

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

Затверджено

На засіданні
кафедри системного проектування
факультету електроніки та комп'ютерних
технологій
Львівського національного університету
імені Івана Франка
(протокол № 1 від 30.08.2021 р.)

Завідувач кафедри:



Роман ШУВАР

Силабус з навчальної дисципліни
“Вступ в інженерію програмного забезпечення”,
що викладається в межах ОПП
“ Високопродуктивний комп'ютинг ”
першого (бакалаврського) рівня вищої освіти для здобувачів з
спеціальності 121 – Інженерія програмного забезпечення

Львів 2021 р.

Назва дисципліни	Вступ в інженерію програмного забезпечення
Адреса викладання дисципліни	м. Львів, вул. Драгоманова, 50
Факультет та кафедра, за якою закріплена дисципліна	Факультет електроніки та комп'ютерних технологій, кафедра системного проектування
Галузь знань, шифр та назва спеціальності	12 Інформаційні технології 121 Інженерія програмного забезпечення (ВПК)
Викладачі дисципліни	Ляшкевич Василь Яремович, доцент
Контактна інформація	vasyl.lyashkevych@lnu.edu.ua , https://electronics.lnu.edu.ua/employee/liashkevych-v-ya
Консультації з питань навчання по дисципліні відбуваються	Консультації в день проведення лекційних занять (за попередньою домовленістю). Також можливі он-лайн консультації через MS Teams або систему електронного навчання Moodle. Для погодження часу онлайн консультацій слід писати на електронну пошту викладача.
Сторінка дисципліни	https://moodle.elct.lnu.edu.ua/course/view.php?id=103
Інформація про дисципліну	Дисципліна «Вступ в інженерію програмного забезпечення» є нормативною дисципліною з спеціальності 121 Інженерія програмного забезпечення для освітньої програми «Високопродуктивний комп'ютинг», яка викладається в 1 семестрі в обсязі 4 кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Коротка анотація дисципліни	Навчальну дисципліну розроблено таким чином, щоб надати учасникам необхідні знання і розуміння сфери програмної інженерії, щоб оволодіти базовими поняттями, пов'язаними з виробничими процесами в індустрії програмного забезпечення, принципи організації компаній та правила роботи на проекті, правила взаємодії між працівниками, порядок підготовки та набору, підготовка та освіта. Саме тому у дисципліні подано огляд питань працевлаштування та роботи, моделі і принципи розробки програмного забезпечення, а також питання управління персоналом.
Мета та цілі дисципліни	Метою вивчення нормативної дисципліни «Вступ в індустрію програмного забезпечення» є оволодіння базовими поняттями, теоретичними знаннями та практичними навичками для роботи в індустрії програмного забезпечення з метою швидкої адаптації у виробничі процеси та для роботи в багато національних командах.
Література для вивчення дисципліни	Основна література: 1. Rajib Mall. Fundamentals of software engineering. - [Режим доступу]: https://davcollegeititilagarh.org/wp-content/uploads/2020/09/fundamentals-of-software-engineering-fourth-edition-rajib-mall.pdf

	<p>2. Вступ до інженерії програмного забезпечення: навч. посібник / Є.В. Левус, Н.Б. Мельник – Львів: Видавництво Львівської політехніки, 2017. - 280 с.</p> <p>3. Лавріщева К.М. Програмна інженерія. - К. - 2008. - 319 с. - ISBN: 978-966-02-5052-9.</p> <p>4. Software engineering / Ian Sommerville. - 9th ed. - 790 p. - ISBN-13: 978-0-13-703515-1.</p> <p>5. Vencel, L., Cook, S. & Matthews, D., A Systems Analysis of the DCPG Capability Domain Architecture - Part One: Characteristics of an Idealized Domain and Domain Set. DSTO Technical Report, Edinburgh, Australia, 2002</p>
Обсяг курсу	Кількість кредитів ЄКТС: 4 (120 год), з них: 64 годин аудиторних занять (лекції: 32 год, лабораторні: 32 год.) та 56 год. самостійної роботи.
Очікувані результати навчання	<p>Після вивчення даного курсу здобувачі набудуть таких Загальних(ЗК)/Фахових(ФК) компетентностей та Програмних результатів навчання (ПРН):</p> <p>ЗК01. Здатність до абстрактного мислення, аналізу та синтезу.</p> <p>ЗК02. Здатність застосовувати знання у практичних ситуаціях.</p> <p>ЗК04. Здатність спілкуватися іноземною мовою як усно, так і письмово.</p> <p>ЗК07. Здатність працювати в команді.</p> <p>ФК13. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.</p> <p>ФК20. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.</p> <p>ФК25. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.</p> <p>ФК29. Здатність здійснювати розробку програмного забезпечення використовуючи різні методології та засоби програмування з метою забезпечення їх високої надійності та продуктивності в роботі.</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>ПРН04. Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.</p> <p>ПРН06. Вміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.</p> <p>ПРН09. Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.</p> <p>ПРН20. Знати підходи щодо оцінки та забезпечення якості програмного забезпечення.</p>
Ключові слова	Інженерія програмного забезпечення, робота в ІТ, моделі розробки програмного забезпечення, управління персоналом, система доставок, Software Engineering, Project Management,

