середовищі, слід мати на увазі й зовнішні фактори, в яких він реалізується. Такими факторами є політичні, економічні, соціальні, правові, науково-технічні, природні та екологічні.

До внутрішніх належать чинники, пов'язані з організацією проєкту. Організація проєкту є розподілом прав, відповідальності та обов'язків між учасниками проєкту.

Учасники проєкту - це люди або організації, які залучені до виконання проєкту, а також ті, хто залежить або зацікавлений у результатах проєкту, його успішному виконанні.

Учасники проєкту реалізують різні інтереси у процесі здійснення проєкту, формують власні вимоги відповідно до цілей та мотивації і впливають

на проєкт, виходячи зі своїх інтересів, компетенцій та ступеню залучення до проєкту.

Склад учасників проєкту, їх ролі, розподіл функцій і відповідальності залежать від типу, виду, масштабу й складності проєкту, а також від фаз його життєвого циклу.

Учасники можуть бути активними, тобто такими, які самостійно реалізують діяльність по проєкту чи діяльність, результати якої впливають на проєкт (взаємодіють з проєктом), та пасивними, тобто тими, хто відчуває вплив зі сторони проєкту.

Крім того, учасники можуть бути безпосередніми (активними чи пасивними), тобто учасниками самої діяльності по проєкту, або непрямими (активними чи пасивними), тобто учасниками діяльності, яка реалізується об'єктами навколишнього середовища і впливає на проєкт чи відчуває вплив проєкту.

Згідно з РМВоК розрізняють таких учасників проєкту:

Менеджер (керівник) проєкту (Project Manager) – особа, відповідальна за управління проєктом.

Спонсор (куратор) проєкту (Project Sponsor) – особа усередині або поза організацією, що забезпечує фінансові ресурси проєкту.

Замовник (Project Customer) – особа (організація), яка приймає результати роботи і платить за її виконання.

Користувач продукту проєкту (User) – особа усередині або поза організацією, яке використовуватиме результати проєкту.

Зацікавлені сторони в проєкті (Stakeholders) – учасники проєкту.

Виконуюча організація (Performing organization) – організація, співробітники якої безпосередньо залучені у виконання проєктних робіт.

Члени проєктної команди (Project team members) – група, що виконує роботу за проєктом.

Впливові особи (Influencers) – особи або групи осіб, які прямо не будуть розпоряджатися або використовувати результати проєкту, але через своє положення в організації можуть вплинути, позитивно або негативно, на просування проєкту.

Проєктний офіс (Project Management Office) — підрозділ, що прямо або побічно відповідає за результат проєкту.

Команда управління проєктом — частка проєктної команди, що бере участь в управлінні.

Наведений перелік учасників може змінюватися і доповнюватися залежно від умов конкретного проєкту.

Виділяють ще учасника – «ініціатор» - учасник проєкту, який є носієм основної ідеї проєкту та ініціативи його реалізації. В якості ініціатора може виступати будь-який з учасників проєкту.

Серед численних учасників ІТ-проєктів можна виділити 8 функціональних ролей, найбільш типових для більшості організацій. Ці ролі умовно названі: топ менеджер, керівник проєкту, ресурсний менеджер,

керівник програми, співробітник офісу управління проєктами, виконавець (програміст або розробник), клієнт (кінцевий споживач продукту, замовник), адміністратор системи управління проєктами.

Синонімом словосполучення «учасники проєкту» є «зацікавлені сторони» (stakeholders) — цей термін був схвалений ISO і прийнятий стандартом компетенцій IPMA (ICB). Всі зацікавлені сторони можуть здійснювати вплив на проєкт прямо і опосередковано. Такі джерела впливу, як інтереси зацікавлених сторін, організаційна зрілість в управлінні проєктом, встановлений порядок управління проєктом, стандарти, проблеми, тенденції і повноваження можуть впливати на зародження і розвиток проєкту.

Тому успіх проєкту сьогодні визначається не стільки в умінні реалізувати проєкт в рамках обмежень за змістом, термінами, вартості, якості, ресурсів і ризику, скільки в умінні узгоджувати зміни з замовником і вищим керівництвом. Навіть останній РМВОК заявляє, що «Успіх проєкту повинен зв'язуватися з останніми базовими планами, схваленими уповноваженими зацікавленими сторонами».

Ось для того, щоб знати, хто з цих зацікавлених сторін зацікавлений в успіху проєкту, а хто в провалі, і створюється Реєстр зацікавлених сторін (або стейкхолдерів). Причому створюється він на самій ранній стадії, а саме на етапі ініціації проєкту.

Реєстр стейкхолдерів містить ідентифікаційну інформацію, оцінку та класифікацію зацікавлених сторін (ЗС) проєкту. У Реєстрі містяться:

1. Ідентифікаційна інформація: прізвище, ім'я та по батькові, посаду в організації, місце розташування, роль у проєкті, контактна інформація;
2. Оціночна інформація: основні вимоги та очікування, потенційний вплив в проєкті, найбільш цікавить фаза в життєвому циклі проєкту;
3. Класифікація зацікавлених сторін: внутрішні /зовнішні, підтримують / нейтральні / пручаються і т.д.

У Реєстрі стейкхолдерів також відбивається аналіз зацікавлених сторін проєкту. Аналіз найпростіше проводити по матриці влади / інтересів, групуючи стейкхолдерів на основі їх рівня повноважень («влади») та рівня зацікавленості («інтересу») щодо результатів проєкту. Метод дозволяє «загнати» стейкхолдерів в кластери і сформуванати стратегію відносин з кожним кластером ЗС.

Таблична частина Реєстру зацікавлених сторін може виглядати так, як показано на рис.1.

ПІП	Компанія, посада	Роль в проєкті	Контакти	Додаткова інформація	ВЛАДА	ЦІКАВІСТЬ
Іванов І.І.	Керівник проєктів	PM	+380.....		60%	100%
Петренко П.П.	Директор	Спонсор	+380.....		100%	50%
Сідоренко С.С.	Директор	Замовник	+380.....		90%	100%
Дятлов Д.Д.	Перевіряючий	Державна перевірка	+380.....		70%	0%
...
Макакін М.М.	Java-девелопер	Розробник	+380.....		0%	60%
...

Рис. 1. Реєстр зацікавлених сторін

Заповнення таблиці та аналіз ЗС:

1. Визначити всі потенційні зацікавлені сторони проєкту і істотну інформацію про них, таку як ролі, відділи, інтереси, знання, очікування та рівні впливу. Таким чином заповнюються перші 5 колонок зазначеної вище таблиці. Ключові ЗС, такі як спонсор або замовник, виявляються легко. Решта зазвичай визначаються під час спілкування зі стейкхолдерами. в Реєстрі бажано вказувати всіх потенційних зацікавлених сторін.

2. Заповнюємо колонку «Влада». Негативною влада не буває, тому обираємо одного з ЗС з максимальною владою і ставимо йому 100% (як правило це спонсор проєкту). Переглядаємо список ЗС знаходимо осіб з мінімальними можливостями впливу на проєкт і ставимо йому 0% влади. таким чином градуюємо шкалу влади, розставляємо іншим стейкхолдерам проценти впливу, попарно порівнюючи їх один з одним.

3. Заповнюємо колонку «Інтерес». Мається на увазі інтерес до успіху проєкту, який може бути, як позитивним, так і негативним. Як правило максимальна зацікавленість в успіху проєкту у керівника, йому і ставимо 100%. Знаходимо цю людину і ставимо йому -100%. Потім відповідно до шкали інтересу, розставляємо відсотки іншим ЗС.

4. Потім в MS Excel за даними стовпців «П.І.Б.», «Влада» і «Цікавість (інтерес)» створюємо матрицю (рис.2.)

5. Візуально об'єднуємо ЗС в кластера і формуємо для них стратегії управління:

- цікавість (інтерес) від -50% до + 50%, і влада від 0% до 50% - на того витрачаємо мінімум зусиль. Наша стратегія - «Спостерігати».
- інтерес від -50% до + 50%, і влада від 50% до 100% - як правило, це наглядові або регламентуючі органи, яким все одно що буде з результатом проєкту, але їм важливо щоб у проєкті були виконані певні законодавчі та нормативні стандарти. Стратегія - «Задовольняти», тобто виконувати те що вони вимагають або переконати, що в нашому проєкті це не потрібно.
- цікавість від + 50% до + 100%, влада від 0% до 50% - це група людей, які постійно відволікають команду нешкідливими, на перший погляд

питаннями, типу «Як там у вас справи?». Такі особи настирливо забирають величезну кількість дорогоцінного часу команди. Стратегія - «Інформувати». Бажано інформувати пасивно, тобто створити, наприклад, розсилку, або подібність соціальної мережі, або повісити великий монітор над входом до кімнати команди проєкту, та демонструвати інформацію.

– інтерес від -100% до -50%, влада від 0% до 50% - це клас людей, які намагаються зробити дрібну капость проєкту або просто «наврочити» щось. Стратегія - «Приховувати інформацію» або «дезінформувати».

– інтерес від + 50% до + 100%, влада від 50% до 100% - це ключові зацікавлені сторони проєкту, про які пишуть всі книги і стандарти з УП, і з якими керівникові проєкту потрібно працювати щільно і завжди. Це саме ті люди, які в підсумку вирішують був проєкт успішним чи ні. Стратегія - «Активно залучати». І не потрібно боятися зайвого участі цих стейкхолдерів в справах проєкту.

– інтерес від -100% до -50%, влада від 50% до 100% - це клас людей, з якими керівнику проєкту доведеться дуже тісно співпрацювати. Більш того, активно залучаючи людей з попереднього пункту, як ресурс для «боротьби» з ними. Стратегія - «Активно оборонятися» або навіть «Нападати».

На наступному кроці необхідно розробити план дій – як впливати на кожного з стейкхолдерів, в залежності від того, в якому він квадраті, і з урахуванням внутрішніх умов проєкту. По кожному з стейкхолдерів у менеджера проєкту повинні бути готові відповіді на питання:

– У чому особа зацікавлена? Які цілі особи в проєкті?

– Як зміниться життя особи, якщо проєкт буде успішним? А якщо неуспішним?

– Чому людина знаходиться саме в цьому квадраті? Чи можна перетягнути її до іншого?

– Як особа може допомогти та/або зашкодити проєкту?

– Як не дати нашкодити і, якщо це сталося, як нейтралізувати наслідки?

– Як посилити позитивний вплив цієї особи на проєкт?

Коли план дій готовий, можна працювати з кожним і регулярно оновлювати матрицю, адже з'являються нові стейкхолдери, старі переходять в інші квадранти

–

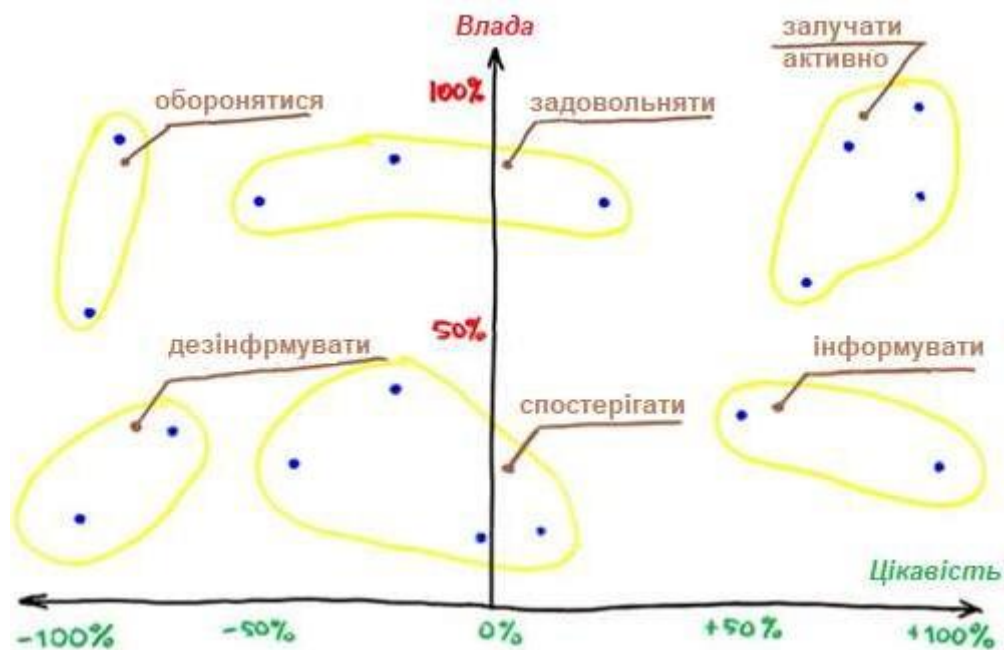


Рис. 2. Матриця зацікавлених сторін

Завдання на лабораторну роботу

1. Створити реєстр зацікавлених сторін, матрицю зацікавлених сторін та сформувані стратегії поведінки для груп зацікавлених сторін проєкту відповідно до проєкту який було обрано в роботі 2.
2. Визначити групу стейкхолдерів, до яких будуть належати користувачі продукту. Навести короткий опис їх ролей.
3. Підготувати презентацію та звіт з лабораторної роботи та презентацію завантажити до СЕН ЧДТУ.

