

Чернівецький національний університет імені Юрія Федьковича

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



РОБОЧА ПРОГРАМА
навчальної дисципліни

ТЕХНОЛОГІЯ СТВОРЕННЯ ПРОГРАМНИХ ПРОДУКТІВ
обов'язкова

Освітньо-професійна програма «Інформаційні системи та технології»

Спеціальність 126 Інформаційні системи та технології

Галузь знань 12 Інформаційні технології

Рівень вищої освіти перший бакалаврський

Мова навчання українська

Робоча програма навчальної дисципліни «ТЕХНОЛОГІЯ СТВОРЕННЯ ПРОГРАМНИХ ПРОДУКТІВ» складена відповідно до освітньо-професійної програми першого (бакалаврського) рівня вищої освіти «Інформаційні системи та технології» за спеціальністю 126 Інформаційні системи та технології галузі знань 12 Інформаційні технології, затвердженої Вченою радою Чернівецького національного університету імені Юрія Федьковича (Протокол № 7 від «31» серпня 2020 року).

Розробник: Томка Юрій Ярославович, доцент кафедри комп'ютерних наук, кандидат фізико-математичних наук

Погоджено з гарантом ОПП і затверджено на засіданні кафедри інформаційних технологій та комп'ютерної фізики

Протокол № 1 від “28” серпня 2023 року

Завідувачка кафедри ІТКФ  Борча М.Д.

Робоча програма затверджена на засіданні кафедри комп'ютерних наук

Протокол № 1 від “28” серпня 2023 року

Завідувач кафедри КН  Ушенко Ю.О.

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

Протокол № 1 від “29” серпня 2023 року

Голова методичної ради ННІФТКН  Струк Я.М.

1. Мета навчальної дисципліни

Дисципліна «Технологія створення програмних продуктів» є частиною освітньо-професійної програми «Інформаційні системи та технології» першого рівня вищої освіти за спеціальністю 126 Інформаційні системи та технології галузі знань 12 Інформаційні технології.

Метою викладання навчальної дисципліни "Технологія створення програмних продуктів" є формування системи теоретичних знань і набуття практичних умінь і навичок з питань теоретичних основ та використання сучасних методологій та технологій розроблення, тестування та експлуатації програмних продуктів, та для моделювання і проектування інформаційних систем, документування та маркетингу програмного забезпечення.

Завдання

Для досягнення мети поставлені такі основні завдання:

- вивчення понять, принципів, методологій та технологій створення програмних продуктів як сукупності процесів розроблення програмних систем на засадах життєвого циклу (ЖЦ) програмного забезпечення інформаційних систем;
- вивчення призначення засобів об'єктного підходу до проектування програмного забезпечення інформаційних систем;
- вивчення стандартів програмної інженерії;
- вивчення методів створення вимог під час розроблення програмних продуктів;
- вивчення методів та засобів тестування програмне забезпечення сучасними засобами тестування програмних продуктів;
- вивчення стандартів якості під час створення програмного забезпечення інформаційних систем;
- вивчення інструментальних засобів програмної інженерії та їх практичне застосування під час проектування програмне забезпечення інформаційних систем.

Об'єктом навчальної дисципліни є процеси, що відбивають різні аспекти створення та супроводу програмних продуктів.

Предметом навчальної дисципліни є технології розроблення, тестування та експлуатації програмних модулів інформаційних систем на основі сучасних методологій та засобів об'єктно-орієнтованого підходу для проектування інформаційних систем, вивчення методології щодо розроблення програмного забезпечення інформаційних систем, застосування мови Universal Modeling Language (UML) для моделювання і проектування інформаційних систем, застосування програмного інструментарію – IBM Rational Rose, вивчення основних положень з проектування інформаційних систем на базі сучасних технічних, програмних, інструментальних та комунікаційних засобів

2. Результати навчання

В результаті вивчення курсу студент повинен:

знати:

- вибирати стратегії для планування життєвого циклу системи;
- визначати організаційну, економічну, технічну та операційну
- здійсненність проекту;
- реалізовувати та тестувати компоненти програмного забезпечення;
- аналізувати вимоги замовника до програмних продуктів;

вміти:

- вибирати стратегії для планування життєвого циклу системи;
- визначати організаційну, економічну, технічну та операційну
- здійсненність проекту;

