

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет
імені Івана Пулюя
Кафедра програмної інженерії



Методичні вказівки для самостійної роботи

з дисципліни

«Управління ІТ-проєктами»

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 01.09.2025.

Методичні вказівки для самостійної роботи з дисципліни «Управління ІТ-проектами» для здобувачів другого (магістерського) рівня вищої освіти ОПП «Інженерія програмного забезпечення» всіх форм навчання / Укладач: Стадник М.А., Цебрій О.Р., Стефанишин В.М., – Тернопіль: ТНТУ ім. І. Пулюя, 2025 – 41 с.

Відповідальний за випуск: Стефанишин В.М.

АНОТАЦІЯ

Самостійна робота студента (СРС) є невід'ємною складовою освітнього процесу та спрямована на поглиблення, узагальнення й закріплення теоретичних знань, отриманих під час лекційних занять, формування практичних навичок самостійного опрацювання навчального матеріалу, а також підготовку до виконання і захисту лабораторних робіт.

У межах вивчення вибіркової дисципліни «Управління ІТ-проектами» самостійна робота спрямована на формування у здобувачів другого (магістерського) рівня вищої освіти знань і практичних навичок планування, організації, реалізації, моніторингу та завершення ІТ-проектів. Особлива увага приділяється визначенню цілей і результатів проекту, формуванню проектної команди, плануванню робіт, ресурсів і термінів, управлінню ризиками, змінами, якістю, пріоритетами та зацікавленими сторонами.

Зміст дисципліни передбачає вивчення життєвого циклу ІТ-проекту, процесів його ініціації та планування, методів оцінювання трудомісткості й тривалості розроблення, принципів кількісного управління, організації командної взаємодії та застосування гнучких методологій розроблення програмного забезпечення. Практична складова спрямована на підготовку проекту, визначення ролей учасників, аналіз ризиків, оцінювання спроможності команди, управління ресурсами, стейкхолдерами і змінами, а також контроль прогресу, якості та завершення проекту.

Основні види самостійної роботи:

- опрацювання лекційного матеріалу та рекомендованої навчальної, наукової і професійної літератури;
- підготовка до виконання та захисту лабораторних робіт;
- самостійне опрацювання окремих розділів навчальної програми, які не виносяться на лекційні заняття;
- аналіз прикладів організації та реалізації ІТ-проектів;
- підготовка проектних матеріалів, моделей, планів, реєстрів ризиків та інших результатів, передбачених лабораторними завданнями;
- аналіз методів планування, оцінювання, моніторингу та контролю ІТ-проектів; опрацювання професійних стандартів, методологій, технічної документації та сучасних програмних засобів управління проектами;
- підготовка до поточного, модульного та підсумкового контролю знань.

Самостійна робота повинна сприяти розвитку здатності здобувачів аналізувати умови реалізації ІТ-проектів, визначати їхні цілі, обмеження, ресурси та ризики, обґрунтовувати управлінські рішення, організовувати взаємодію учасників проекту, оцінювати альтернативні варіанти дій і контролювати досягнення запланованих результатів.

ЗМІСТ

ВСТУП.....	6
ТЕМАТИЧНИЙ ПЛАН НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.....	9
КРИТЕРІЇ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ СТУДЕНТІВ.....	13
ТЕМАТИКА ТА ЗМІСТ САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ (СРС).....	17
МАТЕРІАЛИ ДЛЯ САМОСТІЙНОГО ВИВЧЕННЯ.....	22
ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ.....	32
РЕКОМЕНДОВАНІ ІНСТРУМЕНТАРІЇ ТА ДОДАТКОВІ РЕСУРСИ.....	36
ВИСНОВКИ.....	40
РЕКОМЕНДОВАНІ ДЖЕРЕЛА.....	41

ВСТУП

Сучасні інформаційні технології характеризуються високими темпами розвитку, постійною зміною вимог, обмеженістю ресурсів і необхідністю швидкого впровадження програмних продуктів. Успішна реалізація ІТ-проєкту залежить не лише від правильності технічних рішень, а й від якості планування, організації командної роботи, управління ресурсами, термінами, ризиками, змінами, комунікаціями та очікуваннями зацікавлених сторін.

Вибіркова навчальна дисципліна «Управління ІТ-проєктами» спрямована на формування у здобувачів знань і практичних навичок планування, організації, реалізації, моніторингу та завершення проєктів у сфері інформаційних технологій. Під час вивчення дисципліни розглядаються основні поняття проєктного управління, критерії успішності проєкту, його життєвий цикл, організаційна структура проєктної команди, процеси ініціації та планування, управління ризиками, ресурсами, пріоритетами, змінами, якістю і зацікавленими сторонами.

Важливою складовою освітнього процесу є самостійна робота студента (СРС). Вона забезпечує поглиблення та систематизацію знань, отриманих під час лекційних занять, підготовку до виконання лабораторних робіт, опрацювання окремих питань навчальної програми та формування здатності самостійно працювати з навчальною, науковою, нормативною і професійною інформацією. Робочою програмою дисципліни передбачено 78 годин самостійної роботи для денної та 104 години для заочної (дистанційної) форми навчання.

Головна мета самостійної роботи з дисципліни полягає у поглибленні теоретичних знань і розвитку практичних навичок підготовки, планування, організації, моніторингу та контролю ІТ-проєктів, а також формуванні здатності приймати обґрунтовані управлінські рішення в умовах невизначеності, ресурсних обмежень і зміни вимог.

Самостійна робота повинна сприяти формуванню здатності не лише відтворювати вивчений матеріал, а й аналізувати реальні та навчальні проєктні ситуації, порівнювати альтернативні підходи до управління, оцінювати ризики, визначати пріоритети, обґрунтовувати розподіл ресурсів і пропонувати способи досягнення цілей проєкту. Такий підхід відповідає програмним результатам навчання дисципліни, зокрема здатності оцінювати методи і моделі розроблення програмного забезпечення, управляти відповідними процесами на етапах життєвого циклу, приймати організаційно-управлінські рішення, оцінювати ризики та контролювати якість результатів.

У процесі самостійної роботи студенти повинні:

- поглибити знання щодо сутності, ознак і критеріїв успішності ІТ-проєктів;
- дослідити життєвий цикл проєкту, його основні фази, процеси та результати;

- навчитися визначати цілі, очікувані результати, припущення, обмеження і критерії приймання проєкту;
- опрацювати методи планування змісту, складу робіт, ресурсів і термінів виконання проєкту;
- поглибити знання щодо методів оцінювання трудомісткості та тривалості розроблення програмного забезпечення;
- засвоїти процеси ідентифікації, аналізу, оцінювання, моніторингу та контролю проєктних ризиків;
- навчитися аналізувати спроможність команди, визначати ролі учасників і планувати використання ресурсів;
- дослідити підходи до мотивації учасників і забезпечення ефективної командної взаємодії;
- сформулювати навички визначення, аналізу та залучення ключових учасників і зацікавлених сторін;
- опрацювати методи управління змінами, пріоритетами, якістю та прогресом виконання проєкту;
- засвоїти принципи кількісного управління та використання метрик для контролю стану проєкту;
- поглибити знання щодо гнучких методологій розроблення програмного забезпечення, зокрема Agile, Scrum і Kanban;
- підготуватися до виконання лабораторних робіт із підготовки проєкту, аналізу ризиків, управління ресурсами, стейкхолдерами, змінами, прогресом і якістю;
- набути навичок самостійного опрацювання професійної літератури, стандартів, методологій, технічної документації та сучасних інструментів управління ІТ-проєктами.

Перелік лекційних тем дисципліни охоплює основи управління проєктами, ініціацію і планування проєкту, управління ризиками, оцінювання трудомісткості й термінів розроблення, формування команди, реалізацію проєкту, управління пріоритетами, організаційні структури, мотивацію та взаємодію учасників, кількісне управління, роботу із зацікавленими сторонами та застосування Agile-методологій.

Лабораторні заняття доповнюють теоретичний матеріал практичними завданнями з підготовки проєкту і визначення ролей, аналізу ризиків, оцінювання спроможності команди, управління ресурсами, стейкхолдерами і змінами, а також оцінювання прогресу, контролю якості та завершення проєкту. Самостійна підготовка до лабораторних робіт передбачає попереднє опрацювання відповідних теоретичних положень, вибір методів та інструментів, підготовку вихідних даних і формування обґрунтованих висновків.

У межах самостійної роботи студентам рекомендується використовувати не лише навчальні матеріали курсу, а й сучасну професійну літературу, міжнародні

стандарти, методичні настанови, офіційну документацію та програмні засоби управління проєктами. Особливу увагу слід приділяти практичному аналізу проєктних ситуацій, оцінюванню управлінських рішень, роботі з проєктними обмеженнями, ризиками, метриками, пріоритетами та очікуваннями зацікавлених сторін.

Матеріали для виконання самостійної роботи, теоретичні питання, лабораторні завдання та інші навчально-методичні матеріали розміщуються в системі дистанційного навчання. Робочою програмою передбачено використання електронного курсу «Управління ІТ-проєктами» для організації навчальної та самостійної роботи студентів.

Важливою умовою виконання всіх видів самостійної роботи є дотримання принципів академічної доброчесності. Студент повинен самостійно виконувати поставлені завдання, коректно використовувати та цитувати інформаційні джерела, чітко позначати запозичені матеріали, не допускати академічного плагіату, фабрикації або фальсифікації результатів. Ці вимоги поширюються на підготовку проєктних матеріалів, аналітичних висновків, лабораторних звітів та виконання всіх форм контролю.

