

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення

Декан ФІТ _____

ЗАТВЕРДЖУЮ
Говорушченко Т.О.
1 вересня 2024 р.



СИЛАБУС

Навчальна дисципліна Управління проектами і якістю програмних систем
Освітньо-наукова програма Інженерія програмного забезпечення
Рівень вищої освіти Другий (магістерський)
Загальна інформація

Позиція	Зміст інформації
Викладач(і)	Яшина Оксана Миколаївна
Профайл викладача	https://ipz.khmnmu.edu.ua/yashyna-o-m/
E-mail викладача(ів)	yashynao@khmnmu.edu.ua
Контактний телефон	Заповнюється за домовленістю
Сторінка дисципліни в ІСУ	https://msn.khmnmu.edu.ua/course/view.php?id=7009
Навчальний рік	2024-2025
Консультації	Очні: четвер, 3 пара, 1-204 Онлайн: за необхідністю та попередньою домовленістю

Загальна характеристика дисципліни

Форма здобуття освіти	Курс	Семестр	Загальний обсяг	Кількість годин						Курсовий проект	Курсова робота	Форма семестрового контролю	
				Аудиторні заняття				Індивідуальна робота студента	Самостійна робота, в т.ч. ІРС			Залік	Іспит
				Разом	Лекції	Лабораторні роботи	Практичні заняття						
Очна (денна)	1	2	5	54	18	18	18		96				+

Анотація навчальної дисципліни

Дисципліна «Управління проектами і якістю програмних систем» є однією із спеціальних профілюючих дисциплін і тому займає провідне місце у підготовці магістрів інженерії програмного забезпечення. Дисципліна викладається для здобувачів другого (магістерського) рівня вищої освіти денної форми навчання спеціальностей галузі інформаційні технології.

При викладанні дисципліни використовуються активні і творчі форми проведення занять, зокрема, методи проблемного навчання. Дисципліна входить до циклу освітньо-професійної підготовки студента в галузі інженерії програмного забезпечення. Для успішного засвоєння даного курсу здобувач вищої освіти повинен мати навички самостійної роботи з різними джерелами інформації (мережа Інтернет, друковані видання, тощо), умінням

застосовувати набуті знання на практиці, володіти умінням узагальнювати інформацію, отриману з різних джерел, коректно представляти результат своєї роботи.

Мета і завдання дисципліни

Мета дисципліни. Формування системного базового представлення, належного рівня знань, умінь та навичок студентів в області управління проектами і якістю програмних систем, що об'єднує стандарти та характеристики, процеси оцінювання та процеси керування, необхідні для розробки конкурентоспроможного програмного забезпечення, які достатні для фахової діяльності та подальшого продовження освіти та самоосвіти в області інформаційних технологій.

Завдання дисципліни: надати студентам знання і практичні навички практичного вирішення прикладних задач з обґрунтованим вибором мов програмування, в тому числі мультипарадигмових.

Очікувані результати навчання

Відповідно до Стандарту вищої освіти та освітньої програми дисципліна має забезпечити:

компетентності: Здатність проводити дослідження на відповідному рівні. Здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення. Здатність розробляти і реалізовувати наукові та/або прикладні проекти у сфері інженерії програмного забезпечення. Здатність розробляти, аналізувати та застосовувати специфікації, стандарти, правила і рекомендації в сфері інженерії програмного забезпечення. Здатність розробляти і координувати процеси, етапи та ітерації життєвого циклу програмного забезпечення на основі застосування сучасних моделей, методів та технологій розроблення програмного забезпечення. Здатність забезпечувати якість програмного забезпечення.

програмні результати навчання: Знати і застосовувати сучасні професійні стандарти і інші нормативно-правові документи з інженерії програмного забезпечення. Розробляти, аналізувати, обґрунтовувати та систематизувати вимоги до програмного забезпечення. Розробляти і оцінювати стратегії проектування програмних засобів; обґрунтовувати, аналізувати і оцінювати варіанти проектних рішень з точки зору якості кінцевого програмного продукту, ресурсних обмежень та інших факторів. Забезпечувати якість на всіх стадіях життєвого циклу програмного забезпечення, у тому числі з використанням релевантних моделей та методів оцінювання, а також засобів автоматизованого тестування і верифікації програмного забезпечення. Приймати ефективні організаційно-управлінські рішення в умовах невизначеності та зміни вимог, порівнювати альтернативи, оцінювати ризики. Планувати, організувати та здійснювати тестування, верифікацію та валідацію програмного забезпечення. Вміти використовувати методи фундаментальних і прикладних дисциплін інженерії програмного забезпечення при проектуванні архітектури та розробці програмних систем із використанням поглиблених знань з оптимізації, реінжинірингу програмних систем, управління проектами та якістю з вибором парадигм програмування та застосуванням машинного навчання.