- реалізовувати та тестувати компоненти програмного забезпечення;
- аналізувати вимоги замовника до програмних продуктів;

Під час вивчення даної дисципліни студенти набудуть:

загальних компетентностей:

- КЗ 2. Здатність застосовувати знання у практичних ситуаціях;
- КЗ 5. Здатність вчитися і оволодівати сучасними знаннями;
- КЗ 6. Здатність до пошуку, оброблення та аналізу інформації з різних джерел;

спеціальних (фахових, предметних) компетентностей:

КС 3. Здатність до проектування, розробки, налагодження та вдосконалення системного, комунікаційного та програмноапаратного забезпечення інформаційних систем та технологій, Інтернету речей (IoT), комп'ютерно-інтегрованих систем та системної мережної структури, управління ними.

КС 4. Здатність проектувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші).

КС 8. Здатність управляти якістю продуктів і сервісів інформаційних систем та технологій протягом їх життєвого циклу.

КС 11. Здатність до аналізу, синтезу і оптимізації інформаційних систем та технологій з використанням математичних моделей і методів.

Програмні результати навчання:

ПРН 2. **Застосовувати** знання фундаментальних і природничих наук, системного аналізу та технологій моделювання, стандартних алгоритмів та дискретного аналізу при розв'язанні задач проектування і використання інформаційних систем та технологій.

ПРН 5. **Аргументувати** вибір програмних та технічних засобів для створення інформаційних систем та технологій на основі аналізу їх властивостей, призначення і технічних характеристик з урахуванням вимог до системи і експлуатаційних умов; мати навички налагодження та тестування програмних і технічних засобів інформаційних систем та технологій.

ПРН 7. **Обґрунтовувати** вибір технічної структури та **розробляти** відповідне програмне забезпечення, що входить до складу інформаційних систем та технологій.

3. Опис навчальної дисципліни

3.1. Загальна інформація

Назва навчальної дисципліни <u>Технологія створення програмних продуктів</u>												
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	3	26	-	-	39	55	-	екзамен

3.2. Структура змісту навчальної дисципліни

Назви змістових модулів і тем	Кількість годин					
	денна форма					
	усього	у тому числі				
л		п	лаб	інд	с.р.	
1	2	3	4	5	6	7
Теми лекційних занять	Змістовий модуль 1. Загальні питання та парадигми, моделі та методології створення ПЗ. Функціональні, нефункціональні вимоги.					
Тема 1. Вступ. Класифікація процесів розробки ПЗ. Стратегії розробки програмних засобів та реалізуючі їх моделі життєвого циклу	3	1	-	-	-	2
Тема 2. Методології розробки програмного забезпечення. Класичні та гнучкі методології. Основні стандарти інженерії програмного забезпечення.	12	2	-	6	-	4
Тема 3. Керівництво програмним проектом	4	2	-	-	-	2
Тема 4. Формування та аналіз вимог до програмного забезпечення	4	2	-	-	-	2
Разом за ЗМ 1	23	7	-	6	-	10
Теми лекційних занять	Змістовий модуль 2. Структурне проектування програмного забезпечення. Об'єктно-орієнтований аналіз та проектування програмного забезпечення. Діаграми UML. Контроль версій					
Тема 5. Структурне проектування програмних систем та засоби його автоматизації	4	1	-	-	-	3
Тема 6. Об'єктно-орієнтований аналіз та проектування програмного забезпечення. Мова UML	15	2	-	8	-	5
Тема 7. Метрики об'єктно-орієнтованих систем	4	1	-	-	-	3
Тема 8. Питання архітектури програмного забезпечення та стандарти опису архітектур	4	1	-	-	-	3
Тема 9. Стадія кодування. Системи контролю версій: локальні та розподілені. Основи GIT.	7	2	-	2	-	3
Разом за ЗМ2	34	7	-	10	-	17
Теми лекційних занять	Змістовий модуль 3. Якість програмного забезпечення. Основи тестування програмних додатків					
Тема 10. Управління якістю програмного забезпечення. Надійність та моделі оцінки надійності програмного забезпечення.	3	1	-	-	-	2
Тема 11. Основи тестування програмного забезпечення. Базові поняття.	4	1	-	-	-	3