Формою підсумкового семестрового контролю з дисципліни є залік. Підготовка до нього передбачає систематизацію теоретичних знань, повторення основних понять і процесів управління ІТ-проєктами, аналіз результатів лабораторних робіт та опрацювання питань для самоконтролю.

Таким чином, самостійна робота з дисципліни «Управління ІТ-проєктами» повинна забезпечити системне поєднання теоретичних знань із практичними навичками проєктного управління та сприяти формуванню здатності самостійно планувати й аналізувати проєктну діяльність, приймати обґрунтовані організаційно-управлінські рішення, організовувати командну взаємодію та контролювати досягнення запланованих результатів ІТ-проєкту.

ТЕМАТИЧНИЙ ПЛАН НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Дисципліна: «Управління ІТ-проєктами».

Статус дисципліни: вибіркова.

Рівень: другий (магістерський) рівень вищої освіти.

Спеціальність: F2 «Інженерія програмного забезпечення».

Форма підсумкового контролю: залік.

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

Метою дисципліни є формування у здобувачів комплексу знань і практичних навичок із планування, організації, реалізації, моніторингу, контролю та завершення ІТ-проєктів. У межах дисципліни розглядаються принципи управління проєктами розроблення програмного забезпечення та інформаційних систем, організація проєктних команд, взаємодія учасників, управління ресурсами, термінами, ризиками, змінами, якістю, пріоритетами і зацікавленими сторонами.

Лекційний курс спрямований на формування системного розуміння основних процесів управління ІТ-проєктами. У процесі навчання розглядаються сутність і критерії успішності проєкту, його життєвий цикл, ініціація та планування, визначення цілей і результатів, оцінювання трудомісткості й термінів розроблення, формування команди, управління ризиками, пріоритетами, ресурсами, змінами, якістю та зацікавленими сторонами. Окрему увагу приділено принципам кількісного управління, мотивації учасників, організації ефективної взаємодії та застосуванню гнучких методологій Agile, Scrum і Kanban.

Метою проведення лабораторних занять є формування практичних навичок підготовки, планування, аналізу та контролю ІТ-проєктів. Лабораторна складова дисципліни передбачає підготовку проєкту і визначення ролей його учасників, аналіз проєктних ризиків, оцінювання спроможності команди, управління ресурсами, стейкхолдерами та змінами, а також оцінювання прогресу, контроль якості й завершення проєкту.

Важливою частиною вивчення дисципліни є самостійна робота здобувачів, спрямована на поглиблення лекційного матеріалу, підготовку до лабораторних занять, опрацювання окремих розділів навчальної програми, роботу з додатковою навчальною, науковою і професійною літературою, а також підготовку до поточного, модульного та підсумкового контролю. Робочою програмою передбачено 78 годин самостійної роботи для денної та 104 години для заочної

(дистанційної) форми навчання.

Завданням дисципліни є сформувати здатність аналізувати умови реалізації ІТ-проектів, визначати їхні цілі, результати, припущення й обмеження, планувати зміст і послідовність робіт, оцінювати трудомісткість, терміни та потребу в ресурсах, формувати проектну команду, управляти ризиками, змінами, якістю і взаємодією із зацікавленими сторонами. Дисципліна також повинна забезпечити формування здатності приймати обґрунтовані організаційно-управлінські рішення в умовах невизначеності та зміни вимог.

За результатами вивчення дисципліни студент повинен продемонструвати такі результати навчання:

- РН01. Знати і застосовувати сучасні професійні стандарти та інші нормативно-правові документи з інженерії програмного забезпечення.
- РН02. Оцінювати і вибирати ефективні методи й моделі розроблення, впровадження та супроводу програмного забезпечення, а також управління відповідними процесами на всіх етапах життєвого циклу.
- РН05. Розробляти, аналізувати, обґрунтовувати та систематизувати вимоги до програмного забезпечення.
- РН06. Розробляти й оцінювати стратегії проектування програмних засобів; обґрунтовувати, аналізувати та оцінювати варіанти проектних рішень з погляду якості кінцевого програмного продукту, ресурсних обмежень та інших чинників.
- РН08. Розробляти і модифікувати архітектуру програмного забезпечення для реалізації вимог замовника.
- РН12. Приймати ефективні організаційно-управлінські рішення в умовах невизначеності та зміни вимог, порівнювати альтернативи й оцінювати ризики.
- РН13. Конфігурувати програмне забезпечення, керувати його змінами та розробленням програмної документації на всіх етапах життєвого циклу.
- РН15. Здійснювати реінжиніринг програмного забезпечення відповідно до вимог замовника.
- РН16. Планувати, організовувати та здійснювати тестування, верифікацію і валідацію програмного забезпечення.
- РН18. Володіти критеріями успішності розроблення програмного забезпечення з метою визначення інвестиційної привабливості проектів.

Вивчення навчальної дисципліни передбачає формування та розвиток у студентів таких компетентностей.

Інтегральна компетентність. Здатність розв'язувати складні задачі та проблеми у сфері інженерії програмного забезпечення або у процесі навчання, що передбачає проведення досліджень та здійснення інновацій і характеризується невизначеністю умов та вимог.

Загальні компетентності:

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

Спеціальні (фахові) компетентності:

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

Знання, уміння, комунікація, автономія та відповідальність

Знання:

- Зн1. Спеціалізовані концептуальні знання, що охоплюють сучасні підходи, методи, моделі та інструменти управління ІТ-проєктами й утворюють основу для прийняття обґрунтованих управлінських рішень.
- Зн2. Критичне осмислення проблем управління ІТ-проєктами, зокрема питань планування, ресурсних обмежень, ризиків, зміни вимог, командної взаємодії та забезпечення якості.

Уміння:

- Ум1. Спеціалізовані уміння і навички розв'язання проблем, необхідні для планування, організації, реалізації, моніторингу та завершення ІТ-проєктів.
- Ум2. Здатність інтегрувати знання з програмної інженерії, проєктного менеджменту, управління ресурсами, ризиками, якістю та комунікаціями для розв'язання складних проєктних задач.
- Ум3. Здатність приймати рішення у нових або незнайомих ситуаціях за наявності неповної чи обмеженої інформації, оцінювати альтернативи та враховувати професійну, соціальну й етичну відповідальність.

Комунікація:

- K1. Зрозуміле і недвозначне представлення проєктних цілей, планів, результатів, управлінських рішень і висновків фахівцям, представникам замовника та іншим зацікавленим сторонам.

- K2. Організація ефективної взаємодії між учасниками проєктної команди та узгодження інтересів ключових зацікавлених сторін.

Автономія та відповідальність:

- AB1. Управління робочими або навчальними процесами, які є складними, непередбачуваними та потребують застосування нових стратегічних підходів.

- AB2. Відповідальність за прийняті проєктні рішення, використання ресурсів, управління ризиками, організацію командної роботи та досягнення запланованих результатів.

- AB3. Здатність продовжувати навчання з високим ступенем автономії, самостійно опановувати сучасні методології, стандарти та інструменти управління IT-проєктами.

КРИТЕРІЇ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ СТУДЕНТІВ

Теоретичний курс (тестування). Оцінюється рівень засвоєння теоретичного матеріалу дисципліни, розуміння основних процесів і принципів управління ІТ-проєктами, життєвого циклу проєкту, його ініціації та планування, управління ресурсами, термінами, ризиками, змінами, якістю, пріоритетами і зацікавленими сторонами. Враховується також здатність студента аналізувати проєктні ситуації, порівнювати альтернативні підходи та аргументовано застосовувати вивчені методи для прийняття управлінських рішень.

Лабораторні роботи. Оцінюється правильність і повнота виконання лабораторних завдань, практичне застосування теоретичних знань, обґрунтованість прийнятих управлінських рішень, коректність використання методів та інструментів управління проєктами, якість підготовлених проєктних матеріалів і сформульованих висновків. Лабораторний практикум охоплює підготовку проєкту та визначення ролей, аналіз ризиків, оцінювання спроможності команди, управління ресурсами, стейкхолдерами і змінами, а також оцінювання прогресу, контроль якості та завершення проєкту.

Форма підсумкового семестрового контролю — залік.

Розподіл балів за видами навчальної роботи

Вид навчальної роботи	Модуль 1	Модуль 2	Підсумковий контроль	Разом
Теоретичний курс (тестування)	15	20	—	35
Практична робота	20	20	—	40
Підсумковий контроль (залік)	—	—	25	25
Разом	35	40	25	100

Розподіл балів за лабораторні роботи

Модуль	Вид роботи	Кількість балів
Модуль 1	Лабораторна робота № 1. Підготовка проєкту та визначення ролей	7
	Лабораторна робота № 2. Аналіз ризиків та управління ними	6
	Лабораторна робота № 3. Аналіз спроможності команди та управління ресурсами	7
Разом за практичну складову модуля 1		20
Модуль 2	Лабораторна робота № 4. Управління стейкхолдерами	6
	Лабораторна робота № 5. Управління змінами в проєкті	6
	Лабораторна робота № 6. Оцінювання прогресу, контроль якості та завершення проєкту	8
Разом за практичну складову модуля 2		20

Теоретична складова першого модуля охоплює вступну лекцію та лекції № 1–7 і оцінюється максимально у 15 балів. Теоретична складова другого модуля охоплює лекції № 8–14 і оцінюється максимально у 20 балів.

Практична складова першого модуля формується за результатами виконання та захисту лабораторних робіт № 1–3 і становить 20 балів. Практична складова другого модуля формується за результатами виконання та захисту лабораторних робіт № 4–6 і також становить 20 балів.

