

Силабус « Проектний Практикум»

№	Назва поля	Детальний контент, коментарі
1.	Назва факультету	-
2.	Рівень вищої освіти	<i>Бакалаврський</i>
3.	Код і назва спеціальності	121 «Інженерія програмного забезпечення»
4.	Тип і назва освітньої програми	Програмна інженерія
5.	Код і назва дисципліни	<i>CS.3454, CS.5170 Проектний практикум</i>
6.	Кількість ЄКТС кредитів	4
7.	Структура дисципліни (розподіл за видами та годинами навчання)	Лекції - 24 Практичні - 8 Лабораторні - 16 Консультації - 60 Самостійна робота - 58 Сем. Контроль – комбінований іспит
8.	Графік вивчення дисципліни	4, весняний семестр навчання
9.	Передумови для навчання за дисципліною	-
10.	Анотація дисципліни	Змістовий модуль 1. Поняття якості програмного забезпечення. Характер і роль стандартів інженерії програмного забезпечення. Технології розробки програмного забезпечення. Документування. Тема 1. Вступ. Вартість якості і втрати від поганої якості. Роль документування в програмному забезпеченні. Тема 2. Якість процесу та якість продукту – взаємозв'язок. Проект. Класифікація проектів Тема 3. Походження і значення стандартів програмного забезпечення. Порівняльний аналіз. Тема 4. Візуальне моделювання. UML нотація. Case-засоби проектування програмної системи. Тема 5. Повторне використання програмного забезпечення. Тема 6. Вибір технологій та шаблонів процесів при розробці програмного забезпечення. Процеси/ фази розробки проекту. Системи контролю версій. Тема 7. Документування автоматизованої системи. Визначення образу та меж програмної системи. Вимоги до програмної системи. Змістовий модуль 2. Розробка програмної системи. Характер і роль архітектурних рішень Тема 8. Поведінкові діаграми. Моделювання процесів.

		Тема 9. Процес розробки програмної системи - прототипування. Тема 10. Структурні діаграми. Проектування за допомогою мозкового штурму. Тема 11. Еволюція тривірневої архітектури.
11.	Компетентності, знання, вміння, розуміння, якими оволодіє здобувач вищої освіти в процесі навчання	Загальні компетентності (ЗК) ЗК-1. Здатність до абстрактного мислення, аналізу та синтезу. ЗК-2. Здатність застосовувати знання у практичних ситуаціях. ЗК-6. Здатність до пошуку, оброблення та аналізу інформації з різних джерел. ЗК-7. Здатність працювати в команді. ЗК-8. Здатність діяти на основі етичних міркувань. Фахові компетентності спеціальності (ФК) ФК-2. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування. ФК-3. Здатність розробляти архітектури, модулі та компоненти програмних систем. ФК-4. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами, технічним завданням та стандартами. ФК-5. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу. ФК-8. Здатність застосовувати і розвивати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення. ФК-9. Здатність оцінювати і враховувати економічні, соціальні, технологічні та екологічні чинники, що впливають на сферу професійної діяльності. ФК-10. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя. ФК-11. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення. ФК-14. Здатність до алгоритмічного та логічного мислення. Програмні результати навчання ПР-3 Знати основні процеси, фази та ітерації життєвого циклу ПЗ ПР-4 Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення. ПР-9 Знати та вміти використовувати методи та засоби

		збору, формулювання та аналізу вимог до програмного забезпечення. ПР-10 Проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування. ПР-11 Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання. ПР-14 Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення. ПР-16 Мати навички командної розробки, погодження, оформлення і випуску всіх видів програмної документації. ПР-20 Знати підходи щодо оцінки та забезпечення якості програмного забезпечення. ПР-23 Уміння документувати та презентувати результати розробки ПЗ 1. За результатом вивчення дисципліни студенти повинні: – отримати знання вимог стандартів на автоматизовані системи; – отримати вміння вести розробку ІТ-проектів в команді; – мати навички виконувати всі види проектних робіт щодо створення системи. – мати навички володіння методами та інструментальними засобами проектування системи. Таким чином, завданням дисципліни є вивчення понятійного апарату дисципліни, основних теоретичних положень і методів, формування умінь і прищеплення навичок застосування теоретичних знань для вирішення практичних і прикладних задач. знати: – життєвий цикл програмного забезпечення – різновиди, етапи; – повторне використання програмного забезпечення; – стандарти, методології розробки і супроводження автоматизованої системи; – сучасний ринок програмного забезпечення та інформаційних систем; – методи, методології та технології аналізу прикладної галузі, інформаційні потреби системи, підходи до формулювання вимог; – методи, методології та технології проектування підсистем, в тому числі архітектури і її фізичної реалізації; – класифікацію та загальні характеристики сучасних CASE-засобів; – культуру якості при розробці програмного забезпечення, якість процесу та якість продукту – взаємозв'язок; – візуальну мову моделювання UML для створення систем;
--	--	--