Зміст звіту

1. Титульний аркуш (додаток А).
2. Тема і мета роботи.
3. Короткі теоретичні відомості.
4. Реєстр зацікавлених сторін, матриця зацікавлених сторін та стратегія поведінки для груп зацікавлених сторін.
5. Група стейкхолдерів, до яких будуть належати користувачі продукту з описом їх ролей
6. Висновки.

ЛАБОРАТОРНА РОБОТА №4

Планування та ідентифікація ризиків

Мета роботи: навчитися планувати та ідентифікувати ризики ІТ- проекту

Порядок виконання роботи

1. Вивчити теоретичні відомості.
2. Виконати завдання до лабораторної роботи згідно проекту лабораторної роботи № 2.
3. Скласти та оформити звіт.

Теоретичні відомості

Управління ризиками - це цілеспрямовані процеси, пов'язані з ідентифікацією, аналізом ризиків та прийняттям управлінських рішень, спрямованих на зниження ймовірності виникнення несприятливого результату і мінімізацію можливих втрат протягом усього життєвого циклу проекту.

Під класифікацією ризику слід розуміти розподіл ризику на конкретні групи за певними ознаками для досягнення поставлених цілей. Класифікаційна система ризиків включає групу, категорії, види, підвиди і різновиди ризиків. Науково обґрунтована класифікація ризику дозволяє чітко визначити місце кожного ризику в їх загальній системі. Вона створює можливості для ефективного застосування відповідних методів, прийомів управління ризиком. Кожному ризику відповідає своя система прийомів управління ризиком.

Метою використання класифікаторів є більш зручне використання інформації про різні види ризиків, які відрізняються один від одного місцем, часом, причиною виникнення, і застосування доступних методів управління ризиками в залежності від їх класифікації.

В ІТ-проектах класифікація ризиків дає основу для стандартизації термінології, необхідної для моніторингу та звітності, є незамінною для створення бази знань. Під час виявлення класифікації ризиків проектна команда впорядковує одночасну роботу з великим числом ризиків, надаючи відповідний спосіб групування схожих ризиків. Незважаючи на багато рекомендацій професійних асоціацій та державних установ, до сих пір не існує єдиної класифікації ризиків. Причин може бути декілька:

- одні і ті ж ризики можуть трохи відрізнитися за змістом для різних видів діяльності і різних типів проектів;
- на практиці існує дуже велика кількість різних проявів і ризиків, що залежать від специфіки конкретного проекту і різних факторів;
- в силу традицій один і той же вид ризику може позначатися різними термінами в залежності від прийнятої термінології;
- часто виявляється досить складним розмежувати окремі види ризику і чітко сформулювати вимоги до визначення видів ризику;
- вибір класифікації ризиків буде залежати від професійних уподобань менеджера проекту.

Для гарантованого виявлення якомога повнішого списку ризиків, потенційно впливають на успіх проєкту, а також групування схожих ризиків для подальшого аналізу за доцільне провести класифікацію ризиків відповідно до певними ознаками.

Виділяють наступні ознаки для класифікації ризиків:

1. Функціональні. До функціонального різновиду ризиків відносяться ризики, обумовлені функціональною спрямованістю, тобто фінансові ризики, ризики підрядників тощо.

2. За областями прояви. До цієї групи ризиків належать політичні ризики, інформаційні, стратегічні тощо

3. Структурні. Структурні ризики характеризують управлінську структуру та етапи ІТ-проєкту. На підставі аналізу структури проєкту можна виділити ризики програмування, тестування, налагодження тощо.

4. За часом виникнення. Тимчасові ризики визначаються етапами життєвого циклу проєкту, який починається з процесу його підготовки та розробки техніко- економічного обґрунтування і закінчується етапом його реалізації та остаточного завершення. Етапність життєвого циклу ІТ-проєкту дозволяє говорити про наступні види тимчасових ризиків: ризики розробки, узгодження, приймання, супроводу тощо.

5. За областями управління. За основу беруться найбільш типові області управління проєктом, і виділяються ризики, пов'язані з цією областю: ризики інтеграції, фінансові ризики, тимчасові ризики, ризики персоналу, комунікаційні ризики, ризики невідповідності якості, ризики постачальників та інші.

6. За метриками якості. Ризики, пов'язані з можливим зниженням якісних характеристик ІТ-проєкту, такі як ризики безпеки, доступності, продуктивності тощо

За іншими специфічними ознаками, наприклад: за типом джерела ризику; за ступенем впливу і характером наслідків (критичні / допустимі); за ступенем керованості (контрольовані / частково контрольовані); за ступенем передбачуваності (передбачувані / непередбачувані); за областю походження (відомі / невідомі, зовнішні / внутрішні ризики).

Вибір ознак класифікації ризиків буде залежати від специфіки ІТ-проєкту та професійних переваг менеджера проєкту. Одні ті ж ризики можуть трохи відрізнятися за змістом для різних видів діяльності і різних типів проєктів. На рис. 1 в якості ознаки класифікації обрана ступінь керованості ризиків.

Використовуючи класифікатори, можна виділити ризики, які негативно впливають на реалізацію ІТ-проєкту, і скласти список ідентифікованих ризиків.

Основні ризики, як правило, характерні для будь-яких ІТ-проєктів, полягають у недотриманні термінів реалізації проєкту, перевищенні вартості і недотриманні параметрів якості. Часто причиною виникнення цих ризиків, особливо в ІТ- проєктах, є неготовність підприємства до реалізації подібних проєктів



Рис. 1. Приклад класифікації ризиків ІТ-проекту за ступенем керованості

Розглянемо ризики, які є характерними майже для всіх ІТ-проектів.

Операційні ризики. Операційні ризики припускають можливість непередбачених втрат внаслідок технічних помилок при проведенні операцій, навмисних і ненавмисних дій персоналу, аварійних ситуацій, збоїв апаратури, несанкціонованого доступу до інформаційних систем тощо.

Технологічні ризики. Характеризують здатність інтегрованого рішення «коректно накладатися» на наявну інфраструктуру та інформаційні потоки і часто пов'язані з одноосібним прийняттям проектним менеджером рішень щодо ризиків (ідентифікація, аналіз, вибір методу реагування). Чим більше і складніше проект, тим вище цей ризик, особливо в масштабах промислової організації. Говорячи про мінімізацію технологічних ризиків, слід робити інвестиції більш «короткими», а проекти дробити на кілька взаємозамінних складових. Це допоможе уникнути неприємних наслідків при помилковому інвестуванні на одному з етапів.

Фінансові ризики. Припускають погіршення бізнес-показників в результаті неправильно обраної ІТ-стратегії, неякісне впровадження може спровокувати виникнення фінансових ризиків. Що стосується відповідальності за ризики інвестицій в ІТ, то більшість експертів вважають, що керівник ІТ-департаменту не повинен нести одноосібну відповідальність за невдалі впровадження.

Технічні ризики. Практично в будь-якому ІТ-проекті в промисловості існують ризики, пов'язані з технікою (відмова і збої в роботі обладнання, помилки в монтажі і т. П.). Технічні ризики включають в себе ризики отримання негативних результатів впровадження проекту, недосягнення запланованих параметрів, виникнення різного роду проблем при використанні нових технологій і продуктів, збій і поломку обладнання.

Ризики оцінки термінів. Для більшості ІТ-проектів, особливо в проектах по розробці і впровадженню програмного забезпечення (ПЗ) великих

промислових холдингів, характерні помилки в оцінках термінів робіт проекту. При цьому реальні терміни робіт можуть відрізнятись від запланованих в рази.

Інтеграційні ризики. Завжди високі, особливо в великих промислових організаціях, оскільки будь-яке ІТ-рішення повинно бути інтегровано в існуючу інфраструктуру організації, не порушуючи при цьому безперервності операційної діяльності та виробництва продукції. Найбільш характерні ризики переходу до нової системи, які включають в себе витрати на зупинку підприємства або його складових під час впровадження ІТ-рішень, навчання персоналу та інше

Ризики неприйняття продукту проекту користувачами. Впровадження технологічних рішень без підтримки організаційних перетворень. Будь-який проект, в тому числі в сфері ІТ, - це в першу чергу зміна технології роботи. Технічна складова будь-якого проекту безумовно важлива, але не менш важлива організаційна частина. Необхідно переконатися, що всі майбутні користувачі системи в промисловій організації розуміють необхідність введення нових технологій, процесів, регламентів взаємодії та інше.

Комерційні ризики (або ризики третім особам). Можуть бути визначені на основі факторів бізнесу, наприклад надійності постачальників, платоспроможності замовників, досвіду співвиконавців в подібних проектах, пов'язані з вибором технології і постачальника. Необхідно оцінити успішність технології на ринку, її актуальність протягом життєвого циклу ІТ-проекту, доступність необхідного апаратного та програмного забезпечення, його якість і частоту модернізації та інше. Також слід оцінити надійність постачальника необхідних технологій.

Ризики персоналу. Ризики персоналу пов'язані з кваліфікацією, професіоналізмом і бажанням у персоналу якісно і в строк виконувати свою роботу. Найбільш часто зустрічаються складності при роботі з персоналом – це нездатність або недосвідченість співробітників, які беруть участь в проекті, застосовувати використовувані технології, недостатність досвіду менеджера проекту, часте і суперечливе зміна вимог замовником, плінність кадрів.

В окремих випадках доцільно застосовувати техніку структурування ризиків з використанням структурної декомпозиції ризику (Risk Breakdown Structure, RBS), створюється за аналогією, групуючи ризики за джерелами появи і служить для визначення сумарного впливу ризиків.

З огляду на різні способи впровадження інформаційних систем в табл. 1 розглянуто ризики для кожного виду впровадження.

Для більш ефективної роботи та уникнення двозначних трактувань ризику бажано на початковому етапі продумати чітке формулювання ризику. Природно, що одні й ті ж ризики можуть мати різні формулювання у різних менеджерів проекту в залежності від використовуваної термінології. І навпаки, одна й та ж формулювання ризику може відрізнятись для різних видів діяльності і різних типів проектів.

Формулювання ризику - це «вираження природною мовою причинно-наслідкового зв'язку між реально існуючим фактором проекту (поточним

станом справ) і потенційно можливим, ще не те, що сталося подією або ситуацією».

Перша частина формулювання ризику називається умовою і містить опис існуючого чинника або особливості проекту, які, на думку проектної групи, можуть зробити результат проекту збитковим або скоротити одержувану від проекту прибуток.

Друга частина формулювання ризику називається наслідком і описує ту небажану ситуацію, якої слід уникнути. Перша і друга частини формулювання описують не жорстку, але можливу, з імовірністю менше одиниці, при чинно-наслідковий зв'язок (рис. 2).

Найчастіше виявляється вельми складним дотримуватися заданої формулювання ризику протягом всього проекту. Однак детальна форма опису ризику зручна тим, що на ранніх етапах процесу управління ризиками вона пов'язує наслідок ризику з видимими (і потенційно контрольованими) умовами. Використання альтернативного підходу, при якому проектна група не концентрується на описі умов на фазі виявлення ризиків, веде до того, що на більш пізньому етапі (при розробці стратегій управління) доводиться відновлювати умови та причини ризиків.