Максимальна кількість балів за аудиторну та самостійну роботу протягом семестру становить 75 балів. Підсумковий контроль оцінюється максимально у 25 балів. Відповідно до робочої програми за кожні три бали семестрової оцінки студент автоматично отримує один бал підсумкової семестрової оцінки. Загальна максимальна оцінка з дисципліни становить 100 балів.

Оцінювання самостійної роботи

Результати самостійної роботи студентів враховуються під час поточного та модульного контролю, перевірки й захисту лабораторних робіт, тестування, опрацювання теоретичних питань і підготовки до заліку. Самостійна робота не розглядається окремо від інших видів навчальної діяльності, а інтегрується в загальну систему оцінювання дисципліни.

Під час оцінювання самостійної роботи враховуються:

- глибина опрацювання теоретичного матеріалу — розуміння основних понять, процесів, методів і принципів управління IT-проектами, здатність пояснювати їх взаємозв'язок та практичне значення;
- якість підготовки до лабораторних робіт — попереднє опрацювання відповідних методів, інструментів, проектних документів і теоретичних положень;
- самостійність аналізу проектних ситуацій — здатність визначати цілі, обмеження, ризики, ресурси, пріоритети, ролі учасників і потреби зацікавлених сторін;
- уміння працювати з інформаційними джерелами — використання навчальної та наукової літератури, професійних стандартів, методичних настанов, офіційної документації та інших достовірних джерел;
- обґрунтованість управлінських рішень — здатність порівнювати альтернативні варіанти, аргументувати вибір методів і пояснювати вплив запропонованих рішень на терміни, ресурси, ризики, якість і результати проекту;
- якість підготовлених матеріалів — повнота, структурованість, логічність і коректність проектних планів, реєстрів ризиків, переліків стейкхолдерів, аналітичних матеріалів та інших результатів роботи;
- дотримання академічної доброчесності — самостійне виконання завдань, належне використання і цитування джерел, відсутність академічного плагіату, фабрикації та фальсифікації результатів.

Критерії оцінювання результатів навчання

Оцінка «А» (90–100 балів)

Студент демонструє системне та глибоке розуміння принципів управління ІТ-проєктами, вільно оперує поняттями життєвого циклу, ініціації, планування, змісту, ресурсів, термінів, ризиків, якості, змін, пріоритетів і зацікавлених сторін. Лабораторні роботи виконані повністю та коректно, усі рішення належно обґрунтовані. Студент здатний аналізувати складні проєктні ситуації, порівнювати альтернативні підходи, оцінювати наслідки управлінських рішень і самостійно застосовувати отримані знання до нових задач.

Оцінка «В» (82–89 балів)

Студент добре володіє теоретичним матеріалом, правильно застосовує основні методи планування, організації, моніторингу та контролю ІТ-проєктів. Лабораторні завдання виконані якісно, можливі окремі незначні недоліки, які не впливають на загальну правильність результатів. Студент здатний обґрунтовувати розподіл ролей і ресурсів, аналізувати ризики, зміни та інтереси стейкхолдерів, але може допускати окремі неточності під час розгляду складніших проєктних ситуацій.

Оцінка «С» (74–81 бал)

Студент розуміє основний зміст дисципліни та здатний застосовувати вивчені підходи для розв'язання типових задач управління ІТ-проєктами. Лабораторні роботи виконані переважно правильно, але містять окремі помилки, неповні розрахунки або недостатньо обґрунтовані рішення. Розуміння окремих питань планування, оцінювання трудомісткості, управління ризиками, ресурсами, змінами, якістю або стейкхолдерами може бути неповним.

Оцінка «D–E» (60–73 бали)

Студент володіє мінімально необхідним обсягом знань із дисципліни, розуміє базові процеси управління ІТ-проєктами та може виконувати прості типові завдання. Лабораторні роботи містять суттєві недоліки, окремі завдання виконані неповно або недостатньо обґрунтовано. Під час пояснення прийнятих управлінських рішень студент допускає помилки, однак демонструє загальне розуміння основних тем курсу.

Оцінка «FX» (35–59 балів)

Студент демонструє фрагментарне розуміння теоретичного матеріалу. Наявні суттєві прогалини у знаннях процесів ініціації, планування, організації, моніторингу та контролю ІТ-проєктів. Студент має труднощі з визначенням цілей, обмежень, ризиків, ресурсів, ролей учасників і потреб зацікавлених сторін. Лабораторні роботи виконані частково або містять значні помилки, а прийняті рішення не мають достатнього обґрунтування. Для отримання позитивного результату необхідне додаткове опрацювання матеріалу та повторне проходження відповідних форм контролю.

Оцінка «F» (0–34 бали)

Студент не демонструє достатнього розуміння основних положень дисципліни, не володіє базовими принципами управління ІТ-проектами та не здатний застосовувати їх під час виконання практичних завдань. Значна частина лабораторних робіт не виконана або отримані результати не відповідають поставленим завданням. Студент не може обґрунтувати прийняті рішення, визначити основні параметри проєкту чи пояснити порядок застосування методів проєктного управління. Необхідне повторне вивчення дисципліни.

Під час виставлення підсумкової оцінки особлива увага приділяється не лише формальному виконанню завдань, а й самостійності роботи, розумінню прийнятих управлінських рішень, коректності використання професійної термінології, здатності аналізувати проєктні ситуації, оцінювати альтернативи та аргументовано обирати відповідні методи й інструменти управління ІТ-проектами.

ТЕМАТИКА ТА ЗМІСТ САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ (СРС)

Самостійна робота магістрантів передбачає поглиблене опрацювання теоретичного матеріалу дисципліни, підготовку до лабораторних занять, роботу з професійною літературою, стандартами, методичними настановами та документацією, а також самостійний аналіз проєктних ситуацій і управлінських рішень. Робочою програмою передбачено опрацювання окремих розділів, які не виносяться на лекційні заняття, підготовку до лабораторних робіт, поточного, модульного та підсумкового контролю.

Нижче наведено рекомендований перелік тем для самостійного опрацювання, сформований відповідно до змісту лекційного і лабораторного курсу дисципліни.

Тема 1. Ініціація, обґрунтування та визначення параметрів ІТ-проєкту

Мета: поглибити розуміння процесів ініціації ІТ-проєкту, навчитися формулювати його концепцію, цілі, очікувані результати, обмеження та критерії успішності.

Завдання для самостійного опрацювання:

1. Дослідити ознаки проєкту та визначити відмінності між проєктною, операційною і процесною діяльністю.
2. Проаналізувати основні фази життєвого циклу ІТ-проєкту та результати, які повинні бути отримані на кожній із них.
3. Для обраної проєктної ідеї сформулювати проблему, концепцію, мету, основні завдання та очікувані результати.
4. Визначити припущення, обмеження, основні ресурси, орієнтовні терміни та критерії приймання результатів проєкту.
5. Обґрунтувати корисність проєкту для замовника, користувачів та організації-виконавця.
6. Визначити основних учасників проєкту, розподілити їхні ролі, повноваження та відповідальність.
7. Сформулювати перелік критеріїв успішності ІТ-проєкту з урахуванням змісту, термінів, ресурсів, якості та задоволеності зацікавлених сторін.

Тема безпосередньо пов'язана з лекціями про сутність управління проєктами, життєвий цикл, ініціацію, концепцію, цілі та результати проєкту, а також із лабораторною роботою № 1 «Підготовка проєкту та визначення ролей».

Питання для самоконтролю:

1. Чим проєкт відрізняється від операційної діяльності?
2. Які фази утворюють життєвий цикл ІТ-проєкту?
3. Які відомості повинна містити концепція проєкту?
4. Чим мета проєкту відрізняється від його завдань і результатів?
5. Яку роль відіграють припущення та обмеження під час ініціації проєкту?

6. За якими критеріями можна оцінити успішність ІТ-проєкту?
7. Чому технічно реалізований проєкт не завжди можна вважати успішним?

Тема 2. Планування робіт, термінів, команди та ресурсів ІТ-проєкту

Мета: поглибити знання щодо планування змісту і складу робіт, оцінювання трудомісткості та тривалості проєкту, формування команди й ефективного використання ресурсів.

Завдання для самостійного опрацювання:

1. Дослідити принципи декомпозиції змісту проєкту та побудови ієрархічної структури робіт.
2. Для обраного ІТ-проєкту визначити основні етапи, пакети робіт, завдання та очікувані результати їх виконання.
3. Порівняти експертне, аналогове, параметричне і триточкове оцінювання трудомісткості та тривалості робіт.
4. Опрацювати основні положення методу функціональних точок і визначити можливості його застосування для оцінювання програмних проєктів.
5. Сформувані послідовність виконання робіт, визначити залежності між ними та підготувати базовий розклад проєкту.
6. Проаналізувати наслідки необґрунтовано скорочених або «агресивних» термінів виконання проєкту.
7. Визначити ролі учасників команди, їхню доступність, компетентності та відповідальність за виконання робіт.
8. Оцінити спроможність команди й порівняти її з плановим обсягом робіт.
9. Запропонувати способи усунення перевантаження, дефіциту компетентностей або нерівномірного розподілу ресурсів.

Зміст теми відповідає лекційним питанням планування проєкту, оцінювання трудомісткості й термінів розроблення, формування команди та організаційної структури. Тема також пов'язана з лабораторною роботою № 3 «Аналіз спроможності команди та управління ресурсами».

Питання для самоконтролю:

1. Для чого виконується декомпозиція змісту проєкту?
2. Які характеристики повинно мати коректно визначене проєктне завдання?
3. Чому оцінка трудомісткості має ймовірнісний характер?
4. У чому полягають переваги та обмеження методу функціональних точок?
5. Які чинники впливають на тривалість програмного проєкту?
6. Які негативні наслідки може спричинити необґрунтовано скорочений розклад?

