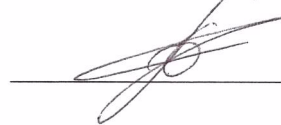


**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Львівський національний університет імені Івана Франка**  
**Факультет електроніки та комп'ютерних технологій**  
**Кафедра оптоелектроніки та інформаційних технологій**

**Затверджено**

На засіданні  
кафедри оптоелектроніки та  
інформаційних технологій  
факультету електроніки та комп'ютерних  
технологій  
Львівського національного університету  
імені Івана Франка  
(протокол №6 від 29.08, 2023 р.)

Завідувач кафедри:



Олег КУШНІР

**Силабус з навчальної дисципліни**  
**“Життєвий цикл ПЗ. Управління ІТ проектами”,**  
**що викладається в межах ОПП**  
**“ Високопродуктивний комп'ютинг ”**  
**першого (бакалаврського) рівня вищої освіти для здобувачів з**  
**спеціальності 121 – Інженерія програмного забезпечення**

Львів 2023 р.

<b>Назва дисципліни</b>	Життєвий цикл ПЗ. Управління ІТ проектами
<b>Адреса викладання дисципліни</b>	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
<b>Факультет та кафедра, за якою закріплена дисципліна</b>	Факультет електроніки та комп'ютерних технологій Кафедра оптоелектроніки та інформаційних технологій
<b>Галузь знань, шифр та назва спеціальності</b>	12 – інформаційні технології 121 – Інженерія програмного забезпечення
<b>Викладачі дисципліни</b>	Франів В.А., кандидат фізико-математичних наук, доцент кафедри оптоелектроніки та інформаційних технологій
<b>Контактна інформація викладачів</b>	<a href="mailto:volodymyr.franiv@lnu.edu">volodymyr.franiv@lnu.edu</a>
<b>Консультації з питань навчання по дисципліні відбуваються</b>	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
<b>Сторінка курсу</b>	
<b>Коротка анотація дисципліни</b>	Дисципліна “Життєвий цикл ПЗ. Управління ІТ проектами ” є нормативною дисципліною з спеціальності 121 – Інженерія програмного забезпечення для освітньої програми “Високопродуктивний комп'ютинг”, яка викладається в 7-му семестрі в обсязі 3.5-ох кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
<b>Інформація про дисципліну</b>	Курс розроблено для ознайомлення студентів з основними поняттями та методами які використовуються у життєвому циклі ПЗ
<b>Мета та цілі дисципліни</b>	<i>Мета:</i> Надати студентам глибоке розуміння ключових концепцій і методів, пов'язаних із життєвим циклом програмного забезпечення та управління проектами 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 ІТ-середовищі. <i>Цілі:</i> 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 ІТ-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

	<p>навичок та знань для ефективного управління та розробки програмного забезпечення в інформаційних технологіях.</p>
<p><b>Література для вивчення дисципліни</b></p>	<ol style="list-style-type: none"> <li>1) Навчальний посібник з дисципліни «Технології розробки програмного забезпечення» для студентів спеціальності 123 «Комп'ютерна інженерія». – Полтава: ПолтНТУ, 2017. – 218 с.</li> <li>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: <a href="http://dx.doi.org/10.30970/eli.23.7">http://dx.doi.org/10.30970/eli.23.7</a></li> <li>3) Pressman, Roger S. "Software Engineering: A Practitioner's Approach." McGraw-Hill Education, 2014.</li> <li>4) McConnell, Steve. "Code Complete: A Practical Handbook of Software Construction." Microsoft Press, 2004.</li> <li>5) Beck, Kent. "Extreme Programming Explained: Embrace Change." Addison-Wesley Professional, 2004.</li> <li>6) Brooks, Frederick P. "The Mythical Man-Month: Essays on Software Engineering." Addison-Wesley Professional, 1995.</li> <li>7) Gamma, Erich, Richard Helm, Ralph Johnson, and John Vlissides. "Design Patterns: Elements of Reusable Object-Oriented Software." Addison-Wesley Professional, 1994.</li> </ol>
<p><b>Обсяг курсу</b></p>	<p>Загальний обсяг: 105 годин. Аудиторних занять: 64 год., з них 32 год. лекційних та 32 годин лабораторних робіт. Самостійної роботи: 41 год.</p>
<p><b>Очікувані результати навчання</b></p>	<p>У результаті вивчення даного курсу студент буде:</p> <p><b>знати:</b>  основні поняття, визначення і проблеми курсу; вимоги до постановки основних задач життєвого циклу програмного забезпечення; призначення й особливості застосування різних життєвих циклів програмного забезпечення;</p> <p><b>вміти:</b>  володіти основними інструментами та підходами для створення життєвого циклу ПЗ; реалізувати життєвий цикл ПЗ для тестового програмного забезпечення.</p> <p>Після вивчення даного курсу «Життєвий цикл ПЗ» здобувачі набудуть таких Загальних та Фахових компетентностей та Програмних результатів навчання:</p> <p>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.  ФК13. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.  ФК14. Здатність брати участь у проектуванні програмного забезпечення,</p>