	Product Management, Agile, Delivery.
Формат курсу	Очний. Проведення лекцій, лабораторних робіт та консультації для кращого розуміння тем.
Теми	Див. СХЕМА КУРСУ
Підсумковий контроль, форма	Екзамен в кінці семестру
Пререквізити	Для вивчення курсу студенти не потребують жодних додаткових знань окрім тих, що отримали протягом навчання в середній школі.
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	Презентація, лекції, лабораторні роботи, обговорення, дискусія.
Необхідне обладнання	Мультимедіа, платформа Moodle
Критерії оцінювання (окремо для кожного виду навчальної діяльності)	<p>Оцінювання проводиться упродовж семестру за 100-бальною шкалою, де враховано бали за два контрольні заміри по 25 балів за кожний модуль та 50 балів за складання екзамену.</p> <p>Бали нараховуються за такими видами робіт з наступним співвідношенням:</p> <ul style="list-style-type: none"> • контрольні заміри (2 модулі): 50% семестрової оцінки; максимальна кількість балів 50, а саме: <ul style="list-style-type: none"> - лабораторні роботи: 60% оцінки контрольного заміру; максимальна кількість балів 32 (6 лабораторних робіт). - теоретичний матеріал: 40% оцінки контрольного заміру; максимальна кількість балів 18 (2 тести по 9 балів кожний, де 1 питання має вагу 0.5 бала). • екзамен: 50% семестрової оцінки; максимально 50 балів. <p>Бали за лабораторними роботами розподіляються так:</p> <ul style="list-style-type: none"> - лаб. робота 1 оцінюється в 6 балів, де від 0 до 5 балів за виконання та від 0 до 1 балу за тестування чи опитування. - лаб. робота 2 оцінюється в 8 балів, де від 0 до 7 балів за виконання та від 0 до 1 балу за тестування чи опитування. - лаб. робота 3 оцінюється в 2 бали, де від 0 до 1 балу за виконання та від 0 до 1 балу за тестування чи опитування. - лаб. робота 4 оцінюється в 6 балів, де від 0 до 5 балів за виконання та від 0 до 1 балу за тестування чи опитування. - лаб. робота 5 оцінюється в 4 бали, де від 0 до 3 балів за виконання та від 0 до 1 балу за тестування чи опитування. - лаб. робота 6 оцінюється в 6 балів, де від 0 до 5 балів за виконання та від 0 до 1 балу за тестування чи опитування. <p>Оцінювання екзаменаційних питань:</p> <p>20-25 балів - розглянута тема відтворюється в повному обсязі, правильно, обґрунтовано, логічно, які містять аналіз і систематизацію, аргументовані висновки. Засвідчено глибоке володіння матеріалом. Наведені приклади коду повністю робочі та</p>