Студент, який успішно завершив вивчення дисципліни, повинен: *проводити* аналітичне дослідження параметрів функціонування програмних систем для їх валідації та верифікації; *вміло використовувати* сучасні професійні стандарти та інші нормативно-правові документи з інженерії програмного забезпечення; *вміти* приймати організаційно-управлінські рішення в умовах невизначеності; *набувати* нові наукові і професійні знання, *вдосконалювати* навички, прогнозувати розвиток програмних систем та інформаційних технологій; *знати і застосовувати* на практиці моделі та методи оцінювання та забезпечення якості програмних систем на всіх стадіях життєвого циклу.

Тематичний і календарний план вивчення дисципліни

№ тижня	Тема лекції	Тема практичного заняття*	Тема лабораторного заняття	Самостійна робота студента		
				зміст	год	література
1	2	3	4	5	6	7
1-2	Концепція інженерії якості програмних систем	Метрики цикломатичної складності по Маккейбу. Ч.1	Оптимізація програмних систем із використанням метрик. Ч.1	Пророблення теоретичного матеріалу та підготовка до лабораторної та практичної роботи		[1, 2, 3,5,7]
3-4	Методології та критерії підвищення якості програмних систем	Метрики цикломатичної складності по Маккейбу. Ч.2	Оптимізація програмних систем із використанням метрик. Ч.2	Пророблення теоретичного матеріалу та підготовка до лабораторної та практичної роботи.		[1, 2, 3,5,7]
5-6	Моделі та метрики якості програмних систем.	Метрика Джилба. Ч.1	Розробка програм оцінки складності програмних систем на	Пророблення теоретичного матеріалу та підготовка до лабораторної та практичної роботи		[1, 2, 3,5,7]

			основі окремих метрик розміру програм. Ч.1		
7-8	Побудова метрик та моделей якості.-	Метрика Джилба. Ч.2	Розробка програм оцінки складності програмних систем на основі окремих метрик розміру програм. Ч.	Пророблення теоретичного матеріалу та підготовка до лабораторної та практичної роботи	[1, 2, 3,5,7]
9-10	Управління ризиком проєктів.	Метрика Холстеда. Ч.1	Розробка програм оцінки складності програмних систем на основі окремих метрик складності потоку управління програмних систем. Ч.1	Пророблення теоретичного матеріалу та підготовка до лабораторної та практичної роботи	[1, 2, 3,5,7]
11-12	Методології підвищення якості програмних систем -	Метрика Холстеда. Ч.1	Розробка програм оцінки складності програмних систем на основі окремих метрик складності потоку управління програмних систем. Ч.2	Пророблення теоретичного матеріалу та підготовка до лабораторної та практичної роботи	-
13-14	Технології керування якістю програмних систем	Метрика Чена. Ч.1	Розробка програм оцінки складності програмних систем на метрик складності потоку даних. Ч.1	Пророблення теоретичного матеріалу та підготовка до лабораторної та практичної роботи	[1, 2, 3,5,7]
15-16	Методології оцінки якості програмних систем.	Метрика Чена. Ч.1	Розробка програм оцінки складності програмних систем на основі окремих метрик складності програм – метрик складності потоку даних. Ч. 2	Пророблення теоретичного матеріалу та підготовка до лабораторної та практичної роботи	[1, 2, 3,5,7]
17-18	Методології оцінювання зрілості організацій-розробників..	Підсумкове заняття		Пророблення теоретичного матеріалу та підготовка до лабораторної та практичної роботи	[1, 2, 3,5,7]

Примітка:* Лекції, практичні та лабораторні заняття проводяться раз у два тижні по дві та чотири години відповідно.

Політика дисципліни.

Організація освітнього процесу з дисципліни відповідає вимогам положень про організаційне і навчально-методичне забезпечення освітнього процесу, освітній програмі та навчальному плану. Студент зобов'язаний відвідувати лекції, лабораторні заняття згідно з розкладом, не запізнюватися на заняття, завдання виконувати відповідно до графіка. Пропущене лабораторне заняття студент зобов'язаний опрацювати самостійно у повному обсязі і відвітати перед викладачем не пізніше, ніж за тиждень до чергової атестації. До лабораторних занять студент має підготуватися за відповідною темою і проявляти активність. Набуті особою знання з дисципліни або її окремих розділів у неформальній освіті зараховуються відповідно до Положення про порядок перезарахування результатів навчання у ХНУ (<https://www.khnu.km.ua/root/files/01/10/03/006.pdf>).

