

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний
університет імені Івана Пулюя

Кафедра програмної інженерії



Конспект лекцій

з дисципліни

«Управління ІТ-проектами»

для здобувачів другого (магістерського)
рівня вищої освіти ОПП «Інженерія
програмного забезпечення»
всіх форм навчання

Тернопіль — 2025

УДК 004.41(076.5)

Укладачі:

Стадник М.А. к.т.н., доцент

Цебрій О.Р., к.ф.-м.н., доцент каф. ПІ

Стефанишин В.М., ас. каф. ПІ

Рецензент:

І.О. Боднарчук, к.т.н., доцент

завідувач кафедри комп'ютерних наук ТНТУ ім. І. Пулюя

Розглянуто й затверджено на засіданні кафедри програмної інженерії. Протокол № 1 від 29.08.2025.

Розглянуто й затверджено на засіданні методичної комісії факультету комп'ютерно-інформаційних систем та програмної інженерії Тернопільського національного технічного університету імені Івана Пулюя.

Протокол № 1 від 1.09.2025.

Конспект лекцій з дисципліни «Управління ІТ-проєктами» для здобувачів другого (магістерського) рівня вищої освіти ОПІ «Інженерія програмного забезпечення» всіх форм навчання / Укладач: Стадник М.А., Цебрій О.Р., Стефанишин В.М.– Тернопіль: ТНТУ ім. І. Пулюя, 2025 – 217 с.

Відповідальний за випуск: ас. каф. ПІ, Стефанишин В.М.

© Стадник М.А., Цебрій О.Р., Стефанишин В.М., 2025

© Тернопільський національний технічний університет імені Івана Пулюя, 2025

АНОТАЦІЯ

Конспект лекцій з дисципліни «Управління ІТ-проєктами» призначений для здобувачів другого (магістерського) рівня вищої освіти спеціальності 121/F2 «Інженерія програмного забезпечення».

У виданні розглянуто життєвий цикл ІТ-проєкту, процеси ініціації, планування, реалізації, моніторингу та завершення. Викладено методи оцінювання трудомісткості й термінів розроблення, управління ресурсами, ризиками, змінами, якістю та пріоритетами. Висвітлено формування проєктної команди, мотивацію учасників, організацію комунікацій і взаємодію із зацікавленими сторонами. Окрему увагу приділено принципам кількісного управління, цінностям і принципам Agile, застосуванню Scrum і Kanban, вибору традиційного, гнучкого або гібридного підходу.

Матеріали конспекту призначені для опрацювання лекційних тем, підготовки до лабораторних робіт, самостійного навчання та підсумкового контролю. Зміст видання спрямований на формування здатності планувати й контролювати виконання ІТ-проєктів, організовувати командну роботу та обґрунтовувати управлінські рішення в умовах невизначеності й ресурсних обмежень.

ЗМІСТ

ВСТУП.....	8
ТЕМАТИЧНИЙ ПЛАН НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.....	12
Лекція 1 Що таке управління проєктами.....	16
1.1 Поняття, ознаки та основні характеристики проєкту.....	16
1.2 Проєкт і операційна діяльність.....	17
1.3 Обмеження проєкту.....	18
1.4 Проєкт і організаційна структура компанії.....	23
1.5 Організація проєктної команди.....	25
1.6 Життєвий цикл проєкту, його фази та основні результати.....	27
Висновки.....	30
Лекція 2 Ініціація проєкту.....	32
2.1 Сутність, мета та завдання ініціації проєкту.....	32
2.2 Управління пріоритетами проєктів.....	34
2.3 Формування концепції та обґрунтування корисності проєкту.....	36
2.4 Визначення цілей, завдань і очікуваних результатів.....	39
2.5 Припущення, обмеження та критерії приймання проєкту.....	41
2.6 Ключові учасники та зацікавлені сторони.....	42
2.7 Попереднє оцінювання ресурсів, термінів і ризиків.....	44
Висновки.....	47
Лекція 3 Планування проєкту.....	49
3.1 Мета, завдання та результати планування проєкту.....	49
3.2 Уточнення змісту і складу проєктних робіт.....	52
3.3 Планування управління змістом проєкту.....	56
3.4 Планування організаційної структури та розподілу відповідальності.	58
3.5 Планування управління конфігураціями та якістю.....	60
3.6 Формування базового розкладу проєкту.....	63
Лекція 4 Управління ризиками проєкту.....	69
4.1 Основні поняття та принципи управління проєктними ризиками....	69
4.2 Планування управління ризиками.....	71
4.3 Ідентифікація та класифікація ризиків.....	72
4.4 Якісний і кількісний аналіз ризиків.....	74
4.5 Планування заходів реагування на ризики.....	77
4.6 Головні ризики програмних проєктів і способи реагування на них.	78

4.7 Управління проєктом, спрямоване на зниження ризиків.....	79
4.8 Моніторинг і контроль ризиків.....	80
Висновки.....	81
Лекція 5 Оцінювання трудомісткості й термінів розроблення.....	82
5.1. Особливості оцінювання трудомісткості програмних проєктів.....	82
5.2. Оцінка як імовірнісне твердження.....	84
5.3. Негативні наслідки «агресивного» розкладу.....	85
5.4. Прагматичний підхід до оцінювання.....	86
5.5. Метод функціональних точок.....	88
5.6. Оцінювання багатокомпонентного програмного продукту.....	90
5.7. Оцінювання тривалості проєкту.....	92
Висновки.....	94
Лекція 6 Формування проєктної команди.....	96
6.1. Проєктна команда та її роль у реалізації проєкту.....	96
6.2. Основні ролі та відповідальність учасників команди.....	98
6.3. Лідерство й управління проєктною командою.....	100
6.4. Добір учасників і формування необхідного набору компетентностей.....	102
6.5. Мотивація учасників проєкту.....	104
6.6. Організація ефективної командної взаємодії.....	105
Лекція 7 Реалізація проєкту.....	109
7.1. Мета, завдання та основні процеси реалізації проєкту.....	109
7.2. Організація робочого планування.....	111
7.3. Координація команди та виконання проєктних робіт.....	113
7.4. Моніторинг прогресу й управління відхиленнями.....	114
7.5. Принципи кількісного управління проєктом.....	116
7.6 Контроль результатів і завершення проєкту.....	119
Висновки.....	122
Лекція 8 Управління пріоритетами проєктів.....	124
8.1. Поняття та значення пріоритету проєкту.....	124
8.2. Чинники, що впливають на визначення пріоритетів.....	126
8.3. Критерії оцінювання та порівняння ІТ-проєктів.....	128
8.4. Основні методи визначення пріоритетів.....	129
8.5. Багатокритеріальне оцінювання та ранжування проєктів.....	132
8.6. Практичний приклад визначення пріоритетів.....	134
8.7. Перегляд пріоритетів у разі зміни умов реалізації.....	135

Висновки.....	136
Лекція 9 Проект і організаційна структура компанії.....	138
9.1. Поняття проекту та організаційної структури компанії.....	138
9.2. Роль організаційної структури в управлінні проектами.....	139
9.3. Функціональна організаційна структура.....	141
9.4. Матрична організаційна структура.....	143
9.5. Проектна організаційна структура.....	145
9.6. Вплив організаційної структури на повноваження керівника та роботи команди.....	147
9.7. Практичні приклади застосування різних організаційних структур...	149
Висновки.....	150
Лекція 10 Мотивація та ефективна взаємодія.....	151
10.1. Поняття мотивації та ефективної взаємодії.....	151
10.2. Внутрішня і зовнішня мотивація учасників проекту.....	152
10.3. Методи мотивації проектної команди.....	154
10.4. Вплив мотивації на продуктивність і результати проекту.....	157
10.5. Особливості взаємодії учасників проекту.....	158
10.6. Комунікація як основа командної роботи.....	160
10.7. Взаємодія із зацікавленими сторонами та роботодавцями.....	163
10.8. Зворотний зв'язок і попередження конфліктів.....	166
Висновки.....	167
Лекція 11 Головні ризики програмних проєктів та способи реагування. 169	
11.1. Особливості ризиків програмних проєктів.....	169
11.2. Ризики вимог і неконтрольованого розширення змісту.....	170
11.3. Ризики оцінювання, термінів і ресурсного забезпечення.....	172
11.4. Командні, організаційні та комунікаційні ризики.....	174
11.5. Технічні ризики, ризики якості та інформаційної безпеки.....	175
11.6. Способи запобігання та реагування на ризики.....	177
11.7. Практичні приклади успішного управління ризиками.....	178
Висновки.....	179
Лекція 12 Принципи кількісного управління.....	181
12.1. Поняття та призначення кількісного управління.....	181
12.2. Основні принципи кількісного управління проєктом.....	182
12.3. Вимоги до проєктних показників і вихідних даних.....	183
12.4. Вимірювання прогресу та продуктивності команди.....	185

12.5. Кількісне оцінювання ризиків.....	186
12.6. Метрики термінів, ресурсів і якості.....	187
12.7. Використання метрик для моніторингу та контролю проєкту.....	188
12.8. Інтерпретація показників і прийняття управлінських рішень.....	189
Висновки.....	190
Лекція 13 Ключові учасники і зацікавлені сторони.....	191
13.1. Поняття ключових учасників і зацікавлених сторін.....	191
13.2. Основні ролі та відповідальність учасників проєкту.....	192
13.3. Внутрішні та зовнішні стейкхолдери.....	193
13.4. Ідентифікація та аналіз зацікавлених сторін.....	194
13.5. Оцінювання впливу, інтересів та очікувань стейкхолдерів.....	196
13.6. Планування залучення та комунікацій.....	197
13.7. Методи взаємодії із зацікавленими сторонами.....	199
13.8. Практичні приклади управління стейкхолдерами.....	200
Висновки.....	201
Лекція 14 Agile-методології розроблення програмного забезпечення 203	
14.1. Передумови виникнення та сутність Agile.....	203
14.2. Цінності й основні принципи Agile.....	204
14.3. Ітеративне та інкрементальне розроблення програмного забезпечення.....	206
14.4. Scrum: ролі, події та артефакти.....	207
14.5. Kanban: візуалізація потоку робіт та обмеження незавершеної роботи.....	209
14.6. Інші гнучкі підходи до розроблення програмного забезпечення.....	211
14.7. Переваги та недоліки Agile порівняно з традиційними підходами.....	212
14.8. Вибір гнучкого, традиційного або гібридного підходу.....	213
Висновки.....	215
Список використаної літератури.....	217

ВСТУП

Розроблення програмного забезпечення потребує узгодження технічних рішень із цілями замовника, можливостями команди, обмеженнями ресурсів і термінів. Зміна вимог, складність оцінювання трудомісткості, залежності між компонентами та учасниками створюють невизначеність, яку необхідно враховувати протягом усього життєвого циклу проєкту. Тому підготовка інженера програмного забезпечення передбачає опанування методів планування, організації командної роботи, управління ризиками та контролю результатів.

Вибіркова дисципліна «Управління ІТ-проєктами» є складовою підготовки здобувачів другого (магістерського) рівня вищої освіти спеціальності F2 «Інженерія програмного забезпечення». Мета дисципліни полягає у формуванні знань і практичних навичок планування, моніторингу та управління проєктами від локального до корпоративного рівня. У межах курсу розглядаються принципи управління розробленням інформаційних систем, організація команди та взаємодія її учасників.

Управління ІТ-проєктом охоплює визначення цілей і очікуваних результатів, формування змісту робіт, оцінювання потреби в ресурсах, розподіл відповідальності, координацію виконання та перевірку досягнутих результатів. Управлінські рішення мають враховувати взаємозалежність змісту, термінів, ресурсів і якості. Зміна одного з цих параметрів може потребувати перегляду плану, пріоритетів або способу виконання робіт.

Матеріал конспекту викладено послідовно: від основних понять, критеріїв успішності та життєвого циклу проєкту до його ініціації, планування, оцінювання трудомісткості, формування команди й реалізації. Наступні лекції поглиблюють питання управління пріоритетами, організаційних структур, мотивації, ризиків, кількісного контролю та взаємодії із зацікавленими сторонами. Завершальна лекція присвячена гнучким підходам до розроблення програмного забезпечення.

Особливу увагу приділено обґрунтованості планування. Оцінки трудомісткості й тривалості розглядаються з урахуванням невизначеності, наявної інформації та характеристик програмного продукту. Аналіз ризиків допомагає визначити можливі причини відхилень, підготувати заходи реагування та своєчасно скоригувати подальші дії. Використання метрик

забезпечує перевірку стану проєкту за фактичними результатами виконання робіт.

Важливою складовою курсу є організація роботи з людьми. Формування команди, розподіл ролей, мотивація, комунікація та залучення зацікавлених сторін впливають на узгодженість рішень і досягнення цілей проєкту. Розгляд організаційних структур дає змогу зрозуміти, як повноваження керівника, підпорядкування учасників і доступність ресурсів визначають умови проєктної діяльності.

Вивчення Agile, Scrum і Kanban доповнює розгляд процесів проєктного управління. Студенти аналізують принципи адаптивного планування, організації роботи короткими ітераціями, управління потоком завдань і регулярного отримання зворотного зв'язку. Порівняння гнучких і традиційних підходів спрямоване на формування здатності вибирати спосіб управління відповідно до вимог, обмежень і рівня невизначеності проєкту.

Основні завдання курсу:

- сформуванати розуміння сутності ІТ-проєкту, його життєвого циклу та критеріїв успішності;
- навчити визначати цілі, результати, припущення, обмеження та критерії приймання проєкту;
- опанувати методи планування змісту, складу й послідовності робіт, ресурсів і термінів;
- засвоїти підходи до оцінювання трудомісткості та тривалості розроблення програмного забезпечення;
- сформуванати навички ідентифікації, аналізу, моніторингу ризиків і планування заходів реагування;
- розглянути принципи формування команди, розподілу відповідальності, лідерства та мотивації;
- навчити визначати пріоритети й враховувати вплив організаційної структури на виконання проєкту;
- опанувати методи взаємодії з ключовими учасниками та зацікавленими сторонами;
- сформуванати розуміння кількісного управління та використання метрик для оцінювання прогресу;
- засвоїти підходи до управління змінами, контролю якості та завершення проєкту;

- ознайомитися з принципами Agile, застосуванням Scrum і Kanban та умовами вибору підходу до управління розробленням.

Практична спрямованість дисципліни забезпечується лабораторними роботами з підготовки проєкту та визначення ролей, аналізу ризиків, оцінювання спроможності команди й управління ресурсами. Окремі роботи присвячено управлінню стейкхолдерами та змінами, оцінюванню прогресу, контролю якості й завершенню проєкту. Теоретичний матеріал конспекту слугує основою для виконання цих завдань і обґрунтування прийнятих рішень.

Конспект охоплює 14 лекцій відповідно до тематичного плану робочої програми дисципліни.

Лекція 1. Що таке управління проєктами. Поняття управління проєктами, проєкт як основа інновацій, критерії успішності. Зв'язок проєкту з організаційною структурою компанії, організація проєктної команди, життєвий цикл, фази та продукти проєкту.

Лекція 2. Ініціація проєкту. Формування концепції, визначення цілей і результатів, припущень та обмежень. Встановлення пріоритетів, визначення ключових учасників і зацікавлених сторін, ресурсів, термінів, ризиків та критеріїв приймання. Обґрунтування корисності проєкту.

Лекція 3. Планування проєкту. Уточнення змісту та складу робіт, планування управління змістом і організаційної структури. Планування управління конфігураціями та якістю, формування базового розкладу проєкту.

Лекція 4. Управління ризиками проєкту. Основні поняття та планування управління ризиками. Ідентифікація, якісний і кількісний аналіз, планування реагування, моніторинг та контроль ризиків. Організація управління проєктом, спрямованого на зниження ризиків.

Лекція 5. Оцінка трудомісткості і термінів розробки. Імовірнісний характер оцінювання, негативні наслідки «агресивного» розкладу та прагматичний підхід до визначення трудомісткості. Метод функціональних точок, оцінювання багатокомпонентного продукту і тривалості проєкту.

Лекція 6. Формування команди. Принципи формування проєктної команди, лідерство та управління. Добір учасників відповідно до потреб проєкту й мотивація командної роботи.

Лекція 7. Реалізація проєкту. Організація виконання проєктних робіт, робоче планування, застосування принципів кількісного управління та завершення проєкту.

Лекція 8. Управління пріоритетами проєктів. Поняття пріоритету проєкту, чинники його визначення та методи встановлення пріоритетів. Приклади практичного застосування цих методів.

Лекція 9. Проєкт і організаційна структура компанії. Поняття та типи організаційних структур, їхня роль у проєктній діяльності. Вплив структури компанії на управління проєктами та приклади організації роботи в різних структурах.

Лекція 10. Мотивація. Ефективна взаємодія (за участі зацікавлених сторін/роботодавців). Поняття мотивації та ефективної взаємодії, види мотивації та їхній вплив на проєкт. Особливості взаємодії учасників і роль комунікації.

Лекція 11. Головні ризики програмних проєктів та способи реагування. Найпоширеніші ризики розроблення програмного забезпечення та їхній вплив на виконання проєкту. Способи запобігання ризикам, заходи реагування та приклади успішного управління ними.

Лекція 12. Принципи кількісного управління. Поняття й основні принципи кількісного управління. Вимірювання продуктивності, оцінювання ризиків і використання метрик для контролю проєкту.

Лекція 13. Ключові учасники і зацікавлені сторони. Визначення ключових учасників та зацікавлених сторін, їхніх ролей у проєкті. Методи взаємодії та приклади їх практичного застосування.

Лекція 14. Agile-методології розробки програмного забезпечення. Поняття та основні принципи Agile, підходи Scrum, Kanban та інші способи організації гнучкого розроблення. Переваги й обмеження Agile порівняно з традиційними підходами.

Самостійне опрацювання конспекту передбачає вивчення лекційних тем, аналіз проєктних ситуацій, роботу з рекомендованими джерелами та підготовку до лабораторних занять і контролю знань. Формою підсумкового семестрового контролю є залік.

Опанування дисципліни має сформувати здатність планувати й організовувати виконання ІТ-проєктів, оцінювати альтернативні рішення та ризики, координувати взаємодію учасників і контролювати досягнення запланованих результатів. Отримані знання застосовуються під час розроблення, впровадження та супроводу програмного забезпечення в умовах зміни вимог і ресурсних обмежень.

ТЕМАТИЧНИЙ ПЛАН НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Дисципліна: «Управління IT-проєктами».

Статус дисципліни: вибіркова.

Рівень вищої освіти: другий (магістерський).

Спеціальність: F2 «Інженерія програмного забезпечення».

Форма підсумкового семестрового контролю: залік.

Обсяг дисципліни: 4 кредити ECTS (120 годин). Для денної форми навчання передбачено 42 години аудиторної роботи, з них 28 годин лекційних та 14 годин лабораторних занять, а також 78 годин самостійної роботи. Для заочної (дистанційної) форми навчання передбачено 16 годин аудиторної роботи, з них 10 годин лекційних та 6 годин лабораторних занять, а також 104 години самостійної роботи.

Метою вивчення дисципліни є формування у здобувачів знань і практичних навичок планування, організації, моніторингу та управління IT-проєктами від локального до корпоративного рівня. Дисципліна спрямована на розвиток здатності обґрунтовувати управлінські рішення, координувати роботу команди та забезпечувати досягнення цілей проєкту в умовах невизначеності, зміни вимог і ресурсних обмежень.

У межах дисципліни розглядаються сутність і критерії успішності IT-проєкту, його життєвий цикл, процеси ініціації, планування, реалізації та завершення. Вивчаються методи оцінювання трудомісткості й термінів розроблення, управління ризиками та пріоритетами, принципи формування команди, мотивації та ефективної взаємодії. Окрему увагу приділено впливу організаційної структури компанії на проєктну діяльність, кількісному управлінню, роботі із зацікавленими сторонами та застосуванню гнучких підходів до розроблення програмного забезпечення.

Мета проведення лекційних занять полягає у формуванні теоретичної бази з управління IT-проєктами та розвитку здатності аналізувати взаємозв'язки між цілями, змістом робіт, ресурсами, термінами, ризиками й очікуваними результатами. Під час лекцій студенти вивчають способи визначення концепції та меж проєкту, формування планів, оцінювання трудомісткості, розподілу відповідальності й контролю виконання робіт.

Лекційний курс охоплює 14 тем, послідовність яких відповідає робочій програмі дисципліни. Матеріал передбачає розгляд методів управління ризиками, використання метрик для оцінювання прогресу,

організації комунікацій та залучення зацікавлених сторін. Вивчення принципів Agile, фреймворку Scrum і підходу Kanban дає змогу порівнювати способи організації розроблення та визначати умови їх доцільного застосування.

Мета проведення лабораторних занять полягає у формуванні практичних навичок підготовки, планування, аналізу та контролю ІТ-проектів. Лабораторний практикум передбачає підготовку проекту й визначення ролей його учасників, аналіз ризиків та управління ними, оцінювання спроможності команди й розподіл ресурсів. Окремі роботи присвячено управлінню стейкхолдерами, опрацюванню змін у проекті, оцінюванню прогресу, контролю якості та завершенню проекту.

Основними завданнями дисципліни є:

- сформувати розуміння сутності ІТ-проекту, його життєвого циклу, основних процесів управління та критеріїв успішності;
- навчити визначати концепцію, цілі, очікувані результати, припущення, обмеження та критерії приймання проекту;
- сформувати навички планування змісту, складу й послідовності робіт, організаційної структури та базового розкладу проекту;
- опанувати методи оцінювання трудомісткості й тривалості розроблення програмного забезпечення з урахуванням невизначеності;
- навчити ідентифікувати ризики, здійснювати їх якісний і кількісний аналіз, планувати заходи реагування та контролювати їх виконання;
- сформувати розуміння принципів добору учасників команди, розподілу ролей, лідерства, мотивації та управління ресурсами;
- навчити визначати пріоритети проектів і враховувати вплив організаційної структури компанії на їх реалізацію;
- розвинути навички взаємодії з ключовими учасниками та зацікавленими сторонами, узгодження очікувань і організації комунікацій;
- опанувати принципи кількісного управління та використання метрик для оцінювання продуктивності, ризиків і прогресу;
- сформувати навички аналізу змін, контролю якості результатів і завершення проекту;
- ознайомити з принципами Agile, застосуванням Scrum і Kanban, перевагами та обмеженнями гнучких підходів;

- розвинути здатність порівнювати альтернативні управлінські рішення й обґрунтовувати їх вибір відповідно до умов проєкту.

Самостійна робота спрямована на поглиблення лекційного матеріалу, підготовку до виконання та захисту лабораторних робіт, опрацювання окремих розділів програми й рекомендованої літератури, а також підготовку до поточного, модульного та підсумкового контролю знань.

У результаті опанування дисципліни студент повинен бути здатним планувати й координувати процеси розроблення програмного забезпечення, оцінювати проєктні рішення з урахуванням якості та ресурсних обмежень, приймати організаційно-управлінські рішення в умовах невизначеності, управляти змінами й оцінювати успішність проєкту.

За результатами вивчення дисципліни ” Управління ІТ-проєктами” студент повинен продемонструвати такі результати навчання (*кінцеві, підсумкові та інтегративні результати навчання, що визначають нормативний зміст підготовки*):

1. РН01 Знати і застосовувати сучасні професійні стандарти і інші нормативно-правові документи з інженерії програмного забезпечення.

2. РН02 Оцінювати і вибирати ефективні методи і моделі розроблення, впровадження, супроводу програмного забезпечення та управління відповідними процесами на всіх етапах життєвого циклу.

3. РН05 Розробляти, аналізувати, обґрунтовувати та систематизувати вимоги до програмного забезпечення.

4. РН06 Розробляти і оцінювати стратегії проєктування програмних засобів; обґрунтовувати, аналізувати і оцінювати варіанти проєктних рішень з точки зору якості кінцевого програмного продукту, ресурсних обмежень та інших факторів.

5. РН12 Приймати ефективні організаційно-управлінські рішення в умовах невизначеності та зміни вимог, порівнювати альтернативи, оцінювати ризики.

6. РН13 Конфігурувати програмне забезпечення, керувати його змінами та розробленням програмної документації на всіх етапах життєвого циклу.

7. РН15 Здійснювати реінжиніринг програмного забезпечення відповідно до вимог замовника.

8. РН16 Планувати, організовувати та здійснювати тестування, верифікацію та валідацію програмного.

9. РН18 Володіти критеріями успішності розробки програмного забезпечення в цілях визначення інвестиційної привабливості проєктів.

Вивчення навчальної дисципліни передбачає формування та розвиток у студентів компетентностей:

інтегральна компетентність: Здатність особи розв'язувати складні задачі і проблеми у певній галузі професійної діяльності або у процесі навчання, що передбачає проведення досліджень та/або здійснення інновацій та характеризується невизначеністю умов та вимог.

загальні компетентності:

- 1 ЗК01. Здатність до абстрактного мислення, аналізу та синтезу
- 2 ЗК04. Здатність спілкуватися з представниками інших професійних груп
- 3 ЗК05. Здатність генерувати нові ідеї (креативність)

спеціальні (фахові) компетентності:

- 4 СК01. Здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення
- 5 СК02. Здатність розробляти і реалізовувати наукові та/або прикладні проєкти у сфері інженерії програмного забезпечення.
- 6 СК04. Здатність розвивати і реалізовувати нові конкурентоспроможні ідеї в інженерії програмного забезпечення.
- 7 СК05. Здатність розробляти, аналізувати та застосовувати специфікації, стандарти, правила і рекомендації в сфері інженерії програмного забезпечення.
- 8 СК06. Здатність ефективно керувати ресурсами проєкту
- 9 СК07. Здатність критично осмислювати проблеми у галузі інформаційних технологій та на межі галузей знань, інтегрувати відповідні знання та розв'язувати складні задачі у широких або мультидисциплінарних контекстах.
- 10 СК08. Здатність розробляти і координувати процеси, етапи та ітерації життєвого циклу програмного забезпечення на основі застосування сучасних моделей, методів та технологій розроблення програмного забезпечення.
- 11 СК09. Здатність забезпечувати якість програмного забезпечення

Лекція 1 Що таке управління проєктами

Тема: Що таке управління проєктами

Мета: Сформувати у студентів розуміння сутності проєкту та управління проєктами, їхньої ролі у створенні інновацій, ознайомити з критеріями успішності, особливостями взаємодії проєкту з організаційною структурою компанії, принципами формування проєктної команди та основними фазами життєвого циклу проєкту.

План лекції

1. Поняття, ознаки та основні характеристики проєкту.
2. Управління проєктами та роль проєктів у створенні інновацій.
3. Цілі та критерії успішності проєкту.
4. Проєкт і організаційна структура компанії.
5. Організація проєктної команди.
6. Життєвий цикл проєкту, його фази та основні результати.
7. Висновки.

Організації постійно виконують роботу двох принципово різних типів: операційну та проєктну. Операційна робота підтримує повсякденне функціонування організації та повторюється за відносно стабільними правилами. Проєктна робота, навпаки, спрямована на створення нового або суттєво зміненого продукту, послуги чи результату. Саме через проєкти організації реалізують стратегічні зміни, упроваджують інформаційні системи, модернізують процеси, створюють програмні продукти та перевіряють інноваційні ідеї.

Управління проєктами потрібне не лише для складання календарного плану. Воно забезпечує узгодження очікувань, визначення цілей і меж робіт, розподіл відповідальності, залучення ресурсів, роботу з ризиками та змінами, контроль прогресу і формальне передавання результату замовникові. Тому проєктне управління слід розглядати як цілісну систему створення цінності, а не як набір ізольованих адміністративних процедур.

1.1 Поняття, ознаки та основні характеристики проєкту

Проєкт. це тимчасова діяльність, яку здійснюють для створення унікального продукту, послуги або результату. Тимчасовий характер означає, що проєкт має визначені початок і завершення. Завершення настає, коли цілі досягнуто, коли встановлено, що їх неможливо або недоцільно досягати, або коли потреба в проєкті зникла.

Слово «проект» у побуті застосовують до багатьох видів діяльності — від ремонту кімнати до спорудження автомагістралі. У професійному управлінні поняття є вузьким. Діяльність вважають проектом, якщо вона має конкретну мету, обмежена в часі, орієнтована на визначених замовників або інших стейкхолдерів, виконується в умовах обмежень і невизначеності та має вимірювані критерії успішності.

Унікальність не обов'язково означає абсолютну відсутність аналогів. Організація могла раніше створювати подібні системи, однак новий проект відрізнятиметься замовником, командою, набором вимог, технологіями, даними, строками або умовами експлуатації. Наприклад, перенесення університетського сайту на нову платформу є проектом, навіть якщо команда вже виконувала аналогічні міграції: конкретна конфігурація системи, склад даних і вимоги цього замовника залишаються унікальними.

Основні ознаки проекту узагальнено на рисунку 1.1.



Рисунок 1 – Основні характеристики проекту

1.2 Проект і операційна діяльність

Проекти відрізняються від операцій передусім тимчасовістю та унікальністю. Операційна діяльність є безперервною і повторюваною: вона не має заздалегідь визначеної дати завершення, а однакові або подібні процеси багаторазово виконуються для отримання стабільного результату. Її призначення — підтримувати функціонування організації. Призначення проекту — досягти конкретної мети й додати цінність шляхом створення нового або зміненого результату.

Розроблення нової інформаційної системи є проєктом; її щоденне адміністрування після введення в експлуатацію — операційною діяльністю. Випуск першої версії мобільного застосунку є проєктом; регулярне опрацювання звернень користувачів — операцією. Водночас велике оновлення застосунку, що має окрему мету, бюджет, команду та строк, знову організовують як проєкт.

Відмінності між двома типами діяльності систематизовано в таблиці 1.1.

Таблиця 1 – Порівняння проєкту й операційної діяльності

Ознака	Проект	Операційна діяльність
Тривалість	Тимчасовий: має визначені початок і завершення	Безперервна або регулярно повторювана
Результат	Унікальний продукт, послуга або результат	Стабільний, повторюваний результат
Мета	Досягнення конкретної зміни або створення нової цінності	Підтримання поточної діяльності організації
Команда	Формується на період проєкту; може бути міжфункціональною	Зазвичай закріплена в постійних підрозділах
Невизначеність	Зазвичай вища через унікальність і зміни	Нижча завдяки повторюваним процесам
Завершення	Після досягнення цілей, скасування або втрати потреби	Не має визначеного завершення, доки існує функція

1.3 Обмеження проєкту

Кожен проєкт реалізується в системі взаємопов'язаних обмежень. Традиційно виділяють зміст, час і вартість — так зване потрійне обмеження. Зміна одного елемента зазвичай впливає на інші. Якщо замовник збільшує зміст робіт, команді може знадобитися більше часу й коштів. Якщо кінцевий строк скорочують, необхідно залучити додаткові ресурси, зменшити зміст або прийняти підвищений ризик.

Сучасне розуміння обмежень є ширшим: окрім змісту, строків і бюджету, керівник має враховувати якість, ризики та доступність ресурсів. Баланс не означає механічної рівності. Він означає свідоме узгодження пріоритетів із замовником і стейкхолдерами, щоб результат залишався

корисним і прийнятним. Взаємозалежність обмежень показано на рисунку 1.2.

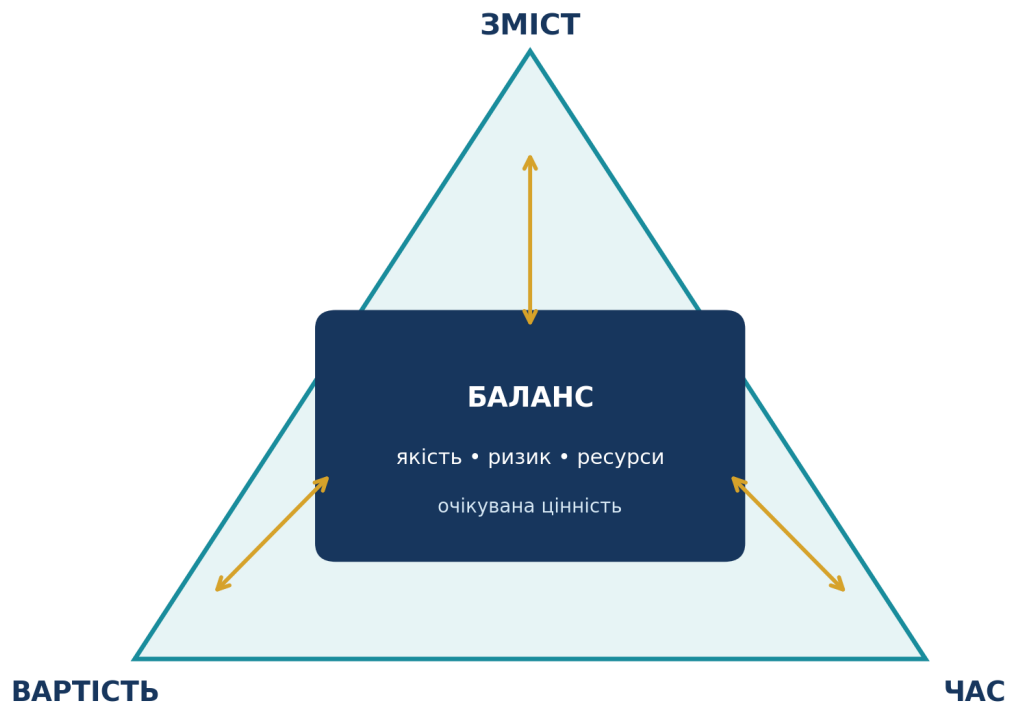


Рисунок 2 – Система взаємопов’язаних обмежень проєкту

Зміст проєкту. сукупність результатів і робіт, потрібних для їх створення. Він визначає, що саме має бути зроблено і що не входить до проєкту.

Час. тривалість виконання проєкту та календарні межі його робіт. Керування часом передбачає визначення завдань, їх послідовності, тривалості й ресурсного забезпечення.

Вартість. затверджений бюджет, який охоплює витрати, необхідні для створення результатів проєкту.

Якість. сукупність стандартів і критеріїв, яким повинен відповідати результат, щоб забезпечити очікувану функціональність, надійність, доступність і придатність до використання.

Ризик. можлива внутрішня або зовнішня подія, настання якої може вплинути на проєкт. Ризики оцінюють за ймовірністю та наслідками і завчасно планують реагування.

Ресурси. люди, обладнання, приміщення, фінансування, інформація та інші засоби, необхідні для виконання проєктних робіт.

1.2. Управління проєктами та роль проєктів у створенні інновацій

Управління проєктами. це застосування знань, навичок, інструментів і методів до проєктної діяльності з метою виконання вимог проєкту. Воно охоплює планування, виконання плану, вимірювання прогресу й результативності, управління змінами та завершення проєкту.

Підхід «просто почати робити» часто призводить до того, що команда створює продукт, який не відповідає очікуванням замовника. Формалізоване управління не усуває всі проблеми, ризики або несподіванки, але забезпечує спільні правила й процедури для роботи з ними. Команда заздалегідь уточнює вимоги, погоджує цілі, документує рішення, визначає відповідальних, планує ресурси та встановлює порядок реагування на відхилення.

Управління проєктом починається з розуміння того, що потрібно отримати і для кого створюється результат. Якщо учасники однаково розуміють мету та погодили основні очікування, зменшується ймовірність конфліктів і повторного виконання робіт. Тому технічні рішення в IT-проєкті мають бути пов'язані з бізнес-потребою, користувацькою проблемою та очікуваними вигодами.

Організації створюють цінність через нові або вдосконалені продукти, процеси та послуги. Проєкти є одним із головних механізмів такого створення цінності, а управління проєктами — цілеспрямованим способом її отримання. Для комерційної організації цінність може проявлятися у доході, зниженні витрат, швидшому виході на ринок або підвищенні лояльності клієнтів. Для університету чи державної установи — у доступності послуг, якості освіти, прозорості процесів або економії часу громадян.

Інновація не обмежується винайденням принципово нової технології. Вона може полягати у новому способі використання відомої технології, зміні бізнес-процесу, новій моделі взаємодії з користувачем або інтеграції раніше відокремлених систем. Щоб інноваційна ідея стала практичним результатом, її необхідно перетворити на керовану ініціативу: визначити проблему, очікувану цінність, результати, ресурси, строк, ризики та відповідальних.

Наприклад, ідея використати штучний інтелект для автоматичного опрацювання звернень студентів стає проєктом після визначення цільових груп, функціональних вимог, джерел даних, критеріїв точності, правил

захисту інформації, складу команди, бюджету та строків. Результатом проєкту може бути пілотна система, інтегрована із сайтом університету. Подальша щоденна підтримка цієї системи вже належатиме до операційної діяльності.

Проектне управління допомагає інноваціям не лише завдяки контролю. Воно створює простір для перевірки припущень, залучення стейкхолдерів, швидкого отримання зворотного зв'язку та перегляду рішень. Водночас організація повинна зберігати зв'язок проєкту зі стратегією: інноваційний результат має не просто існувати, а розв'язувати важливу проблему й забезпечувати вимірювану користь.

1.3. Цілі та критерії успішності проєкту

Проект виникає через потребу, проблему або можливість. На початку важливо пояснити не лише те, що команда планує створити, а й те, навіщо це потрібно. Ціль формулює бажану зміну, тоді як результати описують конкретні продукти, послуги або інші об'єкти, які команда має передати. Вигоди з'являються тоді, коли передані результати використовуються в діяльності організації та створюють очікувану цінність.

Наприклад, створення електронної системи запису до університетської лабораторії є результатом. Скорочення часу очікування, зменшення кількості помилок у розкладі та підвищення завантаження обладнання є очікуваними вигодами. Якщо система технічно розроблена вчасно, але не використовується через незручний інтерфейс або відсутність навчання персоналу, управління роботами могло бути успішним, однак сам проєкт не створив очікуваної цінності.

Успіх проєкту та успіх управління проєктом пов'язані, але не тотожні. Успіх управління стосується процесу: чи завершено проєкт, чи передано погоджений зміст, чи дотримано бюджету, строків і вимог до якості. Успіх проєкту стосується впливу кінцевого результату на замовника, користувачів та інших стейкхолдерів. Окремо розглядають організаційний успіх — внесок проєкту у стратегічні цілі, розвиток спроможностей та реалізацію запланованих вигод.

Співвідношення трьох рівнів успіху подано на рисунку 1.3, а критерії їх оцінювання — у таблиці 1.2.



Рисунок 1.3 – Три взаємопов’язані рівні успішності

Таблиця 1.2 – Критерії оцінювання успішності проекту

Рівень	Основне запитання	Приклади критеріїв
Успіх управління проектом	Наскільки якісно організовано й виконано проектну роботу?	Дотримання погодженого змісту, строків, бюджету та вимог до якості; контроль змін і ризиків
Успіх проекту	Чи задовольняє результат потреби замовника й користувачів?	Приймання результату; придатність до використання; задоволеність стейкхолдерів; досягнення очікуваного ефекту
Організаційний успіх	Чи підтримує проект стратегію та створює довгострокову цінність?	Реалізація вигод; внесок у стратегічні показники; розвиток компетентностей; інвестиційна доцільність

Проект може бути виконаний у межах бюджету й строку, але сприйматися як невдалий, якщо користувачі не отримали очікуваної користі. Можлива й протилежна ситуація: результат створив значну цінність, хоча строк або бюджет було перевищено. Тому критерії успіху потрібно визначати на початку, погоджувати зі стейкхолдерами та періодично переглядати протягом життєвого циклу.

До чинників, які підвищують імовірність успіху, належать зрозуміле обґрунтування, чіткі цілі, компетентний керівник, підтримка вищого керівництва, належне планування, якісні комунікації та реальна прихильність учасників до проекту. Відсутність цих чинників може звести нанівець навіть сильне технічне рішення.

1.4 Проєкт і організаційна структура компанії

Організаційна структура визначає, як поділено завдання, розподілено ресурси, скоординовано підрозділи та побудовано формальні відносини підпорядкування. Вона безпосередньо впливає на те, де перебуває керівник проєкту, кому підпорядковуються члени команди, хто контролює бюджет і наскільки швидко можна отримати потрібні ресурси.

Повноваження. право ухвалювати рішення, віддавати розпорядження та розподіляти ресурси для досягнення бажаних результатів.

Відповідальність. обов'язок належним чином виконувати призначену роботу в межах формальної ролі.

Підзвітність. обов'язок відповідати перед стейкхолдерами за рішення й дії та надавати видимі докази виконаної роботи.

У загальному вигляді розрізняють функціональну, матричну та проєктну структури. Матрична структура має слабкий, збалансований і сильний різновиди. Ці форми утворюють континуум: від мінімальних повноважень керівника проєкту у функціональній організації до високих або повних повноважень у проєктній. Цю залежність показано на рисунку 1.4.



Рисунок 1.4 – Повноваження керівника проєкту в різних організаційних структурах

У функціональній організації працівників групують за спеціалізацією: розроблення, тестування, інформаційна безпека, фінанси, маркетинг тощо. Така структура ефективна для постійної діяльності й професійного розвитку, однак міжпідроздільні проєкти можуть

уповільнюватися через функціональні «сховища», різні пріоритети та необхідність погоджувати ресурси через керівників підрозділів.

Керівник проєкту у функціональній структурі часто має обмежені повноваження і виконує роль координатора. Успіх залежить від співпраці з функціональними керівниками, оскільки саме вони визначають доступність працівників і можуть змінювати їхні пріоритети.

Проектна організація зосереджує ресурси на виконанні проєктів. Працівників призначають до конкретного проєкту, вони звітують керівникові проєкту, а той має значні повноваження щодо ресурсів, персоналу й бюджету. Такий підхід зменшує кількість бюрократичних рівнів і дає команді змогу зосередитися на спільному результаті.

Водночас після завершення проєкту членам команди може бути складно знайти наступне призначення. Крім того, лояльність працівника інколи зміщується від професійної спеціалізації до конкретного проєкту. Проектна структура часто використовується в будівництві, інженерії, дослідницьких організаціях і компаніях, що розробляють програмне забезпечення на замовлення.

Матрична структура намагається поєднати переваги функціональної та проектної організації. Працівник зберігає зв'язок зі своїм функціональним підрозділом, але одночасно працює в проєкті. Керівник проєкту й функціональний керівник спільно розподіляють повноваження та відповідальність.

Перевагою є руйнування комунікаційних бар'єрів між підрозділами та можливість використовувати фахівців у кількох проєктах. Недоліком є подвійне підпорядкування: працівник може отримувати суперечливі пріоритети від двох керівників. У слабкій матриці переважає функціональний керівник, у збалансованій повноваження розподілено приблизно порівну, а в сильній провідну роль відіграє керівник проєкту.

Порівняльні характеристики структур наведено в таблиці 1.3.

Таблиця 1.3 – Роль керівника проєкту в різних організаційних структурах

Характеристика	Функціональна	Слабка матриця	Збалансована матриця	Сильна матриця	Проектна
Повноваження керівника	Мінімальні або відсутні	Низькі	Низькі або помірні	Помірні або високі	Високі або повні

Характеристика	Функціональна	Слабка матриця	Збалансована матриця	Сильна матриця	Проектна
Роль керівника	Часткова зайнятість	Часткова зайнятість	Повна зайнятість	Повна зайнятість	Повна зайнятість
Контроль бюджету	Функціональний керівник	Функціональний керівник	Спільний	Керівник проекту	Керівник проекту
Доступність ресурсів	Мала або відсутня	Низька	Низька або помірною	Помірною або високою	Високою

1.5 Організація проектної команди

Навіть якщо проєкт виконується в межах постійної компанії, для нього створюється тимчасова організація з власними ролями, каналами комунікації та правилами ухвалення рішень. У PM² зазначено, що існує одна проектна команда, до якої входять люди, що виконують ролі рівнів спрямування, управління та виконання. Для успіху вони повинні діяти як єдина команда, а не як представники відокремлених сторін.

Модель PM² розділяє організацію проєкту на рівні: керівний рівень організації, керівний рівень проєкту, рівень спрямування, управління та виконання. Вона також підтримує баланс між стороною замовника, яка визначає потребу й приймає результат, і стороною постачальника, яка створює рішення. Узагальнену модель подано на рисунку 1.5.

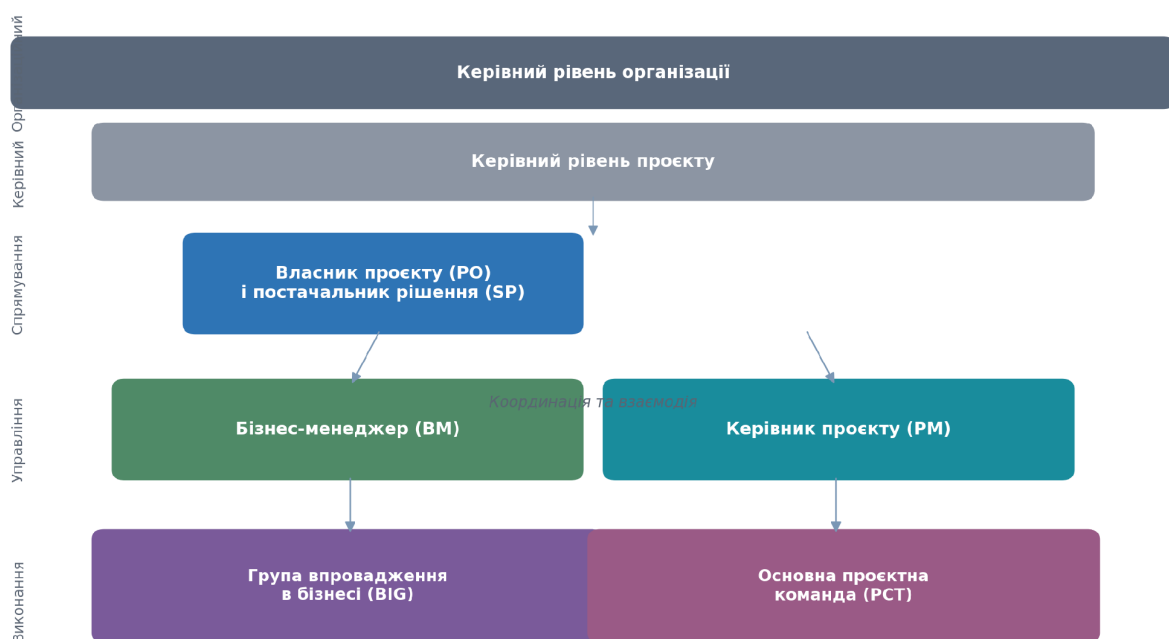


Рисунок 1.5 – Рівні та основні ролі проектної організації PM²

Вищий керівний рівень визначає бачення і стратегію організації, установлює пріоритети, ухвалює інвестиційні рішення та розподіляє ресурси. Керівний рівень проекту забезпечує загальне спрямування і зберігає фокус на цілях. Рівень управління організовує, контролює та координує щоденну роботу. Рівень виконання створює проектні результати й забезпечує їх упровадження в організації.

Загальний підручник виділяє як критично важливі ролі спонсора й вищого керівництва, керівника проекту, членів команди та предметних експертів, зокрема бізнес-аналітика. PM² деталізує ці ролі та розділяє бізнесову й технічну сторони управління. Узгоджене трактування ролей наведено в таблиці 1.4.

Таблиця 4 – Основні ролі проекту та їхня відповідальність

Роль	Основна відповідальність
Власник / спонсор проекту	Підтримує проект на рівні керівництва, володіє бізнес-обґрунтуванням, забезпечує стратегічне спрямування, допомагає отримати ресурси та ухвалює ключові рішення.
Керівник проекту	Планує й координує проект протягом життєвого циклу, організовує виконання, контролює прогрес, комунікації, зміни та передавання результату.
Бізнес-менеджер / представник замовника	Забезпечує відповідність результатів бізнес-потребам, координує впровадження, приймання та організаційні зміни.
Основна проектна команда	Створює результати проекту, надає технічну й предметну експертизу, оцінює роботи та повідомляє про прогрес, проблеми й ризики.
Бізнес-аналітик	Виявляє та документує потреби й вимоги, пропонує життєздатні рішення, перевіряє відповідність результату погодженим вимогам.
Група впровадження в бізнесі	Готує організацію до використання результату, забезпечує навчання, перехід, зміни процесів і реалізацію очікуваних вигод.

