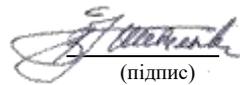


Міністерство освіти і науки України
Національний аерокосмічний університет
«Харківський авіаційний інститут»

Кафедра інженерії програмного забезпечення (№ 603)

ЗАТВЕРДЖУЮ

Гарант освітньої програми


(підпис)

Ілона ШЕВЧЕНКО
(ініціали та прізвище)

«29» серпня 2025 р.

СИЛАБУС ОBOB'ЯЗKОВОЇ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОЗНАЙОМЧА ПРАКТИЧНА

(назва навчальної дисципліни)

Галузь знань: 12 (F) Інформаційні технології
(шифр і найменування галузі знань)

Спеціальність: 121 (F2) Інженерія програмного забезпечення
(код і найменування спеціальності)

Освітня програма: Інженерія програмного забезпечення
(найменування освітньої програми)

Рівень вищої освіти: перший (бакалаврський)

Силабус введено в дію з **01.09.2025 р.**

Розробник: ст. викл. каф. 603, Оксана ЛУЧШЕВА
(посада, науковий ступінь і вчене звання, ім'я та прізвище)


(підпис)

Силабус навчальної дисципліни розглянуто на засіданні кафедри _____
інженерії програмного забезпечення
(назва кафедри)

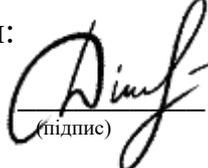
Протокол № 1 від «29» серпня 2025 р.

Завідувач кафедри д.т.н., проф.
(науковий ступінь і вчене звання)


(підпис)

Ігор ТУРКІН
(ім'я та прізвище)

Погоджено з представником здобувачів освіти:


(підпис)

Діана ДИКУН
(ініціали та прізвище)

1. Загальна інформація про викладача



ПІБ: Лучшева Оксана Вадимівна

Посада: старший викладач кафедри інженерії програмного забезпечення

Науковий ступінь: немає

Вчене звання: немає

Перелік дисциплін, які викладає:

Обчислювальна техніка та програмування,
Основи статистики у Python

Напрями наукових досліджень:

інженерія програмного забезпечення зі застосуванням
обробки даних

Контактна інформація:

e-mail: *o.luchsheva@khai.edu*

2. Опис навчальної дисципліни

Форма здобуття освіти	Денна
Семестр	4
Мова викладання	Українська
Тип дисципліни	Обов'язкова
Обсяг дисципліни: кредити ЄКТС/ кількість годин	3 кредитів ЄКТС / 90 годин (СРЗ – 90)
Види навчальної діяльності	Самостійна робота
Види контролю	Семестровий контроль – диференційний залік
Пререквізити	«Основи програмної інженерії», «Основи програмування», «Програмування мовою С#», «Об'єктно-орієнтоване програмування», «UI/UX дизайн»

3. Мета та завдання навчальної дисципліни, переліки компетентностей та очікуваних результатів навчання

Мета – формування у студентів цілісного уявлення про професійне середовище розробки програмного забезпечення та ознайомлення з особливостями функціонування ІТ-компанії. Дисципліна спрямована на вивчення практичного інструментарію розробки, процесів управління проєктами, принципів командної взаємодії, а також на розуміння ролей учасників розробки та вимог до технічної документації.

Завдання – опанування студентами практичних навичок щодо організації життєвого циклу розробки програмного забезпечення та роботи з сучасними інженерними інструментами. Студенти мають навчитися керувати задачами в середовищі Jira (Scrum/Kanban), застосовувати систему контролю версій Git для виконання базових та командних операцій, проєктувати бізнес-процеси у нотації BPMN та моделювати структуру сайтів. Також завданням є набуття досвіду тестування API через Postman, дослідження принципів архітектури SOLID, ідентифікація антипатернів та розвиток навичок систематизації результатів роботи у формі щоденника практики та підсумкового технічного звіту.

Компетентності, які набуваються

Інтегральна компетентність:

Здатність розв'язувати складні спеціалізовані завдання або практичні проблеми інженерії програмного забезпечення, що характеризуються комплексністю та невизначеністю умов, із застосуванням теорій та методів інформаційних технологій.

