

**Чернівецький національний університет імені Юрія Федьковича**

(повне найменування закладу вищої освіти)

**Навчально-науковий інститут фізико-технічних та комп'ютерних наук**

(назва навчально-наукового інституту / факультету)

Кафедра комп'ютерних систем та мереж

**“ЗАТВЕРДЖУЮ”**

Директор навчально-наукового інституту  
фізико-технічних та комп'ютерних наук

\_\_\_\_\_ О. В. Ангельський

“ \_\_\_\_ ” \_\_\_\_\_ 2022 року

**РОБОЧА ПРОГРАМА  
навчальної дисципліни**

Інженерія програмного забезпечення

(назва навчальної дисципліни)

обов'язкова

(вказати: обов'язкова / вибіркова )

**Освітньо-професійна програма** Комп'ютерна інженерія,

Програмування мобільних і вбудованих комп'ютерних систем

та засобів Інтернету речей

(назва програми)

**Спеціальність** 123 Комп'ютерна інженерія

(вказати: код, назва)

**Галузь знань** 12 Інформаційні технології

(вказати: шифр, назва)

**Рівень вищої освіти** перший (бакалаврський)

(вказати: перший бакалаврський/другий магістерський)

**навчально-науковий інститут фізико-технічних та комп'ютерних наук**

(назва факультету/ навчально-наукового інституту,  
на якому здійснюється підготовка фахівців за вказаною освітньо-професійною програмою)

**Мова викладання** українська

Чернівці 2022 рік

Робоча програма навчальної дисципліни  
ОК28. Інженерія програмного забезпечення  
(назва навчальної дисципліни)

складена відповідно до освітньо-професійної програми

Комп'ютерна інженерія, Програмування мобільних і вбудованих  
комп'ютерних систем та засобів Інтернету речей, 123 Комп'ютерна інженерія,  
(назва освітньо-професійної програми, код та назва спеціальності)

12 Інформаційні технології, 30 червня 2022 р.  
(галузь знань: шифр та назва; дата останнього затвердження)

Розробники:

Танасюк Юлія Володимирівна, доцент кафедри КСМ, кандидат фіз.-мат. наук,  
Вацек Діана Орестівна, асистент кафедри КСМ,  
Одайська Христина Савеліївна, асистент кафедри КСМ, канд. техн. наук.

Затверджено на засіданні кафедри комп'ютерних систем та мереж

Протокол № 1 від “ 29 ” серпня 2022 року

Завідувач кафедри \_\_\_\_\_ (Воробець Г.І.)  
(підпис) (прізвище та ініціали)

Схвалено методичною радою навчально-наукового інституту  
фізико-технічних та комп'ютерних наук

Протокол № 1 від “ 31 ” серпня 2022 року

Голова методичної ради навчально-наукового інституту  
фізико-технічних та комп'ютерних наук  
\_\_\_\_\_ (Струк Я. М.)  
(підпис) (прізвище та ініціали)

## 1. Мета навчальної дисципліни

**Мета:** забезпечити засвоєння студентами основних понять і методів системного керування процесом розроблення програмного забезпечення, оволодіння базовими принципами програмної інженерії; формування умінь створення технічного завдання, планування виконання робіт по проекту, набуття навичок прийняття рішень та розподілу обов'язків у команді і використання сучасних інформаційних технологій при створенні професійного програмного забезпечення.

## 2. Результати навчання

У результаті вивчення навчальної дисципліни студенти повинні

### 2.1. Знати:

- основні поняття інженерії програмного забезпечення;
- підходи до управління процесом розробки програмного забезпечення;
- принципи архітектурного та об'єктно-орієнтованого проектування програмного забезпечення;
- основні типи інструментарію для розробки ПЗ;
- принципи та моделі розробки ПЗ, методології програмування;
- засоби управління вимогами до розробки ПЗ;
- основні методи забезпечення якості програмного забезпечення та тестування.