7. Як визначити спроможність проєктної команди?
8. Чим роль учасника проєкту відрізняється від його посади в організації?
9. Як ресурсні обмеження впливають на зміст, терміни та якість проєкту?

Тема 3. Управління ризиками, змінами, якістю та прогресом проєкту

Мета: сформувати здатність виявляти й оцінювати проєктні ризики, планувати реагування на них, управляти змінами та використовувати кількісні показники для контролю прогресу і якості результатів.

Завдання для самостійного опрацювання:

1. Дослідити основні категорії ризиків ІТ-проєктів: технічні, організаційні, ресурсні, часові, комунікаційні та зовнішні.
2. Для обраного проєкту ідентифікувати потенційні ризики, визначити їхні причини, можливі наслідки та відповідальних осіб.
3. Виконати якісний аналіз ризиків за показниками ймовірності та впливу й визначити їхні пріоритети.
4. Сформувати реєстр ризиків і запропонувати заходи уникнення, зниження, передачі або прийняття кожного суттєвого ризику.
5. Визначити показники, за якими можна своєчасно виявити настання ризикової події.
6. Дослідити порядок подання, аналізу, погодження та впровадження змін у проєкті.
7. Проаналізувати вплив запропонованої зміни на зміст, терміни, ресурси, ризики та якість проєкту.
8. Визначити кількісні показники для оцінювання прогресу, продуктивності команди та якості результатів.
9. Сформувати критерії готовності результатів, приймання робіт і завершення проєкту.
10. Запропонувати порядок моніторингу проєкту та підготовки звітності про його поточний стан.

Тема пов'язана з лекціями про управління ризиками, реалізацію проєкту, способи реагування на ризики та принципи кількісного управління. Практичне опрацювання здійснюється під час виконання лабораторних робіт № 2 «Аналіз ризиків та управління ними», № 5 «Управління змінами в проєкті» і № 6 «Оцінювання прогресу, контроль якості та завершення проєкту».

Питання для самоконтролю:

1. Чим проєктний ризик відрізняється від проблеми, яка вже виникла?
2. Які етапи охоплює процес управління ризиками?
3. За якими показниками визначається пріоритет ризику?
4. Які основні стратегії реагування на негативні ризики?
5. Яку інформацію доцільно включати до реєстру ризиків?
6. Чому зміни в ІТ-проєкті повинні проходити формалізовану процедуру

оцінювання?

7. Які наслідки може мати неконтрольоване розширення змісту проєкту?
8. Які метрики можуть використовуватися для оцінювання прогресу проєкту?
9. Як пов'язані контроль якості та критерії приймання результатів?
10. За яких умов IT-проєкт може вважатися завершеним?

Тема 4. Управління стейкхолдерами, пріоритетами та командною взаємодією в гнучких проєктах

Мета: сформувати навички аналізу зацікавлених сторін, визначення пріоритетів, організації комунікацій і командної взаємодії, а також вибору доцільного підходу до управління IT-проєктом.

Завдання для самостійного опрацювання:

1. Визначити внутрішніх і зовнішніх стейкхолдерів обраного IT-проєкту та проаналізувати їхні інтереси, очікування, вплив і ставлення до проєкту.
2. Побудувати матрицю «влада — зацікавленість» і визначити стратегію взаємодії з кожною групою стейкхолдерів.
3. Сформувати план комунікацій із зазначенням одержувачів інформації, її змісту, формату, періодичності та відповідальних осіб.
4. Проаналізувати можливі конфлікти інтересів між замовником, командою, керівництвом та користувачами продукту.
5. Дослідити методи визначення пріоритетів проєктів і вимог, зокрема експертне оцінювання, бальне ранжування та метод MoSCoW.
6. Проаналізувати вплив мотивації, стилю лідерства, розподілу відповідальності та якості комунікацій на результативність команди.
7. Порівняти традиційний, гнучкий і гібридний підходи до управління IT-проєктами.
8. Дослідити основні ролі, події та артефакти Scrum, а також принципи візуалізації потоку робіт і обмеження незавершеної роботи в Kanban.
9. Визначити умови, за яких доцільно застосовувати Scrum, Kanban або гібридну модель управління.
10. Запропонувати спосіб організації взаємодії команди та стейкхолдерів в умовах зміни вимог.

Тема безпосередньо пов'язана з лекціями про управління пріоритетами, організаційну структуру компанії, мотивацію, ефективну взаємодію, ключових учасників, зацікавлені сторони та Agile-методології. Практичне застосування відповідних підходів передбачено лабораторною роботою № 4 «Управління стейкхолдерами».

Питання для самоконтролю:

1. Хто належить до зацікавлених сторін IT-проєкту?
2. За якими ознаками можна класифікувати стейкхолдерів?

3. Для чого використовується матриця «влада — зацікавленість»?
4. Які відомості повинен містити план комунікацій?
5. Які чинники впливають на визначення пріоритетів проєктів і вимог?
6. Як мотивація та стиль лідерства впливають на роботу проєктної команди?
7. У чому полягають основні відмінності між традиційним і гнучким управлінням?
8. Які ролі, події та артефакти визначено у Scrum?
9. Для чого в Kanban обмежують кількість незавершених завдань?
10. У яких випадках доцільно застосовувати гібридний підхід до управління проєктом?

Опрацювання наведених тем повинно бути спрямоване не лише на засвоєння додаткового теоретичного матеріалу, а й на формування здатності самостійно аналізувати проєктні ситуації, визначати цілі, обмеження, ризики, ресурси та інтереси стейкхолдерів, порівнювати альтернативні управлінські рішення й формулювати обґрунтовані висновки. Такий підхід узгоджується з програмними результатами навчання дисципліни щодо управління процесами життєвого циклу програмного забезпечення, прийняття організаційно-управлінських рішень, оцінювання ризиків, координації проєктної діяльності та забезпечення якості результатів.

МАТЕРІАЛИ ДЛЯ САМОСТІЙНОГО ВИВЧЕННЯ

Самостійне опрацювання матеріалу з дисципліни «Управління ІТ-проектами» спрямоване на поглиблення знань щодо планування, організації, реалізації, моніторингу та завершення проєктів у сфері інформаційних технологій, а також на формування здатності приймати обґрунтовані управлінські рішення в умовах невизначеності, ресурсних обмежень і зміни вимог.

Під час самостійної роботи студент повинен не лише ознайомлюватися з теоретичними положеннями, а й аналізувати їх практичне застосування, порівнювати альтернативні підходи, оцінювати вплив прийнятих рішень на зміст, терміни, ресурси, ризики та якість проєкту, а також формувати власні аргументовані висновки.

1. ІТ-проект та його життєвий цикл

Проект є тимчасовою діяльністю, спрямованою на створення унікального продукту, послуги або результату. ІТ-проект може передбачати розроблення програмного забезпечення, упровадження інформаційної системи, модернізацію цифрової інфраструктури, інтеграцію програмних компонентів або реалізацію іншої технологічної ініціативи.

Основними ознаками ІТ-проекту є:

- наявність визначеної мети;
- унікальність запланованого результату;
- обмежені терміни виконання;
- обмежені людські, технічні та фінансові ресурси;
- наявність взаємопов'язаних робіт;
- участь різних зацікавлених сторін;
- невизначеність і проєктні ризики;
- необхідність контролю якості результатів.

Необхідно розрізняти проєктну й операційну діяльність. Проєкт має визначений початок і завершення та спрямований на отримання унікального результату. Операційна діяльність є постійною або повторюваною і забезпечує функціонування організації.

Життєвий цикл ІТ-проекту охоплює послідовність фаз від виникнення проєктної ідеї до завершення робіт і приймання результатів. До основних фаз належать:

- ініціація;
- планування;
- реалізація;
- моніторинг і контроль;
- завершення.

Кожна фаза повинна мати визначені цілі, вхідні дані, перелік робіт,

відповідальних осіб і результати. Перехід між фазами доцільно здійснювати після перевірки досягнутих результатів і прийняття відповідного управлінського рішення.

Успішність ІТ-проєкту оцінюється не лише за дотриманням початкових термінів і ресурсних обмежень. Важливими критеріями також є відповідність результату вимогам, належна якість продукту, задоволеність замовника і користувачів, досягнення очікуваної цінності та можливість подальшого використання створеного продукту.

2. Ініціація та обґрунтування ІТ-проєкту

Ініціація передбачає початкове визначення сутності проєкту та прийняття рішення про доцільність його реалізації. На цьому етапі формується загальне бачення майбутнього продукту, визначаються проблема, мета, очікувані результати, основні учасники, обмеження та критерії приймання.

Під час ініціації необхідно визначити:

- проблему або потребу, яку повинен розв'язати проєкт;
- мету і завдання проєкту;
- очікуваний продукт або результат;
- основних замовників, користувачів та інших стейкхолдерів;
- попередній зміст робіт;
- припущення й обмеження;
- орієнтовні терміни та потребу в ресурсах;
- основні ризики;
- критерії приймання та успішності;
- очікувану користь від реалізації проєкту.

Мета проєкту повинна бути конкретною, вимірюваною, досяжною, актуальною та обмеженою в часі. Загальні формулювання, які не дозволяють перевірити результат, ускладнюють планування і подальше оцінювання успішності проєкту.

Припущення відображають умови, які вважаються правдивими для цілей планування, але ще не підтверджені. Обмеження визначають межі, у яких повинен реалізовуватися проєкт, зокрема доступні терміни, ресурси, технології, бюджет, нормативні вимоги або склад команди.