Загальні компетентності (ЗК)

Після закінчення цієї програми здобувач освіти буде мати:

- ЗК01. Здатність до абстрактного мислення, аналізу та синтезу.
- ЗК02. Здатність застосовувати знання у практичних ситуаціях.
- ЗК03. Здатність спілкуватися державною мовою як усно, так і письмово.
- ЗК04. Здатність спілкуватися іноземною мовою як усно, так і письмово.
- ЗК05. Здатність вчитися і оволодівати сучасними знаннями.
- ЗК06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
- ЗК07. Здатність працювати в команді.
- ЗК08. Здатність діяти на основі етичних міркувань.
- ЗК10. Здатність діяти соціально відповідально та свідомо.
- ЗК13. Здатність ухвалювати рішення та діяти, дотримуючись принципу неприпустимості корупції та будь-яких інших проявів не доброчесності.

Спеціальні компетентності (СК або ФК)

Після закінчення цієї програми здобувач освіти буде мати:

- ФК02. Здатність брати участь у проєктуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.
- ФК03. Здатність розробляти архітектури, модулі та компоненти програмних систем.
- ФК07. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних.
- ФК08. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.

ФК09. Здатність оцінювати і враховувати економічні, соціальні, технологічні та екологічні чинники, що впливають на сферу професійної діяльності.

ФК10. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.

ФК11. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.

ФК12. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.

ФК13. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

ФК14. Здатність до алгоритмічного та логічного мислення.

ФК16. Здатність розробляти методичні, інформаційні, математичні, алгоритмічні та програмні засоби реалізації інформаційних технологій

Програмні результати навчання (ПРН):

ПРН01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.

ПРН02. Знати кодекс професійної етики, розуміти соціальну значимість та культурні аспекти інженерії програмного забезпечення і дотримуватись їх в професійній діяльності.

ПРН03. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.

ПРН04. Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.

ПРН05. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.

ПРН07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

ПРН08. Вміти розробляти людино-машинний інтерфейс.

ПРН12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.

ПРН13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

ПРН14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.

ПРН15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.

ПРН16. Мати навички командної розробки, погодження, оформлення і випуску всіх видів програмної документації.

ПРН23. Вміти документувати та презентувати результати розробки програмного забезпечення.

4. Зміст навчальної дисципліни

МОДУЛЬ 1

Змістовний модуль 1. Інструментальні засоби та процеси розробки програмного забезпечення

Тема 1. Організація ІТ-компанії та життєвий цикл програмного забезпечення

Стисла анотація:

Вивчення особливостей організації та функціонування сучасної ІТ-компанії: структура підрозділів, взаємодія між відділами та корпоративна культура. Детальний розгляд фаз життєвого циклу розробки програмного забезпечення (SDLC): аналіз вимог до програмних продуктів, проєктування архітектури, написання коду, тестування та впровадження. Ознайомлення з принципами командної розробки, поняттям безперервної інтеграції та доставки (CI/CD) та роллю систем контролю версій у промисловому виробництві програмного забезпечення.

Самостійна робота здобувача освіти:

Складання та узгодження індивідуального графіку проходження практики. Ознайомлення з регламентом роботи ІТ-підприємства. Аналіз етапів життєвого циклу розробки ПЗ (SDLC) та організаційної структури компанії. Опрацювання навчального матеріалу щодо стандартів CI/CD та підготовка питань до керівника. Початок ведення щоденника практики: фіксація плану роботи та перших результатів ознайомлення з процесами.

Тема 2. Управління проєктами та задачами у середовищі Jira

Стисла анотація:

Ознайомлення з гнучкими методологіями розробки Scrum та Kanban. Практичне опанування інструменту Jira для створення та ведення життєвого циклу задач: робота з типами задач (Epic, Story, Task, Sub-task, Bug). Участь у проєктному борді: управління беклогом (backlog), планування та ведення активних спринтів (active sprint), налаштування робочих процесів (workflow). Виконання базових дій: зміна статусів, додавання коментарів, прикріплення файлів та оцінка часових витрат (estimation).