Команду формують відповідно до потреб і цілей проєкту. До неї можуть входити працівники, повністю закріплені за проєктом, фахівці з частковою зайнятістю та зовнішні експерти. Важливо не лише зібрати потрібні компетентності, а й забезпечити доступність людей у потрібний час. Формальне внесення прізвища до списку команди не гарантує, що працівник справді має ресурс для виконання завдань.

Керівник проєкту може відповідати за успіх, не маючи повної влади над ресурсами. Тому критичними є комунікація, переговори, лідерство, мотивація та побудова довіри. У функціональній або матричній структурі керівник погоджує залучення фахівців із їхніми функціональними керівниками, а в проєктній — безпосередньо визначає пріоритети команди.

Під час формування команди доцільно:

- визначити ролі, відповідальність, повноваження та порядок підзвітності;
- зіставити потреби проєкту з компетентностями й доступністю кандидатів;
- залучити представників замовника, користувачів і постачальника рішення;
- домовитися про канали комунікації, порядок ескалації та ухвалення рішень;
- установити спільні правила роботи, критерії якості й очікувану поведінку;
- забезпечити обмін знаннями, взаємну підтримку та регулярний зворотний зв'язок.

PM² наголошує, що проєктні ролі слід призначати людям, які найкраще відповідають потребам проєкту, а команда має підтримувати культуру співпраці, чіткої комунікації та підзвітності. Методологія повинна служити проєкту, тому складність ролей і документів допускається адаптувати до масштабу й контексту, не втрачаючи їхнього призначення.

1.6 Життєвий цикл проєкту, його фази та основні результати

Життєвий цикл проєкту. це структура і потік проєктної роботи від початку до завершення. Кожен проєкт має початок, середній період, упродовж якого роботи наближають його до цілі, та завершення — успішне або неуспішне.

Фаза проєкту. група пов'язаних видів діяльності з відносно визначеними початком і завершенням. У прогнозованих підходах фази здебільшого послідовні; в адаптивних окремі роботи можуть перекриватися або повторюватися.

Фаза не є тим самим, що група процесів управління. Наприклад, планування як управлінська діяльність може повторюватися протягом усього проєкту, хоча формально існує окрема фаза планування. Моніторинг і контроль також виконуються паралельно з іншими роботами, оскільки керівник повинен постійно порівнювати фактичний стан із планом.

PM² визначає чотири послідовні фази, що не перекриваються формально: ініціювання, планування, виконання та завершення. У кожній фазі переважає певний тип діяльності, але окремі управлінські роботи можуть продовжуватися в наступних фазах. Моніторинг і контроль здійснюються впродовж усього життєвого циклу.

Між фазами розташовані контрольні точки, або фазові переходи. Вони дають змогу перевірити, чи виконано необхідну роботу, чи схвалено ключові документи та чи готовий проєкт переходити далі. Схему життєвого циклу й переходів показано на рисунку 1.6.



Рисунок 1.6 – Життєвий цикл проєкту за методологією PM²

Ініціювання починається з виявлення потреби, проблеми або можливості. Учасники визначають бажаний результат, аналізують доцільність і формують підстави для подальшого планування. Основними результатами є запит на ініціювання проєкту, бізнес-кейс і статут проєкту. Бізнес-кейс пояснює, чому проєкт потрібний і які обмеження існують, а

статут уточнює зміст, високорівневі вимоги, результати, основні етапи, організацію та критерії успіху.

Недооцінювання ініціювання створює проблеми в наступних фазах: команда може почати детально планувати або розробляти рішення без спільного розуміння потреби та очікуваних вигод. На завершенні фази проєкт проходить контрольну точку RfP — Ready for Planning, тобто перевірку готовності до планування.

Під час планування ціль проєкту розкладають на конкретні результати й роботи, оцінюють необхідні ресурси, тривалість і витрати, визначають залежності, ризики, правила комунікації, управління якістю та змінами. Основними інтегрованими результатами PM² є довідник проєкту, що описує підхід до управління, ролі й правила, та план робіт проєкту, що містить декомпозицію робіт, оцінки трудомісткості й вартості та календарний план.

План не слід сприймати як незмінний прогноз. Він є погодженою базою для координації та контролю, яку уточнюють у межах визначеного порядку. Наприкінці фази здійснюється перехід RfE — Ready for Executing, тобто оцінювання готовності до виконання.

Під час виконання команда створює заплановані результати, а організація-замовник готується до їх упровадження. Це найбільш ресурсомістка фаза, яка потребує інтенсивного моніторингу. Керівник координує людей і постачальників, контролює прогрес, якість і приймання проміжних результатів, опрацьовує проблеми, ризики та запити на зміни.

Моніторинг і контроль включають вимірювання фактичного виконання, порівняння з планом, оцінювання відхилень, прогнозування наслідків і погодження коригувальних дій. Ці процеси не є окремим завершальним кроком: вони відбуваються паралельно з виконанням, а частково — з усіма фазами життєвого циклу. Коли результати створено й виконано умови переходу, проєкт проходить RfC — Ready for Closing.

У фазі завершення готові результати офіційно передаються замовникові або операційному підрозділу. Команда перевіряє виконання зобов'язань, закриває контракти, звільняє ресурси, документує невирішені питання та готує підсумковий звіт. Окремо аналізують досвід: що спрацювало, які труднощі виникли та що потрібно змінити в майбутніх проєктах.

Завершення проєкту не означає завершення життєвого циклу продукту. Після передавання програмне забезпечення переходить в

експлуатацію, супровід і розвиток. Реалізація частини вигод може тривати вже після формального закриття, тому організація повинна призначити відповідальних за використання результату й подальше оцінювання ефекту.

Основні цілі, рушійні ролі та результати фаз узагальнено в таблиці 1.5.

Таблиця 5 – Фази життєвого циклу PM² та їхні основні результати

Фаза	Основна мета	Головний рушій	Ключові результати
Ініціювання	Обґрунтувати потребу та визначити проєкт на високому рівні	Власник проєкту	Запит на ініціювання; бізнес-кейс; статут проєкту; рішення RfP
Планування	Уточнити зміст і підготувати реалістичний спосіб виконання	Керівник проєкту	Довідник проєкту; план робіт; допоміжні плани й журнали; рішення RfE
Виконання	Створити результати та підготувати організацію до впровадження	Основна проєктна команда	Проектні результати; звіти про стан і прогрес; прийняті зміни; рішення RfC
Завершення	Передати результати, оцінити виконання й закрити проєкт	Стейкхолдери та керівник проєкту	Формальне приймання і передавання; підсумковий звіт; засвоєні уроки; архів

Висновки

Проект є тимчасовою діяльністю з визначеною метою, унікальним результатом, обмеженнями, стейкхолдерами, критеріями успіху та певним рівнем невизначеності.

Проектна діяльність відрізняється від операційної тимчасовістю й унікальністю; після передавання результату його постійне використання та підтримка зазвичай переходять до операцій.

Управління проєктами поєднує знання, навички, методи та інструменти для планування, організації виконання, вимірювання прогресу, управління змінами й завершення проєкту.

Проекти є механізмом реалізації стратегії, створення інновацій і цінності через нові або вдосконалені продукти, процеси та послуги.

Успіх управління проектом, успіх самого проекту та організаційний успіх потрібно оцінювати окремо, але взаємопов'язано.

Організаційна структура визначає повноваження керівника, доступність ресурсів, порядок підпорядкування й комунікації; функціональна, матрична та проектна структури створюють різні умови для виконання проекту.

Проектна команда є тимчасовою системою ролей, у якій сторони замовника й постачальника повинні співпрацювати заради спільної мети.

Життєвий цикл забезпечує кероване проходження від потреби до результату через ініціювання, планування, виконання та завершення, а моніторинг і контроль супроводжують увесь проект.

Лекція 2 Ініціація проєкту

Тема: Ініціація проєкту

Мета: Сформувати у студентів розуміння призначення та змісту процесу ініціації проєкту, ознайомити з підходами до формування концепції, визначення цілей і результатів, аналізу припущень та обмежень, ідентифікації зацікавлених сторін, попереднього оцінювання ресурсів, термінів і ризиків, а також обґрунтування доцільності реалізації проєкту.

План лекції

1. Сутність, мета та завдання ініціації проєкту.
2. Управління пріоритетами проєктів.
3. Формування концепції та обґрунтування корисності проєкту.
4. Визначення цілей, завдань і очікуваних результатів.
5. Припущення, обмеження та критерії приймання проєкту.
6. Ключові учасники та зацікавлені сторони.
7. Попереднє оцінювання ресурсів, термінів і ризиків.
8. Висновки.

2.1 Сутність, мета та завдання ініціації проєкту

Ініціація є першою фазою проєкту. Вона починається з виявлення потреби, проблеми або можливості та завершується формальним рішенням щодо подальшої реалізації проєкту. Належна ініціація має критичне значення для успішного планування і виконання проєкту, оскільки на цій фазі визначаються його цілі, очікувані результати, основні обмеження та організаційна підтримка.

Підставою для ініціювання ІТ-проєкту може бути необхідність усунення певної проблеми, задоволення потреб користувачів, використання нової технологічної можливості, виконання вимог законодавства, підвищення ефективності діяльності або реалізація стратегічної мети організації. Наприклад, проєкт може бути ініційований через низьку швидкість оброблення замовлень, недостатню захищеність інформаційної системи, потребу в автоматизації бізнес-процесу або необхідність створення нового цифрового продукту.

Під час ініціації необхідно отримати відповіді на такі питання:

- яка потреба, проблема або можливість зумовлює появу проєкту;
- чому проєкт необхідно реалізувати;
- яких результатів планується досягти;
- як проєкт пов'язаний зі стратегією організації;

- хто є замовником, власником і основним користувачем результатів;

- які ресурси необхідні для реалізації проєкту;
- скільки часу орієнтовно триватиме проєкт;
- які обмеження, припущення та ризики вже відомі;
- за якими показниками оцінюватиметься успішність;
- хто має повноваження схвалити або відхилити проєкт.

Основна мета ініціації полягає не в розробленні детального плану, а у створенні достатнього інформаційного підґрунтя для прийняття рішення про доцільність проєкту. Деталізація складу робіт, календарного плану, ресурсів і бюджету відбувається під час планування. На фазі ініціації ці характеристики визначаються на високому рівні.

Процес ініціації та його основні результати показано на рисунку 2.1.

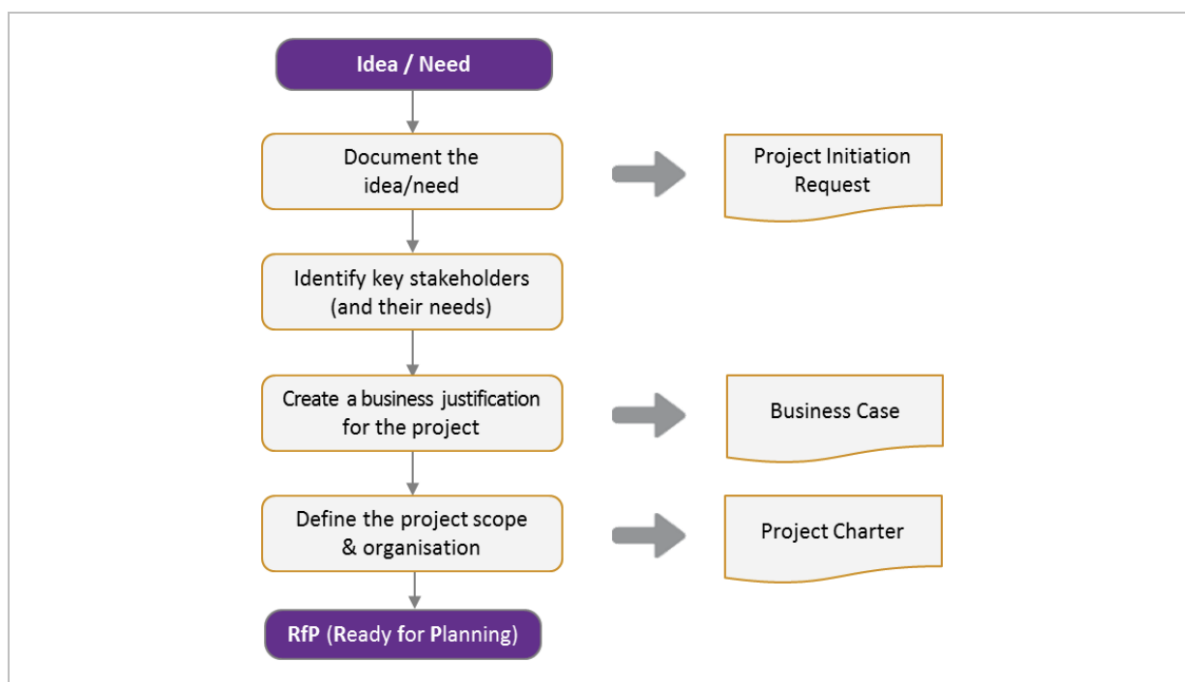


Рисунок 2.1 – Основні дії та результати фази ініціації

У межах фази ініціації послідовно створюються три основні документи: запит на ініціювання проєкту, бізнес-кейс і статут проєкту. Разом вони забезпечують поступовий перехід від початкової ідеї до формально схваленого проєкту.

Таблиця 2.1 – Основні документи фази ініціації

Документ	Призначення	Основний зміст	Результат розгляду
----------	-------------	----------------	--------------------

Запит на ініціювання проєкту	Формалізація проблеми, потреби або можливості	Контекст, очікуваний вплив, попередні цілі, орієнтовні витрати і терміни, припущення, обмеження та ризики	Рішення про підготовку бізнес-кейсу
Бізнес-кейс	Обґрунтування доцільності інвестування часу, коштів і ресурсів	Стратегічна відповідність, альтернативні рішення, вигоди, витрати, ризики, вплив і основні етапи	Рішення про схвалення або відхилення проєкту
Статут проєкту	Формальне визначення та авторизація проєкту	Цілі, високорівневі вимоги, межі, результати, віхи, бюджет, ризики, учасники та повноваження	Дозвіл на перехід до планування

Як видно з таблиці 2.1, зміст документів поступово деталізується. Запит на ініціювання фіксує вихідну ідею, бізнес-кейс обґрунтовує корисність проєкту, а статут надає проєкту офіційний статус.

До основних завдань ініціації належать:

- формалізація потреби, проблеми або можливості;
- визначення зв'язку проєкту зі стратегією організації;
- формулювання попереднього змісту і меж проєкту;
- визначення цілей та очікуваних результатів;
- встановлення показників успішності;
- ідентифікація ключових учасників і зацікавлених сторін;
- попереднє оцінювання ресурсів, вартості й термінів;
- визначення основних припущень, обмежень і ризиків;
- підготовка бізнес-кейсу та статуту;
- отримання рішення про схвалення, відхилення або доопрацювання проєкту.

2.2 Управління пріоритетами проєктів

Організації зазвичай мають більше проєктних ідей, ніж можуть реалізувати за наявних ресурсів і часових обмежень. Тому важливо не лише визначити можливі проєкти, а й обрати ті, які найбільше сприяють досягненню стратегічних цілей.

Проєкти можна поділити на три основні категорії:

- **стратегічні проекти**, спрямовані на створення нових продуктів, послуг, технологій або конкурентних переваг;
- **операційні проекти**, спрямовані на вдосконалення наявних процесів, зменшення витрат, скорочення часу виконання робіт або підвищення якості;
- **обов'язкові проекти**, необхідні для виконання законодавчих, нормативних, безпекових або галузевих вимог.

Під час визначення пріоритетів проектів необхідно враховувати їхню відповідність місії, стратегії та цілям організації. Проект із високою стратегічною цінністю зазвичай отримує ресурси раніше, ніж проект, користь якого обмежується локальним удосконаленням.

Процес відбору можна розглядати як систему послідовних фільтрів. Спочатку аналізується велика кількість ідей, потреб і можливостей. Після перевірки стратегічної відповідності, технічної здійсненності, очікуваної корисності, вартості та ризиків залишається обмежена кількість проектів, рекомендованих до реалізації.

Для відбору можуть застосовуватися якісні та кількісні методи. До якісних належать контрольні списки, експертне оцінювання і зважені моделі. До кількісних — строк окупності, чиста приведена вартість, внутрішня норма прибутковості та співвідношення вигід і витрат.

Використання лише фінансових показників не завжди забезпечує правильний вибір. Наприклад, проект із невисокою прямою прибутковістю може бути необхідним для виконання вимог кібербезпеки або забезпечення відповідності законодавству. Тому для пріоритизації доцільно поєднувати фінансові та нефінансові критерії.

Одним із поширених методів є зважена модель оцінювання. Спочатку визначаються критерії, після чого кожному критерію надається вага відповідно до його важливості. Сума ваг має дорівнювати 100 %. Кожен проект оцінюється за встановленою шкалою, а загальний результат визначається як сума добутків ваг критеріїв на відповідні оцінки.

Таблиця 2.2 – Приклад зваженої моделі оцінювання проектів

Критерій	Вага	Проект А: оцінка	Проект А: результат	Проект В: оцінка	Проект В: результат	Проект С: оцінка	Проект С: результат
Можливість	30	1	30	2	60	2	60

створення додаткових продуктів							
Можливість отримання запланованого доходу	40	2	80	2	80	1	40
Можливість виконання без зовнішніх консультантів	20	2	40	2	40	1	20
Позитивний вплив на репутацію організації	10	0	0	2	20	2	20
Загальний результат	100		150		200		140

Як видно з таблиці 2.2, найвищий результат отримав проєкт В. За умови правильного визначення критеріїв і ваг він має найвищий пріоритет. Водночас зважена модель не усуває суб'єктивність повністю, оскільки вибір критеріїв, ваг та оцінок залежить від рішень експертів і керівництва.

Доцільність відбору повинна також перевірятися з урахуванням доступних ресурсів, строків, взаємозалежностей між проєктами та прийнятного рівня ризику. Остаточне рішення може полягати у схваленні, відхиленні, відкладенні або об'єднанні проєктів.

2.3 Формування концепції та обґрунтування корисності проєкту

Концепція проєкту являє собою узагальнене бачення майбутнього проєкту. Вона пояснює, яку проблему необхідно розв'язати, яку можливість використати, які зміни мають відбутися та яку цінність отримає організація або користувач.

Формування концепції доцільно починати з аналізу трьох складових:

1. **Поточний стан** — наявна ситуація, проблема, потреба або недолік.
2. **Бажаний майбутній стан** — результат, якого організація прагне досягти.

3. **Розрив між станами** — зміни, які необхідно здійснити для переходу від поточного стану до бажаного.

Наприклад, поточним станом може бути ручне оброблення заявок клієнтів, яке триває декілька днів і супроводжується значною кількістю помилок. Бажаним станом є автоматизована система, що забезпечує оброблення заявки протягом декількох хвилин. Проєкт повинен створити програмне рішення, інтегрувати його з наявними системами та забезпечити підготовку користувачів.

Початкову концепцію формалізують у запиті на ініціювання проєкту. У ньому фіксуються контекст, очікуваний вплив, попередні цілі, орієнтовні витрати, строки, основні припущення, обмеження і ризики. Після схвалення запиту готується бізнес-кейс.

Бізнес-кейс містить обґрунтування проєкту, описує його відповідність стратегічним цілям і пояснює необхідність інвестування часу, коштів та організаційних ресурсів. Він повинен надати керівництву достатньо інформації для визначення того, чи варто реалізовувати проєкт.

Послідовність створення основних документів ініціації показано на рисунку 2.2.

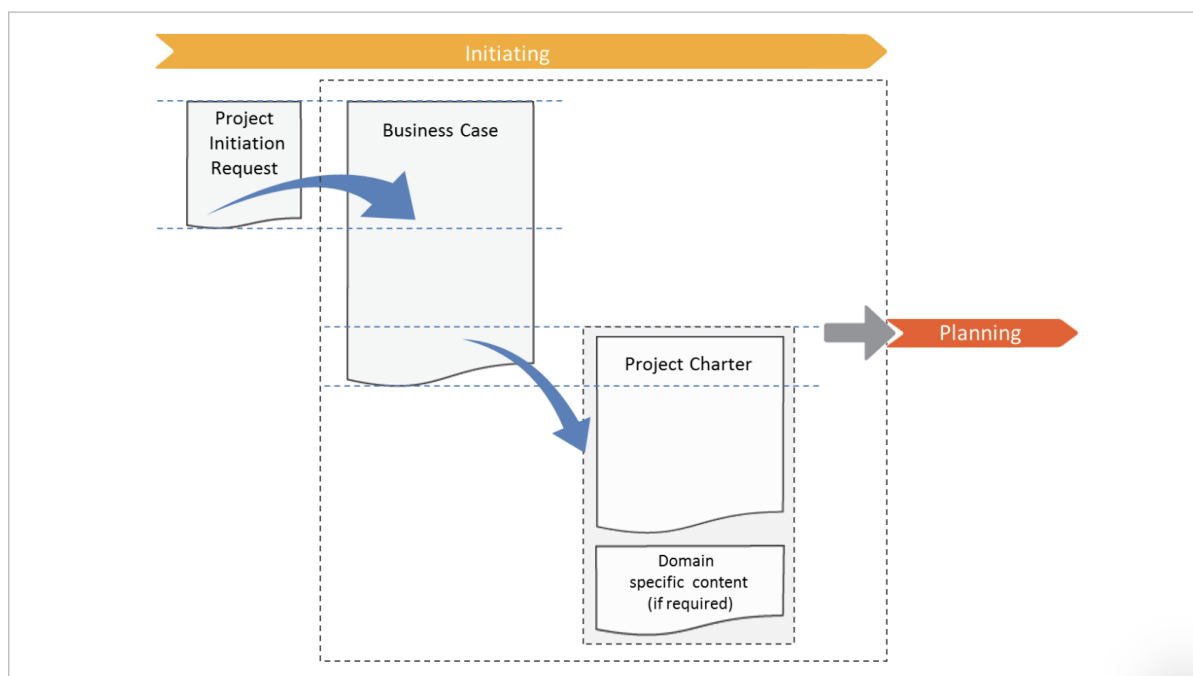


Рисунок 2.2 – Взаємозв’язок документів, створюваних під час ініціації проєкту

Бізнес-кейс повинен розглядати не лише запропоноване рішення, а й альтернативні варіанти. До таких варіантів можуть належати розроблення власної системи, придбання готового програмного продукту, використання хмарного сервісу, модернізація наявного рішення або відмова від реалізації проєкту.

Для кожного варіанта оцінюють:

- відповідність потребам і стратегічним цілям;
- технічну здійсненність;
- орієнтовну вартість;
- тривалість реалізації;
- потребу в людських та інших ресурсах;
- очікувані вигоди;
- можливий вплив на діяльність організації;
- основні ризики;
- залежності від інших систем, проєктів і підрозділів.

Таблиця 2.3 – Основні складові бізнес-кейсу

Складова	Зміст
Опис потреби	Проблема, потреба або можливість, яка є підставою для проєкту
Стратегічне обґрунтування	Зв'язок проєкту з місією, цілями та пріоритетами організації
Поточний стан	Наявні процеси, системи, показники та недоліки
Бажаний стан	Очікувані зміни та характеристики майбутнього рішення
Зацікавлені сторони	Основні користувачі, замовники, учасники та їхні потреби
Альтернативні рішення	Можливі способи розв'язання проблеми
Вигоди	Фінансові, операційні, соціальні, стратегічні та якісні результати
Витрати	Орієнтовні витрати на створення, впровадження, експлуатацію і підтримку
Ризики та обмеження	Основні невизначеності та межі реалізації
Основні віхи	Попередня послідовність ключових результатів і строків
Рекомендоване рішення	Варіант, запропонований для реалізації, та його обґрунтування

Корисність проєкту може бути виражена у фінансових і нефінансових показниках. До фінансових вигід належать збільшення доходів, скорочення витрат, зменшення втрат і підвищення продуктивності. До нефінансових — покращення якості обслуговування, скорочення часу виконання операцій, підвищення безпеки, задоволеності користувачів, відповідності нормативним вимогам або конкурентоспроможності.

Бізнес-кейс не є документом, який створюється один раз і більше не переглядається. Під час реалізації проєкту його необхідно повторно оцінювати у ключових точках, щоб перевірити, чи залишаються очікувані вигоди досяжними, а витрати і строки — прийнятними. Якщо проєкт втрачає актуальність або перестає забезпечувати очікувану цінність, може бути прийняте рішення про його зміну або припинення.

2.4 Визначення цілей, завдань і очікуваних результатів

Мета проєкту описує загальний бажаний стан, якого необхідно досягти. Цілі конкретизують мету і визначають вимірювані зміни. Завдання описують основні дії, необхідні для досягнення цілей. Очікувані результати визначають те, що має бути створено або змінено внаслідок реалізації проєкту.

При формулюванні цілей доцільно застосовувати підхід SMART. Відповідно до нього цілі повинні бути конкретними, вимірюваними, досяжними, релевантними та обмеженими в часі.

Таблиця 2.4 – Критерії формулювання цілей за підходом SMART

Критерій	Характеристика	Контрольне питання
Конкретність	Ціль чітко визначає необхідний результат	Що саме має бути досягнуто?
Вимірюваність	Для цілі встановлені кількісні або перевірювані показники	Як буде визначено ступінь досягнення цілі?
Досяжність	Ціль відповідає наявним ресурсам, компетентностям і умовам	Чи можна досягти цілі за наявних умов?
Релевантність	Ціль відповідає потребі та стратегії організації	Чому досягнення цієї цілі є важливим?
Обмеженість у часі	Встановлено строк або дату досягнення цілі	Коли має бути отримано результат?

Нечітке формулювання може мати такий вигляд: «Покращити обслуговування користувачів». Воно не встановлює конкретного результату, показників і строку.

Формулювання за підходом SMART може бути таким: «Протягом шести місяців після запуску системи скоротити середній час оброблення звернення користувача з 24 до 4 годин і забезпечити рівень задоволеності користувачів не нижче 85 %».

Для визначення цілей необхідно знати початкове значення показника. Без базового показника неможливо об'єктивно визначити зміни. Якщо планується скоротити час оброблення заявок на 50 %, спочатку необхідно встановити його поточне середнє значення.

Цілі проєкту повинні бути пов'язані з очікуваними результатами. При цьому необхідно розрізняти безпосередні результати та вигоди. Безпосереднім результатом ІТ-проєкту може бути розроблена і впроваджена інформаційна система. Вигодою є скорочення часу виконання операцій, зменшення кількості помилок або підвищення задоволеності користувачів.

Наприклад, для проєкту впровадження системи електронного документообігу можна визначити:

- **мету** — підвищити ефективність оброблення внутрішніх документів;
- **ціль** — протягом трьох місяців після впровадження скоротити середній час погодження документа з п'яти до двох робочих днів;
- **завдання** — проаналізувати процеси, визначити вимоги, налаштувати систему, інтегрувати її з наявними сервісами, перенести дані та навчити користувачів;
- **безпосередній результат** — введена в експлуатацію система електронного документообігу;
- **очікувана вигода** — скорочення тривалості погодження документів і зменшення витрат на паперове опрацювання.

Високорівневі цілі й результати фіксуються у статуті проєкту. Статут формально підтверджує існування проєкту, надає керівнику проєкту повноваження використовувати організаційні ресурси та створює основу для подальшого планування.

До статуту доцільно включити:

- призначення проєкту;
- вимірювані цілі;
- високорівневі вимоги;
- загальний опис і межі проєкту;
- основні результати;
- припущення та обмеження;
- загальний рівень ризику;
- ключові віхи;
- попередньо схвалений бюджет;
- ключових зацікавлених сторін;
- вимоги до схвалення результатів;
- критерії завершення або припинення;
- склад проєктної команди;
- ім'я та повноваження особи, яка схвалює проєкт.

Для підготовки документа рекомендується використовувати шаблон Project Charter із комплекту артефактів PM² [7].

2.5 Припущення, обмеження та критерії приймання проєкту

На початку проєкту частина інформації залишається невідомою або неповною. Для забезпечення можливості попереднього оцінювання формуються припущення. Одночасно визначаються обмеження, у межах яких проєкт має бути реалізований.

Припущення — це твердження або умови, які на етапі ініціації вважаються істинними, хоча ще не мають повного підтвердження. Наприклад, команда може припускати, що необхідне фінансування буде виділено вчасно, ключові фахівці будуть доступні, а зовнішній постачальник надасть програмний інтерфейс до визначеної дати.

Припущення потрібно перевіряти та документувати. Якщо припущення виявиться неправильним, воно може стати джерелом ризику. Наприклад, припущення про доступність фахівця може перетворитися на ризик затримки, якщо цей фахівець буде залучений до іншого проєкту.

Обмеження — це встановлені межі або умови, які звужують можливості команди. Обмеження можуть стосуватися строків, бюджету, технологій, ресурсів, нормативних вимог, безпеки або сумісності з наявними системами. Наприклад, проєкт повинен бути завершений до початку навчального року, його бюджет не може перевищувати

встановлену суму, а програмне рішення має працювати з наявною базою даних.

Критерії приймання — це конкретні умови, за якими замовник або уповноважена сторона визначає, чи відповідає результат установленим вимогам. Вони повинні бути чіткими, вимірюваними, перевірюваними та погодженими до початку виконання основних робіт.

Таблиця 2.5 – Відмінності між припущеннями, обмеженнями та критеріями приймання

Поняття	Сутність	Приклад для ІТ-проєкту
Припущення	Умова, яка тимчасово вважається істинною	Представники замовника щотижня братимуть участь у погодженні вимог
Обмеження	Обов’язкова межа реалізації проєкту	Систему необхідно впровадити до 1 вересня в межах затвердженого бюджету
Критерій приймання	Умова, виконання якої підтверджує придатність результату	Система повинна підтримувати не менше 1000 одночасних користувачів із часом відповіді до двох секунд

Усі припущення, обмеження та критерії приймання мають бути сформульовані однозначно. Наприклад, критерій «система повинна працювати швидко» не є придатним для перевірки. Натомість критерій «95 % запитів повинні оброблятися не довше ніж за дві секунди» можна виміряти під час тестування.

На фазі ініціації визначаються переважно високорівневі критерії приймання. Під час планування і розроблення вимог вони уточнюються та деталізуються для окремих функцій, модулів і результатів.

2.6 Ключові учасники та зацікавлені сторони

Зацікавленими сторонами є особи або групи, які можуть впливати на проєкт, зазнавати впливу його діяльності чи результатів або вважати себе такими, на кого цей проєкт впливає. Вони можуть безпосередньо брати участь у роботі, належати до інших підрозділів організації або бути зовнішніми щодо неї.

До зацікавлених сторін ІТ-проєкту можуть належати:

- замовник;
- спонсор або власник проєкту;
- керівник проєкту;

- бізнес-аналітики;
- розробники, тестувальники і системні адміністратори;
- майбутні користувачі;
- керівники функціональних підрозділів;
- постачальники програмного забезпечення та обладнання;
- регуляторні органи;
- служби інформаційної безпеки;
- служби підтримки та експлуатації;
- інші проекти або підрозділи, які залежать від результату.

Кількість зацікавлених сторін залежить від складності й масштабу проекту. Чим більше людей і процесів охоплюють майбутні зміни, тим вища ймовірність появи сторін, які мають значний вплив на проект. Зацікавлені сторони можуть підтримувати проект або перешкоджати його реалізації, тому їх необхідно визначити якомога раніше.

Загальну структуру проектної організації доцільно показати на рисунку 2.3.

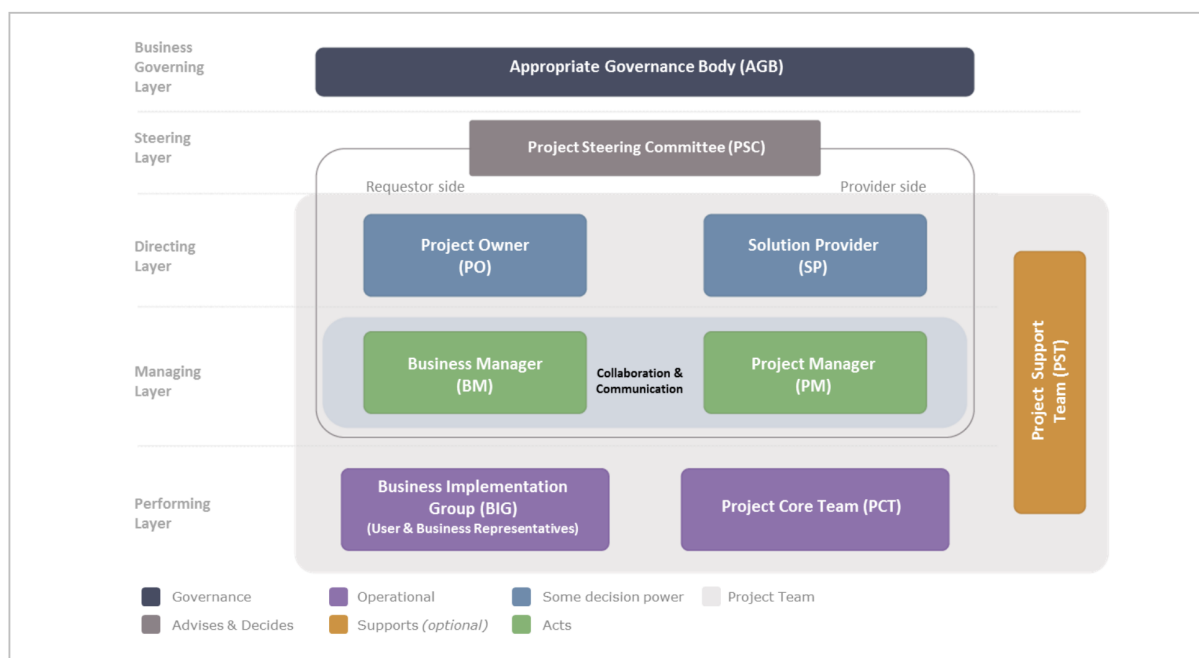


Рисунок 2.3 – Організація проекту та основні рівні управління

На фазі ініціації особливо важливими є такі учасники.

Таблиця 2.6 – Ключові учасники процесу ініціації

Учасник	Основна роль
---------	--------------

Ініціатор	Формулює початкову проблему, потребу або можливість і подає запит на ініціювання
Спонсор або власник проєкту	Відповідає за бізнес-доцільність, забезпечує підтримку, приймає ключові рішення і схвалює проєкт
Бізнес-менеджер або бізнес-аналітик	Аналізує потреби, формує бізнес-кейс, збирає високорівневі вимоги та представляє інтереси бізнесу
Постачальник рішення	Оцінює технічну здійсненність і можливості створення необхідного результату
Керівник проєкту	Завершує підготовку статуту, координує учасників і після схвалення організовує планування
Проектна команда	Надає професійні оцінки та надалі виконує роботи зі створення результатів
Орган управління або керівний комітет	Розглядає бізнес-кейс і статут та приймає рішення про схвалення або відхилення
Користувачі	Формулюють потреби, вимоги та критерії приймання майбутнього продукту

Спонсор є однією з найвпливовіших зацікавлених сторін. Він має повноваження підтримувати проєкт, забезпечувати виділення ресурсів, усувати організаційні перешкоди, схвалювати статут, а за необхідності — ініціювати призупинення або припинення робіт.

Керівник проєкту може бути призначений після схвалення бізнес-кейсу або до завершення підготовки статуту. Підписаний статут офіційно надає йому повноваження розпочати планування та залучати ресурси організації.

Ідентифікація зацікавлених сторін повинна передбачати визначення їхніх інтересів, очікувань, впливу, ставлення до проєкту та інформаційних потреб. Результати доцільно оформити у вигляді матриці зацікавлених сторін. Для цього можна використати шаблон **Project Stakeholder Matrix** із комплекту артефактів PM² [7].

2.7 Попереднє оцінювання ресурсів, термінів і ризиків

На фазі ініціації оцінювання виконується на високому рівні. Його мета полягає в тому, щоб встановити загальний масштаб проєкту, перевірити його здійсненність та забезпечити достатню інформацію для прийняття рішення. Детальний розподіл ресурсів, календарний план і бюджет розробляються під час планування.

До основних ресурсів ІТ-проєкту належать:

- працівники та їхні компетентності;
- робочий час;
- фінансування;
- комп'ютерне обладнання;
- програмні засоби та ліцензії;
- хмарна і мережева інфраструктура;
- тестові середовища;
- дані;
- послуги зовнішніх постачальників;
- приміщення та засоби комунікації.

Попереднє оцінювання людських ресурсів визначає, які ролі та компетентності будуть потрібні. Наприклад, для створення вебсистеми можуть знадобитися керівник проєкту, бізнес-аналітик, архітектор, розробники, дизайнер інтерфейсу, тестувальник, фахівець із кібербезпеки та системний адміністратор.

Попереднє оцінювання термінів ґрунтується на високорівневому змісті, основних результатах, відомих залежностях і доступності ресурсів. Замість детального календарного плану визначають орієнтовні дати початку і завершення та основні віхи.

Віха є значущою подією або точкою контролю, яка не має власної тривалості. Прикладами віх можуть бути «бізнес-кейс схвалено», «вимоги погоджено», «прототип прийнято», «систему введено в експлуатацію».

Ризиком є невизначена подія або умова, яка у разі виникнення може вплинути на досягнення цілей. На фазі ініціації визначаються насамперед високорівневі ризики, здатні поставити під сумнів доцільність або здійсненність усього проєкту.

До типових початкових ризиків ІТ-проєктів належать:

- нечіткість потреб і цілей;
- недостатність фінансування;
- відсутність потрібних фахівців;
- залежність від зовнішнього постачальника;
- складність інтеграції з наявними системами;
- низька якість вихідних даних;
- зміни законодавчих або безпекових вимог;
- недостатня підтримка керівництва;

- опір користувачів;
- нереалістичні строки;
- використання нової або недостатньо перевіреної технології.

Таблиця 2.7 – Попереднє оцінювання основних параметрів проєкту

Напрямок оцінювання	Що необхідно визначити	Результат ініціації
Зміст	Основні межі, функції та результати	Високорівневий опис проєкту
Людські ресурси	Необхідні ролі, компетентності й приблизна чисельність команди	Попередній склад команди
Технічні ресурси	Обладнання, програмне забезпечення, дані й інфраструктура	Перелік основних потреб
Фінанси	Орієнтовні витрати та джерела фінансування	Попередній бюджет
Терміни	Початок, завершення, ключові віхи й зовнішні строки	Високорівнева дорожня карта
Ризики	Найбільш значущі невизначеності та їхній можливий вплив	Початковий перелік ризиків
Залежності	Інші проєкти, системи, постачальники та рішення	Перелік критичних залежностей
Критерії успішності	Показники, за якими оцінюватиметься результат	Вимірювані показники успіху

Для кожного початкового ризику необхідно коротко визначити причину, можливу подію, наслідок і відповідального за подальший аналіз. На цьому етапі може бути створений початковий реєстр ризиків, який деталізуватиметься під час планування.

Перед схваленням проєкту необхідно перевірити:

- чи сформульовано потребу або проблему;
- чи відповідає проєкт стратегії організації;
- чи визначено вимірювані цілі;
- чи зрозумілі межі та очікувані результати;
- чи розглянуто альтернативні рішення;
- чи є очікувані вигоди достатніми;
- чи можуть бути надані необхідні ресурси;
- чи є орієнтовні строки реалістичними;
- чи визначено основні ризики;

- чи встановлено критерії приймання;
- чи призначено спонсора та керівника проєкту;
- чи погоджено бізнес-кейс і статут.

Місце ініціації у загальному циклі проєкту показано на рисунку 2.4.

****[Місце для вставлення рисунка]****

****Рисунок 2.4 – Передпроєктна робота, життєвий цикл проєкту та післяпроєктна діяльність****

Звідки взяти рисунок: джерело [1], розділ 2, підрозділ 2.3 «Business Case», Figure 2.3.1 «Pre-Project Work, Project Life Cycle, and Post-Project Work».

Рішенням за результатами ініціації може бути схвалення проєкту, повернення документів на доопрацювання, відкладення проєкту або відмова від його реалізації. У разі схвалення бізнес-кейсу та статуту проєкт переходить до фази планування.

Висновки

Ініціація забезпечує перехід від початкової ідеї до формально визначеного та схваленого проєкту. Вона починається з виявлення потреби, проблеми або можливості та завершується прийняттям управлінського рішення щодо подальшої реалізації.

Під час ініціації встановлюється зв'язок проєкту зі стратегією організації, обґрунтовується його корисність, визначаються цілі, високорівневі вимоги, результати, припущення, обмеження і критерії приймання. Одночасно ідентифікуються основні учасники та зацікавлені сторони, попередньо оцінюються ресурси, строки, витрати і ризики.

Головними документами фази є запит на ініціювання, бізнес-кейс і статут проєкту. Запит формалізує початкову потребу, бізнес-кейс пояснює, чому проєкт варто реалізувати, а статут офіційно підтверджує існування проєкту та надає керівнику необхідні повноваження.

Якісна ініціація не гарантує успіху автоматично, але суттєво зменшує ризик реалізації проєкту з нечіткими цілями, невизначеними результатами, недостатніми ресурсами або сумнівною корисністю. Саме

тому перехід до детального планування повинен відбуватися лише після погодження основних параметрів і підтвердження доцільності проєкту.

Лекція 3 Планування проєкту

Тема: Планування проєкту

Мета: Сформувати у студентів розуміння процесів планування IT-проєкту, ознайомити з підходами до визначення змісту і складу робіт, планування організаційної структури, управління конфігураціями та якістю, а також формування базового розкладу проєкту.

План лекції

1. Мета, завдання та результати планування проєкту.
2. Уточнення змісту і складу проєктних робіт.
3. Планування управління змістом проєкту.
4. Планування організаційної структури та розподілу відповідальності.
5. Планування управління конфігураціями та якістю.
6. Формування базового розкладу проєкту.
7. Висновки.

3.1 Мета, завдання та результати планування проєкту

Планування є другою фазою життєвого циклу проєкту. Воно починається після схвалення бізнес-кейсу і статуту проєкту та завершується після розроблення, погодження й затвердження основних проєктних планів і базових показників.

Мета планування полягає у визначенні того, що саме необхідно створити, які роботи для цього потрібно виконати, хто їх виконуватиме, коли вони мають бути завершені, які ресурси будуть використані та яким чином контролюватимуться зміни, якість і прогрес проєкту.

Планування перетворює загальну концепцію проєкту на конкретну модель його реалізації. Статут містить високорівневе визначення проєкту, тоді як під час планування його зміст уточнюється, результати декомпонуються, визначаються роботи, відповідальні особи, ресурси, залежності, терміни та критерії якості.

Основні дії та результати фази планування показано на рисунку 3.1.

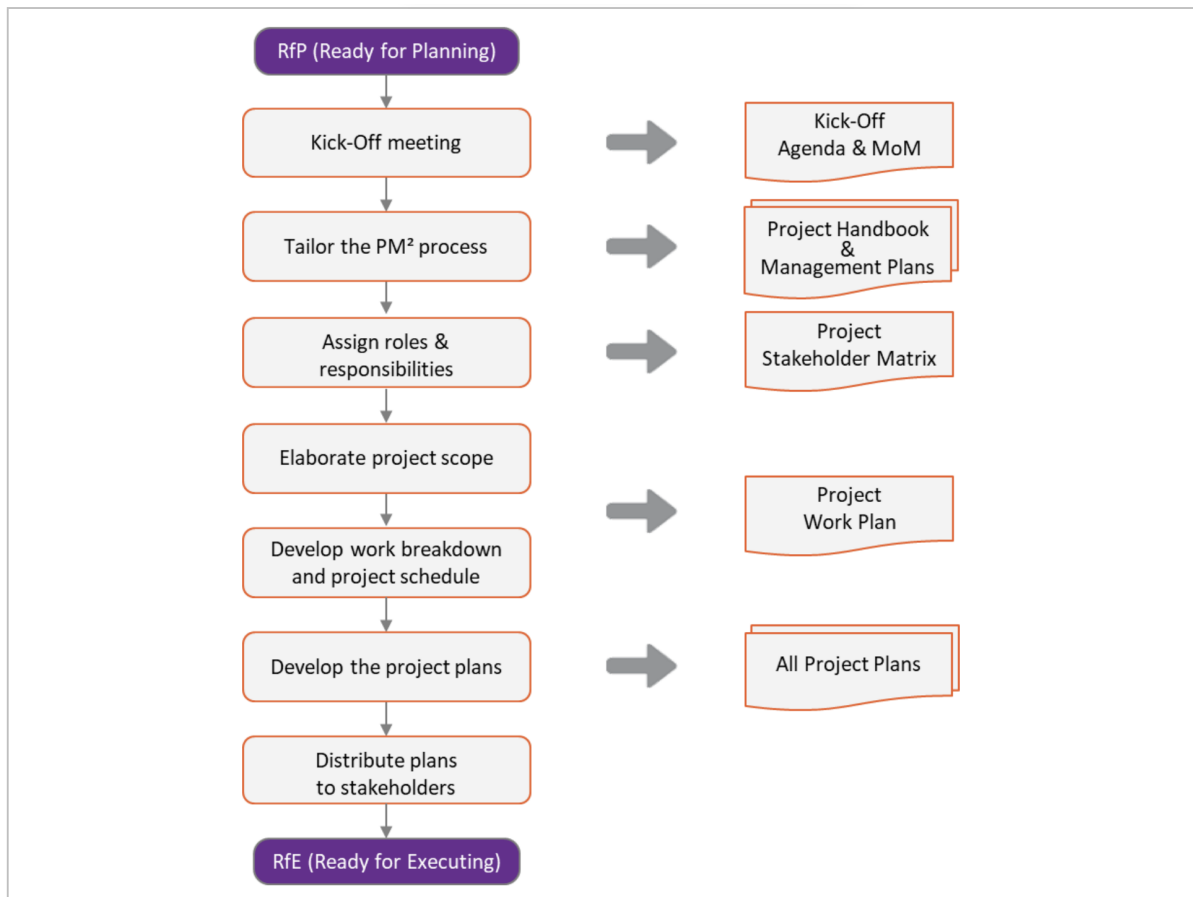


Рисунок 3.1 – Основні дії та результати фази планування проєкту

Планування не є одноразовою дією. У міру отримання нової інформації окремі елементи плану уточнюються. У прогнозованих проєктах більшість робіт намагаються визначити до початку реалізації. В адаптивних проєктах загальні межі встановлюються на початку, а детальне планування здійснюється поступово для окремих ітерацій або частин продукту.

Під час планування необхідно:

- уточнити цілі, межі та результати проєкту;
- зібрати і систематизувати вимоги;
- визначити склад проєктних робіт;
- розробити ієрархічну структуру робіт;
- призначити відповідальних за результати;
- визначити потребу в людських, технічних і фінансових ресурсах;
- встановити залежності між роботами;
- оцінити їхню тривалість;
- сформувати календарний розклад;

- визначити вимоги та показники якості;
- встановити порядок управління змінами і конфігураціями;
- погодити правила приймання результатів;
- сформувані базові показники змісту, розкладу та бюджету.

Основними документами фази планування є настанова з управління проектом, робочий план проекту, матриця зацікавлених сторін, плани управління змістом, вимогами, змінами, ризиками, якістю, комунікаціями, ресурсами та прийманням результатів.

Таблиця 3.1 – Основні результати планування проекту

Результат	Призначення
Настанова з управління проектом	Визначає загальний підхід, правила, процеси, ролі та порядок управління
Опис змісту проекту	Встановлює межі, результати, включення та виключення
Ієрархічна структура робіт	Поділяє проект на керовані компоненти і пакети робіт
Матриця відповідальності	Закріплює ролі учасників щодо окремих результатів і робіт
План управління якістю	Визначає вимоги, показники, перевірки та відповідальних за якість
Процедура управління конфігураціями	Визначає порядок ідентифікації, версіонування та зміни проектних об'єктів
Робочий план проекту	Об'єднує роботи, оцінки трудомісткості, витрати, ресурси та розклад
Базовий розклад	Фіксує затверджені дати початку і завершення робіт та основні віхи
План приймання результатів	Визначає критерії, процедури, строки та відповідальних за приймання

Загальний зв'язок між плануванням змісту, розкладу і бюджету показано на рисунку 3.2.