	<p>включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.</p> <p>ФК20. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.</p> <p>ФК21. Здатність оцінювати і враховувати економічні, соціальні, технологічні та екологічні чинники, що впливають на сферу професійної діяльності.</p> <p>ФК22. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.</p> <p>ФК23. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.</p> <p>ФК27. Здатність розробляти високопродуктивні програмні комплекси для вирішення задач наук про дані, систем штучного інтелекту, вбудованих та інших інноваційних систем.</p> <p>ПРН01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.</p> <p>ПРН02. Знати кодекс професійної етики, розуміти соціальну значимість та культурні аспекти інженерії програмного забезпечення і дотримуватись їх в професійній діяльності.</p> <p>ПРН03. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.</p> <p>ПРН04. Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.</p> <p>ПРН06. Вміти вибирати та використовувати відповідну до задачі методологію створення програмного забезпечення.</p> <p>ПРН07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.</p> <p>ПРН09. Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.</p> <p>ПРН14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.</p> <p>ПРН19. Знати та вміти застосовувати методи верифікації та валідації програмного забезпечення.</p> <p>ПРН22. Знати та вміти застосовувати методи та засоби управління проектами.</p> <p>ПРН23. Вміти документувати та презентувати результати розробки програмного забезпечення.</p> <p>ПРН24. Вміти проводити розрахунок економічної ефективності програмних систем.</p>
<b>Ключові слова</b>	Програмне забезпечення, життєвий цикл, розробка.
<b>Формат курсу</b>	Очний