		вміти: – мати чітке представлення щодо життєвого циклу розробки проекту; – використовувати методи аналізу, такі як: аналіз потреб, цілей та варіантів використання для побудови вимог до будь якої програмної системи; – виявляти інформаційні потреби, розробляти вимоги до проекту; – розробляти концепт-документ до проекту; – Застосовувати мову UML для моделювання вимог до програмної системи. – Проводити аналіз ПЗ в різних професійних галузях та за обраною предметною галуззю майбутнього проекту; – створювати специфікації на автоматизовані системи з використанням UML; – Мати базові навички прототипування використовуючи Case-засоби – виконувати роботи на всіх стадіях ЖЦ проекту; – створювати інтерфейс, знати взаємодію, еволюцію програм та даних, методи інтеграції, проблеми взаємодії різномовних програм і даних у сучасних середовищах, а також методи еволюційної зміни компонентів і систем, характеристика стандарту iso/iec 11404:2007 з опису даних, незалежних мов програмування; – обирати сучасні інструментальні засоби і технології; – проводити формалізацію і реалізацію рішень прикладних задач: – розробляти проекти в прикладних галузях; – використовувати методи верифікації і базове тестування програм, визначення формального апарату специфікації, верифікації і тестування програм; – використовувати інженерію розробки програмних продуктів, базові характеристики інженерії розробки компонентів, предметної галузі готових компонентів та лінії розробки програмних продуктів, особливості сучасних середовищ для колективної розробки; – використовувати моделі якості та надійності програмних систем; моделі якості, метрики і методи виміру показників якості; класифікацію математичних моделей надійності та підходи до оцінки надійності програмного продукту за деякими моделями. – працювати у команді; володіти компетенціями: – методами програмної інженерії для реалізації програмного забезпечення з урахуванням вимог до його якості, надійності; – володіти базовими уявленнями про сучасні стандарти та процеси управління якістю ПЗ; – сучасними уявленнями про основи інженерії вимог до ПЗ;
--	--	---

		– аналізувати предметну галузь, створювати та працювати: ідентифікувати, формалізувати та класифікувати вимоги до програмних систем, - розробляти специфікацію вимог, виконувати їхню верифікацію. – моделювати різні аспекти системи, для якої створюється ПЗ, розробляти алгоритми та структури даних для програмної системи; – обґрунтовано обирати програмно-технічні засоби для розробки та супроводження системи; – навички роботи з інструментальними засобами проектування програмної системи; – створення технічної документації до програмного проекту; – розуміння особливостей виконання колективного проекту при створенні системи, які спрямовані на набуття перших навичок практичного застосування отриманих раніше знань в умовах колективної праці з використанням відповідних методів та інструментів;
12.	Результати навчання здобувача вищої освіти	1. Необхідний обсяг знань для одержання позитивної оцінки. Знати: – життєвий цикл програмного забезпечення – різновиди, етапи; – повторне використання програмного забезпечення; – стандарти, методології розробки і супроводження автоматизованої системи; – сучасний ринок програмного забезпечення та інформаційних систем; – методи, методології та технології аналізу прикладної галузі, інформаційні потреби системи, підходи до формулювання вимог; – методи, методології та технології проектування підсистем, в тому числі архітектури і її фізичної реалізації; – класифікацію та загальні характеристики сучасних CASE-засобів; – культуру якості при розробці програмного забезпечення, якість процесу та якість продукту – взаємозв'язок; – візуальну мову моделювання UML для створення систем; Необхідний обсяг умінь для одержання позитивної оцінки: – мати чітке представлення щодо життєвого циклу розробки проекту; – використовувати методи аналізу, такі як: аналіз потреб, цілей та варіантів використання для побудови вимог до будь якої програмної системи; – виявляти інформаційні потреби, розробляти вимоги до проекту; – розробляти концепт-документ до проекту; – Застосовувати мову UML для моделювання вимог до програмної системи. – Проводити аналіз ПЗ в різних професійних галузях та за