Project Planning Areas		
Scope	Schedule	Budget
• Project Scope & Deliverables • Requirements • Responsibilities	• Activities • Resources • Relationships	• Activity-Related Costs • Other Project Costs • Reserves

Рисунок 3.2 – Основні напрями планування проєкту

У результаті планування всі учасники повинні однаково розуміти, що, як, ким і коли буде створено. Після погодження планів проєкт може переходити до виконання.

3.2 Уточнення змісту і складу проєктних робіт

Зміст проєкту визначає характеристики його майбутніх результатів і всю роботу, яку необхідно виконати для їх створення. Він установлює, що входить до проєкту, а що перебуває поза його межами.

Зміст необхідно визначити достатньо чітко, щоб замовник, керівник і команда однаково розуміли очікуваний результат. Нечіткі межі призводять до появи додаткових вимог, неконтрольованого збільшення обсягу робіт, затримок і перевищення бюджету.

Опис змісту проєкту зазвичай містить:

- назву та ідентифікатор проєкту;
- посилання на статут;
- мету та вимірювані цілі;
- опис продукту або послуги;
- високорівневі вимоги;
- основні результати;
- критерії приймання;
- роботи та результати, включені до проєкту;
- виключення з проєкту;
- припущення;
- обмеження.

Наприклад, для проєкту створення університетської інформаційної системи зміст можна сформулювати так, як показано в таблиці 3.2.

Таблиця 3.2 – Приклад опису змісту ІТ-проєкту

Елемент	Приклад
Назва проєкту	Система електронного подання студентських заяв
Мета	Скорочення часу подання, погодження та опрацювання заяв
Основні результати	Вебзастосунок, база даних, особисті кабінети, модуль погодження, журнал операцій, документація
Включено до проєкту	Авторизація користувачів, створення заяв, завантаження документів, погодження, повідомлення про статус
Не включено до проєкту	Мобільний застосунок, інтеграція з бухгалтерською системою, електронний цифровий підпис
Основні користувачі	Студенти, викладачі, працівники деканату, адміністратори
Критерії приймання	Підтримка не менше 500 одночасних користувачів; час відповіді до двох секунд; коректне проходження погодження
Обмеження	Використання наявної серверної інфраструктури; завершення до початку навчального року

Під час визначення змісту необхідно описати всі проміжні та кінцеві результати. Результатом може бути програмний продукт, окремий модуль, база даних, проєктна документація, навчальні матеріали, налаштоване середовище або звіт про тестування.

Кожен результат має бути конкретним і перевірюваним. Формулювання «підготувати документацію» є недостатньо точним. Необхідно визначити вид документації: технічне завдання, специфікація вимог, інструкція користувача, інструкція адміністратора або опис програмного інтерфейсу.

Після визначення результатів виконується декомпозиція проєкту — послідовний поділ складного результату на менші компоненти. Для цього використовується ієрархічна структура робіт, або WBS.

WBS поділяє весь зміст проєкту на компоненти, які можна оцінити, призначити виконавцям, включити до розкладу та контролювати. Кожен нижчий рівень структури містить детальніше представлення роботи.

Приклад ієрархічної структури робіт наведено на рисунку 3.3.

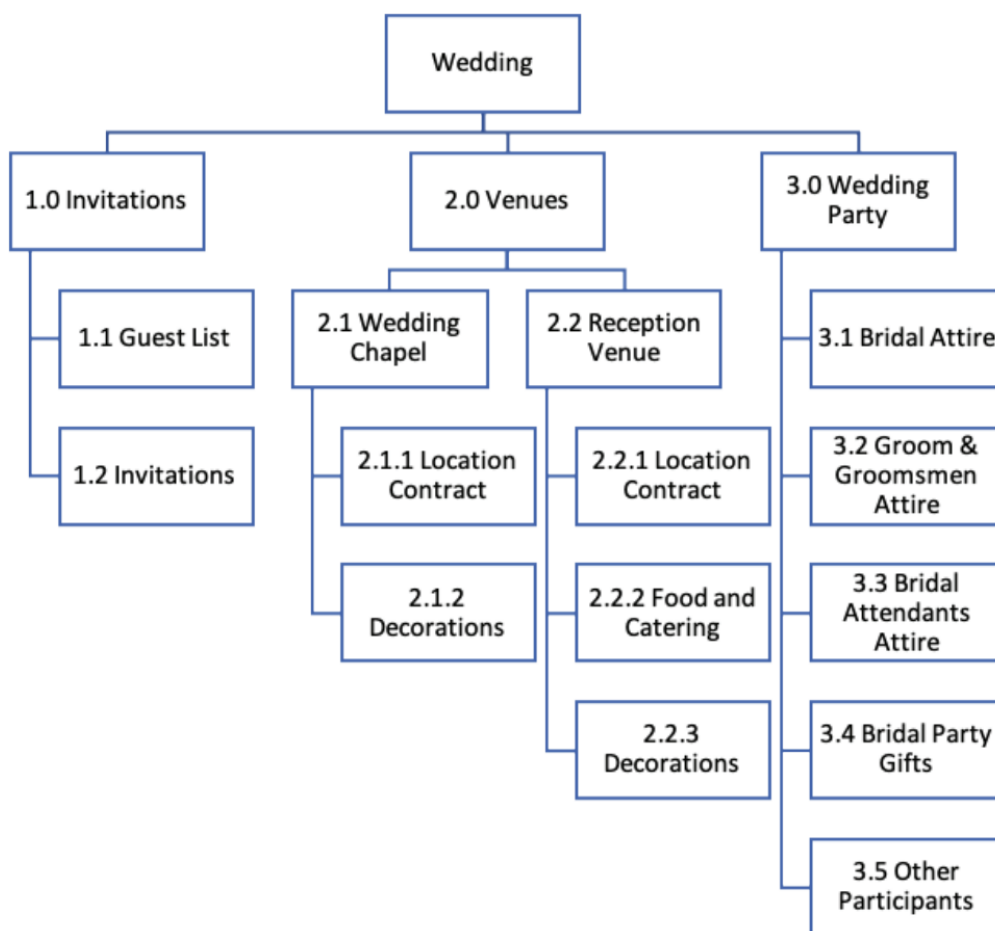


Рисунок 3.3 – Приклад ієрархічної структури робіт проекту

Нижній рівень WBS називається пакетом робіт. Пакет робіт повинен бути достатньо малим для оцінювання трудомісткості, вартості й тривалості, але не настільки деталізованим, щоб керування ним створювало зайву складність.

Таблиця 3.3 – Приклад WBS проекту створення інформаційної системи

Код WBS	Компонент або пакет робіт
1.0	Управління проектом
1.1	Підготовка плану проекту
1.2	Моніторинг і звітність
1.3	Управління змінами
2.0	Аналіз вимог
2.1	Збір потреб зацікавлених сторін
2.2	Формування функціональних вимог

2.3	Формування нефункціональних вимог
2.4	Погодження специфікації вимог
3.0	Проектування системи
3.1	Проектування архітектури
3.2	Проектування бази даних
3.3	Проектування користувацького інтерфейсу
4.0	Розроблення
4.1	Модуль авторизації
4.2	Модуль створення і подання заяв
4.3	Модуль погодження
4.4	Модуль повідомлень
4.5	Журналювання операцій
5.0	Тестування
5.1	Модульне тестування
5.2	Інтеграційне тестування
5.3	Навантажувальне тестування
5.4	Приймальне тестування
6.0	Впровадження
6.1	Підготовка середовища
6.2	Розгортання системи
6.3	Перенесення початкових даних
6.4	Навчання користувачів
7.0	Завершення проєкту
7.1	Передавання результатів
7.2	Архівування документації
7.3	Формування підсумкового звіту

Під час побудови WBS необхідно дотримуватися правила 100 %. Сукупність дочірніх компонентів повинна охоплювати 100 % роботи, визначеної батьківським компонентом. У структурі не повинно бути робіт, які не належать до змісту проєкту, і водночас не можна пропускати роботи, необхідні для отримання результату.

WBS не визначає послідовність виконання. Спочатку необхідно встановити повний склад робіт, а вже потім визначати залежності, ресурси і строки.

Для кожного пакета робіт може створюватися словник WBS, який містить:

- код та назву пакета;
- опис результату;
- відповідального;
- вимоги;
- критерії приймання;
- припущення й обмеження;
- орієнтовні ресурси;
- трудомісткість;
- залежності;
- контрольні точки.

3.3 Планування управління змістом проєкту

План управління змістом визначає, як зміст проєкту буде формуватися, погоджуватися, перевірятися та контролюватися. Він описує не сам зміст, а правила роботи з ним.

План повинен установлювати:

- хто відповідає за визначення змісту;
- як збираються і документуються вимоги;
- хто погоджує вимоги та результати;
- як створюється й оновлюється WBS;
- як перевіряється повнота результатів;
- яким чином подаються запити на зміну;
- хто оцінює вплив змін;
- хто має право схвалювати зміни;
- як оновлюються базові показники;
- яким чином попереджається неконтрольоване розширення змісту.

Таблиця 3.4 – Приклад структури плану управління змістом

Розділ	Зміст
Визначення змісту	Методи збору вимог і підготовки опису змісту
Декомпозиція	Правила створення та кодування WBS

Перевірка змісту	Порядок перевірки результатів і критерії приймання
Контроль змісту	Порядок виявлення відхилень від погодженого змісту
Управління змінами	Реєстрація, аналіз, схвалення і реалізація змін
Ролі та повноваження	Особи, відповідальні за підготовку, перевірку і схвалення
Документування	Перелік документів, журналів і репозиторіїв
Звітність	Порядок повідомлення про стан змісту та зміни

Однією з основних проблем є неконтрольоване розширення змісту — поступове додавання нових функцій, робіт або результатів без формального погодження і відповідного коригування ресурсів, бюджету та строків.

Наприклад, після погодження вебсистеми замовник може попросити додати мобільний застосунок, інтеграцію з додатковою базою даних і новий аналітичний модуль. Якщо ці вимоги реалізуються без оцінювання впливу та зміни плану, команда фактично виконує більший обсяг робіт у межах попередніх строків і ресурсів.

Схожим явищем є надмірне вдосконалення, коли члени команди самостійно додають функції, яких не вимагав замовник. Навіть якщо такі функції здаються корисними, вони можуть збільшити складність, тривалість тестування і витрати на підтримку.

Зміна змісту повинна проходити формальну процедуру:

1. Реєстрація запиту на зміну.
2. Опис причини й очікуваної цінності.
3. Аналіз впливу на зміст, строки, бюджет, ресурси, ризики та якість.
4. Прийняття рішення уповноваженою особою або органом.
5. Реалізація схваленої зміни.
6. Оновлення планів, вимог, конфігураційних одиниць і базових показників.
7. Інформування зацікавлених сторін.

Процес опрацювання змін можна проілюструвати рисунком 3.4.

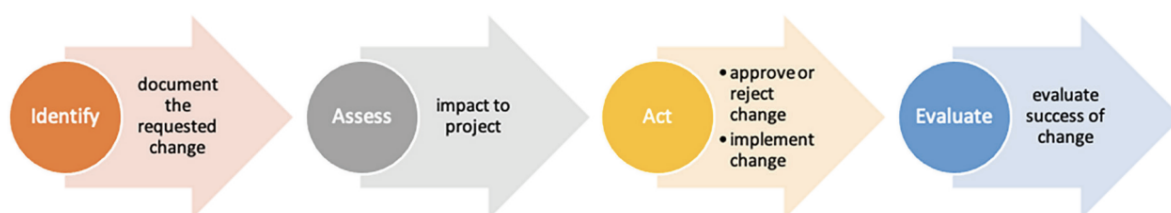


Рисунок 3.4 – Спрощений процес управління змінами проєкту

Після затвердження опис змісту, WBS і пов’язані вимоги утворюють базовий зміст проєкту. Він використовується для порівняння запланованих і фактично виконаних робіт. Зміна базового змісту допускається лише після проходження встановленої процедури.

3.4 Планування організаційної структури та розподілу відповідальності

Після визначення результатів і пакетів робіт необхідно встановити, хто відповідатиме за їх виконання, погодження, перевірку та приймання.

Організаційна структура проєкту повинна враховувати:

- структуру компанії;
- склад і розташування команди;
- компетентності учасників;
- повноваження керівника проєкту;
- участь функціональних керівників;
- роль замовника і спонсора;
- участь зовнішніх постачальників;
- порядок прийняття та ескалації рішень.

До основних проєктних ролей можуть належати спонсор, власник проєкту, керівник проєкту, бізнес-аналітик, архітектор, розробники, тестувальники, фахівці з інформаційної безпеки, системні адміністратори, представники користувачів і працівники підтримки.

Для закріплення відповідальності використовується матриця розподілу відповідальності. Одним із її найпоширеніших варіантів є матриця RACI:

- R — Responsible: виконує роботу;
- A — Accountable: несе остаточну відповідальність і схвалює результат;
- C — Consulted: надає консультації та бере участь в обговоренні;
- I — Informed: отримує інформацію про стан або результат.

Для кожного пакета робіт бажано визначати лише одну особу з роллю А. Водночас декілька осіб можуть мати ролі R, C або I.

Приклад матриці RACI з підручника показано на рисунку 3.5.

Deliverable	Bride	Groom	Bride's Mother	Maid of Honor	Best Man
3.0 Wedding Party					
3.1 Bridal Attire	A	I	R	C	I
3.2 Groom & Groomsmen Attire	C	A	C	I	R
3.3 Bridal Attendants Attire	A	I	C	R	I
3.4 Bridal Party Gifts	C	C	A	C	R
3.5 Other Participants	C	I	R	A	I

Legend Responsible Accountable Consulted Informed

Рисунок 3.5 – Приклад матриці розподілу відповідальності RACI

Таблиця 3.5 – Приклад матриці RACI для IT-проєкту

Результат або робота	Спонсор	Керівник проєкту	Бізнес-аналітик	Розробник	Тестувальник	Представник користувачів
Специфікація вимог	I	A	R	C	C	C
Архітектура системи	I	A	C	R	C	I
Програмна реалізація	I	A	C	R	C	I
План тестування	I	A	C	C	R	C
Приймальне тестування	I	A	C	C	R	R
Рішення про введення в експлуатацію	A	R	C	I	C	C
Підсумковий звіт	I	A/R	C	I	I	I

Як видно з таблиці 3.5, керівник проєкту координує більшість робіт і відповідає за їх організацію, але не обов'язково виконує їх особисто. Матриця також допомагає виявити роботи без відповідального виконавця, надмірну концентрацію повноважень або дублювання відповідальності.

Крім розподілу відповідальності, необхідно визначити:

- порядок прийняття рішень;
- межі повноважень кожної ролі;
- процедуру ескалації проблем;

- правила комунікації;
- порядок заміни учасників;
- потребу в навчанні;
- умови залучення зовнішніх фахівців;
- порядок вивільнення ресурсів після завершення робіт.

Інформація про ролі та відповідальних осіб може бути оформлена у настанові з управління проєктом і матриці зацікавлених сторін. Для цього доцільно використати шаблони **Project Handbook** та **Project Stakeholder Matrix** із комплексу артефактів PM² [7].

3.5 Планування управління конфігураціями та якістю

Управління конфігураціями забезпечує ідентифікацію, зберігання, версіонування, зміну і простежуваність проєктних документів та результатів. Його мета полягає в тому, щоб команда використовувала правильні й погоджені версії матеріалів, а несанкціоновані зміни не потрапляли до кінцевого продукту.

Конфігураційними одиницями ІТ-проєкту можуть бути:

- специфікація вимог;
- архітектурна документація;
- вихідний програмний код;
- схема бази даних;
- конфігураційні файли;
- сценарії розгортання;
- тестові сценарії;
- набори даних;
- бібліотеки й залежності;
- інструкції користувача;
- інструкції адміністратора;
- склад програмного випуску.

Для кожної конфігураційної одиниці визначаються її ідентифікатор, назва, версія, власник, місце зберігання, поточний статус і порядок внесення змін.

Таблиця 3.6 – Приклад реєстру конфігураційних одиниць

ID	Конфігураційна одиниця	Поточна версія	Відповідальний	Місце зберігання	Статус
CI-01	Специфікація вимог	1.2	Бізнес-	Репозиторій	Затверджено

			аналітик	документів	
CI-02	Архітектура системи	1.0	Архітектор	Репозиторій документів	Затверджено
CI-03	Вихідний код серверної частини	0.8	Провідний розробник	Git-репозиторій	У розробленні
CI-04	Схема бази даних	0.6	Розробник бази даних	Git-репозиторій	У розробленні
CI-05	Набір інтеграційних тестів	0.4	Тестувальник	Система тестування	На перевірці
CI-06	Сценарій розгортання	0.3	DevOps-інженер	Git-репозиторій	Чернетка
CI-07	Інструкція користувача	0.2	Технічний автор	Репозиторій документів	Чернетка

Процедура управління конфігураціями повинна визначати:

- правила найменування файлів, документів і версій;
- перелік конфігураційних одиниць;
- структуру репозиторіїв і папок;
- права доступу;
- порядок створення і погодження нових версій;
- правила формування програмних випусків;
- порядок резервного копіювання;
- строки зберігання;
- порядок архівування або видалення даних;
- зв'язок між запитами на зміну і зміненими об'єктами.

Управління конфігураціями тісно пов'язане з управлінням якістю. Якщо команда не контролює версії вимог, коду, тестів і документації, вона не може гарантувати, що перевіряється та передається саме погоджена версія результату.

План управління якістю визначає вимоги до якості, підхід до її забезпечення, процедури контролю, показники та відповідальних осіб. Необхідно розрізняти забезпечення і контроль якості.

Забезпечення якості спрямоване на перевірку правильності процесів, які використовуються для створення результатів. До нього належать аудити, перевірка дотримання стандартів, аналіз процесів розроблення, перевірка правил роботи з кодом і документацією.

Контроль якості спрямований на перевірку самих результатів. До нього належать тестування, інспекції, аналіз дефектів, перевірка продуктивності, безпеки та відповідності критеріям приймання.

Планування якості має охоплювати:

- характеристики якості;
- вимоги й стандарти;
- показники та допустимі значення;
- методи перевірки;
- строки проведення перевірок;
- відповідальних осіб;
- порядок реєстрації дефектів;
- правила виправлення і повторної перевірки;
- критерії приймання результатів.

Таблиця 3.7 – Приклад показників якості програмного продукту

Характеристика	Показник	Цільове значення	Метод перевірки	Відповідальний
Функціональна придатність	Частка реалізованих і перевірених вимог	100 % критичних вимог	Матриця простежуваності, приймальні тести	Бізнес-аналітик, тестувальник
Надійність	Частка успішно виконаних тестів	Не менше 95 %	Автоматизоване тестування	Тестувальник
Продуктивність	Час відповіді системи	До 2 секунд для 95 % запитів	Навантажувальне тестування	Тестувальник
Масштабованість	Кількість одночасних користувачів	Не менше 500	Навантажувальне тестування	DevOps-інженер
Якість коду	Частка коду, що пройшла перевірку	100 % змін у головній гілці	Code review	Провідний розробник
Тестове покриття	Покриття модульними тестами	Не менше 80 % критичних модулів	Звіт системи тестування	Розробник

Безпека	Кількість критичних вразливостей	0	Сканування та тестування безпеки	Фахівець із безпеки
Зручність використання	Частка користувачів, які виконали основний сценарій без допомоги	Не менше 90 %	Користувацьке тестування	UX-фахівець
Документація	Повнота обов'язкових документів	100 %	Експертна перевірка	Керівник проєкту

Показники якості мають бути вимірюваними, пов'язаними з вимогами та включеними до проєктного розкладу. Тестування, перевірки, аудити й виправлення дефектів потребують часу і ресурсів, тому їх не можна розглядати як додаткові роботи, що виконуються лише за наявності вільного часу.

Для оформлення цих процесів можна використати шаблони **Quality Management Plan, Quality Review Checklist, Deliverables Acceptance Plan i Deliverables Acceptance Checklist** із комплексу артефактів PM² [7].

3.6 Формування базового розкладу проєкту

Розклад є часовою моделлю виконання проєкту. Він визначає роботи, їхню тривалість, залежності, ресурси, дати початку і завершення та основні контрольні точки.

Формування розкладу починається після створення WBS. Для кожного пакета робіт визначаються конкретні дії, необхідні для отримання результату.

Послідовність розроблення розкладу включає:

1. Визначення переліку робіт.
2. Установлення логічних залежностей.
3. Оцінювання необхідних ресурсів.
4. Оцінювання трудомісткості та тривалості.
5. Визначення календарних обмежень.
6. Розрахунок дат початку і завершення.
7. Визначення віх і критичного шляху.
8. Перевірку доступності ресурсів.

9. Урахування робіт із тестування, якості, приймання та управління ризиками.

10. Погодження і затвердження базового розкладу.

Перелік робіт повинен бути безпосередньо пов'язаний із пакетами WBS. Для кожної роботи доцільно вказати:

- унікальний ідентифікатор;
- код WBS;
- назву й опис;
- попередників;
- тривалість;
- трудомісткість;
- призначені ресурси;
- припущення та обмеження;
- планові дати.

Необхідно розрізняти тривалість і трудомісткість. Трудомісткість показує кількість робочого часу, необхідного для виконання завдання. Тривалість визначає календарний період між початком і завершенням.

Наприклад, завдання може мати трудомісткість 16 людино-годин. Один працівник виконуватиме його два робочі дні, а два працівники за певних умов можуть виконати за один день. Водночас збільшення кількості працівників не завжди пропорційно скорочує тривалість. Деякі роботи мають технологічні або організаційні обмеження.

Для оцінювання тривалості використовуються:

- експертне оцінювання;
- аналогове оцінювання за попередніми проєктами;
- параметричне оцінювання;
- оцінювання знизу вгору;
- триточкове оцінювання;
- аналіз резервів.

Після оцінювання роботи розташовуються у логічній послідовності. Деякі з них можуть виконуватися паралельно, а інші починаються лише після завершення попередніх.

Таблиця 3.8 – Приклад переліку робіт для формування розкладу

ID	Робота	Тривалість	Попередник	Ресурс	Результат
A	Уточнення вимог	5 днів	—	Бізнес-аналітик	Погоджені вимоги

B	Проектування архітектури	4 дні	A	Архітектор	Опис архітектури
C	Проектування інтерфейсу	4 дні	A	UX/UI-дизайнер	Макети інтерфейсу
D	Розроблення серверної частини	10 днів	B	Backend-розробник	Серверний модуль
E	Розроблення клієнтської частини	10 днів	B, C	Frontend-розробник	Клієнтський модуль
F	Інтеграційне тестування	5 днів	D, E	Тестувальник	Звіт про тестування
G	Виправлення дефектів	4 дні	F	Розробники	Виправлена версія
H	Приймальне тестування	3 дні	G	Замовник, тестувальник	Акт приймання
I	Розгортання системи	2 дні	H	DevOps-інженер	Робоча система

У таблиці 3.8 роботи B і C можуть виконуватися паралельно після погодження вимог. Інтеграційне тестування починається лише після завершення серверної і клієнтської частин. Такі залежності використовуються для створення мережевої моделі та визначення критичного шляху.

Віхи є контрольними подіями з нульовою тривалістю. Прикладами можуть бути:

- вимоги погоджено;
- архітектуру затверджено;
- прототип продемонстровано;
- тестування завершено;
- продукт прийнято;
- систему введено в експлуатацію.

Розклад може бути представлений як таблиця, мережевий графік, план віх або діаграма Ганта. Діаграма Ганта відображає роботи у вигляді горизонтальних смуг на часовій шкалі та дозволяє показати їхню тривалість, послідовність і залежності.

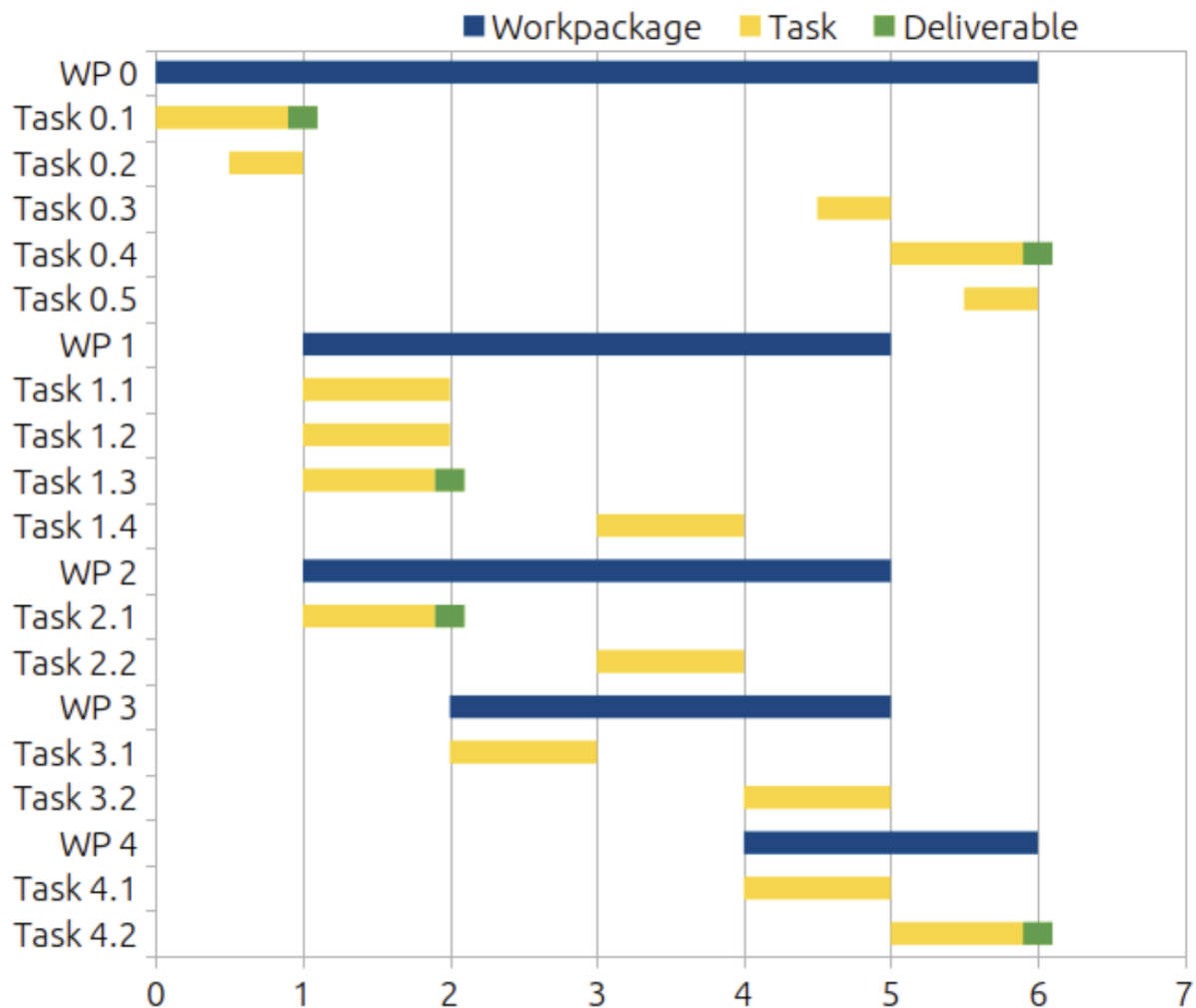


Рисунок 3.6 – Приклад календарного розкладу у вигляді діаграми Ганта

Під час підготовки розкладу необхідно перевірити доступність ресурсів. Якщо один фахівець одночасно призначений на декілька повноцінних робіт, виникає перевантаження. Для його усунення можна змінити послідовність робіт, перенести строки, залучити додатковий ресурс або переглянути зміст.

Після погодження розклад затверджується як базовий. Базовий розклад є офіційною версією, відносно якої надалі вимірюється фактичний прогрес.

Таблиця 3.9 – Приклад базового розкладу проєкту

Віха або результат	Планова дата	Критерій досягнення	Відповідальний
Вимоги погоджено	15 вересня	Специфікацію затверджено замовником	Бізнес-аналітик

Архітектуру затверджено	25 вересня	Архітектурне рішення пройшло перевірку	Архітектор
Прототип підготовлено	10 жовтня	Основні сценарії доступні для демонстрації	Команда розроблення
Розроблення завершено	10 листопада	Усі критичні функції реалізовано	Провідний розробник
Тестування завершено	25 листопада	Критичні дефекти відсутні	Тестувальник
Продукт прийнято	5 грудня	Підписано документ про приймання	Замовник
Систему введено в експлуатацію	10 грудня	Робоча версія розгорнута у виробничому середовищі	DevOps-інженер

Базовий розклад не означає, що план не може змінюватися. Однак кожна зміна, яка впливає на затверджені строки, повинна бути обґрунтована, проаналізована і схвалена відповідно до процедури управління змінами. Початкова версія базового розкладу зберігається для аналізу відхилень і оцінювання якості планування.

Для створення робочого плану й розкладу можна використати шаблон **Project Work Plan** із комплекту артефактів PM² [7]. Він повинен об'єднувати WBS, оцінки трудомісткості й вартості та календарний розклад.

Висновки

Планування перетворює загальне визначення проєкту на погоджену модель його виконання. На цій фазі уточнюються зміст, результати, вимоги, склад робіт, відповідальні учасники, ресурси, показники якості та строки.

Основою планування змісту є чітке визначення того, що входить до проєкту, а що перебуває поза його межами. Результати проєкту декомпонуються за допомогою WBS до рівня пакетів робіт, які можна оцінити, призначити виконавцям і включити до розкладу.

Розподіл відповідальності доцільно фіксувати за допомогою матриці RACI. Вона дає змогу встановити виконавців, відповідальних за остаточне рішення, консультантів та осіб, яких потрібно інформувати.

Управління конфігураціями забезпечує використання правильних версій вимог, програмного коду, документації та інших результатів. План

управління якістю визначає вимоги, показники, процедури забезпечення і контролю якості та відповідальних осіб.

Розклад формується на основі WBS, переліку робіт, залежностей, оцінок тривалості та доступності ресурсів. Після погодження він стає базовим і використовується для моніторингу фактичного прогресу.

Якісне планування створює спільне розуміння порядку реалізації проєкту, зменшує невизначеність, забезпечує узгодженість дій команди та формує основу для виконання, моніторингу і контролю ІТ-проєкту.

Лекція 4 Управління ризиками проєкту

Тема: Управління ризиками проєкту

Мета: Сформувати у студентів системне розуміння процесу управління ризиками IT-проєкту, ознайомити з методами їх ідентифікації, якісного і кількісного аналізу, планування реагування, а також моніторингу та контролю ризиків протягом життєвого циклу проєкту.

План лекції

1. Основні поняття та принципи управління проєктними ризиками.
2. Планування управління ризиками.
3. Ідентифікація та класифікація ризиків.
4. Якісний і кількісний аналіз ризиків.
5. Планування заходів реагування на ризики.
6. Головні ризики програмних проєктів і способи реагування на них.
7. Управління проєктом, спрямоване на зниження ризиків.
8. Моніторинг і контроль ризиків.
9. Висновки.

4.1 Основні поняття та принципи управління проєктними ризиками

Кожен проєкт реалізується в умовах невизначеності. На початку проєкту неможливо точно передбачити всі майбутні події, зміни вимог, технічні складнощі, доступність ресурсів і поведінку зацікавлених сторін. Частина таких невизначеностей може вплинути на досягнення цілей проєкту.

Ризик проєкту — це невизначена подія або умова, яка в разі настання позитивно або негативно впливає на одну чи декілька цілей проєкту: зміст, строки, вартість, якість, ресурси або очікувані результати.

Негативні ризики називають загрозами. Вони можуть спричинити затримки, перевищення бюджету, погіршення якості або невиконання вимог. Позитивні ризики називають можливостями. Їх реалізація може скоротити строки, зменшити витрати, підвищити якість або створити додаткову цінність.

Ризик стосується майбутньої події, яка ще не відбулася. Якщо подія вже настала і потребує негайного вирішення, вона розглядається як

проблема. Наприклад, можливе звільнення ключового розробника є ризиком. Якщо розробник уже звільнився, це проблема, яку необхідно зареєструвати та вирішувати.

Таблиця 4.1 – Основні поняття управління ризиками

Поняття	Характеристика
Ризик	Невизначена майбутня подія або умова, яка може вплинути на цілі проекту
Загроза	Ризик із можливими негативними наслідками
Можливість	Ризик із можливими позитивними наслідками
Проблема	Подія, яка вже настала і потребує вирішення
Тригер ризику	Ознака або подія, що свідчить про наближення чи настання ризику
Власник ризику	Особа, відповідальна за спостереження та управління ризиком
Залишковий ризик	Ризик, який залишається після реалізації заходів реагування
Вторинний ризик	Новий ризик, що виникає внаслідок реалізації заходів реагування

Ризик доцільно описувати за схемою «причина — ризикова подія — наслідок». Наприклад: «Через недостатній досвід роботи команди з новою технологією може виникнути значна кількість технічних помилок, що призведе до затримки інтеграції та збільшення витрат на виправлення».

Управління ризиками є систематичним і безперервним процесом ідентифікації, аналізу, планування реагування, контролю та документування ризиків. Його завдання полягає не в усуненні всієї невизначеності, а у зменшенні ймовірності та впливу загроз і збільшенні ймовірності реалізації можливостей.

Основними принципами управління ризиками є:

- раннє виявлення ризиків;
- систематичне повторне оцінювання;
- узгодження дій із цілями проекту;
- призначення відповідального за кожен значущий ризик;
- визначення конкретних заходів реагування;
- своєчасне інформування зацікавлених сторін;
- включення заходів реагування до робочого плану;
- документування прийнятих рішень;
- використання досвіду попередніх проєктів.

Загальну організацію процесу управління ризиками показано на рисунку 4.1.

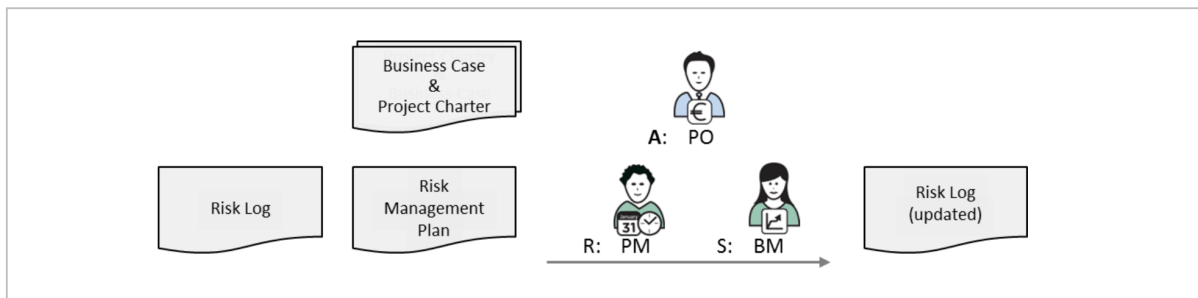


Рисунок 4.1 – Управління ризиками: вхідні дані, результати та основні ролі

4.2 Планування управління ризиками

Планування управління ризиками визначає, як саме команда працюватиме з ризиками протягом життєвого циклу проєкту. Результатом цього процесу є план управління ризиками.

План не є переліком конкретних ризиків. Він установлює правила, методи, шкали, відповідальність і періодичність виконання процесів ризик-менеджменту. Самі ризики фіксуються у реєстрі ризиків.

План управління ризиками повинен визначати:

- методи ідентифікації ризиків;
- категорії ризиків;
- шкали ймовірності та впливу;
- порядок визначення загального рівня ризику;
- допустимий рівень ризику;
- критерії пріоритизації;
- стратегії реагування;
- ролі й відповідальність;
- порядок ескалації критичних ризиків;
- періодичність перегляду реєстру;
- правила звітування;
- резерви часу і бюджету;
- структуру реєстру ризиків.

Схильність до ризику характеризує загальний рівень ризику, який організація готова прийняти для досягнення своїх цілей. Толерантність до ризику визначає допустимий діапазон відхилення, а порогове значення

встановлює конкретну межу, після перевищення якої потрібне реагування або ескалація.

Наприклад, організація може погодитися з незначними затримками окремих некритичних робіт, але вимагати негайної ескалації будь-якого ризику, який може затримати випуск продукту більше ніж на два тижні.

Керівник проєкту готує план управління ризиками, а спонсор або власник проєкту його схвалює. До розроблення плану залучаються члени команди, представники замовника, технічні експерти та інші зацікавлені сторони.

Для документування процесу доцільно використовувати шаблони **Risk Management Plan** та **Risk Log** із комплекту артефактів PM² [7].

4.3 Ідентифікація та класифікація ризиків

Ідентифікація ризиків полягає у визначенні подій і умов, які можуть вплинути на проєкт, та документуванні їхніх характеристик. Цей процес починається під час ініціації, деталізується під час планування і повторюється протягом усього проєкту.

Джерелами інформації для ідентифікації є:

- бізнес-кейс і статут;
- опис змісту та вимоги;
- WBS і перелік робіт;
- календарний розклад;
- ресурсний і фінансовий плани;
- припущення та обмеження;
- договори з постачальниками;
- технічна й архітектурна документація;
- досвід попередніх проєктів;
- інформація від зацікавлених сторін.

Для ідентифікації ризиків використовують експертні обговорення, мозковий штурм, інтерв'ю, контрольні списки, аналіз документів, аналіз припущень, аналіз першопричин, SWOT-аналіз, PESTLE-аналіз і спеціальні наради з ризиків.

Ідентифікація повинна охоплювати не лише загрози, а й можливості. Наприклад, поява нового програмного інструменту може створити ризики несумісності, але водночас дати можливість автоматизувати частину робіт і скоротити строк розроблення.

Класифікація допомагає забезпечити системність і не пропустити важливі групи ризиків. Для цього може використовуватися ієрархічна структура ризиків.

Основні категорії ризиків ІТ-проєкту:

- бізнесові та стратегічні;
- пов'язані з вимогами і змістом;
- технічні й архітектурні;
- організаційні та кадрові;
- календарні й фінансові;
- пов'язані з якістю;
- безпекові;
- контрактні та постачальницькі;
- нормативні й правові;
- зовнішні.

Після ідентифікації ризик записується до реєстру. Реєстр є централізованим джерелом інформації про ризики, їхні оцінки, власників, заходи реагування та поточний статус.

Приклад структури реєстру показано на рисунку 4.2.

ID	Description	Trigger	Probability	Severity	Response	Owner
R01						
R02						
R03						
R04						
R05						
R06						

Рисунок 4.2 – Приклад реєстру ризиків проєкту

У реєстрі доцільно фіксувати:

- унікальний ідентифікатор;
- категорію;
- назву й опис;
- причину та можливі наслідки;
- пов'язаний пакет робіт;
- тригер;
- ймовірність;
- вплив;
- загальний рівень;

- власника;
- стратегію реагування;
- конкретні заходи;
- відповідального за виконання заходів;
- строк виконання;
- поточний статус.

Таблиця 4.2 – Скорочений приклад реєстру ризиків ІТ-проєкту

ID	Ризик	Ймовірність	Вплив	Рівень	Стратегія	Власник
R-01	Зміна ключових вимог після початку розроблення	4	4	16	Зниження	Бізнес-аналітик
R-02	Недоступність зовнішнього API у визначений строк	3	5	15	Зниження/ передавання	Технічний керівник
R-03	Звільнення ключового розробника	2	5	10	Зниження	Керівник проєкту
R-04	Виявлення критичної вразливості перед випуском	3	5	15	Уникнення/з ниження	Фахівець із безпеки

Реєстр повинен регулярно оновлюватися. Нові ризики можуть виникнути в будь-який момент, а оцінки вже відомих ризиків можуть змінитися.

4.4 Якісний і кількісний аналіз ризиків

Після ідентифікації ризики необхідно проаналізувати і визначити їхню пріоритетність. Не всі ризики потребують однакового обсягу уваги. Значні ресурси насамперед спрямовуються на ризики з високою ймовірністю або серйозними наслідками.

Якісний аналіз передбачає оцінювання ймовірності виникнення ризику та сили його впливу. Для цього часто використовується п'ятирівнева шкала:

1 — дуже низький;

- 2 — низький;
 3 — середній;
 4 — високий;
 5 — дуже високий.

Загальний рівень ризику може визначатися за формулою:

$$RL = L * times I,$$

де (RL) — рівень ризику, (L) — оцінка ймовірності, (I) — оцінка впливу.

Наприклад, ризик із ймовірністю 4 та впливом 5 отримує рівень 20 і потребує першочергового реагування. Ризик із ймовірністю 1 та впливом 2 отримує рівень 2 і може лише спостерігатися.

Для наочного представлення оцінок використовується матриця ймовірності та впливу, наведена на рисунку 4.3.

Risk Assessment Matrix		Probability (Expected likelihood or frequency)				
		Frequent: Continuous, regular, or inevitable	Likely: Several or numerous occurrences	Occasional: Intermittent chance of happening	Seldom: Infrequently happens	Unlikely: Could possibly occur but improbable
Severity (Impact or consequence)		A	B	C	D	E
Catastrophic: company readiness eliminated, unacceptable loss or damage	I	EH	EH	H	H	M
Critical: company ability to operate significantly reduced, severe injury, loss, or damage	II	EH	H	H	M	L
Moderate: somewhat reduced ability to operate, minor injury, loss, or damage	III	H	M	M	L	L
Negligible: little or no impact on organization, minimal injury, loss, or damage	IV	M	L	L	L	L
Legend: EH=Extremely High Risk, H=High Risk, M=Medium Risk, L=Low Risk						

Рисунок 4.3 – Матриця оцінювання проєктних ризиків

Крім ймовірності та впливу, можуть враховуватися:

- терміновість;

- близькість очікуваної події;
- можливість раннього виявлення;
- складність управління;
- взаємозв'язок з іншими ризиками;
- вплив на критичні результати;
- ставлення ключових зацікавлених сторін.

Якісний аналіз значною мірою ґрунтується на експертному судженні, тому на оцінки можуть впливати суб'єктивні погляди. Для зменшення упередженості необхідно використовувати погоджені шкали, залучати декількох експертів і вимагати обґрунтування оцінок.

Кількісний аналіз застосовується переважно до найбільш значущих ризиків. Він дає змогу оцінити можливий числовий вплив ризиків на строки, вартість або інші показники проєкту.

Основними методами кількісного аналізу є:

- аналіз очікуваної грошової вартості;
- дерева рішень;
- триточкове оцінювання;
- аналіз чутливості;
- моделювання Монте-Карло.

Очікувана грошова вартість визначається за формулою:

$$EMV = P * V,$$

де (P) — ймовірність події, (V) — її грошовий наслідок.

Наприклад, якщо ймовірність додаткових витрат у розмірі 100 000 грн становить 30 %, очікувана грошова вартість ризику дорівнює:

$$EMV = 0,3 * 100,000 = 30,000 \text{ грн.}$$

Це значення може бути враховане під час формування резерву.

Моделювання Монте-Карло використовується у складних проєктах із великою кількістю взаємопов'язаних невизначеностей. Воно багаторазово моделює різні комбінації тривалості, вартості та ризикових подій і визначає ймовірність завершення проєкту в межах заданих показників.

Кількісний аналіз не завжди є необхідним. Для невеликих проєктів може бути достатньо якісного оцінювання. Глибина аналізу повинна відповідати складності, вартості й загальному рівню невизначеності проєкту.

4.5 Планування заходів реагування на ризики

Після оцінювання ризиків команда визначає стратегії та конкретні дії реагування. Обрана стратегія повинна відповідати типу ризику, його рівню, доступним ресурсам і допустимому рівню невизначеності.

Таблиця 4.3 – Стратегії реагування на ризики

Тип ризику	Стратегія	Сутність
Загроза	Уникнення	Зміна плану для усунення причини або можливості виникнення ризику
Загроза	Зниження	Зменшення ймовірності або негативного впливу
Загроза	Передавання	Передавання частини відповідальності іншій стороні
Загроза	Прийняття	Усвідомлене рішення не виконувати попереджувальних дій
Загроза	Ескалація	Передавання ризику на рівень, який має необхідні повноваження
Можливість	Використання	Дії, спрямовані на гарантовану реалізацію можливості
Можливість	Посилення	Збільшення ймовірності або позитивного впливу
Можливість	Спільне використання	Залучення партнера, здатного допомогти реалізувати можливість
Можливість	Прийняття	Використання можливості в разі її появи без активного стимулювання
Можливість	Ескалація	Передавання можливості на вищий рівень управління

Для кожного значущого ризику необхідно визначити:

- обрану стратегію;
- конкретні попереджувальні дії;
- власника ризику;
- відповідального за виконання заходів;
- необхідні ресурси;
- строк виконання;
- тригер;
- резервний план;
- умови ескалації.

Заходи реагування, які потребують часу і ресурсів, повинні бути включені до WBS, календарного розкладу та бюджету. Недостатньо лише записати, що ризик необхідно контролювати.

Для критичних ризиків розробляється резервний план, який виконується після настання визначеного тригера. Якщо резервний план виявляється недостатнім, можуть застосовуватися додаткові обхідні рішення.

Після реалізації реагування необхідно оцінити залишковий ризик і перевірити, чи не створили заплановані дії нових вторинних ризиків.

4.6 Головні ризики програмних проєктів і способи реагування на них

Програмні проєкти характеризуються мінливістю вимог, складністю технологій, високою залежністю від компетентності команди та необхідністю інтеграції великої кількості компонентів.

Таблиця 4.4 – Типові ризики програмних проєктів

Ризик	Можливі наслідки	Основні способи реагування
Нечіткі або мінливі вимоги	Перероблення, розширення змісту, затримки	Прототипування, раннє погодження, матриця простежуваності, формальне управління змінами
Нереалістичні оцінки	Зрив строків і перевищення бюджету	Експертне та триточкове оцінювання, декомпозиція, резерви
Використання неперевіреної технології	Технічні помилки, низька продуктивність	Технічний прототип, дослідницьке завдання, залучення експертів
Залежність від ключового фахівця	Зупинення критичних робіт	Обмін знаннями, документація, дублювання компетентностей
Проблеми інтеграції	Несумісність компонентів і затримки	Рання інтеграція, контрактне тестування, тестове середовище
Низька якість програмного коду	Велика кількість дефектів і перероблень	Code review, автоматизовані тести, статичний аналіз
Вразливості безпеки	Втрата даних, відкладення випуску	Безпекові вимоги, аналіз загроз, сканування і тестування
Залежність від зовнішнього постачальника	Затримка або зміна умов надання послуг	SLA, резервний постачальник, контрольні точки, передавання ризику
Недостатнє залучення користувачів	Невідповідність продукту потребам	Регулярні демонстрації, користувацьке і приймальне

		тестування
--	--	------------

Наведені ризики не є універсальним переліком. Для кожного проєкту необхідно враховувати його предметну область, архітектуру, технології, організаційне середовище та зацікавлені сторони.

4.7 Управління проєктом, спрямоване на зниження ризиків

Ризик-орієнтоване управління передбачає, що найбільш невизначені та критичні елементи проєкту опрацьовуються якомога раніше. Відкладення складних технічних або організаційних питань збільшує вартість їх вирішення.

До практик зниження ризиків належать:

- раннє уточнення критичних вимог;
- створення прототипів;
- перевірка архітектурних рішень;
- короткі ітерації;
- поступове постачання результатів;
- рання та регулярна інтеграція;
- автоматизоване тестування;
- аналіз програмного коду;
- регулярні демонстрації замовнику;
- резервування часу і бюджету;
- документування рішень та обмін знаннями;
- використання досвіду попередніх проєктів.

План проєкту повинен враховувати ризики під час визначення послідовності робіт. Наприклад, інтеграцію з критичною зовнішньою системою доцільно перевірити за допомогою прототипу на початку, а не залишати до завершення розроблення.

У прогнозованому проєкті зниження ризиків забезпечується детальним аналізом вимог, резервами, контрольними точками та формальним управлінням змінами. В адаптивному проєкті ризики зменшуються завдяки коротким ітераціям, постійному зворотному зв'язку, пріоритизації та поступовому уточненню продукту.

Вартість реагування повинна бути пропорційною можливим наслідкам. Витрачати значні ресурси на малоімовірний ризик із незначним впливом недоцільно. Водночас ризик із низькою ймовірністю, але катастрофічним впливом може потребувати окремого резервного плану.

