

СИЛАБУС

Базова інформація про дисципліну	
Назва дисципліни	СЕ017 Командна розробка програмних проєктів / Software Projects Team Development
Рівень фахової передвищої освіти	Фаховий молодший бакалавр
Галузь знань	12 Інформаційні технології
Спеціальність	121 «Інженерія програмного забезпечення»
Освітня програма	Інженерія програмного забезпечення
Семестр	5 семестр (на базі повної загальної середньої освіти) 7 семестр (на базі базової середньої освіти)
Курс	3 курс (на базі повної загальної середньої освіти) 4 курс (на базі базової середньої освіти)
Анотація курсу	Предмет передбачає ісципл вивчення принципів організації спільної роботи над програмними продуктами, розподілу ролей у команді, планування та контролю виконання завдань. Розглядаються етапи життєвого циклу програмного забезпечення, методології Agile, Scrum, Kanban, а також сучасні інструменти керування проєктами та системи контролю версій (Git, GitHub, GitLab). У процесі навчання студенти набувають умінь ефективно працювати в команді розробників, використовувати інструменти спільної роботи, тестувати та документувати результати. Результатом вивчення курсу є створення колективного програмного продукту з урахуванням вимог якості та сучасних стандартів розробки.
Сторінка курсу в MOODLE	http://78.137.2.119:2929/course/view.php?id=824
Мова викладання	українська
Лектор курсу	Захарова Марія В'ячеславівна, к.т.н., доцент Канали комунікації: E-mail: lecturer2020student@gmail.com Чат на сторінці дисципліни у системі Moodle
Місце дисципліни в освітній програмі	
Освітньо-професійна програма	http://csbc.edu.ua/documents/otdel/koop_pr23.pdf
Інтегральна компетентність	Здатність вирішувати типові спеціалізовані задачі інженерії програмного забезпечення, що вимагає застосування положень і методів відповідних наук (математики, інформатики, інформаційних технологій, тощо) та може характеризуватися певною невизначеністю умов; нести відповідальність за результати своєї діяльності; здійснювати контроль інших осіб у визначених ситуаціях.

Перелік загальних компетентностей (ЗК)	ЗК05. Знання та розуміння предметної області та розуміння професійної діяльності. ЗК06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел. ЗК07. Здатність застосовувати знання у практичних ситуаціях.
Перелік спеціальних компетентностей (СК)	СК01. Здатність алгоритмічно та логічно мислити. СК02. Здатність вдосконалювати знання і навички в галузі інформаційних технологій та усвідомлення важливості навчання протягом усього життя. СК04. Здатність дотримуватися стандартів при розробці програмного забезпечення. СК07. Здатність розробляти модулі і компоненти програмного забезпечення за допомогою типових алгоритмів та інструментів.
Перелік програмних результатів навчання	РН03. Застосовувати спеціалізовані емпіричні та теоретичні знання у сфері інженерії програмного забезпечення. РН05. Розробляти та супроводжувати програмне забезпечення. РН08. Аналізувати вимоги до програмного забезпечення. РН09. Розуміти основні принципи командної роботи при розробці програмного забезпечення. РН15. Аналізувати та узагальнювати необхідну інформацію з різних джерел та ресурсів для розв'язання професійних задач з урахуванням сучасних досягнень.
Опис дисципліни	
Структура навантаження на студента	Загальна кількість годин – 180 Кількість кредитів – 6 Кількість лекційних годин – 54 Кількість практичних занять – 54 Кількість годин для самостійної роботи студентів – 72 Форма підсумкового контролю – іспит
Методи навчання	Словесні (інформаційна, самостійна робота з джерелами інформації, науково-популярна розповідь); Наочні (презентаційні повідомлення); Практичні роботи; Інтерактивні методи (дистанційні консультації); Індивідуальні завдання
Зміст дисципліни	
Змістовий модуль 1. Основи командної розробки програмних проєктів.	

Тема 1. Вступ до командної розробки програмного забезпечення.	Поняття командної роботи. Ролі учасників команди розробників. Етапи життєвого циклу програмного забезпечення. Основи ефективної комунікації та взаємодії в команді.
Тема 2. Організація процесу створення програмного проєкту.	Планування роботи команди, постановка цілей, визначення ролей і завдань. Принципи управління часом, ресурсами та ризиками. Документування етапів розробки.
Тема 3. Методології управління командними проєктами.	Основи Waterfall, Agile, Scrum, Kanban. Планування спринтів, розподіл завдань, проведення стендапів і ретроспектив. Порівняння підходів до організації роботи команди.
Змістовий модуль 2. Інструменти та практична реалізація командних проєктів.	
Тема 4. Інструменти командної взаємодії та управління завданнями.	Планування спринтів, розподіл завдань, проведення стендапів і ретроспектив. Порівняння підходів до організації роботи команди.
Тема 5. Системи контролю версій у командній розробці.	Основи роботи з Git. Репозиторії, гілки, коміти, пул-запити. Використання GitHub, GitLab, Bitbucket для спільної розробки. Розв'язання конфліктів версій, організація командного внеску.
Тема 6. Забезпечення якості командних програмних проєктів.	Тестування програмного забезпечення: модульне, інтеграційне, системне. Автоматизація тестування. Інструменти CI/CD для перевірки якості коду (GitHub Actions, GitLab CI/CD).
Тема 7. Аналіз ефективності командної взаємодії	Представлення створеного програмного продукту. Оформлення технічної, користувацької та презентаційної документації, публікація результатів у репозиторії команди.
Політика дисципліни	
Політика відвідування	Регулярне відвідування всіх видів занять, своєчасність виконання самостійної роботи. За об'єктивних причин студент може отримати індивідуальний графік навчання за погодженням із керівником курсу, завідувачем ЦК та завідувачем відділення.
Політика щодо дедлайнів та перескладання	Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку.
Академічна доброчесність	У випадку недотримання політики академічної доброчесності (плагіат, самоплагіат, фабрикація, фальсифікація, списування, обман, хабарництво) передбачено повторне проходження оцінювання.
Використання ШІ	Використання ШІ під час виконання завдань регламентується Політикою «Використання ШІ в