Обґрунтування проєкту повинно пояснювати, яку проблему він розв'язує, яку цінність створює, які ресурси потребує та які наслідки матиме відмова від його реалізації. На підставі цього обґрунтування приймається рішення про початок, відхилення або додаткове опрацювання проєктної ініціативи.

3. Планування змісту та структури робіт

Планування змісту визначає, які роботи повинні бути виконані для отримання запланованого результату. Недостатньо чітке визначення змісту може спричинити постійне додавання нових завдань, конфлікти із замовником, перевантаження команди та порушення термінів.

Під час планування необхідно розрізняти:

- зміст продукту — функції та характеристики створюваного результату;
- зміст проєкту — сукупність робіт, необхідних для створення цього результату.

Одним з основних засобів планування є ієрархічна структура робіт, яка передбачає послідовну декомпозицію проєкту на етапи, пакети робіт і конкретні завдання. Декомпозиція повинна здійснюватися до рівня, на якому можна оцінити трудомісткість, визначити відповідального, встановити термін і перевірити результат виконання роботи.

Для кожного завдання доцільно визначити:

- зміст і очікуваний результат;
- відповідального виконавця;
- необхідні ресурси;
- попередні та наступні роботи;
- орієнтовну трудомісткість;
- термін виконання;
- критерії готовності;
- можливі ризики.

Після визначення робіт встановлюються логічні залежності між ними та формується базовий розклад проєкту. Базовий план використовується як основа для порівняння запланованих і фактичних результатів під час реалізації проєкту.

4. Оцінювання трудомісткості та тривалості проєкту

Оцінювання трудомісткості й термінів є складним завданням, оскільки виконується на підставі неповної інформації та містить елемент невизначеності. Оцінка повинна розглядатися як обґрунтоване ймовірнісне твердження, а не як гарантоване значення.

До поширених методів оцінювання належать:

- експертне оцінювання;
- оцінювання за аналогією;
- параметричне оцінювання;
- оцінювання знизу вгору;
- триточкове оцінювання;
- метод функціональних точок;
- відносне оцінювання в гнучких проєктах.

Експертне оцінювання ґрунтується на досвіді фахівців. Оцінювання за аналогією використовує дані раніше реалізованих подібних проєктів. Параметричний підхід застосовує кількісні залежності між характеристиками продукту і витратами на його створення. Оцінювання знизу вгору передбачає окреме визначення трудомісткості кожного завдання з подальшим підсумовуванням результатів.

Триточкове оцінювання враховує оптимістичний, найбільш імовірний та песимістичний сценарії. Такий підхід дозволяє відобразити невизначеність і сформулювати реалістичніший прогноз.

Метод функціональних точок використовується для оцінювання функціонального розміру програмного продукту на основі його зовнішньої поведінки та кількості інформаційних функцій. Метод дозволяє здійснювати оцінювання до завершення програмної реалізації, однак потребує коректного визначення функціональних складових системи.

Необґрунтовано скорочений, або «агресивний», розклад може призвести до перевантаження працівників, збільшення кількості помилок, зниження якості, накопичення технічного боргу, конфліктів у команді та зриву запланованих термінів.

5. Формування команди та управління ресурсами

Проектна команда об'єднує фахівців, які виконують взаємопов'язані роботи для досягнення спільної мети. Ефективність команди залежить від правильного визначення ролей, достатності компетентностей, доступності ресурсів, якості комунікацій, мотивації та стилю керівництва.

На організацію роботи команди впливає структура підприємства. Основними типами організаційних структур є:

- функціональна;
- матрична;
- проектна;
- комбінована.

У функціональній структурі працівники підпорядковуються керівникам відповідних підрозділів. У проектній структурі команда формується безпосередньо для реалізації конкретного проекту. Матрична структура поєднує ознаки функціонального і проектного управління та передбачає розподіл повноважень між функціональним і проектним керівництвом.

Під час формування команди необхідно враховувати:

- професійні компетентності учасників;
- досвід виконання подібних робіт;
- доступність у запланований період;
- здатність до командної взаємодії;
- повноваження та відповідальність;
- можливість заміни критично важливих фахівців;
- потребу в навчанні або залученні зовнішніх спеціалістів.

Для представлення розподілу відповідальності може використовуватися матриця RACI, у якій для кожної роботи визначають виконавця, відповідального за результат, осіб, із якими необхідно консультуватися, та осіб, яких потрібно інформувати.

Спроможність команди визначається обсягом роботи, який вона може реально виконати за певний період. Планування не повинно базуватися на припущенні про стовідсоткову доступність усіх учасників, оскільки необхідно враховувати комунікації, організаційні завдання, відпустки, навчання, можливі затримки та інші витрати часу.

6. Управління ризиками ІТ-проєкту

Ризик є невизначеною подією або умовою, настання якої може позитивно чи негативно вплинути на цілі проєкту. Управління ризиками повинно здійснюватися протягом усього життєвого циклу, а не лише під час початкового планування.

Процес управління ризиками охоплює:

- планування управління ризиками;
- ідентифікацію ризиків;
- якісний аналіз;
- за необхідності кількісний аналіз;
- планування реагування;
- реалізацію запланованих заходів;
- моніторинг і контроль.

Ризики ІТ-проєктів можуть бути пов'язані з:

- нестабільністю або неповнотою вимог;
- використанням нових технологій;
- помилками в оцінюванні;
- дефіцитом фахівців;
- залежністю від окремих учасників;
- недостатньою якістю комунікацій;
- змінами пріоритетів замовника;
- зовнішніми постачальниками;
- інформаційною безпекою;
- недостатньою якістю продукту;
- порушенням запланованих термінів.

Для кожного суттєвого ризику доцільно визначити причину, можливу подію, наслідки, ймовірність, вплив, пріоритет, відповідальну особу, заплановану реакцію та ознаки наближення ризикової події.

Основними стратегіями реагування на негативні ризики є уникнення, зниження, передача та прийняття. Обрана стратегія повинна бути практично здійсненою, економічно доцільною та узгодженою з пріоритетами проєкту.

7. Управління стейкхолдерами та комунікаціями

Стейкхолдерами є особи, групи або організації, які впливають на проєкт, залежать від його результатів або вважають, що проєкт стосується їхніх інтересів. До них можуть належати замовники, користувачі, керівництво, члени команди, інвестори, постачальники, контролюючі органи та інші учасники.

Управління стейкхолдерами передбачає:

- їх ідентифікацію;
- аналіз інтересів, очікувань і рівня впливу;
- визначення поточного та бажаного рівня залучення;
- планування комунікацій;
- управління очікуваннями;
- моніторинг змін у ставленні до проєкту.

Для аналізу може використовуватися матриця «влада — зацікавленість». Вона дозволяє визначити, яких стейкхолдерів потрібно активно залучати, інтереси яких слід постійно враховувати, кого необхідно регулярно інформувати, а для кого достатньо періодичного спостереження.

План комунікацій повинен визначати:

- яка інформація передається;
- хто є її одержувачем;
- хто відповідає за підготовку;
- у якому форматі вона подається;
- які канали використовуються;
- з якою періодичністю здійснюється комунікація;
- яким чином забезпечується зворотний зв'язок.

Неефективна комунікація може спричинити неправильне розуміння вимог, конфлікти, дублювання робіт, затримки та неприйняття результатів проєкту.

8. Управління пріоритетами

Пріоритет визначає порівняльну важливість проєкту, вимоги, функції або завдання. Управління пріоритетами необхідне в умовах обмежених ресурсів, коли неможливо одночасно реалізувати всі запропоновані ініціативи.

На визначення пріоритету можуть впливати:

- стратегічна відповідність цілям організації;
- очікувана цінність;
- терміновість;
- рівень ризику;
- ресурсомісткість;
- залежності від інших робіт;
- вимоги законодавства або стандартів;
- потреби користувачів;
- наслідки відкладення;
- технічна доцільність.

Для визначення пріоритетів можуть застосовуватися експертне ранжування, бальне оцінювання, попарне порівняння, матриці цінності та складності, метод MoSCoW та інші підходи.

Пріоритети необхідно періодично переглядати, оскільки умови реалізації,

доступність ресурсів, потреби користувачів і стратегічні цілі організації можуть змінюватися. Зміна пріоритету повинна бути обґрунтованою та доведеною до відома учасників проєкту.

9. Управління змінами та конфігураціями

Зміни є природною складовою ІТ-проєктів, оскільки вимоги, технології, ринкові умови та очікування користувачів можуть уточнюватися протягом реалізації. Водночас неконтрольоване внесення змін створює ризик розширення змісту, перевантаження команди, зростання трудомісткості та порушення термінів.

Управління змінами передбачає:

- реєстрацію запиту на зміну;
- опис причини та очікуваного результату;
- аналіз впливу на зміст, терміни, ресурси, ризики та якість;
- розгляд альтернатив;
- погодження або відхилення;
- оновлення планів і проєктної документації;
- реалізацію затвердженої зміни;
- перевірку отриманого результату;
- інформування зацікавлених сторін.

Необхідно розрізняти зміну вимоги, виправлення дефекту, попереджувальну дію та коригувальну дію. Кожна з них має різне походження і може потребувати окремого порядку розгляду.

Управління конфігураціями забезпечує ідентифікацію, збереження і контроль версій проєктних артефактів. До таких артефактів можуть належати вимоги, плани, моделі, програмний код, налаштування, тестова документація та інструкції користувача.