Критерії оцінювання результатів навчання.

Кожний вид роботи з дисципліни оцінюється за **чотирибальною** шкалою. Семестрова підсумкова оцінка визначається як середньозважена з усіх видів навчальної роботи, виконаних і зданих позитивно з врахуванням коефіцієнта вагомості. Вагові коефіцієнти змінюються залежно від структури дисципліни і важливості окремих її видів робіт. При оцінюванні знань студентів використовуються різні засоби контролю, зокрема: усне опитування перед допуском до виконання лабораторної роботи – здійснюється на її початку; засвоєння теоретичного матеріалу з тем перевіряється тестовим контролем; якість виконання, набуття теоретичних знань і практичних навичок

перевіряється шляхом захисту кожної лабораторної роботи згідно з робочою програмою дисципліни і робочим навчальним планом.

Структурування дисципліни за видами робіт і оцінювання результатів навчання студентів денної форми навчання у семестрі за ваговими коефіцієнтами

Аудиторна робота				Самостійна, індивідуальна робота				Іспит		
<i>Другий семестр</i>										
Лабораторні роботи №:				Практичні роботи №				Контроль:		іспит
1	2	3	4	1	2	3	4	ТК Т1-2	ТК Т 3-4	
ВК*:		0,2		0,2				0,2		0,4

Умовні позначення: Т – тема дисципліни; ВК – ваговий коефіцієнт; ТК – тестовий контроль.

Оцінювання тестових завдань. Тематичний тест для кожного студента складається з двадцяти тестових завдань, кожне з яких оцінюється одним балом. Максимальна сума балів, яку може набрати студент, складає 20. На тестування відводиться 20 хвилин. Тестування проводиться з використанням модульного середовища для навчання MOODLE.

Оцінювання здійснюється за чотирибальною шкалою.

Відповідність набраних балів за тестове завдання оцінці, що виставляється студенту, представлена у нижченаведеній таблиці.

Сума балів за тестове завдання	1–11	12–14	15–18	19-20
Оцінка	2	3	4	5

Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ЄКТС	Інституційна інтервальна шкала балів	Вітчизняна оцінка, критерії	
A	4,75–5,00	5	Відмінно – глибоке і повне опанування навчального матеріалу і виявлення відповідних умінь та навиків
B	4,25–4,74	4	Добре – повне знання навчального матеріалу з кількома незначними помилками
C	3,75–4,24	4	Добре – в загальному правильна відповідь з двома-трьома суттєвими помилками
D	3,25–3,74	3	Задовільно – неповне опанування програмного матеріалу, але достатнє для практичної діяльності за професією
E	3,00–3,24	3	Задовільно – неповне опанування програмного матеріалу, що задовольняє мінімальні критерії оцінювання
FX	2,00–2,99	2	Незадовільно – безсистемність одержаних знань і неможливість продовжити навчання без додаткових знань з дисципліни
F	0,00–1,99	2	Незадовільно – необхідна серйозна подальша робота і повторне вивчення дисципліни

Питання для підсумкового контролю з дисципліни

1. Вказати основні компоненти поняття управління якістю програмних систем.
2. Для чого, коли та ким було введено кодекс етики та професійної практики інженерії програмного забезпечення?
3. Вказати основні положення Кодексу етики та професійної практики програмної інженерії, в яких йде мова про якість.
4. Дати стандартизоване визначення поняття якості програмного забезпечення.
5. Сформулювати формалізоване визначення якості програмних систем.
6. Охарактеризуйте структурну модель якості програмних систем.
7. Описати багаторівневу структуру характеристик якості програмних систем (ПС).
8. Які класи якості виділені в стандартах ISO/IEC?
9. Перерахувати та охарактеризувати фактори якості програмних систем.
10. Описати схему структурної моделі якості Бома.