В результаті процесу виявлення ризиків повинні бути отримані чіткі, однозначні й узгоджені формулювання, представлені у вигляді списку ризиків для подальшого аналізу. Коректне формулювання ризиків зазвичай надає значний обсяг додаткової корисної інформації, включаючи виявлення першопричин ризиків.

Ефективність управління залежить від правильного і повного розуміння тих ризиків, з якими стикається проектна група.

Таблиця 1 –Класифікація ризиків в залежності від типів впровадження

Тип впровадження	Ризики	Приклад / коментар
1. «Коробкове впровадження» (приклад: «1С»), що не потребує доробки системи	Ризики порушення строків	Затягнуті строки впровадження у зв'язку з труднощами перенесення даних
	Інтеграційні ризики	Відсутність інтеграції з наявною базою даних
	Ризики неприйняття продукту проекту	Відсутність залученості користувачів під час вибору системи. Протидія при впровадженні
	Комерційні ризики	Помилки в розрахунку вартості кількості ліцензій, вибір ненадійного постачальника
2. «Впровадження типового рішення» (приклад: «SAP») з необхідністю налаштування системи	Технічні ризики	Збої та поломки системи, відмова чи збої в роботі обладнання, помилки в налаштуванні системи
	Ризики порушення строків	Помилки в оцінці строків впровадження й налаштування системи
	Інтеграційні ризики	Відсутність інтеграції з наявною системою/базою даних, помилки перенесення даних
	Ризики неприйняття продукту	Відсутність залученості користувачів під час впровадження й налаштування системи

	Комерційні ризики	Перевитрати коштів на додаткові строки впровадження й налаштування нової функціональності
	Ризики відсутності технології	Відсутність єдиної методології та планів впровадження, взаємодії учасників команди
3. Розробка й впровадження системи «з нуля» з необхідністю кодування й розробки системи	Технічні ризики	Помилки кодування, системні збої, несумісність з використаним обладнанням
	Ризики порушення строків	Помилки в оцінці строків розробки, тестування й налаштування системи
	Інтеграційні ризики	Відсутність планів по інтеграції на етапі розробки системи
	Ризики неприйняття продукту	Відсутність залученості користувачів під час розробки
	Комерційні ризики	Вибір неякісних постачальників чи ненадійних зовнішніх розробників
	Ризики відсутності технології	Відсутність методології та планів впровадження, взаємодії учасників команди

Усвідомлення різноманіття можливих проблем і рівня потенційних збитків може навести зневіру на співробітників. Як наслідок проєктна група буде дивитися на виявлення ризиків як на негативну діяльність. Знання про існування ризиків – необхідна умова ефективної роботи над ними. Отже, перспективи успіху проєктної групи лише збільшуються від проведення роботи по виявленню ризиків.



Рис. 2. Формулювання ризику, згідно з методологією MSF

Відкрите протоколюються обговорення ризиків призводить до чіткого роз'яснення ролей і відповідальності членів проєктної групи як під час планових заходів щодо профілактики ризиків, так і при вирішенні проблем, які можуть виникнути. Це звільняє від додаткових трудовитрат і дозволяє проєктній групі зосередитися на своїй роботі.

Члени проєктної групи і особливо її керівники повинні завжди трактувати виявлення ризиків як позитивний фактор. Це дає можливість отримати максимально повну інформацію про виникаючі ризики. Негативне сприйняття ризиків заважає проєктній групі обговорювати тему.

Як і будь-яка інша діяльність в проєкті, управління ризиками вимагає часу і витрат ресурсів. Тому ця робота обов'язково повинна плануватися. На етапі планування ризиків (першому етапі процесу управління ризиками) відбуваються організація робіт і планування дій з управління ризиками з

прив'язкою до життєвого циклу ІТ-проєкту. Тобто планування управління ризиками - це процес визначення підходів і планування операцій з управління ризиками проєкту.

Під час етапу планування необхідно відповісти на наступні питання: як буде реалізовуватися процес управління ризиками? З яких кроків складається цей процес? Які саме дії, ролі учасників і їх обов'язки, результати характеризують кожен крок? Хто буде здійснювати дії з управління ризиками? Які для цього потрібні навички/кваліфікація? Який інструментарій та які методики будуть застосовані? Який термінологічний апарат буде використаний для класифікації та оцінки? Яка процедура пріоритетності ризиків? Як будуватимуться плани управління ризиками та плани заходів щодо пом'якшення можливих негативних наслідків? Які дії будуть робити окремі члени проєктної групи для управління ризиками? Як буде виконуватися моніторинг управління ризиками? Яка інфраструктура (бази даних, програмні інструменти, сховища даних) буде використана для забезпечення процесу управління ризиками?

Планування управління ризиками може включати в себе рішення по організації, кадрового забезпечення процедур управління ризиками проєкту, вибір кращої методології, джерел даних для ідентифікації ризиків, регламент виконання процедур. Важливо спланувати управління ризиками, адекватне як рівню невизначеності і ризику, так і важливості проєкту для організації.

За статистикою компанії IBM, на стадії попереднього опрацювання та планування впровадження проєкту (перші 10%) закладається більше 90% його корисності. Саме в цей час закладаються найбільш болючі наслідки: зрив термінів, нереальний бюджет, збиткова функціональність. При цьому консультанти і замовники бачать проблеми неоднаково, що призводить до різних методів управління ризиками, які плануються на ранніх етапах. Кожна компанія повинна розробити власний план управління ризиками, що містить загальні регламенти та рекомендовані методики. Метою етапу планування управління ризиками є розробка плану управління ризиками, який містить:

обґрунтування вибору методології управління ризиками. В залежності від обраної методології можуть виконуватися певні етапи, використовуватися різні засоби управління ризиками;

детальні плани управління ризиками. Ретельне і докладне планування управління ризиками дозволяє виділити достатню кількість часу і ресурсів для виконання операцій з управління ризиками, визначити загальні підстави для оцінки ризиків, підвищити ймовірність успішного досягнення результатів проєкту. Дані про ресурси та оцінка вартості заходів, необхідних для управління ризиками, включаються в базовий план вартості проєкту. Необхідно передбачити забезпечення планів управління ризиками за допомогою їх інтеграції до загального процесу управління ІТ-проєктом;

визначення ролі і відповідальності в процесі управління ризиками та забезпечення залучення учасників. Крім розподілу ролей і відповідальності, необхідно виділити список позицій виконання, підтримки та управління ризиками для кожного виду операцій, включених до плану управління

ризиками, призначення співробітників на ці позиції і роз'яснення їх відповідальності;

визначення підходів, інструментів і джерел даних, які можуть використовуватися для управління ризиками в даному проєкті. Організації можуть мати заздалегідь розроблені підходи до управління ризиками, наприклад категорії ризиків, загальні визначення понять і термінів, стандартні шаблони, схеми розподілу ролей і відповідальності, а також певні рівні доступу до прийняття рішень, що необхідно прописати в плані управління ризиками на етапі планування;

визначення граничних рівнів ризиків. Ставлення до ризику і толерантність до ризику організацій і осіб, що беруть участь в проєкті, впливає на план управління проєктом. Воно повинно бути зафіксовано в викладі основних принципів і підходів до управління ризиками;

визначення методів ідентифікації та оцінки ризиків, критеріїв пріоритетності ідентифікованих ризиків. Залежно від обраної методології відбувається визначення термінів і частоти виконання ідентифікації та оцінки ризиків протягом всього життєвого циклу проєкту, а також визначення методів по ідентифікації та оцінки ризиків;

принципи обліку і документування. На етапі планування відбуваються визначення частоти і формату звітності, розробка шаблонів для документування процесу управління ризиками, визначаються регламенти та правила написання та оформлення документів.

Планування управління ризиками має бути завершено на ранній стадії планування проєкту, оскільки воно вкрай важливо для успішного виконання інших процесів. План управління ризиками проєкту розробляється на основі внутрішніх документів підприємства, а також контрактів із зовнішніми зацікавленими сторонами.

Процедура ідентифікації ризиків повинна проводитися регулярно протягом життєвого циклу ІТ-проєкту для виявлення втрачених і нових утворилися потенційних ризиків. Заходи щодо виявлення ризиків можуть прив'язуватися до часовим графіком (наприклад, бути щоденними, щотижневими або щомісячними) або віх проєкту.

Після формування плану управління ризиками починається етап ідентифікації ризиків, на якому визначаються ризики, здатні вплинути на проєкт, і документуються характеристики цих ризиків. Ідентифікація ризиків - це ітеративний процес, який періодично повторюється на всьому протязі проєкту, оскільки в рамках його життєвого циклу можуть виявлятися нові ризики. Тому процедура ідентифікації ризиків повинна проводитися регулярно протягом реалізації проєкту для виявлення втрачених і нових утворилися потенційних ризиків. В ідентифікації ризиків має бути задіяно якомога більше учасників: менеджерів проєкту, замовників, користувачів, незалежних фахівців.

В рамках процесу ідентифікації ризиків ІТ-проєкту необхідно:

- визначити, які ризики можуть вплинути на проєкт;
- забезпечити ітеративний процес виявлення ризиків;
- організувати методи ідентифікації та наради за проєктом;

- аналізувати всі можливі припущення і обмеження;
- аналізувати сильні і слабкі сторони, загрози і можливості;
- забезпечити своєчасне і повне документування ризиків проєкту;
- забезпечити чіткі і своєчасні комунікації.

В ідентифікації ризиків має бути задіяно якомога більше учасників: менеджерів проєкту, замовників, користувачів, незалежних фахівців, оскільки потрібно розуміння місії проєкту, його предметної області, цілей замовника, інвестора та інших зацікавлених осіб.

Для збору інформації про ризики можуть застосовуватися різні підходи.

Серед цих підходів найбільш поширені:

- 1) огляд документації;
- 2) метод мозкового штурму;
- 3) метод Дельфі (Delphi);
- 4) SWOT-аналіз;
- 5) інтерв'ю (опитування експертів);
- 6) контрольні списки;
- 7) аналіз припущень / сценаріїв;
- 8) причинно-наслідкові діаграми;
- 9) структурування ризиків з використанням структурної декомпозиції

ризиків.

Огляд документації. Метод використовується для початкового ознайомлення з проєктом і передбачає проведення огляду існуючих документів групою управління ризиками проєкту, включає структурований аналіз плану проєкту та наявних пропозицій (обмежень) як на рівні всього проєкту, так і на рівні окремих робіт.

Вихідні дані для виявлення і опису характеристик ризиків можуть братися з різних джерел. В першу чергу це база знань організації. Інформація про виконання колишніх проєктів може бути доступна в архівах попередніх проєктів. Слід пам'ятати, що проблеми завершених і виконуваних проєктів - це, як правило, ризики в нових проєктах.

Іншим джерелом даних про ризики проєкту може служити різноманітна інформація з відкритих джерел, наукових праць, маркетингова аналітика і інші дослідницькі роботи в даній області. Нарешті, багато форумів з програмування можуть дати безцінну інформацію про виниклі раніше проблеми в схожих проєктах (рис. 3).

Серед документів, які можуть бути ефективно використані з метою ідентифікації ризиків для максимально повного списку ризиків, пропонуються наступні документи: опис продукту; розклад проєкту; передбачувані витрати і терміни; ресурсний план; план поставок; перелік намірів та обмежень.

Додатково може використовуватися документація попередніх проєктів і доступна зовнішня інформація - комерційні бази даних, навчальні посібники, спеціалізовані видання, відкриті публікації по схожим проєктам та інше.

При аналізі планів проєкту проводиться аналіз відповідності завдань за планом договірних документів і раніше прийнятим угодам, відповідності виділяється часу масштабами завдань, можливості керувати і контролювати

роботи по представленому плану, аналіз процедур взаємодії, методологій і підходів до роботи і порядку виконання робіт.