<b>Теми</b>	<p>Тема 1. <i>Вихідні вимоги до програмного забезпечення.</i></p> <p>Тема 2. <i>Підходи до розробки програмного забезпечення.</i></p> <p>Тема 3. <i>Вивчення обов'язків ролей у сучасних підходах до розробки ПЗ</i></p> <p>Тема 4. <i>Аналіз прикладу життєвого циклу ПЗ при розробці ПЗ.</i></p> <p>Тема 5. <i>Інтеграція. Розгортання. Постачання (CICD)</i></p> <p>Тема 6. <i>Аналіз сучасних інструментів для впровадження CICD</i></p> <p>Тема 7. <i>Версії програмного забезпечення.</i></p> <p>Тема 8. <i>Видалення програмного забезпечення. Розробка документація</i></p>
<b>Підсумковий контроль, форма</b>	Екзамен у кінці семестру
<b>Пререквізити</b>	Для вивчення даного курсу студентам потрібні базові знання з курсів: - об'єктно-орієнтоване програмування.
<b>Навчальні методи та техніки, які будуть використовуватися під час викладання курсу</b>	Інформаційні методи (лекція, бесіда, ілюстрація, демонстрація); дедуктивні методи на основі узагальнень; евристичні методи (проблемна лекція); інтерактивні методи (дискусія).
<b>Необхідне обладнання</b>	<p>Для проведення лекційних занять: комп'ютер (мінімальні характеристики: процесор Intel Core i3(4 ядра/8 потоків), 8ГБ оперативної пам'яті, 50ГБ вільного місця на диску, відеокарта Nvidia GEFORCE GT1030 2048MB), доступ до мережі Internet, засоби мультимедіа (в т.ч. проектор).</p> <p>Для проведення лабораторних занять: Комп'ютер (мінімальні характеристики: процесор Intel Core i3(4 ядра/8 потоків), 8ГБ оперативної пам'яті, 50ГБ вільного місця на диску, відеокарта Nvidia GEFORCE GT1030 2048MB). ОС Win10. MS Visual Studio 2017.</p>
<b>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</b>	<p>Оцінювання проводиться за 100-бальною шкалою. Бали нараховуються за наступним співвідношенням:</p> <ul style="list-style-type: none"> <li>• Виконання лабораторних робіт: до 50 балів.</li> <li>• Екзамен 50 балів.</li> </ul> <p>Підсумкова максимальна кількість балів 100. Звітність за курс – екзамен у кінці семестру.</p> <p><b>Академічна доброчесність:</b> 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 незарахування викладачем, незалежно від масштабів плагіату чи обману. Жодні форми порушення академічної доброчесності не толеруються.</p> <p><b>Відвідання занять</b> є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції та лабораторні заняття 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.</p> <p><b>Література.</b> Уся література, яку студенти не зможуть знайти самостійно, буде надана викладачем виключно в освітніх цілях без права її передачі</p>

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 запізнь 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** (7 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 (0.71) 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.

**Критерії оцінювання результатів неформальної освіти:**

Нарахування балів відбувається за написання студентом тез доповідей на конференціях, наукових статей, участь у діяльності наукових гуртків, участь у наукових семінарах та круглих столах, конкурсах, участь у заходах неформальної освіти за отримання сертифікатів про проходження навчання на різних освітніх платформах (Coursera, Prometheus тощо), курсах на

	<p>провідних ІТ компаніях за тематикою навчальної дисципліни. Кількість балів визначається відсотком покриття результатів відповідної активності до вимог результатів навчання з навчальної дисципліни.</p>
<p><b>Питання до заліку чи екзамену.</b></p>	<p>Вихідні вимоги до програмного забезпечення?          Основні принципи розробки програмного забезпечення?          Agile?          Порівняйте Scrum 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 CI/CD, 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?</p>
<p><b>Опитування</b></p>	<p>Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.</p>

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

<b>Тиж.</b>	<b>Тема, план, короткі тези</b>	<b>Форма діяльності (заняття)* *лекція, самостійна, дискусія, групова робота)</b>	<b>Література.** * Ресурси в інтернеті</b>	<b>Завдання, год</b>	<b>Термін виконання</b>
1	Вихідні вимоги до програмного забезпечення.	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
1	Лаб.1.Створення базового плану проєкту (Project Baseline)	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.1 Створення базового плану проєкту (Project Baseline)	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.2. Робота в Atlassian jira (Створення jira issue. Epic, user story)	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.2. Лаб.2. Робота в Atlassian jira (Створення jira issue. Epic, user story, bugs)	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
5	Створення багів	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
5	Лаб.3. Версіонування програмного коду	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.4. Вирішення конфліктів	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
8	Версії програмного забезпечення.	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
8	Лаб.4 Вирішення конфліктів, документація	Лабораторна робота	Сайт курсу	2	кінець поточного тижня



9	Версії програмного забезпечення.	Лекція	[1], [2] Сайт курсу	2	Кінець поточного тижня
9	Лаб.5 Побудова CICD за допомогою 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.5 Побудова CICD за допомогою 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.5 Побудова CICD за допомогою Jenkins	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.6 Побудова CICD за допомогою GitLab CICD	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.6 Побудова CICD за допомогою GitLab CICD	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.6 Побудова CICD за допомогою GitLab CICD	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.7 Розробка документації	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