10. Кількісне управління, контроль прогресу та якості

Кількісне управління передбачає використання вимірюваних показників для оцінювання стану проєкту та прийняття рішень. Метрики повинні бути пов'язані з цілями проєкту, мати зрозумілий спосіб обчислення та забезпечувати можливість своєчасно виявляти відхилення.

Для контролю проєкту можуть використовуватися такі групи показників:

- виконання запланованих робіт;
- дотримання термінів;
- використання ресурсів;
- продуктивність команди;
- стабільність вимог;
- кількість і критичність дефектів;
- результати тестування;
- стан ризиків;
- кількість і вплив змін;

- готовність результатів до приймання.

Під час моніторингу необхідно порівнювати фактичні результати з базовим планом, визначати причини відхилень, прогнозувати можливі наслідки та планувати коригувальні дії.

Контроль якості спрямований на перевірку відповідності проміжних і кінцевих результатів установленим вимогам та критеріям приймання. Якість повинна забезпечуватися протягом усього проєкту, а не перевірятися лише перед його завершенням.

Завершення проєкту передбачає:

- перевірку виконання запланованих робіт;
- приймання результатів замовником;
- закриття невирішених питань;
- передавання продукту в експлуатацію або супровід;
- упорядкування документації;
- звільнення ресурсів;
- підбиття підсумків;
- фіксацію отриманого досвіду.

11. Agile, Scrum і Kanban в управлінні IT-проєктами

Гнучкі підходи орієнтовані на поступове створення продукту, регулярне отримання зворотного зв'язку, тісну взаємодію із замовником та адаптацію до зміни вимог.

Agile ґрунтується на цінностях і принципах, які підкреслюють важливість взаємодії людей, працездатного продукту, співпраці із замовником та готовності реагувати на зміни. Agile не є окремою методологією, а визначає загальний підхід, який реалізується через конкретні фреймворки та практики.

Scrum передбачає ітеративне створення продукту короткими циклами. Основними складовими Scrum є:

- відповідальності Product Owner, Scrum Master і Developers;
- події Sprint, Sprint Planning, Daily Scrum, Sprint Review і Sprint Retrospective;
- артефакти Product Backlog, Sprint Backlog та Increment;
- цілі продукту і спринту;
- визначення готовності результату.

Kanban зосереджується на візуалізації потоку робіт, обмеженні кількості незавершених завдань, контролі часу проходження роботи та безперервному вдосконаленні процесу.

Під час вибору підходу необхідно враховувати:

- стабільність вимог;
- тривалість проєкту;
- можливість регулярно отримувати зворотний зв'язок;

- доступність замовника;
- досвід і розмір команди;
- нормативні та договірні обмеження;
- критичність створюваного продукту;
- потребу у формалізованій документації.

На практиці можуть застосовуватися гібридні моделі, які поєднують передбачуваність традиційного планування з адаптивністю гнучких підходів.

ТЕМИ ДЛЯ САМОСТІЙНИХ ДОСЛІДЖЕНЬ

Для поглибленого вивчення дисципліни студентам може бути запропоновано обрати одну тему та підготувати аналітичний звіт, реферат або презентацію.

Блок 1. Ініціація та стратегічне обґрунтування проєктів

1. Критерії успішності сучасних ІТ-проєктів.
2. Обґрунтування цінності ІТ-проєкту для організації.
3. Методи вибору ІТ-проєктів в умовах обмежених ресурсів.
4. Визначення цілей і результатів проєкту за принципом SMART.
5. Причини невдач ІТ-проєктів на етапі ініціації.
6. Управління припущеннями та обмеженнями проєкту.

Блок 2. Планування та оцінювання

1. Ієрархічна структура робіт як основа планування проєкту.
2. Порівняння методів оцінювання трудомісткості програмного забезпечення.

3. Застосування методу функціональних точок.
4. Триточкове оцінювання в умовах невизначеності.
5. Причини та наслідки необґрунтовано скорочених розкладів.
6. Планування спроможності команди.
7. Використання діаграм Ганта та мережевих моделей.

Блок 3. Команда, стейкхолдери та комунікації

1. Вплив організаційної структури на управління ІТ-проєктом.
2. Використання матриці RACI для розподілу відповідальності.
3. Мотивація учасників проєктної команди.
4. Управління конфліктами в ІТ-проєктах.
5. Особливості управління розподіленими командами.
6. Аналіз стейкхолдерів за матрицею «влада — зацікавленість».
7. Організація комунікацій із замовником і користувачами.

Блок 4. Ризики, зміни та якість

1. Головні ризики програмних проєктів і способи реагування.
2. Якісний і кількісний аналіз проєктних ризиків.
3. Реєстр ризиків як інструмент управління.
4. Неконтрольоване розширення змісту проєкту.
5. Організація процесу управління змінами.

6. Метрики прогресу та продуктивності команди.
7. Управління якістю протягом життєвого циклу проєкту.
8. Завершення проєкту та використання отриманого досвіду.

Блок 5. Гнучкі підходи та цифрові інструменти

1. Порівняння традиційного, гнучкого і гібридного управління.
2. Scrum як фреймворк управління розробленням продукту.
3. Kanban і управління потоком робіт.
4. Формування та пріоритизація беклогу продукту.
5. Використання Jira, Trello, GitHub Projects або Azure DevOps для планування робіт.
6. Застосування інформаційних панелей для моніторингу проєкту.
7. Використання штучного інтелекту для підготовки планів, аналізу ризиків і проєктної звітності.
8. Обмеження та ризики застосування AI-асистентів в управлінні IT-проєктами.

Результати самостійного дослідження повинні демонструвати не лише опрацювання літературних і професійних джерел, а й здатність студента аналізувати проєктні ситуації, наводити практичні приклади, порівнювати альтернативні управлінські підходи, оцінювати їхні переваги й обмеження та формулювати власні аргументовані висновки.

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

Основи управління ІТ-проектами та життєвий цикл проекту

1. Що таке проєкт і за якими ознаками проєктна діяльність відрізняється від операційної?
2. Які особливості розроблення програмного забезпечення необхідно враховувати під час управління ІТ-проектами?
3. У чому полягає роль проєктів у створенні та впровадженні інновацій?
4. Які основні процеси охоплює управління ІТ-проектом?
5. Що таке життєвий цикл проєкту та які результати формуються на його основних етапах?
6. Як пов'язані зміст, терміни, ресурси та якість проєкту?
7. Чим успішне виконання проєкту за планом відрізняється від досягнення очікуваної користі для замовника?

Ініціація проєкту, визначення цілей і пріоритетів

8. У чому полягає призначення ініціації проєкту та які рішення приймаються на цьому етапі?
9. Яку інформацію повинна містити концепція ІТ-проекту?
10. Чим відрізняються проблема, мета, завдання та очікувані результати проєкту?
11. Як сформулювати цілі проєкту так, щоб можна було перевірити їх досягнення?
12. Що таке припущення й обмеження проєкту та як вони впливають на його планування?
13. Як обґрунтувати корисність проєкту та визначити критерії приймання його результатів?
14. Які чинники й методи використовують для визначення пріоритетів проєктів за обмежених ресурсів?

Планування змісту, робіт і термінів проєкту

15. Які основні складові має план управління ІТ-проектом?
16. Чим зміст програмного продукту відрізняється від змісту проєкту?
17. Для чого визначають межі проєкту та перелік робіт, які до нього не входять?
18. Що таке ієрархічна структура робіт (WBS) та за якими принципами її формують?
19. Як визначають залежності між роботами та послідовність їх виконання?
20. Що таке критичний шлях і як затримка робіт на ньому впливає на термін завершення проєкту?
21. Що таке базовий розклад проєкту та як його використовують для контролю виконання робіт?

22. Які рішення щодо організаційної структури, управління конфігураціями та якості необхідно передбачити під час планування?

Оцінювання трудомісткості, тривалості та потреби в ресурсах

23. Чим трудомісткість роботи відрізняється від її календарної тривалості?

24. Чому оцінку трудомісткості або терміну виконання слід розглядати як імовірнісне твердження?

25. Які вихідні дані та припущення необхідні для обґрунтованого оцінювання розроблення програмного забезпечення?

26. У чому полягають експертне оцінювання, оцінювання за аналогією та оцінювання на основі декомпозиції робіт?

27. У чому полягає метод функціональних точок і для чого його застосовують?

28. Як оцінити трудомісткість багатокomпонентного продукту з урахуванням інтеграції та тестування?

29. Які негативні наслідки може спричинити «агресивний» розклад і чому збільшення кількості виконавців не завжди пропорційно скорочує тривалість розроблення?

Управління ризиками та способи реагування

30. Що таке проєктний ризик і чим він відрізняється від проблеми, яка вже виникла?

31. Які основні етапи охоплює управління ризиками ІТ-проєкту?

32. Які методи використовують для ідентифікації ризиків і яку інформацію включають до реєстру ризиків?

33. Чим якісний аналіз ризиків відрізняється від кількісного?

34. Як імовірність виникнення та вплив ризику використовують для визначення його пріоритету?

35. Які ризики є характерними для вимог, технологій, оцінювання термінів і командної роботи в програмних проєктах?

36. У чому полягають стратегії уникнення, зниження, передавання та прийняття ризику? Наведіть приклади їх застосування.

37. Як визначають відповідального за ризик, ознаки його реалізації та порядок моніторингу ефективності заходів реагування?

Формування команди, організаційні структури та управління ресурсами

38. Які чинники необхідно враховувати під час формування команди ІТ-проєкту?

39. Як визначають ролі, повноваження та відповідальність учасників проєкту?

40. Для чого використовують матрицю відповідальності RACI та яку інформацію вона відображає?

41. Чим відрізняються функціональна, матрична та проєктна організаційні

структури?

