МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Львівський національний університет імені Івана Франка
Факультет електроніки та комп'ютерних технологій
Кафедра оптоелектроніки та інформаційних технологій

Затверджено
на засіданні кафедри оптоелектроніки та інформаційних технологій факультету електроніки та комп'ютерних технологій Львівського національного університету імені Івана Франка (протокол № 7 від 27.08.2026 р.)

В. о. завідувача кафедри _____ Юрій КОРЧАК

Силабус з навчальної дисципліни
"Життєвий цикл ПЗ",
що викладається в межах ОПП
" Інженерія програмного забезпечення "
першого (бакалаврського) рівня вищої освіти для здобувачів з спеціальності 121 – Інженерія програмного забезпечення

Львів 2026 р.

Назва дисципліни	Життєвий цикл ПЗ
Адреса викладання дисципліни	Корпус факультету електроніки та комп'ютерних технологій, Львівський національний університет імені Івана Франка, вул. Драгоманова 50, м. Львів, 79005, вул. Ген. Тарнавського 107, м. Львів, 79011
Факультет та кафедра, за якою закріплена дисципліна	Факультет електроніки та комп'ютерних технологій Кафедра оптоелектроніки та інформаційних технологій
Галузь знань, шифр та назва спеціальності	12 – інформаційні технології 121 – Інженерія програмного забезпечення
Викладачі дисципліни	Франів В.А., кандидат фізико-математичних наук, доцент кафедри оптоелектроніки та інформаційних технологій
Контактна інформація викладачів	volodymyr.franiv@lnu.edu
Консультації з питань навчання по дисципліні відбуваються	Консультації в день проведення лекцій/лабораторних занять (за попередньою домовленістю): ауд. 214, корпус факультету електроніки та комп'ютерних технологій, вул. Ген. Тарнавського 107, м. Львів, 79011
Сторінка курсу	https://electronics.lnu.edu.ua/course/zhyttievyy-tsykl-pz-ipz/
Коротка анотація дисципліни	Дисципліна "Життєвий цикл ПЗ" є нормативною дисципліною з спеціальності 121 – Інженерія програмного забезпечення для освітньої програми "Інженерія програмного забезпечення", яка викладається в 8-му семестрі в обсязі 4-ох кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Інформація про дисципліну	Дисципліна "Життєвий цикл ПЗ" є нормативною дисципліною з спеціальності 121 – Інженерія програмного забезпечення для освітньої програми "Інженерія програмного забезпечення", яка викладається в 4-му семестрі в обсязі 4-ох кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Мета та цілі дисципліни	*Мета:* Надати студентам глибоке розуміння ключових концепцій і методів, пов'язаних із життєвим циклом програмного забезпечення в інформаційних технологіях. Мета полягає у формуванні навичок ефективного планування, виконання та моніторингу проектів, а також в оволодінні стратегіями розробки програмного забезпечення для досягнення високої якості та успішного впровадження в сучасному IT-середовищі. *Цілі:* Забезпечення студентів глибоким розумінням теорії та практики життєвого циклу програмного забезпечення. Розвиток навичок практичного використання концепцій та методів у реальних проектах. Навчання студентів вибору та обґрунтування підходів до розробки програмного забезпечення. Розробка у студентів навичок управління IT-проектами, включаючи планування та контроль. Підготовка до вибору та підтримки розроблених продуктів в умовах реального використання.