11. Які задачі необхідно вирішувати під час проектування структурно-функціональної моделі якості ПС?
12. Охарактеризуйте поняття показники якості.
13. Перерахувати види функцій агрегації показників якості.
14. Приведіть способи кодування характеристик якості ПС.
15. Привести графічну схему процесу оцінювання якості ПС.
16. Перерахувати та коротко охарактеризувати основні стандартизовані процеси SQM, орієнтовані на оцінювання ПС як результату розробки та співставити їх із життєвим циклом ПС.
17. Що включають в планування якості ПС? Навести приклад плану. Запропонувати способи автоматизації планування якості ПС.
18. Охарактеризувати процеси верифікації та атестації ПС. Які способи автоматизації цих процесів можна запропонувати?
19. Охарактеризуйте процеси спільного аналізу ПС та приведіть варіанти їх автоматизації.
20. Чим відрізняються управлінські оцінки ПС від технічних?
21. Співставити процес інспекції та процес прогонки ПС. Які рішення по автоматизації цих процесів доцільні?
22. Сформулюйте основні аспекти управління якістю ПС.
23. Навести приклад архітектури системи автоматизації управління якістю ПС на основі формалізованих моделей.
24. Концепція та сутність управління якістю ПС.
25. Що таке комплексний показник якості?
26. Що таке базове значення показника якості?
27. Що відображають характеристики якості ПС?
28. Що називають фактором якості?
29. Дати визначення метрики оцінки якості ПС.
30. Що таке оціночний елемент якості ПС?
31. Яка роль стандартизації в управлінні якістю ПС?
32. Які функції виконує стандартизація?
33. Вказати на важливість стандартів при управлінні якістю ПС.
34. Типові етапи в розробці систем.
35. Визначити склад метрик та їх значення в ефективності управління ПС.
36. Які існують способи захисту ПС від копіювання?
37. Вказати способи захисту даних від копіювання.
38. Від чого здійснюється захист даних?
39. Вказати градацію програмних продуктів із ціллю захисту даних.
40. Охарактеризувати захист на стадії виконання.
41. Важливість авторського права та його використання при управлінні якістю ПС.
42. Що означає характеристика якості програмного продукту?
43. Вказати основні напрямки застосування метрик.
44. Використання метричних шкал.
45. Дати визначення розмірно-орієнтованим метрикам.
46. Вказати недоліки розмірно-орієнтованих метрик.
47. Використання метрик цикломатичної складності.
48. Використання метрик складності потоку даних.
49. Розробка метрик стилістики та зрозумілості програм.
50. Вказати суть метрики коректності програм.

Методичне забезпечення

Навчальний процес з дисципліни забезпечений необхідними навчально-методичними розробками в модульному середовищі.

Рекомендована література.

1. Daniel Galin Software Quality Concepts and Practice / USA.: Wiley-IEEE Press, 2020. – 711 p.
2. Тудоран В. А. Людино-машинна взаємодія : навч. посіб. / В. А. Тудоран; Нац. ун-т кораблебудування ім. адмірала Макарова. Електронний ресурс - Миколаїв : НУК, 2015. - 162 с.
3. І. Бородкіна, Г. Бородкін Інженерія програмного забезпечення. Посібник для студентів вищих навчальних закладів. – К.: Центр навчальної літератури, 2020. –204 с.
4. Г. Галісеєв Системне програмування. – В. : Ун. Україна. – 2019. – 113 с.
5. О. Перевозчикова Інформаційні системи і структури даних. – К. : К-М Академія. – 2021. – 288 с.
6. Д. Сідоров, І. Казак Основи інженерних розрахунків на ПЕОМ. Програмування алгоритмічною мовою Фортран. – В: ЦНЛ. – 186 с.
7. Томас Кормен, Чарльз Лейзерсон, Рональд Рівест, Кліффорд Штайн Алгоритми: побудова та аналіз. Том 2. – К.: Діалектика. – 2020. – 716 с.
8. Говорущенко Т. О. Методологія оцінювання достатності інформації для визначення якості програмного забезпечення: монографія. Хмельницький: Хмельницький національний університет, 2017. 310 с.

9. Бевз О. М. Проектування програмних засобів систем управління: навчальний посібник / О. М. Бевз, В. М. Папінов, Ю. А. Скидан. – Вінниця: ВНТУ, 2020. – 125 с.
10. Лаврищева Е. М. Software Engineering комп'ютерних систем. Парадигми, технології та CASE-засоби програмування. – Київ: Наукова думка, 2020. – 283 с.
11. Говорущенко Т. О. Аналіз, дослідження та оцінювання програмних систем: навчальний посібник. – Хмельницький: Хмельницький національний університет, 2019. 358 с.

Розробники:



к.т.н., доцент Яшина О.М

Погоджено:

Зав. кафедри ІПЗ



д.ф-м. н., проф. Бедратюк Л.П.

Гарант ОП



к.т.н., доцент Яшина О.М.