42. Як організаційна структура компанії впливає на повноваження керівника проєкту та доступність ресурсів?

43. Що таке спроможність команди та як її оцінити з урахуванням компетентностей, доступного часу й інших зобов'язань учасників?

44. Як виявити перевантаження працівників і конфлікти використання ресурсів між проєктами?

45. За якими критеріями слід приймати рішення про перерозподіл робіт, залучення додаткових фахівців або зміну розкладу?

Мотивація, комунікація та зацікавлені сторони

46. Чим внутрішня мотивація відрізняється від зовнішньої та як вони впливають на роботу команди?

47. Які дії керівника сприяють відповідальності, ініціативності та ефективній взаємодії учасників?

48. Хто належить до зацікавлених сторін ІТ-проєкту та як визначити їхні інтереси й очікування?

49. Як оцінити вплив і зацікавленість стейкхолдерів та використати ці оцінки для планування взаємодії?

50. Яку інформацію повинен містити план комунікацій проєкту?

51. Як діяти, якщо очікування замовника, користувачів і команди суперечать одне одному?

52. Як перевірити, що учасники однаково розуміють вимоги, прийняті рішення та критерії готовності результату?

Реалізація проєкту, робоче планування та управління змінами

53. Які основні завдання управління виникають під час реалізації ІТ-проєкту?

54. Чим поточне робоче планування відрізняється від формування загального плану проєкту?

55. Як організувати координацію залежних робіт і своєчасне усунення блокувань?

56. Які дії слід виконати після виявлення відхилення фактичного стану проєкту від плану?

57. Що таке запит на зміну та яку інформацію він повинен містити?

58. Як оцінити вплив запропонованої зміни на зміст, терміни, ресурси, ризики та якість проєкту?

59. Як організувати погодження, документування та виконання змін, щоб запобігти неконтрольованому розширенню змісту проєкту?

Кількісне управління, контроль якості та завершення проєкту

60. У чому полягає сутність кількісного управління ІТ-проєктом?

61. Яким вимогам повинні відповідати метрики, що використовуються для

прийняття управлінських рішень?

62. Які показники дають змогу оцінити прогрес, продуктивність, якість і стан ризиків проєкту?

63. Чому кількість витрачених годин або виконаних завдань не завжди відображає реальний прогрес проєкту?

64. Як порівнювати планові та фактичні показники, аналізувати причини відхилень і прогнозувати завершення робіт?

65. Чим забезпечення якості відрізняється від контролю якості та яку роль у них відіграють перевірки й тестування?

66. Які умови мають бути виконані для приймання результатів і формального завершення проєкту?

67. Які матеріали необхідно передати замовнику або команді супроводу та для чого документують отримані уроки?

Гнучкі підходи до розроблення програмного забезпечення

68. Що таке Agile та які його основні цінності й принципи?

69. Чим ітеративне розроблення відрізняється від інкрементального та як ці підходи поєднуються?

70. Яке призначення Scrum і які відповідальності мають Product Owner, Scrum Master та Developers?

71. Яке призначення Sprint, Sprint Planning, Daily Scrum, Sprint Review та Sprint Retrospective?

72. Що таке Product Backlog, Sprint Backlog та Increment і як із ними пов'язані Product Goal, Sprint Goal та Definition of Done?

73. У чому полягає підхід Kanban та як візуалізація потоку й обмеження незавершеної роботи допомагають управляти виконанням завдань?

74. Чим відрізняються WIP, пропускна здатність, вік робочого елемента та час циклу і для яких управлінських рішень їх використовують?

75. За якими критеріями слід вибирати традиційний, гнучкий або гібридний підхід до управління ІТ-проєктом? Наведіть приклади умов їх застосування.

РЕКОМЕНДОВАНІ ІНСТРУМЕНТАРІЇ ТА ДОДАТКОВІ РЕСУРСИ

Для виконання завдань самостійної роботи з дисципліни «Управління ІТ-проектами» рекомендується використовувати засоби планування робіт, організації командної взаємодії, оцінювання ресурсів, аналізу ризиків, ведення проектної документації та контролю прогресу. Вибір інструментів залежить від завдання, способу організації проекту та доступних ресурсів.

Наведені програмні засоби є рекомендованими варіантами. Одночасне використання всіх інструментів не потрібне. Перевагу доцільно надавати доступним рішенням, які забезпечують виконання завдання, збереження результатів і можливість їх перевірки викладачем.

Засоби календарного планування та управління роботами

- **GanttProject** — настільний застосунок для побудови діаграм Ганта, визначення ієрархії та залежностей робіт, контрольних точок і завантаження ресурсів.
- **OpenProject** — платформа для спільного планування проекту, встановлення дат виконання робіт, відображення залежностей і ведення календарного плану.

Ці засоби доцільно застосовувати для підготовки переліку робіт, визначення послідовності їх виконання, призначення відповідальних і формування розкладу. Під час аналізу плану необхідно враховувати залежності між завданнями, доступність виконавців та вплив затримок на кінцевий термін.

Студент повинен уміти пояснити вихідні припущення, обґрунтувати тривалість робіт і показати, як зміна окремого завдання впливає на розклад проекту.

Платформи управління завданнями та командної роботи

- **Jira** — засіб ведення беклогу, планування спринтів, використання Scrum-і Kanban-дошок та відстеження виконання робіт.
- **GitHub Projects** — інструмент планування у вигляді таблиць, дошок і дорожніх карт із прив'язкою завдань до GitHub Issues та запитів на внесення змін.

Під час самостійної роботи ці платформи можуть використовуватися для:

- формування й упорядкування переліку завдань;
- визначення пріоритетів, виконавців і термінів;
- встановлення станів робочого процесу;
- фіксації блокувань і залежностей;
- документування обговорень та прийнятих рішень;
- перегляду поточного стану проекту.

Для кожного завдання рекомендується зазначати очікуваний результат і критерії його завершення. Назви станів дошки повинні відповідати фактичному процесу роботи. Наявність електронної дошки сама по собі не забезпечує правильного застосування Scrum або Kanban.

Електронні таблиці для оцінювання, аналізу ризиків і ресурсів

Для розрахунків і підготовки проєктних реєстрів можуть використовуватися **Microsoft Excel**, **Google Sheets** або **LibreOffice Calc**.

У межах дисципліни електронні таблиці доцільно застосовувати для:

- оцінювання трудомісткості та тривалості робіт;
- розрахунку доступного робочого часу й завантаження команди;
- підготовки кошторису та порівняння варіантів використання ресурсів;
- ведення реєстру ризиків і матриці ймовірності та впливу;
- багатокритеріального оцінювання пріоритетів;
- формування матриці відповідальності RACI;
- ведення реєстрів зацікавлених сторін і змін;
- порівняння планових та фактичних показників.

Розрахункові матеріали повинні містити зрозумілі назви показників, одиниці вимірювання, вихідні дані та пояснення припущень. Формули слід перевіряти, а навчальні або змодельовані дані — чітко позначати.

Для реєстру ризиків рекомендується передбачити опис ризику, його причини й наслідки, оцінку ймовірності та впливу, відповідального, заходи реагування і поточний стан.

Засоби візуалізації проєктних процесів

- **draw.io (diagrams.net)** — засіб створення схем, який можна використовувати для представлення структури робіт, організаційної структури команди та процесів погодження рішень.

- **Mermaid** — інструмент текстового опису діаграм, придатний для підготовки схем процесів, послідовностей взаємодії та діаграм Ганта.

Візуальні матеріали мають допомагати пояснювати розподіл відповідальності, залежності між роботами та порядок виконання управлінських дій. Наприклад, схема опрацювання зміни може відображати подання запиту, оцінювання впливу, погодження, виконання та перевірку результату.

Схеми повинні узгоджуватися з текстом документації, календарним планом і переліком завдань.

Засоби документування та взаємодії із зацікавленими сторонами

Для підготовки концепції проєкту, планів, протоколів зустрічей і звітів можуть використовуватися текстові редактори та засоби спільного опрацювання документів. Зокрема, **Google Docs** підтримує спільне редагування, коментування та перегляд історії змін.

Як зразки структури проєктних документів доцільно використовувати **PM² Artefacts**. Набір містить шаблони статуту проєкту, робочого плану, матриці зацікавлених сторін, планів управління ризиками, якістю й комунікаціями, запитів на зміни, реєстрів і звітів.

Під час підготовки документації рекомендується:

- використовувати єдині назви проєкту, робіт і ролей;
- зазначати дату, версію та відповідального за документ;
- фіксувати рішення, підстави їх прийняття та подальші дії;
- розрізняти проєкти документів і погоджені версії;
- налаштовувати доступ відповідно до ролей учасників.

Шаблони необхідно адаптувати до навчального завдання. Заповнення документа повинно відображати прийняті рішення та особливості проєкту.

Засоби моніторингу прогресу та контролю якості

Для аналізу виконання робіт можна використовувати електронні таблиці та звіти обраної платформи управління завданнями. Зокрема, **звіти Jira** підтримують аналіз роботи за спринтами та потоку завдань.

Під час самостійної роботи доцільно опрацьовувати:

- порівняння планових і фактичних термінів;
- облік завершених, активних і заблокованих завдань;
- аналіз незавершеної роботи та часу виконання;
- контроль дефектів і результатів перевірок;
- перевірку відповідності результатів критеріям приймання;
- підготовку звіту про стан і завершення проєкту.

Для кожного показника потрібно визначити спосіб обчислення, джерело даних і період оновлення. У звіті необхідно пояснювати причини відхилень та запропоновані коригувальні дії.