	<p>відповідають темі. Можуть бути присутні несуттєві помилки та невідповідності;</p> <p>15-19 балів - відтворюється значна частина розглянутої теми. Виявлено знання і розуміння основних положень навчальної дисципліни, проте присутні неточності та/або невідповідності основній темі. Наведені приклади коду частково робочі, проте в загальному відповідають темі;</p> <p>10-14 балів - відстежується загальне розуміння розглянутої теми. Виявлені множинні неточності та невідповідності, пояснення наведеного коду відсутні, код функціонує із значними неточностями (або відсутні приклади запуску коду на виконання взагалі);</p> <p>5-9 балів – студент погано розуміє розглянуту тему. Виявлені суттєві неточності та невідповідності. Наведені приклади коду з суттєвими недоліками, або не відповідають темі;</p> <p>0-4 бали – студент взагалі не розуміє розглянуту тему. Тему не розкрито, кількість викладеного матеріалу не відповідає загальним нормам обраного виду роботи. Наведений код не робочий, або відсутній як такий.</p> <p>Академічна доброчесність: Очікується, що лабораторні роботи та контрольні роботи студентів будуть їх оригінальними дослідженнями чи міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в роботі студента є підставою для її незарахування викладачем.</p> <p>Відвідання занять є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції і лабораторні заняття курсу. Студенти мають інформувати викладача про неможливість відвідати заняття. Студенти зобов'язані дотримуватися усіх термінів визначених для виконання усіх видів робіт, передбачених курсом.</p> <p>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.</p> <p>Політика виставлення балів. Враховуються бали набрані на поточному тестуванні, самостійній роботі та бали підсумкового тестування. При цьому обов'язково враховуються присутність на заняттях та активність студента під час лабораторного 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. Жодні форми порушення академічної доброчесності не толеруються.</p>
<p>Питання до контрольних робіт</p>	<p>Перелік питань та завдань для проведення підсумкової оцінки знань певних тем до контрольних робіт:</p> <ol style="list-style-type: none"> 1. Розкрийте поняття “професійної розробки програмного

	<p>забезпечення”. (Поняття програмної інженерії, характеристики ПЗ, програмний продукт, типи програмних продуктів, різноманітність в програмній інженерії, типи ПЗ, типи програмних систем, поняття веб-застосунків.)</p> <ol style="list-style-type: none"> 2. Робота і кар’єра в ІТ компанії. (Робота в галузі інформаційних технологій. Правила комунікації та взаємодії в ІТ компанії. Кар’єра в ІТ: менеджер, розробник, тестувальник та ін. Загальна ієрархія позицій в ІТ компанії. R&D активності. L&D активності. Освітні активності.) 3. Поясніть “Концепцію програмної інженерії”. (Елементи важливості програмної інженерії. Основи програмної інженерії. Етика програмної інженерії. Робота в галузі інформаційних технологій. Типові обов’язки та компетенції інженерів програмного забезпечення. Наведіть приклад організації системи схематично.) 4. Поясніть особливості підготовки резюме, розділи резюме. (Процес рекрутингу в ІТ компанії. Процес працевлаштування в ІТ компанію. Пошук позиції. Резюме. Основні розділи резюме та їх призначення з наведеними прикладами. Додаткові розділи в резюме. Портфоліо. Рекомендаційні листи.) 5. Опишіть процеси в програмній інженерії. (Виробничі процеси. Класифікація процесів ЖЦ ПЗ. Складові елементи еталонної моделі інженерії ПЗ. Активності і моделі ЖЦ ПЗ. Типи моделей. Каскадна модель. Еволюційна модель. Різниця між каскадною та еволюційною моделями.) 6. Опишіть процеси програмної інженерії в моделях зрілості ІТ компаній. (Модель зрілості процесів Capability Maturity Model. Рівні зрілості процесів ІТ-компаній. Активності і моделі ЖЦ ПЗ. Персонал.) 7. Поясніть процеси в програмній інженерії. (Моделі ЖЦ ПЗ. Типи моделей. Модель ПЗ, орієнтована на повторне використання. Спіральна модель. Інкрементна модель. Різниця між спіральною та інкрементною моделями. Загальна модель проектування процесів ПЗ.) 8. Процеси в програмній інженерії пов’язані зі складністю програмного забезпечення. (Складність ПЗ. Комплексні потоки даних. “Бермудський трикутник” моделювання. Активності у процесах програмної інженерії. Керовані планом та Agile процеси. Розробка зручності використання згідно ISO 9241. Модель ПЗ, орієнтована на повторне використання. Прототипування.) 9. Визначення вимог до програмних систем. (Класифікація вимог. Функціональні і не функціональні вимоги. Аналіз і збирання вимог. Інженерія вимог. Фіксація вимог. Трасування вимог. Об’єктно-орієнтована інженерія вимог.) 10. Поняття системного моделювання. (Поняття бізне-логіки та бізне-процесів. CASE-технології: концепція, цілі, засоби. Архітектура CIMOSA. Планування на основі можливостей / Capability-based planning. Загальний процес СВР, наведіть приклади. Впровадження СВР. Оцінка можливостей.) 11. Поняття моделювання системи. (Поняття бізне-процесів. Поняття планування на основі можливостей / Capability-based planning. Моделювання системи. Контекстні моделі. Моделі процесів. Моделі взаємодії. Моделювання варіантів використання. Діаграма послідовності. Структурні моделі. Поведінкові моделі.) 12. Поняття моделювання системи на основі ідеї. (Стартап ідеї в бізнес середовищі. Пошук ідеї для продукту. Реалізація стаптану як інноваційного проекту. Проектування сценаріїв. Прототипування.)
--	---