Самостійна робота здобувача освіти:

Опрацювання навчальних матеріалів (відео, гайдів) щодо інтерфейсу Jira та принципів Agile-розробки. Моделювання робочого процесу та керування завданнями у середовищі Jira: створення, редагування та оцінювання задач різних типів. Практична робота з проєктним бордом та переведення завдань за статусами. Оформлення звіту з виконаної роботи та щоденне додавання результатів (скріншотів) у щоденник практики.

Тема 3. Робота з системою контролю версій Git та інтеграція з Jira

Стисла анотація:

Вивчення принципів роботи з розподіленими системами контролю версій на прикладі Git. Опанування базового та розширеного інструментарію: ініціалізація

репозиторіїв, робота з областю підготовки файлів (staging area), фіксація змін та синхронізація з віддаленими серверами (GitHub / GitLab). Детальний розгляд стратегій розгалуження та методів безпечного злиття гілок із розв'язанням можливих конфліктів у програмному коді. Налаштування інтеграції системи контролю версій із середовищем Jira для автоматизації процесів: оновлення статусів завдань через коментарі до фіксацій змін, візуалізація активності розробника та забезпечення прозорості етапів реалізації проєкту.

Самостійна робота здобувача освіти:

Проходження інтерактивного тестування та опрацювання навчальних матеріалів з Git (W3Schools, GitHowTo). Практичне виконання операцій із розгалуження та злиття коду у власному репозиторії. Керування версіями програмного коду та налаштування взаємодії репозиторію з системою управління проєктами (Jira). Перевірка коректності інтеграції: створення зв'язків між фіксаціями змін та відповідними завданнями. Фіксація посилань на репозиторії у щоденнику практики та підготовка питань до керівника.

Тема 4. Моделювання бізнес-процесів за допомогою Bizagi Modeler

Стисла анотація:

Опанування методології візуалізації бізнес-процесів із використанням міжнародного стандарту нотації BPMN 2.0. Детальний розгляд архітектури діаграм: робота з об'єктами потоку (події, дії, шлюзи), сполучними об'єктами та об'єктами даних. Вивчення принципів структурування процесів за допомогою контейнерів (pools) та доріжок (lanes) для чіткого розподілу відповідальності між учасниками системи. Практичне застосування інструментарію Bizagi Modeler для побудови логічних моделей, проведення симуляції процесів та аналізу на предмет виявлення логічних помилок або неефективних етапів («вузьких місць») у роботі програмного продукту.

Самостійна робота здобувача освіти:

Опрацювання навчальних матеріалів щодо стандартів нотації BPMN та інтерфейсу Bizagi Modeler. Розробка графічної моделі бізнес-процесу в нотації BPMN для обраного програмного продукту. Валідація побудованої схеми на логічну коректність та відсутність помилок у потоках даних. Експорт графічного результату та внесення опису виконаних кроків моделювання у щоденник практики.

Тема 5. Проєктування структури сайту та тестування інтерфейсів взаємодії (API)

Стисла анотація:

Вивчення принципів побудови ієрархічних моделей програмних продуктів за допомогою сервісів ментальних карт (Coggle.it). Візуалізація архітектури та карти ресурсу з детальним описом функціонального призначення сторінок та логіки переходів. Ознайомлення з інструментарієм Postman для перевірки працездатності серверної частини застосунку. Опанування механізмів формування та виконання запитів за протоколом HTTP (GET, POST), аналіз кодів відповідей сервера, робота з параметрами та структурою даних у форматі JSON.

Самостійна робота здобувача освіти:

Опрацювання інструкцій та методичних рекомендацій щодо проектування мап ресурсів та основ роботи з інтерфейсами програмування додатків (API). Візуалізація архітектури сайту та функціональне тестування інтерфейсів взаємодії: побудова ієрархічної структури проєкту в середовищі Coggle.it. Практичне виконання перевірочних запитів до вебсервісів за допомогою Postman. Фіксація розробленої архітектури та результатів тестування у звіті, а також щоденне занесення відомостей про виконану роботу до щоденника практики.