4.8 Моніторинг і контроль ризиків

Моніторинг ризиків здійснюється протягом усього життєвого циклу проєкту. Його мета полягає у відстеженні відомих ризиків, виявленні нових, перевірці виконання заходів реагування та оцінюванні їхньої результативності.

У процесі моніторингу необхідно:

- переглядати реєстр ризиків;
- відстежувати тригери;
- повторно оцінювати ймовірність і вплив;
- контролювати виконання заходів;
- визначати нові ризики;
- аналізувати залишкові та вторинні ризики;
- закривати неактуальні ризики;
- контролювати використання резервів;
- ескалувати критичні ризики;
- повідомляти зацікавлені сторони;
- оновлювати робочий план.

Періодичність перегляду залежить від складності та динаміки проєкту. Ризики можуть розглядатися під час щотижневих нарад, статусних зустрічей, завершення ітерацій або проходження контрольних точок.

Для контролю можуть використовуватися такі показники:

- кількість відкритих ризиків;
- кількість ризиків високого рівня;
- загальна експозиція проєкту;
- кількість прострочених заходів реагування;
- частка ризиків без призначеного власника;
- використання резервів;
- зміна рівня ризиків у часі;
- кількість ризиків, що перетворилися на проблеми.

Якщо ризик настає, його наслідки повинні бути зареєстровані як проблема. Команда реалізує заплановані заходи, а якщо вони недостатні — визначає обхідне рішення. За потреби запускається процедура управління змінами.

Моніторинг повинен також визначати, чи були заходи реагування ефективними. Якщо після їх виконання рівень ризику залишається

неприйнятним, необхідно переглянути стратегію, призначити додаткові дії або передати ризик на вищий рівень управління.

Усі суттєві зміни оцінок, статусів, заходів і рішень фіксуються в реєстрі. Після завершення проєкту інформація про реалізовані ризики та ефективність реагування включається до засвоєних уроків.

Висновки

Управління ризиками є систематичним і безперервним процесом, спрямованим на зменшення негативного та збільшення позитивного впливу невизначеності на цілі проєкту.

Ризики необхідно відрізняти від проблем: ризик стосується майбутньої події, а проблема вже настала. Ризики можуть бути загрозами або можливостями та повинні описуватися через причину, подію і можливий наслідок.

План управління ризиками визначає правила, шкали, відповідальність, стратегії, періодичність перегляду і порядок ескалації. Виявлені ризики реєструються, класифікуються та закріплюються за відповідальними особами.

Якісний аналіз використовується для визначення пріоритетності за ймовірністю і впливом. Кількісний аналіз дає змогу оцінити числовий вплив значущих ризиків на вартість і строки.

Для загроз застосовуються уникнення, зниження, передавання, прийняття або ескалація. Для можливостей використовуються стратегії використання, посилення, спільної реалізації, прийняття або ескалації.

Ефективне управління ризиками передбачає ранню перевірку найбільш невизначених елементів, включення заходів реагування до плану та постійне оновлення реєстру. Такий підхід допомагає команді своєчасно приймати рішення і підвищує ймовірність успішного завершення ІТ-проєкту.

Лекція 5 Оцінювання трудомісткості й термінів розроблення

Тема: Оцінювання трудомісткості й термінів розроблення

Мета: Сформувати у студентів розуміння принципів і методів оцінювання трудомісткості та тривалості розроблення програмного забезпечення, навчити враховувати невизначеність оцінок, аналізувати наслідки необґрунтовано скорочених термінів і застосовувати відповідні підходи до планування програмних проєктів.

План лекції

1. Особливості оцінювання трудомісткості програмних проєктів.
2. Оцінка як імовірнісне твердження.
3. Негативні наслідки «агресивного» розкладу.
4. Прагматичний підхід до оцінювання.
5. Метод функціональних точок.
6. Оцінювання багатокomпонентного програмного продукту.
7. Оцінювання тривалості проєкту.
8. Висновки.

5.1. Особливості оцінювання трудомісткості програмних проєктів

Оцінювання трудомісткості й термінів є одним із найскладніших завдань управління програмним проєктом. На відміну від матеріального виробництва, програмне забезпечення не має фізично вимірюваного обсягу. Кількість рядків програмного коду, модулів або файлів не завжди характеризує реальний обсяг роботи та складність майбутнього продукту.

Оцінювання виконується для того, щоб визначити, скільки праці, часу, людей і фінансових ресурсів необхідно для створення програмного продукту. Отримані оцінки використовуються під час формування бюджету, календарного плану, складу команди, договірних зобов'язань і базового розкладу проєкту.

Точність оцінювання залежить від повноти вимог, складності продукту, досвіду команди, застосовуваних технологій, кількості інтеграцій, вимог до якості та безпеки, а також від рівня невизначеності. На ранніх етапах проєкту відомостей про майбутній продукт недостатньо, тому початкові оцінки мають орієнтовний характер. У міру уточнення вимог, архітектури й складу робіт вони повинні переглядатися.

Під час оцінювання необхідно розрізняти трудомісткість, тривалість, чисельність команди та вартість. Відмінності між цими поняттями наведено в таблиці 5.1.

Таблиця 5.1 – Основні величини, що використовуються під час оцінювання

Величина	Зміст	Типові одиниці вимірювання
Трудомісткість	Обсяг праці, необхідний для виконання роботи	людино-години, людино-дні, людино-місяці
Тривалість	Календарний час від початку до завершення роботи	години, дні, тижні, місяці
Чисельність команди	Кількість виконавців, залучених до роботи	особи або еквівалент повної зайнятості
Вартість	Грошові витрати на виконання роботи	гривні, євро, долари
Продуктивність	Обсяг результату, створений за одиницю праці	функціональні точки на людино-місяць, завдання на ітерацію

Трудомісткість і тривалість не є взаємозамінними поняттями. Якщо одна людина виконує роботу протягом десяти робочих днів, трудомісткість становить десять людино-днів, а тривалість — десять днів. Проте залучення двох людей не завжди скорочує тривалість до п'яти днів. Частина робіт не може виконуватися паралельно, а додаткові виконавці потребують координації, обміну інформацією та узгодження результатів.

Різницю між трудомісткістю і тривалістю доцільно проілюструвати за допомогою рисунка 5.1.

Activity	Duration	Effort
A) Order the paint and supplies	1 week	1 hour
B) Remove the furniture and prepare the room	2 hours	2 hours
C) Paint the room	6 hours	6 hours
D) Let the paint dry	2 days	0 hours
E) Restore the furniture back to the room	2 hours	2 hours

Рисунок 5.1 – Порівняння тривалості та трудомісткості роботи

Наприклад, очікування надання доступу до зовнішнього програмного інтерфейсу може тривати п'ять днів, але вимагати лише однієї години праці для оформлення заявки. Натомість розроблення програмного модуля може потребувати п'яти людино-днів безперервної роботи. Тому під час планування необхідно окремо оцінювати активну працю виконавців і календарне очікування.

Основою обґрунтованого оцінювання є структура декомпозиції робіт. Проектний результат поділяють на компоненти, робочі пакети та окремі операції, які можна призначити конкретним виконавцям і для яких можна визначити необхідні ресурси. Чим точніше визначено склад робіт, тим надійнішою буде оцінка.

До оцінки необхідно включати не лише програмування, а й аналіз вимог, проєктування, створення архітектури, тестування, виправлення дефектів, налаштування середовищ, документування, розгортання, навчання користувачів, управління проєктом і комунікації із зацікавленими сторонами.

5.2. Оцінка як імовірнісне твердження

Оцінка не є гарантією точного результату. Вона є обґрунтованим припущенням щодо майбутнього, сформованим на основі доступної інформації, досвіду, статистики та прийнятих припущень.

Формулювання «робота триватиме десять днів» створює враження абсолютної точності. Коректніше зазначити, що робота, наприклад, триватиме від восьми до дванадцяти днів із визначеним рівнем впевненості за умови незмінності вимог і доступності необхідних ресурсів.

Повна оцінка повинна містити:

- очікуване значення або діапазон значень;
- рівень упевненості;
- перелік припущень;
- відомі обмеження;
- основні джерела невизначеності;
- дату виконання оцінювання;
- метод, за допомогою якого отримано оцінку.

На ранніх етапах проєкту діапазон оцінки є широким. Після уточнення вимог, декомпозиції продукту, вибору технологій і проведення

експериментальної розробки невизначеність зменшується. Тому оцінка повинна уточнюватися протягом усього життєвого циклу проєкту.

Одним зі способів урахування невизначеності є триточкове оцінювання. Для кожної роботи визначають:

- оптимістичну оцінку (O) — мінімально можливу тривалість за сприятливих умов;
- найбільш імовірну оцінку (M) — тривалість за нормального перебігу робіт;
- песимістичну оцінку (P) — тривалість за несприятливих, але реалістичних умов.

Очікувану тривалість можна розрахувати за формулою:

$$T_e = (O + 4M + P) / 6,$$

де T_e — очікувана тривалість роботи.

Мірою розкиду оцінки є дисперсія:

$$\sigma^2 = \left(\frac{P - O}{6} \right)^2$$

Триточкове оцінювання змушує команду аналізувати не лише найбільш імовірний сценарій, а й чинники, які можуть прискорити або затримати виконання роботи. Воно особливо корисне для нових технологій, складних інтеграцій і робіт, щодо яких відсутня достатня історична інформація.

Необхідно також відрізняти оцінку, цільовий термін і зобов'язання. Оцінка відображає очікуваний обсяг праці або часу. Цільовий термін визначає бажану дату завершення. Зобов'язання є погодженим рішенням щодо того, що команда має поставити до певної дати. Підміна оцінки бажаною датою призводить до необґрунтованого планування.

5.3. Негативні наслідки «агресивного» розкладу

«Агресивним» називають розклад, у якому терміни встановлено істотно коротшими за обґрунтовану оцінку без відповідного скорочення змісту, зміни способу виконання робіт або надання додаткових ресурсів.

Причиною такого розкладу може бути бажання випередити конкурентів, виконати контрактне зобов'язання, представити продукт до визначеної події або зменшити витрати. Проте сам факт установлення ранньої дати завершення не зменшує реальної трудомісткості проєкту.

Наслідками необґрунтовано скороченого розкладу можуть бути:

- постійна понаднормова робота та професійне вигорання;

- зниження якості програмного продукту;
- скорочення обсягів тестування;
- відмова від документування й аналізу;
- накопичення технічного боргу;
- збільшення кількості дефектів і повторних робіт;
- приховування проблем і викривлення звітності;
- втрата передбачуваності процесу розроблення;
- зростання загальної вартості проєкту;
- погіршення взаємодії із замовником і всередині команди.

За умов постійного дефіциту часу команда починає відкладати роботи, результат яких не одразу помітний: автоматизоване тестування, перевірку безпеки, рефакторинг, підготовку документації та аналіз архітектури. У короткостроковій перспективі це може створювати видимість прискорення, але в подальшому кількість помилок і повторної роботи збільшується.

Додавання людей до проєкту також не гарантує пропорційного скорочення тривалості. Нові учасники потребують ознайомлення з продуктом, технологіями та правилами роботи. Зі збільшенням команди зростає кількість комунікаційних зв'язків, витрати на координацію та ймовірність неузгоджених рішень.

Обґрунтоване скорочення розкладу може виконуватися двома основними способами. Перший спосіб — додавання ресурсів до робіт критичного шляху. Це може скоротити тривалість, але збільшує вартість. Другий спосіб — паралельне виконання робіт, які спочатку планувалися послідовно. Такий підхід може зекономити час, але підвищує ризик повторного виконання робіт.

Якщо розрахункова тривалість перевищує очікування замовника, необхідно не приховувати розбіжність, а переглянути зміст, пріоритети, ресурси, послідовність робіт і дату завершення. Рішення про скорочення термінів повинно супроводжуватися явним аналізом його вартості та ризиків.

5.4. Прагматичний підхід до оцінювання

Прагматичний підхід передбачає використання оцінки як інструмента прийняття рішень, а не як спроби точно передбачити майбутнє. Мета оцінювання полягає в тому, щоб надати достатньо

достовірну інформацію для планування ресурсів, бюджету, змісту та термінів проєкту.

Процес прагматичного оцінювання доцільно виконувати в такій послідовності:

1. Уточнити мету, межі та очікувані результати проєкту.
2. Сформувати структуру декомпозиції робіт.
3. Визначити функціональні та нефункціональні вимоги.
4. Залучити до оцінювання майбутніх виконавців і фахівців предметної області.
5. Вибрати метод оцінювання відповідно до доступної інформації.
6. Оцінити роботи у вигляді діапазонів.
7. Зафіксувати припущення, обмеження і виключення.
8. Урахувати ризики, резерви та роботи із забезпечення якості.
9. Перевірити оцінку за допомогою іншого методу.
10. Регулярно уточнювати оцінку на основі фактичних результатів.

Основні методи оцінювання узагальнено в таблиці 5.2.

Таблиця 5.2 – Основні методи оцінювання проєктних робіт

Метод	Сутність	Доцільність застосування
Експертне оцінювання	Оцінка формується фахівцями, які мають досвід виконання подібних робіт	Коли доступні компетентні експерти
Оцінювання за аналогією	Використовуються фактичні дані попередніх подібних проєктів	На ранніх етапах або для типових робіт
Параметричне оцінювання	Застосовується математична залежність між обсягом продукту та витратами праці	За наявності перевіреної моделі й історичних даних
Триточкове оцінювання	Використовуються оптимістична, найбільш імовірна та песимістична оцінки	За значної невизначеності
Оцінювання «знизу вгору»	Окремо оцінюються роботи нижнього рівня, після чого результати підсумовуються	Для формування детального базового плану
Оцінювання «згори вниз»	Загальна оцінка розподіляється між основними складовими проєкту	Для попереднього планування

На початку проєкту часто використовується оцінювання «згори вниз». Воно дає змогу швидко визначити орієнтовний масштаб проєкту, однак не забезпечує достатньої деталізації для контролю виконання. Після

формування структури декомпозиції робіт доцільно перейти до оцінювання «знизу вгору».

Співвідношення цих підходів показано на рисунку 5.2.

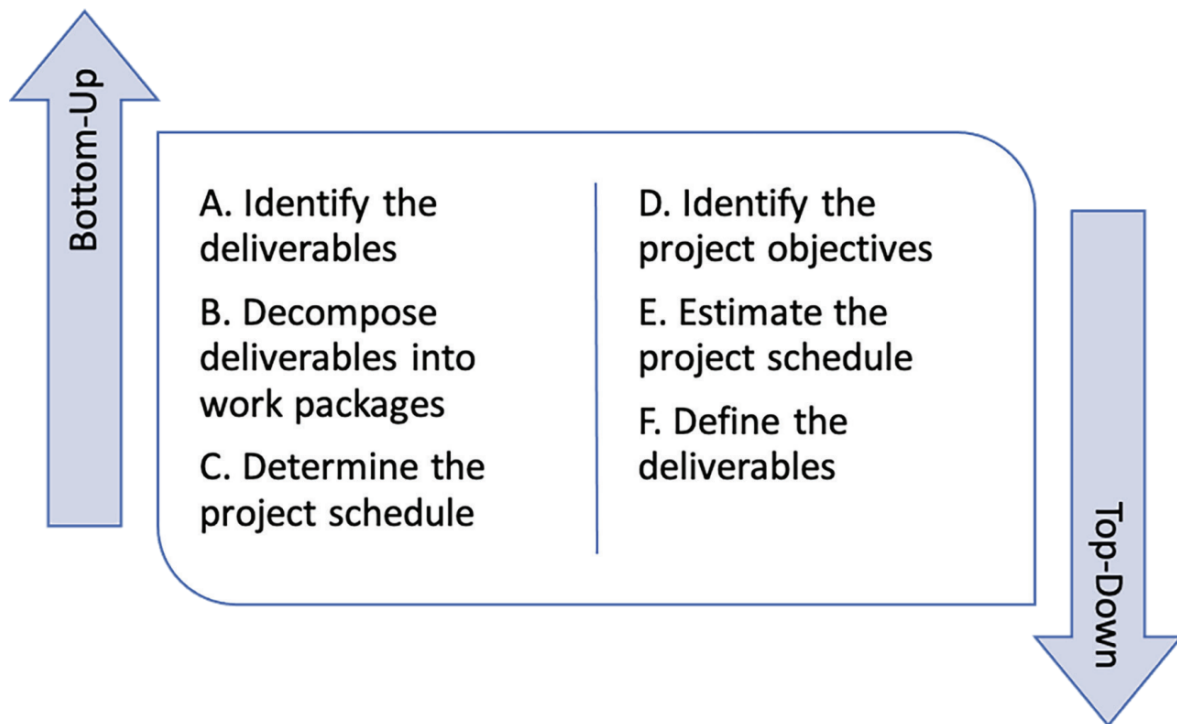


Рисунок 5.2 – Формування розкладу «згори вниз» і «знизу вгору»

Для підвищення надійності оцінки рекомендується комбінувати кілька методів. Наприклад, результат детального оцінювання робіт можна порівняти з даними попередніх проєктів або з параметричною моделлю. Значна розбіжність між результатами означає, що необхідно повторно перевірити склад робіт, припущення або коефіцієнти продуктивності.

Команда повинна зберігати фактичні дані про виконані роботи: початкову оцінку, реальну трудомісткість, причини відхилень, кількість дефектів і повторної роботи. Така інформація забезпечує поступове калібрування моделей оцінювання відповідно до особливостей організації.

5.5. Метод функціональних точок

Метод функціональних точок призначений для оцінювання функціонального розміру програмного продукту з погляду користувача. Він базується на функціях, які система повинна надавати, і не залежить

безпосередньо від мови програмування, кількості рядків коду або конкретної технології реалізації.

Перевагою методу є можливість виконувати оцінювання на основі вимог ще до завершення проєктування та програмування. Функціональний розмір може використовуватися для порівняння проєктів, визначення продуктивності команди й розрахунку трудомісткості.

Під час підрахунку визначають п'ять основних типів функцій:

- зовнішні введення (EI) — дані або команди, які надходять до системи та змінюють її внутрішній стан;
- зовнішні виведення (EO) — сформовані системою результати, що містять оброблені або розраховані дані;
- зовнішні запити (EQ) — операції отримання інформації без істотної зміни внутрішніх даних;
- внутрішні логічні файли (ILF) — логічні групи даних, які підтримуються системою;
- зовнішні інтерфейсні файли (EIF) — логічні групи даних, які використовуються системою, але підтримуються іншою системою.

Кожну функцію відносять до низького, середнього або високого рівня складності. Стандартні вагові коефіцієнти наведено в таблиці 5.3.

Таблиця 5.3 – Вагові коефіцієнти функціональних точок

Тип функції	Низька складність	Середня складність	Висока складність
Зовнішнє введення (EI)	3	4	6
Зовнішнє виведення (EO)	4	5	7
Зовнішній запит (EQ)	3	4	6
Внутрішній логічний файл (ILF)	7	10	15
Зовнішній інтерфейсний файл (EIF)	5	7	10

Кількість нескоригованих функціональних точок розраховується за формулою:

$$UFP = \sum_{i=1}^n N_i W_i$$

де N_i — кількість функцій відповідного типу і складності;

W_i — ваговий коефіцієнт;

UFP — кількість нескоригованих функціональних точок.

Розглянемо стислий приклад. Нехай інформаційна система містить вісім зовнішніх введень середньої складності, п'ять зовнішніх виведень, чотири зовнішні запити, три внутрішні логічні файли та два зовнішні інтерфейсні файли. Усі функції віднесено до середнього рівня складності.

Тоді функціональний розмір становитиме:

$$UFP = 8 \cdot 4 + 5 \cdot 5 + 4 \cdot 4 + 3 \cdot 10 + 2 \cdot 7 = 117$$

Для перетворення функціонального розміру на трудомісткість необхідно використати фактичну продуктивність команди. Якщо за історичними даними команда реалізує в середньому вісім функціональних точок на один людино-місяць, орієнтовна трудомісткість дорівнюватиме:

$$E = \frac{117}{8} = 14,625 \approx 15 \text{ людино-місяців}$$

Отриманий результат не слід автоматично використовувати як остаточну оцінку. Функціональні точки переважно характеризують функціональність, яку бачить користувач. Додатково необхідно врахувати роботи з архітектури, міграції даних, інформаційної безпеки, налаштування інфраструктури, автоматизації розгортання, тестування, документування й управління проектом.

Продуктивність у функціональних точках повинна визначатися за даними конкретної організації. Використання випадкового коефіцієнта з іншого проекту може призвести до значної похибки через відмінності в технологіях, кваліфікації команди, вимогах до якості та складності предметної області.

5.6. Оцінювання багатокomпонентного програмного продукту

Багатокomпонентний програмний продукт може включати клієнтські застосунки, серверну частину, базу даних, зовнішні інтеграції, аналітичні модулі, інфраструктуру розгортання та засоби інформаційної безпеки. Оцінювання такого продукту одним загальним числом приховує відмінності між компонентами й залежності між ними.

Спочатку продукт необхідно поділити на логічні компоненти. Кожний компонент оцінюють окремо з урахуванням його функціонального

розміру, технічної складності, вимог до якості, доступності фахівців і зовнішніх залежностей.

Крім компонентів продукту, необхідно окремо оцінити наскрізні роботи:

- інтеграційне й системне тестування;
- проєктування архітектури;
- забезпечення продуктивності та безпеки;
- міграцію й очищення даних;
- налаштування середовищ;
- автоматизацію збирання та розгортання;
- документування і навчання користувачів;
- управління випусками;
- координацію учасників проєкту.

Приклад узагальненої оцінки багатокomпонентного продукту наведено в таблиці 5.4.

Таблиця 5.4 – Приклад оцінювання багатокomпонентного програмного продукту

Складова продукту або роботи	Оцінка трудомісткості, людино-місяців
Аналіз вимог і проєктування	2–3
Клієнтський вебзастосунок	4–6
Серверна частина та API	6–9
База даних і міграція даних	3–5
Інтеграція із зовнішніми системами	4–7
Тестування та перевірка безпеки	4–6
Розгортання, документація і навчання	2–4
Інтеграційний резерв	3–5
Загальна трудомісткість	28–45

Підсумковий діапазон у таблиці становить від 28 до 45 людино-місяців. Ширина діапазону пояснюється невизначеністю вимог, складністю інтеграцій і можливістю повторного виконання частини робіт.

Під час підсумовування необхідно уникати подвійного врахування. Наприклад, якщо тестування вже включено до оцінки кожного компонента, його не слід повторно повністю додавати як окремий вид робіт. Водночас інтеграційне та системне тестування потрібно оцінювати окремо, оскільки

воно перевіряє взаємодію компонентів, а не лише кожний компонент окремо.

Сума трудомісткості компонентів не визначає календарну тривалість проєкту. Частина компонентів може розроблятися паралельно, а частина — лише після завершення залежних робіт. Тому після оцінювання трудомісткості необхідно сформувати мережу робіт і календарний розклад.

5.7. Оцінювання тривалості проєкту

Тривалість проєкту визначається не простим діленням загальної трудомісткості на кількість учасників команди. Для її розрахунку необхідно врахувати послідовність робіт, можливості паралельного виконання, доступність ресурсів, календарі, очікування, погодження, обмеження і ризику.

Процес оцінювання тривалості включає такі дії:

1. Визначення переліку робіт на основі структури декомпозиції.
2. Оцінювання трудомісткості та тривалості кожної роботи.
3. Визначення залежностей між роботами.
4. Призначення необхідних ресурсів.
5. Урахування доступності фахівців та інших ресурсів.
6. Побудову мережевої моделі проєкту.
7. Визначення критичного шляху.
8. Урахування резервів і формування базового розкладу.

Мережева модель відображає логічну послідовність проєктних робіт. Одні роботи можуть виконуватися паралельно, тоді як інші повинні починатися лише після завершення попередніх. Найдовша послідовність залежних робіт від початку до завершення проєкту утворює критичний шлях.

У спрощеному вигляді тривалість проєкту можна подати як найбільшу сумарну тривалість робіт на одному з його шляхів:

$$T_{\text{проєкту}} = \max \left(\sum T_i \right)$$

де T_i — тривалість роботи, що належить відповідному шляху.

Принцип визначення критичного шляху показано на рисунку 5.3.

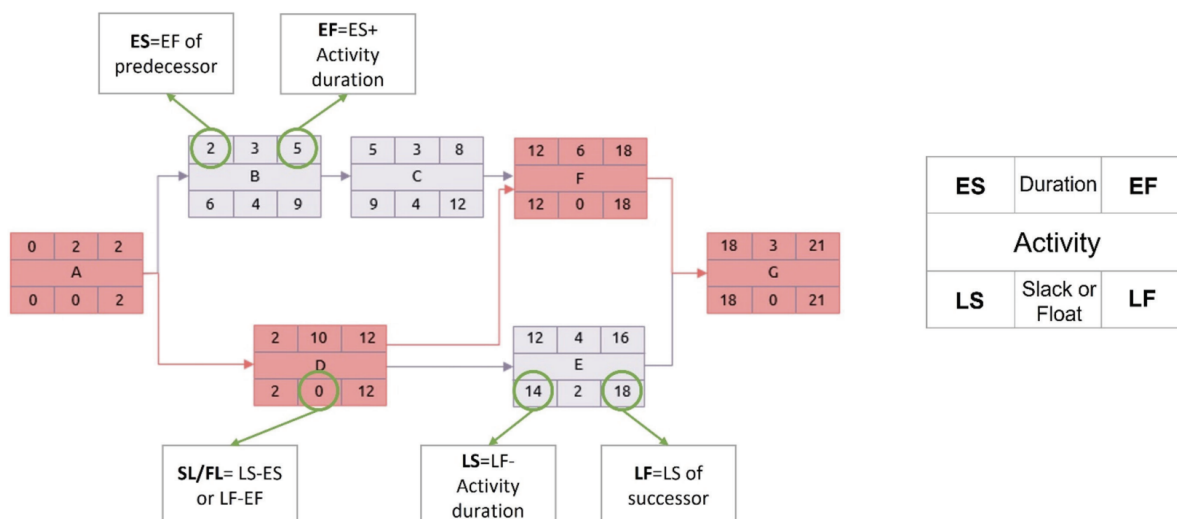


Рисунок 5.3 – Аналіз критичного шляху проєкту

Затримка роботи критичного шляху за відсутності резерву призводить до затримки всього проєкту. Роботи, які не належать до критичного шляху, можуть мати часовий резерв. Проте використання цього резерву може перетворити некритичну роботу на критичну.

Після визначення логічної послідовності робіт необхідно врахувати ресурсні обмеження. Якщо один фахівець одночасно призначений на кілька робіт, ці роботи неможливо виконати паралельно. Усунення таких конфліктів називається вирівнюванням ресурсів. Воно забезпечує реалістичність розкладу, але може збільшити загальну тривалість проєкту.

Наприклад, загальна трудомісткість продукту, наведеного в таблиці 5.4, становить 28–45 людино-місяців. Наявність команди з п'яти осіб не означає, що проєкт автоматично триватиме від 5,6 до 9 місяців. Частина учасників може бути залучена не на повний робочий час, деякі роботи виконуються лише послідовно, а інтеграційне тестування починається після отримання результатів розроблення.

До розкладу потрібно включати не лише активну роботу, а й календарне очікування:

- погодження вимог і проєктних рішень;
- отримання доступів;
- надання тестових даних;
- перевірку результатів замовником;
- закупівлю обладнання або ліцензій;
- проходження аудиту безпеки;
- розміщення застосунку в зовнішніх сервісах;

- приймання результатів проєкту.

За необхідності скорочення розкладу насамперед аналізують роботи критичного шляху. Додавання ресурсів до некритичних робіт не змінить дату завершення проєкту. Застосування додаткових ресурсів може збільшити бюджет, а паралельне виконання залежних робіт — підвищити ризик повторної роботи.

Остаточний базовий розклад повинен містити дати початку і завершення робіт, залежності, призначені ресурси, контрольні події та резерви. Після затвердження він використовується для порівняння запланованих і фактичних результатів. У разі істотної зміни вимог, ресурсів або зовнішніх умов оцінка та розклад повинні бути переглянуті відповідно до встановленого процесу управління змінами.

Висновки

Оцінювання трудомісткості й термінів програмного проєкту є процесом послідовного уточнення, а не одноразовим розрахунком. На ранніх етапах оцінки мають широкий діапазон, який звужується після деталізації вимог, архітектури, компонентів і проєктних робіт.

Трудомісткість характеризує обсяг праці, тоді як тривалість визначає календарний час виконання. Збільшення кількості виконавців не завжди пропорційно скорочує тривалість через залежності між роботами, обмежену можливість паралельного виконання та додаткові комунікаційні витрати.

Оцінку потрібно розглядати як імовірнісне твердження, що містить діапазон, припущення та рівень упевненості. Для врахування невизначеності можуть використовуватися триточкове оцінювання, резервний аналіз і періодичне уточнення прогнозів.

Прагматичний підхід передбачає декомпозицію продукту, залучення виконавців, використання історичних даних, поєднання декількох методів і порівняння оцінок із фактичними результатами.

Метод функціональних точок дає змогу визначити функціональний розмір продукту незалежно від технології реалізації. Для перетворення функціонального розміру на трудомісткість необхідно використовувати перевірені показники продуктивності конкретної команди або організації.

Тривалість проєкту встановлюється на основі залежностей між роботами, критичного шляху, календарів і ресурсних обмежень. Необґрунтовано «агресивний» розклад збільшує ризик дефектів, повторної

роботи, технічного боргу, перевищення бюджету та професійного вигорання команди. Тому розклад повинен бути не лише бажаним, а й технічно та ресурсно обґрунтованим.

Лекція 6 Формування проєктної команди

Тема: Формування проєктної команди

Мета: Сформувати у студентів розуміння принципів формування ефективної проєктної команди, ознайомити з роллю лідерства та управління, підходами до добору учасників відповідно до необхідних компетентностей, розподілу ролей і відповідальності, а також методами мотивації та організації командної взаємодії.

План лекції

1. Проєктна команда та її роль у реалізації проєкту.
2. Основні ролі та відповідальність учасників команди.
3. Лідерство й управління проєктною командою.
4. Добір учасників і формування необхідного набору компетентностей.
5. Мотивація учасників проєкту.
6. Організація ефективної командної взаємодії.
7. Висновки.

6.1. Проєктна команда та її роль у реалізації проєкту

Проєктна команда — це група людей із взаємодоповнювальними знаннями, уміннями та повноваженнями, які спільно працюють над досягненням цілей проєкту і створенням його результатів. Учасники команди виконують різні функції, але несуть спільну відповідальність за узгоджене виконання проєктних робіт.

Команда відрізняється від звичайної робочої групи наявністю спільної мети, взаємозалежністю учасників і необхідністю постійної координації. Результат одного учасника часто є вхідною інформацією для роботи іншого. Наприклад, розробник використовує вимоги, підготовлені бізнес-аналітиком, а тестувальник перевіряє програмний компонент відповідно до визначених критеріїв приймання.

Проєктна команда створюється на обмежений період і може включати працівників різних структурних підрозділів, зовнішніх фахівців, підрядників, представників замовника та майбутніх користувачів. Її склад залежить від масштабу, складності, змісту, технологій і способу організації проєкту.

Основними завданнями проєктної команди є:

- участь у визначенні змісту та плануванні проєкту;
- виконання робіт відповідно до затвердженого плану;

- створення і перевірка проєктних результатів;
- надання інформації про стан виконання робіт;
- виявлення ризиків, проблем і потреби у змінах;
- участь у проєктних нарадах і прийнятті рішень;
- забезпечення відповідності результатів установленим вимогам;
- накопичення й передавання досвіду, отриманого під час реалізації проєкту.

Проєктна команда може мати декілька рівнів. Керівний рівень визначає стратегічний напрям і приймає ключові рішення. Управлінський рівень організовує, координує та контролює роботу. Виконавський рівень безпосередньо створює результати проєкту. Допоміжні ролі забезпечують адміністративну, методичну, правову, архітектурну та якісну підтримку.

Взаємозв'язок основних рівнів і ролей проєктної організації показано на рисунку 6.1.

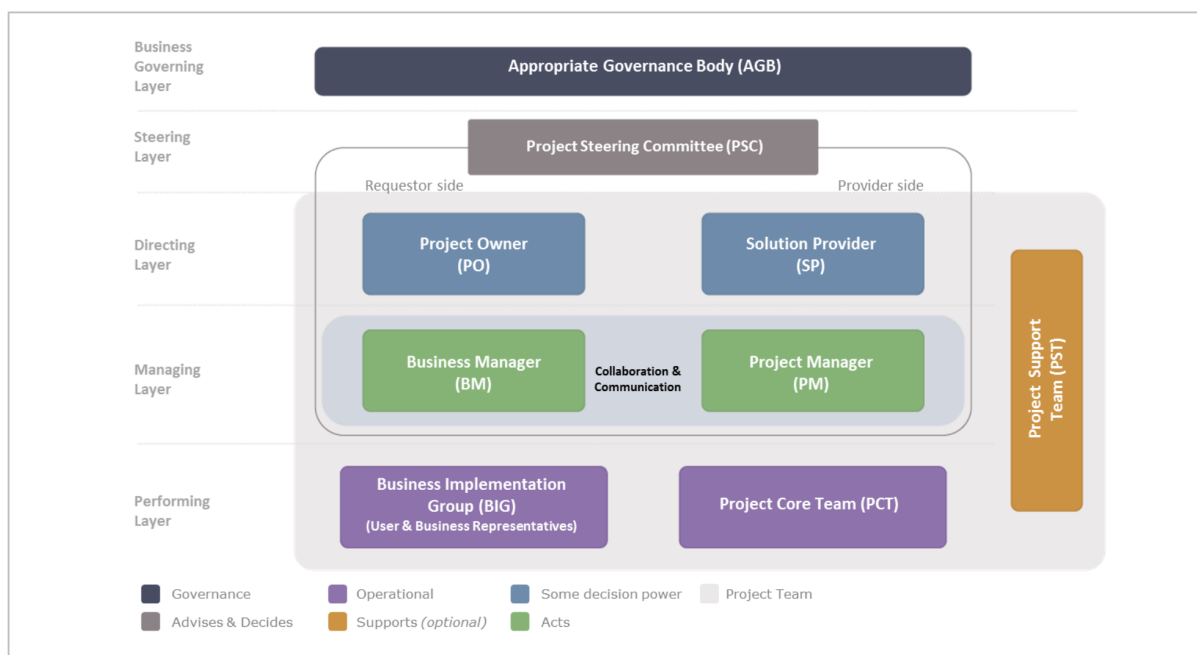


Рисунок 6.1 – Організація проєкту та основні рівні управління

Склад команди може змінюватися протягом життєвого циклу проєкту. На початку більше залучаються бізнес-аналітики, архітектори й фахівці предметної області. Під час розроблення основне навантаження переходить до програмістів, дизайнерів, інженерів баз даних та інших технічних фахівців. На етапах тестування і впровадження зростає роль тестувальників, інженерів із розгортання, представників користувачів і служби підтримки.

Менеджер проєкту повинен завчасно визначити, які спеціалісти та коли будуть потрібні. Несвоєчасне залучення учасника може спричинити затримку залежних робіт, а надто раннє залучення — неефективне використання ресурсів.

Ефективність проєктної команди визначається не лише індивідуальною кваліфікацією учасників. Важливими є чіткість спільної мети, узгодженість ролей, довіра, обмін інформацією, здатність приймати рішення та готовність підтримувати інших учасників.

6.2. Основні ролі та відповідальність учасників команди

Роль визначає очікувану поведінку, повноваження і сферу відповідальності учасника проєкту. Одна людина може виконувати декілька ролей, особливо в невеликому проєкті. У складному проєкті одну роль можуть виконувати декілька людей, але відповідальність і порядок прийняття рішень повинні залишатися однозначними.

Типовий склад команди ІТ-проєкту наведено в таблиці 6.1.

Таблиця 6.1 – Основні ролі учасників ІТ-проєкту

Роль	Основна відповідальність
Власник або спонсор проєкту	Визначає бізнес-цілі, забезпечує фінансування, приймає ключові рішення та відповідає за доцільність проєкту
Менеджер проєкту	Планує і координує роботи, управляє ресурсами, термінами, ризиками, змінами та комунікаціями
Бізнес-менеджер або представник замовника	Забезпечує відповідність результатів потребам організації та координує впровадження змін
Бізнес-аналітик	Збирає, аналізує, узгоджує і документує вимоги
Архітектор програмного забезпечення	Формує архітектуру, визначає технічні принципи та контролює узгодженість технічних рішень
Керівник технічної команди	Координує технічну роботу, розподіляє завдання та контролює дотримання прийнятих рішень
Розробники	Проєктують і реалізують програмні компоненти, виконують модульне тестування та виправляють дефекти
Тестувальники	Планують і проводять тестування, реєструють дефекти та перевіряють відповідність результатів вимогам
UX/UI-дизайнер	Досліджує потреби користувачів і створює інтерфейси програмного продукту

DevOps-інженер	Налаштовує середовища, автоматизацію збирання, тестування, розгортання і моніторингу
Фахівець з інформаційної безпеки	Визначає вимоги безпеки та контролює їх виконання
Представник користувачів	Уточнює потреби користувачів, бере участь у перевірці та прийманні результатів

Менеджер проєкту здійснює щоденне управління проєктом, координує основну команду, контролює використання ресурсів і забезпечує досягнення цілей у межах установлених обмежень. Він також управляє очікуваннями зацікавлених сторін, ризиками, проблемами, змінами та проєктними комунікаціями.

Водночас менеджер проєкту не повинен виконувати всю роботу особисто або приймати кожне технічне рішення. Його завдання полягає у створенні умов, за яких компетентні учасники можуть ефективно виконувати доручені їм завдання.

Для розподілу відповідальності застосовується матриця RAM. Одним із її варіантів є матриця RASCI, у якій використовуються такі позначення:

- R — Responsible: виконує роботу;
- A — Accountable: несе остаточну відповідальність за результат;
- S — Supports: допомагає виконавцю;
- C — Consulted: надає консультації;
- I — Informed: отримує інформацію про перебіг або результат роботи.

Для кожного завдання бажано призначати одного відповідального за виконання та одного учасника, який несе остаточну відповідальність. Це запобігає ситуаціям, коли робота залишається без власника або декілька осіб очікують дій одна від одної.

Приклад розподілу відповідальності в ІТ-проєкті наведено в таблиці 6.2.

Таблиця 6.2 – Приклад матриці RASCI для ІТ-проєкту

Проектна робота	Власник проєкту	Менеджер проєкту	Бізнес-аналітик	Розробники	Тестувальники	Представник користувачів
Визначення	A	C	R	C	C	S

та погодження вимог						
Планування виконання робіт	I	A/R	C	S	S	I
Розроблення програмного компонента	I	A	C	R	S	I
Тестування компонента	I	A	C	S	R	C
Приймання результату	A	R	C	I	S	C

Матриця не замінює посадових інструкцій або детального опису завдань. Вона стисло показує, хто виконує роботу, хто приймає остаточне рішення, кого потрібно залучити до консультацій і кому необхідно повідомити про результат.

Розподіл відповідальності необхідно погоджувати з учасниками команди. Формальне призначення ролі без перевірки повноважень, компетентності й доступності людини не забезпечує фактичного виконання роботи.

6.3. Лідерство й управління проєктною командою

Управління та лідерство є взаємопов'язаними, але не тотожними видами діяльності. Управління забезпечує планування, організацію, контроль і передбачуваність. Лідерство формує напрям, мотивує людей і допомагає команді працювати в умовах невизначеності та змін.

Менеджер проєкту як управлінець:

- планує та розподіляє роботи;
- призначає ресурси;
- контролює терміни й витрати;
- відстежує виконання завдань;
- управляє ризиками, проблемами і змінами;
- організовує звітність;
- забезпечує дотримання встановлених процесів.

Менеджер проєкту як лідер:

- пояснює мету і цінність проєкту;

- формує довіру;
- підтримує команду під час складних ситуацій;
- сприяє професійному розвитку учасників;
- допомагає вирішувати конфлікти;
- заохочує відповідальність та ініціативність;
- створює умови для спільного прийняття рішень.

Єдиного універсального стилю лідерства для всіх проєктів не існує. Стиль повинен відповідати складності проєкту, зрілості команди, рівню невизначеності та поточному етапу життєвого циклу.

На початку роботи новій команді може бути потрібне більш директивне управління: чіткі завдання, правила й порядок прийняття рішень. Після набуття досвіду та узгодження способів роботи менеджер може передавати команді більше повноважень і переходити до підтримувального або делегувального стилю.

У складних проєктах може застосовуватися розподілене лідерство. Учасники беруть на себе лідерські функції у тих сферах, де мають найбільшу компетентність. Архітектор очолює роботу над технічними рішеннями, бізнес-аналітик — уточнення вимог, а керівник тестування — організацію контролю якості. Менеджер проєкту координує ці напрями й забезпечує їх узгодженість.

Проектний менеджер часто відповідає за результат, не маючи прямої адміністративної влади над усіма учасниками. Тому він повинен уміти впливати за допомогою професійної компетентності, довіри, аргументації, комунікації та особистого прикладу.

Основні джерела влади, які може використовувати менеджер проєкту, показано на рисунку 6.2.



Рисунок 6.2 – Джерела влади менеджера проєкту

До основних джерел влади належать:

- посадові повноваження;
- можливість надавати винагороди;
- можливість застосовувати санкції;
- професійні знання і досвід;
- доступ до важливої інформації;
- особистий авторитет і довіра.

Найстійкіший позитивний вплив забезпечують професійна компетентність, надійність, відкритість і справедливість. Надмірне використання примусу або формальних повноважень може забезпечити короточасне виконання вимог, але послаблює довіру й ініціативність команди.

Лідер повинен бути послідовним у рішеннях, виконувати власні зобов'язання, пояснювати причини змін і однаково застосовувати правила до всіх учасників. Довіра зменшується, якщо менеджер приховує інформацію, перекладає відповідальність, допускає фаворитизм або змінює вимоги без пояснення.

6.4. Добір учасників і формування необхідного набору компетентностей

Формування команди починається з визначення робіт, які необхідно виконати, і компетентностей, потрібних для створення результатів проекту. Спочатку формується структура декомпозиції робіт, після чого для кожного робочого пакета визначаються необхідні знання, навички, досвід і рівень відповідальності.

Під час добору учасників необхідно враховувати:

- професійні знання і технічні навички;
- досвід виконання подібних робіт;
- знання предметної області;
- здатність працювати в команді;
- комунікаційні навички;
- вміння приймати рішення;
- готовність навчатися;
- доступність у необхідний період;
- вартість залучення;
- можливість роботи в умовах невизначеності;
- відповідність культурі й цінностям команди.

До компетентностей учасника проєкту належать не лише технічні знання. Фахівець може мати високу індивідуальну продуктивність, але створювати проблеми через небажання ділитися інформацією, неприйняття спільних рішень або нездатність конструктивно вирішувати конфлікти.

Для оцінювання складу команди можна застосовувати матрицю компетентностей. Її приклад наведено в таблиці 6.3.

Таблиця 6.3 – Приклад матриці компетентностей команди ІТ-проєкту

Компетентність	Аналітик	Архітектор	Розробник	Тестувальник	DevOps-інженер
Аналіз і документування вимог	Високий	Середній	Базовий	Середній	Базовий
Проектування архітектури	Базовий	Високий	Середній	Базовий	Середній
Програмування	Базовий	Високий	Високий	Середній	Середній
Тестування	Середній	Середній	Середній	Високий	Середній
Автоматизація розгортання	Базовий	Середній	Середній	Базовий	Високий
Знання предметної області	Високий	Середній	Середній	Середній	Базовий
Командна взаємодія	Високий	Високий	Високий	Високий	Високий

Матриця дає змогу виявити критичні компетентності, якими володіє лише одна людина, а також навички, відсутні в команді. Залежність від одного фахівця створює ризик для проєкту, тому важливі знання необхідно документувати й передавати іншим учасникам.

Дефіцит компетентностей може бути усунутий за допомогою:

- навчання учасників;
- наставництва;
- перерозподілу ролей;
- залучення внутрішнього експерта;
- наймання нового працівника;
- залучення консультанта;

- передавання окремих робіт підряднику;
- зміни технічного рішення або змісту проєкту.

Команда повинна поєднувати досвідчених фахівців і учасників, які розвивають компетентності. Використання лише найдосвідченіших працівників може бути економічно недоцільним, тоді як команда без достатнього рівня досвіду потребуватиме надмірного контролю й матиме підвищений ризик помилок.

Необхідно також перевіряти реальну доступність людей. Формальне включення фахівця до команди не означає, що він може приділяти проєкту весь робочий час. Учасник може одночасно працювати над кількома проєктами, виконувати функціональні обов'язки або бути доступним лише на певних етапах.

Після залучення нового учасника необхідно організувати його введення в проєкт. Він повинен отримати інформацію про цілі, вимоги, архітектуру, правила роботи, інструменти, відповідальність, порядок комунікації та прийняття рішень.

6.5. Мотивація учасників проєкту

Мотивація визначає готовність людини докладати зусиль для досягнення проєктних цілей. Вона впливає на продуктивність, якість роботи, ініціативність, відповідальність і бажання залишатися в команді.

Мотивація може бути зовнішньою і внутрішньою. Зовнішня мотивація пов'язана із заробітною платою, преміями, підвищенням, пільгами та іншими винагородами. Внутрішня мотивація виникає через інтерес до роботи, професійний розвиток, автономію, відчуття значущості результату та належності до команди.

Матеріальна винагорода є важливою, але сама по собі не забезпечує високої залученості. Навіть добре оплачуваний фахівець може втратити мотивацію через нечіткі цілі, постійну зміну пріоритетів, несправедливий розподіл навантаження, мікроменеджмент або відсутність визнання.

Менеджер проєкту може підтримувати мотивацію за допомогою таких дій:

- пояснювати мету та цінність проєкту;
- установлювати чіткі й досяжні цілі;
- залучати учасників до планування і прийняття рішень;
- надавати необхідну автономію;
- забезпечувати справедливий розподіл роботи;

- своєчасно надавати конструктивний зворотний зв'язок;
- визнавати особисті та командні досягнення;
- створювати можливості для навчання;
- усувати перешкоди для виконання роботи;
- підтримувати безпечне середовище для обговорення проблем;
- запобігати постійному перевантаженню.

Підхід менеджера до учасників впливає на їхню поведінку. Якщо керівник вважає, що працівники уникають відповідальності та потребують постійного контролю, він схильний застосовувати мікроменеджмент. Унаслідок цього команда втрачає самостійність та ініціативність.

Якщо менеджер довіряє команді, встановлює зрозумілі очікування і надає повноваження, учасники частіше беруть відповідальність за результат. Водночас довіра не означає відсутності контролю. Вона передбачає прозорі цілі, погоджені критерії готовності та регулярний обмін інформацією.

Визнання повинно бути своєчасним, конкретним і справедливим. Замість загальної похвали доцільно зазначати, який саме внесок зробив учасник і як він вплинув на результат проекту. Необхідно визнавати не лише індивідуальні досягнення, а й ефективну співпрацю та допомогу іншим.

Мотивацію знижують суперечливі пріоритети, нереалістичні терміни, постійні переробки, відсутність необхідних ресурсів і невирішені конфлікти. Тому завдання менеджера полягає не тільки в заохоченні команди, а й в усуненні організаційних причин демотивації.

6.6. Організація ефективної командної взаємодії

Ефективна взаємодія передбачає своєчасний обмін достовірною інформацією, узгоджені способи роботи, взаємну підтримку та спільну відповідальність за результат. Команда повинна розуміти, що необхідно зробити, хто відповідає за кожен роботу, як приймаються рішення і куди звертатися у разі виникнення проблеми.

На початку спільної роботи доцільно провести установчу зустріч команди. Під час неї необхідно:

- представити цілі, зміст і очікувані результати проекту;
- ознайомити учасників із планом робіт;
- уточнити ролі та відповідальність;
- визначити правила комунікації;

- погодити порядок прийняття рішень;
- представити процеси управління ризиками, проблемами і змінами;
- визначити порядок ескалації;
- погодити правила командної роботи;
- уточнити критерії готовності та приймання результатів.