Тема 12. Основи Unit-тестування	11	2	-	6	-	3
Тема 13. Основи інтеграційного тестування. Автоматизоване тестування.	11	2	-	6	-	3
Тема 14. Основи рефакторингу коду	4	1	-	-	-	3
Разом за ЗМ2	33	7	-	12	-	14
Теми лекційних занять	Змістовий модуль 4. Задачі контейнеризації, розгортання, супроводу ПЗ. Питання документування всіх етапів розробки ПЗ					
Тема 15. Задачі розгортання програмних додатків. Контейнеризація	7	1	-	2	-	4
Тема 16. Continuous integration/Continuous delivery/Continuous deployment. Azure DEVOPS	15	2	-	9	-	4
Тема 17. Супровід програмних продуктів.	4	1	-	-	-	3
Тема 18. Експлуатаційна, операційна, рекламна документація на ПЗ та маркетинг ПП	4	1	-	-	-	3
Разом за ЗМ3	30	5	-	11	-	14
ВСЬОГО	120	26	-	39	-	55

3.3. Самостійна робота

№	Назва теми
1	Тема 1. Вступ. Класифікація процесів розробки ПЗ. Стратегії розробки програмних засобів та реалізуючі їх моделі життєвого циклу Каскадна модель життєвого циклу. Каскадна модель ЖЦ ПЗ із зворотнім зв'язком. V-образна та пилообразна моделі ЖЦ ПЗ. Спиральна модель ЖЦ. Стандарт ISO/IEC 12207 "Процеси ЖЦ ПЗ". Архітектура процесів ЖЦ ПЗ. Процеси розробки ПЗ згідно стандарту ISO/IEC 12207.
2	Тема 2. Методології розробки програмного забезпечення. Класичні та гнучкі методології. Основні стандарти інженерії програмного забезпечення. RAD RUP SAFe SCRUM TSP UP XP
3	Тема 3. Керівництво програмним проектом Project Management Knowledge Areas The Project Life Cycle The Triple Constraint of Project Management The Project Management Phases: How to Manage a Project Step by Step Project Management Methodologies Project Management Tools Project Management Templates Project Management Roles Project Management Certifications

4	<p>Тема 4. Формування та аналіз вимог до програмного забезпечення</p> <p>Збирання та виявлення вимог. Аналіз вимог до ПЗ. Специфікація вимог. Стандарт IEEE 830 "Рекомендовані прийоми специфікації вимог до ПЗ". Структура стандарту. Вимоги замовника та вимоги розробника. Перевірка вимог. Онтологія домену та модель динамічних явищ домену.</p> <p>Об'єктно-орієнтована інженерія вимог. Метод Шлеєр і Меллора. Сценарний підхід до моделювання вимог (метод Джекобсона). Концепція методу UML щодо специфікації вимог до ПЗ. Продукти інженерії вимог.</p>
5	<p>Тема 5. Структурне проектування програмних систем та засоби його автоматизації</p> <p>Структурний аналіз та проектування ПС. Методологія SADT. Сімейство стандартів IDEF. Метод функціонального моделювання IDEF0 та його графічна нотація. Моделювання потоків даних за допомогою діаграм DFD. Нотації Йордана-ДеМарко та Гейна-Сарсона.</p> <p>Концептуальне проектування. Методологія опису бізнес-процесів IDEF3. Логічне моделювання даних згідно методології IDEF1X. Реалізація баз даних. CASE-засоби BPWin та ERWin – інструментальні засоби підтримки технологій моделювання. Додаткові можливості IDEF-моделей.</p>
6	<p>Тема 6. Об'єктно-орієнтований аналіз та проектування програмного забезпечення. Мова UML</p> <p>Об'єктно-орієнтоване проектування програмних продуктів. Методологія UML – потенційний стандарт ООП програмних систем. Мова UML. Діаграми варіантів використання. Діаграми класів. Діаграми взаємодії (діаграми послідовності та кооперації). Діаграми поведінки (діаграми станів та діяльності). Діаграми компонентів. Діаграма пакетів. Діаграма розгортання.</p> <p>Відмінності мови UML 2.0. Основні пакети метамоделі. Діаграма композитної структури. Додаткові діаграми структури (діаграми пакетів та об'єктів). Допоміжні діаграми взаємодії (діаграма комунікації, огляду взаємодії та часова діаграма). Діаграма скінченного автомата (State machine diagram)</p>
7	<p>Тема 7. Метрики об'єктно-орієнтованих систем</p> <p>Насиченість класу (англ. Weighted methods per class, WMC) Глибина спадкування (англ. Depth of Inheritance tree, DIT) Кількість нащадків (англ. Number of children, NOC) Зв'язаність класів(англ. Coupling between object classes, CBO) Відповідь класу (англ. Response for a class, RFC) Непов'язаність методів (англ. Lack of cohesion in Methods, LCOM)</p>
8	<p>Тема 8. Питання архітектури програмного забезпечення та стандарти опису архітектур</p> <p>Multitier architecture Model–view–controller</p>