Тема 6. Стандарти якості коду: принципи SOLID та антипатерни

Стисла анотація:

Ознайомлення з принципами SOLID як базовими правилами написання зрозумілого та якісного програмного коду. Розгляд основних концепцій: єдиної відповідальності, можливості розширення коду без зміни його структури, правильного використання успадкування та розділення інтерфейсів. Вивчення поняття антипатернів — поширених помилок у програмуванні, що ускладнюють подальшу роботу з проєктом. Формування навичок аналізу коду для пошуку та усунення таких недоліків.

Самостійна робота здобувача освіти:

Опрацювання навчальних матеріалів щодо принципів SOLID та типових помилок (антипатернів) у програмуванні. Дослідження архітектурних підходів та виявлення недоліків у коді: аналіз фрагментів програмного забезпечення на відповідність стандартам якості. Пошук порушень принципів SOLID у структурі проєкту. Підготовка коротких висновків із прикладами виявлених помилок та варіантами їх виправлення. Узагальнення результатів у фінальному звіті з практики та заповнення щоденника.

Тема 7. Систематизація результатів, підготовка презентації та захист результатів практики

Стисла анотація:

Оформлення щоденника практики з фіксацією щоденних записів про виконані завдання, застосовані інструменти та методи розв'язання технічних проблем. Підготовка підсумкового технічного звіту, що містить графічні результати виконання завдань у середовищах Jira, Git, Postman, Vizagi та Coggle. Розробка мультимедійної презентації за результатами практики та підготовка відеозапису захисту (демонстрації результатів), у якому студент презентує ключові досягнення та аналізує отримані практичні навички. Заповнення підсумкової анкети зворотного зв'язку.

Самостійна робота здобувача освіти:

Комплектування звітної документації, створення презентації та запис відеозахисту. Упорядкування щоденника практики та перевірка повноти технічного звіту. Розробка структури та візуального оформлення презентації. Підготовка сценарію та запис демонстраційного відео про результати проходження практики. Заповнення фінальної форми оцінювання результатів навчання та подання матеріалів на перевірку керівнику.

5. Індивідуальні завдання

Не передбачені навчальним планом.

6. Методи навчання

Словесні, наочні, практичні.

7. Методи контролю

Оцінювання здійснюється за результатами самостійного опрацювання теоретичного матеріалу та якістю виконання практичних завдань за кожною темою. Підсумковий контроль (диференційований залік) полягає у верифікації здатності здобувача освіти самостійно застосовувати професійний інструментарій, а саме: Jira, Git, Postman, Bizagi Modeler та Coggle, для розв'язання прикладних інженерних питань. Результати навчання підтверджуються поданим технічним звітом, оформленим щоденником практики та публічним захистом отриманих результатів.

8. Критерії оцінювання та розподіл балів, які отримують здобувачі освіти

Таблиця 8.1 – Розподіл балів, які отримують здобувачі освіти

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Модуль			
Змістовний модуль 1			
Виконання завдання за Темою 1 "Організація ІТ-компанії та SDLC" (аналіз структури та фаз розробки)	0...5	1	0...5
Виконання завдання за Темою 2 "Управління проектами у Jira" (налаштування беклогу та спринтів)	0...10	1	0...10
Виконання завдання за Темою 3 "Робота з Git та інтеграція з Jira" (репозиторій та зв'язок із задачами)	0...10	1	0...10
Виконання завдання за Темою 4 "Моделювання у Bizagi Modeler" (діаграма бізнес-процесу в BPMN)	0...15	1	0...15
Виконання завдання за Темою 5 "Проектування структури та API" (мапа ресурсу та тести у Postman)	0...15	1	0...15
Виконання завдання за Темою 6 "Стандарти якості коду: SOLID" (аналіз архітектури та антипатернів)	0...15	1	0...15
Виконання завдання за Темою 7 "Звіт та щоденник практики" (систематизація документації та СРЗ)	0...20	1	0...20
Публічний захист (відеопрезентація результатів)	0...10	1	0...10
Усього за дисципліну			0...100

Підсумкова оцінка за ознайомчу практику складається з балів за виконання завдань за темами 1-6 та підготовку звітної документації (тема 7). Максимально за ці етапи можна отримати 90 балів. Публічний захист (презентація) додає ще 10 балів і є обов'язковим для отримання оцінки «відмінно» (90–100 балів).