Правила командної роботи можуть визначати робочий час, канали зв'язку, очікуваний час відповіді, порядок проведення нарад, правила перевірки програмного коду, використання системи контролю версій і процедуру вирішення конфліктів.

Команда проходить декілька стадій розвитку. Відповідно до моделі Такмана виділяють формування, конфронтацію, нормування, продуктивну роботу та завершення. Ці стадії показано на рисунку 6.3.

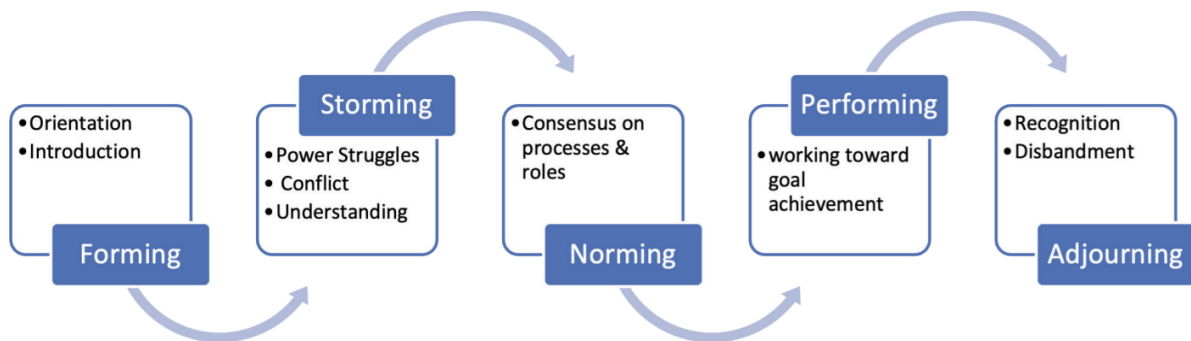


Рисунок 6.3 – Стадії розвитку проєктної команди за моделлю Такмана

На стадії формування учасники знайомляться, уточнюють мету та намагаються зрозуміти власні ролі. Менеджер повинен надати чіткий напрям, пояснити правила та забезпечити доступ до необхідної інформації.

На стадії конфронтації виникають суперечки щодо ролей, пріоритетів, методів роботи й розподілу впливу. Завдання менеджера полягає не в придушенні всіх суперечностей, а в забезпеченні конструктивного обговорення та запобіганні особистим конфліктам.

На стадії нормування команда погоджує правила, розподіляє відповідальність і виробляє спільні способи роботи. Менеджер діє переважно як фасилітатор і допомагає закріпити ефективні практики.

На стадії продуктивної роботи учасники зосереджуються на досягненні результатів, самостійно координують значну частину діяльності та конструктивно вирішують проблеми. Менеджер може

делегувати більше повноважень і приділяти основну увагу зовнішнім перешкодам та очікуванням зацікавлених сторін.

На стадії завершення команда передає результати, систематизує отриманий досвід і переходить до інших проєктів. Необхідно провести підсумковий аналіз, зафіксувати засвоєні уроки та визнати внесок учасників.

Повернення до попередніх стадій є нормальним явищем. Воно може відбутися після зміни складу команди, керівника, вимог, технології або способу організації робіт.

Основою ефективної взаємодії є довіра. За високого рівня довіри учасники відкрито повідомляють про проблеми, звертаються по допомогу та обмінюються знаннями. За низького рівня довіри інформація приховується, рішення затримуються, а проблеми стають помітними лише після виникнення значних відхилень.

У команді повинна підтримуватися психологічна безпека — можливість висловити сумнів, поставити запитання, повідомити про помилку або запропонувати інше рішення без страху приниження чи покарання. Психологічна безпека не скасовує відповідальності, а дає змогу виявляти проблеми на ранньому етапі.

Конфлікти в проєктній команді можуть виникати через ресурси, терміни, технічні рішення, пріоритети, розподіл відповідальності або особисті відмінності. Конструктивний конфлікт допомагає порівняти альтернативи й знайти краще рішення. Деструктивний конфлікт переходить від обговорення проблеми до особистого протистояння.

Під час вирішення конфлікту необхідно:

1. Визначити факти та предмет розбіжності.
2. Надати сторонам можливість викласти власну позицію.
3. Відокремити проблему від особистостей.
4. Уточнити спільну мету та обмеження.
5. Розглянути можливі варіанти рішення.
6. Погодити дії, відповідальних і терміни.
7. Перевірити виконання домовленостей.

Для розподіленої команди особливого значення набувають письмова фіксація рішень, спільні інформаційні системи, узгоджені канали зв'язку та правила асинхронної взаємодії. Важлива інформація не повинна залишатися лише в особистому листуванні або усних домовленостях.

Ефективна команда використовує єдине джерело актуальної інформації про завдання, вимоги, рішення, ризики й дефекти. Регулярність комунікації повинна відповідати потребам проєкту: надмірна кількість зустрічей зменшує час продуктивної роботи, а недостатня комунікація призводить до неузгоджених рішень.

Висновки

Проектна команда є групою взаємозалежних учасників, які спільно створюють результати проєкту. Її ефективність визначається не лише професійними знаннями окремих фахівців, а й узгодженістю ролей, довірою, мотивацією та якістю взаємодії.

Формування команди повинно починатися з аналізу змісту робіт і визначення необхідних компетентностей. Під час добору учасників необхідно враховувати професійні знання, досвід, доступність, комунікаційні навички та здатність працювати в умовах невизначеності.

Ролі й відповідальність мають бути чітко визначені та погоджені. Матриця RASCI дає змогу встановити виконавців, відповідальних за результат, учасників підтримки, консультантів і отримувачів інформації.

Менеджер проєкту поєднує функції управління та лідерства. Він повинен планувати і контролювати роботи, але водночас формувати довіру, підтримувати мотивацію, розвивати команду та адаптувати стиль керівництва до поточної ситуації.

Мотивація підтримується через зрозумілі цілі, справедливий розподіл роботи, автономію, визнання досягнень, професійний розвиток і своєчасне усунення перешкод. Надмірний контроль, нереалістичні терміни та відсутність довіри знижують залученість учасників.

Ефективна командна взаємодія ґрунтується на прозорій комунікації, спільних правилах, психологічній безпеці та конструктивному вирішенні конфліктів. Завдання менеджера полягає у створенні умов, за яких компетентні учасники можуть узгоджено працювати, приймати відповідальність і досягати спільних цілей проєкту.

Лекція 7 Реалізація проєкту

Тема: Реалізація проєкту

Мета: Сформувати у студентів розуміння процесів реалізації ІТ-проєкту, організації поточної роботи команди, координації виконання запланованих завдань, застосування принципів кількісного управління, контролю результатів і належного завершення проєкту.

План лекції

1. Мета, завдання та основні процеси реалізації проєкту.
2. Організація робочого планування.
3. Координація команди та виконання проєктних робіт.
4. Моніторинг прогресу й управління відхиленнями.
5. Принципи кількісного управління проєктом.
6. Контроль результатів і завершення проєкту.
7. Висновки.

7.1. Мета, завдання та основні процеси реалізації проєкту

Реалізація проєкту — це сукупність процесів, під час яких команда виконує заплановані роботи та створює визначені проєктом результати. На цьому етапі затверджені плани перетворюються на конкретні програмні компоненти, документи, налаштовані середовища, інтеграції, результати тестування та інші продукти проєкту.

Основою реалізації є план управління проєктом, який містить базові плани змісту, розкладу й бюджету, а також правила управління ресурсами, якістю, комунікаціями, ризиками, змінами та зацікавленими сторонами. Ці документи визначають, що необхідно створити, хто виконує роботи, у які терміни вони повинні бути завершені та як перевірятимуться отримані результати.

Основними завданнями етапу реалізації є:

- організація виконання запланованих робіт;
- розподіл завдань між учасниками команди;
- забезпечення виконавців необхідною інформацією і ресурсами;
- створення та перевірка проєктних результатів;
- координація взаємодії учасників;
- управління якістю;
- реалізація погоджених змін;
- виконання заходів реагування на ризики;
- збирання інформації про фактичний стан проєкту;

- інформування зацікавлених сторін;
- підготовка результатів до приймання і передавання замовнику.

Реалізація не є ізольованим процесом. Паралельно з виконанням робіт здійснюються моніторинг і контроль. Команда створює результати, а менеджер проєкту порівнює фактичний стан із базовим планом, виявляє відхилення та організовує коригувальні дії.

Основні процеси й результати етапу реалізації показано на рисунку 7.1.

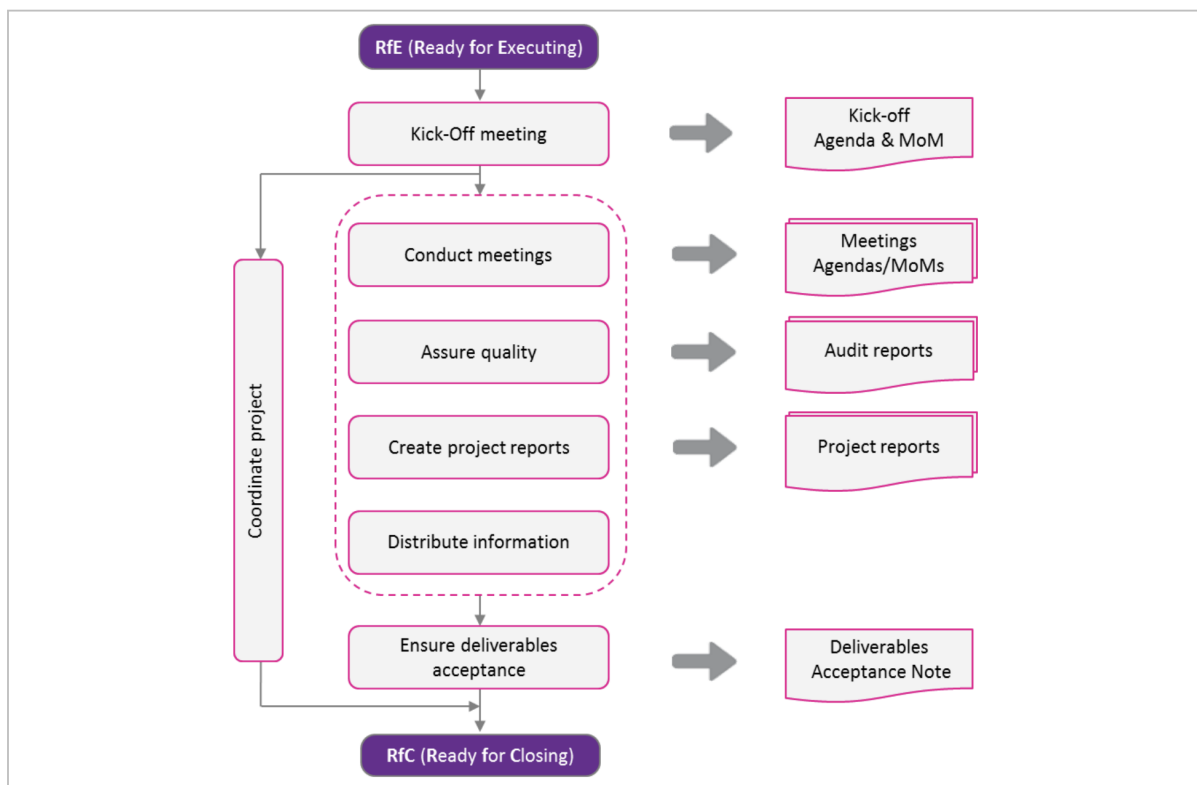


Рисунок 7.1 – Основні процеси та результати фази реалізації проєкту

Перед початком активного виконання робіт доцільно провести установчу зустріч команди. Під час неї уточнюються цілі проєкту, план робіт, ролі, правила комунікації, порядок управління ризиками, проблемами й змінами, критерії приймання результатів і процедура ескалації.

Менеджер проєкту повинен переконатися, що кожний учасник розуміє:

- мету та очікувані результати проєкту;
- власні завдання і відповідальність;
- зв'язок своєї роботи з іншими роботами;

- терміни виконання;
- критерії готовності результату;
- правила звітування;
- порядок повідомлення про проблеми;
- межі власних повноважень.

На етапі реалізації використовується значна частина запланованих ресурсів і бюджету. Тому помилки, виявлені із запізненням, можуть призвести до значних витрат. Регулярний контроль проміжних результатів дає змогу виявляти проблеми раніше, ніж вони вплинуть на весь проєкт.

7.2. Організація робочого планування

Базовий план проєкту визначає погоджені межі змісту, термінів і бюджету. Проте для щоденної роботи команди його необхідно деталізувати до рівня конкретних завдань, виконавців, строків і результатів. Цей процес називається робочим плануванням.

Робоче планування забезпечує зв'язок між загальним планом і поточною діяльністю команди. Воно дає відповіді на такі запитання:

- які роботи повинні виконуватися зараз;
- хто відповідає за кожну роботу;
- який результат необхідно отримати;
- коли робота повинна початися і завершитися;
- від яких попередніх робіт вона залежить;
- які ресурси потрібні;
- як перевірятиметься готовність результату;
- які перешкоди можуть вплинути на виконання.

Основою робочого плану є структура декомпозиції робіт, календарний розклад і перелік проєктних результатів. Робочі пакети поділяються на завдання, які можна оцінити, призначити виконавцям і проконтролювати.

Основні поля робочого плану наведено в таблиці 7.1.

Таблиця 7.1 – Приклад структури робочого плану IT-проєкту

Поле	Зміст
Ідентифікатор	Унікальний номер роботи або завдання
Назва роботи	Стисле формулювання дії або очікуваного результату
Відповідальний	Учасник, який організовує виконання і звітує про результат

Виконавці	Учасники, безпосередньо залучені до роботи
Дата початку	Запланована або фактична дата початку
Дата завершення	Запланована або прогнозована дата завершення
Попередні роботи	Завдання, від яких залежить початок або завершення роботи
Необхідні ресурси	Людські, технічні, інформаційні та фінансові ресурси
Критерії готовності	Умови, за яких результат вважається завершеним
Статус	Не розпочато, виконується, заблоковано, завершено або прийнято
Примітки	Припущення, обмеження, проблеми та додаткова інформація

Робочий план повинен бути достатньо детальним для координації роботи, але не перевантаженим надмірною кількістю дрібних операцій. Рівень деталізації залежить від складності проєкту, тривалості робіт і потреби в контролі.

Якщо на початку проєкту відсутня достатня інформація для детального планування всього обсягу робіт, використовується планування методом набіжної хвилі. Найближчі роботи плануються докладно, а віддалені залишаються на вищому рівні та уточнюються в міру отримання інформації.

Під час поточного планування необхідно розрізняти:

- базовий план — офіційно затверджене представлення змісту, строків і бюджету;
- поточний план — актуальний стан робіт з урахуванням фактичного виконання;
- прогноз — очікуваний майбутній стан проєкту за наявних тенденцій.

Внесення фактичних даних до робочого плану не означає автоматичної зміни базового плану. Базовий план змінюється лише після погодження відповідного запиту на зміну.

Робоче планування повинно виконуватися за участю майбутніх виконавців. Учасники, які безпосередньо виконуватимуть роботу, краще розуміють її технічну складність, залежності, необхідні ресурси та можливі ризики. Залучення команди також підвищує реалістичність плану і відповідальність за його виконання.

7.3. Координація команди та виконання проєктних робіт

Координація проєкту полягає в узгодженні дій учасників, розподілі ресурсів, передаванні необхідної інформації та підтриманні послідовного виконання запланованих робіт.

Менеджер проєкту координує щоденну діяльність команди, але не повинен перетворювати координацію на постійне втручання в роботу кожного фахівця. Його основним завданням є створення умов для виконання робіт і своєчасне усунення перешкод.

Координація команди включає:

- призначення завдань і уточнення очікуваних результатів;
- перевірку доступності виконавців;
- надання інформації, доступів та інших ресурсів;
- узгодження залежностей між роботами;
- контроль виконання критичних завдань;
- перевірку проміжних результатів;
- організацію взаємодії між технічними й бізнес-учасниками;
- підтримання мотивації команди;
- вирішення конфліктів;
- передавання проблем на вищий рівень управління;
- забезпечення актуальності проєктної документації.

Кожне завдання повинно мати зрозумілий результат і критерії готовності. Формулювання «розробити модуль авторизації» є недостатньо точним, якщо не визначено підтримувані сценарії, вимоги безпеки, необхідність автоматизованих тестів, документації та перевірки програмного коду.

Критерії готовності можуть включати:

- реалізацію погоджених функцій;
- успішне проходження модульних та інтеграційних тестів;
- перевірку програмного коду;
- усунення критичних дефектів;
- оновлення документації;
- розгортання в тестовому середовищі;
- підтвердження відповідності критеріям приймання.

Під час реалізації важливо підтримувати регулярний обмін інформацією. Для цього можуть використовуватися короткі координаційні зустрічі, щотижневі наради, демонстрації проміжних результатів, письмові звіти та спільні системи обліку завдань.

Комунікація повинна бути спрямована на прийняття рішень, а не лише на повідомлення про виконані дії. Інформація про проблему повинна містити її вплив, причину, запропоновані варіанти реагування та рішення, яке необхідно прийняти.

Доцільно розмежовувати ризик, проблему, зміну і рішення:

- ризик — невизначена подія, яка може вплинути на проєкт;
- проблема — подія або обставина, яка вже виникла і потребує реагування;
- зміна — запропоноване коригування погодженого змісту, термінів, бюджету або способу реалізації;
- рішення — погоджений вибір способу подальших дій.

Відповідна інформація фіксується в реєстрі ризиків, журналі проблем, журналі змін і журналі рішень. Ведення окремих журналів дає змогу простежити історію подій, призначити відповідальних і контролювати виконання погоджених дій.

Менеджер проєкту повинен приділяти особливу увагу роботам критичного шляху, залежностям між командами та завданням, які блокують подальше виконання. Запізнення з передаванням результату однієї команди може спричинити простої декількох інших учасників.

7.4. Моніторинг прогресу й управління відхиленнями

Моніторинг полягає в систематичному збиранні та аналізі інформації про стан проєкту. Контроль передбачає порівняння фактичних результатів із базовим планом, визначення відхилень і виконання необхідних управлінських дій.

Моніторинг і контроль здійснюються протягом усього проєкту, але найбільша їх інтенсивність припадає на етап реалізації.

Менеджер проєкту повинен відстежувати:

- виконання запланованих завдань;
- досягнення контрольних подій;
- готовність і приймання проєктних результатів;
- використання ресурсів;
- фактичні витрати;
- відповідність календарному плану;
- зміни змісту та вимог;
- стан ризиків і проблем;
- кількість і критичність дефектів;

- результати контролю якості;
- залученість зацікавлених сторін;
- стан і продуктивність команди.

Для кожного показника необхідно визначити базове значення, допустиме відхилення та поріг ескалації. Незначні відхилення можуть усуватися менеджером проєкту в межах його повноважень. Критичні відхилення, які загрожують досягненню основних цілей, повинні передаватися керівному органу або спонсору проєкту.

Основні види відхилень і можливі дії наведено в таблиці 7.2.

Таблиця 7.2 – Управління основними відхиленнями проєкту

Вид відхилення	Можлива причина	Управлінська дія
Відставання від розкладу	Нереалістична оцінка, блокування, нестача ресурсів	Перегляд послідовності робіт, усунення блокування, залучення ресурсів
Перевищення бюджету	Додаткові роботи, низька продуктивність, зростання вартості ресурсів	Аналіз причин, перегляд витрат, коригування прогнозу
Невідповідність змісту	Нечіткі вимоги, неконтрольовані зміни	Уточнення вимог, застосування процесу управління змінами
Низька якість	Недостатнє тестування, помилки проєктування, поспішна розробка	Аналіз першопричини, виправлення дефектів, посилення контролю
Перевантаження команди	Надмірна кількість паралельних завдань	Перерозподіл робіт, зміна пріоритетів, вирівнювання ресурсів
Низька залученість стейкхолдерів	Неефективна комунікація, суперечливі очікування	Перегляд способу взаємодії та плану комунікацій

Виявлення відхилення повинно супроводжуватися аналізом його першопричини. Усунення лише зовнішнього прояву проблеми може призвести до її повторного виникнення.

За результатами аналізу можуть застосовуватися:

- коригувальні дії — повертають фактичне виконання до погодженого плану;
- запобіжні дії — зменшують імовірність майбутнього відхилення;

- виправлення дефектів — усуває невідповідність у створеному результаті;
- запит на зміну — формально коригує погоджені параметри проєкту.

Зміна не повинна впроваджуватися без оцінювання її впливу. Необхідно проаналізувати, як вона змінить зміст, розклад, бюджет, якість, ризики, ресурси й очікування зацікавлених сторін.

Після погодження зміни оновлюються відповідні плани, журнали, вимоги, розклад і бюджет. Команда повинна отримати інформацію про нове рішення та момент набуття ним чинності. Відхилені або відкладені зміни також фіксуються для збереження історії прийнятих рішень.

Звіт про стан проєкту повинен показувати не лише виконану роботу, а й прогноз завершення. До нього доцільно включати:

- загальний стан проєкту;
- досягнуті результати;
- стан розкладу і бюджету;
- основні ризики та проблеми;
- погоджені зміни;
- прогнозовані дати й витрати;
- необхідні управлінські рішення;
- заплановані дії на наступний звітний період.

7.5. Принципи кількісного управління проєктом

Кількісне управління передбачає використання вимірюваних показників для оцінювання фактичного стану проєкту, виявлення тенденцій і прогнозування майбутніх результатів. Показники повинні допомагати приймати рішення, а не збиратися лише для звітності.

До основних груп проєктних показників належать:

- показники виконання змісту;
- показники календарного плану;
- показники витрат;
- показники використання ресурсів;
- показники якості;
- показники ризиків;
- показники продуктивності;
- показники задоволеності зацікавлених сторін.

Одним із методів комплексного оцінювання стану проекту є аналіз освоєного обсягу. Він порівнює заплановану вартість робіт, фактично виконаний обсяг і реальні витрати.

Основними величинами методу є:

- PV (Planned Value) — планова вартість робіт, які повинні бути виконані на дату контролю;
- EV (Earned Value) — планова вартість робіт, фактично виконаних на дату контролю;
- AC (Actual Cost) — фактичні витрати на виконану роботу;
- BAC (Budget at Completion) — затверджений бюджет проекту після завершення всіх робіт.

Взаємозв'язок основних показників показано на рисунку 7.2.

Initial Data	Variance Analysis	Forecasting
• Planned Value (PV) • Actual Costs (AC) • Earned Value (EV) • Budget at Completion (BAC)	• Cost Variance (CV) • Schedule Variance (SV) • Cost Performance Index (CPI) • Schedule Performance Index (SPI) • Percent Complete Index-Budget (PCIB) • Percent Complete Index-Cost (PCIC)	• Estimate to Complete (ETC) • Estimate At Completion (EAC) • Variance At Completion (VAC) • To Complete Performance Index (TCPI)

Рисунок 7.2 – Основні показники аналізу освоєного обсягу

На основі початкових величин розраховують відхилення та індекси виконання.

Відхилення від розкладу:

$$SV = EV - PV.$$

Якщо $SV < 0$, виконано менше робіт, ніж планувалося. Якщо $SV > 0$, виконано більше запланованого обсягу.

Індекс виконання розкладу:

$$SPI = \frac{EV}{PV}$$

Значення $SPI < 1$ свідчить про відставання, $SPI = 1$ — про виконання відповідно до плану, а $SPI > 1$ — про випередження плану.

Відхилення за вартістю:

$$CV = EV - AC.$$

Якщо $CV < 0$, виконані роботи коштували дорожче, ніж планувалося. Якщо $CV > 0$, фактичні витрати є нижчими за планові.

Індекс виконання вартості:

$$CPI = \frac{EV}{AC}$$

Значення $CPI < 1$ вказує на перевищення витрат, $CPI = 1$ — на відповідність бюджету, а $CPI > 1$ — на економніше використання коштів.

Основні формули та їх інтерпретацію наведено в таблиці 7.3.

Таблиця 7.3 – Показники аналізу освоєного обсягу

Показник	Формула	Інтерпретація
Відхилення від розкладу	$SV = EV - PV$	Від'ємне значення означає відставання
Індекс виконання розкладу	$SPI = EV / PV$	Значення менше 1 означає відставання
Відхилення за вартістю	$CV = EV - AC$	Від'ємне значення означає перевищення витрат
Індекс виконання вартості	$CPI = EV / AC$	Значення менше 1 означає перевищення бюджету
Прогноз вартості завершення	$EAC = BAC / CPI$	Очікувана загальна вартість за збереження поточної ефективності
Прогноз залишкових витрат	$ETC = EAC - AC$	Кошти, необхідні для завершення решти робіт
Прогнозоване відхилення бюджету	$VAC = BAC - EAC$	Від'ємне значення означає очікуване перевищення бюджету

Розглянемо стислий приклад. На дату контролю планова вартість робіт становить 80 000 грн, освоєний обсяг — 70 000 грн, а фактичні витрати — 90 000 грн.

Тоді:

$$SV = 70,000 - 80,000 = -10,000 \text{ грн}$$

$$SPI = \frac{70,000}{80,000} = 0,875$$

$$CV = 70,000 - 90,000 = -20,000 \text{ грн}$$

$$CPI = \frac{70,000}{90,000} \approx 0,78$$

Отже, проєкт відстає від розкладу та перевищує заплановані витрати. На кожную фактично витрачену гривню команда створює приблизно 0,78 грн запланованої вартості результату.

Показники освоєного обсягу необхідно інтерпретувати разом із календарним планом, критичним шляхом, досягнутими контрольними подіями та показниками якості. Високе значення виконаного обсягу не є позитивним результатом, якщо створені компоненти містять критичні дефекти або не відповідають вимогам.

Набір показників повинен відповідати масштабу і способу організації проєкту. Для невеликого ІТ-проєкту може бути достатньо відстеження виконаних завдань, контрольних подій, дефектів і фактичних витрат. Складний довготривалий проєкт потребує детальнішого кількісного аналізу.

7.6 Контроль результатів і завершення проєкту

Контроль результатів полягає у перевірці того, чи відповідають створені продукти встановленим вимогам, стандартам якості та критеріям приймання. Перевірка повинна здійснюватися не лише наприкінці проєкту, а протягом усього етапу реалізації.

Для контролю результатів використовуються:

- перевірки програмного коду;
- модульне, інтеграційне і системне тестування;
- тестування безпеки та продуктивності;
- демонстрації проміжних результатів;
- перевірка критеріїв готовності;
- матриця простежуваності вимог;
- контрольні списки якості;
- аудити й експертні перевірки;
- приймальне тестування користувачами.

Невідповідності повинні бути зареєстровані, класифіковані за пріоритетом, призначені відповідальним і повторно перевірені після виправлення. Критичні дефекти, які перешкоджають використанню продукту, повинні бути усунені до його приймання.

Завершення проєкту починається після створення та приймання запланованих результатів. Воно не зводиться лише до припинення виконання завдань. Необхідно формально підтвердити виконання зобов'язань, передати продукт, закрити документацію і звільнити ресурси.

Основні процеси та результати фази завершення показано на рисунку 7.3.

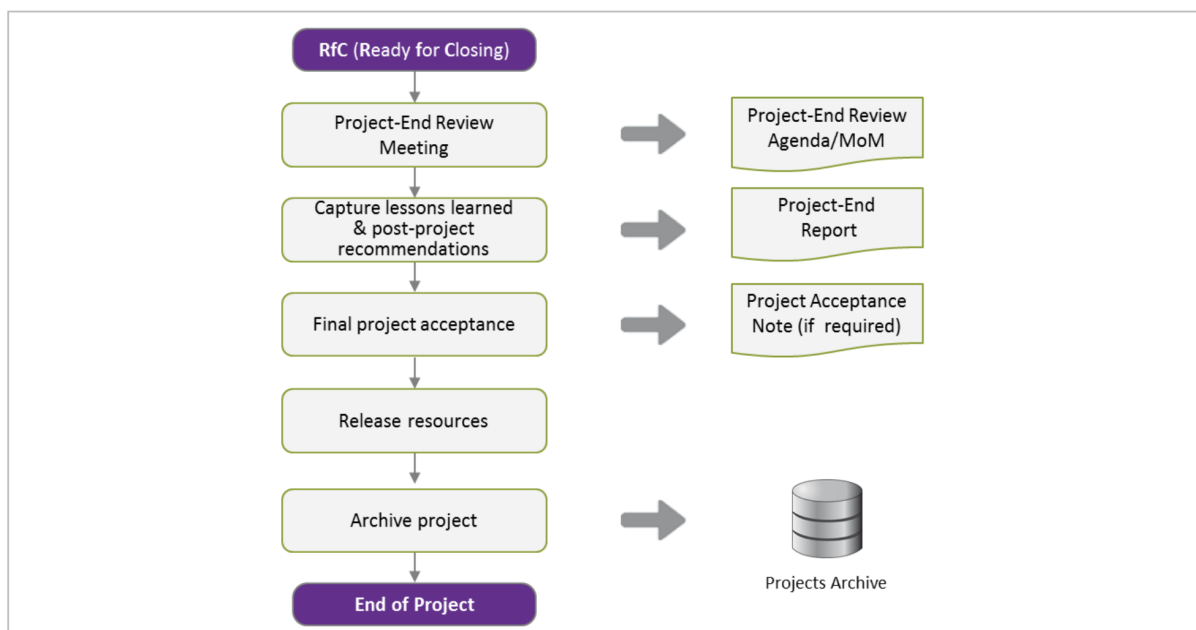


Рисунок 7.3 – Основні процеси та результати завершення проєкту

Основні дії завершення узагальнено в таблиці 7.4.

Таблиця 7.4 – Контрольний перелік завершення проєкту

Напря́м	Необхі́дні дії
Результати	Перевірити готовність, відповідність вимогам і формальне приймання
Зміст	Підтвердити виконання всіх погоджених робіт або зафіксувати невиконані
Розклад	Зафіксувати фактичні дати та причини відхилень
Бюджет	Закрити витрати, платежі, резерви й фінансові зобов'язання

Ризики та проблеми	Закрити завершені записи, а залишкові передати відповідальним
Закупівлі	Перевірити виконання договорів і формально закрити контракти
Передавання продукту	Передати продукт, дані, документацію та права доступу експлуатаційній команді
Досвід	Провести підсумковий аналіз і зафіксувати засвоєні уроки
Документація	Перевірити, упорядкувати й архівувати проєктні матеріали
Команда	Оцінити внесок учасників, надати зворотний зв'язок і звільнити ресурси
Підсумковий звіт	Узагальнити результати, відхилення, проблеми та рекомендації

Приймання результатів є однією з найважливіших дій завершення. Менеджер проєкту повинен перевірити виконання вимог, критеріїв приймання й технічних характеристик. Факт приймання необхідно підтвердити відповідним документом або погодженим електронним записом.

Якщо залишилися незначні недоліки, формується перелік незавершених дій із відповідальними та термінами. Наявність значних невиконаних робіт означає, що проєкт ще не готовий до закриття.

Перед закриттям необхідно переглянути ризики та проблеми. Неактуальні ризики закриваються, а залишкові ризики, пов'язані з експлуатацією продукту, передаються власнику продукту або операційній команді.

Під час підсумкової зустрічі аналізуються:

- досягнення початкових цілей;
- відповідність фактичного змісту затвердженому;
- виконання розкладу і бюджету;
- якість отриманих результатів;
- ефективність управління ризиками та змінами;
- робота команди й підрядників;
- задоволеність замовника;
- успішні практики;
- помилки та причини проблем;
- рекомендації для майбутніх проєктів.

Метою аналізу засвоєних уроків є не пошук винних, а збереження досвіду. Доцільно визначити, що було виконано добре, що потребує вдосконалення, чому виникли проблеми та які дії необхідно застосовувати в майбутньому.

Підсумковий звіт повинен містити оцінку результативності проєкту, фактичні показники, основні відхилення, перелік прийнятих змін, інформацію про якість, ризики, проблеми, команду, засвоєні уроки та подальші рекомендації.

Документація повинна бути впорядкована й розміщена у визначеному сховищі. До архіву включають затверджені плани, вимоги, проєктні рішення, журнали, протоколи, звіти, результати тестування, акти приймання та матеріали щодо засвоєних уроків.

Після приймання продукту команда проєкту може бути розформована, але життєвий цикл створеного продукту продовжується. Тому необхідно завчасно визначити порядок технічної підтримки, супроводу, моніторингу, резервного копіювання, оновлення та реагування на інциденти.

Проєкт вважається формально завершеним після приймання результатів, виконання договірних зобов'язань, передавання продукту, закриття документації, звільнення ресурсів і отримання остаточного погодження власника або спонсора проєкту.

Висновки

Реалізація проєкту є етапом, на якому команда виконує заплановані роботи та створює визначені результати. Водночас здійснюються моніторинг, контроль якості, управління ризиками, проблемами, змінами й очікуваннями зацікавлених сторін.

Робоче планування деталізує базовий план до рівня конкретних завдань, виконавців, термінів і критеріїв готовності. Воно повинно виконуватися за участю команди та регулярно уточнюватися на основі фактичної інформації.

Координація проєктних робіт передбачає розподіл завдань, забезпечення ресурсами, узгодження залежностей, перевірку проміжних результатів і своєчасне усунення перешкод. Важлива інформація про ризики, проблеми, зміни та рішення повинна фіксуватися у відповідних журналах.

Моніторинг забезпечує збирання інформації про стан проєкту, а контроль — порівняння фактичних результатів із базовим планом та реалізацію управлінських дій. Коригування базового плану допускається лише після погодження зміни.

Кількісне управління дає змогу об'єктивніше оцінювати стан проєкту та прогнозувати його завершення. Показники освоєного обсягу характеризують виконання розкладу і бюджету, але повинні аналізуватися разом із даними про якість, критичний шлях і готовність результатів.

Завершення проєкту включає формальне приймання результатів, закриття бюджету і контрактів, передавання продукту, аналіз досвіду, архівування документації та звільнення ресурсів. Належне завершення забезпечує перехід від проєктної діяльності до експлуатації створеного програмного продукту.

Лекція 8 Управління пріоритетами проєктів

Тема: Управління пріоритетами проєктів

Мета: Сформувати у студентів розуміння поняття пріоритету проєкту, його значення для розподілу обмежених ресурсів, ознайомити з основними чинниками, критеріями та методами визначення пріоритетів, а також практичними підходами до порівняння і відбору ІТ-проєктів.

План лекції

1. Поняття та значення пріоритету проєкту.
2. Чинники, що впливають на визначення пріоритетів.
3. Критерії оцінювання та порівняння ІТ-проєктів.
4. Основні методи визначення пріоритетів.
5. Багатокритеріальне оцінювання та ранжування проєктів.
6. Практичні приклади визначення пріоритетів.
7. Перегляд пріоритетів у разі зміни умов реалізації.
8. Висновки.

8.1. Поняття та значення пріоритету проєкту

Пріоритет проєкту — це визначений організацією рівень його важливості порівняно з іншими проєктами, програмами та поточною діяльністю. Пріоритет впливає на порядок реалізації проєктів, виділення фінансування, призначення фахівців, доступ до технічних ресурсів і рівень управлінської підтримки.

Організації зазвичай мають більше проєктних ідей, ніж можуть реалізувати. Обмеженнями виступають бюджет, час, кількість кваліфікованих працівників, технічна інфраструктура та спроможність організації одночасно впроваджувати зміни. Тому необхідно не лише оцінювати кожний проєкт окремо, а й порівнювати його з іншими можливими проєктами.

Високий пріоритет означає, що проєкт має суттєве значення для досягнення стратегічних цілей, виконання обов'язкових вимог або отримання важливих результатів. Такий проєкт зазвичай отримує ресурси раніше за проєкти з нижчим пріоритетом.

Пріоритет проєкту не слід ототожнювати з його терміновістю. Терміновий проєкт потребує швидкого реагування, але не завжди забезпечує значну довгострокову цінність. Водночас стратегічно важливий проєкт може не мати найближчого крайнього терміну, але визначати майбутній розвиток організації.

Управління пріоритетами виконує такі функції:

- забезпечує зв'язок проєктів зі стратегією організації;
- спрямовує ресурси на найважливіші ініціативи;
- запобігає одночасному запуску надмірної кількості проєктів;
- забезпечує прозорість управлінських рішень;
- дає змогу порівнювати різні типи проєктів;
- допомагає збалансувати користь, витрати й ризики;
- створює підстави для призупинення або припинення

неефективних проєктів.

Організації зазвичай реалізують стратегічні, операційні та обов'язкові проєкти. Стратегічні проєкти створюють нові продукти, послуги або конкурентні можливості. Операційні проєкти покращують наявні процеси, знижують витрати або підвищують якість. Обов'язкові проєкти виконуються для дотримання законодавчих, нормативних, безпекових або договірних вимог.

Визначення пріоритетів повинно починатися з аналізу стратегії організації. Цілі проєкту порівнюють із місією, стратегічними цілями, бюджетом, планами розвитку та ключовими показниками діяльності.

Одним зі способів поєднання стратегічних цілей і проєктних ініціатив є збалансована система показників, приклад якої наведено на рисунку 8.1.

Perspective	Objectives	Measures	Targets (2024)
<u>Stewardship</u> (Financial Strength into Perpetuity)	Increase smartphone sales	Number of smartphones sold	+1.5% of global shipments
	Offset advertising costs	Percent of users returning through direct channel after first purchase	5% of purchasers
	Increased margin on premium services	Increase the profit margin on premium data services	At least 3.2% margin
<u>Customers</u> (A uniquely satisfying experience)	Enhance customer shopping experience	Average session duration to purchase	Maintain the lead in online sales and increase gap between rivals
	Build brand loyalty	Number of returning users with a purchase	25%
<u>Process</u> (Efficient and effective Operations)	Develop a next-gen smartphone	Time to market with first product	Launch by January 2023
	Incorporate sustainability practices in manufacturing	Reduce waste from smart device production	Reduce by 10%
<u>People and Learning</u> (Be a great place to work)	Train sales team in new sales platform	Percentage of sales staff with certificate of completion	95%
	Improve employee benefits options	Add vision and dental services	70% election by EOY

Рисунок 8.1 – Приклад збалансованої системи стратегічних показників організації

Збалансована система показників дає змогу оцінити проєкт із декількох позицій: фінансової, клієнтської, процесної, кадрової та інноваційної. Завдяки цьому перевага не надається автоматично лише найбільш прибутковому проєкту.

8.2. Чинники, що впливають на визначення пріоритетів

Пріоритет формується під впливом сукупності стратегічних, фінансових, технічних, організаційних і зовнішніх чинників. Рішення не повинно ґрунтуватися лише на одному показнику, оскільки проєкт із

високою очікуваною користю може мати неприйнятний рівень ризику або потребувати недоступних ресурсів.

Основні чинники визначення пріоритету наведено в таблиці 8.1.

Таблиця 8.1 – Чинники визначення пріоритету ІТ-проєкту

Чинник	Питання для оцінювання
Стратегічна відповідність	Наскільки проєкт підтримує місію, цілі та пріоритети організації?
Обов'язковість	Чи пов'язаний проєкт із законодавчими, безпековими або договірними вимогами?
Очікувана користь	Які фінансові, операційні, соціальні або технологічні результати буде отримано?
Терміновість	До якої дати необхідно отримати результат і що станеться в разі затримки?
Вартість	Які початкові та подальші витрати потребує проєкт?
Ризик	Якою є ймовірність невдачі та можливий вплив негативних подій?
Технічна здійсненність	Чи доступні необхідні технології, інфраструктура та знання?
Ресурсна забезпеченість	Чи є доступними необхідні фахівці, обладнання, дані та фінансування?
Тривалість отримання користі	Коли організація почне отримувати результати від проєкту?
Залежності	Чи залежить проєкт від інших ініціатив або чи створює основу для них?
Вплив на користувачів	Скільки користувачів охоплює проєкт і наскільки суттєво змінює їхню роботу?
Репутаційний вплив	Як реалізація або відмова від проєкту вплине на репутацію організації?

Стратегічна відповідність є одним із найважливіших чинників. Проєкт може бути технічно цікавим і потенційно прибутковим, але не підтримувати актуальні цілі організації. У такому разі його реалізація відволікатиме ресурси від важливіших напрямів.

Обов'язкові проєкти можуть отримувати пріоритет незалежно від фінансової привабливості. Наприклад, оновлення системи захисту персональних даних може не створювати безпосереднього доходу, але бути

необхідним для дотримання нормативних вимог і запобігання значним втратам.

Під час оцінювання вартості необхідно враховувати не лише витрати на розроблення, а й сукупну вартість володіння. До неї входять витрати на ліцензії, хмарну інфраструктуру, технічну підтримку, оновлення, навчання користувачів, безпеку та подальший супровід.

Важливим є альтернативний дохід або альтернативна користь, яку організація втрачає через відмову від іншого проєкту. Якщо ресурси спрямовано на одну ініціативу, вони не можуть одночасно використовуватися в іншій. Ця втрачена можливість є альтернативною вартістю рішення.

Витрати, які вже здійснені й не можуть бути повернуті, не повинні визначати майбутній пріоритет. Рішення про продовження проєкту має ґрунтуватися на порівнянні майбутніх витрат, ризиків і користі, а не на бажанні виправдати вже витрачені кошти.

8.3. Критерії оцінювання та порівняння ІТ-проєктів

Критерій оцінювання — це ознака, за якою визначається цінність, здійсненність або ризик проєкту. Щоб порівняння було обґрунтованим, усі проєкти повинні оцінюватися за однаковими правилами.

Критерії мають бути:

- пов'язаними зі стратегічними цілями;
- зрозумілими для учасників оцінювання;
- вимірюваними або однозначно інтерпретованими;
- придатними для порівняння різних проєктів;
- незалежними настільки, наскільки це можливо;
- обмеженими до кількості, необхідної для прийняття рішення;
- стабільними протягом одного циклу оцінювання.

Критерії доцільно поділяти на обов'язкові та порівняльні. Обов'язкові критерії використовуються як початковий фільтр. Якщо проєкт не відповідає хоча б одній критично важливій умові, його не допускають до подальшого ранжування.

До обов'язкових критеріїв можуть належати:

- відповідність законодавству;
- узгодженість зі стратегією;
- наявність відповідального власника;
- технічна здійсненність;

- допустимий рівень ризику;
- наявність мінімально необхідного фінансування;
- можливість завершення до обов'язкової дати.

Порівняльні критерії використовуються для визначення відносної привабливості проєктів, які пройшли початковий відбір. Для ІТ-проєктів такими критеріями можуть бути очікувана економія, приріст доходу, підвищення якості обслуговування, скорочення часу виконання процесу, зменшення кількості помилок, рівень автоматизації та кількість охоплених користувачів.

Необхідно уникати подвійного врахування однієї переваги. Наприклад, критерії «зниження операційних витрат» і «підвищення фінансової ефективності» можуть фактично характеризувати однаковий результат. Їх одночасне використання необґрунтовано збільшить вплив фінансової складової.

Для кожного критерію потрібно визначити напрям оцінювання. Вищий дохід є позитивним результатом, а більша вартість або вищий ризик — негативним. У бальній моделі шкалу доцільно побудувати так, щоб більша оцінка завжди означала кращий результат. Тоді проєкт із низькою вартістю або низьким ризиком отримуватиме вищий бал.

8.4. Основні методи визначення пріоритетів

Для визначення пріоритетів використовують якісні, фінансові, багатокритеріальні та математичні методи. Організація може поєднувати декілька методів, оскільки жоден із них не враховує всі аспекти проєкту.

Основні методи наведено в таблиці 8.2.

Таблиця 8.2 – Методи визначення пріоритетів проєктів

Метод	Сутність	Основне обмеження
Експертне оцінювання	Пріоритет визначається на основі думок керівників і фахівців	Залежність від суб'єктивності експертів
Контрольний перелік	Перевіряється відповідність проєкту встановленим умовам	Не показує відносну важливість критеріїв
Проста бальна модель	Проекти отримують оцінки за однаковими критеріями	Усі критерії фактично вважаються однаково важливими
Зважена бальна модель	Оцінки множаться на ваги критеріїв і підсумовуються	Результат залежить від правильності ваг і оцінок

Період окупності	Визначається час повернення вкладених коштів	Не враховує вигоди після періоду окупності
Чиста приведена вартість	Майбутні грошові потоки приводяться до поточної вартості	Потребує прогнозу потоків і ставки дисконтування
Внутрішня норма дохідності	Визначається очікувана відсоткова дохідність	Може неправильно відображати масштаб користі
Співвідношення вигід і витрат	Порівнюється приведена вартість вигід і витрат	Складно точно оцінити нематеріальні вигоди
Попарне порівняння	Проекти або критерії послідовно порівнюються між собою	Кількість порівнянь швидко зростає
Оптимізація за обмеженнями	Обирається найкраща комбінація проектів за заданих ресурсних обмежень	Потребує формальної математичної моделі

Контрольний перелік є найпростішим методом. Для кожного проекту перевіряється виконання визначених умов. Проект, який відповідає більшій кількості критеріїв, може отримати вищий пріоритет. Однак такий метод не враховує різної важливості умов.

Зважена бальна модель дає змогу надати критеріям різну важливість. Вона робить процес оцінювання прозорішим, оскільки показує, чому певний проект отримав вищу підсумкову оцінку.

Приклад такої моделі з підручника показано на рисунку 8.2.

Criteria	Weight	Project A Score	Project A Value	Project B Score	Project B Value	Project C Score	Project C Value
Likely to produce spin-offs or other products	30	1	30	2	60	2	60
Likely to make \$30,000 or more in revenue per year	40	2	80	2	80	1	40
Doesn't require outside consultants	20	2	40	2	40	1	20
Likely to gain us good press	10	0	0	2	20	2	20
Total	100		150		200		140

Рисунок 8.2 – Приклад зваженої бальної моделі відбору проєктів

Фінансові методи оцінюють здатність проєкту створювати дохід, зменшувати витрати або забезпечувати інший економічний результат.

Період окупності за умови рівномірного щорічного ефекту можна визначити за формулою:

$$T_{\text{окупності}} = \frac{\text{початкові інвестиції}}{\text{щорічний чистий економічний ефект}}$$

Коротший період окупності зазвичай вважається кращим, але цей метод не враховує результатів, отриманих після повернення інвестицій, і зміну вартості грошей із часом.

Рентабельність інвестицій можна визначити за формулою:

$$ROI = \frac{\text{вигоди} - \text{витрати}}{\text{витрати}} \cdot 100\%$$

Чиста приведена вартість у загальному вигляді розраховується так:

$$NPV = \sum_{t=1}^n \frac{CF_t}{(1+r)^t} - I_0$$

де CF_t — грошовий потік у періоді (t);

r — ставка дисконтування;

I_0 — початкові інвестиції.

Додатне значення NPV свідчить, що очікувана приведена користь перевищує початкові витрати. За інших однакових умов вищий показник є кращим.

Співвідношення вигід і витрат визначається за формулою:

$$BCR = \frac{\text{приведена вартість вигід}}{\text{приведена вартість витрат}}$$

Значення $BCR > 1$ означає, що очікувані вигоди перевищують витрати.

Порівняння основних фінансових методів наведено на рисунку 8.3.

Payback Period	Shows the time it will take to recover our investment in the project but doesn't account for the time value of money. With payback calculations, a lower number is better.
Net Present Value (NPV)	Accounts for the time value of money and is expressed as a dollar figure. Higher values are better.
Internal Rate of Return (IRR)	Accounts for the time value of money and is expressed as a percentage. Higher percentages are better. IRR doesn't express the magnitude (amount of dollars) of the project return.
Benefit-Cost Ratio (BCR)	Accounts for the time value of money and is expressed as a ratio. Higher numbers are better. Like IRR, BCR doesn't express the magnitude (amount of dollars) of the project return.

Рисунок 8.3 – Основні фінансові методи оцінювання проєктів

Фінансові показники не повинні використовуватися ізольовано. Проєкт із нижчою фінансовою дохідністю може мати вищий пріоритет, якщо він забезпечує дотримання законодавства, захист критичних даних або створює технологічну основу для інших проєктів.