	<p> Domain-driven design Blackboard pattern Sensor–controller–actuator Presentation–abstraction–control Component-based Monolithic application Layered Pipes and filters Database-centric Blackboard Rule-based Event-driven aka implicit invocation Publish-subscribe Asynchronous messaging Microkernel Reflection Client-server (multitier architecture exhibits this style) Shared nothing architecture Space-based architecture Object request broker Peer-to-peer Representational state transfer (REST) Service-oriented </p>
9	<p> Тема 9. Стадія кодування. Системи контролю версій: локальні та розподілені. Основи GIT. Основи Git Створення Git-репозиторія Запис змін до репозиторія Перегляд історії комітів Скасування речей Взаємодія з віддаленими сховищами Тегування Псевдоніми Git Галуження в git Гілки у кількох словах Основи галуження та зливання Управління гілками Процеси роботи з гілками Віддалені гілки Перебазування Git на сервері Протоколи Отримання Git на сервері Генерація вашого публічного ключа SSH Налаштування Серверу Демон Git </p>

	<p>Розумний HTTP GitWeb GitLab Варіанти стороннього хостингу Розподілений Git Розподілені процеси роботи Внесення змін до проекту Супроводжування проекту</p>
10	<p>Тема 10. Управління якістю програмного забезпечення. Надійність та моделі оцінки надійності програмного забезпечення. Software Failure Mechanisms Hardware vs. Software Reliability Software Reliability Measurement Techniques Reliability Metrics Software Fault Tolerance</p>
11	<p>Тема 11. Основи тестування програмного забезпечення. Базові поняття.</p>
12	<p>Тема 12. Основи Unit-тестування Understand the MSTest framework Understand the NUnit framework Understand the xUnit framework Decide between the frameworks</p> <p>Console Applications ASP.NET Core console project overview Set up the console testing project xUnit Theory with InlineData Challenge: Test add and average methods Solution: Test add and average methods xUnit Theory with MemberData xUnit Theory with ClassData</p> <p>Web API Applications ASP.NET Core web API project overview Set up web API testing project Unit testing HttpGet API endpoint Challenge: Test HttpGet API endpoint with id Solution: Test HttpGet API endpoint with id Unit testing HttpPost API endpoint Unit testing HttpDelete API endpoint</p> <p>MVC Applications ASP.NET Core MVC project overview Set up MVC testing project Unit testing index ActionResult Challenge: Unit testing details ActionResult Solution: Unit testing details ActionResult Unit testing create ActionResult Unit testing remove ActionResult</p>