Таблиця 8.2 – Шкали оцінювання: бальна і традиційна

Сума балів	Оцінка за традиційною шкалою	
	Іспит, диференційний залік	Залік
90 – 100	Відмінно	Зараховано
75 – 89	Добре	
60 – 74	Задовільно	
0 – 59	Незадовільно	Не зараховано

Критерії оцінювання роботи здобувача протягом семестру

Задовільно (60–74). Виставляється, якщо здобувач виявив мінімально необхідні знання щодо процесів розробки ПЗ. Він володіє базовими навичками роботи з інструментарієм (Jira, Git), здатний за зразком побудувати просту модель бізнес-процесу або виконати запит у Postman. Аналіз якості коду та системні висновки робляться лише за суттєвої допомоги викладача. Звітна документація (звіт та щоденник) оформлена з певними недоліками, але відображає виконання основних етапів практики.

Добре (75–89). Виставляється, якщо здобувач демонструє достатньо повні знання та впевнено володіє програмним інструментарієм. Здобувач самостійно налаштовує робоче середовище в Jira, коректно працює з гілками в Git, будує логічно правильні BPMN-схеми в Bizagi та проводить тестування API. Допускаються незначні помилки в аналізі архітектурних принципів SOLID. Звітна документація підготовлена вчасно, результати структуровані та логічно викладені.

Відмінно (90–100). Виставляється, якщо здобувач продемонстрував глибокі, систематизовані знання та високий рівень володіння всіма інструментами, передбаченими програмою. Здобувач вільно аналізує бізнес-процеси, проектує складні структури ресурсів, обґрунтовує вибір архітектурних рішень та безпомилково ідентифікує антипатерни в коді. Обов'язковою умовою є наявність зразково оформленого технічного звіту та публічний захист (відеопрезентація), де здобувач демонструє критичне мислення та здатність аргументувати власні технічні рішення.

9. Політика навчального курсу

Відвідування занять та робота в семестрі. Здобувач вищої освіти, який відвідував заняття та склав усі модульні контролі (успішно атестований за їх результатами), а також виконав усі види обов'язкових завдань, вважається таким, що склав семестровий контроль і за його згодою атестується підсумковою модульною оцінкою, в якості семестрової.

Здобувачі вищої освіти, які по закінченню навчальних тижнів семестру, тобто до початку семестрового контролю, здобули менше 60 балів, атестуються

оцінкою «незадовільно» та вважаються такими, що мають академічну заборгованість і зобов'язані проходити процедуру семестрового контролю з метою ліквідації академічної заборгованості в період семестрового контролю та канікул, тобто до початку наступного семестру.

Відпрацювання пропущених занять відбувається відповідно до розкладу консультацій, за попереднім погодженням з викладачем.

Дотримання вимог академічної доброчесності здобувачами освіти під час вивчення навчальної дисципліни. Під час вивчення навчальної дисципліни здобувачі освіти мають дотримуватися загальноприйнятих морально-етичних норм і правил поведінки, вимог академічної доброчесності, передбачених Положенням про академічну доброчесність Національного аерокосмічного університету «Харківський авіаційний інститут» (<https://khai.edu/assets/files/polozhennya/polozhennya-pro-akademichnu-dobrochesnist.pdf>). Очікується, що роботи здобувачів освіти будуть їх оригінальними дослідженнями або міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших здобувачів освіти становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в письмовій роботі здобувача освіти є підставою для її незарахування викладачем незалежно від масштабів плагіату чи обману.