### 2.2. Вміти:

- формулювати вимоги до програмного продукту;
- розробляти проект специфікації та технічного завдання на розробку;
- розв'язувати задачі з використанням декомпозиції;
- створювати діаграми різних типів;
- розробляти структуру проекту ПЗ;
- проектувати та реалізовувати зручний користувацький інтерфейс;
- оформляти документацію до програмного проекту;
- працювати з кількома версіями програмного проекту;
- виконувати різного роду тестування ПЗ;
- визначати техніко-економічні показники програмного продукту;
- організовувати та підтримувати роботу в команді.

### 2.3. Набути компетентностей:

#### ЗК - загальних

ЗК1. Здатність до абстрактного мислення, аналізу і синтезу.

ЗК2. Здатність вчитися і оволодівати сучасними знаннями.

ЗК3. Здатність застосовувати знання у практичних ситуаціях.

ЗК4. Здатність спілкуватися державною мовою як усно, так і письмово.

ЗК6. Навички міжособистісної взаємодії.

ЗК7. Вміння виявляти, ставити та вирішувати проблеми.

ЗК8. Здатність працювати в команді.

ФК - фахових

ФК1. Здатність застосовувати законодавчу та нормативно-правову базу, а також державні та міжнародні вимоги, практики і стандарти з метою здійснення професійної діяльності в галузі комп'ютерної інженерії.

ФК2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.

ФК3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.

ФК5. Здатність використовувати засоби і системи автоматизації проектування до розроблення компонентів комп'ютерних систем та мереж, Інтернет додатків, кіберфізичних систем тощо.

ФК6. Здатність проектувати, впроваджувати та обслуговувати комп'ютерні системи та мережі різного виду та призначення.

ФК8. Готовність брати участь у роботах з впровадження комп'ютерних систем та мереж, введення їх до експлуатації на об'єктах різного призначення.

ФК9. Здатність системно адмініструвати, використовувати, адаптувати та експлуатувати наявні інформаційні технології та системи.

ФК11. Здатність оформляти отримані робочі результати у вигляді презентацій, науково-технічних звітів.

ФК12. Здатність ідентифікувати, класифікувати та описувати роботу програмно-технічних засобів, комп'ютерних та кіберфізичних систем, мереж та їхніх компонентів шляхом використання аналітичних методів і методів моделювання.

ФК13. Здатність вирішувати проблеми у галузі комп'ютерних та інформаційних технологій, визначати обмеження цих технологій.

ФК14. Здатність проектувати системи та їхні компоненти з урахуванням усіх аспектів їх життєвого циклу та поставленої задачі, включаючи створення, налаштування, експлуатацію, технічне обслуговування та утилізацію.

ФК15. Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення.

ФК16. Здатність застосовувати технології комп'ютерних систем і мереж, дискретної обробки інформації та числових методів для реалізації інформаційно-вимірювальних систем і систем передачі даних.

ПРН – програмних результатів навчання

ПРН2. Мати навички проведення експериментів, збирання даних та моделювання в комп'ютерних системах.

ПРН3. Знати новітні технології в галузі комп'ютерної інженерії.

ПРН4. Знати та розуміти вплив технічних рішень в суспільному, економічному, соціальному і екологічному контексті.

ПРН5. Мати знання основ економіки та управління проектами.

ПРН6. Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.

ПРН7. Вміти розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності.

ПРН8. Вміти системно мислити та застосовувати творчі здібності до формування нових ідей.

ПРН10. Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати типове для спеціальності обладнання.

ПРН11. Вміти здійснювати пошук інформації в різних джерелах для розв'язання задач комп'ютерної інженерії.

ПРН12. Вміти ефективно працювати як індивідуально, так і у складі команди.

ПРН13. Вміти ідентифікувати, класифікувати та описувати роботу комп'ютерних систем та їх компонентів.

ПРН14. Вміти поєднувати теорію і практику, а також ухвалювати рішення та виробляти стратегію діяльності для вирішення завдань спеціальності з урахуванням загальнолюдських цінностей, суспільних, державних та виробничих інтересів.

ПРН16. Вміти оцінювати отримані результати та аргументовано захищати прийняті рішення.

ПРН18. Використовувати інформаційні технології та для ефективного спілкування на професійному та соціальному рівнях.