Кожен проєкт замислюється і розробляється на підставі ряду гіпотез, сценаріїв і припущень. Як правило, в описі змісту проєкту перераховуються прийняті допущення - фактори, які для цілей планування вважаються вірними, реальними або певними без залучення доказів. Невизначеність в припущеннях проєкту слід також обов'язково розглядати як потенційне джерело виникнення ризиків проєкту.

Аналіз допущення дозволяє ідентифікувати ризики проєкту, що походять від неточності, несумісності або неповноти припущень.

Метод мозкового штурму. Метод «мозкового штурму» - найбільш поширений метод ідентифікації ризиків. Мета - скласти перелік можливих ризиків, які пізніше можуть бути відібрані в процесі остаточного формування списку ризиків. При використанні цього методу організовуються зустрічі з найбільш кваліфікованими фахівцями, які висловлюють свої ідеї за ризиками проєкту.

На першому етапі всі члени команди називають ризики, які здаються їм найбільш актуальними. Всі пропозиції записуються без обговорення, це триває до тих пір, поки є ідеї, які можна записати, або до настання певного терміну. Зустріч може виявитися більш вдалою, якщо учасники підготуються заздалегідь, вибравши певні категорії ризиків. На другому етапі відбувається обговорення запропонованих ризиків. Залишають тільки ті ризики, які команда визнала цікавими для подальшого розгляду. Зустрічі проходять без перерви і без коментарів з приводу судження кожного - суперечки і зауваження не допускаються. Потім ризики групуються за типами та їх характеристикам.

Основним недоліком цього методу є те, що на зустрічах учасники не мають можливості виступити анонімно, що може вплинути на якість результату, в разі якщо учасник боїться виступити з ідеєю, зворотної думки начальника або більш авторитетного лідера.

Метод Дельфі. Метод Дельфі дозволяє досягти однакості експертів і фахівців в галузі управління ризиками. В даному методі використовується група експертів від семи до десяти чоловік. Метод багато в чому схожий на метод мозкового штурму, однак є важливі відмінності. По-перше, при застосуванні цього методу експерти беруть участь в опитуванні анонімно. Метод передбачає анонімне заповнення анкет або опитувальників експертами з подальшим коректуванням оцінки на підставі опрацьованих комплексних результатів експертизи. Всі учасники опитування визначаються заздалегідь, але в експертизі виступають, не зустрічаючись одне з одним. Тому результат характеризується меншою суб'єктивністю, меншою упередженістю і меншим впливом окремих експертів. По - друге, опитування експертів проводиться в кілька етапів. На кожному етапі ведучий розсилає анкети, збирає і обробляє відповіді. Отримані результати розсилаються експертам для уточнення з урахуванням думки інших експертів. Кожен експерт має можливість ознайомитися з комплексними результатами експертизи, а потім дати нову, більш зважену оцінку. Погоджений список ризиків може бути отриманий в

результаті декількох ітерацій заочних погоджень. У підсумку виходить список ризиків в порядку їх важливості, заснований на незалежній думці кожного залученого експерта. Даний метод дозволяє досягти якогось спільної думки фахівців про ризики, а також зменшити необ'єктивність, упередженість аналізу і передчасне вплив окремих членів групи на думки інших експертів. Недоліком цього методу може бути складність реалізації всіх етапів.

SWOT-аналіз. SWOT-аналіз (Strength, Weaknesses, Opportunities and Threats) – оцінка якостей проєкту ІТ з точки зору сильних і слабких сторін, можливостей і загроз. SWOT-аналіз забезпечує аналіз проєкту з кожної з перерахованих сторін, щоб сформулювати припущення про основні загрози реалізації проєкту в цілому.

Так, наприклад, при проведенні SWOT-аналізу сильні сторони (Strength) описують більш розвинені, опрацьовані складові проєкту ІТ, такі як наявність досвіду персоналу в подібних проєктах, наявність необхідних навичок і кваліфікацій, наявність безперебійних технологій, що забезпечують досягнення цілей проєкту ІТ. Слабкі сторони (Weakness) описують складові проєкту, що представляють загрозу своїй неясністю, неповнотою, слабкою опрацюванням або організацією, наприклад нечітка постановка цілей або небажання співробітників переходити на нові технології. Можливості (Opportunities) представляють можливості по стратегії реалізації проєкту, додаткові переваги, одержувані за рахунок мінімізації витрат і максимізації результату, такі як зменшення трудовитрат за рахунок використання більш розвинених інформаційних технологій, забезпечення безпеки даних або збільшення швидкості обміну інформацією між співробітниками. Загрози (Threats) визначають чинники, які можуть перешкодити виконати проєкт з плановими результатами або взагалі зробити його реалізацію неможливою, не вигідною тощо. Наприклад несумісність технологій, неможливість перенесення даних з однієї системи в іншу, припинення фінансування проєкту в наслідок адміністративної реформи та організаційних змін (рис. 3).

При всіх позитивних характеристиках SWOT-аналізу основних і істотним недоліком є низька ступінь деталізації отриманих параметрів і можливість неоднозначної інтерпретації ризиків через занадто загальних категорій аналізу сторін проєкту.

Інтерв'ю (опитування експертів). Інтерв'ювання - метод проведення опитувань і інтерв'ю експертів фахівцями з управління ризиками проєкту. Цей метод зазвичай використовується для уточнення деталей ризику, дослідження нових ризиків або перевірки тих областей проєкту, які були переплановані. Ризики можуть бути ідентифіковані за допомогою опитувань, проведених фахівцями з управління ризиками проєкту. Особи, відповідальні за ідентифікацію ризиків, визначають відповідних фахівців в різних функціональних областях проєкту. Фахівці, що дають інтерв'ю, базуються на своєму досвіді, інформації про проєкт та інших джерелах і можуть надати величезну допомогу в уникненні повторного рішення одних і тих же проблем. Однак при використанні думки експертів потрібно дотримуватися обережності.

Якщо експерту довіряти беззастережно і брати його поради, не ставлячи питань, то проєкт може піти в неправильному напрямку.

Оскільки впровадження нової ІТ має здійснюватися з підтримки бізнесу, це обумовлює необхідність проводити інтерв'ю як з бізнес-користувачами, так і з відділом ІТ. У зв'язку з цим обмеженням методу може бути різниця в термінології і мовою спілкування між бізнесом та ІТ. Ще одним обмеженням є той факт, що залучення експертів, особливо найманих за межами своєї організації, може коштувати дорого. Тому необхідно забезпечити продуктивне і ефективне їх використання - перед опитуванням експерт повинен отримати ввідну інформацію і ясно зрозуміти мету опитування.

Контрольні списки/таблиці. Контрольні таблиці представляють собою списки типових ризиків, структуровані відповідно до деякої класифікацією, яка використовувалася на попередньому проєкті. Контрольні таблиці ідентифікації ризиків можуть бути розроблені не тільки відповідно до накопиченим досвідом по попереднім подібним проєктам, але і на основі інших джерел (платних баз даних, іншого). Перевагою використання контрольних таблиць є можливість швидкої і простої роботи за умови, що предметна область і цілі проєкту близькі. Недолік полягає в неможливості складання повної, вичерпної контрольної таблиці, так як користувач обмежений існуючими видами ризиків. З метою складання більш повного переліку потенційних подій контрольні таблиці слід використовувати на початковому етапі планування ризиків, доповнюючи їх специфічними ризиками конкретного проєкту.

<p>Сильні сторони</p> <ul style="list-style-type: none"> - Високий попит на подібні проєкти - Модульність, гнучкість - Популярність і зручність використання єдиного інформаційного середовища 	<p>Слабкі сторони</p> <ul style="list-style-type: none"> - Недостатність фахівців - Великі витрати на підтримку інформаційної системи - Складність підтримки через великий комплекс завдань - Небажання співробітників переходити на нові технології
<p>Можливості</p> <ul style="list-style-type: none"> - Нові технології доступу та обміну інформацією - Поліпшення структури і якості послуг - Збільшення швидкості обміну інформацією 	<p>Загрози</p> <ul style="list-style-type: none"> - Несумісність технологій - Неможливість перенесення даних з однієї системи в іншу - Припинення фінансування проєкту

Рис. 3. Приклад SWOT-аналізу для ідентифікації ризиків ІТ-проєкту

Аналіз припущень / сценаріїв. Аналіз припущень (мережеві методи аналізу) – метод, який досліджує правильність припущень і потім ідентифікує ризики, виходячи з правильності, повноти і послідовності припущень. Даний метод ідентифікації спрямований на визначення областей, які можуть бути схильні до ризику, виходячи з правильності, повноти і послідовності припущень про цій галузі. Аналіз припущень проводиться при аудиті документів проєкту і дозволяє формулювати потенційні ризики, виходячи з того, що висунуте припущення про проєкт може виявитися невірним. Часто для аналізу застосовується гіпотеза «що, якщо», за допомогою якої можна досліджувати правильність припущень про хід виконання проєкту ІТ на

кожному етапі життєвого циклу і ідентифікувати ризики, виходячи з правильності, повноти і послідовності припущень. Наприклад, якщо припущення, що «в найближчі 5 років в компанії планується збільшення даних в n раз і для передачі даних досить буде використовувати мережевий кабель з пропускною спроможністю» невірне, то існує потенційний ризик неможливості обміну комунікаціями при збільшенні інформації більш ніж в n раз.

Аналіз сценаріїв допомагає вибрати один з декількох варіантів реагування в разі настання одного з факторів ризику. Після визначення фактору ризику або невизначеності слід використовувати такі принципи аналізу сценаріїв:

- 1) визначити учасників прийняття рішень і відповідальних за виникнення ризику (хто приймає рішення)
- 2) визначити наслідки невизначеності (на що впливає)
- 3) визначити характер змін і спосіб реагування на ризики (що змінити і як)

Наприклад, **якщо** проєкт відчуває нестачу кваліфікованих фахівців, **то** можуть виникнути складнощі з впровадженням системи і її подальшою підтримкою. Або ж **якщо** система має надлишкової функціональності, **то** подальша підтримка системи виявиться складною і дорогою.

Перегляд і оцінка великої кількості сценаріїв є трудомістким процесом, але при цьому розширюють область пошуку рішень і збільшують ймовірність отримання найкращого рішення. Для кожного зміни передбачається, як правило, кілька альтернативних сценаріїв, з яких можна вибрати найбільш оптимістичний варіант.

Для оцінки сценаріїв змін необхідно використовувати класифікацію змін (соціальні, політичні, економічні, технологічні, технічні та інші), яка може відрізнитися в залежності від проєкту впровадження і стратегічних цілей в програмі ІТ-проєкту.

Головний недолік методу полягає в суб'єктивності вибору сценарію реалізації проєкту і відсутності обліку інформаційної невизначеності.

Причинно-наслідкові діаграми. Застосування причинно-наслідкових діаграм при ідентифікації ризиків ІТ-проєкту (рис. 4) представляє аналіз графічного відображення невизначеностей та інших аспектів проєкту. Застосування причинно-наслідкових діаграм дозволяє виявити причинно-наслідкові зв'язки, які можуть привести до виникнення ризику. Використання діаграм слід пов'язувати з використанням майбутніх ІТ, які оцінюються із застосуванням різних критеріїв і факторів на всіх етапах життєвого циклу, включаючи розробку і використання ІТ в конкретній організації.

Під час побудови діаграми, спочатку необхідно окреслити проблему, що вимагає рішення.