13	<p>Тема 13. Основи інтеграційного тестування. Автоматизоване тестування.</p> <p>Selenium C# Tutorial: Setting Up Selenium In Visual Studio</p> <p>Running First Selenium C# Script With NUnit</p> <p>Selenium C# Tutorial: Using Implicit Wait in Selenium</p> <p>Selenium C# Tutorial: Using Explicit and Fluent Wait in Selenium</p> <p>Selenium C# Tutorial: Handling Alert Windows</p> <p>Selenium C# Tutorial: Handling Multiple Browser Windows</p> <p>Selenium C# Tutorial: Handling Frames & iFrames With Examples</p> <p>Selenium C#: Page Object Model Tutorial With Examples</p> <p>MSTest Tutorial: Environment Setup For Selenium Testing</p> <p>MSTest Tutorial: Running First Selenium Automation Script</p>
14	<p>Тема 14. Основи рефакторингу коду</p> <p>Роль патернів у задачах об'єктно-орієнтованого проектування та об'єктноорієнтованого аналізу</p> <p>Вступ до рефакторингу</p> <p>Запахи коду. Роздувальники:</p> <p>Запахи коду. Порушники об'єктного дизайну:</p> <p>Запахи коду. Забруднювачі коду:</p> <p>Запахи коду. Заплутувальники зв'язками:</p> <p>Техніки рефакторингу. Складання методів:</p> <p>Техніки рефакторингу. Переміщення функцій між об'єктами:</p> <p>Техніки рефакторингу. Організація даних:</p> <p>Техніки рефакторингу. Спрощення умовних виразів:</p> <p>Техніки рефакторингу. Спрощення викликів методів:</p> <p>Роль патернів у задачах рефакторингу</p>
15	<p>Тема 15. Задачі розгортання програмних додатків. Контейнеризація</p> <p>Basic Docker Commands</p> <p>Docker Run Commands</p> <p>Docker Images</p> <p>Environment Variables</p> <p>Command & Entrypoint</p> <p>Docker Compose</p> <p>Docker Storage</p> <p>Docker Networking</p>
16	<p>Тема 16. Continuous integration/Continuous delivery/Continuous deployment. Azure DEVOPS</p> <p>Agile Planning and Portfolio Management with Azure Boards Managing Project Schedules across Teams with Delivery Plans</p> <p>Version Controlling with Azure Repos</p> <p>Working with Pull Requests in VS Code and Azure DevOps</p> <p>Configuring pipelines as Code with YAML</p> <p>Enabling Continuous Integration with Azure Pipelines</p> <p>Embracing Continuous Delivery with Azure Pipelines</p> <p>Package Management with Azure Artifacts</p> <p>Collaborating with Azure DevOps Wiki</p> <p>Test Planning and Management with Azure Test Plans</p> <p>Exploratory Testing with Azure Test Plans</p> <p>Web Application Load and Performance Testing</p> <p>Monitoring App Performance with Application Insights</p>

	Working with Pull Requests in VS Code
17	Тема 17. Супровід програмних продуктів. Основні концепції (Basic Concepts), Процес супроводження (Process Maintenance), Ключові питання супроводу ПЗ (key Issue in Software Maintenance) , Техніки супроводу (Techniques for Maintenance).
18	Тема 18. Експлуатаційна, операційна, рекламна документація на ПЗ та маркетинг ПП Process documentation Planning documentation Estimate documentation Standards documentation Metrics documentation Scheduling documentation Product documentation System documentation Requirements documentation Architecture and design documentation Technical documentation User documentation End-user documentation System administrator documentation

4. Система контролю та оцінювання

Поточний контроль (ПК) здійснюється під час проведення лекційних, практичних та індивідуально-консультативних занять з метою перевірки рівня засвоєння теоретичних знань та практичних навичок студента. ПК проводиться у формі написання письмових робіт, проміжних тестувань та активності й влучності обговорення відповідних тем під час навчальних занять. Згідно з навчальним планом семестровий контроль з дисципліни «Технології створення програмних продуктів» відбувається у формі іспиту.

Відвідування занять. Відсутність на аудиторному занятті не передбачає нарахування штрафних балів, оскільки фінальний рейтинговий бал студента формується виключно на основі оцінювання результатів навчання. Разом з тим, обговорення результатів виконання тематичних завдань, а також презентація / публічний виступ та участь у обговореннях та доповнення на практичних заняттях оцінюватимуться під час аудиторних занять.

Оцінювання пропущених контрольних заходів. Кожен студент має право відпрацювати пропущені з поважної причини (лікарняний, мобільність тощо) заняття за рахунок самостійної роботи.

Процедура оскарження результатів контрольних заходів оцінювання. Студент може підняти будь-яке питання, яке стосується процедури контрольних заходів та очікувати, що воно буде розглянуто згідно із наперед визначеними процедурами Положенням про апеляцію на результати підсумкового семестрового контролю знань студентів Чернівецького національного університету імені Юрія Федьковича (<https://drive.google.com/file/d/16FPnHMJXd2al362HvDwmvoZ5uEih42ks/view>).

Академічна доброчесність. Політика та принципи академічної доброчесності визначені Етичним кодексом Чернівецького національного університету імені Юрія Федьковича (https://drive.google.com/file/d/1CB4AIMVXSAYkF_СepI-k98GPc9E8KznQ/view).

Інклюзивне навчання. Засвоєння знань та умінь в ході вивчення дисципліни «Технології створення програмних продуктів» може бути доступним для більшості осіб з особливими освітніми потребами, окрім здобувачів з серйозними вадами зору, які не дозволяють виконувати завдання за допомогою персональних комп'ютерів, ноутбуків та/або інших технічних засобів.