	освітньому процесі ЧДБК» Завдання мають маркування регламенту використання ІІІ.
Підсумковий контроль	іспит

Система оцінювання

Система оцінювання підсумкової успішності студентів поділяється на **поточний контроль** та **семестровий контроль**.

Поточний контроль здійснюється протягом семестру і охоплює всі види аудиторної роботи (практичні, семінарські заняття) та виконання індивідуальних завдань. Максимальна кількість балів, яку студент може набрати за цей вид контролю, становить 100.

Підсумковий контроль відбувається у формі іспиту.

Розрахунок підсумкової оцінки.

Підсумкова оцінка (О) розраховується як сума балів за роботу в семестрі (S) та балів за іспит (Т), кожен з яких має максимальну вагу 100 балів. Вагові коефіцієнти для обох компонентів однакові та дорівнюють 0,5.

Формула: $O = (S \times 0.5) + (T \times 0.5)$

Види навчальної роботи	Загальна кількість балів
Практичні роботи №1-5 (по 5 балів)	25
Тести №1-4 (по 10 балів)	40
Модульні контрольні роботи (№1-10 балів, №2-15 балів)	25
Індивідуальне завдання	10
Всього (поточний контроль)	100
Іспит	100

Критерії оцінювання для кожного виду навчальної роботи:

Критерії оцінювання практичних робіт:

5 б.- повне вірне виконання поставленої задачі, з відповідним оформленням, яке здане у встановлені викладачем терміни.

4 б. - наявність незначних помилок при оформленні роботи, або при вирішенні поставленої задачі, або робота здана у терміни, пізніші відведених.

3 б.- наявність незначних помилок при оформленні роботи і при вирішенні поставленої задачі а також робота здана у терміни, пізніші відведених.

2 б. - 50% завдань виконані з помилками, але робота здана у відповідному оформленні та у встановлені викладачем терміни.

1 б. - 50% завдань виконані з помилками,робота не оформлена,або частково оформлена

0 б. - відсутність зданої на оцінювання роботи

Критерії оцінювання модульних робіт

10 б. - студент демонструє повне розуміння теоретичного матеріалу і вміє його використовувати при виконанні практичних завдань, а також вміє пояснити алгоритм рішення кожного завдання. Всі поставлені задачі виконані у повному обсязі

9 б. - студент демонструє повне розуміння теоретичного матеріалу і вміє його використовувати при виконанні практичних завдань, а також вміє пояснити алгоритм рішення кожного завдання.90 % поставлених задач виконані у повному обсязі

8 б. - студент демонструє достатнє розуміння теоретичного матеріалу і вміє його використовувати при виконанні практичних завдань, але допускає помилки при поясненні.

- 7 б. - студент демонструє задовільне розуміння теоретичного матеріалу і вміє його використовувати епізодично при виконанні деяких практичних завдань, та припускається грубих помилок при поясненні.
- 6 б. - студент вирішив всі практичні завдання, але не може дати теоретичного обґрунтування
- 5 б. - студент вирішив деякі завдання, але не може дати теоретичного обґрунтування
- 4 б. - студент вирішив 50% завдання, але не може дати теоретичного обґрунтування
- 3 б. - студент вирішив одне практичне завдання, та може дати теоретичне обґрунтування
- 2 б. - студент вирішив одне практичне завдання, але не може дати теоретичного обґрунтування
- 1 б. - студент може знайти відповідь на завдання з допомогою ІІІ в присутності викладача
- 0 б. - студент не виконав жодного завдання