ПРН19. Здатність адаптуватись до нових ситуацій, обґрунтовувати, ухвалювати та реалізовувати у межах компетенції рішення.

ПРН20. Усвідомлювати необхідність навчання впродовж усього життя з метою поглиблення набутих та здобуття нових фахових знань, удосконалення креативного мислення.

### 3. Опис навчальної дисципліни

#### 3.1. Загальна інформація

Назва навчальної дисципліни <i>OK28 Інженерія програмного забезпечення</i>												
Форма навчання	Рік підготовки	Семестр	Кількість			Кількість годин						Вид підсумкового контролю
			кредитів	годин	змістових модулів	лекції	практичні	семінарські	лабораторні	самостійна робота	індивідуальні завдання	
Денна	3	6	4	120	2	30	-	-	30	60	-	іспит
Заочна	3	6	4	120	2	8	-	-	8	104	-	іспит

Примітка. Співвідношення кількості годин аудиторних занять до самостійної роботи становить:

для денної форми навчання – 1.0 ((30+30)/60)

для заочної форми навчання – 0.15 ((8+8)/104)

### 3.2. Структура змісту навчальної дисципліни

Назви змістових модулів і тем	Кількість годин											
	Денна форма						Заочна форма					
	усього	у тому числі					усього	у тому числі				
		л	п	лаб	інд	с.р.		л	п	лаб	інд	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	13
<b>Змістовий модуль 1. Проектування і розробка програмного забезпечення</b>												
Тема 1. Життєвий цикл програмного забезпечення	18	6	-	4	-	8	22	1	-	1	-	20
Тема 2. Розробка архітектури ПЗ	26	6	-	8	-	12	32	1	-	1	-	30
Разом за змістовим модулем 1	44	12	-	12	-	20	54	2	-	2	-	50
<b>Змістовий модуль 2. Управління програмними проектами, методи забезпечення та контролю якості ПЗ</b>												
Тема 3. Моделювання програмного забезпечення	22	8	-	4	-	10	16	1	-	1	-	14
Тема 4. Управління програмними проектами	32	6	-	8	-	18	25	3	-	2	-	20
Тема 5. . Забезпечення та контроль якості ПЗ	22	4	-	6	-	12	35	2	-	3	-	30
Разом за змістовим модулем 2	76	18	-	18	-	40	76	6	-	6	-	64
<b>Усього годин</b>	<b>120</b>	<b>30</b>	<b>-</b>	<b>30</b>	<b>-</b>	<b>60</b>	<b>120</b>	<b>8</b>	<b>-</b>	<b>8</b>	<b>-</b>	<b>104</b>

### 3.3. Тематика лабораторних занять

№	Назва теми	Кількість годин
1.	Етапи розробки програм і програмної документації. Розробка технічного завдання.	4
2.	Розробка ескізного проекту ПЗ. Створення діаграми прецедентів	4
3.	Взаємодія об'єктів. Створення діаграми послідовності.	4
4.	Проектування користувацького інтерфейсу	4
5.	Створення діаграми класів	6
6.	Розрахунок техніко-економічних показників програмного продукту	4
7.	Тестування програмного продукту	4
	<b>Разом</b>	<b>30</b>

### 3.4. Самостійна робота студента

№	Назва теми	Кількість годин
1.	Моделі і методи розробки ПЗ	6
2.	Мови моделювання	6
3.	Шаблони проектування	8
4.	Методології моделювання ELM, OOAD	4
5.	Методи управління ресурсами та змістом програмного продукту	6
6.	Принципи постійної інтеграції ПЗ	4
7.	Верифікація і валідація програмного забезпечення	6
8.	Метрики програмного забезпечення	4
9.	Оптимізація коду та рефакторинг	4
10.	Аспекти продуктивності програмного забезпечення	6
11.	Надійність програмного забезпечення	6
<b>Разом</b>		<b>60</b>