	<p>Компоненти системи та зв'язки між ними. Сервіс-орієнтований підхід.)</p> <p>13. Методи об'єктного аналізу і моделювання. (Основні поняття. Об'єктно-орієнтовані методи аналізу і побудови моделей. Метод побудови об'єктної моделі предметної області. Проектування архітектури програмних систем. Загальні підходи до проектування. Різні види архітектур. Поняття і шаблони архітектурних стилів.)</p> <p>14. Поняття проектування програмного продукту чи системи. (Шлях від ідеї до застосунку. Перетворення труднощів у цілі. Основні атрибути програмного продукту. Вимоги ПЗ та їх класифікація. Класифікація компонентів проектування ПЗ. Тестування, якість та підтримка ПЗ.)</p> <p>15. Поняття архітектури ПЗ. (Поняття ПЗ. Поняття архітектури ПЗ: цілі та концепція. Особливості застосунків. Архітектурні стилі. Архітектура Trade-offs. Інвентаризаційна архітектура. Шлях користувача. Діаграма послідовностей. Високо-рівнева архітектура застосунків. Поняття мікросервісної архітектури.)</p> <p>16. Поняття архітектури рішення ПЗ. (Поняття архітектури ПЗ. Особливості застосунків. Шлях користувача. Поняття розповіді / story. Проектування сценаріїв на основі розповідей. Технічне рішення. Архітектура рішення на основі сценаріїв. Поділ розповідей на задачі. Розподіл задач для їх виконання серед команди. Оцінювання задач та розповідей для виконання. Засоби моніторингу виконання задач: Jira, Trello.)</p> <p>17. Розкрийте поняття надійності програмного забезпечення. (Базові характеристики якості програмного продукту. Надійність ПЗ. Моделі оцінки надійності ПЗ. Грунтовні поняття проблематики надійності. Класифікація моделей надійності. Процеси оцінки надійності.)</p> <p>18. Розкрийте поняття якості програмного забезпечення. (Базові характеристики якості програмного продукту. Модель якості програмних систем. Стандартні показники якості. Метрики якості. Стандартна оцінка показників якості. Керування якістю ПЗ. Сертифікація ПЗ.)</p> <p>19. Розкрийте поняття гарантії якості програмного забезпечення. (Базові характеристики якості програмного продукту. Процеси для гарантії якості ПЗ. Вклад QA інженера та його обов'язки у життєвий цикл розробки ПЗ. Ручне тестування. Автоматичне тестування. Типи тестування ПЗ. Навички QA інженера.)</p> <p>20. Поясніть активності по тестуванню ПЗ. (Поняття тестування ПЗ. Теоретичні обмеження. Практичні обмеження. Приклад обмеження на алгоритмі Дейкстри. Модульне тестування /Unit test. Інтеграційний тест / Integration test. Функціональний тест / Functional test. Тести продуктивності / Performance test. Тест прийняття / Acceptance test. V модель тестування. Планування тестів. Керування відмовами. Гарантії якості.)</p> <p>21. Тестування ПЗ. (Поняття тестування ПЗ. Типи тестування. Системне тестування. Модульне тестування. Чорний ящик. Білий ящик. Тестування циклів. Побудова діаграми логічного потоку. Пошук тестових випадків.)</p> <p>22. Стратегії та евристики тестування ПЗ. (Поняття тестування ПЗ. Типи тестування. Етапи тестування / Розроблення / Випуск / Користувач. Тестові випадки. Пошук тестових випадків. Стратегія Out of time. Побудова тестових випадків для Out of time. Креативність в тестуванні. Тестування в моделі ЖЦ Водоспаду / Waterfall.)</p>
--	--