	Стимулювання творчого та інноваційного підходу в розробці програмного забезпечення. Ці цілі спрямовані на комплексне формування у студентів навичок та знань для ефективного управління та розробки програмного забезпечення в інформаційних технологіях.
Література для вивчення дисципліни	1) Навчальний посібник з дисципліни «Технології розробки програмного забезпечення» для студентів спеціальності 123 «Комп'ютерна інженерія». – Полтава: ПолтНТУ, 2017. – 218 с. 2) Comparison usage of agile, waterfall and other approaches for software development under unstable situations caused by russian invasion of Ukraine / Franiv V.A., Vasylyk S.V., Franiv I.A. // Electroniks and Information Technologies 2023. V. 22 pp. 70-82 DOI: http://dx.doi.org/10.30970/eli.23.7 3) Pressman, Roger S. Software Engineering: A Practitioner's Approach. McGraw-Hill Education, 2014. 4) Ian Sommerville. Software Engineering (10th Edition). Pearson, 2015. 5) Martin Fowler. Refactoring: Improving the Design of Existing Code (2nd Edition). Addison-Wesley Professional, 2018. 6) Robert Cecil Martin. Clean Architecture: A Craftsman's Guide to Software Structure and Design. Prentice Hall, 2017. 7) Andrew Hunt, David Thomas. The Pragmatic Programmer: Your Journey to Mastery (20th Anniversary Edition). Addison-Wesley Professional, 2019.
Обсяг курсу	Загальний обсяг: 120 годин. Аудиторних занять: 64 год., з них 32 год. лекційних та 32 годин лабораторних робіт. Самостійної роботи: 56 год.
Очікувані результати навчання	У результаті вивчення даного курсу студент буде: **знати**: основні поняття, визначення і проблеми курсу; вимоги до постановки основних задач життєвого циклу програмного забезпечення; призначення й особливості застосування різних життєвих циклів програмного забезпечення; **вміти**: володіти основними інструментами та підходами для створення життєвого циклу ПЗ; реалізувати життєвий цикл ПЗ для тестового програмного забезпечення. Після вивчення даного курсу «Життєвий цикл ПЗ» здобувачі набудуть таких Загальних та Фахових компетентностей та Програмних результатів навчання: К01. Здатність до абстрактного мислення, аналізу та синтезу. К02. Здатність застосовувати знання у практичних ситуаціях. К03. Здатність спілкуватися державною мовою як усно, так і письмово. К04. Здатність спілкуватися іноземною мовою мовою як усно, так і письмово. К14. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування. К15. Здатність розробляти архітектури, модулі та компоненти програмних систем. К20. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення. К26. Здатність до алгоритмічного та логічного мислення

	ПР23: Вміти документувати та презентувати результати розробки програмного забезпечення.
Ключові слова	Програмне забезпечення, життєвий цикл, розробка.
Формат курсу	Очний
Теми	Тема 1. *Вихідні вимоги до програмного забезпечення.* Тема 2. *Підходи до розробки програмного забезпечення.* Тема 3. *Вивчення обов'язків ролей у сучасних підходах до розробки ПЗ* Тема 4. *Аналіз прикладу життєвого циклу ПЗ при розробці ПЗ.* Тема 5. *Інтеграція. Розгортання. Постачання (CICD)* Тема 6. *Аналіз сучасних інструментів для впровадження CICD* Тема 7. *Версії програмного забезпечення.* Тема 8. *Видалення програмного забезпечення. Розробка документація*
Підсумковий контроль, форма	Екзамен у кінці семестру
Пререквізити	Для вивчення даного курсу студентам потрібні базові знання з курсів: - основи програмування; - алгоритми і структури даних; - об'єктно-орієнтоване програмування.
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	Інформаційні методи (лекції, презентації, лабораторні роботи, написання рефератів, виконання індивідуальних завдань, робота у групі, командна робота, обговорення, консультації для поглибленого розуміння тем, бесіда, ілюстрація, демонстрація), дедуктивні методи на основі узагальнень, евристичні методи (проблемна лекція), інтерактивні методи (дискусія).
Необхідне обладнання	Для проведення лекційних занять: • монітор TFT 23"; • системний блок (процесор Intel i5-6500, 8GB оперативної пам'яті, HDD 256GB) ; • мультимедійне обладнання (проектор, проекційний екран, дошка настінна, звуковий підсилювач та аудіосистема); • комутатор мережевий для доступу до мережі Internet. Для проведення лабораторних занять: • комп'ютерна лабораторія з 12-14 робочими місцями; • монітори TFT 23"; • системні блоки (процесор Intel i5-6500, 8GB оперативної пам'яті, HDD 256GB); • мультимедійне обладнання (проектор, проекційний екран, дошка настінна, звуковий підсилювач та аудіосистема); • комутатор мережевий для доступу до мережі Internet.
Критерії оцінювання (окремо для кожного виду навчальної діяльності)	Оцінювання проводиться за 100-бальною шкалою. Бали нараховуються за наступним співвідношенням: • Виконання лабораторних робіт: до 50 балів. • Екзамен 50 балів. Підсумкова максимальна кількість балів 100. Звітність за курс – екзамен в кінці семестру.