8.5. Багатокритеріальне оцінювання та ранжування проєктів

Багатокритеріальне оцінювання застосовується тоді, коли рішення залежить від декількох чинників, які мають різну важливість. Одним із найбільш зрозумілих методів є зважена бальна модель.

Процес її застосування включає такі кроки:

1. Визначення переліку проєктів для порівняння.
2. Перевірку обов'язкових умов.
3. Формування критеріїв оцінювання.
4. Визначення ваг критеріїв.

5. Установлення єдиної шкали.
6. Оцінювання кожного проєкту.
7. Обчислення зважених оцінок.
8. Ранжування проєктів.
9. Перевірку результатів і прийняття управлінського рішення.

Сума ваг повинна дорівнювати одиниці або 100 %:

$$\sum_{i=1}^n w_i = 1$$

Підсумкова оцінка проєкту розраховується за формулою:

$$S_j = \sum_{i=1}^n w_i r_{ij}$$

де (S_j) — підсумкова оцінка проєкту (j);

w_i — вага критерію i ;

r_{ij} — оцінка проєкту j за критерієм i .

Ваги повинні відображати актуальні стратегічні пріоритети організації. Якщо найважливішим напрямом є інформаційна безпека, критерії зменшення кіберризиків повинні мати більшу вагу. Якщо організація прагне швидко вивести новий продукт на ринок, більшої ваги набувають час отримання користі та вплив на клієнтів.

Шкала оцінювання має бути чітко визначена. Наприклад:

- 1 бал — дуже низька відповідність;
- 2 бали — низька відповідність;
- 3 бали — середня відповідність;
- 4 бали — висока відповідність;
- 5 балів — дуже висока відповідність.

Для критеріїв ризику, вартості та складності шкалу потрібно інвертувати: низький ризик або низька вартість отримує високий бал, а високий ризик або висока вартість — низький.

До оцінювання доцільно залучати представників різних підрозділів: керівництва, фінансів, ІТ, інформаційної безпеки, операційної діяльності та користувачів. Це зменшує вплив інтересів одного підрозділу на підсумкове рішення.

Результат моделі не замінює управлінського рішення. Він забезпечує структуровану інформацію для обговорення. Після розрахунку необхідно

перевірити ресурсні обмеження, взаємозалежності проєктів і збалансованість портфеля.

8.6. Практичний приклад визначення пріоритетів

Припустимо, організація розглядає три ІТ-проєкти:

проєкт А — модернізація CRM-системи;

проєкт Б — упровадження платформи кібербезпеки;

проєкт В — розроблення мобільного застосунку для клієнтів.

Для порівняння визначено шість критеріїв. Кожний проєкт оцінюється за шкалою від 1 до 5. Розрахунок наведено в таблиці 8.3.

Таблиця 8.3 – Приклад багатокритеріального оцінювання ІТ-проєктів

Критерій	Вага	Проєкт А: оцінка	Проєкт А: результат	Проєкт Б: оцінка	Проєкт Б: результат	Проєкт В: оцінка	Проєкт В: результат
Стратегічна відповідність	0,25	5	1,25	4	1	4	1
Очікувана користь	0,2	4	0,8	3	0,6	5	1
Терміновість	0,2	3	0,6	5	1	2	0,4
Здійсненність і рівень ризику	0,15	4	0,6	4	0,6	3	0,45
Доступність ресурсів	0,1	4	0,4	3	0,3	4	0,4
Швидкість отримання користі	0,1	4	0,4	4	0,4	5	0,5
Підсумкова оцінка	1		4,05		3,9		3,75

Для проєкту А підсумкова оцінка розраховується так:

$$S_A = 0,25 \cdot 5 + 0,20 \cdot 4 + 0,20 \cdot 3 + 0,15 \cdot 4 + 0,10 \cdot 4 + 0,10 \cdot 4 = 4,05$$

Результати ранжування:

Проєкт А — 4,05 бала.

Проєкт Б — 3,90 бала.

Проєкт В — 3,75 бала.

За результатами моделі найвищий пріоритет має модернізація CRM-системи. Вона найкраще відповідає стратегії організації, має високу очікувану користь і може бути реалізована з прийнятним рівнем ризику.

Платформа кібербезпеки посідає друге місце завдяки високій терміновості. Якщо її реалізація стане обов'язковою через нові нормативні вимоги або критичну загрозу, вона може перейти на перше місце незалежно від підсумкової бальної оцінки.

Мобільний застосунок має найвищу оцінку очікуваної користі та швидкості її отримання, але поступається іншим проектам через нижчу терміновість і вищий рівень технічної невизначеності.

Після ранжування необхідно перевірити, чи можуть проекти виконуватися паралельно. Якщо всі три потребують одних і тих самих архітекторів або фахівців із безпеки, організація повинна визначити послідовність запуску або змінити ресурсний план.

Доцільно також провести аналіз чутливості. Для цього ваги або оцінки незначно змінюють і перевіряють, чи залишається порядок проєктів незмінним. Якщо невелика зміна ваг повністю змінює ранжування, результат моделі є нестійким і потребує додаткового обговорення.

8.7. Перегляд пріоритетів у разі зміни умов реалізації

Пріоритет проєкту не є незмінним протягом усього життєвого циклу. Зовнішні та внутрішні умови можуть змінитися, тому організація повинна періодично переглядати актуальність проєктів.

Причинами перегляду можуть бути:

- зміна стратегії організації;
- прийняття нових законодавчих або нормативних вимог;
- скорочення чи збільшення бюджету;
- поява нової технології;
- зміна ринкових умов;
- виникнення критичної загрози;
- зміна потреб користувачів;
- втрата ключових фахівців;
- істотне збільшення прогнозованої вартості;
- відставання від розкладу;
- погіршення очікуваної користі;
- зміна залежностей між проєктами;
- поява нового проєкту з вищою цінністю.

Перегляд може здійснюватися через установлені проміжки часу, наприклад щоквартально, а також після завершення основних фаз або досягнення контрольних подій. Позаплановий перегляд проводиться в разі суттєвої зміни умов.

Під час перегляду необхідно використовувати актуальні оцінки майбутніх витрат, строків, користі та ризиків. Уже понесені витрати не повинні автоматично бути аргументом на користь продовження проєкту.

За результатами перегляду може бути прийнято одне з таких рішень:

- продовжити проєкт без змін;
- підвищити його пріоритет;
- знизити пріоритет;
- змінити зміст або спосіб реалізації;
- прискорити виконання;
- тимчасово призупинити;
- об'єднати з іншим проєктом;
- припинити реалізацію.

Зміна пріоритету повинна бути обґрунтована, погоджена уповноваженим органом і повідомлена всім зацікавленим сторонам. Необхідно також оновити ресурсні плани, розклади, бюджети та залежності між проєктами.

Надто часті зміни пріоритетів негативно впливають на продуктивність. Команди витрачають час на перемикання між завданнями, незавершена робота накопичується, а відповідальність за результати послаблюється. Тому організація повинна визначити пороги, за яких зміна пріоритету є обґрунтованою.

Управління пріоритетами має здійснюватися на рівні всього портфеля. Високі оцінки окремих проєктів не гарантують, що їх одночасна реалізація сформує оптимальний портфель. Необхідно забезпечити баланс між стратегічними, операційними й обов'язковими проєктами, короткостроковими та довгостроковими результатами, високоризиковими й стабільними ініціативами.

Висновки

Пріоритет проєкту характеризує його відносну важливість для організації та визначає порядок надання фінансових, людських і технічних ресурсів. Управління пріоритетами необхідне через обмеженість ресурсів і

наявність більшої кількості проєктних ідей, ніж організація може реалізувати.

Основою визначення пріоритетів є відповідність проєкту стратегії. Додатково враховуються обов'язковість, очікувана користь, терміновість, вартість, ризик, технічна здійсненність, доступність ресурсів, залежності й час отримання результату.

Критерії оцінювання повинні бути зрозумілими, вимірюваними та однаково застосовуватися до всіх проєктів. Обов'язкові критерії використовуються як початковий фільтр, а порівняльні — для подальшого ранжування.

Для визначення пріоритетів застосовуються контрольні переліки, експертні оцінки, фінансові показники, зважені бальні моделі, попарне порівняння й оптимізація за обмеженнями. Найбільш обґрунтований результат зазвичай забезпечує поєднання декількох методів.

Зважена бальна модель дає змогу врахувати різну важливість критеріїв і зробити процес прийняття рішень прозорішим. Водночас її результат залежить від правильності обраних ваг, шкал і експертних оцінок.

Пріоритети необхідно регулярно переглядати у разі зміни стратегії, бюджету, ризиків, ресурсів, технологій або очікуваної користі. Рішення про продовження проєкту має ґрунтуватися на майбутніх витратах і результатах, а не на коштах, уже витрачених у минулому.

Ефективне управління пріоритетами забезпечує спрямування ресурсів на проєкти, які створюють найбільшу цінність і найкраще підтримують довгострокові цілі організації.

Лекція 9 Проект і організаційна структура компанії

Тема: Проект і організаційна структура компанії

Мета: Сформувати у студентів розуміння взаємозв'язку між проектом та організаційною структурою компанії, ознайомити з основними типами організаційних структур, особливостями розподілу повноважень і відповідальності та їхнім впливом на управління ІТ-проектами.

План лекції

1. Поняття проекту та організаційної структури компанії.
2. Роль організаційної структури в управлінні проектами.
3. Функціональна організаційна структура.
4. Матрична організаційна структура.
5. Проектна організаційна структура.
6. Вплив організаційної структури на повноваження керівника та роботу команди.
7. Практичні приклади застосування різних організаційних структур.
8. Висновки.

9.1. Поняття проекту та організаційної структури компанії

Проект є тимчасовою діяльністю, спрямованою на створення унікального продукту, послуги або результату. Він має визначені цілі, початок і завершення, обмежені ресурси, а також установлені вимоги до змісту, термінів, вартості та якості.

Організація, на відміну від проекту, зазвичай створюється для довготривалої діяльності. Вона виконує регулярні операції, підтримує бізнес-процеси, розподіляє ресурси та забезпечує досягнення стратегічних цілей.

Організаційна структура — це система розподілу завдань, повноважень, відповідальності та зв'язків між підрозділами й працівниками компанії. Вона визначає:

- які підрозділи існують в організації;
- які функції виконує кожний підрозділ;
- кому підпорядковуються працівники;
- хто має право приймати рішення;
- хто розподіляє ресурси;
- якими шляхами передається інформація;

- хто відповідає за результати діяльності.

Проект реалізується всередині організаційного середовища та використовує його ресурси. Навіть якщо для проекту створюється окрема команда, вона залежить від правил, культури, фінансових процедур, кадрової політики, технічної інфраструктури й системи управління компанії.

Організаційна структура може підтримувати проектну діяльність або створювати для неї додаткові перешкоди. Наприклад, якщо всі рішення повинні послідовно проходити через декілька рівнів управління, швидкість реагування проекту зменшується. Якщо менеджер проекту має достатні повноваження, він може швидше розподіляти ресурси та приймати поточні рішення.

Організаційну структуру не слід розглядати лише як схему посад. Формальна структура показує встановлені зв'язки підпорядкування, а реальна взаємодія також залежить від організаційної культури, особистого авторитету, професійних зв'язків і неформальних способів комунікації.

9.2. Роль організаційної структури в управлінні проектами

Організаційна структура визначає місце проекту в компанії та впливає на спосіб його управління. Від структури залежить, хто затверджує проект, призначає менеджера, формує команду, контролює бюджет і приймає рішення щодо змін.

Основними характеристиками організаційної структури є повноваження, відповідальність і підзвітність.

Повноваження — це право приймати рішення, давати доручення та розподіляти ресурси.

Відповідальність — це обов'язок виконати визначену роботу або забезпечити отримання результату.

Підзвітність — це обов'язок пояснювати прийняті рішення, повідомляти про виконані дії та відповідати за їхні наслідки.

Повноваження і відповідальність повинні бути збалансованими. Якщо менеджер проекту відповідає за терміни й результати, але не може впливати на призначення ресурсів або пріоритети команди, його здатність управляти проектом є обмеженою.

Організаційна структура впливає на:

- рівень повноважень менеджера проекту;
- порядок формування команди;

- доступність працівників;
- контроль проєктного бюджету;
- швидкість прийняття рішень;
- порядок ескалації проблем;
- інтенсивність комунікації;
- розподіл відповідальності;
- лояльність учасників до проєкту або функціонального підрозділу;
- можливість професійного розвитку учасників;
- спосіб оцінювання роботи команди.

Основні типи організаційних структур утворюють певний континуум: від функціональної структури, де менеджер проєкту має мінімальні повноваження, до проєктної структури, де він практично повністю управляє ресурсами, бюджетом і командою. Цю залежність показано на рисунку 9.1.

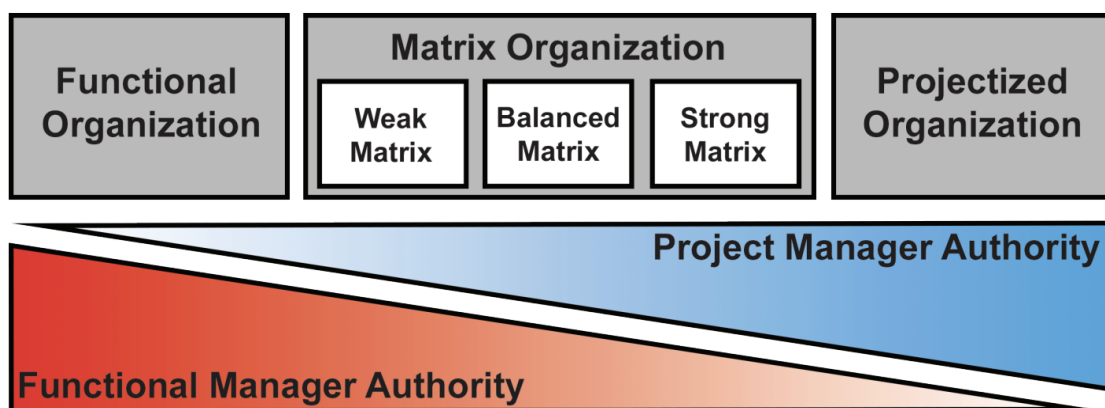


Рисунок 9.1 – Повноваження менеджера залежно від організаційної структури

В організації також можуть діяти керівний орган проєкту та офіс управління проєктами. Керівний орган приймає стратегічні рішення, погоджує значні зміни, вирішує ескаловані проблеми й контролює досягнення цілей. Офіс управління проєктами може розробляти стандарти, надавати шаблони, підтримувати інформаційні системи, навчати менеджерів і координувати використання ресурсів між проєктами.

Вибір структури залежить від виду діяльності компанії, кількості одночасних проєктів, їхньої складності, доступності спеціалістів і необхідної швидкості реагування.

9.3. Функціональна організаційна структура

У функціональній структурі працівники групуються за спеціалізацією. Окремі підрозділи можуть відповідати за розроблення програмного забезпечення, тестування, інформаційну безпеку, фінанси, маркетинг, закупівлі, кадрове забезпечення та інші напрями.

Кожний працівник підпорядковується функціональному керівнику, який визначає його завдання, професійні стандарти, навантаження та розвиток. Ресурси належать функціональним підрозділам і надаються проектам за погодженням із їхніми керівниками.

Перевагами функціональної структури є:

- концентрація спеціалістів одного профілю;
- обмін професійними знаннями;
- єдині технічні стандарти;
- стабільний професійний розвиток;
- ефективне використання вузькопрофільних фахівців;
- зрозумілий порядок адміністративного підпорядкування;
- збереження накопиченої експертизи в підрозділах;
- безперервність операційної діяльності.

У такій структурі тестувальники можуть працювати в окремому підрозділі забезпечення якості, інженери інформаційної безпеки — у службі безпеки, а програмісти — у відділі розроблення. Це дає змогу підтримувати професійні стандарти та використовувати фахівців у декількох проектах.

Основними недоліками функціональної структури є:

- слабка орієнтація на загальний результат проекту;
- повільна комунікація між підрозділами;
- виникнення функціональних «сховищ» інформації;
- різні пріоритети підрозділів;
- складність швидкого перерозподілу ресурсів;
- обмежені повноваження менеджера проекту;
- тривалий процес погодження рішень;
- складність визначення відповідального за кінцевий результат.

Розміщення менеджера і команди проекту у функціональній організації показано на рисунку 9.2.

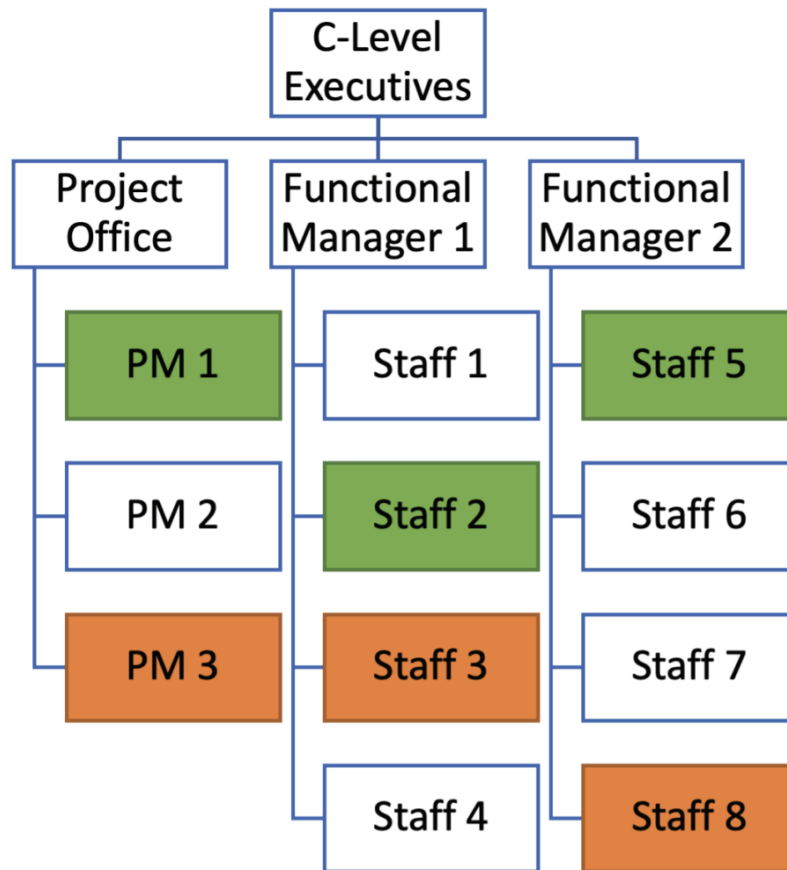


Рисунок 9.2 – Менеджер і команда проєкту у функціональній організації

У функціональній структурі менеджер проєкту може виконувати роль координатора. Він формує розклад, збирає інформацію, організовує зустрічі та контролює взаємозв'язки між роботами, але не має прямої влади над учасниками.

Успішність проєкту значною мірою залежить від співпраці менеджера проєкту з функціональними керівниками. Для залучення працівника необхідно погодити обсяг його участі, терміни, пріоритети та критерії оцінювання роботи.

Конфлікт виникає, якщо функціональний керівник вимагає виконання поточних операцій, а менеджер проєкту — проєктних завдань. Без чіткого погодження пріоритетів працівник опиняється між суперечливими вимогами.

Функціональна структура підходить для організацій зі стабільними повторюваними процесами та відносно невеликою кількістю міжфункціональних проєктів. Вона також ефективна, коли професійна

спеціалізація і стандартизація важливіші за швидкість реалізації окремого проєкту.

9.4. Матрична організаційна структура

Матрична структура поєднує ознаки функціональної та проєктної організації. Працівник залишається членом функціонального підрозділу, але одночасно залучається до одного або декількох проєктів.

У такій структурі функціональний керівник і менеджер проєкту спільно управляють ресурсами. Функціональний керівник відповідає за професійну компетентність працівників, технічні стандарти та кадрові питання. Менеджер проєкту відповідає за зміст, терміни, бюджет, координацію та досягнення проєктних цілей.

Матрична структура дає змогу:

- залучати спеціалістів із різних підрозділів;
- зберігати професійні центри компетентності;
- ефективно використовувати рідкісних фахівців;
- створювати міжфункціональні команди;
- прискорювати обмін інформацією;
- одночасно підтримувати операційну та проєктну діяльність;
- швидко формувати команди для нових проєктів.

Водночас подвійне підпорядкування може спричинити:

- суперечливі доручення;
- конфлікти пріоритетів;
- перевантаження працівників;
- боротьбу за обмежені ресурси;
- збільшення кількості комунікацій;
- нечіткий розподіл відповідальності;
- затримку рішень через необхідність погодження;
- різні оцінки результативності одного працівника.

Позитивні й негативні наслідки матричної структури наведено на рисунку 9.3.

Negative Outcomes	Positive Outcomes
- Employees may have two supervisors to whom they must report, breaking the solitary chain of command rule. - Employees must balance work between the project's needs and their functional unit. - Supervisors may find it more challenging to achieve a consistent rate of progress since employees are often pulled in different directions. - Costs and communication channels can increase.	- It significantly disrupts the communication silos of a functional organization, creating a more horizontal structure for teams and increasing the flow of information. - Allows people to concentrate on their specialty areas and bring those strengths to current projects.

Рисунок 9.3 – Позитивні та негативні наслідки матричної структури

Залежно від розподілу повноважень розрізняють слабку, збалансовану та сильну матрицю. Їхні характеристики наведено в таблиці 9.1.

Таблиця 9.1 – Типи матричної організаційної структури

Тип матриці	Повноваження менеджера проєкту	Роль функціонального керівника	Особливості
Слабка	Низькі	Домінує в управлінні ресурсами та працівниками	Менеджер проєкту переважно координує роботи
Збалансована	Середні	Розділяє повноваження з менеджером проєкту	Рішення потребують постійної взаємодії двох керівників
Сильна	Високі	Забезпечує професійну й ресурсну підтримку	Менеджер проєкту контролює більшість проєктних рішень

У слабкій матриці менеджер проєкту виконує функції координатора, а рішення про ресурси приймають функціональні керівники. За своїми характеристиками вона наближена до функціональної структури.

У збалансованій матриці повноваження розподіляються приблизно порівну. Такий варіант потребує чіткого розмежування відповідальності й постійної комунікації між керівниками.

У сильній матриці менеджер проєкту має значний вплив на ресурси, бюджет і пріоритети. Функціональні керівники забезпечують професійну підтримку, розвиток компетентностей і надання спеціалістів.

Для ефективної матричної структури необхідно встановити:

- хто визначає пріоритети працівника;
- хто погоджує його навантаження;
- хто оцінює результати роботи;
- хто приймає технічні рішення;
- хто відповідає за проєктні результати;
- як вирішуються ресурсні конфлікти;
- кому передаються проблеми, які не вдалося вирішити.

9.5. Проєктна організаційна структура

У проєктній, або проєктно-орієнтованій, структурі основна діяльність компанії організована навколо проєктів. Працівники призначаються до конкретної команди та безпосередньо підпорядковуються менеджеру проєкту.

Менеджер проєкту має високий рівень повноважень щодо:

- формування команди;
- розподілу робіт;
- використання ресурсів;
- управління бюджетом;
- установа пріоритетів;
- оцінювання результативності учасників;
- комунікації із замовником;
- погодження поточних рішень;
- управління ризиками та змінами.

Організацію команд у проєктній структурі показано на рисунку 9.4.

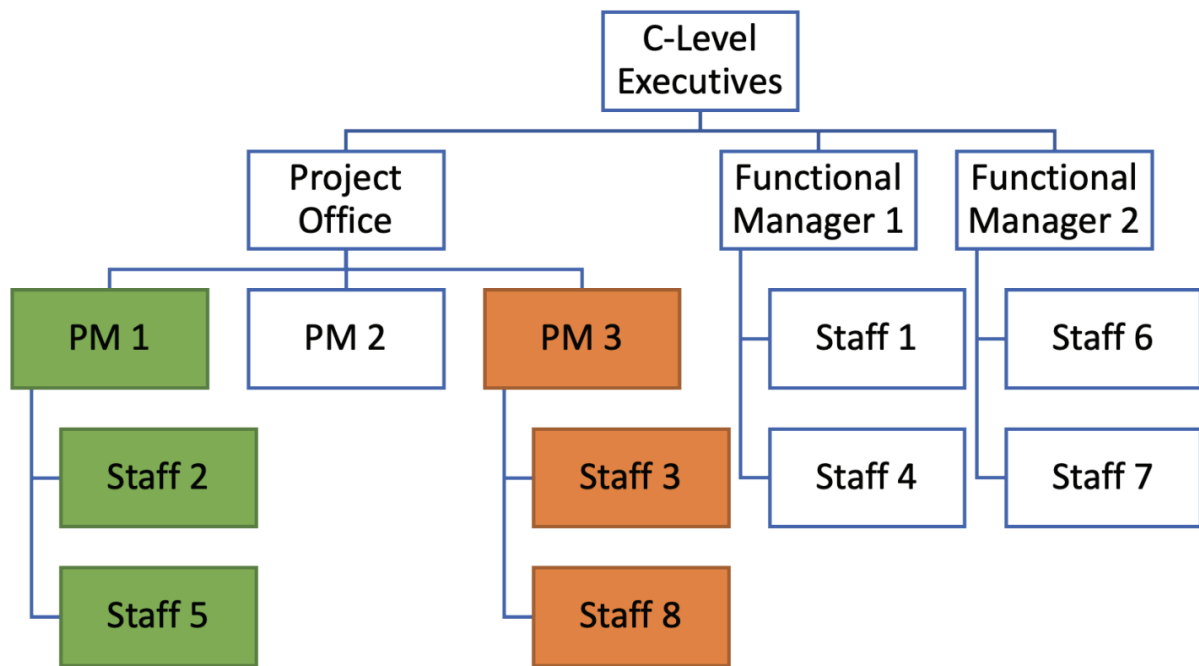


Рисунок 9.4 – Проектні команди у проектно-орієнтованій організації

Перевагами проектної структури є:

- чітка відповідальність за кінцевий результат;
- високий рівень повноважень менеджера;
- швидке прийняття рішень;
- безпосередня комунікація всередині команди;
- висока зосередженість на цілях проекту;
- можливість швидко реагувати на зміни;
- сильна командна ідентичність;
- простіший контроль ресурсів і бюджету.

До недоліків належать:

- дублювання спеціалістів і технічних ресурсів у різних проектах;
- слабший обмін професійними знаннями між командами;
- можливі відмінності у стандартах роботи;
- нестабільність зайнятості після завершення проекту;
- складність повернення працівників до функціональних підрозділів;
- конкуренція між проектами;
- орієнтація на короткостроковий проектний результат замість довгострокового розвитку компетентностей.

Проектна структура найбільш придатна для організацій, основна діяльність яких полягає у виконанні великих, складних або унікальних проєктів. До них можуть належати компанії, що розробляють програмні рішення на замовлення, системні інтегратори, консалтингові організації та науково-дослідні центри.

Навіть у функціональній компанії для критично важливого проєкту може бути створена окрема виділена команда. Це дає їй переваги проєктної структури без повної реорганізації компанії. Однак після завершення такого проєкту можуть виникнути складнощі з поверненням учасників до функціональних підрозділів.

9.6. Вплив організаційної структури на повноваження керівника та роботу команди

Організаційна структура визначає не лише формальні повноваження менеджера, а й практичні умови роботи команди. Основні відмінності між структурами наведено в таблиці 9.2.

Таблиця 9.2 – Порівняння організаційних структур

Характеристика	Функціональна	Матрична	Проектна
Повноваження менеджера проєкту	Низькі або відсутні	Від низьких до високих	Високі
Основний керівник працівника	Функціональний керівник	Функціональний і проєктний керівники	Менеджер проєкту
Контроль бюджету	Функціональний керівник	Спільний або менеджер проєкту	Менеджер проєкту
Доступність ресурсів	Обмежена	Середня	Висока
Зосередженість команди на проєкті	Низька	Середня	Висока
Швидкість прийняття рішень	Низька	Середня	Висока
Збереження професійної спеціалізації	Високе	Високе	Середнє

Імовірність конфлікту пріоритетів	Висока між підрозділами	Висока через подвійне підпорядкування	Нижча всередині проєкту
Комунікація між функціями	Переважно через керівників	Безпосередня й через керівників	Переважно безпосередня
Стабільність роботи після завершення проєкту	Висока	Висока	Може бути низькою

У функціональній структурі менеджер проєкту часто змушений управляти через переговори та особистий вплив. Він не може самостійно вимагати від працівника змінити пріоритет або збільшити участь у проєкті.

У матричній структурі менеджер має формальні повноваження, але повинен погоджувати рішення з функціональними керівниками. Для нього особливо важливими є переговори, управління конфліктами та прозоре планування завантаження.

У проєктній структурі менеджер має найбільшу свободу, але водночас несе більшу відповідальність за використання ресурсів, бюджет, розвиток команди та кінцевий результат.

Структура впливає і на поведінку учасників. У функціональній організації людина насамперед ідентифікує себе з професійним підрозділом. У проєктній — із конкретним проєктом. У матричній структурі працівник одночасно належить до функціонального підрозділу та проєктної команди.

Незалежно від типу структури потрібно чітко визначити:

- ролі учасників;
- повноваження менеджера проєкту;
- правила призначення ресурсів;
- порядок установа пріоритетів;
- відповідального за бюджет;
- порядок оцінювання роботи;
- процедуру прийняття рішень;
- рівні ескалації;
- правила взаємодії з функціональними підрозділами.

Якщо менеджер відповідає за результат, але не має достатніх повноважень, організація повинна забезпечити механізм підтримки з боку спонсора або керівного органу проєкту.

9.7. Практичні приклади застосування різних організаційних структур

Функціональна структура може застосовуватися під час реалізації внутрішнього проєкту оновлення системи електронного документообігу університету. Працівники IT-відділу, бухгалтерії, кадрової служби й юридичного підрозділу продовжують підпорядковуватися своїм керівникам і періодично виконують проєктні завдання. Менеджер координує роботи, але призначення ресурсів погоджує з керівниками підрозділів.

Перевагою такого підходу є використання наявних спеціалістів без створення нової структури. Основним ризиком є те, що поточні обов'язки учасників можуть отримувати вищий пріоритет, ніж проєкт.

Матрична структура доцільна в IT-компанії, яка одночасно розробляє декілька програмних продуктів. Розробники, тестувальники, дизайнери й DevOps-інженери належать до відповідних професійних підрозділів, але призначаються до продуктових або проєктних команд.

Менеджер проєкту визначає, які результати та коли необхідно створити, а функціональний керівник відповідає за професійний розвиток фахівців і технічні стандарти. Така структура дає змогу використовувати рідкісних спеціалістів у декількох проєктах, але потребує узгодженого планування навантаження.

Проєктна структура може застосовуватися під час створення великої інформаційної системи для зовнішнього замовника. Для неї формується окрема команда, до якої входять менеджер, аналітики, архітектори, розробники, тестувальники, дизайнери та інженери з розгортання. Учасники працюють переважно або виключно над цим проєктом.

Менеджер самостійно розподіляє роботи й бюджет у межах погоджених повноважень. Команда швидко приймає рішення і безпосередньо взаємодіє із замовником. Після завершення проєкту необхідно завчасно спланувати переведення учасників до нових команд.

У межах однієї компанії можуть одночасно використовуватися різні структури. Поточні операції можуть виконуватися функціональними підрозділами, більшість проєктів — матричними командами, а для стратегічної ініціативи може бути створена окрема виділена команда.

Тому завдання полягає не у виборі структури, яка є найкращою за будь-яких умов, а у визначенні структури, яка відповідає виду діяльності,

складності проєкту, доступності ресурсів і необхідній швидкості прийняття рішень.

Висновки

Організаційна структура визначає розподіл завдань, повноважень, відповідальності та зв'язків між підрозділами й працівниками. Вона безпосередньо впливає на місце проєкту в компанії, доступність ресурсів і можливості менеджера проєкту.

У функціональній структурі працівники групуються за професійними напрямками та підпорядковуються функціональним керівникам. Така структура підтримує спеціалізацію і стабільність, але може уповільнювати міжфункціональні проєкти.

Матрична структура поєднує функціональне та проєктне управління. Вона забезпечує гнучке використання спеціалістів, але створює подвійне підпорядкування та ризик конфлікту пріоритетів. Залежно від розподілу повноважень матриця може бути слабкою, збалансованою або сильною.

У проєктній структурі учасники переважно працюють над конкретним проєктом і підпорядковуються його менеджеру. Це забезпечує високу швидкість прийняття рішень і зосередженість на результаті, але може спричинити дублювання ресурсів та невизначеність після завершення проєкту.

Організаційна структура повинна забезпечувати відповідність між відповідальністю менеджера та наданими йому повноваженнями. Якщо менеджер не має прямої влади над ресурсами, необхідними стають підтримка спонсора, переговори та чітка процедура ескалації.

Універсальної структури, придатної для всіх організацій і проєктів, не існує. Її вибір повинен враховувати характер діяльності компанії, кількість і складність проєктів, доступність спеціалістів, необхідність професійної спеціалізації та швидкість реалізації змін.

Лекція 10 Мотивація та ефективна взаємодія

Тема: Мотивація та ефективна взаємодія

Мета: Сформувати у студентів розуміння ролі мотивації та ефективної взаємодії в реалізації ІТ-проектів, ознайомити з основними видами мотивації, принципами організації комунікацій, особливостями співпраці учасників проекту, зацікавлених сторін і роботодавців.

План лекції

1. Поняття мотивації та ефективної взаємодії.
2. Внутрішня і зовнішня мотивація учасників проекту.
3. Методи мотивації проектної команди.
4. Вплив мотивації на продуктивність і результати проекту.
5. Особливості взаємодії учасників проекту.
6. Комунікація як основа командної роботи.
7. Взаємодія із зацікавленими сторонами та роботодавцями.
8. Зворотний зв'язок і попередження конфліктів.
9. Висновки.

10.1. Поняття мотивації та ефективної взаємодії

Результати ІТ-проекту залежать не лише від правильності планування, вибору технологій і наявності необхідних ресурсів. Навіть добре підготовлений проект може не досягти поставлених цілей, якщо його учасники не розуміють значення своєї роботи, не зацікавлені в результаті або не можуть ефективно взаємодіяти між собою.

Мотивація — це сукупність внутрішніх і зовнішніх чинників, які спонукають людину до діяльності, визначають її спрямованість, інтенсивність і наполегливість. У контексті управління ІТ-проектами мотивація показує, чому учасник команди готовий докладати зусиль, брати відповідальність, розв'язувати складні проблеми та підтримувати досягнення спільних цілей.

Керівник проекту не може безпосередньо «створити» мотивацію іншої людини. Він може сформулювати умови, за яких учасники команди:

- розуміють цілі й цінність проекту;
- бачать зв'язок між власною роботою та загальним результатом;
- мають необхідні повноваження і ресурси;
- отримують справедливую оцінку результатів;
- можуть професійно розвиватися;
- довіряють керівнику та іншим учасникам;

- не боятися повідомляти про проблеми і помилки;
- відчувати свою належність до команди.

Ефективна взаємодія — це узгоджена спільна діяльність учасників проєкту, під час якої вони своєчасно обмінюються інформацією, координують дії, приймають рішення, розв'язують проблеми та несуть відповідальність за досягнення спільного результату.

Взаємодія не обмежується проведенням нарад або обміном повідомленнями. Вона передбачає наявність спільного розуміння цілей, правил роботи, ролей, відповідальності, способів прийняття рішень і розв'язання конфліктів.

Необхідно розрізняти мотивацію, задоволеність роботою та залученість. Задоволений працівник може позитивно оцінювати умови праці, але не обов'язково докладати додаткових зусиль. Мотивований учасник має причини виконувати певну діяльність. Залучений учасник емоційно, інтелектуально і професійно пов'язує себе з роботою команди та активно підтримує її результати.

Ефективна проєктна команда характеризується не постійною відсутністю суперечностей, а здатністю конструктивно обговорювати різні погляди, своєчасно виявляти проблеми й узгоджувати рішення. Для цього необхідні довіра, відкритість і психологічна безпека.

Психологічна безпека означає, що учасник може поставити запитання, висловити сумнів, повідомити про помилку або запропонувати альтернативне рішення без страху приниження чи необґрунтованого покарання. У сфері розроблення програмного забезпечення це особливо важливо, оскільки приховування дефектів, ризиків і технічних проблем може призвести до значних втрат на пізніших етапах проєкту.

10.2. Внутрішня і зовнішня мотивація учасників проєкту

За джерелом виникнення розрізняють внутрішню та зовнішню мотивацію.

Внутрішня мотивація виникає тоді, коли сама діяльність має цінність для людини. Учасник працює не лише заради винагороди, а тому, що завдання є цікавим, складним, корисним або дає можливість досягти професійної майстерності.

Основними джерелами внутрішньої мотивації в ІТ-проєктах можуть бути:

- інтерес до предметної області або технології;

- складність і нестандартність завдання;
- можливість навчатися та розвивати компетентності;
- самостійність у виборі способу виконання роботи;
- розуміння суспільної або бізнесової цінності продукту;
- відчуття особистого внеску в результат;
- професійне визнання з боку колег;
- належність до сильної команди.

Зовнішня мотивація пов'язана з наслідками діяльності, які формуються середовищем або іншими людьми. До зовнішніх мотиваторів належать заробітна плата, премії, кар'єрне просування, формальний статус, додаткові пільги, публічне визнання, а також можливі санкції за невиконання обов'язків.

Основні характеристики внутрішньої і зовнішньої мотивації наведено в таблиці 10.1.

Таблиця 10.1 – Порівняння внутрішньої і зовнішньої мотивації

Ознака	Внутрішня мотивація	Зовнішня мотивація
Джерело	Інтерес, професійний розвиток, зміст і цінність роботи	Винагорода, статус, оцінка, кар'єрне просування або санкції
Орієнтація	На процес діяльності та особисту майстерність	На отримання визначеного результату або винагороди
Типовий результат	Ініціативність, творчість, відповідальність, прагнення до розвитку	Виконання встановлених вимог і досягнення формальних показників
Тривалість впливу	Часто зберігається протягом тривалого часу	Може зменшуватися після отримання винагороди
Основний ризик	Втрата інтересу через надмірний контроль або відсутність розвитку	Формальне виконання завдань, конкуренція за винагороди, маніпулювання показниками
Роль керівника	Забезпечити автономію, змістовність роботи й можливість розвитку	Установити справедливі, прозорі та зрозумілі правила винагороди

Внутрішня і зовнішня мотивація не суперечать одна одній. Учасник може одночасно отримувати справедливую матеріальну винагороду, цікавитися завданням і відчувати значення своєї роботи. Проблеми виникають тоді, коли зовнішні винагороди починають витісняти

внутрішній інтерес або стимулюють поведінку, яка шкодить загальному результату.

Наприклад, винагорода лише за кількість закритих завдань може спонукати учасників обирати прості роботи, поділяти одне завдання на надмірну кількість дрібних або нехтувати якістю. Тому система мотивації повинна бути пов'язана не тільки з обсягом виконаної роботи, а й із цінністю результату, якістю, командною взаємодією та дотриманням професійних принципів.

Підходи керівника до мотивації значною мірою залежать від того, як він сприймає учасників команди. Відповідно до теорій Х і Y, керівник може виходити з різних припущень.

Теорія Х передбачає, що працівники намагаються уникати роботи, не бажають брати відповідальність і потребують постійного контролю. Керівник, який дотримується такого погляду, схильний до мікроменеджменту, докладних вказівок і контролю кожної операції. Надмірне застосування цього підходу може знизити ініціативність і сформувати ситуацію, коли учасники справді перестають приймати самостійні рішення.

Теорія Y виходить із припущення, що за відповідних умов люди прагнуть якісно виконувати роботу, готові брати відповідальність і можуть працювати без постійного контролю. Такий керівник більше уваги приділяє довірі, делегуванню, розвитку компетентностей і контролю результатів, а не кожної окремої дії.

На практиці стиль управління повинен ураховувати зрілість команди, складність завдання, досвід учасників і рівень ризику. Новому працівникові може бути потрібна докладна інструкція, тоді як досвідченому фахівцеві надмірний контроль лише заважатиме.

10.3. Методи мотивації проєктної команди

Методи мотивації повинні відповідати особливостям проєкту, організаційній культурі, складу команди й індивідуальним потребам учасників. Універсальної системи, яка однаково ефективно діятиме на всіх людей, не існує.

До основних методів мотивації проєктної команди належать такі.

Забезпечення зрозумілої мети. Учасники повинні знати не лише те, яке завдання потрібно виконати, а й те, чому воно важливе. Розуміння

очікуваної цінності продукту підсилює відповідальність і допомагає приймати рішення в умовах неповної інформації.

Участь у плануванні та прийнятті рішень. Якщо учасники беруть участь у визначенні способів виконання робіт, оцінюванні завдань і формуванні плану, вони краще розуміють прийняті рішення та частіше відчують відповідальність за їх реалізацію.

Надання самостійності. Керівник повинен визначити очікуваний результат, обмеження та критерії якості, але не завжди має детально встановлювати спосіб виконання роботи. Делегування повноважень сприяє розвитку професійної відповідальності.

Справедливий розподіл навантаження. Постійне перевантаження найбільш результативних учасників поступово знижує їхню мотивацію. Розподіл робіт має враховувати компетентності, доступність, складність завдань і потребу в професійному розвитку.

Усунення перешкод. Мотиваційні промови не компенсують відсутності необхідних інструментів, доступів, вихідних даних або своєчасних рішень. Одним із головних завдань керівника є виявлення і видалення перешкод, які заважають команді працювати.

Визнання результатів. Визнання повинно бути своєчасним, конкретним і відповідати реальному внеску. Необхідно враховувати індивідуальні та культурні відмінності: одні учасники позитивно сприймають публічну подяку, тоді як інші надають перевагу особистому визнанню.

Професійний розвиток. До мотиваційних заходів можуть належати наставництво, навчання, складніші завдання, участь в архітектурних рішеннях, ротація ролей і можливість представити результат зацікавленим сторонам.

Підтримання командної ідентичності. Спільні цілі, правила роботи, взаємодопомога і колективне визнання результатів сприяють згуртованості команди. Водночас командна єдність не повинна перетворюватися на придушення критичних думок.

Мотивація також пов'язана зі здатністю керівника впливати на інших людей. У проєктах керівник не завжди має достатні формальні повноваження, тому часто повинен використовувати аргументацію, консультації, залучення до прийняття рішень і формування спільного бачення.

Основні тактики впливу доцільно показати на рисунку 10.1.



Рисунок 10.1 – Основні тактики впливу

До тактик впливу належать раціональне переконання, надихаюче звернення, консультація, особисте звернення, обмін, формування коаліції, тиск і посилення на формальні повноваження. У роботі з проєктною командою найбільш стійкий результат зазвичай забезпечують:

- аргументація на основі фактів і даних;
- пояснення цінності й очікуваного результату;
- залучення учасників до підготовки рішення;
- підтримка у виконанні складної роботи;
- чесне визнання внеску;
- узгодження взаємних зобов'язань.

Реакцією на вплив може бути опір, формальна згода або справжня прихильність до рішення. Тиск і постійне використання формальної влади можуть забезпечити короткострокове виконання вказівок, але рідко формують тривалу мотивацію. Метою керівника має бути не лише підпорядкування, а й усвідомлена підтримка спільного рішення.

Матеріальне стимулювання є важливим, але воно повинно відповідати таким принципам:

- критерії винагороди мають бути зрозумілими;
- оцінювання повинно бути справедливим;
- винагорода має бути пов'язана з результатами, на які людина справді могла впливати;
- індивідуальні стимули не повинні руйнувати командну співпрацю;
- досягнення результату не повинно виправдовувати порушення вимог до якості, безпеки або професійної етики.

Мотивація не може замінити належну організацію роботи. Якщо команда постійно отримує суперечливі вимоги, не має доступу до

необхідних ресурсів або змушена працювати за нереалістичним розкладом, окремі заохочення не усунуть основної причини проблеми.

10.4. Вплив мотивації на продуктивність і результати проєкту

Мотивація впливає на поведінку учасників, але її не можна безпосередньо виміряти одним показником. Вона проявляється через ставлення до роботи, готовність співпрацювати, відповідальність, якість рішень і реакцію на проблеми.

Високий рівень мотивації може сприяти:

- зосередженості на цілях проєкту;
- своєчасному виконанню зобов'язань;
- підвищенню якості результатів;
- ранньому повідомленню про ризики і проблеми;
- обміну знаннями;
- готовності допомагати іншим учасникам;
- пошуку способів удосконалення продукту і процесу;
- зменшенню плинності кадрів;
- відповідальному ставленню до замовника та користувачів.

Низька мотивація може проявлятися у формальному виконанні завдань, небажанні брати відповідальність, приховуванні проблем, збільшенні кількості дефектів, уникненні командних обговорень і зростанні конфліктності.

Характерні прояви мотиваційних проблем та можливі управлінські дії наведено в таблиці 10.2.

Таблиця 10.2 – Ознаки мотиваційних проблем у проєктній команді

Ознака	Можлива причина	Доцільна дія керівника
Учасники виконують лише прямо поставлені завдання	Відсутність самостійності або нерозуміння загальної мети	Пояснити цілі, делегувати рішення в межах визначених повноважень
Проблеми повідомляються із запізненням	Страх покарання або недовіра	Відокремлювати аналіз причин від пошуку винних, підтримувати відкритість
Знижується якість результатів	Перевантаження, поспіх або неправильні стимули	Переглянути навантаження, строки та критерії оцінювання
Виникає конкуренція між учасниками	Надмірний акцент на індивідуальних показниках	Поєднати індивідуальні та командні критерії

		результативності
Учасники уникають нарад і обговорень	Неефективний формат комунікації або відсутність впливу на рішення	Переглянути склад, тривалість і мету зустрічей
Зростає кількість звільнень або ознак виснаження	Тривале перевантаження, відсутність розвитку чи визнання	Переглянути робочий ритм, навантаження і можливості розвитку

Продуктивність не можна оцінювати лише кількістю створених рядків програмного коду, закритих завдань або відпрацьованих годин. Такі показники не відображають складності роботи, якості рішень, повторного використання компонентів і внеску в командну взаємодію.

Для оцінювання результатів доцільно використовувати поєднання показників:

- досягнення цілей і контрольних подій;
- прогнозованість виконання робіт;
- тривалість проходження завдань;
- кількість і критичність дефектів;
- обсяг повторної роботи;
- якість документації;
- рівень задоволеності користувачів;
- здатність команди обмінюватися знаннями;
- виконання погоджених командних правил.

Необхідно відрізнити короткострокове збільшення інтенсивності роботи від стійкої продуктивності. Понаднормова праця може тимчасово збільшити обсяг виконаних завдань, але її постійне застосування призводить до втоми, помилок, зниження якості та професійного вигорання.

Стійка продуктивність забезпечується реалістичним плануванням, стабільним робочим ритмом, чіткими пріоритетами, достатньою кількістю часу на перевірку якості та можливість відновлення. Відповідальність керівника полягає в тому, щоб не перетворювати надзвичайний режим роботи на постійну норму.

10.5. Особливості взаємодії учасників проєкту

Проектна команда складається з учасників, які мають різні компетентності, професійний досвід, інтереси, повноваження і способи роботи. В ІТ-проєкті можуть взаємодіяти бізнес-аналітики, архітектори,

розробники, тестувальники, дизайнери, фахівці з інформаційної безпеки, адміністратори, представники замовника та керівники функціональних підрозділів.

Ефективна взаємодія потребує узгодження таких складових:

- спільної мети;
- ролей і відповідальності;
- очікуваних результатів;
- залежностей між роботами;
- правил передавання результатів;
- способів прийняття рішень;
- порядку ескалації проблем;
- комунікаційних каналів;
- правил роботи з конфіденційною інформацією.

Важливим інструментом є командна угода — погоджений набір принципів і правил спільної роботи. У ній можуть бути визначені:

- години доступності учасників;
- очікуваний час відповіді на повідомлення;
- порядок проведення нарад;
- правила документування рішень;
- критерії готовності завдань;
- порядок перевірки результатів;
- правила надання зворотного зв'язку;
- способи розв'язання розбіжностей.