23. Поняття менеджменту проекту. (Проектування на основі можливостей / Capability-based planning - СВР. Етапи СВР підходу: сценарії, бізнес процес, робочі задачі, можливості, активності. Поняття та характеристики проекту. Поняття та характеристики проекту. Активності менеджменту.)
24. Поняття сфери діяльності менеджменту проекту. (Сфера діяльності менеджменту проекту. План керування проектом ПЗ: структура, ключові етапи, графік планування. Наскрізне управління проектами: ініціалізація, початок проекту, затримки. Декомпозиція проекту: функції, активності, задачі. Приклад активностей проекту.)
25. Оцінка ризиків в діяльності менеджменту проекту. (Сфера діяльності менеджменту проекту. Декомпозиція проекту та аналіз складності. Оцінка задач проекту. План керування проектом ПЗ: структура, ключові етапи, графік планування. Призначення ролей на проекті. Взаємозв'язок задач, активностей та ролей на проекті. Поняття ризиків. Ефекти ризиків. Модель зрілості керування ризиками.)
26. Методи планування проекту. (Поняття проекту: особливості, цілі та результати. Дерево рішень. Трикутник проектного менеджменту. "Залізний трикутник" обмеження проекту. Основні функції проектного менеджера. Типи робіт, що складають процес менеджменту. Система менеджменту якості. Загальні поняття оцінювання проекту. Цільова аудиторія. Конкуренція. Формування команди. Формування команди за Agile та "гібридним" підходами.)
27. Особливості керування проекту. (Поняття проекту: особливості, цілі та результати. Поняття керування проекту. Керування розробкою ПЗ. Профіль процесу керування проекту. Інфраструктура програмного проекту. Організаційне забезпечення проекту. Модель відповідності осіб.)
28. Методи керування проекту. (Поняття керування проекту. Процес керування проектом. Метод критичного шлязу СРМ. Метод аналізу й оцінки проекту - PERT. Види планів організації проекту. Етапи графіку робіт на проекті. Операційни граф плану проекту.)
29. Моделі якості інтерфейсу користувача. (Поняття якості ПЗ. Людино-машинний інтерфейс. Шаблони інтерфейсу користувача. Мінімальні критерії до інтерфейсу користувача. Метрики якості та критерії прийняття / Acceptance criteria.)
30. Різновиди тестування. (Поняття тестування ПЗ. Функціональне та не функціональне тестування. Статичне та динамічне тестування. Класифікація видів тестування. Умови тестування: нормальні, кестремальні та виняткові ситуації. Граничне випробування. Документація.)
31. Розроблення програмного забезпечення згідно Agile методології. (Поняття Agile технології. План-орієнтована і "жива" розробка. Модель екстремального програмування. Основні фази моделі. Розповіді. Правила Agile технології: живе планування, зміна версій, прості проектні рішення, розробка на основі тестування, постійна переробка, програмування парами та постійна інтеграція. Масштабування agile методів.)
32. Проектування архітектури програмного забезпечення. (Поняття архітектури ПЗ. Проектування рішення архітектури. Архітектурні види. Архітектурні шаблони і стилі. Архітектура застосунку.)
33. Проектування та розроблення програмного забезпечення. (Поняття архітектури ПЗ. Поняття UML. Об'єктно-орієнтоване проектування. Шаблони проектування. Проблеми розроблення. Відкриті ресурси