	Академічна доброчесність: Очікується, що роботи студентів будуть оригінальними дослідженнями чи міркуваннями. Списування та втручання в роботу інших студентів становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в написанні завдань є підставою для її незарахуванння викладачем, незалежно від масштабів плагіату чи обману. Жодні форми порушення академічної доброчесності не толеруються. **Відвідання занять** є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції та лабораторні зайняття курсу. Студенти повинні інформувати викладача про неможливість відвідати заняття. У будь-якому випадку студенти зобов'язані дотримуватися термінів визначених для виконання всіх видів робіт, передбачених курсом. **Література.** Уся література, яку студенти не зможуть знайти самостійно, буде надана викладачем виключно в освітніх цілях без права її передачі третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих. **Політика виставлення балів.** Враховуються бали, набрані при поточному контролі та бали за виконання лабораторних робіт. При цьому обов'язково враховуються присутність на заняттях та активність студента під час практичного заняття; недопустимість пропусків та запізнень на заняття; користування мобільним телефоном, планшетом чи іншими мобільними пристроями під час заняття в цілях не пов'язаних з навчанням; списування та плагіат; несвоєчасне виконання поставленого завдання і т. ін. **Оцінювання лабораторних робіт** (9 лабораторних робіт, максимальна кількість балів: 50) відбувається шляхом оцінки роботи студента під час проведення лабораторної роботи в аудиторії (0-5 балів за одну роботу) та захисту звіту по виконаній лабораторній роботі (0-5 балів за одну роботу). У підсумку, всі набрані бали множаться на коефіцієнт (0.56) для переведення у 50-ти бальну шкалу. Бали оцінювання лабораторних робіт нараховуються за наступним співвідношенням: 5 – студент в повному обсязі володіє навчальним матеріалом, має повне розуміння розглянутої теми, надає правильні відповіді на запитання по темі, код програми функціонує відповідно до завдання; 4 – студент достатньо розуміє розглянутий матеріал та принципи написаного ним коду програми, присутні неточності та незначні помилки у відповідях на запитання по темі, код програми функціонує відповідно до завдання (або з несуттєвими недоліками); 3 – студент не досить добре розуміє розглянутий матеріал та написаний ним код програми, вагається та надає неточні/не конкретні відповіді на запитання по темі, код програми функціонує неточно, або з помірними недоліками; 2 – студент погано розуміє розглянутий матеріал та написаний ним код програми, студент в більшості надає помилкові відповіді на питання по темі, код програми функціонує з суттєвими недоліками; 1 - студент погано розуміє розглянутий матеріал та написаний ним код програми, код програми не функціонує належним чином; 0 - студент зовсім не засвоїв розглянутий матеріал, написаний ним код програми не відповідає темі/не функціонує взагалі.