Навчання іноземною мовою. У ході виконання завдань студентам може бути рекомендовано звернутися до англомовних джерел.

Підготовка до практичних занять та контрольних заходів здійснюється під час самостійної роботи студентів з можливістю консультування з викладачем у визначений час консультацій або за допомогою електронного листування (електронна пошта, месенджери).

Неформальна освіта. При наявності сертифікатів про проходження професійноспрямованих психолого-педагогічних курсів, тренінгів, майстер-класів з неформальної освіти, участь у неформальній освіті студентам зараховувалось до 10% змістового матеріалу, що відповідає прослуханому матеріалу, за умови підготовки ними презентацій та нотаток за матеріалами прослуханого курсу, чи веб-заходу та їх публічного захисту на практичних заняттях.

Також, як можливості неформальної освіти студентам під час вивчення курсу «Технологія створення програмних продуктів» пропонується проходження курсів з отриманням сертифікатів, як індивідуальне завдання (ІНДЗ). На один модуль не більше 5 балів при 100% (46 – 80-99%, 36 – 60-79% тощо) проходженні курсів на віртуальних платформах (Prometheus, Coursera та інші).

Критерії оцінювання результатів навчання з навчальної дисципліни

Екзамен проводиться у формі, визначеній кафедрою. Екзаменаційний білет включає теоретичні і практичні завдання та містить 4 завдання (2 теоретичних питання, 2 практичні задачі). Результати екзамену оцінюються за національною чотирибальною шкалою. Максимальна кількість балів на екзамені за шкалою ВНЗ становить 40 балів.

Загальна підсумкова оцінка з дисципліни складається з суми балів за результатами ПМК та за виконання завдань, що виносяться на екзамен.

Якщо студент на екзамені отримав незадовільну оцінку, то це вважається як академічна заборгованість і набрані бали не заносяться до відомості. За графіком деканату студент перескладає екзамен і його результати заносяться до окремої відомості.

За результатами складання екзамену студент отримає:

27,0-30,0 бала – дана розгорнута вичерпна відповідь на теоретичні питання та правильно виконане практичне завдання;

24,0-26,0 бала – студентом допущені незначні помилки у відповіді на теоретичні питання чи практичне завдання;

21,0-23,0 бала – студент допустив значні помилки у відповіді на одне з теоретичних питань чи в практичному завданні;

18,0-20,0 бала – студент, допустивши значні помилки, не дав чіткої відповіді на теоретичні питання, не в повному обсязі виконав практичне завдання;

15,0-17,0 бала – студент не дав відповіді на одне із теоретичних питань, практичне завдання виконане не в повному обсязі;

10,5-14,0 бала – студент не дав відповіді на одне із теоретичних питань, не виконане практичне завдання.

0,1-10,3 бала – студент виконав частину одного з теоретичних питань, практичне завдання не виконане. 0 балів – студент не з'явився на екзамені.

Шкала оцінювання результатів екзамену

Підсумкова кількість балів за екзамен	Оцінка за національною шкалою	Оцінка за шкалою ECTS
27 - 30	5 (відмінно)	A (відмінно)
24 – 26	4 (добре)	B (дуже добре)
21 – 23		C (добре)
18 – 20	3 (задовільно)	D (задовільно)
15 – 17		E (достатньо)
10,5 – 14	2 (незадовільно) з можливістю повторного складання	FX
0 – 10,4	2 (незадовільно) з обов'язковим повторним курсом	F

Шкала оцінювання результатів навчальних досягнень при вивченні дисципліни

Кількість балів за 100 бальною шкалою (max- 100 балів)	Підсумкова оцінка за національною шкалою (max- 5 балів)	Підсумкова оцінка за шкалою ECTS
90-100	5 (відмінно)	A
80-89	4 (добре)	B
70-79	4 (добре)	C
60-69	3 (задовільно)	D
50-59	3 (задовільно)	E
35-49	2 (незадовільно (з можливістю складання іспиту)	FX
1-34	2 (незадовільно (з додатковим вивченням дисципліни)	F

Розподіл балів, які отримують студенти

Поточне оцінювання (аудиторна та самостійна робота)																ІНДЗ	Кількість балів (екзамен)	Загальна кількість балів		
Змістовий модуль № 1				Змістовий модуль № 2					Змістовий модуль № 4					Змістовий модуль № 4						
T 01	T 02	T 03	T 04	T 05	T 06	T 07	T 08	T 09	T 10	T 11	T 12	T 13	T 14	T 15	T 16	T 17	T 18	5	30	100
	12				15			5			9	9		6	8	6				
12				20					18					20						

*T1, T2 ... T15 – теми змістових модулів.