- для розроблення.)
34. Доведення і верифікація програмного забезпечення. (Поняття доведення / validation ПЗ. Поняття верифікації ПЗ. Мови специфікації програм. Методи доведення правильності програм. Верифікація і валідація програм. Визначення зробленого.)
 35. Групова динаміка та комунікації. (Поняття групової динаміка. Колективна робота в галузі ІТ. Комунікації у групі. Поняття групи. Діловий та соціальний аспекти групи. Формальні і неформальні групи. Місце індивіда в групі. Рольова напруга і рольовий конфлікт. Командні ролі.)
 36. Особливості сервіс-орієнтованої архітектури. (Поняття архітектури ПЗ. Компоненти, що можна повторно використовувати. Сервіси як компоненти, які можна повторно використовувати. Інженерія сервісів. Веб-сервіси. Поняття API. Сервіс-орієнтована розробка ПЗ.)
 37. Моделювання та моделі системи. (Поняття архітектури системи. Компоненти програмної системи. Контекстні моделі. Перехресні моделі. Структурні моделі. Поведінкові моделі. Модель-керована інженерія.)
 38. Поняття побудови керованих систем. (Поняття архітектури системи. Системи, що керовані даними / Data-driven systems. Архітектура потоків даних. Системи, що керовані подіями / Event-driven systems. Поняття події. Архітектури ситсеми, що керуються подіями. Опрацювання подій.)
 39. Поняття даних в програмних системах. (Поняття бази даних. Поняття сховища даних. Поняття розподілених систем. Архітектура даних. Архітектура розподілених ситеми. Поняття про великі дані / Big Data.)
 40. Поняття хмарних систем та сервісів. (Поняття хмарної системи. Поняття хмарних сервісів. Можливості хмарних сервісів. Провайдери хмарних ресурсів: Амазон / AWS, Гугл / GCP, Мікрософт / Azure, ІВМ / IBM Cloud / IBM Watson.)
 41. Визначте вимоги та спроектуйте виробничі процеси для заданого програмного продукту (Веб-магазин, цифровий лікар, електронний помічник і т.п.) згідно моделі (Водопаду, Ітераційна водопаду, інкремента та ін.) розробки програмного забезпечення.
 42. Запропонуйте чотири сценарії та основі них розробіть систему прототипування для заданого програмного продукту (Веб-магазин, цифровий лікар, електронний помічник і т.п.). Зазначте ризики, переваги та недоліки запропонованої ситеми прототипів.
 43. Запропонуйте систему тестування на основі V-моделі для заданого програмного продукту (Веб-магазин, цифровий лікар, електронний помічник і т.п.), який розробляється за заданою моделлю (Водопаду, Ітераційна водопаду, інкремента та ін.).
 44. Виконайте плануванн на основі можливостей - Capability-based planning використовуючи CASE технології (Діаграма послідовності, поведінкова модель, моделі процесів та ін.). Задайте метрики та їх критерії для оцінки сценаріїв можливостей в часі (3 довільні часові періоди).
 45. Запропонуйте рішення для заданого програмного продукту (Веб-магазин, цифровий лікар, електронний помічник і т.п.) і представте його у вигляді шляху користувача (> 5 етапів), архітектури рішення (> 5 компонентів) та сервіс-орієнтованої архітектури (>2 сервісів) та заданих CASE діаграм (Діаграма послідовності, діаграма прецедентів, і т.п.).
 46. Запропонуйте рішення для заданого програмного продукту (Веб-