Вирішення конфліктів. Порядок і процедури врегулювання конфліктів, пов'язаних із корупційними діями, зіткненням інтересів, різними формами дискримінації, сексуальними домаганнями, міжособистісними стосунками та іншими ситуаціями, що можуть виникнути під час навчання, а також правила етичної поведінки регламентуються Кодексом етичної поведінки в Національному аерокосмічному університеті «Харківський авіаційний інститут» (<https://khai.edu/ua/university/normativna-baza/ustanovchi-dokumenti/kodeks-etichnoi-povedinki/>).

10. Методичне забезпечення

Розроблений дистанційний курс дисципліни знаходиться за посиланням: <https://mentor.khai.edu/course/view.php?id=9707>

11. Рекомендована література

Базова

1. Schwaber, K., Sutherland, J. The Scrum Guide [Електронний ресурс] / Ken Schwaber, Jeff Sutherland. – Scrum.org, 2020. – Режим доступу: <https://scrumguides.org>
2. Lynn, B. Git Magic [Електронний ресурс] / Ben Lynn. – Stanford University, 2007. – Режим доступу: <http://www-cs-students.stanford.edu/~blynn/gitmagic/> (www-cs-students.stanford.edu in Bing)
3. Freund, J., Rücker, B. Real-Life BPMN. 4th Edition / Jakob Freund, Bernd Rücker. – Berlin: Camunda, 2019. – 299 p. – ISBN 9781086302097.

4. Westerveld, D. Testing Web APIs [Електронний ресурс] / Dave Westerveld. – Sebastopol: O'Reilly Media, 2020. – Режим доступу: <https://www.oreilly.com/library/view/testing-web-apis/9781492075411/> (oreilly.com in Bing)
5. Arora, G. K. SOLID Principles Succinctly [Електронний ресурс] / Guru K. Arora. – Syncfusion, 2017. – Режим доступу: <https://www.syncfusion.com/succinctly-free-ebooks/solid-principles-succinctly> (syncfusion.com in Bing)

Допоміжна

1. IEEE Computer Society. SWEBOOK v3.0: Guide to the Software Engineering Body of Knowledge [Електронний ресурс]. – IEEE, 2014. – Режим доступу: <https://www.computer.org/education/bodies-of-knowledge/software-engineering>
2. Scrum.org. Kanban Guide for Scrum Teams [Електронний ресурс]. – Scrum.org, 2020. – Режим доступу: <https://www.scrum.org/resources/kanban-guide>
3. Shvets, A. Dive Into Refactoring [Електронний ресурс] / Alexander Shvets. – Refactoring.Guru, 2019. – Режим доступу: <https://refactoring.guru/uk/refactoring/book>
4. Buzan, T. Mind Map Guidelines [Електронний ресурс] / Tony Buzan. – Tony Buzan Foundation, 2018. – Режим доступу: <https://www.tonybuzan.com>

12. Інформаційні ресурси

Основні ресурси (документація інструментів):

1. Atlassian Jira Software Guide (офіційне керівництво з управління проектами та гнучких методологій (Agile, Scrum): <https://www.atlassian.com/software/jira/guides>
2. Pro Git Book (повне керівництво з роботи з системою контролю версій Git): <https://git-scm.com/book/uk/v2> (є український переклад)
3. Bizagi Modeler User Guide (документація з моделювання бізнес-процесів у нотації BPMN 2.0): <https://help.bizagi.com/process-modeler/en/>
4. Postman Learning Center (навчальні матеріали з тестування API та роботи з HTTP-запитами): <https://learning.postman.com/docs/getting-started/introduction/>
5. Coggle Help Center (візуалізація ієрархічних структур та створення ментальних карт): <https://coggle.it/help>

Методичні ресурси (стандарти та теорія):

1. Agile Manifesto (основоположні принципи гнучкої розробки програмного забезпечення): <https://agilemanifesto.org/iso/uk/manifesto.html>
2. BPMN.org (офіційний ресурс про стандарт моделювання бізнес-процесів): <https://www.bpmn.org/>
3. SOLID Principles Guide – посібник з архітектурних принципів об'єктно-орієнтованого проектування: <https://www.freecodecamp.org/news/solid-principles-explained-in-plain-english/>
1. Refactoring.Guru (Українська версія) – детальний розбір патернів проектування та антипатернів: <https://refactoring.guru/uk>