Особливу увагу необхідно приділяти точкам передавання робіт між учасниками. Наприклад, передавання функціональності від розробника до тестувальника має супроводжуватися визначеними критеріями готовності, описом реалізованих змін, результатами автоматичних перевірок і відомими обмеженнями. Якщо правила передавання не встановлені, виникають затримки, повторна робота і взаємні звинувачення.

Учасники проекту можуть взаємодіяти формально і неформально. Формальна взаємодія відбувається через затверджені документи, звіти, протоколи, наради та запити на зміни. Неформальна взаємодія охоплює короткі консультації, професійні обговорення і взаємодопомогу. Обидві форми важливі: формальна забезпечує відповідальність і простежуваність, а неформальна прискорює обмін знаннями та підтримує довіру.

У дистанційних і розподілених командах необхідно додатково враховувати:

- часові пояси;
- мовні й культурні відмінності;
- обмеження синхронних зустрічей;
- доступність інформації для відсутніх учасників;
- необхідність письмового документування рішень;
- ризик професійної ізоляції окремих працівників.

Довіра є основою командної взаємодії. За низького рівня довіри учасники починають приховувати інформацію, надмірно контролювати одне одного, уникати відповідальності й витратити час на захист власної позиції. За високого рівня довіри інформація поширюється швидше, а проблеми обговорюються раніше.

Довіра формується через послідовність дій, виконання обіцянок, чесність, компетентність, повагу та справедливість. Вона не виключає контролю, але змінює його характер: контроль спрямовується на результат і ризики, а не на постійне спостереження за кожною дією учасника.

10.6. Комунікація як основа командної роботи

Комунікація — це процес створення, передавання, отримання та інтерпретації інформації. У проєкті важливо не лише передати повідомлення, а й переконатися, що отримувач правильно його зрозумів і може виконати необхідні дії.

Комунікаційне повідомлення повинно відповідати на такі запитання:

- хто потребує інформації;
- яка інформація потрібна;
- для якого рішення або дії вона використовується;
- хто відповідає за її підготовку;
- коли і як часто її потрібно передавати;
- який канал є найбільш доцільним;
- у якому форматі вона має бути представлена;
- як буде підтверджено її отримання і розуміння.

Спосіб передавання інформації повинен відповідати змісту повідомлення. Складні, неоднозначні або конфліктні питання краще обговорювати під час особистої чи відеозустрічі. Рішення, вимоги, зобов'язання та зміни необхідно фіксувати письмово. Регулярні показники прогресу доцільно подавати у вигляді коротких звітів або інформаційних панелей.

Комунікації поділяють на синхронні й асинхронні. Синхронна комунікація відбувається за одночасної участі сторін: під час особистої зустрічі, телефонної розмови, відеоконференції або обміну миттєвими повідомленнями. Вона дає можливість швидко отримати реакцію, уточнити позиції та розв'язати неоднозначність.

Асинхронна комунікація не вимагає одночасної присутності учасників. До неї належать електронна пошта, документи, записи в системі управління завданнями, бази знань і коментарі до програмного коду. Вона забезпечує зручність, документування і можливість працювати в різних часових поясах.

Зі збільшенням кількості учасників зростає потенційна кількість комунікаційних зв'язків. Для команди з (n) учасників їхню максимальну кількість можна визначити за формулою:

$$C = \frac{n(n-1)}{2}$$

де C — кількість потенційних комунікаційних зв'язків.

Наприклад, для восьми учасників можливі 28 парних зв'язків. Це не означає, що всі учасники повинні постійно спілкуватися між собою. Формула показує, чому зі збільшенням команди зростає потреба в чітких ролях, визначених каналах і систематизації інформації.

Основним документом для організації обміну інформацією є план комунікацій. Приклад такого плану доцільно показати на рисунку 10.2.

Communication Type (What)	Method (How)	Frequency (When)	Goal and Content	Owner	Audience
Project status report	Email	Weekly	Review project status and discuss potential issues or delays	Project manager	Project team + project sponsor
Team standup	Meeting	Daily	Discuss what each team member did yesterday, what they'll do today, and any blockers	Project manager	Project team
Project review	Meeting	At milestones	Present project deliverables, gather feedback, and discuss next steps	Project manager	Project team + project sponsor
Post-mortem meeting	Meeting	At end of the project	Assess what did and did not work and discuss actionable takeaways. (Sometimes referred to as "lessons learned" or an "after-action review.")	Project manager	Project team
Task progress updates	Activity List	Daily	Share daily progress made on project tasks	Project manager	Project team

Рисунок 10.2 – Приклад плану комунікацій проекту

Адаптований приклад основних елементів плану комунікацій наведено в таблиці 10.3.

Таблиця 10.3 – Фрагмент плану комунікацій IT-проекту

Комунікація	Мета і зміст	Канал	Періодичність	Відповідальний	Отримувачі
Координація поточних робіт	Виконані й заплановані завдання, перешкоди	Коротка командна зустріч	Щодня або відповідно до робочого циклу	Керівник команди	Проектна команда
Звіт про стан проекту	Прогрес, строки, витрати, ризики, потрібні рішення	Інформацій на панель і короткий звіт	Щотижня	Керівник проекту	Спонсор, ключові стейкхолдери

Демонстрація результату	Перевірка створеної функціональності та отримання відгуків	Відеозустріч із демонстрацією	Наприкінці ітерації	Команда розроблення	Замовник, користувачі
Повідомлення про критичну проблему	Опис події, вплив, вжиті дії, необхідні рішення	Визначений канал інцидентів	Негайно після виявлення	Власник проблеми	Керівник проєкту, відповідальні сторони
Фіксація управлінського рішення	Рішення, обґрунтування, відповідальні та строки	Журнал рішень	Після прийняття рішення	Секретар зустрічі або керівник проєкту	Усі сторони, яких стосується рішення

Для зменшення інформаційного перевантаження доцільно:

- визначити єдине місце зберігання актуальної інформації;
- розділяти оперативні повідомлення і затверджені рішення;
- використовувати зрозумілі назви документів і версій;
- обмежувати склад зустрічей учасниками, потрібними для конкретного питання;
- установлювати мету й порядок денний наради;
- завершувати нараду переліком рішень, відповідальних і строків;
- регулярно переглядати актуальність звітів і комунікаційних заходів.

Комунікаційний план не є незмінним. Його потрібно переглядати після зміни складу команди, появи нових зацікавлених сторін, переходу до іншої фази проєкту або виявлення комунікаційних проблем.

10.7. Взаємодія із зацікавленими сторонами та роботодавцями

Зацікавлені сторони мають різні інтереси, рівні повноважень, очікування й інформаційні потреби. Спонсор зосереджується на цінності, витратах і стратегічних результатах. Користувачі оцінюють зручність і відповідність продукту їхнім потребам. Функціональні керівники контролюють доступність спеціалістів. Фахівці з безпеки або юридичні підрозділи перевіряють відповідність установленим вимогам.

Тому однаковий зміст і формат комунікації не можуть бути ефективними для всіх сторін. Керівник проєкту повинен адаптувати повідомлення до ролі отримувача, не спотворюючи факти.

Під час взаємодії із зацікавленими сторонами необхідно:

- визначити їхні інтереси й очікування;
- узгодити критерії успішності;
- установити формат і періодичність комунікацій;
- залучати відповідних осіб до прийняття рішень;
- своєчасно повідомляти про відхилення та ризики;
- пояснювати наслідки альтернативних варіантів;
- фіксувати погоджені рішення;
- контролювати зміни очікувань протягом проєкту.

Важливо не приховувати негативну інформацію. Якщо відхилення стало відомим, керівник повинен повідомити не тільки про саму проблему, а й про її причини, можливий вплив, варіанти реагування та рішення, яке необхідно отримати від зацікавленої сторони.

Взаємодія з роботодавцем може здійснюватися через спонсора, функціонального керівника, керівника підрозділу, HR-службу або керівництво організації. Така взаємодія охоплює:

- виділення й перерозподіл ресурсів;
- погодження повноважень учасників;
- визначення умов роботи;
- навчання і професійний розвиток;
- оцінювання результативності;
- вирішення питань навантаження;
- винагородження та кар'єрне просування;
- дотримання корпоративних правил.

Керівник проєкту повинен узгоджувати інтереси проєкту з інтересами організації. Наприклад, проєкт може потребувати повної зайнятості певного фахівця, але функціональний підрозділ може одночасно потребувати його участі в операційній діяльності. Таку суперечність потрібно розв'язувати через прозоре обговорення пріоритетів і наслідків, а не через неформальне перевантаження працівника.

Технічну інформацію необхідно перекладати мовою управлінських наслідків. Замість повідомлення лише про необхідність перероблення програмного компонента доцільно пояснити:

- яку проблему має поточне рішення;

- як вона впливає на користувачів, строки, витрати або безпеку;
- які варіанти дій можливі;
- які ресурси потрібні;
- які наслідки матиме відмова від змін.

Для пояснення причин управлінського рішення або необхідності певної ініціативи можна застосувати візуалізацію, запропоновану на рисунку 10.3.

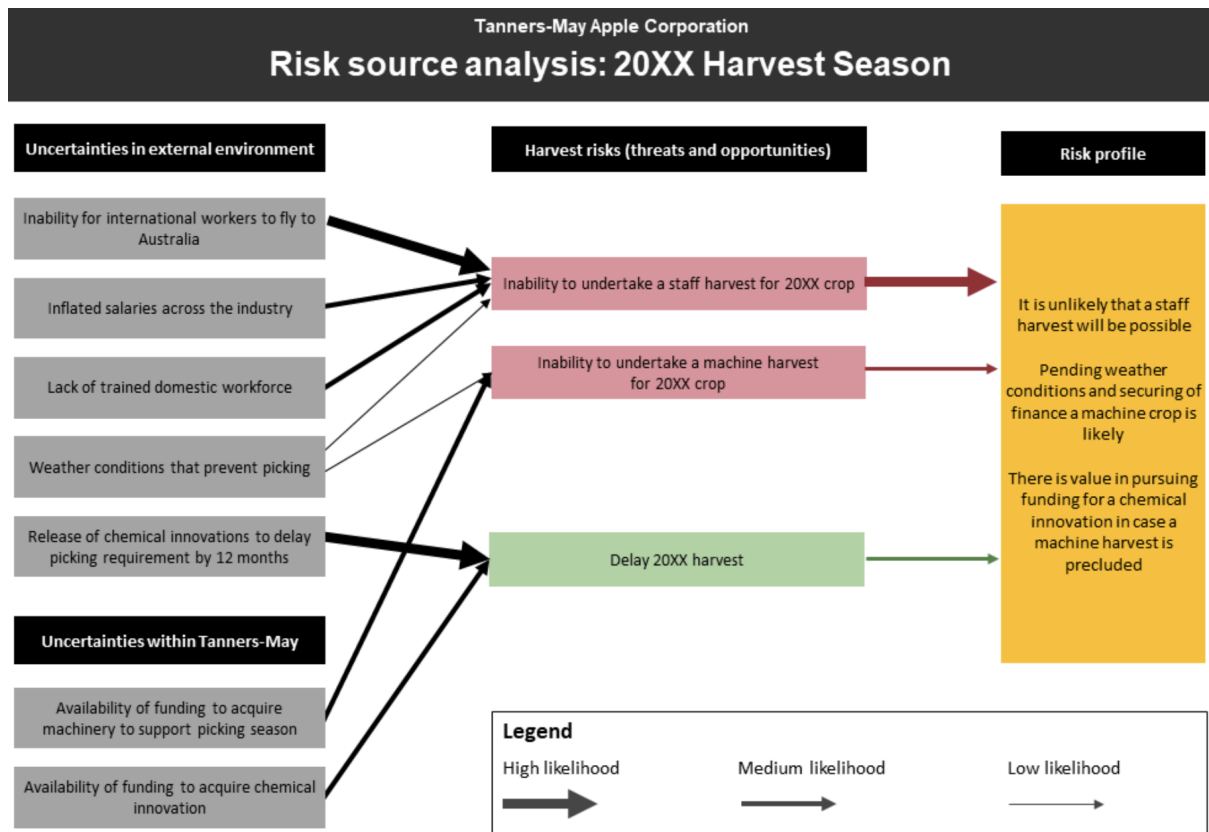


Рисунок 10.3 – Візуалізація причин і чинників управлінського рішення

Візуальні матеріали повинні спрощувати розуміння інформації, а не замінювати її аналіз. Для керівництва доцільно подавати лише ті дані, які допомагають оцінити стан проєкту або прийняти рішення. Надмірно складні схеми, велика кількість кольорів і дрібний текст знижують ефективність комунікації.

10.8. Зворотний зв'язок і попередження конфліктів

Зворотний зв'язок — це інформація про поведінку, виконану роботу або отриманий результат, яка допомагає учасникові зрозуміти сильні сторони, проблеми та можливості вдосконалення.

Ефективний зворотний зв'язок має бути:

- своєчасним;
- конкретним;
- заснованим на спостережуваних фактах;
- спрямованим на поведінку або результат, а не на особистість;
- зрозумілим;
- орієнтованим на подальші дії;
- двостороннім.

Доцільно використовувати таку послідовність:

1. Описати конкретну ситуацію.
2. Назвати спостережувану дію або результат.
3. Пояснити її вплив на роботу, команду або проєкт.
4. Вислухати позицію іншої сторони.
5. Узгодити подальші дії.
6. Перевірити виконання домовленостей.

Наприклад, замість загального твердження «Ви погано комунікуєте» потрібно вказати, яка інформація не була передана, коли це сталося, до яких наслідків призвело та як необхідно діяти надалі.

Позитивний зворотний зв'язок також має бути конкретним. Фраза «Добра робота» є приємною, але не пояснює, яку поведінку варто повторювати. Натомість конкретне визнання своєчасного виявлення ризику або якісного документування рішення допомагає закріпити корисну практику.

Конфлікт — це зіткнення різних цілей, інтересів, цінностей, способів роботи або поглядів. У проєктах конфлікти можуть виникати через:

- нечіткий розподіл відповідальності;
- суперечливі пріоритети;
- обмежені ресурси;
- нереалістичні строки;
- залежності між роботами;
- різні технічні підходи;
- недостатню або несвоєчасну інформацію;
- відмінності в очікуваннях;

- порушення домовленостей;
- особисті суперечності.

Не кожен конфлікт є негативним. Професійне обговорення різних технічних рішень може підвищити якість продукту. Небезпечним є перехід від обговорення проблеми до особистого протистояння.

Для попередження деструктивних конфліктів необхідно:

- чітко визначати ролі та межі відповідальності;
- погоджувати пріоритети;
- документувати важливі рішення;
- забезпечувати прозорий розподіл ресурсів;
- установлювати правила командної взаємодії;
- своєчасно обговорювати відхилення;
- не допускати накопичення невирішених проблем;
- відокремлювати факти від припущень;
- підтримувати повагу до професійних позицій.

Розв'язання конфлікту доцільно починати зі встановлення фактів і визначення спільної мети. Після цього сторони повинні висловити свої позиції, визначити основні інтереси, розглянути можливі варіанти та погодити подальші дії.

Керівник не повинен автоматично підтримувати сторону, яка має вищий формальний статус або висловлюється активніше. Рішення має враховувати цілі проєкту, фактичні дані, професійні вимоги і наслідки для команди.

Якщо конфлікт стосується порушення етичних норм, дискримінації, переслідування, безпеки або законодавчих вимог, його не можна розглядати лише як звичайну міжособистісну суперечність. У такому разі необхідно застосовувати встановлені організаційні процедури й залучати відповідальних осіб.

Після розв'язання конфлікту потрібно зафіксувати домовленості, відповідальних і строки, а також перевірити, чи було усунуто першопричину. Формальне припинення суперечки без зміни умов, які її спричинили, часто призводить до повторного конфлікту.

Висновки

Мотивація та ефективна взаємодія є важливими умовами успішної реалізації ІТ-проєкту. Мотивація визначає спрямованість, інтенсивність і

стійкість зусиль учасників, а ефективна взаємодія забезпечує узгодження їхніх дій і своєчасний обмін інформацією.

Внутрішня мотивація пов'язана зі змістом роботи, професійним розвитком, самостійністю і відчуттям особистого внеску. Зовнішня мотивація ґрунтується на винагородах, визнанні, статусі та інших наслідках діяльності. Ефективна система управління повинна поєднувати обидва види мотивації, не допускаючи, щоб зовнішні стимули погіршували якість або командну співпрацю.

До основних засобів підтримання мотивації належать зрозумілі цілі, участь у плануванні, делегування повноважень, справедливий розподіл навантаження, усунення перешкод, визнання результатів і можливості професійного розвитку.

Ефективна взаємодія потребує довіри, психологічної безпеки, чіткого розподілу відповідальності, погоджених правил і відповідних комунікаційних каналів. План комунікацій допомагає визначити, яка інформація потрібна різним зацікавленим сторонам, хто відповідає за її підготовку, коли та в якому форматі її необхідно передавати.

Зворотний зв'язок повинен бути своєчасним, конкретним і спрямованим на вдосконалення подальших дій. Конфлікти потрібно виявляти на ранніх етапах, відокремлювати професійні розбіжності від особистого протистояння та розв'язувати на основі фактів, спільних цілей і погоджених правил.

Отже, завдання керівника проєкту полягає не тільки в розподілі робіт і контролі строків. Він повинен формувати середовище, у якому учасники розуміють цінність проєкту, готові брати відповідальність, відкрито обговорюють проблеми та спільно працюють над досягненням запланованого результату.

Лекція 11 Головні ризики програмних проєктів та способи реагування

Тема: Головні ризики програмних проєктів та способи реагування

Мета: Поглибити знання студентів щодо найбільш поширених ризиків програмних проєктів, їхніх причин і можливих наслідків, сформуванню розуміння методів запобігання, зниження та контролю ризиків, а також практичного застосування стратегій реагування.

План лекції

1. Особливості ризиків програмних проєктів.
2. Ризики вимог і неконтрольованого розширення змісту.
3. Ризики оцінювання, термінів і ресурсного забезпечення.
4. Командні, організаційні та комунікаційні ризики.
5. Технічні ризики, ризики якості та інформаційної безпеки.
6. Способи запобігання та реагування на ризики.
7. Практичні приклади успішного управління ризиками.
8. Висновки.

11.1. Особливості ризиків програмних проєктів

Загальний процес управління ризиками, який охоплює їх ідентифікацію, аналіз, планування реагування, моніторинг і контроль, розглянуто в лекції 4. У цій лекції основна увага приділяється характерним ризиковим ситуаціям, що виникають під час створення програмного забезпечення, їхнім раннім ознакам і практичним способам реагування.

Програмні проєкти реалізуються в умовах значної невизначеності. На початку робіт замовник не завжди може повністю сформулювати потреби, команда може не мати досвіду застосування вибраної технології, а реальна складність інтеграції часто стає зрозумілою лише після створення прототипу.

До особливостей програмних проєктів, які збільшують ризики, належать:

- нематеріальний характер проміжних результатів;
- складність об'єктивного визначення фактичного прогресу;
- залежність результату від знань і досвіду окремих фахівців;
- висока взаємозалежність програмних компонентів;
- можливість змінювати вимоги протягом розроблення;
- складність точного оцінювання трудомісткості;

- залежність від зовнішніх бібліотек, платформ, сервісів і постачальників;
- швидка зміна технологій;
- накопичення технічного боргу;
- необхідність забезпечення якості, безпеки та захисту даних;
- складність перенесення програмного продукту в реальне середовище експлуатації.

Ризики програмного проєкту часто взаємопов'язані. Нечіткі вимоги спричиняють помилки в оцінюванні, помилки в оцінюванні — дефіцит часу, дефіцит часу — скорочення тестування, а недостатнє тестування — зростання кількості дефектів після випуску.

Тому важливо аналізувати не лише окрему ризикову подію, а й можливий ланцюг її наслідків.

Основні групи ризиків програмних проєктів:

- ризики вимог і змісту;
- ризики оцінювання та розкладу;
- ресурсні ризики;
- командні й організаційні ризики;
- комунікаційні ризики;
- технічні й інтеграційні ризики;
- ризики якості;
- ризики інформаційної безпеки;
- зовнішні та постачальницькі ризики;
- ризики впровадження й експлуатації.

Ризик існує доти, доки подія залишається невизначеною. Якщо вона вже настала, її необхідно розглядати як проблему та переходити до виконання запланованих дій.

11.2. Ризики вимог і неконтрольованого розширення змісту

Ризики вимог виникають через неповне, неоднозначне, суперечливе або нестабільне розуміння того, який програмний продукт потрібно створити.

До найбільш поширених належать:

- неповне визначення потреб замовника;
- різне розуміння вимог учасниками проєкту;
- суперечливі вимоги різних підрозділів;
- недостатнє залучення користувачів;

- відсутність критеріїв приймання;
- пропуск нефункціональних вимог;
- часта зміна пріоритетів;
- пізнє виявлення законодавчих або галузевих обмежень;
- неконтрольоване розширення змісту.

Неконтрольоване розширення змісту виникає, коли до проєкту додають нові функції, роботи або вимоги без оцінювання їхнього впливу та формального погодження змін. Окремі зміни можуть здаватися незначними, але їх накопичення збільшує трудомісткість, ускладнює архітектуру й відсуває дату завершення.

Джерелом розширення змісту може бути не лише замовник. Команда також може додавати технічно цікаві, але не обов'язкові функції, проводити надмірне вдосконалення окремих компонентів або створювати рішення, складніше за необхідне.

Основні ризики вимог, їхні ранні ознаки та способи реагування наведено в таблиці 11.1.

Таблиця 11.1 – Ризики вимог і змісту програмного проєкту

Ризик	Ранні ознаки	Запобіжні та коригувальні дії
Неповні вимоги	Під час розроблення постійно виникають нові запитання	Інтерв'ю, аналіз процесів, прототипування, уточнення сценаріїв використання
Неоднозначні вимоги	Розробники, тестувальники й замовник по-різному розуміють результат	Приклади, моделі, критерії приймання, спільний перегляд вимог
Суперечливі очікування	Різні представники замовника дають несумісні вказівки	Визначення уповноваженої особи, фіксація рішень, узгодження пріоритетів
Неформалізовані зміни	Нові функції додають без оновлення плану	Реєстрація запиту на зміну та аналіз впливу на терміни, ресурси і якість
Недостатня участь користувачів	Зворотний зв'язок надходить лише наприкінці розроблення	Регулярні демонстрації, раннє тестування прототипів, пілотне впровадження
Відсутність критеріїв приймання	Неможливо однозначно визначити готовність результату	Узгодження перевірюваних критеріїв до початку реалізації
Пропущені нефункціональні	Проблеми продуктивності, безпеки або зручності	Формування окремого переліку нефункціональних вимог і рання

вимоги	виявляють перед запуском	перевірка
--------	--------------------------	-----------

Реагування на ризики вимог має починатися до написання значного обсягу програмного коду. Чим пізніше виявлено неправильне розуміння потреби, тим дорожчим стає виправлення.

До основних способів запобігання належать:

- визначення відповідального представника замовника;
- установлення критеріїв готовності вимоги до реалізації;
- документування критеріїв приймання;
- використання прототипів і демонстрацій;
- пріоритизація функцій;
- простежуваність вимог до проєктних рішень і тестів;
- формальний аналіз впливу змін;
- регулярна перевірка відповідності результату потребам користувачів.

11.3. Ризики оцінювання, термінів і ресурсного забезпечення

Методи оцінювання трудомісткості й тривалості програмних проєктів, а також наслідки агресивного розкладу розглянуто в лекції 5. У цьому розділі увага зосереджується на ознаках того, що затверджена оцінка або ресурсний план стають нереалістичними.

Поширені причини помилок оцінювання:

- оцінювання за неповними вимогами;
 - використання однієї точної цифри замість діапазону;
 - ігнорування інтеграції, тестування, документування та розгортання;
 - неврахування залежностей між роботами;
 - відсутність досвіду роботи з вибраною технологією;
 - перенесення оцінок з іншого за складністю проєкту;
 - тиск керівництва або замовника;
 - неврахування відпусток, навчання, нарад та операційних обов'язків;
 - припущення про стовідсоткову доступність усіх фахівців.
- Ранніми ознаками проблем із розкладом і ресурсами можуть бути:
- повторне перенесення контрольних точок;
 - систематичне невиконання плану періоду;
 - збільшення кількості незавершених завдань;

- зростання часу виконання робіт;
- накопичення понаднормових годин;
- скорочення тестування або документування;
- часте перемикання працівників між проєктами;
- затримки через відсутність одного фахівця;
- залежність наступних робіт від непідтверджених результатів;
- зростання кількості термінових завдань.

До ресурсних ризиків належать:

- дефіцит потрібних компетентностей;
- недоступність критичного фахівця;
- звільнення учасника команди;
- конкуренція кількох проєктів за однакові ресурси;
- нестача обладнання або тестових середовищ;
- затримка закупівель;
- недостатній бюджет;
- недоступність представників замовника.

Основні способи реагування:

- декомпозиція робіт до рівня, придатного для оцінювання;
- використання історичних даних;
- залучення виконавців до формування оцінок;
- визначення діапазону та рівня невизначеності;
- резервування часу й ресурсів;
- окреме планування інтеграції, тестування та впровадження;
- аналіз залежностей і критичних обмежень;
- підготовка резервних виконавців;
- поширення знань між учасниками;
- раннє залучення зовнішніх фахівців;
- перегляд пріоритетів і змісту за неможливості дотриматися

всіх обмежень.

Залучення додаткових працівників до проєкту, який уже суттєво відстає, не завжди забезпечує прискорення. Нові учасники потребують ознайомлення з продуктом і процесами, а досвідчені працівники витрачають час на їх навчання. Тому ресурсне рішення необхідно приймати з урахуванням характеру робіт і можливості їх паралельного виконання.

11.4. Командні, організаційні та комунікаційні ризики

Принципи формування команди й розподілу ролей розглянуто в лекції 6, а вплив організаційної структури — у лекції 9. Для управління ризиками важливо виявити ситуації, у яких організація роботи може призвести до затримок, помилок або втрати контролю.

До командних ризиків належать:

- залежність від одного ключового фахівця;
- недостатня компетентність окремих учасників;
- висока плинність кадрів;
- конфлікти та низький рівень довіри;
- професійне виснаження;
- нерівномірний розподіл навантаження;
- відсутність відповідальності за результат;
- небажання повідомляти про проблеми.

Залежність від однієї людини особливо небезпечна, якщо лише вона розуміє архітектуру, володіє доступами до критичної системи або може виконувати розгортання продукту. Для зниження такого ризику застосовують спільний перегляд коду, документування, парну роботу, ротацію завдань і підготовку резервного виконавця.

Організаційні ризики можуть бути пов'язані з:

- нечітким розподілом повноважень;
- відсутністю активної підтримки спонсора;
- повільним прийняттям рішень;
- конкуренцією підрозділів за ресурси;
- частою зміною керівництва або пріоритетів;
- подвійним підпорядкуванням учасників;
- невизначеною процедурою ескалації;
- відсутністю відповідального за приймання результату.

Для їх зниження необхідно встановити ролі, межі повноважень, порядок погодження, процедуру ескалації та строки прийняття рішень. Важливі рішення потрібно фіксувати, щоб уникнути їх повторного обговорення або різного тлумачення.

Комунікаційні ризики виникають через:

- несвоєчасне передавання інформації;
- різне розуміння домовленостей;
- надмірну кількість каналів;
- відсутність протоколів рішень;

- приховування негативного стану проєкту;
- мовні та культурні відмінності;
- розподілену роботу в різних часових поясах;
- недостатню взаємодію між розробниками, тестувальниками й замовником.

Важливим способом запобігання є створення середовища, у якому учасники можуть своєчасно повідомляти про помилки й ризики без побоювання необґрунтованого покарання. Прихована проблема зазвичай завдає проєкту більшої шкоди, ніж помилка, виявлена і повідомлена на ранньому етапі.

11.5. Технічні ризики, ризики якості та інформаційної безпеки

Технічні ризики пов'язані з технологіями, архітектурою, інтеграціями, даними, інфраструктурою та умовами експлуатації програмного продукту.

Найбільш поширеними є:

- використання недостатньо перевіреної технології;
- невідповідність архітектури очікуваному навантаженню;
- складність інтеграції із зовнішніми системами;
- зміна або припинення підтримки зовнішнього сервісу;
- проблеми сумісності;
- низька якість успадкованого коду;
- помилки міграції даних;
- накопичення технічного боргу;
- відмінності між тестовим і робочим середовищами;
- неможливість безпечного повернення до попередньої версії.

Ризики якості виникають, якщо заходи перевірки планують запізно, критерії якості не визначено або команда скорочує тестування для дотримання термінів.

До їх ранніх ознак належать:

- зростання кількості повторно відкритих дефектів;
- збільшення часу виправлення;
- часті збої автоматичного збирання;
- нестабільність тестового середовища;
- накопичення неперевірених змін;
- відсутність регресійного тестування;
- значна кількість ручних операцій під час випуску;

- виявлення критичних помилок безпосередньо перед запуском.

Ризики інформаційної безпеки пов'язані з:

- неправомірним доступом;
- витоком або втратою даних;
- неналежним управлінням обліковими записами;
- уразливими зовнішніми компонентами;
- незахищеним передаванням інформації;
- неправильним зберіганням конфіденційних даних;
- відсутністю журналювання;
- невідповідністю законодавчим і організаційним вимогам;
- недостатньою готовністю до інцидентів.

Основні технічні ризики й відповідні способи їх зниження узагальнено в таблиці 11.2.

Таблиця 11.2 – Технічні ризики та способи їх зниження

Ризик	Спосіб ранньої перевірки	Запобіжні заходи
Нова або неперевірена технологія	Технічний прототип	Обмежене експериментальне впровадження, підготовка альтернативи
Недостатня продуктивність	Навантажувальне тестування	Визначення вимог до продуктивності, раннє профілювання
Проблеми інтеграції	Перевірка критичного сценарію обміну	Раннє інтеграційне тестування, заглушки, контроль версій інтерфейсу
Помилки міграції даних	Пробне перенесення вибірки	Перевірка правил перетворення, резервні копії, звіряння результатів
Низька якість коду	Аналіз коду та результатів тестування	Перегляд коду, автоматизовані тести, контроль технічного боргу
Нестабільне розгортання	Тестовий випуск у наближеному середовищі	Автоматизація розгортання, резервне копіювання, план повернення
Уразливі залежності	Перевірка компонентів і конфігурацій	Регулярне оновлення, контроль походження бібліотек, обмеження доступу
Витік конфіденційних даних	Аналіз загроз і тестування захисту	Мінімізація даних, шифрування, розмежування прав, журналювання

Для технічних ризиків особливо важливе раннє отримання доказів. Якщо невизначеність стосується критичної технології або інтеграції, її потрібно перевіряти на початку, а не залишати до завершального етапу.

11.6. Способи запобігання та реагування на ризики

Загальні стратегії реагування на загрози — уникнення, зниження, передавання, прийняття та ескалація — розглянуто в лекції 4. Практична цінність стратегії залежить від того, наскільки конкретно визначено порядок її реалізації.

Для критичного ризику недостатньо записати «знизити» або «контролювати». План реагування повинен визначати:

- власника ризику;
- конкретні запобіжні дії;
- відповідальних за виконання заходів;
- строки виконання;
- ресурси та резерви;
- ранні індикатори;
- умову активації реагування;
- дії після настання події;
- резервний варіант;
- порядок повідомлення та ескалації;
- залишковий ризик.

Необхідно розрізняти:

- запобіжні дії — виконуються до настання ризикової події та зменшують її ймовірність або вплив;
- план реагування — активується після появи визначеного сигналу;
- резервний план — застосовується, якщо основне реагування не дало очікуваного результату;
- обхідне рішення — формується для проблеми, яка настала без попередньо підготовленого плану.

Умову активації потрібно формулювати так, щоб її можна було однозначно перевірити. Наприклад, замість «постачальник затримує інтеграцію» варто зазначити: «тестовий інтерфейс не надано до визначеної дати» або «час відповіді зовнішнього сервісу перевищує погоджене значення».

Заходи реагування, які потребують часу, бюджету або участі команди, повинні бути включені до робочого плану чи переліку завдань. Інакше вони можуть залишитися лише записами в реєстрі ризиків.

Робота з ризиком завершується не після виконання запобіжної дії, а після повторного оцінювання залишкового ризику. Якщо його рівень залишається неприйнятним, потрібно запланувати додаткові заходи або змінити спосіб реалізації проєкту.

11.7. Практичні приклади успішного управління ризиками

Розглянемо проєкт створення вебплатформи для електронного запису користувачів на послуги. Система повинна обробляти персональні дані, взаємодіяти із зовнішнім сервісом ідентифікації та витримувати значне навантаження в періоди масової реєстрації.

Фрагмент плану реагування на ключові ризики такого проєкту наведено в таблиці 11.3.

Таблиця 11.3 – Приклад плану реагування на ризики програмного проєкту

Ризик	Рання ознака або умова активації	Запобіжні дії	Реагування
Затримка погодження вимог	Вимога не погоджена до початку запланованої реалізації	Призначення уповноваженого представника замовника, регулярний перегляд вимог	Перенести пов'язану функцію, виконувати незалежні роботи, ескалувати затримку
Недоступність зовнішнього сервісу	Тестовий доступ не надано до контрольної дати	Ранній запит доступу, створення програмної заглушки	Продовжити розроблення із заглушкою, активувати альтернативний спосіб ідентифікації
Недостатня продуктивність	Результати навантажувального тестування не відповідають вимогам	Ранній прототип, визначення цільових характеристик	Оптимізувати критичні операції, масштабувати інфраструктуру, обмежити некритичні функції
Відсутність ключового розробника	Фахівець недоступний довше встановленого періоду	Спільний перегляд коду, документація, резервний виконавець	Передати завдання резервному фахівцю, переглянути пріоритети

Виявлення критичної вразливості	Вразливість підтверджено під час перевірки	Аналіз загроз, перевірка залежностей, контроль доступу	Заблокувати випуск, усунути вразливість, повторити перевірку
Помилка під час випуску	Контрольні перевірки після розгортання не пройдено	Автоматизація випуску, резервна копія, тестове розгортання	Повернути попередню версію, відновити дані, провести аналіз причини

У цьому прикладі успішне управління полягає в тому, що критичні невизначеності перевіряються до завершального етапу. Інтеграцію перевіряють на початку, продуктивність — до запуску всієї функціональності, безпеку — протягом розроблення, а процедуру повернення версії готують до першого робочого випуску.

Якщо зовнішній сервіс не надано вчасно, команда не чекає бездіяльно, а використовує заглушку та продовжує незалежні роботи. Якщо навантажувальне тестування виявляє проблему, команда має час змінити архітектурне рішення. Якщо критичний фахівець стає недоступним, передавання роботи можливе завдяки завчасному поширенню знань.

Отже, успішне реагування полягає не у відсутності ризикових подій, а в готовності проєкту вчасно їх виявити, обмежити наслідки та продовжити досягнення запланованих результатів.

Висновки

Програмні проєкти характеризуються високою невизначеністю вимог, складністю оцінювання, залежністю від компетентностей команди, технологій, зовнішніх систем і умов експлуатації. Тому ризики можуть виникати в усіх складових проєкту та підсилювати один одного.

Ризики вимог знижують за допомогою раннього залучення користувачів, прототипування, критеріїв приймання, пріоритизації та контрольованого управління змінами. Ризики оцінювання і термінів потребують декомпозиції робіт, аналізу невизначеності, резервів і регулярного перегляду прогнозів.

Командні й організаційні ризики зменшують через чіткий розподіл відповідальності, поширення знань, підготовку резервних виконавців, своєчасну ескалацію та відкриту комунікацію. Технічні ризики необхідно

перевіряти за допомогою прототипів, ранньої інтеграції, автоматизованого тестування й підготовки резервних рішень.

План реагування повинен визначати не лише загальну стратегію, а й конкретні дії, власника, строки, умови активації, необхідні ресурси та резервний варіант. Заплановані заходи мають бути включені до робочого плану й регулярно контролюватися.

Ефективне управління ризиками не гарантує відсутності проблем, але забезпечує готовність команди своєчасно їх виявляти, обмежувати наслідки та зберігати керованість програмного проєкту.

Лекція 12 Принципи кількісного управління

Тема: Принципи кількісного управління

Мета: Сформувати у студентів розуміння сутності та принципів кількісного управління ІТ-проєктами, ознайомити з показниками продуктивності, прогресу, якості та ризиків, а також навчити використовувати метрики для контролю стану проєкту та прийняття обґрунтованих управлінських рішень.

План лекції

1. Поняття та призначення кількісного управління.
2. Основні принципи кількісного управління проєктом.
3. Вимоги до проєктних показників і вихідних даних.
4. Вимірювання прогресу та продуктивності команди.
5. Кількісне оцінювання ризиків.
6. Метрики термінів, ресурсів і якості.
7. Використання метрик для моніторингу та контролю проєкту.
8. Інтерпретація показників і прийняття управлінських рішень.
9. Висновки.

12.1. Поняття та призначення кількісного управління

У попередніх лекціях було розглянуто планування проєкту, оцінювання трудомісткості й термінів, управління ризиками, організацію роботи команди, моніторинг прогресу та контроль відхилень. Кількісне управління об'єднує отримані в межах цих процесів дані в єдину систему підтримки управлінських рішень.

Кількісне управління проєктом — це систематичне використання вимірюваних показників для оцінювання поточного стану проєкту, виявлення тенденцій, прогнозування результатів і вибору необхідних управлінських дій.

Його призначення полягає не в накопиченні максимальної кількості числової інформації, а в отриманні відповідей на конкретні управлінські питання:

- чи відповідає фактичний стан проєкту затвердженому плану;
- чи є виявлене відхилення допустимим;
- чи має показник стійку негативну тенденцію;
- які наслідки матиме збереження поточного стану;
- чи потрібно втручатися в перебіг проєкту;
- які дії можуть забезпечити найкращий результат.

Кількісне управління спирається на базові плани змісту, термінів і ресурсів, розглянуті в лекції 3. Методи формування оцінок і врахування їхньої невизначеності було подано в лекції 5, а порядок зіставлення фактичного виконання з планом — у лекції 7. Тому в цій лекції основна увага зосереджується на побудові узгодженої системи метрик та інтерпретації отриманих значень.

Потрібно розрізняти такі поняття:

- дані — зафіксовані факти про перебіг або результати проєкту;
- метрика — визначений спосіб кількісного оцінювання певної характеристики;
- показник — конкретне значення метрики за встановлений період;
- індикатор — показник, який сигналізує про стан проєкту або потребу в реагуванні;
- цільове значення — результат, якого необхідно досягти;
- граничне значення — межа, після перевищення якої необхідне управлінське втручання.

Наприклад, час виконання завдання є метрикою, середнє значення за поточний місяць — показником, а його стійке зростання вище встановленої межі — індикатором можливої проблеми в організації робіт.

Таким чином, кількісне управління перетворює проєктні дані на інформацію, придатну для аналізу та прийняття рішень.

12.2. Основні принципи кількісного управління проєктом

Система метрик повинна відповідати цілям, масштабу, складності та способу реалізації конкретного проєкту. Не існує універсального набору показників, однаково придатного для всіх ІТ-проєктів.

Вибирати метрики доцільно за такою логікою:

1. Визначити управлінську ціль.
2. Сформулювати питання, відповідь на яке потрібна для прийняття рішення.
3. Вибрати показники, які дають змогу отримати цю відповідь.
4. Установити джерела даних, періодичність вимірювання та правила інтерпретації.
5. Визначити можливі дії у разі відхилення.

Наприклад, якщо управлінською ціллю є забезпечення передбачуваного випуску результатів, недостатньо лише підраховувати

кількість завершених завдань. Потрібно також контролювати час їх виконання, обсяг незавершеної роботи, кількість заблокованих завдань і частку результатів, прийнятих без повторного доопрацювання.

Основні принципи кількісного управління наведено в таблиці 12.1.

Таблиця 12.1 – Принципи побудови системи проєктних метрик

Принцип	Зміст
Зв'язок із цілями	Кожна метрика повинна характеризувати досягнення певної цілі або підтримувати конкретне рішення
Однозначність	Формула, одиниця вимірювання та правила обчислення мають бути однаково зрозумілими всім учасникам
Порівнюваність	Значення повинні мати основу для порівняння: базовий план, ціль, допустиму межу або попередній період
Регулярність	Дані збирають із визначеною та стабільною періодичністю
Збалансованість	Показники термінів, ресурсів, якості, прогресу і ризиків аналізують у взаємозв'язку
Орієнтація на тенденції	Висновки роблять на основі зміни показника, а не лише його одиничного значення
Практична придатність	Для суттєвого відхилення має бути передбачена можлива управлінська дія
Економічність	Витрати на збирання й аналіз даних повинні відповідати користі від їх використання

Кількість метрик потрібно обмежувати. Надмірна деталізація збільшує витрати на звітування, ускладнює сприйняття інформації та відволікає увагу від справді важливих сигналів.

Доцільно виділяти невеликий набір ключових показників, які використовуються керівником проєкту, та додаткові операційні метрики для роботи окремих команд і функціональних напрямів.

12.3. Вимоги до проєктних показників і вихідних даних

Результати кількісного аналізу залежать не лише від правильно вибраних формул, а й від якості вихідних даних. Неповні, застарілі або суперечливі дані можуть створити хибне уявлення про стан проєкту.

Проєктні дані повинні відповідати таким вимогам:

- **достовірність** — дані правильно відображають фактичний стан;

- повнота — враховано всю інформацію, суттєву для розрахунку;
- актуальність — значення оновлено відповідно до встановленої періодичності;
- послідовність — в усіх періодах застосовуються однакові правила;
- порівнюваність — показники мають спільну основу вимірювання;
- простежуваність — можна визначити джерело кожного значення;
- перевірюваність — результати можна повторно розрахувати й підтвердити;
- достатня деталізація — рівень деталізації відповідає управлінській потребі.

Для кожної ключової метрики доцільно визначити її паспорт. Рекомендовану структуру такого паспорта подано в таблиці 12.2.

Таблиця 12.2 – Структура паспорта проєктної метрики

Елемент	Зміст
Назва	Однозначна назва показника
Управлінська мета	Рішення або питання, для якого використовується метрика
Визначення	Характеристика, яку відображає показник
Спосіб розрахунку	Формула або правило отримання значення
Одиниця вимірювання	Відсотки, години, дні, грошові одиниці, кількість подій тощо
Джерело даних	Система управління завданнями, розклад, реєстр дефектів, фінансовий звіт тощо
Періодичність	Щоденно, щотижнево, після завершення ітерації або контрольної точки
Відповідальний	Особа, яка забезпечує отримання та перевірку даних
Цільове і граничне значення	Запланований результат і межі допустимого відхилення
Правило реагування	Дії, які виконуються в разі виходу показника за встановлені межі

Перед початком використання метрики потрібно переконатися, що команда однаково розуміє визначення вихідних подій. Наприклад,

необхідно однозначно встановити, коли завдання вважається завершеним, дефект — усуненим, а результат — прийнятим.

Дані можуть надходити із систем управління завданнями, обліку робочого часу, контролю версій, тестування, фінансового обліку, моніторингу програмного продукту, а також із реєстрів ризиків, проблем і змін. Автоматичне отримання даних зменшує трудомісткість, але не усуває необхідності перевіряти правильність їх тлумачення.

12.4. Вимірювання прогресу та продуктивності команди

Загальні процеси моніторингу прогресу та контролю виконання проектних робіт було розглянуто в лекції 7. У межах кількісного управління важливо правильно вибрати основу, за якою визначається фактичний прогрес.

Витрачений час не є безпосереднім свідченням виконання роботи. Прогрес повинен підтверджуватися завершеними та прийнятими результатами, досягнутими контрольними точками або виконаними критеріями готовності.

Для оцінювання прогресу можуть використовуватися:

- частка завершених і прийнятих робіт;
- частка досягнутих контрольних точок;
- обсяг функціональності, готової до використання;
- кількість результатів, прийнятих без доопрацювання;
- прогнозована дата завершення залишкових робіт;
- стабільність виконання плану за окремими періодами.

Під час оцінювання продуктивності важливо враховувати не лише кількість створених результатів, а й витрачені ресурси, якість та цінність цих результатів. Збільшення кількості завершених завдань не можна вважати покращенням, якщо одночасно зросли дефектність, понаднормова робота або обсяг повторного виконання.

До показників робочого процесу належать:

- пропускна здатність команди;
- середній час виконання завдання;
- обсяг незавершеної роботи;
- частка заблокованих завдань;
- частка повторного виконання;
- стабільність результатів між послідовними періодами.

Показники продуктивності доцільно використовувати для аналізу тенденцій усередині однієї команди. Безпосереднє порівняння різних команд може бути некоректним через відмінності у складності завдань, технологіях, компетентностях, критеріях готовності та способах оцінювання.

Кількість рядків програмного коду, комітів або годин присутності не варто використовувати як основні показники результативності окремого працівника. Такі значення характеризують активність, але не обов'язково відображають створену цінність.

12.5. Кількісне оцінювання ризиків

Методи ідентифікації, якісного і кількісного аналізу ризиків, а також планування реагування було розглянуто в лекції 4. Тому в системі кількісного управління основна увага приділяється не повторному розрахунку кожного ризику, а контролю зміни загального ризикового профілю проєкту.

Доцільно відстежувати:

- кількість активних критичних ризиків;
- сумарний очікуваний вплив ризиків;
- зміну рівня ключових ризиків;
- кількість нових і закритих ризиків за період;
- частку виконаних заходів реагування;
- рівень залишкового ризику;
- обсяг використаного резерву;
- кількість ризиків, що перетворилися на фактичні проблеми.

Окреме значення має не лише поточний рівень ризику, а й напрям його зміни. Незмінна кількість ризиків може приховувати погіршення ситуації, якщо зростає їхній можливий вплив або не виконуються заплановані заходи реагування.

Кількісні показники ризиків повинні бути пов'язані з реєстром ризиків і відповідними управлінськими діями. Якщо перевищено граничний рівень ризику, це може бути підставою для використання резерву, зміни способу виконання робіт, залучення додаткової експертизи або ескалації питання керівництву.

12.6. Метрики термінів, ресурсів і якості

Методи оцінювання термінів і трудомісткості розглянуто в лекції 5, а порядок моніторингу виконання — у лекції 7. У кількісному управлінні ці показники потрібно поєднати з метриками ресурсів, якості та ризиків.

Приклад збалансованого набору показників наведено в таблиці 12.3.

Таблиця 12.3 – Збалансований набір показників IT-проєкту

Напрямок	Управлінське питання	Приклади показників
Терміни	Чи буде проєкт завершено відповідно до затвердженого розкладу?	Відхилення контрольних точок, прогнозована дата завершення, частка своєчасно виконаних робіт
Ресурси	Чи відповідає фактичне використання ресурсів запланованій спроможності?	Завантаження команди, понаднормова робота, дефіцит критичних компетентностей
Робочий процес	Наскільки стабільно команда створює результати?	Пропускна здатність, час виконання, незавершена робота, заблоковані завдання
Якість	Чи відповідають створені результати встановленим вимогам?	Кількість критичних дефектів, частка повторного виконання, результати приймання і тестування
Ризики	Який рівень невизначеності залишається в проєкті?	Критичні ризики, залишковий вплив, виконання заходів реагування, використання резерву

Показники різних напрямів потрібно аналізувати спільно. Наприклад, скорочення строків виконання може супроводжуватися збільшенням понаднормової роботи, кількості дефектів і повторного виконання. Формально розклад може покращитися, але загальний стан проєкту погіршиться.

Повне завантаження всіх учасників також не завжди є позитивним результатом. Відсутність резерву спроможності обмежує можливість реагувати на непередбачені роботи й може спричинити накопичення незавершених завдань.

Набір показників повинен відповідати особливостям продукту. Для сервісу, що працює безперервно, важливими будуть доступність і час відновлення. Для проєкту розроблення нового програмного продукту — прогрес реалізації функціональності, стабільність розкладу та результати приймального тестування.

12.7. Використання метрик для моніторингу та контролю проєкту

Метрики повинні використовуватися в межах регулярного циклу управління:

1. Отримання даних із визначених джерел.
2. Перевірка їх повноти й достовірності.
3. Розрахунок установлених показників.
4. Порівняння з базовими, цільовими та граничними значеннями.
5. Аналіз тенденцій і взаємозв'язків.
6. Визначення причин суттєвих відхилень.
7. Підготовка управлінського рішення.
8. Контроль результативності виконаних дій.