	<p>магазин, цифровий лікар, електронний помічник і т.п.) та зробіть його декомпозицію за допомогою розповідей (> 2 розповідей) та їх підзадач (> 3 підзадач для однієї розповіді). Сформууйте команду, визначте ролі та призначте відповідальних за виконання кожної розповіді, підзадачі. Зображте загальну схему розповідей та підзадач. Одна розповідь може вміщати як інші розповіді так і задачі чи підзадачі.</p> <p>47. Розробіть систему функціонального тестування для заданого програмного продукту (Веб-магазин, цифровий лікар, електронний помічник і т.п.) та опишіть життєвий цикл тестування ПЗ (Software Testing Life Cycle - STLC) використовуючи V-модель тестування.</p> <p>48. Розробіть систему НЕ функціонального тестування для заданого програмного продукту (Веб-магазин, цифровий лікар, електронний помічник і т.п.) та опишіть життєвий цикл тестування ПЗ (Software Testing Life Cycle - STLC) використовуючи критерії приймання користувача.</p> <p>49. Побудуйте алгоритмічну структуру (> 10 компонентів) чи покомпонентну модель (> 5 компонентів) для заданого програмного продукту (Веб-магазин, цифровий лікар, електронний помічник і т.п.) та відшукайте тестові випадки. Далі, розробіть відповідну систему тестування, де зазначено яким підходом до тестування (Модульне тестування, Чорна скринька, Інтеграційне тестування, Тестування несвоєчасного виконання та ін.) які тестові випадки забезпечуються.</p> <p>50. Нехай задано проект (Веб-магазин, цифровий лікар, електронний помічник і т.п.), в команді якого є представники компанії замовника (>3 людей) та представники компанії виконавця (> 8 людей). Розробіть найперспективнішу для компанії виконавця карту сили / Power Map для заданого проекту використовуючи соціальні стилі поведінки (Аналітичний / Analytical, Водійський / Driving, Дружній / Amiable, Виразний / Expressive), при умові, що у компанії замовника такі ролі: “Аналітичний” є підопічним “Водійського”, “Виразний” консультується з “Аналітичним”, а у компанії виконавця є такі люди: 2 людей - Водіями, 2 людей - Аналітиками, 3 людини - Дружніми, та 1 людина - Виразний. Поясніть своє рішення і можливі потенційні конфлікти в колективі.</p>
Опитування	Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.

СХЕМА КУРСУ

Тиж.	Тема, план, короткі тези	Форма діяльності	Література. Ресурси в Інтернеті.	Завдання, год	Термін виконання, тиж.
1	Поняття інженерії програмного забезпечення. Основні поняття та характеристики програмного забезпечення. Професійна розробка програмного забезпечення. Етика програмної інженерії. Робота в галузі інформаційних технологій. Типові обов'язки інженерів програмного забезпечення. Вимоги до м'яких навичок.	лекція	1-5	2	кінець поточного тижня
	Оцінка кар'єрних очікувань	лаб. робота	1-2	6	кінець 3-го тижня
	Професійна розробка програмного забезпечення.	сам. робота	1-3	3,5	кінець поточного тижня
2	Виробничі процеси. Виробничі процеси, активності і моделі життєвого циклу програмного забезпечення. Типи моделей. Каскадна модель. Інкрементна модель. Спіральна модель. Еволюційна модель.	лекція	1-5	2	кінець поточного тижня
	Моделі життєвого циклу програмного забезпечення.	сам. робота	1-4	3,5	кінець поточного тижня
3	Стандарти розроблення програмних систем. Вимоги до програмних систем. Стандарти розроблення ПЗ. Поняття об'єктного аналізу. Вимоги до інженерів програмного забезпечення.	лекція	3-4	2	кінець поточного тижня
	Еволюційні моделі життєвого циклу програмного забезпечення.	сам. робота	1-4	3,5	кінець поточного тижня
4	Методологія "Agile". Гнучка розробка програмного забезпечення. Методологія Agile. Розробка, керована планом. Гнучке керування проектами. Керування	лекція	2, 3	2	кінець поточного тижня

	продуктом. Аналіз продукту. Поняття історії і функції продукту.				
	Оцінка ринку для бізнес стартап-ідеї	лаб. робота	1, 5	8	кінець 7-го тижня
	Вимоги до програмних систем.	сам. робота	1-4	3,5	кінець поточного тижня
5	Вступ до інженерії вимог. Функціональні й нефункціональні вимоги. Виявлення та аналіз вимог. Специфікації вимог. Вимоги до виробничих процесів. Вимоги до валідації та управління.	лекція	1, 4	2	кінець поточного тижня
	Методології розроблення програмних систем.	сам. робота	1-3	3,5	кінець поточного тижня
6	Поняття системного моделювання. Поняття системного моделювання. Бізнес аналіз. Планування на основі можливостей. Моделі зрілості. Моделі: контекстна, інтерактивна, структурна, поведінкова. Поняття UML.	лекція	1, 3-4	2	кінець поточного тижня
	Методологія Agile.	сам. робота	1, 4	3,5	кінець поточного тижня
7	Вступ до проектування архітектури. Поняття архітектурного проектування. Архітектурні рішення, вигляди, шаблони. Архітектура програмного забезпечення. Шаблони проектування. Проблеми реалізації. Розробка з відкритим кодом.	лекція	1, 4-5	2	кінець поточного тижня
	Інженерія вимог.	сам. робота	1-4	3,5	кінець поточного тижня
8	Гарантія якості і тестування. Що таке гарантія якості. Поняття тестування та еволюції програмного забезпечення. Тестування розробки. Тестування випуску. Тестування користувача. Статичні і динамічні методи тестування. Функціональне тестування. Інженерія керування тестуванням. Еволюційні процеси.	лекція	1, 4	2	кінець поточного тижня