### 3.5. Темы курсових робіт

№	Назва теми	№	Назва теми
1.	Органайзер	14.	Бібліотека
2.	Особисті дані студентів	15.	Автосервіс
3.	Облік виконання лабораторних робіт	16.	Служба доставки
4.	Облік успішності студентів	17.	Готель
5.	Екзаменаційна сесія	18.	Інтернет-магазин
6.	Телефонна служба	19.	Кабельне телебачення
7.	Облік порушень правил дорожнього руку	20.	Автовокзал
8.	Агентство нерухомості	21.	Туристична фірма
9.	Продаж залізничних квитків	22.	Ресторан
10.	Книжковий магазин	23.	Магазин побутової техніки
11.	Авіакаса	24.	Розважальний центр
12.	Поліклініка	25.	Спортивний клуб
13.	Кадрове агентство	26.	Комп'ютерна фірма

## 4. Форми і методи навчання

**Форми навчання** – це демонстраційні лекції оглядового та практичного характеру, лабораторні заняття, заняття із застосуванням комп'ютерної та телекомунікаційної техніки, інтерактивні заняття з навчанням одних студентів іншими, інтегровані заняття, заняття для колективного обговорення результатів та ухвалення рішень, відеоматеріали лекцій та відеоконференції засобами Google Meet, Cisco Webex, заняття з використанням системи електронного навчання Moodle.

**Методи:** проблемний виклад матеріалу, частково-пошукові та дослідницькі лабораторні практикуми, презентації, консультації та дискусії, робота в онлайн-середовищах: електронні лекції, лабораторні роботи, дистанційні консультації та ін., спрямовані на активізацію і стимулювання навчально-пізнавальної діяльності студентів.

**Підходи до навчання:** студенто-центрований, проблемно-орієнтований, діяльнісний, комунікативний, професійно-орієнтований і міждисциплінарний підходи.

**Реалізація навчального процесу** здійснюється під час лекційних, лабораторних занять, самостійної роботи з використанням сучасних інформаційних технологій навчання, консультацій з викладачами, інтерактивних засобів закріплення практичних навичок, онлайн-симуляторів і середовищ.

Для **формувань умінь та навичок** застосовуються такі **методи навчання**:

- вербальні/словесні (*лекція, пояснення, розповідь, бесіда, інструктаж*);
- наочні (*спостереження, ілюстрація, демонстрація*);
- практичні (*проведення експерименту, практики*);
- пояснювально-ілюстративний або інформаційно-рецептивний, який передбачає пред'явлення готової інформації викладачем та її засвоєння студентами;
- репродуктивний (*виконання лабораторних завдань за зразком*);
- метод проблемного викладу матеріалу на лекційних заняттях.

## 5. Критерії оцінювання результатів навчання з навчальної дисципліни

### Шкала оцінювання: національна та ЄКТС

Оцінка за шкалою ЄКТС	Пояснення	Оцінка за 100-бальною шкалою	Оцінка за національною шкалою
<b>A</b>	відмінно	<b>90 – 100</b>	<b>відмінно</b>
<b>B</b>	дуже добре	<b>80-89</b>	<b>добре</b>
<b>C</b>	добре	<b>70-79</b>	
<b>D</b>	задовільно	<b>60-69</b>	<b>задовільно</b>
<b>E</b>	достатньо	<b>50-59</b>	
<b>FX</b>	(незадовільно) з можливістю повторного складання	<b>35-49</b>	<b>незадовільно</b>
<b>F</b>	(незадовільно) з обов'язковим повторним вивченням курсу	<b>1-34</b>	



## 6. Засоби оцінювання

Засобами оцінювання та демонстрування результатів навчання є:

- завдання на лабораторні роботи;
- тестові завдання по темах;
- презентації результатів виконаних завдань;
- демонстрування поточних результатів виконання курсової роботи;
- практичні завдання по варіантах.

Формами поточного контролю рівня знань є усна та письмова відповідь студента при захисті виконаних лабораторних робіт, підготовка звітів до лабораторних робіт, відповіді на контрольні запитання, складання тестів у системі дистанційної освіти, виконання практичних завдань, а також письмова відповідь при написанні модульних контрольних робіт.