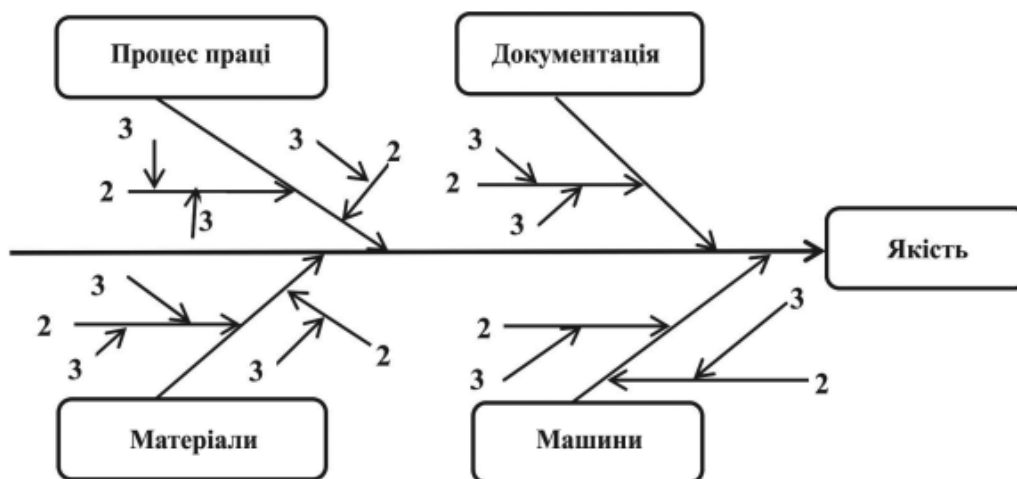


Рис.4. Приклад причинно-наслідкового діаграми (діаграми Ісікави) для ідентифікації ІТ-ризиків

Основним завданням є оцінка вищого рівня виходячи з взаємодії різних рівнів, а не з безпосередньої залежності від елементів на цих рівнях і виявлення найбільш значущих чинників. Розрахункову основу методу складають матриці попарних порівнянь, які заповнюються експертами. На підставі цих матриць обчислюються ваги цілей, критеріїв і альтернатив. Напрямок, який одержав найбільшу вагу, вносить найбільший внесок у досягнення головної мети.

Процедура побудови причинно-наслідкового діаграми складається з наступних етапів.

Етап 1. Побудова діаграми здійснюється справа – тут встановлюється основна мета дослідження – виявляється основна комплексна проблема – перший рівень.

Етап 2. Виявлення головних причин, що впливають на проблему - фактори 2 рівня. Далі будується «риб'ячий скелет», «хребет» якого становить пряма лінія, що відходить від показника якості. Головні причини з'єднуються з «хребтом» за допомогою стрілок, що представляють собою «великі кістки». Для опису основних причин проблем можна використовувати правило 5 W's – What, Who, When, Where, Why (рис. 5).

Так само для опису основних причин можна використовувати і інші категорії:

- Методи, машини, матеріали, підготовка персоналу ;
- Місце, Процедура, Люди, Політика;
- Оточення, Постачальники, Системи, Навички;

Етап 3. Знаходження вторинних причин, що впливають на головні причини – фактори 3 рівня. Їх розташовують у вигляді «середніх кісток», прилеглих до «великих». Тут також при необхідності визначаються причини третинного порядку, які впливають на вторинні і складають «дрібні кістки», – примикають до «середнім». На цьому побудова діаграми закінчено і необхідно провести заключний етап – ранжування і розрахунок.

Етап 4. Ранжуються причини за їх значимістю, виділяються особливо важливі, які імовірно мають найбільший вплив на той чи інший фактор проекту.

Використання діаграм є дуже наочним, хоча і трудомістким методом ідентифікації ризиків.

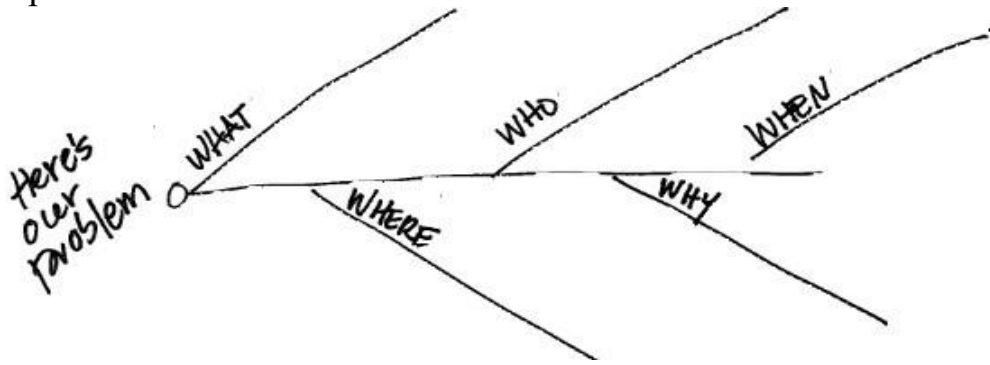


Рис. 5. Вид діаграми Ісікави, яка створена за правилом 5W's – What, Who, When, Where, Why

Структурування ризиків з використанням структурної декомпозиції ризику (Risk Breakdown Structure, RBS) сформовано за аналогією зі *структурної декомпозицією робіт проекту (WBS)* і дозволяє групувати ризики за джерелами появи для визначення сумарного впливу ризиків. Універсальний приклад ієрархічних рівнів RBS IT-проекту представлений в табл. 2.

Використання даного методу в області IT вимагає великого знання предметної області та наявності достатньої інформації.

Для ідентифікації ризиків необхідно залучати якомога більше людей з команди проекту. Якщо в ідентифікації бере участь тільки керівник проекту, він може не врахувати всіх можливих ризиків в силу обмеженості своїх знань. В результаті ідентифікації ризиків повинні бути отримані списки ризиків та ризикообразуючих чинників, при настанні яких можливе отримання негативного впливу на проект. Кожен ризик необхідно зареєструвати:

- дати найменування ризику і привласнити йому унікальний номер;
- визначити відповідального за даний ризик;
- описати причини виникнення ризику;
- описати наслідки настання ризику;
- зафіксувати статус ризику (новий / в роботі / закритий).

Таблиця 2. Структурна декомпозиція ризиків IT-проекту

Рівень 0	Рівень 1	Рівень 2	Фактори ризику
Сумарний ризик проекту	Ризики управління проектом	Корпоративні ризики	Історія, досвід, культура Стабільність організації Фінанси
		Ризики замовника та зацікавлених осіб	Досвід подібних проектів Визначення потреб та стабільність взаємовідносин
	Зовнішні фактори ризику	Ризики навколишнього середовища	Зовнішнє середовище Інфраструктура
	Внутрішні фактори ризику	Ризики реалізації	Складність реалізації Технологічні обмеження

			Нечітка постановка задачі та формулювання вимог Підтримка з боку вищого керівництва
		Організаційні ризики	Організаційний досвід керівників проекту Персональні навички та досвід ризик-менеджера Наявність достатніх ресурсів

Завдання на лабораторну роботу

1. Виконати ідентифікацію основних ризиків проекту з лабораторної роботи № 2 одним з методів, розглянутих в теоретичній частині роботи.
2. Підготувати презентацію.

Зміст звіту

1. Титульний аркуш (додаток А).
2. Тема і мета роботи.
3. Короткі теоретичні відомості.
4. Ризики проекту
5. Висновки.

ЛАБОРАТОРНА РОБОТА №5

Планування та проведення спринту (Scrum simulation)

Мета роботи: Навчитися планувати спринт, формувати беклог продукту, працювати з артефактами Scrum та відслідковувати прогрес за допомогою метрик.

Порядок виконання роботи

1. Вивчити теоретичні відомості.
2. Виконати завдання до лабораторної роботи згідно свого варіанту.
3. Скласти та оформити звіт.

Теоретичні відомості

Scrum – це гнучкий інструмент управління проектами, що базується на ітеративному підході та поступовій розробці продукту невеликими частинами.

1. Беклог продукту (Product Backlog)

Беклог – це впорядкований список вимог до продукту у вигляді user stories. Кожен елемент описує функціонал чи завдання, необхідне для досягнення цілей проекту.

- Беклог постійно змінюється та оновлюється.
- Пріоритизація здійснюється Product Owner-ом.
- Записи можуть оцінюватися у story points – відносних одиницях складності.

Оцінювання завдань у story points

Story points – це відносна одиниця вимірювання складності користувацьких історій у беклозі. Вони не виражаються в годинах чи днях, а показують:

- скільки зусиль потрібно для виконання задачі;
- наскільки завдання складне у порівнянні з іншими;
- які ризики та невизначеність воно має.

Як визначаються story points

1.1. Порівняння з еталонною задачею

- Команда обирає «базову» user story, наприклад просту задачу (реєстрація користувача = 2 SP).
- Інші завдання оцінюються відносно цієї.

1.2. Шкала чисел

- Найчастіше використовується ряд Фібоначчі: 1, 2, 3, 5, 8, 13, 21...
- Великі числа відображають завдання з високою складністю та невизначеністю.

1.3. Фактори, які враховуються при оцінюванні:

- Обсяг роботи (кількість функцій, обсяг коду).
- Складність (технічні труднощі, нові технології).
- Ризики (невідомі залежності, потреба у додаткових дослідженнях).

1.4. Методи оцінки

- Planning Poker: команда одночасно показує карти з оцінкою, потім

обговорює різницю.

- Афіне порівняння: задачі впорядковуються за складністю на шкалі.
- Bucket System: завдання розподіляються у «кошики» зі значеннями (1, 2, 3, 5, 8...).

Приклад

- «Реєстрація користувача» – 2 SP (проста задача).
- «Авторизація з підтвердженням e-mail» – 3 SP (трохи складніше).
- «Інтеграція з платіжною системою» – 8 SP (багато залежностей, складніша логіка).

2. Планування спринту (Sprint Planning)

Спринт – це фіксований проміжок часу (зазвичай 1 – 4 тижні), протягом якого команда реалізує певний обсяг роботи. На зустрічі з планування команда:

- формулює мету спринту (sprint goal);
- відбирає завдання з беклогу для Sprint Backlog;
- оцінює складність і розподіляє завдання між учасниками.

3. Щоденні зустрічі (Daily Scrum)

Daily Scrum – це короткі (до 15 хвилин) щоденні зустрічі команди. Їх мета:

- синхронізувати роботу;
- обговорити прогрес і проблеми;
- уточнити план дій на найближчу добу.

Формат часто будується навколо трьох запитань:

1. Що я зробив учора для досягнення мети спринту?
2. Що я зроблю сьогодні?
3. Які перешкоди заважають моїй роботі?

4. Огляд і ретроспектива (Review та Retrospective)

- Review – зустріч наприкінці спринту, де команда демонструє результат (інкремент продукту) стейкхолдерам.
- Retrospective – внутрішня зустріч команди для аналізу процесів: що вдалося, що варто покращити.

5. Burn-down Chart

Burn-down chart – це графік, що показує обсяг роботи, який залишився у спринті, у співвідношенні з часом (рис.1).

- Вертикальна вісь – кількість завдань чи story points.
- Горизонтальна вісь – дні спринту.
- Лінія спадання – очікуваний темп виконання роботи.
- Реальна крива показує фактичний прогрес.

Цей інструмент дозволяє швидко виявити затримки та оцінити, чи встигає команда досягнути мети спринту.

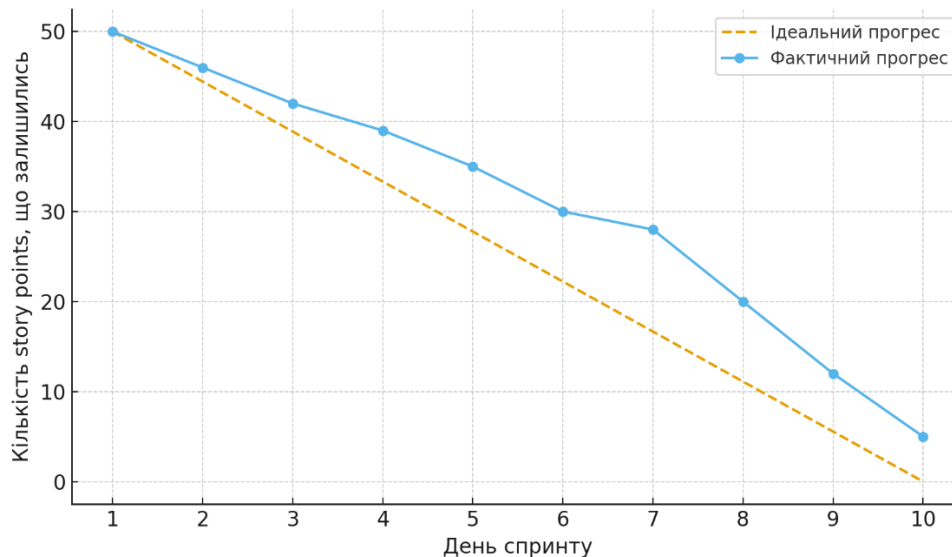


Рис.1. Приклад Burn-down chart

Завдання на лабораторну роботу

1. Складання Product Backlog
 - Сформувані список мінімум із 15 user stories для обраного проєкту.
 - Для кожної user story вказати: опис, пріоритет (Must/Should/Could).
 - Проведіть оцінювання їхньої складності у story points, використовуючи: ряд Фібоначчі (1, 2, 3, 5, 8, 13, 21) та метод Planning Poker (кожен учасник обирає оцінку, після чого команда доходить згоди).
 - Занесіть результати оцінювання у таблицю.
 - Проаналізуйте, які історії потребують найбільших зусиль, і чи можна розділити великі задачі на менші.