	Підтримка програмного забезпечення.				
	Вибір команди для реалізації проекту	лаб. робота	2-3	2	кінець поточного тижня
	Моделювання, поняття UML	сам. робота	1-2	3,5	кінець поточного тижня
9	Методи доведення та верифікація програмного забезпечення. Методи доведення і верифікації програмного забезпечення. Мови специфікації програм. Методи доведення правильності програм. Верифікація і валідація програм. Визначення зробленого.	лекція	3, 4	2	кінець поточного тижня
	Технічне проектування	лаб. робота	1, 4	6	кінець 11-го тижня
	Показники якості програмного забезпечення.	сам. робота	1-4	3,5	кінець поточного тижня
10	Керування якістю. Моделі та метрики якості програмних систем. Стандартні показники якості. Метрики якості. Керування якістю програмних систем. Моделі оцінки надійності програмних систем. Процеси оцінки надійності. Сертифікація програмного продукту.	лекція	1, 3	2	кінець поточного тижня
	Поняття та метрики якості програмного забезпечення.	сам. робота	1-4	3,5	кінець поточного тижня
11	Менеджмент проекту. Поняття управління проектом. Основні поняття та задачі. Головні цілі та процес менеджменту проекту. Командна робота. Управління персоналом.	лекція	1, 3-4	2	кінець поточного тижня
	Поняття менеджменту проекту.	сам. робота	2-4	3,5	кінець поточного тижня
12	Менеджмент стартап-проектів. Управління стартапами як формою інноваційного бізнесу. Організація стартапу: від команди до підприємства. Бізнес-ідея стартап-проекту та створення	лекція	1, 3-4	2	кінець поточного тижня

	життєздатного продукту. Бізнес-моделювання стартапу. Маркетинговий менеджмент. Бізнес-планування стартапу.				
	Організація виробничих процесів	лаб. робота	1-3	4	кінець 13-го тижня
	Методи керування проектом.	сам. робота	2-3	3,5	кінець поточного тижня
13	Планування і методи керування проектом. Методи керування і планування проектом. Метод критичного шляху (CPM). Метод аналізу й оцінки проекту (PERT). Керування якістю і ризиками у проекті. Процеси удосконалення. Процеси аналізу. Процеси змін.	лекція	1, 3-4	2	кінець поточного тижня
	Робота в команді	сам. робота	2	3,5	кінець поточного тижня
14	Групова динаміка та комунікації. Групова динаміка та комунікації. Колективна робота в галузі ІТ. Поняття групи. Діловий та соціальний аспекти групи. Формальні і неформальні групи. Місце індивіда в групі. Рольова напруга і рольовий конфлікт. Командні ролі.	лекція	3	2	кінець поточного тижня
	Підготовка до співбесіди	лаб. робота	2-3	6	кінець 16-го тижня
	Соціальні стилі та поведінка людей.	сам. робота	2	3,5	кінець поточного тижня
15	Ділове спілкування. Процес комунікації. Комунікаційні ролі. Поняття карти сили. Вербальна і невербальна комунікація. Пошук роботи. Написання резюме. Підготовка до співбесіди. Підготовка до ділових перемовин.	лекція	2	2	кінець поточного тижня
	Поняття м'яких навичок.	сам. робота		3,5	кінець поточного тижня

16	Пошук роботи в ІТ на прикладі компанії GlobalLogic. Кар'єра в GlobalLogic. Освіта. GL BaseCamp. Програма стажування. Резюме та співбесіда. Проходження співбесіди. Загальне резюме.	лекція	2	2	кінець ПОТОЧНОГО ТИЖНЯ
	Успішне ділове спілкування.	сам. робота	2	3,5	кінець ПОТОЧНОГО ТИЖНЯ