Формами підсумкового контролю рівня знань є усна та письмова відповідь студента при складанні іспиту, презентація результатів виконання курсової роботи, обґрунтування ухвалених рішень.

## 7. Форми поточного та підсумкового контролю

Формами поточного контролю рівня знань є усна та письмова відповідь студента при захисті виконаних лабораторних робіт, кількість отриманих балів при виконанні тестового завдання, а також письмова відповідь при написанні модульних контрольних робіт.

Поточне оцінювання (аудиторна і самостійна робота)							Підсумковий контроль (іспит)	Сума балів
Змістовий модуль 1			Змістовий модуль 2				40	100
T1	T2	M1	T3	T4	T5	M2		
10	10	10	5	5	5	15		

T1, T2, ... T5 – теми змістових модулів, M1, M2 – модульні контрольні роботи

### Розподіл балів за виконання курсової роботи

Пояснювальна записка	Проект розробленого ПЗ. Функціональна частина.	Захист роботи	Загальна сума балів
20 балів	40 балів	40 балів	100

## Змістовий модуль 1. Проектування і розробка програмного забезпечення

### T1. Життєвий цикл програмного забезпечення

1.1. Інженерія програмного забезпечення. Технології програмування в історичному аспекті.

1.2. Життєвий цикл (ЖЦ) програмного забезпечення. Моделі ЖЦ.

**Лабораторна робота №1.** Етапи розробки програм і програмної документації. Розробка технічного завдання (5 балів).

1.3. Методологія розробки програмного забезпечення. Гнучка розробка застосунків. Принципи Agile-розробки. Scrum, RAD. XP-програмування.

1.4. Управління вимогами до програмного забезпечення.

**Лабораторна робота № 2.** Розробка ескізного проекту ПП. Створення діаграми прецедентів (5 балів).

**T2. Розробка архітектури ПЗ**

2.1. Проектування архітектури ПЗ. Моделі структурування системи.

2.3. Моделювання управління і декомпозиція на модулі.

2.4. Проектування інтерфейсу користувача.

**Лабораторна робота № 3.** Взаємодія об'єктів. Створення діаграми послідовності (5 балів).

**Лабораторна робота № 4.** Проектування користувацького інтерфейсу (5 балів).

**M1. Модульна контрольна робота № 1 (10 балів).**

**Змістовий модуль 2. Управління програмними проектами, методи забезпечення та контролю якості ПЗ**

**T3. Моделювання програмного забезпечення**

3.1. Структурний підхід до моделювання. Методологія SADT.

3.2. Моделювання потоків даних.

3.3. Моделювання структур даних. Діаграма переходів станів.

3.4. Основи мови UML. Діаграми класів.

**Лабораторна робота № 5.** Створення діаграми класів (5 балів).

**T4. Управління програмними проектами**

4.1. Задачі управління проектами.

4.2. Концепції проекту. Управління ризиками програмного продукту.

4.3. Планування програмних проектів. SMART. WBS. PERT. CMP. Gant Chart.

4.3. Формування команди розробників. Розподіл ролей та обов'язків.

**Лабораторна робота № 6.** Розрахунок техніко-економічних показників програмного продукту (5 балів).

**T5. Забезпечення та контроль якості ПЗ**

5.1. Метрики та якість ПЗ.

5.2. Верифікація та тестування програмного забезпечення.