	Критерії отримання додаткових балів: написання тез, статті, участь у міжнародних, всеукраїнських конкурсах, навчання в школах та курсах за тематикою навчальної дисципліни. Кількість балів визначається процентом покриття змісту навчальної дисципліни.
Питання до заліку чи екзамену.	Вихідні вимоги до програмного забезпечення? Основні принципи розробки програмного забезпечення? Agile? Порівняйте Scurm та Kanban. Які основні ролі існують у методології Scrum? Які основні ролі існують у методології Kanban? Наведіть приклад завдання для розробника. Наведіть приклад Bug репорту. Для чого використовуються розділи та з чого вони складаються. Порівняйте роль тестувальника та розробника у Scrum? Порівняйте роль тестувальника та розробника у Kanban? Які основні обов'язки обов'язки Scrum майстера? Опишіть обов'язки власника продукту. Як відбувається версіонування коду? Як відбувається процедура вирішення конфліктів у програмному коді? Опишіть процес постійної інтеграції продукту. Опишіть процес постійної розробки коду. Опишіть процес постійної доставки коду. Які сучасні інструменти використовуються для CICD, їхні переваги та недоліки. Основні складові конвеєру по доставці програмного забезпечення? Конвеєр у Jenkins. Конвеєр у Kanban. Версіонування програмного забезпечення? Документація при розробці програмного забезпечення? Як відбувається видалення програмного забезпечення? Які наради відбуваються у Scrum? Основна ціль денної наради у Scrum?
Опитування	Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.

Схема курсу "Життєвий цикл ПЗ. Управління ІТ проектами"
для студентів спеціальності 121 – Інженерія програмного забезпечення

Тиж.	Тема, план, короткі тези	Форма діяльності (заняття)** *лекція, самостійна, дискусія, групова робота)	Література.*** Ресурси в інтернеті	Завдання, год	Термін виконання
1	Вихідні вимоги до програмного забезпечення.	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
1	Лаб.1. Створення project baseline для ІТ продукту	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
2	Вихідні вимоги до програмного забезпечення	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
2	Лаб.2 Ознайомлення з підходом до розробки Waterfall	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
3	Вивчення обов'язків ролей у сучасних підходах до розробки ПЗ	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
3	Лаб.3 Ознайомлення з підходом до розробки SCRUM	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
4	Вивчення обов'язків ролей у сучасних підходах до розробки ПЗ	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
4	Лаб.4 Ознайомлення з підходом до розробки Kanban	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
5	Створення багів	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
5	Лаб.5 Версіонування програмного забезпечення. локальна система контролю версій.	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
6	Аналіз прикладу життєвого циклу ПЗ при розробці ПЗ	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
6	Підсумкове заняття ЗМ 1	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
7	Версіонування програмного коду	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
7	Лаб.5 Версіонування програмного забезпечення. локальна система контролю версій.	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
8	Версії програмного забезпечення.	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
8	Лаб.6 Версіонування програмного забезпечення. децентралізована система контролю версій.	Лабораторна робота	Сайт курсу	2	кінець поточного тижня

9	Версії програмного забезпечення.	Лекція	[1], [2] Сайт курсу	2	Кінець поточного тижня
9	Лаб.7 Неперервна інтеграція та неперервна доставка (ci/cd). впровадження ci/cd використовуючи програмний пакет jenkins.	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
10	Інтеграція. Розгортання. Постачання (CICD).	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
10	Лаб.7 Неперервна інтеграція та неперервна доставка (ci/cd). впровадження ci/cd використовуючи програмний пакет jenkins.	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
11	Інтеграція. Розгортання. Постачання (CICD).	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
11	Лаб.8 Неперервна інтеграція та неперервна доставка (ci/cd). впровадження ci/cd використовуючи веб сервіс github actions workflow	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
12	Аналіз сучасних інструментів для впровадження CICD	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
12	Лаб.8 Неперервна інтеграція та неперервна доставка (ci/cd). впровадження ci/cd використовуючи веб сервіс github actions workflow	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
13	Аналіз сучасних інструментів для впровадження CICD	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
13	Лаб.9 Ведення документація у IT проекті.	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
14	Видалення програмного забезпечення. Розробка документація	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
14	Лаб.9 Ведення документація у IT проекті.	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
15	Видалення програмного забезпечення. Розробка документація	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
15	Лаб.9 Ведення документація у IT проекті.	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
16	Видалення програмного забезпечення. Розробка документація	Лекція	[1], [2] Лаб.6 Сайт курсу	2	кінець поточного тижня
16	Підсумкове заняття ЗМ 2	Лабораторна робота	Сайт курсу	2	кінець поточного тижня