		обраною предметною галуззю майбутнього проекту; – створювати специфікації на автоматизовані системи з використанням UML; – Мати базові навички прототипування використовуючи Case-засоби – виконувати роботи на всіх стадіях ЖЦ проекту; – створювати інтерфейс, знати взаємодію, еволюцію програм та даних, методи інтеграції, проблеми взаємодії різномовних програм і даних у сучасних середовищах, а також методи еволюційної зміни компонентів і систем, характеристика стандарту iso/iec 11404:2007 з опису даних, незалежних мов програмування; – обирати сучасні інструментальні засоби і технології; – проводити формалізацію і реалізацію рішень прикладних задач: розробляти проекти в прикладних галузях; – використовувати методи верифікації і базове тестування програм, визначення формального апарату специфікації, верифікації і тестування програм; – використовувати інженерію розробки програмних продуктів, базові характеристики інженерії розробки компонентів, предметної галузі готових компонентів та лінії розробки програмних продуктів, особливості сучасних середовищ для колективної розробки; – використовувати моделі якості та надійності програмних систем; моделі якості, метрики і методи виміру показників якості; класифікацію математичних моделей надійності та підходи до оцінки надійності програмного продукту за деякими моделями. – працювати у команді;
13.	Система оцінювання відповідно до кожного завдання для складання заліку/екзамену	Критерії оцінювання роботи студента протягом семестру. Відмінно, А,В (90-100). Відпрацювати та захистити усі лабораторні роботи, КР. Досконало знати й використовувати мову моделювання UML при описі вимог до програмної системи. Застосовувати методи керування вимогами, володіти уміннями розробляти усі шаблони специфікації до програмної системи. Використовувати моделі удосконалення вимог. Чітко розуміти та використовувати фази розробки проекту в команді. Добре, С (75-89). Відпрацювати та захистити усі лабораторні роботи, КР. Володіти уміннями розробляти Vision до системи в повному обсязі. Володіти уміннями описувати та застосувати методи аналізу, такі як: аналіз потреб, цілей та варіантів використання для побудови вимог до програмної системи. Застосовувати мову моделювання UML. Чітко розуміти та використовувати фази розробки проекту в команді. Задовільно, D, E (60-74). Відпрацювати та захистити усі лабораторні роботи, КР.

		Володіти уміннями розробляти документ образу та меж проекту. Мати мінімальний рівень знання мови UML. Розуміти фази розробки проектів. Критерії оцінювання знань та вмінь студента на комбінованому іспиті. Завданням підсумкового контролю (комбінованого іспиту, заліку) є підсумкова перевірка глибини засвоєння студентом програмного матеріалу дисципліни, логіки та взаємозв'язків між окремими її розділами, здатність творчого використання набутих знань, уміння сформулювати своє ставлення до певної проблеми, що впливає зі змісту дисципліни тощо. Для визначення ступеню оволодіння навчальним матеріалом із подальшим його оцінюванням застосовуються наступні рівні досягнень студентів. Відмінно, А,В (90-100). Студент вільно володіє навчальним матеріалом на підставі вивченої основної та додаткової літератури, аргументовано висловлює свої думки, проявляє творчий підхід до виконання індивідуальних та колективних завдань при самостійній роботі. Добре, С (75-89). Студент володіє певним обсягом навчального матеріалу, здатний його аналізувати, але не має достатніх знань та вмінь для формулювання висновків, допускає несуттєві неточності. Задовільно, D, E (60-74). Студент володіє навчальним матеріалом на репродуктивному рівні або володіє частиною навчального матеріалу, уміє використовувати знання в стандартних ситуаціях. Студент володіє навчальним матеріалом поверхово й фрагментарно. Незадовільно, FX, F (1-59). Студент не володіє навчальним матеріалом.
14.	Якість освітнього процесу	Дотримання норм академічної доброчесності забезпечується заборонаю плагіату та списування. Очікується, що роботи студента вирізнятимуться оригінальністю, власними міркуваннями в межах команди. Відсутність посилань на використані джерела, фабрикування, списування, часткове чи повне запозичення матеріалу з робіт інших студентів є проявом академічної недоброчесності. Виявлення зазначених ознак у письмовій роботі студентів є підставою для її незарахування. Жодні форми порушення академічної доброчесності не толеруються.
15.	Методичне забезпечення	Використовується безкоштовний ліцензійний продукт, навчально-методичні матеріали, електронний посібник, КНМЗ з дисципліни.
16.	Розробник силабусу	Доцент кафедри ІІІ Афанасьєва Ірина Віталіївна iryna.afanasieva@nure.ua