Приклад таблиці Product Backlog:

ID	User Story	Пріоритет	Story Points	Обґрунтування	Статус
US-01	Як користувач, я хочу зареєструватися, щоб мати власний профіль	Must have	2	Проста форма, стандартна логіка	To Do
US-02	Як адміністратор, я хочу блокувати акаунти для контролю порушень	Should have	8	Адмін-панель, перевірка прав доступу	To Do

2. Планування спринту
 - Обрати мету спринту (Sprint Goal).
 - Відібрати 5–7 найважливіших user stories у Sprint Backlog.
 - Розподілити завдання між учасниками.

Приклад таблиці Sprint Backlog

User Story	Завдання	Виконавець	Статус
US-01 Реєстрація	Створення форми	Студент 1	To Do
US-01 Реєстрація	Перевірка email	Студент 2	To Do

3. Симуляція Daily Scrum

– Кожен учасник під час симуляції відповідає на три питання:

Що я зробив учора?

Що зроблю сьогодні?

Які перешкоди існують?

– Записати короткі підсумки у таблицю.

Приклад запису Daily Scrum:

Учасник	Вчора	Сьогодні	Перешкоди
Студент 1	Створив макет форми	Реалізація перевірки email	Немає
Студент 2	Налаштував БД	Тестування авторизації	Чекаю API

4. Контроль прогресу (Burn-down Chart)

– Створити Burn-down Chart у Excel або Google Sheets:

по осі X – дні спринту,

по осі Y – кількість story points, що залишились.

– Порівняти ідеальний темп виконання (пряма лінія) з фактичним.

Зміст звіту

1. Титульний аркуш (додаток А).
2. Тема і мета роботи.
3. Короткі теоретичні відомості.
4. Таблиці з даними та діаграми.
5. Висновки.

ЛАБОРАТОРНА РОБОТА №6

Використання Kanban-дошки та управління потоком

Мета роботи: Набути практичних навичок щодо створення Kanban-дошки, застосування WIP-лімітів, аналізу руху завдань та обчислювання ключових метрик потоку.

Порядок виконання роботи

1. Вивчити теоретичні відомості.
2. Виконати завдання до лабораторної роботи відповідно до свого проєкту.
3. Скласти та оформити звіт.

Теоретичні відомості

Коли необхідно оптимізувати робочі процеси, звичайних to-do листів може бути недостатньо. У цьому випадку варто пошукати «чарівний» функціонал, який буде відслідковувати всі завдання, над якими працює команда проєкту, завдання, які тільки плануються, а також показувати повну картину по вже виконаним задачам.

Kanban-дошка – відмінне рішення цих проблем. Цей інструмент для спільної роботи над проєктами широко використовується в розробці ПЗ, маркетингу, будівництві, логістиці та в будь-яких рішеннях, де присутній постійний потік завдань. Kanban-підхід допомагає командам візуалізувати робочі процеси, грамотно аналізувати їх та підвищувати ефективність управління завданнями.

Метод Kanban вперше був застосований у виробничій системі Toyota у 40-х роках ХХ століття. Його завданням було зменшення витрат і прискорення виробничого циклу. В ІТ Kanban почав широко використовуватись на початку 2000-х років як більш гнучка альтернатива класичним методологіям управління проєктами.

Принципи Kanban

- Візуалізуй роботу. Завдання завжди відображаються у вигляді карток, що переміщуються між колонками.
- Обмежуй WIP. Надмірна кількість незавершених завдань гальмує команду.
- Керуй потоком. Завдання повинні рухатися без «заторів».
- Зроби правила явними. Команда погоджується на чіткі критерії завершення (Definition of Done).
- Впроваджуй петлі зворотного зв'язку. Регулярні наради дозволяють коригувати процес.
- Прагни вдосконалення. Kanban орієнтований на поступове підвищення ефективності.

3. Елементи Kanban-дошки

- Картки (Cards). Представляють завдання або user story.
- Колонки (Columns). Відображають етапи процесу (To Do → In Progress → Review → Done).

- WIP-ліміти. Встановлюють максимальну кількість завдань у колонці.
- Swimlanes. Горизонтальні розділи дошки для групування задач (наприклад, «Front-end», «Back-end»).

4. Метрики Kanban

1. Lead Time – загальний час від постановки завдання до його завершення. Відображає швидкість обслуговування клієнта. Розраховується за формулою:

$$\text{Lead Time} = \text{Дата}_{Done} - \text{Дата}_{ToDo}$$

Lead Time важливий для клієнтів (час очікування результату).

2. Cycle Time – час від початку роботи над завданням (*In Progress*) до його завершення (*Done*). Характеризує ефективність команди. Розраховується за формулою:

$$\text{Cycle Time} = \text{Дата}_{Done} - \text{Дата}_{InProgress}$$

Cycle Time важливий для команди (ефективність роботи).

3. Throughput – кількість завдань, виконаних за певний проміжок часу. Розраховується за формулою:

$$\text{Throughput} = \frac{\text{Кількість завершених завдань}}{\text{Одиниця часу}}$$

Наприклад, якщо за тиждень команда завершила 12 задач, то Throughput = 12 задач/тиждень

Throughput показує продуктивність на часовому інтервалі.

4. Cumulative Flow Diagram (CFD) – графік, що показує кількість завдань у кожній колонці дошки в часі. По осі X відкладається час, а по осі Y – кількість завдань. Дозволяє виявляти «затор» у конкретних етапах. Для кожного дня показують накопичену кількість завдань у колонках (*To Do*, *In Progress*, *Review*, *Done*). Аналізуючи CFD, можна зрозуміти: чи стабільний потік задач; чи є зростання черги в якійсь колонці; наскільки рівномірно завдання потрапляють у Done.

CFD дає візуальне уявлення про стабільність процесу та допомагає знаходити «вузькі місця».

5. Практичні правила ефективного використання Kanban

- Завжди підтримуйте дошку в актуальному стані.
- Якщо завдання «застряє» у колонці, необхідно визначити причини (нестача ресурсів, неясні вимоги, залежності від інших задач).
- Використовуйте візуальні маркери: кольори карток (баг, нова функція, документація), позначки пріоритетів.
- Встановлюйте реалістичні WIP-ліміти: наприклад, якщо команда має 3 розробників, у колонці In Progress має бути не більше 3–4 завдань.
- Використовуйте щотижневі або щоденні короткі обговорення, щоб

відстежувати стан завдань.

6. Kanban у порівнянні з іншими підходами

Kanban vs Scrum:

- Scrum працює ітераціями (спринтами), Kanban – безперервним потоком.
- У Scrum чітко визначені ролі, у Kanban – гнучкіша організація.
- Scrum планує фіксований обсяг роботи на спринт, Kanban дозволяє змінювати обсяг на ходу.

Kanban vs Waterfall:

- Каскадна модель суворо лінійна, Kanban – адаптивний і візуальний.

7. Інструменти для роботи з Kanban

- Онлайн-платформи: Trello, Jira, Asana, Miro, Notion та інші (табл.1)
- Фізичні дошки: стікери, маркери, дошка з колонками

Правильно підібране програмне забезпечення з функціоналом Kanban дозволяє зосередитися на пріоритетах, тримати команду в курсі того, що актуально і що буде відбуватися в найближчому майбутньому, а також контролювати всі робочі процеси.

Що включає в себе потужне ПЗ Kanban? На перший погляд, система Kanban може здатися складною і не завжди прийнятною. Але справжня сила Kanban в тому, що, як тільки команда починає застосовувати його принципи, то відразу може оптимізувати і стандартизувати робочі процеси. Проте, завжди існують рішення, які виконують роботу краще і ефективніше інших.

Таблиця 1. Перелік онлайн інструментів, орієнтованих на Kanban

Назва	Найкраще для
JIRA	для розробки програмного забезпечення Agile-командами; потужний функціонал backlog, Scrum, Kanban, інтеграції з CI/CD.
Trello	для приватного та командного використання; простий у використанні Kanban-інтерфейс; підходить для невеликих і середніх команд.
ClickUp	універсальний інструмент; підтримує Kanban, Scrum, Gantt та інші представлення; підходить для команд з різними процесами.
Asana	для персонального використання і для невеликих-середніх команд; хороша Kanban-дошка, базова функціональність керування тасками.
Wrike	для більших команд і бізнесів; має розширені можливості візуалізації робочих процесів, звітів, автоматизації.
Zoho Projects	для команд, що потребують гнучкого PM-інструменту з роботою з задачами, часом і автоматизацією робочих процесів.
Kanban Tool	для тих, хто хоче просту і візуально зрозумілу Kanban-дошку з аналітикою.
KanbanFlow	для простих Kanban-потоків; має таймер типу Pomodoro для фокусованої роботи.
Monday.com	для команд, які цінують кастомізацію, інтерактивні дошки та роботу з різними представленнями задач.
Teamwork.com	для агенцій або команд, що працюють над кількома проєктами одночасно; хороший контроль завдань і комунікація.

Miro	для візуалізації, брейнштормів, спільної роботи; можна використати Kanban-дошки та діаграми для планування.
MeisterTask	простий інтерфейс; інтеграція з інструментами для майндмепів; хороші канбан-функції.
Microsoft Planner	для команд, що вже працюють в Microsoft 365; інтеграція з Teams; базовий Kanban-потік.
LeanKit	для великих/складних робочих потоків; можливості деталізації, сегментації і масштабування.
ZenHub	інтеграція з GitHub; зручний для dev-команд, які використовують GitHub для репозиторіїв.
TeamGantt	якщо потрібні комбіновані представлення: Gantt-діаграми + Kanban; для проєктів зі строгими дедлайнами.

Головними критеріями «кращого» Kanban-інструменту є:

Kanban дошка – система, яка призначена для візуалізації процесів (завдань) за допомогою колонок (Swimlanes), обмежень (WIP limits), сабколонок, а також стандартного набору колонок («To-do», «In Progress» та «Done»).

Kanban картка – елемент системи Канбан, який призначає завдання за допомогою чек-листів та вкладень. Картки дозволяють пов'язувати завдання, додавати ієрархію і призначати необхідні ресурси.

Аналітика – система, що дозволяє створювати звіти.

Інтеграція – здатність інструменту легко інтегруватися з іншими системами управління проєктами.

Автоматизація – можливість налаштувати робочий процес відповідно до умов конкретного проєкту.

Завдання на лабораторну роботу

- Створення Kanban-дошки
 - Оберіть інструмент (Trello, Jira, ClickUp, Asana, Zoho Projects, Miro або ін.).
 - Налаштуйте дошку з мінімум 4 колонками: To Do / In Progress / Review / Done.
 - Створіть не менше 10 завдань на основі вашого продуктового беклогу (JIP-5).
- Налаштування WIP-лімітів
 - Встановіть WIP-ліміт для колонки In Progress (наприклад, ≤ 3 задачі одночасно).
 - Внесіть пояснення, чому обрано саме такий ліміт.
- Симуляція процесу виконання завдань
 - Переміщайте завдання з To Do → In Progress → Review → Done.
 - Зафіксуйте щонайменше 5 змін стану задач.
 - Занесіть дані у таблицю для розрахунку метрик.