**Лабораторна робота № 7.** Тестування програмного продукту (5 балів).

**M2. Модульна контрольна робота № 2 (15 балів).**

**Підсумковий модуль-контроль – іспит (40 балів).**

### Питання для підсумкового контролю

1. Основні поняття та проблема розробки програмного забезпечення. Історія становлення принципів розробки ПЗ.
2. Життєвий цикл ПЗ: класичний (водоспадний), спіральний, Rational Objectory Process, Rapid Application Development (RAD), екстремальне програмування (XP).
3. Гнучка розробка застосунків- Agile. Методологія Scrum.
4. Аналіз, специфікація, верифікація та валідація вимог до ПЗ. Методи з'ясування вимог до ПЗ.
5. Функціональні та не функціональні вимоги до програмного забезпечення. Методи уточнення вимог до ПЗ.
6. Проектування архітектури ПЗ. Структурування системи: моделі репозиторію, клієнт/сервер, трирівнева, багатошарова (абстрактних машин).
7. Проектування архітектури ПЗ. Моделювання управління: моделі централізованого управління та на основі подій. Модель управління на основі подій: широкомовна та під управлінням переривань.
8. Проектування архітектури ПЗ. Модульна декомпозиція: об'єктно-орієнтована модель та модель потоків даних.
9. Принципи проектування інтерфейсу користувача.
10. Рекомендації щодо створення зручного та зрозумілого користувацького інтерфейсу.
11. Типи взаємодії з користувачем. Подання інформації. Оформлення інтерфейсів: кольори, шрифти, розміри об'єктів, розташування. Засоби підтримки користувача. Класифікація помилок. Оцінка інтерфейсів.
12. Методології моделювання. Класифікація моделей при розробці ПЗ.
13. Структурний підхід до проектування програмної системи. Методологія моделювання SADT. Побудова діаграми IDEF0.
14. Методологія моделювання потоків даних. Діаграма DFD.
15. Інформаційне моделювання. Діаграма сутність-зв'язок.
16. Діаграма STD – переходів станів.
17. Мова UML. Діаграми варіантів використання (прецедентів), послідовності, взаємодії.
18. Створення діаграми класів. Об'єкт. Клас. Атрибут. Операція. Реалізація принципів об'єктно-орієнтованого програмування. Стереотипи класів. Зв'язки між класами: узагальнення, асоціація, залежність, агрегація, композиція. Побудова діаграми класів.
19. Задачі управління проектами. Залізний трикутник обмежень.
20. Схема організації менеджменту проекту. Користувацькі і системні вимоги. Процес управління програмним проектом. Критерії успішного проекту.
21. Концепції проекту. Управління ризиками програмного продукту. Причини можливих ризиків План управління ризиками. Вплив ризиків на результат розробки.
22. Менеджмент проекту. Припущення та обмеження. Розподіл ресурсів. Строки виконання проекту. Планування графіку виконання проекту.

23. Організація роботи проектних команди. Ролі і зони відповідальності учасників команди розробників.
24. Якість ПЗ, Метрики і стандарти якості. Типи метрик ПЗ: метрики програмного продукту, процесу і використання.
25. Тестування ПЗ. Помилки ПЗ. Статичний аналіз програм. Забезпечення якості ПЗ. Тест-кейси. Класифікація підходів до тестування.
26. Техніко-економічні показники розробки програмного продукту: трудомісткість, чисельність виконавців, тривалість розробки, продуктивність. Визначення собівартості та вартості програмного забезпечення.

## **8. Рекомендована література**

### **8.1. Базова (основна)**

1. Інженерія програмного забезпечення: конспект лекцій / уклад. : Танасюк Ю В. Вацек Д. О. – Чернівці : Чернівецький національний університет імені Юрія Федьковича, 2022. 204 с. (електронне видання).
2. Інженерія програмного забезпечення: методичні вказівки до лабораторних робіт / уклад. : Танасюк Ю.В, Одайська Х. С. – Чернівці : Чернівецький національний університет імені Юрія Федьковича, 2022. 102 с. (електронне видання).
3. Farley D. Modern software engineering: doing what works to build better software faster. Addison-Wesley Professional, 2021. 256 p.
4. Мартін Р. Чиста архітектура. – Харків: Фабула, 2019. 368 с.
5. Мартін Р. Чистий Agile. Назад до основ. – Харків: Фабула, 2021. 224 с.
6. Ousterhout J. A philosophy of software design. Yaknyam Press, 2018. 190 p.
7. Thomas D. The pragmatic programmer: your journey to mastery, 20<sup>th</sup> anniversary edition. Addison-Wesley, 2019. 352 p.
8. Richards M. Fundamentals of software architecture: an engineering approach. O'Reilly Media, 2020. 419 p.
9. Mohan G. Full stack testing. A practical guide for delivering high quality software. O'Reilly Media, 2022. 406 p.
10. Філдінг Пол. Дж. Як керувати проектами. – Харків: Фабула, 2020. 240 с.
11. Сазерленд Дж. Scrum. Навчись робити вдвічі більше за менший час. – Харків: Клуб сімейного дозвілля, 2022. 280 с.