Періодичність аналізу встановлюється відповідно до швидкості змін у проєкті. Операційні показники можуть оновлюватися щодня або щотижня, а узагальнений звіт про стан проєкту — після завершення ітерації, звітного періоду або досягнення контрольної точки.

Для керівника проєкту доцільно формувати компактну інформаційну панель, яка містить лише ключові показники:

- загальний прогрес;
- стан основних контрольних точок;
- прогноз завершення;
- стан використання ресурсів;
- ключові показники якості;
- критичні ризики та проблеми;
- зміни, що можуть вплинути на базовий план.

Для показників можна встановлювати зелений, жовтий і червоний статуси. Зелений статус означає перебування в допустимих межах, жовтий — появу відхилення або негативної тенденції, червоний — порушення критичного обмеження чи необхідність негайного реагування. Конкретні межі визначаються окремо для кожного проєкту.

Виявлене відхилення не повинно автоматично спричиняти зміну базового плану. Якщо необхідно змінити затверджені терміни, бюджет або зміст, це здійснюють відповідно до встановленого процесу управління змінами.

12.8. Інтерпретація показників і прийняття управлінських рішень

Показник набуває управлінського значення лише після порівняння та інтерпретації. Для цього потрібно враховувати:

- базове або цільове значення;
- установлені межі допустимого відхилення;
- зміну показника протягом кількох періодів;
- взаємозв'язок з іншими показниками;
- поточний етап проєкту;
- вплив зовнішніх і внутрішніх чинників;
- достовірність вихідних даних.

Одиничне відхилення не завжди потребує втручання. Воно може бути випадковим або короткостроковим. Натомість поступове погіршення показника протягом кількох періодів може бути ранньою ознакою системної проблеми.

Доцільно розрізняти:

- випереджальні індикатори, які попереджають про можливе майбутнє відхилення;
- підсумкові індикатори, які фіксують уже отриманий результат.

Наприклад, збільшення незавершеної роботи, кількості блокувань або понаднормових годин може бути випереджальним сигналом майбутньої затримки. Фактичне порушення контрольної точки є вже підсумковим показником.

Перед прийняттям рішення керівник проєкту повинен:

1. Перевірити правильність даних.
2. Оцінити суттєвість відхилення.
3. Визначити його першопричину.
4. Спрогнозувати наслідки без втручання.
5. Розглянути можливі способи реагування.
6. Оцінити вплив кожного варіанта на терміни, ресурси, якість і ризики.
7. Визначити дію, відповідального та строк виконання.
8. Перевірити результат під час наступного циклу моніторингу.

За результатами аналізу може бути прийнято рішення про усунення перешкоди, зміну послідовності робіт, перерозподіл ресурсів, уточнення пріоритетів, реалізацію заходів реагування на ризик, використання резерву, посилення контролю якості або підготовку запиту на зміну.

Необхідно уникати ситуації, коли команда починає працювати заради покращення самого показника. Наприклад, прагнення збільшити кількість завершених завдань може спричинити їх штучне подрібнення, а вимога зменшити кількість дефектів — приховування або неправильну класифікацію проблем.

Тому кількісні показники не замінюють професійного судження керівника та команди. Вони забезпечують інформаційну основу для рішення, але остаточний вибір повинен враховувати цілі проєкту, його обмеження та інтереси зацікавлених сторін.

Висновки

Кількісне управління поєднує результати планування, оцінювання, виконання, управління ризиками та контролю проєкту в єдину систему показників. Його основним призначенням є забезпечення керівника достовірною інформацією для своєчасного прийняття рішень.

Система метрик повинна бути пов'язана з цілями проєкту, спиратися на перевірені дані та містити правила інтерпретації й реагування. Для кожного ключового показника необхідно визначити джерело даних, спосіб розрахунку, періодичність оновлення, цільове значення, допустимі межі та відповідального.

Показники термінів, ресурсів, прогресу, якості й ризиків потрібно розглядати у взаємозв'язку. Покращення одного показника не завжди означає покращення загального стану проєкту, якщо воно досягається за рахунок інших обмежень.

Основна цінність кількісного управління полягає не в кількості зібраних даних, а в здатності виявити тенденцію, пояснити причину відхилення, спрогнозувати наслідки та вибрати обґрунтовану управлінську дію.

Лекція 13 Ключові учасники і зацікавлені сторони

Тема: Ключові учасники і зацікавлені сторони

Мета: Сформувати у студентів розуміння ролі ключових учасників і зацікавлених сторін у реалізації IT-проєкту, ознайомити з методами їх ідентифікації, аналізу, класифікації, залучення та організації ефективної взаємодії.

План лекції

1. Поняття ключових учасників і зацікавлених сторін.
2. Основні ролі та відповідальність учасників проєкту.
3. Внутрішні та зовнішні стейкхолдери.
4. Ідентифікація та аналіз зацікавлених сторін.
5. Оцінювання впливу, інтересів та очікувань стейкхолдерів.
6. Планування залучення та комунікацій.
7. Методи взаємодії із зацікавленими сторонами.
8. Практичні приклади управління стейкхолдерами.
9. Висновки.

13.1. Поняття ключових учасників і зацікавлених сторін

Зацікавлена сторона, або стейкхолдер, — це особа, група осіб чи організація, яка може впливати на проєкт, зазнавати впливу його робіт або результатів чи вважати, що проєкт стосується її інтересів.

Стейкхолдери можуть підтримувати проєкт, ставитися до нього нейтрально або протидіяти його реалізації. Вони можуть безпосередньо брати участь у виконанні робіт або залишатися поза межами проєктної команди, але впливати на фінансування, вимоги, приймання результатів, доступ до ресурсів чи умови використання програмного продукту.

Потрібно розрізняти поняття учасник проєкту і зацікавлена сторона. Учасник виконує визначену функцію або має формальну роль у проєкті. Зацікавлена сторона не обов'язково бере участь у роботах, однак може впливати на рішення або зазнати наслідків впровадження результатів.

Отже:

- кожен учасник проєкту є його стейкхолдером;
- не кожен стейкхолдер є безпосереднім учасником проєкту;
- вплив стейкхолдера не завжди визначається його посадою;
- ставлення та рівень залучення можуть змінюватися протягом життєвого циклу проєкту.

Ключовими стейкхолдерами вважають осіб або групи, від рішень, підтримки, ресурсів, вимог чи приймання яких суттєво залежить успіх проєкту. До них зазвичай належать замовник, спонсор, керівник проєкту, представники користувачів, власники бізнес-процесів, керівники ресурсів і відповідальні за приймання результатів.

Інтереси різних сторін можуть не збігатися. Замовник зацікавлений у швидкому отриманні функціонального продукту, керівництво — у дотриманні бюджету, користувачі — у зручності роботи, а служба інформаційної безпеки — у відповідності встановленим вимогам. Управління стейкхолдерами передбачає виявлення таких інтересів і пошук узгодженого способу досягнення цілей проєкту.

13.2. Основні ролі та відповідальність учасників проєкту

Ролі проєктної команди, принципи добору учасників і розподілу відповідальності розглянуто в лекції 6. Вплив організаційної структури компанії на повноваження керівника та доступ до ресурсів висвітлено в лекції 9. Тому в цій лекції ролі розглядаються передусім з позиції їхньої участі в управлінні зацікавленими сторонами.

До ключових ролей належать:

- спонсор або власник проєкту — забезпечує стратегічну підтримку, фінансування, доступ до ресурсів і прийняття рішень, які виходять за межі повноважень керівника проєкту;
- керівник проєкту — організовує ідентифікацію та аналіз стейкхолдерів, планує комунікації, контролює очікування й координує взаємодію;
- представник замовника або бізнесу — формулює потреби, погоджує вимоги, визначає пріоритети та бере участь у прийманні результатів;
- проєктна команда — створює результати, надає професійні оцінки та взаємодіє зі стейкхолдерами у межах своєї компетентності;
- користувачі — надають інформацію про потреби й умови експлуатації, оцінюють прототипи та підтверджують придатність продукту;
- керівники ресурсів — забезпечують доступ до необхідних фахівців, обладнання й інших організаційних ресурсів;
- постачальники та підрядники — відповідають за виконання договірних зобов'язань і надання зовнішніх продуктів або послуг;

- контрольні та регуляторні сторони — перевіряють відповідність рішення законодавчим, галузевим, безпековим або організаційним вимогам.

Відповідальність за управління стейкхолдерами загалом належить керівнику проєкту. Проте він не повинен самотійно здійснювати всю взаємодію. Для окремих стейкхолдерів доцільно залучати спонсора, представника бізнесу, технічного керівника, фахівця з комунікацій або іншого учасника, який має необхідні повноваження й довіру.

Наприклад, керівник проєкту може готувати інформацію про критичну проблему, але рішення про додаткове фінансування доцільно обговорювати за участю спонсора. Технічні питання з архітектури варто розглядати разом із технічним керівником, а зміни бізнес-процесів — із представником відповідного підрозділу.

13.3. Внутрішні та зовнішні стейкхолдери

За належністю до організації стейкхолдерів поділяють на внутрішніх і зовнішніх.

Внутрішні стейкхолдери належать до організації, яка ініціює, фінансує, виконує або використовує результати проєкту. До них можуть належати:

- вище керівництво;
- спонсор і замовник;
- керівник та команда проєкту;
- функціональні керівники;
- працівники підрозділів-користувачів;
- фінансова, юридична та кадрова служби;
- фахівці з інформаційної безпеки;
- служба технічної підтримки;
- інші проєктні команди.

Зовнішні стейкхолдери перебувають поза організацією, але впливають на проєкт або залежать від його результатів. До них належать:

- зовнішні замовники і користувачі;
- постачальники та підрядники;
- партнери;
- консультанти;
- державні та місцеві органи;
- регуляторні установи;

- професійні об'єднання;
- громадські організації;
- представники суспільства.

Окрім поділу на внутрішніх і зовнішніх, стейкхолдерів можна класифікувати:

- за безпосередністю участі — на прямих і непрямих;
- за ставленням — на прихильників, нейтральних і противників;
- за характером впливу — на тих, хто впливає на проєкт, і тих, на кого впливає проєкт;
- за етапом участі — на залучених під час ініціації, планування, реалізації, приймання або експлуатації;
- за можливістю приймати рішення — на уповноважених, консультативних та інформаційних учасників.

Одна й та сама сторона може належати відразу до декількох категорій. Наприклад, зовнішній замовник може бути одночасно джерелом вимог, стороною, яка фінансує роботи, і стороною, яка приймає кінцевий результат.

13.4. Ідентифікація та аналіз зацікавлених сторін

Початкову ідентифікацію стейкхолдерів проводять під час ініціації проєкту, а її результати уточнюють у процесі планування. Проте склад стейкхолдерів не є незмінним. Нові особи й групи можуть з'являтися внаслідок зміни вимог, організаційної структури, законодавства, постачальників або способу впровадження продукту.

Для ідентифікації стейкхолдерів використовують:

- аналіз статуту й бізнес-обґрунтування проєкту;
- аналіз договорів, вимог і проєктної документації;
- вивчення організаційної структури;
- консультації зі спонсором і замовником;
- інтерв'ю з представниками підрозділів;
- аналіз бізнес-процесів і майбутніх користувачів;
- аналіз постачальників, партнерів і регуляторних вимог;
- вивчення досвіду аналогічних проєктів;
- командні обговорення та експертне оцінювання.

Корисно послідовно відповісти на такі запитання:

1. Хто ініціював і фінансує проєкт?
2. Хто приймає основні рішення?

3. Хто формулює та погоджує вимоги?
4. Хто створює, перевіряє і приймає результати?
5. Хто користуватиметься продуктом?
6. Чиї робочі процеси зміняться після впровадження?
7. Хто надає ресурси та професійну експертизу?
8. Хто може підтримати, затримати або заблокувати проєкт?
9. Хто відповідає за дотримання нормативних вимог?
10. Хто супроводжуватиме продукт після завершення проєкту?

Результати ідентифікації та початкового аналізу фіксують у реєстрі стейкхолдерів. Його рекомендовану структуру наведено в таблиці 13.1.

Таблиця 13.1 – Рекомендована структура реєстру стейкхолдерів

Поле	Зміст
Стейкхолдер	Ім'я особи або назва групи чи організації
Роль	Функція стейкхолдера у проєкті або його зв'язок із результатом
Внутрішній чи зовнішній	Належність до організації
Інтереси та очікування	Результати, вигоди або умови, важливі для стейкхолдера
Вплив	Здатність впливати на рішення, ресурси, вимоги або приймання
Ставлення	Підтримка, нейтральне ставлення або протидія
Потреба в інформації	Яку інформацію необхідно надавати
Потенційні проблеми	Можливі конфлікти, заперечення або обмеження
Стратегія залучення	Запланований спосіб взаємодії
Відповідальний	Учасник команди, відповідальний за комунікацію
Дата перегляду	Час наступного оновлення інформації

Реєстр має регулярно переглядатися. Частина інформації в ньому може бути конфіденційною, особливо оцінки впливу, ставлення, можливих заперечень і неформальних зв'язків. Тому доступ до повної версії документа доцільно обмежувати.

За потреби для команди можна підготувати окрему робочу версію реєстру без персональних оцінок і чутливої інформації.

13.5. Оцінювання впливу, інтересів та очікувань стейкхолдерів

Після ідентифікації необхідно визначити значення кожного стейкхолдера для проєкту. Аналіз може враховувати такі характеристики:

- рівень формальних повноважень;
- вплив на фінансування і ресурси;
- можливість погоджувати або блокувати рішення;
- рівень зацікавленості в результатах;
- ступінь впливу проєкту на діяльність стейкхолдера;
- ставлення до проєкту;
- терміновість його потреб;
- очікуваний внесок у виконання робіт;
- взаємозв'язки з іншими стейкхолдерами.

Необхідно розрізняти інтерес і ставлення. Високий інтерес не означає підтримку: стейкхолдер може активно цікавитися проєктом саме тому, що вважає його небажаним або загрозливим.

Одним із найпоширеніших інструментів аналізу є матриця «вплив — інтерес». Вона допомагає визначити пріоритетність стейкхолдерів і загальну стратегію взаємодії. Приклад такої матриці подано на рисунку 13.1.

		Importance of Project to Stakeholder (Interest)					
		None or Unknown	Little	Some	Moderate	Significant	High
Stakeholder Influence (Power)	High	<u>Influential</u> Keep satisfied			<u>Important</u> Collaborate and involve		
	Significant						
	Moderate						
	Some	<u>Aware</u> Monitor			<u>Interested</u> Keep informed		
	Little						
	None or Unknown						

Рисунок 13.1 – Вибір стратегії залучення стейкхолдерів за рівнем впливу та інтересу

За результатами аналізу використовують такі загальні стратегії:

- високий вплив і високий інтерес — тісно співпрацювати, залучати до планування та прийняття ключових рішень;

- високий вплив і низький інтерес — підтримувати задоволеність, своєчасно повідомляти про суттєві зміни й не перевантажувати деталями;
- низький вплив і високий інтерес — регулярно інформувати, збирати відгуки та залучати до обговорення питань, які безпосередньо їх стосуються;
- низький вплив і низький інтерес — періодично спостерігати та переглядати оцінку у разі зміни умов.

Матриця не є постійною. Наприклад, користувачі можуть мати обмежений формальний вплив під час початкового планування, але їхнє значення суттєво зростає під час тестування, приймання та впровадження продукту. Служба інформаційної безпеки може мати помірний інтерес до більшості робіт, але високий вплив на рішення про введення системи в експлуатацію.

Висновки аналізу повинні визначати не лише частоту комунікацій, а й характер участі стейкхолдера: інформування, консультування, погодження, спільне вироблення рішення або передавання йому відповідальності за певний результат.

13.6. Планування залучення та комунікацій

Залучення стейкхолдерів — це цілеспрямована робота зі створення та підтримання такого рівня участі, який сприяє досягненню цілей проекту.

Для оцінювання поточного і бажаного рівнів залучення можна використовувати шкалу, наведену в таблиці 13.2.

Таблиця 13.2 – Рівні залучення стейкхолдерів

Рівень	Характеристика
Необізнаний	Не знає про проект або його можливий вплив
Противник	Знає про проект, але не підтримує його
Нейтральний	Обізнаний, але не демонструє активної підтримки чи протидії
Прихильник	Підтримує проект і готовий сприяти його реалізації
Лідер	Активно просуває проект, впливає на інших і бере участь у подоланні перешкод

Для кожного ключового стейкхолдера визначають поточний і бажаний рівні. Мета управління не завжди полягає в перетворенні всіх стейкхолдерів на активних прихильників. У деяких випадках достатньо

зменшити протидію, забезпечити нейтральне ставлення або своєчасне виконання конкретного обов'язку.

План комунікацій повинен визначати:

- кому надається інформація;
- яка інформація потрібна;
- для чого вона надається;
- хто відповідає за її підготовку;
- яким каналом вона передається;
- з якою періодичністю;
- у якому форматі;
- як фіксуються рішення та зворотний зв'язок;
- у яких випадках застосовується ескалація.

Приклад фрагмента плану комунікацій для проєкту впровадження інформаційної системи наведено в таблиці 13.3.

Таблиця 13.3 – Приклад плану комунікацій зі стейкхолдерами

Стейкхолдер	Інформація	Формат і періодичність	Відповідальний	Очікуваний результат
Спонсор	Загальний стан, критичні ризики, потреба в рішеннях	Короткий звіт і зустріч раз на місяць	Керівник проєкту	Підтримка та своєчасні управлінські рішення
Представник замовника	Вимоги, пріоритети, готовність результатів	Робоча зустріч щотижня	Керівник проєкту, бізнес-аналітик	Погоджені вимоги та пріоритети
Користувачі	Прототипи, зміни процесів, інструкції	Демонстрації, опитування, навчання	Бізнес-аналітик	Зворотний зв'язок і готовність до використання
Технічна служба	Архітектура, інтеграції, розгортання	Технічні наради за контрольними точками	Технічний керівник	Готовність інфраструктури й супроводу
Постачальник	Завдання, строки, критерії приймання	Робочі зустрічі та письмові підтвердження	Керівник проєкту	Виконання договірних зобов'язань

Комунікація повинна бути двосторонньою. Надсилання звіту не гарантує, що інформацію прочитано, зрозуміло та використано. Для важливих питань необхідно передбачати підтвердження отримання, можливість поставити запитання та формальну фіксацію прийнятого рішення.

13.7. Методи взаємодії із зацікавленими сторонами

Методи взаємодії вибирають відповідно до мети, кількості учасників, складності питання, рівня впливу стейкхолдера та необхідності отримати зворотний зв'язок.

Основними методами є:

- індивідуальні інтерв'ю;
- робочі та координаційні зустрічі;
- наради керівного комітету;
- семінари зі збору й узгодження вимог;
- демонстрації прототипів і проміжних результатів;
- опитування та анкетування;
- фокус-групи;
- презентації та інформаційні повідомлення;
- спільне опрацювання документів;
- переговори;
- навчання користувачів;
- консультації й технічна підтримка;
- процедури погодження та приймання.

За характером обміну інформацією комунікації можна поділити на:

- інтерактивні — зустрічі, відеоконференції, обговорення, переговори;
- активне поширення — надсилання звітів, повідомлень, протоколів і презентацій;
- надання доступу — розміщення інформації в системі управління проектом, базі знань або сховищі документів.

Для складних і конфліктних питань доцільною є інтерактивна взаємодія, яка дає змогу одразу уточнювати позиції. Для регулярного звітування можна використовувати активне поширення інформації. Великі обсяги довідкових матеріалів краще зберігати в спільному інформаційному середовищі.

Ефективна взаємодія передбачає:

- адаптацію змісту до потреб отримувача;
- використання зрозумілої термінології;
- своєчасність повідомлень;
- відокремлення фактів від припущень;
- чесне інформування про проблеми;
- документування погоджень і рішень;
- повагу до конфіденційності;
- контроль виконання домовленостей.

Якщо інтереси сторін суперечать один одному, керівник проєкту повинен організувати переговори та забезпечити прозоре порівняння альтернатив. Рішення бажано обґрунтовувати цілями проєкту, критеріями успішності, вимогами та встановленими обмеженнями, а не особистим впливом окремого учасника.

13.8. Практичні приклади управління стейкхолдерами

Розглянемо проєкт упровадження університетської системи управління навчанням.

До ключових стейкхолдерів такого проєкту можуть належати:

- керівництво університету;
- адміністрація факультетів і кафедр;
- викладачі;
- студенти;
- навчально-методичний підрозділ;
- IT-служба;
- служба інформаційної безпеки;
- постачальник програмного забезпечення;
- служба технічної підтримки.

Керівництво університету має високий вплив, оскільки затверджує фінансування та організаційні рішення. Для нього важливими є строки впровадження, бюджет, охоплення користувачів і досягнення запланованого результату. Тому взаємодія повинна бути стислою, регулярною та орієнтованою на рішення.

Викладачі та студенти мають високий інтерес, оскільки безпосередньо використовуватимуть систему. Їх доцільно залучати до опитувань, демонстрацій, пілотного тестування та оцінювання зручності роботи.

ІТ-служба має значний вплив на інтеграцію, розгортання і подальший супровід. Якщо її залучити лише наприкінці проєкту, можуть виникнути проблеми з архітектурою, сумісністю, доступом до даних і передаванням продукту в експлуатацію.

Служба інформаційної безпеки може не брати участі в щоденній роботі, але має повноваження зупинити впровадження, якщо система не відповідає вимогам захисту даних. Тому її потрібно залучити до погодження архітектури й перевірки готовності до запуску.

Припустимо, що частина викладачів негативно ставиться до нової системи через побоювання збільшення навантаження. Простого надсилання інструкцій у цьому разі буде недостатньо. Необхідно з'ясувати конкретні причини протидії, продемонструвати типові сценарії роботи, організувати пілотне використання, врахувати обґрунтовані зауваження та забезпечити навчання.

Після початку пілотного впровадження аналіз стейкхолдерів потрібно переглянути. Студенти й викладачі можуть отримати більший вплив через результати тестування, а служба підтримки — стати ключовою стороною через необхідність обробляти звернення користувачів.

Цей приклад показує, що управління стейкхолдерами не завершується формуванням початкового реєстру. Воно потребує постійного спостереження за очікуваннями, ставленням, рівнем впливу та результативністю обраної стратегії взаємодії.

Висновки

Зацікавлені сторони можуть підтримувати проєкт, впливати на його рішення, надавати необхідні ресурси, формувати вимоги, приймати результати або протидіяти запланованим змінам. Тому їх своєчасна ідентифікація та залучення є важливою умовою успішної реалізації ІТ-проєкту.

Управління стейкхолдерами охоплює їх ідентифікацію, ведення реєстру, аналіз інтересів, впливу, очікувань і ставлення, визначення поточного та бажаного рівнів залучення, планування комунікацій і регулярний перегляд обраної стратегії.

Матриця «вплив — інтерес» допомагає визначити пріоритетність стейкхолдерів і загальний підхід до взаємодії. Однак її результати не є постійними, оскільки роль та вплив окремих сторін можуть змінюватися на різних етапах життєвого циклу.

Комунікації повинні враховувати потреби конкретних стейкхолдерів, передбачати отримання зворотного зв'язку та сприяти своєчасному прийняттю рішень. Водночас надмірна кількість повідомлень може бути так само неефективною, як і недостатнє інформування.

Ефективне управління стейкхолдерами полягає не в задоволенні всіх індивідуальних побажань, а в узгодженні обґрунтованих інтересів із цілями, вимогами та обмеженнями проєкту.

Лекція 14 Agile-методології розроблення програмного забезпечення

Тема: Agile-методології розроблення програмного забезпечення

Мета: Сформувати у студентів розуміння сутності, цінностей і принципів Agile, ознайомити з особливостями Scrum, Kanban та інших гнучких підходів, а також навчити оцінювати переваги, обмеження та доцільність їх застосування порівняно з традиційними моделями управління розробленням програмного забезпечення.

План лекції

1. Передумови виникнення та сутність Agile.
2. Цінності й основні принципи Agile.
3. Ітеративне та інкрементальне розроблення програмного забезпечення.
4. Scrum: ролі, події та артефакти.
5. Kanban: візуалізація потоку робіт та обмеження незавершеної роботи.
6. Інші гнучкі підходи до розроблення програмного забезпечення.
7. Переваги та недоліки Agile порівняно з традиційними підходами.
8. Вибір гнучкого, традиційного або гібридного підходу.
9. Висновки.

14.1. Передумови виникнення та сутність Agile

Традиційні моделі управління передбачають детальне визначення змісту, термінів, бюджету та послідовності робіт до початку основного виконання. Такий підхід є ефективним, якщо вимоги достатньо стабільні, технологія добре відома, а зміни можна передбачити й контролювати.

У програмних проєктах ці умови виконуються не завжди. Замовник може уточнювати потреби після ознайомлення з першими версіями продукту, технологічні обмеження можуть бути виявлені лише під час реалізації, а зміни ринку або законодавства можуть зробити частину початкових вимог неактуальною.

До проблем, які сприяли поширенню гнучких підходів, належать:

- тривалий період між формуванням вимог і отриманням готового продукту;
- пізнє виявлення неправильного розуміння потреб користувачів;
- висока вартість змін наприкінці проєкту;

- складність точного прогнозування всього обсягу робіт;
- накопичення значної кількості неперевірених компонентів;
- недостатня взаємодія між замовником і командою;
- оцінювання прогресу за документами, а не за працездатним продуктом.

Agile виник як відповідь на потребу швидше перевіряти припущення, отримувати зворотний зв'язок і пристосовувати розроблення до змін.

Agile — це сукупність цінностей і принципів організації роботи, орієнтована на створення цінного результату невеликими частинами, постійну взаємодію із замовником, адаптивне планування та регулярне вдосконалення способу роботи.

Agile не є єдиною методологією з фіксованим набором процедур. Це загальний підхід, на основі якого застосовуються різні фреймворки, методи та практики, зокрема Scrum, Kanban, Extreme Programming, Lean Software Development і Crystal.

Використання Agile не означає відмову від планування, документації, управління ризиками, контролю якості або відповідальності. Змінюється спосіб виконання цих процесів:

- планування проводиться регулярно та з різним рівнем деталізації;
- вимоги уточнюються в міру отримання нової інформації;
- результат перевіряється невеликими частинами;
- зміни розглядаються як очікувана складова роботи;
- якість забезпечується протягом усього розроблення;
- рішення приймаються на основі фактичних результатів і зворотного зв'язку.

Гнучкий підхід не усуває потреби в загальному управлінні проектом. Питання бюджету, строків, ризиків, ресурсів, конфігурацій, стейкхолдерів і приймання результатів залишаються актуальними. PM²-Agile, наприклад, поєднує Agile-практики з організаційним управлінням, життєвим циклом і корпоративними процедурами [4].

14.2. Цінності й основні принципи Agile

Оснoву Agile становлять чотири цінності, сформульовані в Мaніфeсті гнучкого розроблення програмного забезпечення. Їхній зміст наведено в таблиці 14.1.

Таблиця 14.1 – Цінності Agile

Більше цінується	Порівняно з	Практичний зміст
Люди та взаємодія	Процеси й інструменти	Результат залежить від співпраці, компетентності та відповідальності учасників
Працездатне програмне забезпечення	Вичерпна документація	Основним доказом прогресу є готовий і перевірений продукт
Співпраця із замовником	Формальне погодження умов	Регулярна взаємодія допомагає своєчасно уточнювати потреби
Реагування на зміни	Дотримання початкового плану	План переглядають, якщо нова інформація вимагає іншого рішення

Праві складові таблиці не заперечуються. Процеси, інструменти, документація, договори та плани залишаються потрібними, але не повинні перешкоджати створенню цінного результату.

Основні принципи Agile передбачають:

1. Раннє та регулярне постачання цінного програмного забезпечення.
2. Готовність уточнювати вимоги навіть на пізніх етапах.
3. Часте створення працездатних версій продукту.
4. Постійне співробітництво представників бізнесу й розробників.
5. Організацію роботи навколо мотивованих і компетентних учасників.
6. Підтримання відкритої та безпосередньої комунікації.
7. Оцінювання прогресу за готовим програмним забезпеченням.
8. Підтримання сталого темпу роботи без систематичних перевантажень.
9. Постійне забезпечення технічної якості.
10. Простоту рішень і відмову від непотрібної роботи.
11. Самоорганізацію команди в межах визначених цілей і повноважень.

12. Регулярне вдосконалення процесу за результатами аналізу досвіду.

Самоорганізація не означає відсутності керівництва чи відповідальності. Команда самостійно визначає спосіб виконання роботи, але діє в межах погоджених цілей, пріоритетів, стандартів якості та організаційних обмежень.

Agile також не передбачає автоматичного прийняття кожної запропонованої зміни. Зміна повинна бути оцінена з погляду цінності, пріоритету, вартості, ризиків і впливу на вже заплановані роботи.

14.3. Ітеративне та інкрементальне розроблення програмного забезпечення

Гнучкі підходи переважно поєднують ітеративне та інкрементальне розроблення.

Ітерація — це обмежений період роботи, протягом якого команда планує, створює, перевіряє та аналізує частину продукту. Після завершення ітерації отриманий досвід використовується для уточнення наступних дій.

Інкремент — це завершене збільшення функціональності продукту, яке інтегровано з попередніми результатами та відповідає встановленим критеріям якості.

Ітеративність передбачає повторне уточнення рішення, а інкрементальність — поступове збільшення обсягу готового продукту. Проєкт може бути ітеративним без регулярного створення готових інкрементів, якщо команда лише переглядає моделі чи прототипи. Він також може бути інкрементальним, але недостатньо адаптивним, якщо весь зміст визначено наперед і лише розділено на послідовні частини.

Типовий цикл гнучкого розроблення охоплює:

1. Вибір найбільш пріоритетних потреб.
2. Уточнення очікуваного результату.
3. Планування невеликого обсягу робіт.
4. Проєктування, реалізацію та тестування.
5. Інтеграцію результату з продуктом.
6. Демонстрацію зацікавленим сторонам.
7. Отримання зворотного зв'язку.
8. Перегляд пріоритетів і наступного плану.
9. Аналіз способу роботи та його вдосконалення.

Результат кожної ітерації повинен відповідати погодженому визначенню готовності. Воно може містити вимоги щодо:

- завершення реалізації;
- перевірки програмного коду;
- проходження автоматизованих тестів;
- усунення критичних дефектів;
- оновлення необхідної документації;
- відповідності безпековим вимогам;
- інтеграції з поточною версією продукту.

Не кожен інкремент обов'язково передається кінцевим користувачам. Проте він повинен бути достатньо якісним, щоб не створювати прихованого обсягу незавершених робіт.

Ітеративно-інкрементальний підхід дає змогу раніше перевірити найбільш невизначені або цінні частини продукту, зменшити ризик створення непотрібної функціональності та отримати практичні дані для подальшого планування.

14.4. Scrum: ролі, події та артефакти

Scrum — це легкий фреймворк для організації роботи над складними продуктами. Він ґрунтується на емпіричному управлінні, за якого рішення приймаються на основі досвіду та спостережуваних результатів.

Основними засадами Scrum є:

- прозорість — стан роботи, правила та критерії мають бути зрозумілими;
- інспекція — результат і процес регулярно перевіряються;
- адаптація — у разі виявлення проблем або нової інформації спосіб роботи коригується.

У сучасному Scrum використовують поняття відповідальностей, хоча в навчальній і професійній практиці їх часто називають ролями.

Scrum-команда охоплює:

- Product Owner — відповідає за максимізацію цінності продукту, формування його мети й упорядкування Product Backlog;
- Scrum Master — сприяє правильному застосуванню Scrum, допомагає команді підвищувати ефективність і усувати організаційні перешкоди;
- Developers — створюють придатний до використання інкремент, планують роботу і відповідають за її якість.

Структуру Scrum узагальнено в таблиці 14.2.

Таблиця 14.2 – Відповідальності, події та артефакти Scrum

Категорія	Елемент	Призначення
Відповідальність	Product Owner	Визначає мету продукту, пріоритети й порядок елементів Product Backlog
Відповідальність	Scrum Master	Допомагає команді й організації ефективно застосовувати Scrum
Відповідальність	Developers	Створюють якісний інкремент і визначають спосіб виконання робіт
Подія	Sprint	Фіксований період, у межах якого створюється корисний інкремент
Подія	Sprint Planning	Визначення цілі Sprint і плану її досягнення
Подія	Daily Scrum	Коротка щоденна перевірка прогресу та коригування плану
Подія	Sprint Review	Аналіз результату разом із зацікавленими сторонами та перегляд подальших пріоритетів
Подія	Sprint Retrospective	Аналіз способу роботи й визначення покращень
Артефакт	Product Backlog	Упорядкований перелік потреб і можливих змін продукту
Артефакт	Sprint Backlog	Ціль Sprint, вибрані елементи й план їх реалізації
Артефакт	Increment	Інтегрований результат, який відповідає визначенню готовності

З кожним артефактом пов'язане зобов'язання:

- Product Backlog — із Product Goal;
- Sprint Backlog — зі Sprint Goal;
- Increment — із Definition of Done.

Sprint є основним циклом Scrum. Його тривалість не перевищує одного місяця та залишається сталою для забезпечення передбачуваності. Коротші цикли дають змогу швидше отримувати зворотний зв'язок, але збільшують частоту планування й аналізу.

Sprint Planning відповідає на три основні запитання:

1. Чому ця ітерація є цінною?
2. Що можна завершити протягом Sprint?

3. Як команда виконає вибрану роботу?

Daily Scrum не є звітуванням кожного учасника перед керівником. Його мета — перевірити рух до Sprint Goal і за потреби скоригувати план роботи.

Під час Sprint Review оцінюють не лише виконаний обсяг, а й зміни середовища, потреб користувачів і подальших пріоритетів. Sprint Retrospective зосереджується на процесі, взаємодії, інструментах, якості та можливостях удосконалення.

Поширеними помилками застосування Scrum є:

- відсутність чіткої мети продукту або Sprint;
- постійне додавання нових завдань до поточного Sprint;
- використання Daily Scrum як формального звіту;
- демонстрація незавершеної роботи;
- відсутність уповноваженого Product Owner;
- ігнорування технічного боргу;
- проведення ретроспектив без реалізації покращень;
- оцінювання продуктивності окремих працівників за швидкістю команди.

Scrum не усуває управління проєктом. Рішення щодо фінансування, договірних зобов'язань, ризиків, зовнішніх залежностей і взаємодії з керівництвом можуть залишатися поза межами Scrum-команди та потребувати загального проєктного управління.

14.5. Kanban: візуалізація потоку робіт та обмеження незавершеної роботи

Kanban — це стратегія оптимізації потоку цінності через процес. Вона не визначає обов'язкових ролей, ітерацій або фіксованого набору подій, тому може застосовуватися самостійно або доповнювати інші підходи.

Офіційний Kanban Guide 2025 визначає три взаємопов'язані практики:

- Визначення та візуалізація робочого процесу.
- Активне управління елементами в потоці.
- Удосконалення робочого процесу [6].

Визначення робочого процесу має встановлювати:

- які одиниці цінності переміщуються через систему;

- коли робота вважається розпочатою і завершеною;
- через які стани проходять елементи;
- як контролюється незавершена робота;
- які правила переходу діють між станами;
- яке очікування щодо строку надання послуги застосовується.

Процес часто візуалізують у вигляді Kanban-дошки. Найпростіший варіант може містити стани «Заплановано», «У роботі», «Перевірка» і «Завершено». Для реального проєкту структура має відображати фактичний потік, а не лише формальний статус завдань.

Основні елементи Kanban наведено в таблиці 14.3.

Таблиця 14.3 – Основні елементи системи Kanban

Елемент	Призначення
Візуалізація потоку	Робить видимими стан робіт, блокування, черги та перевантаження
Явні правила	Визначають умови початку, переходу й завершення робіт
Контроль WIP	Обмежує кількість розпочатих, але не завершених елементів
Витягувальна система	Нова робота починається лише за наявності вільної спроможності
Очікування щодо рівня обслуговування	Прогнозує строк завершення елемента з визначеною ймовірністю
Активне управління	Передбачає контроль старіння, блокувань і руху елементів
Удосконалення потоку	Змінює процес на основі фактичних спостережень і метрик
WIP	Кількість розпочатих, але не завершених елементів
Throughput	Кількість елементів, завершених за одиницю часу
Work Item Age	Час від початку роботи над активним елементом до поточного моменту
Cycle Time	Час від початку до завершення елемента

Обмеження незавершеної роботи є центральною практикою Kanban. Якщо команда одночасно розпочинає забагато завдань, зростають черги, перемикання контексту і час завершення. Контроль WIP спрямовує увагу на завершення вже розпочатої роботи.

Kanban використовує витягувальний принцип: нове завдання переходить у роботу, коли в системі з'являється вільна спроможність. Це не означає, що працівники повинні бути постійно повністю завантажені.

Метою є оптимізація загального потоку, а не максимізація зайнятості кожного учасника.

Scrum і Kanban мають різні акценти:

1. Scrum організовує роботу у фіксованих Sprint;
2. Kanban підтримує безперервний потік;
3. Scrum установлює конкретні відповідальності, події й артефакти;
4. Kanban адаптується до наявної організаційної структури;
5. Scrum контролює обсяг роботи переважно через Sprint Backlog;
6. Kanban безпосередньо контролює WIP у визначених частинах потоку.

Kanban особливо придатний для технічної підтримки, супроводу програмного забезпечення, обробки інцидентів та інших процесів, у яких завдання надходять безперервно й мають різну терміновість.

14.6. Інші гнучкі підходи до розроблення програмного забезпечення

Окрім Scrum і Kanban, у програмній інженерії застосовуються інші гнучкі підходи та набори практик.

Extreme Programming (XP) зосереджується на технічній якості та здатності програмного коду швидко змінюватися. До характерних практик належать:

- розроблення через тестування;
- парне програмування;
- постійний перегляд і покращення коду;
- безперервна інтеграція;
- малі випуски;
- колективна відповідальність за код;
- простий дизайн;
- тісна взаємодія із замовником.

XP доцільно застосовувати, коли вимоги часто змінюються, а якість і можливість безпечного внесення змін мають критичне значення.

Lean Software Development переносить принципи ощадливого управління у сферу програмного забезпечення. Основними ідеями є створення цінності, усунення втрат, зменшення черг, забезпечення якості в процесі роботи, оптимізація всієї системи та постійне вдосконалення.

Втратами можуть вважатися:

- непотрібна функціональність;
- очікування погоджень;
- надмірна кількість незавершеної роботи;
- передавання завдань між великою кількістю учасників;
- дефекти й повторне виконання;
- зайва документація;
- невикористані знання команди.

Crystal передбачає пристосування способу роботи до чисельності команди, критичності системи та особливостей проєкту. Основна увага приділяється комунікації, компетентності людей і регулярному створенню результату.

Feature-Driven Development організовує розроблення навколо невеликих функцій, цінних для користувача. Підхід передбачає формування загальної моделі, переліку функцій, планування, проєктування та реалізацію за функціями.

Scrumban поєднує окремі елементи Scrum і Kanban. Команда може зберігати планування, огляди й ретроспективи, але використовувати Kanban-дошку, контроль WIP і потокові метрики.

DevOps не є окремою Agile-методологією. Це сукупність культурних і технічних практик, спрямованих на співпрацю розроблення й експлуатації, автоматизацію тестування, інтеграції, розгортання, моніторингу та відновлення. DevOps доповнює Agile, скорочуючи шлях від готового програмного коду до його безпечного використання.

14.7. Переваги та недоліки Agile порівняно з традиційними підходами

До основних переваг Agile належать:

- раннє отримання працездатного результату;
- швидке виявлення неправильного розуміння вимог;
- регулярне отримання зворотного зв'язку;
- можливість змінювати пріоритети;
- прозорість поточного стану робіт;
- зменшення ризику створення непотрібної функціональності;
- поступове зниження технічної та продуктової невизначеності;
- постійне вдосконалення процесу;
- тісніша взаємодія команди й замовника;

- можливість раніше отримувати користь від окремих частин продукту.

Водночас Agile має обмеження. Його ефективне застосування потребує:

- доступного й уповноваженого представника замовника;
- зрілої та відповідальної команди;
- швидкого прийняття рішень;
- підтримки керівництва;
- можливості постачати продукт частинами;
- стабільного контролю якості;
- готовності переглядати початкові припущення;
- прозорого визначення пріоритетів.

До можливих недоліків і проблем належать:

- складність довгострокового прогнозування точного змісту;
- ризик постійної зміни пріоритетів;
- недостатня документація за неправильного розуміння Agile;
- накопичення технічного боргу;
- залежність від компетентності Product Owner і команди;
- складність роботи за контрактом із повністю фіксованим змістом;
- труднощі в організаціях із повільними процедурами погодження;
- формальне проведення подій без реальної адаптації;
- використання командних метрик для оцінювання окремих працівників.

Agile не гарантує автоматичного скорочення строків або бюджету. Його основна перевага полягає у швидшому отриманні інформації та можливості раніше змінити неправильний напрям.

Традиційний підхід також не потрібно розглядати як застарілий або неефективний. Він може бути доцільним за стабільних вимог, регламентованої послідовності робіт, значних договірних обмежень і високої вартості зміни затвердженого рішення.

14.8. Вибір гнучкого, традиційного або гібридного підходу

Вибір способу управління повинен залежати від характеристик проєкту, а не від популярності певної методології.

Потрібно враховувати:

- стабільність вимог;
- рівень технічної невизначеності;
- можливість поступового постачання;
- доступність замовника і користувачів;
- зрілість команди;
- критичність програмного продукту;
- нормативні й договірні вимоги;
- кількість зовнішніх залежностей;
- вартість внесення змін;
- потребу в довгостроковому прогнозуванні;
- організаційну структуру;
- швидкість прийняття рішень.

Основні умови вибору підходу наведено в таблиці 14.4.

Таблиця 14.4 – Вибір підходу до управління розробленням

Чинник	Гнучкий підхід	Традиційний підхід	Гібридний підхід
Вимоги	Змінні або неповні	Стабільні й детально визначені	Загальні вимоги стабільні, деталі уточнюються
Технологія	Висока невизначеність, потрібні експерименти	Добре відома й перевірена	Основна платформа відома, окремі компоненти експериментальні
Постачання	Можливе невеликими інкрементами	Результат має бути переданий як єдине ціле	Передбачено основні етапи та проміжні випуски
Участь замовника	Регулярна й активна	Переважно під час погодження та приймання	Активна для продукту, формальна для управлінських рішень
Планування	Адаптивне й поетапне	Детальне на початку	Загальний план наперед, деталізація за ітераціями
Зміни	Очікуються і регулярно пріоритизуються	Мінімізуються та проходять формальний контроль	Операційні зміни приймаються гнучко, суттєві — формально
Документація	Достатня для створення, використання і супроводу	Детально визначена стандартами та процедурами	Обов'язкова документація поєднується з гнучкими робочими матеріалами

Команда	Самоорганізована, міжфункціональна	Спеціалізовані ролі та формальна координація	Гнучкі команди працюють у межах загальної структури
Контроль	Через інкременти, зворотний зв'язок і поточні метрики	Через базовий план, контрольні точки і звіти	Поєднання контрольних точок із регулярними оглядами інкрементів
Найкраще застосування	Продуктова розробка, інновації, швидкі зміни	Стабільні та регламентовані проекти	Великі корпоративні й державні IT-проекти

Гібридний підхід поєднує загальне проєктне управління з гнучким створенням продукту. Наприклад, бюджет, основні строки, архітектурні обмеження та порядок приймання можуть бути визначені на рівні всього проєкту, а функціональність розроблятися короткими ітераціями.

Для проєкту впровадження університетської інформаційної системи можна заздалегідь визначити бюджет, інтеграції, вимоги до захисту даних і кінцеву дату запуску. Водночас інтерфейс, окремі функції та сценарії користувачів доцільно уточнювати за результатами прототипування й пілотного використання.

Незалежно від вибраного підходу необхідно зберігати:

- чіткі цілі;
- визначену відповідальність;
- контроль якості;
- управління ризиками;
- прозорість стану робіт;
- взаємодію із зацікавленими сторонами;
- обґрунтоване управління змінами;
- регулярну перевірку результатів.

Методологія повинна підтримувати досягнення цілей проєкту, а не перетворюватися на самоціль.

Висновки

Agile є системою цінностей і принципів, орієнтованою на раннє та регулярне створення цінного результату, співпрацю із замовником, адаптивне планування та постійне вдосконалення.

Ітеративне й інкрементальне розроблення дає змогу створювати продукт невеликими частинами, регулярно перевіряти припущення та використовувати зворотний зв'язок для уточнення подальших рішень.

Scrum організовує роботу через визначені відповідальності, події та артефакти. Його основою є Sprint, у межах якого команда створює придатний до використання інкремент і перевіряє рух до цілі продукту.

Kanban зосереджується на оптимізації потоку цінності. Він передбачає візуалізацію робочого процесу, контроль незавершеної роботи, активне управління елементами та використання потокових метрик.

Extreme Programming, Lean Software Development, Crystal, Feature-Driven Development, Scrumban і DevOps доповнюють гнучку розробку різними організаційними та технічними практиками.

Agile доцільний за умов змінних вимог, високої невизначеності та можливості регулярного отримання зворотного зв'язку. Традиційний підхід є придатним для стабільних і регламентованих проєктів, а гібридний дає змогу поєднати корпоративне управління з гнучким створенням продукту.

Успіх залежить не від формального використання назви Scrum, Kanban або Agile, а від здатності команди створювати цінний і якісний продукт, отримувати достовірний зворотний зв'язок та своєчасно пристосовувати спосіб роботи до реальних умов.

Список використаної літератури

1. Oguz A. Project Management: Navigating the Complexity with a Systematic Approach. 2nd ed. Cleveland, Ohio : MSL Academic Endeavors, 2025. ISBN 978-1-956812-06-0. Електронний ресурс. Режим доступу: <https://pressbooks.ulib.csuohio.edu/projectmanagement2ndedition/> (дата звернення: 23.08.2026).
2. Woods J. M., Marshall R. S., Schlesiger L. Delivering Value with Project Management. Dahlonega, Georgia : University of North Georgia Press, 2023. ISBN 978-1-959203-05-6. Електронний ресурс. Режим доступу: <https://alg.manifoldapp.org/projects/project-management-org1> (дата звернення: 23.08.2026).
3. European Commission, Directorate-General for Digital Services. PM² Project Management Methodology: Guide 3.1. Luxembourg : Publications Office of the European Union, 2023. ISBN 978-92-68-10314-2. DOI: 10.2799/970188. Електронний ресурс. Режим доступу: <https://op.europa.eu/en/publication-detail/-/publication/97cc2f12-c648-11ee-95d9-01aa75ed71a1/language-en> (дата звернення: 23.08.2026).
4. European Commission, Directorate-General for Digital Services. The PM²-Agile Guide. Version 3.0.1. Luxembourg : Publications Office of the European Union, 2021. ISBN 978-92-76-39410-5. DOI: 10.2799/162784. Електронний ресурс. Режим доступу: <https://op.europa.eu/en/publication-detail/-/publication/ed85debf-decc-11eb-895a-01aa75ed71a1/language-en> (дата звернення: 23.08.2026).
5. van der Hoorn B. Visuals for Influence: In Project Management and Beyond. Toowoomba : University of Southern Queensland, 2021. ISBN 978-0-6487698-6-6. DOI: 10.26192/q6qy7. Електронний ресурс. Режим доступу: <https://usq.pressbooks.pub/visualsforprojectmanagement/> (дата звернення: 23.08.2026).
6. Coleman J., Vacanti D., Johnson C. та ін. The Kanban Guide. May 2025. Електронний ресурс. Режим доступу: <https://kanbanguides.org/the-kanban-guide/> (дата звернення: 23.08.2026).
7. European Commission, Directorate-General for Digital Services. PM² Project Management Artefacts. Version 3.1. 2025. Електронний ресурс. Режим доступу: https://pm2.europa.eu/pm2-artefacts_en (дата звернення: 23.08.2026).