Приклад таблиці для фіксації:

ID	Завдання	Дата To Do	Дата In Progress	Дата Done	Lead Time (дні)	Cycle Time (дні)
T-01	Реєстрація користувача	01.03	02.03	05.03	4	3
T-02	Авторизація з e-mail	01.03	03.03	07.03	6	4

4. Обчислення метрик

- Для кожного завдання порахуйте Lead Time та Cycle Time.
- Визначте Throughput (кількість задач, завершених за період).
- Зробіть висновок: чи є робочий процес стабільним.

5. Виявлення вузьких місць

- Побудуйте діаграму з кількістю задач у кожній колонці на кінець кожного дня.
- Проаналізуйте, де виникають затримки (In Progress, Review чи інші).
- Запропонуйте дії для покращення (зміна WIP-лімітів, розподіл ролей, автоматизація).

Зміст звіту

1. Титульний аркуш (додаток А).
2. Тема і мета роботи.
3. Короткі теоретичні відомості.
4. Скріншот створеної Kanban-дошки, таблиця руху завдань, розраховані метрики (Lead Time, Cycle Time, Throughput), діаграма для аналізу вузьких місць.
5. Висновки щодо ефективності роботи та пропозиції з оптимізації.

ЛАБОРАТОРНА РОБОТА №7

Agile-метрики та ретроспектива

Мета роботи: навчитися вимірювати та аналізувати ключові Agile-метрики (Velocity, Throughput, Lead Time, Cycle Time, Defect Rate), візуалізувати прогрес за допомогою діаграм (Burn-down, Burn-up, CFD) та опанувати проведення ретроспективи як інструменту безперервного вдосконалення командної роботи.

Порядок виконання роботи

1. Вивчити теоретичні відомості.
2. Виконати завдання до лабораторної роботи відповідно до свого проєкту.
3. Скласти та оформити звіт.

Теоретичні відомості

Agile – це підхід до управління проєктами, що базується на принципах гнучкості, ітеративності та орієнтації на цінність для замовника. Одним із ключових аспектів Agile є безперервне вдосконалення, яке реалізується через регулярні вимірювання результатів роботи та командні обговорення (ретроспективи).

Для оцінки ефективності в Agile-процесах застосовують Agile-метрики – кількісні та якісні показники, які дозволяють:

- відстежувати прогрес виконання завдань у спринті або потоці;
- прогнозувати продуктивність команди в майбутніх ітераціях;
- оцінювати якість продукту та стабільність процесів;
- виявляти проблеми та «вузькі місця»;
- створювати прозору систему звітності для стейкхолдерів.

Метрики використовуються не для контролю індивідуальної продуктивності, а для того, щоб допомогти команді працювати ефективніше. У поєднанні з ретроспективами вони формують цикл inspect & adapt («аналізуй та пристосовуйся»), що є основою Agile.

Основні Agile-метрики

1. Velocity – кількість story points, завершених за спринт. Використовується для прогнозування можливостей команди.
2. Throughput – кількість завершених задач за певний проміжок часу.
3. Cycle Time – час від початку роботи над завданням до завершення.
4. Lead Time – час від створення задачі до її завершення.
5. Defect Rate – кількість дефектів у відношенні до завершених user stories.
6. Burndown/Burnup Charts – графіки прогресу у виконанні задач.
7. Cumulative Flow Diagram (CFD) – показує стабільність потоку завдань.
8. Business Value Delivered, CSAT, NPS – метрики бізнес-цінності та задоволеності користувачів.

Ретроспектива

Ретроспектива – це регулярна зустріч команди після завершення ітерації (спринту), спрямована на аналіз процесів і пошук шляхів покращення. Популярні формати: Start/Stop/Continue, 5 Whys, Sailboat, Mad/Sad/Glad.

Головна мета ретроспективи – вироблення конкретних Action Items, які покращать роботу команди у наступних спринтах.

Баланс метрик

Команда повинна обирати 3–5 ключових метрик, які найбільше відповідають їхнім цілям. Надмірна кількість показників може призвести до мікроменеджменту, а неправильне використання – до спотворення результатів. Метрики мають допомагати команді, а не бути інструментом покарання.

Завдання на лабораторну роботу

1. На основі симуляції Scrum/Kanban з попередніх робіт зібрати дані: кількість завершених user stories; час виконання завдань; кількість помилок/дефектів.
2. Побудувати Cumulative Flow Diagram (CFD) у Excel/Google Sheets.
3. Обчислити Velocity та Throughput.
4. Сmodelювати проведення ретроспективи в команді, заповнивши таблицю.
5. Скласти Action Plan (мінімум 3 дії на наступний спринт).

Приклад таблиці для ретроспективи:

What went well	What can be improved	Action items
Завершили 90% завдань	Були затримки через неясні вимоги	Ввести backlog refinement
Добра комунікація	Недостатньо часу на тестування	Виділити окремий QA-етап

6. Використайте дані зі спринту (ЛР-5 або ЛР-6) та обчисліть Velocity й Throughput. Зробіть висновок: наскільки команда здатна прогнозувати роботу на наступний спринт.

Приклад таблиці:

Спринт	Завершено user stories	Загальна кількість SP	Velocity	Throughput
Sprint 1	6	20	20	6

7. Побудова Cumulative Flow Diagram (CFD). В Excel/Google Sheets створіть таблицю з кількістю задач у колонках Kanban (To Do, In Progress, Review, Done) на кожен день. Побудуйте діаграму з накопиченими значеннями. Проаналізуйте: чи є стабільний потік завдань; чи зростає черга в In Progress; наскільки рівномірно завдання потрапляють у Done.

Фрагмент таблиці для CFD:

День	To Do	In Progress	Review	Done
1	10	0	0	0
2	8	2	0	0
3	6	3	1	0
4	4	3	1	2

Зміст звіту

1. Титульний аркуш (додаток А).
2. Тема і мета роботи.
3. Короткі теоретичні відомості.
4. Розрахунок метрик, побудова Cumulative Flow Diagram.
5. Проведення ретроспективи (формат Start/Stop/Continue або Mad/Sad/Glad)
6. Таблиці з розрахунками метрик, побудовані графіки
7. Аналіз ефективності команди, виявлені проблеми, запропоновані покращення.
8. Висновки.

ВИКОРИСТАНА ЛІТЕРАТУРА

1. Schwaber, K., Sutherland, J. The Scrum Guide. – Scrum.org, 2020.
2. Anderson, D. J. Kanban: Successful Evolutionary Change for Your Technology Business. – Blue Hole Press, 2010.
3. Cohn, M. Agile Estimating and Planning. – Prentice Hall, 2006.
4. Rubin, K. S. Essential Scrum: A Practical Guide to the Most Popular Agile Process. – Addison-Wesley, 2012.
5. Kniberg, H. Scrum and XP from the Trenches. – InfoQ, 2007.
6. Leach, L. P. Critical Chain Project Management. – Artech House, 2nd ed., 2005.
7. PMI. Agile Practice Guide. – Project Management Institute, 2017.
8. ISO 21502:2020 – Project, programme and portfolio management — Guidance.
9. PMBOK® Guide. – Project Management Institute, 7th Edition, 2021.
10. Derby, E., Larsen, D. Agile Retrospectives: Making Good Teams Great. – Pragmatic Bookshelf, 2006.
11. Knaster, R., Leffingwell, D. SAFe® 5.0 Distilled: Achieving Business Agility with the Scaled Agile Framework®. – Addison-Wesley, 2020.
12. Wysocki, R. K. Effective Project Management: Traditional, Agile, Extreme. – Wiley, 2019.
13. Layton, M., Ostermiller, S. Agile Project Management for Dummies. – Wiley, 2020.
14. Newman, S. Building Microservices. – O'Reilly, 2021.
15. Scrum Guides: <https://scrumguides.org>
16. Agile Alliance Resources: <https://www.agilealliance.org/agile101>
17. Atlassian Agile Coach (Scrum, Kanban, Agile): <https://www.atlassian.com/agile>
18. Scaled Agile Framework (SAFe): <https://scaledagileframework.com>
19. EasyRetro – Story Point Calculator: <https://easyretro.io/tools/story-point-calculator>
20. PlanningPoker.live: <https://planningpoker.live>
21. ScrumPoker Online: <https://www.scrumpoker-online.org>

ДОДАТОК А

Міністерство освіти і науки України
Черкаський державний технологічний університет

Факультет інформаційних технологій і систем

Кафедра комп'ютерних наук та системного аналізу

Дисципліна:

«Гнучкі методології управління ІТ-стартапами та ІТ-проєктами»

З В І Т

з лабораторної роботи № 1

Тема: «Аналіз методологій управління проєктами»

студента групи МСТП-2402

Павлова Віктора Олександровича

(Дата)

(Підпис студента)

Оцінка _____

Перевірено _____

(Дата)

Викладач _____ / проф. Колеснікова К.В. /

(Підпис)

(Посада, прізвище та ініціали)

Черкаси – 2024 р.

ДОДАТОК Б

З В І Т

про виконання завдань до лабораторної роботи № 1

Тема: Аналіз методологій управління проектами

Мета роботи: знайомство з гнучкими методологіями управління проектами

1. Короткі теоретичні відомості.

.....

2. Завдання для виконання.

.....

3. Таблиці з даними.

.....

Висновки:

ДОДАТОК В

Приклади навчальних проєктів з дисципліни «Гнучкі методології управління ІТ-стартапами та ІТ-проєктами»

№	Назва проєкту	Мета	Цільова аудиторія	Основні функції
1	Мобільний додаток для фінансів	Контроль витрат і доходів	Студенти, молодь, сім'ї	Внесення витрат, графіки, поради
2	Онлайн-платформа для бронювання спортмайданчиків	Зручне бронювання	Спортсмени, школи, клуби	Календар бронювань, оплата онлайн
3	Додаток для вивчення мов із чат-ботом	Самоосвіта	Учні, студенти, дорослі	Діалоги з ботом, вправи, тести
4	Система документообігу	Автоматизація офісу	Малі бізнеси, стартапи	Завантаження, пошук, контроль версій
5	Онлайн-магазин з рекомендаціями	Продаж товарів	Широка аудиторія	Каталог, корзина, рекомендації ШІ
6	Фітнес-додаток	Здоровий спосіб життя	Активна молодь, спортсмени	Вправи, трекінг кроків, нагадування
7	Платформа для опитувань	Збір думок	Школи, бізнеси, громади	Створення анкет, аналітика
8	Щоденник харчування	Контроль калорій	Ті, хто стежить за харчуванням	Запис страв, підрахунок калорій
9	Сервіс бронювання столиків	Зручність клієнта	Кафе, ресторани, відвідувачі	Бронювання, карта закладів
10	Carpooling-додаток	Спільні поїздки	Студенти, офісні працівники	Пошук попутників, розрахунок витрат
11	Сервіс пошуку майстрів	Домашні послуги	Домогосподарі, офіси	Каталог послуг, відгуки
12	Додаток для шкільної математики	Допомога у навчанні	Школярі	Завдання, тести, пояснення
13	Система управління гуртожитком	Адміністрування	Студенти, коменданти	Заселення, заявки, оплати
14	Онлайн-бібліотека	Доступ до книг	Студенти, читачі	Каталог, завантаження, рецензії
15	Додаток для спортивних подій	Організація турнірів	Клуби, федерації	Сітка турніру, розклад
16	Еко-платформа	Відстеження ініціатив	Міські громади	Карта подій, повідомлення
17	Платформа для репетиторів	Освітні послуги	Учні, вчителі	Бронювання занять, чат
18	Сімейний бюджет	Контроль витрат	Сім'ї	Спільні фінанси, аналітика
19	Онлайн-запис до лікаря	Медпослуги	Пацієнти, клініки	Календар, нагадування

20	Обмін книгами	Бібліотека користувачів	Молодь, студенти	Каталог, чат, обмін
21	Туристичні маршрути	Планування подорожей	Туристи	Карта, маршрути, рекомендації
22	Аналог Trello	Командна робота	Стартапи, команди	Дошки, таски, дедлайни
23	Платформа для курсів	Онлайн-навчання	Викладачі, студенти	Створення курсів, відео, тести
24	Бронювання квитків у кіно	Дозвілля	Глядачі, кінотеатри	Афіша, вибір місць, оплата
25	Онлайн-щоденник	Організація дня	Студенти, працівники	Нагадування, нотатки
26	Система управління бібліотекою	Автоматизація обліку	Школи, університети	Облік книг, пошук
27	Додаток для догляду за тваринами	Турбота	Власники тварин	Календар щеплень, поради
28	Платформа пошуку підробітку	Студентські вакансії	Студенти	Профілі, вакансії, чат
29	Додаток-гра з історії	Освіта + гейміфікація	Учні, студенти	Квести, тести, рівні
30	Платформа для благодійності	Збір коштів	Благодійники, волонтери	Кампанії, звіти, платежі

ДОДАТОК Д

Інструменти Scrum. Покер планування

Planning poker – техніка командної оцінки завдань в відносних одиницях. Вперше описана в 2002 році Джеймсом Гренінга, одним з авторів Agile-маніфесту. Надалі популяризована Майком Коном в книзі «Agile Estimating and Planning».