5. Рекомендована література

5.1. Базова (основна)

1. Tomka Yurii Theoretical and practical aspects of software development process. Monography. LAP LAMBERT Academic Publishing, 2023. 245p. ISBN: 978-620-5-64156-9.
2. Robert Martin. Agile Software Development, Principles, Patterns, and Practices / Robert Martin., 2002. – 552 p.
3. Rui Vilão. Software Development from A to Z: A Deep Dive Into All the Roles Involved in the Creation of Software / Rui Vilão, Olga Filipova., 2018. – 291 p.
4. Robert F Rose. Software Development Activity Cycles: Collaborative Development, Continuous Testing and User Acceptance / Robert F Rose., 2022. – 279 p.
5. Lynn Smith. Agile Software Development with C# and Visual Studio 2022 Third Edition / Lynn Smith., 2022. – 250 с.
6. Sarah Drasner. Engineering Management for the Rest of Us. Kindle Edition- Skill Recordings Inc, 2022 – 259p .
7. Петрик М.Р. Моделювання програмного забезпечення: науково-методичний посібник / М.Р. Петрик, О.Ю. Петрик – Тернопіль : Видво ТНТУ імені Івана Пулюя, 2015. – 200 с.
8. Jeffrey O.Grady. System Requirements Analysis, 2nd Edition. – Elsevier, 2013 – 834 p.
9. Гербер-Кронус Р. Оптимізація ПЗ. Збірник рецептів: підручник / Р. Гербер-Кронус, О. Бік, Д. Сміт. –, К.: Тіан. 2009. – 114с.
10. Табунщик Г. В. Проектування та моделювання програмного забезпечення сучасних інформаційних систем: навчальний посібник/ Г. В. Табунщик, Т.І. Каплієнко, О.А. Петрова – Запоріжжя : Дике Поле, 2016. – 250 с.
11. Грицюк Ю. І. Аналіз вимог до програмного забезпечення: навчальний посібник. / Ю. І. Грицюк – Львів : Видавництво Львівської політехніки, 2018. –456 с.
12. Кватрани Т. Візуальне моделювання за допомогою Rational Rose 2002 та UML. / Т. Кватрани – К.: Вільямс, 2003. – 192 с.

5.2. Допоміжна

1. Leszek Maciaszek. Requirements Analysis and Systems Design (3rd Edition) 3rd Edition. - Pearson Education Canada, 2007. – 656 p.
2. Jez Humble, David Farley. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (Addison-Wesley Signature Series (Fowler)) 1st Edition. - Addison-Wesley Professional, 2010 – 512p.
3. David Hay Requirements Analysis: From Business Views to Architecture 1st Edition. - Prentice Hall, 2002 – 496 p.
4. Andy Carmichael, Dan Haywood. Better Software Faster 1st Edition.- Prentice Hall 2002 - 384p.
5. Lisa Crispin, Janet Gregory. Agile Testing: A Practical Guide for Testers and Agile Teams 1st Edition. - Addison-Wesley Professional, 2008 – 576 p.
6. Alexandra Sunderland. Remote Engineering Management: Managing an Engineering Team in a Remote-First World. - Paperback, 2022 – 226 p.
7. Agile Processes in Software Engineering and Extreme Programming – Workshops: XP 2020 Workshops, Copenhagen, Denmark, June 8–12, 2020, Revised Selected ... Business Information Processing Book 396). - Springer; 1st ed. 2020 edition, Kindle Edition, (September 23, 2020) – 474 p.
8. Бабенко Л.П. Лавріщева К.М. Основи програмної інженерії: Навч. посіб. К.: Т-во «Знання», 2001. 269 с.