Навчальні та методичні ресурси

Для поглибленого опрацювання матеріалу рекомендуються такі відкриті ресурси:

- **PM² Project Management** — матеріали Європейської комісії щодо життєвого циклу проєкту, управлінських ролей, процесів і документації.
- **Agile Manifesto українською мовою** — першоджерело для вивчення цінностей гнучкого розроблення програмного забезпечення.
- **Scrum Guide** — офіційний посібник зі Scrum; на сторінці доступні мовні версії та переклади.
- **The Kanban Guide** — настанова щодо визначення робочого процесу, управління потоком, контролю незавершеної роботи та використання метрик.
- **Project Management: Navigating the Complexity with a Systematic Approach** — відкритий підручник А. Огуза для системного вивчення управління проєктами.

Основою підготовки залишаються конспект лекцій, завдання лабораторних робіт і матеріали електронного курсу дисципліни. Додаткові джерела використовуються для уточнення понять, аналізу прикладів і порівняння управлінських підходів.

Інструменти на основі штучного інтелекту

За умови дотримання правил використання ІІІ, установлених викладачем, АІ-асистенти можуть застосовуватися як допоміжні засоби для:

- пояснення понять і методів управління проєктами;
- підготовки початкових варіантів структури робіт;
- пошуку можливих ризиків і заходів реагування;
- порівняння альтернативних управлінських рішень;
- перевірки узгодженості цілей, завдань і критеріїв приймання;
- редагування проєктної документації.

Студент повинен перевіряти отримані пропозиції, розрахунки та посилання, пояснювати прийняті рішення й зазначати використання ІІІ відповідно до вимог академічної доброчесності. Не можна подавати згенеровані оцінки як фактичні результати роботи команди або передавати зовнішнім сервісам конфіденційні проєктні дані.

Технічні вимоги та вибір інструментів

Для виконання більшості завдань достатньо персонального комп'ютера або ноутбука із сучасним браузером, офісним пакетом і доступом до Інтернету. Локальні застосунки дають змогу виконувати частину роботи без постійного підключення до мережі.

Перед вибором хмарного сервісу необхідно перевірити умови доступу, обмеження тарифного плану, можливість експорту та налаштування конфіденційності. Купівля платного програмного забезпечення не є необхідною, якщо доступні засоби забезпечують виконання навчального завдання.

Для базового комплексу роботи достатньо обрати один засіб планування та обліку завдань, один табличний редактор, текстовий редактор і засіб створення схем. Результати слід зберігати у вихідних форматах, а для перевірки — також у форматах, визначених завданням.

ВИСНОВКИ

Самостійна робота з дисципліни «Управління ІТ-проєктами» спрямована на закріплення теоретичних знань і розвиток навичок планування, організації, моніторингу та завершення проєктів. Методичні матеріали поєднують опрацювання лекційних тем, підготовку до лабораторних робіт, аналіз проєктних ситуацій і перевірку знань за питаннями для самоконтролю.

Виконання запропонованих завдань дає змогу розвивати вміння:

- визначати цілі, очікувані результати, обмеження та критерії успішності ІТ-проєкту;
- планувати склад і послідовність робіт, оцінювати трудомісткість, терміни та потребу в ресурсах;
- визначати ролі учасників, розподіляти відповідальність і враховувати спроможність команди;
- виявляти й оцінювати ризики, обирати заходи реагування та контролювати їх виконання;
- організовувати комунікацію, узгоджувати очікування зацікавлених сторін і визначати пріоритети;
- аналізувати вплив змін на зміст, терміни, ресурси та якість проєкту;
- використовувати метрики для оцінювання прогресу, виявлення відхилень і коригування планів;
- обґрунтовувати вибір традиційного, гнучкого або гібридного підходу до управління проєктом.

Практичне значення самостійної роботи полягає у підготовці взаємопов'язаних проєктних матеріалів: планів робіт, оцінок ресурсів, реєстрів ризиків і змін, матриць відповідальності та звітів про прогрес. Використання програмних інструментів допомагає впорядкувати ці матеріали, виконати розрахунки та представити результати. Студент повинен пояснювати вихідні припущення, перевіряти достовірність даних і обґрунтовувати запропоновані рішення.

Очікуваним результатом є здатність самостійно аналізувати умови реалізації ІТ-проєкту та приймати узгоджені управлінські рішення з урахуванням ризиків, ресурсних обмежень і потреб замовника. Сформовані навички створюють основу для відповідальної участі в командній розробці та управління процесами створення, впровадження й супроводу програмного забезпечення.

РЕКОМЕНДОВАНІ ДЖЕРЕЛА

Базова література

1. Медведєва М. О., Шаров С. В., Колмакова В. О. Управління ІТ-проектами : навчальний посібник. Умань : Видавець «Сочінський М. М.», 2025. 176 с. [Електронний ресурс]. Режим доступу: <https://dspace.udpu.edu.ua/xmlui/handle/123456789/18699?show=full>.
2. Блага Н. В. Управління проектами : навчальний посібник. Львів : Львівський державний університет внутрішніх справ, 2021. 152 с. [Електронний ресурс]. Режим доступу: <https://dspace.lvduvs.edu.ua/handle/1234567890/3870>.
3. Oguz A. *Project Management: Navigating the Complexity with a Systematic Approach*. 2nd ed. Cleveland : MSL Academic Endeavors, 2025. [Електронний ресурс]. Режим доступу: <https://pressbooks.ulib.csuohio.edu/projectmanagement2ndedition/>.
4. Woods J. M., Marshall R. S., Schlesiger L. *Delivering Value with Project Management*. Dahlenega : University of North Georgia Press, 2023. [Електронний ресурс]. Режим доступу: <https://ung.edu/university-press/books/delivering-value-with-project-management.php>.
5. European Commission, Directorate-General for Digital Services. *PM² Project Management Methodology: Guide 3.1*. Luxembourg : Publications Office of the European Union, 2023. DOI: 10.2799/970188. [Електронний ресурс]. Режим доступу: <https://op.europa.eu/en/publication-detail/-/publication/97cc2f12-c648-11ee-95d9-01aa75ed71a1/language-en>.
6. European Commission, Directorate-General for Digital Services. *The PM²-Agile Guide*. Version 3.0.1. Luxembourg : Publications Office of the European Union, 2021. DOI: 10.2799/162784. [Електронний ресурс]. Режим доступу: <https://op.europa.eu/en/publication-detail/-/publication/ed85debf-decc-11eb-895a-01aa75ed71a1/language-en>.

Допоміжна література

7. Управління проектами. Лабораторний практикум : навчальний посібник / уклад. Б. Я. Корнієнко. Київ : КПП ім. Ігоря Сікорського, 2023. 68 с. Посібник містить завдання з аналізу методологій, ризиків, планування та визначення потреб проекту. [Електронний ресурс]. Режим доступу: <https://ela.kpi.ua/bitstreams/e0f9010b-937a-4859-9536-ba60c8271981/download>.
8. van der Hoorn B. *Visuals for Influence: In Project Management and Beyond*. University of Southern Queensland, 2021. Видання присвячене візуальному представленню проектної інформації та взаємодії із зацікавленими сторонами. [Електронний ресурс]. Режим доступу: <https://usq.pressbooks.pub/visualsforprojectmanagement/>.
9. Schwaber K., Sutherland J. *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game*. November 2020. Посібник визначає відповідальності, події, артефакти та правила Scrum. [Електронний ресурс]. Режим доступу:

<https://scrumguides.org/download.html>.

10. Coleman J., Vacanti D., Johnson C. et al. *The Kanban Guide*. May 2025. Настанова охоплює визначення робочого процесу, контроль незавершеної роботи, управління потоком і використання метрик. [Електронний ресурс]. Режим доступу: <https://kanbanguides.org/the-kanban-guide/>.

Інформаційні ресурси в мережі Інтернет

11. European Commission. **PM² Artefacts**. Version 3.1. 2025. Шаблони проєктних документів: статуту, робочого плану, матриці зацікавлених сторін, планів управління ризиками, якістю та комунікаціями, запитів на зміни, реєстрів і звітів. [Електронний ресурс]. Режим доступу: https://pm2.europa.eu/pm2-artefacts_en.

12. Agile-маніфест розробки програмного забезпечення. Український переклад. Першоджерело для опрацювання цінностей гнучкого розроблення; містить посилання на основні принципи Agile. [Електронний ресурс]. Режим доступу: <https://agilemanifesto.org/iso/uk/manifesto.html>.

13. Atlassian. Agile Project Management with Jira. Офіційні матеріали щодо застосування Jira для ведення беклогу, планування спринтів, роботи зі Scrum- і Kanban-дошками та контролю виконання завдань. [Електронний ресурс]. Режим доступу: <https://www.atlassian.com/software/jira/agile>.

14. GitHub Docs. Planning and Tracking with Projects. Офіційна документація щодо організації проєктних таблиць, дощок і дорожніх карт, використання полів, фільтрів та зв'язків із завданнями й запитом на внесення змін. [Електронний ресурс]. Режим доступу: <https://docs.github.com/en/issues/planning-and-tracking-with-projects>.

15. OpenProject. User Guide. Офіційна документація щодо управління роботами, календарного планування, діаграм Ганта, залежностей, командної взаємодії та проєктної документації. [Електронний ресурс]. Режим доступу: <https://www.openproject.org/docs/user-guide/>.

16. GanttProject. Official GanttProject Repository. Офіційний ресурс програмного засобу для побудови діаграм Ганта, визначення ієрархії та залежностей робіт, контрольних точок і завантаження ресурсів. [Електронний ресурс]. Режим доступу: <https://github.com/bardsoftware/ganttproject>.