Мета Scrum poker – не ідеально точно встановити термін виконання завдання, а переконатися в тому, що всі учасники процесу розуміють завдання та алгоритм його виконання однаково правильно.

Переваги техніки:

- Залучає всіх учасників до процесу оцінки.
- Дозволяє сформуванню команди однакове розуміння суті завдання, яке оцінюється. В результаті оцінки завдання за допомогою покеру планування команда розуміє обсяг змін та їх складність – за рахунок передачі знань і досвіду між учасниками.

- Дозволяє команді за обмежений час прийти до консенсусу у визначенні складності завдання. Згідно з дослідженнями, такі оцінки виходять більш точними і менш оптимістичними.

- Допомагає мінімізувати ефект «прив'язування». Ефект «прив'язування» виникає, коли при оцінці завдань один з учасників вголос вимовляє свою версію оцінки складності завдання, наприклад: «Я думаю, що це завдання на 5 днів». У цей момент у інших учасників несвідомо виникає прив'язка до числа 5. Це число стає відправною точкою в їх особистому розумовому процесі, задаючи рамки. Наприклад, якщо хтось думав про оцінку в 1-2 дня, після почутого думки в 5 днів він несвідомо захоче збільшити свою оцінку. Особливо сильно цей ефект проявляється, коли першим говорить людина, що є формальним або неформальним лідером для команди.

Використовувати метод краще в наступних випадках:

- Для малих та середніх за масштабом проєктів (з відповідними командами, не більше 10 осіб)
- Для проєктів з гнучкою методологією розробки
- Для команд, в яких вже є високий рівень комунікацій та залучення (або для команд, які дуже хочуть цього домогтися і готові докладати зусилля)
- Для команд і завдань, в яких є можливість витратити час на досягнення консенсусу

Як зробити Planning Poker по-справжньому продуктивним:

1. Нехай обговорення залишається продуктивним. Двохвилинний таймер є корисним інструментом. Так ви зможете привчити членів команди виражатися лаконічно. Таймер встановлюється, коли хтось в групі починає раунд. І, як тільки таймер спрацював, право голосу переходить до іншого члена команди.

2. Розділіть обговорення на невеликі сесії. Ідеально, якщо є можливість розбити велику групу на більш дрібні підгрупи. Це добре, коли є багато завдань, які потрібно оцінити; часто такий підхід ефективний на початку

нового проекту. У деяких компаніях команда розкидана по різних містах або навіть країнах. У таких випадках можна використовувати онлайн-конференції для спілкування та спеціальні онлайн-платформи для фіналізації результатів.

3. Менеджери зазвичай хочуть, щоб робота зайняла менше часу, тому вони часто голосують дуже неефективно. Однак у них більше досвіду, ніж у середнього члена команди. Надавши менеджеру право вето на консенсус команди в одному конкретному випадку, можна попросити команду розглянути щось, що може змінити виконання завдання.

4. Коли різниця в оцінках незначна, просто обирайте найбільшу та рухайтесь далі. Внесіть обраний дедлайн в терміни завдання.

5. Люди, які беруть участь в оцінці, повинні мати уявлення про найпростіше, правдоподібне рішення. Вони повинні обирати оцінку на основі цього сценарію.

6. Не допускайте неодноразово виставлення оцінок. Тільки синхронність! Інакше є великий ризик того, що учасники будуть підлаштовуватися один під одного, а це зводить нанівець весь ефект методу. Використовуйте гучний таймер на смартфоні або заведіть офісний будильник.

7. Деякі учасники часто настільки захоплюються сесією покерного планування, що не розуміють, коли інші члени команди потребують перерви. Картка «Мені потрібна перерва» може бути використана кимось, щоб привернути увагу до необхідності перерви.

8. Якщо консенсус не може бути досягнутий до кінця третього туру голосування, виберіть найвищий бал та продовжуйте. Після двох раундів обговорення подальші спроби призведуть лише до витрат часу.

9. Уникайте ефекту прив'язування. Як правило, відкрите обговорення оцінок дозволяє уникнути впливу цього ефекту. Коментування завдань в дозволяє кожному говорити відкрито.

Суть покеру в азарті, а суть проектного менеджменту – в системі. Покер планування перетворює тоскний процес визначення дедлайну завдання з неповною інформацією в командний пошук вирішення цього завдання. Такий системний підхід вирішує відразу кілька завдань: команда починає мислити в єдиному ключі; ставляться реалістичні терміни завдань; в процесі знаходиться оптимальне рішення задачі, готове до реалізації.

Як грати у Planning Poker:

Кожен з присутніх використовує набір карт, на яких є цифри-бали (або числа Фібоначчі – 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, або альтернатива – 0, 1/2, 1, 2, 3, 5, 8, 13, 20, 40, 100, знак питання «?») (ним позначається невпевненість учасника), іконка чашки або пивного келиха (натяк на те, що треба зробити паузу)). Ще використовується знак нескінченності (∞). При цьому числа можуть служити позначенням story points, днів, годин та ін. Автором спеціалізованої оціночної шкали є Майкл Кон.

Закономірність чисел на картах, незалежно від того, який ряд обраний, зводиться до наступного: 0 – завдання настільки просте або настільки близьке до завершення, що немає сенсу виділяти йому місце в плані, 1-3 – дрібні

завдання, 5- 13 – завдання середньої складності, 20-40 – великі завдання, 100 – глобальне завдання, ∞ – епохальна за масштабністю та важливістю задача (рис.Д.1).

Скрам-майстер (Scrum master) виступає як ведучий, постійно стежить за дотриманням правил, розв'язує суперечності і захищає гравців від потенційно відволікаючих чинників. Сам він не бере участь в грі, а всього лише документує підсумки дискусій.



Рис.Д.1. Карти для гри в Planning poker

Власник продукту (Product owner) пояснює вимоги та бажання клієнта, а також інші призначені для користувача сценарії. Під час зборів він демонструє учасникам проекту всі вимоги до майбутнього продукту. Крім цього, власник бере на себе місію представлення інтересів кінцевих споживачів створюваного продукту.

Якщо власника продукту немає, то вимоги може озвучувати скрам-майстер або менеджер проекту.

Перед початком гри проектна команда повинна визначити числові позначення своїх карт. Це можуть бути години, дні, або відносні величини складності виконання встановлених завдань.

Якщо є необхідність, ведучий може використовувати таймер. Подібна практика дозволяє команді не затягувати обговорення і забезпечує якісний мозковий штурм.

На початку ведучий оголошує задачу (або кілька одразу). Учасники гри обговорюють її зміст та за необхідності задають уточнюючі питання власнику продукту, який відповідає за ПЗ.

Після цього кожен учасник таємно обирає собі карту, яка відповідає його оцінці та кладе карту сорочкою нагору.

Як тільки всі учасники гри визначилися зі своїми оцінками – гравці показують карти, які демонструють їх поточні оцінки.

Після обмірковування всі учасники обирають карту, яка відповідає їх оцінці завдання та кладуть її на стіл «сорочкою» вгору. Карти відкриваються, оцінки стають надбанням команди (рис.Д.2)

Якщо є категорична розбіжність у думках, це обговорюється і аргументується. Тобто: той учасник, який дав рекордно низьку оцінку часу виконання завдання, пояснює, як він планує домогтися успіху у настільки стислі терміни. Той, хто оцінив час виконання максимально, також повідомляє, в чому, на його думку, полягає складність завдання. Після обговорення проводиться ще один раунд оцінювання, за підсумками якого приймається рішення.



Рис.Д.2. Процес гри у Planning poker

Виглядати процес буде приблизно таким чином:

- Отже, скільки часу займе ця задача?
- Мовчазний розумовий процес розпочато ...
- Карти відкриті. Але чому ж у учасників А і Д показали такі діаметральні відмінності тривалості у порівнянні з учасником С? Це підлягає обговоренню.
- І знову оцінка з урахуванням обговорення. Консенсус досягнутий.

Перелік деяких онлайн-ресурсів для проведення гри Planning poker :

Назва ресурсу	Посилання	Опис / Особливості
PlanningPoker.live	https://planningpoker.live/ (Planning Poker)	Безкоштовний/легкий Planning Poker; реальне голосування; інтеграція з Jira / Linear; анонімне голосування. (Planning Poker)
StoryPoint.Poker	https://storypoint.poker/ (storypoint.poker)	Проста веб-версія Poker; налаштовувані карти; можна створити кімнату без облікового запису; мобільна та веб-версія. (storypoint.poker)
Scrum Poker Online	https://www.scrumpoker-online.org/en/ (Scrum Poker Online)	Дуже простий інструмент для онлайн-Planning Poker; швидко створити сесію, спільна оцінка. (Scrum Poker Online)

PokerPlanning.org	https://pokerplanning.org/ (PokerPlanning.org)	Open source інструмент для команд; реальний час оцінювання; легкий інтерфейс. (PokerPlanning.org)
Parabol – Sprint Poker	https://www.parabol.co/agile/sprint-poker/ (Parabol)	Інтеграції з GitHub/GitLab/Jira; підтримка різних масштабів оцінок (Fibonacci, інші); безкоштовний план. (Parabol)
Kollabe	https://kollabe.com/posts/best-free-online-planning-poker-tool (Kollabe)	Безкоштовний план; підтримка різних шкал (Fibonacci, T-shirt etc); аналітика історії оцінок; зручний для команд. (Kollabe)
Thunderdome (open-source)	https://thunderdome.dev/ (thunderdome.dev)	Відкрите програмне забезпечення; можливість хостингу самостійно; контроль над даними. (thunderdome.dev)

ЗМІСТ

ВСТУП	3
ЛАБОРАТОРНА РОБОТА №1	4
Порядок виконання роботи	4
Теоретичні відомості.....	4
Завдання на лабораторну роботу.....	7
Зміст звіту.....	8
ЛАБОРАТОРНА РОБОТА №2	9
Порядок виконання роботи	9
Теоретичні відомості.....	9
Завдання на лабораторну роботу.....	12
Зміст звіту.....	12
ЛАБОРАТОРНА РОБОТА №3	13
Порядок виконання роботи	13
Теоретичні відомості.....	13
Завдання на лабораторну роботу.....	18
Зміст звіту.....	18
ЛАБОРАТОРНА РОБОТА №4	19
Порядок виконання роботи	19
Теоретичні відомості.....	19
Завдання на лабораторну роботу.....	34
Зміст звіту.....	34
ЛАБОРАТОРНА РОБОТА №5	35
Порядок виконання роботи	35
Теоретичні відомості.....	35
Завдання на лабораторну роботу.....	37
Зміст звіту.....	38
ЛАБОРАТОРНА РОБОТА №6	39
Порядок виконання роботи	39
Теоретичні відомості.....	39

Завдання на лабораторну роботу.....	42
Зміст звіту.....	43
ЛАБОРАТОРНА РОБОТА №7.....	44
Порядок виконання роботи	44
Теоретичні відомості.....	44
Завдання на лабораторну роботу.....	45
Зміст звіту.....	46
ВИКОРИСТАНА ЛІТЕРАТУРА.....	47
ДОДАТОК А.....	48
ДОДАТОК Б.....	49
ДОДАТОК В.....	50
ДОДАТОК Д.....	52