### **8.2. Допоміжна**

1. Левус Є.В. Вступ до інженерії програмного забезпечення: навч. посібник / Левус Є.В, Мельник Н.Б. – Львів: Видавництво Львівської політехніки, 2017. 280 с.
2. Sommerville I. Software Engineering Global Edition. Pearson Education, 2015. 624 p.
3. Sullivan S. Designing for wearables: effective UX for current and future devices, 1<sup>st</sup> edition. O'Reilly Media, 2017. 194 p.
4. Бок Л. Робота рулить! Уроки Google: правила гри у команді мрії. – К.: Наш формат, 2016. 424 с.

5. Lau E. The effective engineer: How to leverage your efforts in software engineering to make a disproportionate and meaningful impact. Palo Alto, CA, 2015. 260 p.
6. Fowler M. UML distilled: A brief guide to the standard object modeling language, 3<sup>rd</sup> edition. Addison-Wesley Professional, 2016. 208 p.
7. Booch G. Unified modeling language user guide. Addison-Wesley Professional, 2017. 504 p.
8. Rumpe B. Modeling with UML: language, concepts, methods, 1<sup>st</sup> edition. Springer, 2016. 295 p.
9. Myers G.J. The Art Of Software Testing - New York: John Wiley & Sons, Inc., 2004. 254 p.
10. Хігні Дж. Управління проектами. – Х.: Фабула, 2020. 272 с.
11. Когон К., Блейкморт С., Вуд Дж. Керування проектами для «неофіційних» проект-менеджерів. – Харків: Фабула, 2019. 240 с.
12. Петренко Н. Управління проектами : навч. посібник / Н. Петренко, Л. Кустрич, М. Гоменюк. – К. : Центр навчальної літератури, 2019. 244 с.
13. Лич Л. Вчасно і в рамках бюджету. Управління проектами за методом критичного ланцюга. – К.: Паблішер, 2018. 352 с.
14. A Guide to project management Body of Knowledge (PMBOK guide) and a Standard for project management, 7<sup>th</sup> edition, 2021. 250 p.
15. Авраменко А.С. Тестування програмного забезпечення: навч. посібник / Авраменко А.С., Авраменко В.С., Косенюк Г.В . – Черкаси : ЧНУ імені Богдана Хмельницького, 2017. 284 с.

### 9. Інформаційні ресурси

1. Agile manifesto: Understanding Agile values and Principles. URL : <https://www.softwaretestinghelp.com/agile-manifesto>
2. Scrum Guides. URL : <https://scrumguides.org/>
3. Як підвищити ефективність топ команди з урахуванням розподілу ролей. URL : <https://hurma.work/blog/yak-pidvishhiti-efektivnist-top-komandi-z-urahuvannyam-rozpodilu-rolej/>
4. Design patterns. URL : <https://refactoring.guru/design-patterns>
5. The clean architecture – Beginner’s guide. URL : <https://betterprogramming.pub/the-clean-architecture-beginners-guide-e4b7058c1165>
6. 5 способів мотивувати технічні команди. URL : <https://peopleforce.io/uk/blog/kak-motivirovat-tehnicheskuuy-komandu-5-sposobov-2/>
7. EQ: сам собі диригент у емоційних станів. URL : <https://hurma.work/blog/eq-sam-sobi-dirigent-emoczijnih-staniv/>
8. Гріневський Р. Система управління ризиками: кому і коли це потрібно. URL : <http://www.management.com.ua/qm/qm245.html>

9. Agile Software Development Metrics and KPIs that Help Optimize Product Delivery.  
URL : <https://www.altexsoft.com/blog/business/agile-software-development-metrics-and-kpis-that-help-optimize-product-delivery/>
10. Agile in a nutshell. URL : <http://www.agilenutshell.com/>