Критерії оцінювання індивідуального завдання

- 10 б. - студент демонструє повне розуміння теоретичного матеріалу і вміє його використовувати при виконанні проєкту, а також вміє пояснити алгоритм виконання проєкту. Постановка задачі виконана у повному обсязі
- 9 б. - студент демонструє розуміння теоретичного матеріалу і вміє його використовувати при виконанні проєкту, а також вміє пояснити алгоритм рішення, 90 % поставлених задач виконані у повному обсязі
- 8 б. - студент демонструє достатнє розуміння теоретичного матеріалу і вміє його використовувати при виконанні проєкту, але допускає помилки при поясненні.
- 7 б. - студент демонструє задовільне розуміння теоретичного матеріалу і вміє його використовувати епізодично при виконанні деяких частин проєкту, та припускається грубих помилок при поясненні.
- 6 б. - студент виконав 70% проєкту, але не може дати теоретичного обґрунтування
- 5 б. - студент вирішив частково завдання, але не може дати теоретичного обґрунтування
- 4 б. - студент вирішив 50% завдання, але не може дати теоретичного обґрунтування
- 3 б. - студент орієнтується у теорії, але не може її використати для розробки проєкту.
- 2 б. - студент може словесно розказати алгоритм виконання проєкту, але не може його відтворити програмними засобами
- 1 б. - студент може знайти відповідь на завдання з допомогою ІІІ в присутності викладача
- 0 б. - студент не виконав проєкт

Критерії оцінювання іспиту

За іспит з предмету можна набрати 100 балів максимально. Білет складається з 4 завдань, двох теоретичних та двох практичних. За виконання першого завдання можна отримати максимально 20 балів, за друге завдання – 20 балів, за виконані практичні завдання – по 30 балів.

Таблиця 1. Бали за виконання першого теоретичного завдання

<i>Бали</i>	<i>Критерії</i>
18-20	Студент (студентка) правильно, повно, чітко і логічно відповідає на в поставлені питання. При незначній неповноті бал знижується.
9-17	Правильно і повно, інколи з деякою неточністю та з допомогою навідні (пояснювальних) питань відповідає на всі поставлені питання
1-8	Відповідь студента (студентки) неповна і неточна, на навідні питання відповідає не зовсім точно, відповідь не є чіткою і логічною
0	Не може відповісти на питання взагалі

Таблиця 2. Бали за виконання другого теоретичного завдання

Завдання має 10 питань. Кожне питання оцінюється у 2 бали.

Бали	Критерії
2	Студент (студентка) правильно, повно, чітко відповідає на і питання
1	Студент (студентка) правильно відповідає на частину питання.
0	відповідь невірна
Таблиця 3. Бали за виконання кожного практичного завдання	
Бали	Критерії
25-30	Студент самостійно деталізує вхідні дані до завдання, аналізує алгоритм виконання завдання та особливості реалізації. Програмна реалізація повністю відповідає поставленому завданню, і студент здатний вносити в неї невеликі корективи.
15-24	Студент самостійно деталізує вхідні дані до завдання, але не може обґрунтувати їх доцільність, аналізує алгоритм виконання завдання. Програмна реалізація переважно відповідає поставленому завданню.
8-14	Студент (студентка) знає алгоритм виконання, потрібні синтаксичні інструменти. Програмна реалізація частково відповідає поставленому завданню.
1-7	Студент (студентка) знає алгоритм виконання, потрібні синтаксичні інструменти, проте програмну реалізацію здійснити не може.
0	Завдання не виконано взагалі

Шкала оцінювання		
ECTS	Бали	Зміст
A	90-100	Бездоганна підготовка в широкому контексті
B	80-89	Повні знання, міцні вміння
C	70-79	Хороші знання та вміння
D	65-69	Задовільні знання, стереотипні вміння
E	60-64	Виконання мінімальних вимог діяльності в стандартних умовах
FX	35-59	Слабкі знання, відсутність умінь
F	1-34	Необхідний повторний курс

Список рекомендованих джерел

1. Командна розробка ІТ-проектів. Лабораторний практикум для студентів денної та заочної форм навчання освітньої програми «Інженерія програмного забезпечення Інтернету речей» першого (бакалаврського) рівня вищої освіти за спеціальністю 121 Інженерія програмного забезпечення галузі знань 12 Інформаційні технології / уклад. О. В. Шпортько, Ю. Г. Лотюк, О. М. Богут. Рівне: ПВНЗ "МЕРУ ім. акад. С. Дем'янчука", 2023. 65 с.
2. Гагаріна Л.Г. Введення до архітектури програмного забезпечення: навчальний

посібник / Л.Г. Гагаріна, А.Р. Федоров, П.А. Федоров. – Москва: ІНФРА-М, 2018. 320 с.

3. Axelrod A. Complete Guide to Test Automation: Techniques, Practices, and Patterns for Building and Maintaining Effective Software Projects / A. Axelrod. – NY: Apress, 2018. 588 p.
4. Richards M. Fundamentals of Software Architecture: An Engineering Approach / M. Richards, N. Ford. – Sebastopol: O'Reilly Media, 2020. 432 p.

Інтернет ресурси

1. Цибульник С.О., Барандич К.С. ТЕХНОЛОГІЇ РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЧАСТИНА 1. ЖИТТЄВИЙ ЦИКЛ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ. Електронне мережне навчальне видання <https://ela.kpi.ua/server/api/core/bitstreams/9521e5f9-421a-4874-a17d-f0853f942856/content>
2. Introduction to Git workflows. URL: https://docs.gitlab.com/ee/topics/gitlab_flow.html.
3. Gitflow Workflow. Atlassian Git Tutorial. URL: <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>.