6. Інформаційні ресурси

1. The Unified Modeling Language. [Електронний ресурс] – Режим доступу: <https://www.uml-diagrams.org/>
2. ANDREJA VELIMIROVIC. What is SDLC? Understand the Software Development Life Cycle [Electronic resource] / ANDREJA VELIMIROVIC // PhoenixNAP'. – 2022. – Resource access mode: <https://phoenixnap.com/blog/software-development-life-cycle>.
3. Project Management [Electronic resource]. – 2022. – Resource access mode: <https://www.projectmanager.com/guides/project-management>.
4. Kirill Yusov. Software Requirements Specification Example and Guide [Electronic resource] / Kirill Yusov. – 2020. – Resource access mode: <https://jelvix.com/blog/software-requirements-specification>.
5. Khalil Stemmler. How to Learn Software Design and Architecture - a Roadmap [Electronic resource] / Khalil Stemmler. – 2019 – Resource access mode: <https://www.freecodecamp.org/news/software-design/>.
6. John Ousterhout. A Philosophy of Software Design, 2nd Edition / John Ousterhout., 2021. – 196 с.
7. Hiren Dhaduk. 10 Best Software Architecture Patterns You Must Know About [Electronic resource] / Hiren Dhaduk. – 2020. – Resource access mode: <https://www.simform.com/blog/software-architecture-patterns/>.
8. Maitray Gadhavi. The 80/20 Way of Comparing Software Architecture Patterns and Choosing the Best One [Electronic resource] / Maitray Gadhavi. – 2022 – Resource access mode: <https://radixweb.com/blog/software-architecture-patterns#layered>.
9. Joel Falconer. How to Write Good Code: 10 Beginner-friendly Techniques for Instant Results [Electronic resource] / Joel Falconer. – 2022. – Resource access mode: <https://www.sitepoint.com/how-to-write-good-code/>.
10. Burak Guzel. Top 18 Best Practices for Writing Super Readable Code [Electronic resource] / Burak Guzel . – 2021. – Resource access mode: <https://code.tutsplus.com/tutorials/top-15-best-practices-for-writing-super-readable-code-net-8118>.
11. Software Reliability [Electronic resource]. – 2022. – Resource access mode: <https://www.javatpoint.com/software-engineering-software-reliability>.
12. Reliability Metrics [Electronic resource]. – 2022. – Resource access mode: <https://www.javatpoint.com/software-engineering-software-reliability-metrics>.
13. What Is Software Deployment? [Electronic resource]. – 2022. – Resource access mode: <https://www.pagerduty.com/resources/learn/what-is-software-deployment/>.
14. Samarpit. CI/CD Pipeline – Learn how to Setup a CI/CD Pipeline from Scratch [Electronic resource] / Samarpit. – 2022. – Resource access mode: <https://www.edureka.co/blog/ci-cd-pipeline/>.
15. GitHub CI/CD Tutorial [Electronic resource]. – 2023. – Resource access mode: <https://mindmajix.com/github-ci-cd-tutorial>.
16. Quality Assurance, Quality Control and Testing — the Basics of Software Quality Management [Electronic resource] // AltexSoft. – 2016. – Resource access mode: <https://www.altexsoft.com/whitepapers/quality-assurance-quality-control-and-testing-the-basics-of-software-quality-management/>.
17. Matthew Jones. Unit Testing Overview with ASP.NET 5.0, XUnit, and Moq [Electronic resource] / Matthew Jones. – 2021. – Resource access mode: <https://exceptionnotfound.net/unit-testing-overview-with-asp-net-core-xunit-and-moq/>
18. Getting Started with xUnit.net [Electronic resource]. – 2022. – Resource access mode: <https://xunit.net/docs/getting-started/netfx/visual-studio>.
19. Unit Test with .Net 6 with xUnit and MOQ [Electronic resource]. – 2022. – Resource access mode: <https://dev.to/moe23/learn-unit-test-with-net-6-with-xunit-and-moq-k9i>.

20. Software Maintenance [Electronic resource] // Javatpoint – Resource access mode: <https://www.javatpoint.com/software-engineering-software-maintenance>.
21. Software Maintenance Overview [Electronic resource] – Resource access mode: https://www.tutorialspoint.com/software_engineering/software_maintenance_overview.htm.
22. Software Maintenance Cost Factors [Electronic resource] // codestudio. – 2022. – Resource access mode: <https://www.codingninjas.com/codestudio/library/softwaremaintenance-cost-factors>.
23. Technical Documentation in Software Development: Types, Best Practices, and Tools [Electronic resource] // altexsoft. – 2020. – Resource access mode: <https://www.altexsoft.com/blog/business/technical-documentation-in-softwaredevelopment-types-best-practices-and-tools/>.