



РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Управління проектами і якістю програмних систем

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Рівень вищої освіти – другий (магістерський)

Освітньо-професійна програма – Інженерія програмного забезпечення

Шифр дисципліни – ОІП104

Статус дисципліни – обов'язкова

Факультет – Програмування та комп'ютерних і телекомунікаційних систем

Кафедра – інженерії програмного забезпечення

Форма навчання	Курс	Семестр	Обсяг дисципліни	Кількість годин							Форма семестрового контролю		
				Аудиторні заняття							Курсова робота	Залік	Іспит
				Кредити ЄКТС	Разом	Лекції	Лабораторні роботи	Практичні заняття	Індивідуальна робота студента	Самостійна робота, в т.ч. ІРС			
Д	5	2	5	150	18	18	18		96			+	
Разом ДФН			5	150	18	18	18		96			1	

Робоча програма складена на основі освітньо-професійної програми

Програма складена  Яшина О.М.

Схвалена на засіданні кафедри ПІЗ

Протокол № 1 від 31 серпня 2021 р.

Зав. кафедри ПІЗ  Бедратюк Л.П.
Підпис Ініціал прізвище

Робоча програма розглянута та схвалена Вченою радою факультету програмування та комп'ютерних і телекомунікаційних систем

Голова Вченої ради  Савенко О.С.
Підпис Ініціал прізвище

Хмельницький 2021

УПРАВЛІННЯ ПРОЕКТАМИ І ЯКІСТЮ ПРОГРАМНИХ СИСТЕМ

Тип дисципліни
Рівень вищої освіти
Мова навчання

Обов'язкова
Другий (магістерський)
Українська

Семестр

другий

Форми навчання, для яких викладається дисципліна

Денна

Результати навчання Відповідно до Стандарту вищої освіти із зазначеної спеціальності 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. Здатність виконувати проектування програмних систем та їх складових компонентів із використанням поглиблених знань з оптимізації, реінжинірингу програмних систем, управління проектами та якістю з вибором парадигм програмування та застосування машинного навчання.

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, організовувати 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 застосування машинного навчання.

Зміст навчальної дисципліни. Концепція інженерії якості програмних систем. Методології та критерії підвищення якості програмних систем. Моделі та метрики якості програмних систем. Побудова метрик та моделей якості. Управління проектами. Управління ризиком проектів. Методології підвищення якості програмних систем. Технології керування якістю програмних систем. Методології оцінки якості програмних систем. Методології оцінювання зрілості організацій-розробників.

Запланована навчальна діяльність: лекції – 18 год., лабораторні заняття – 18 год., практичні заняття – 18; самостійна робота – 96 год., разом – 150 год.

Методи навчання: Методи проблемного викладання і візуалізації пояснювально-ілюстративні, частковопошукові, дослідницькі, самостійна робота.

Форми і методи оцінювання результатів навчання усне опитування, захисти лабораторних та практичних робіт

Вид семестрового контролю: іспит

Навчальні ресурси:

1. Daniel Galin Software Quality Concepts and Practice / USA.: Wiley-IEEE Press, 2018. – 711 p.
2. Дрейзин, В.Э. Управление качеством электронных средств / В.Э. Дрейзин. – М.: Academia, 2018. – 44 с.
3. 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.
4. Claude Y. LaporteAlain Software Quality Assurance, First Edition/ USA: Wiley-IEEE Press, 2017. – 596 p.
5. Mark A. Levin, Ted T. Kalal, Jonathan Rodin Software Quality Analysis Techniques / USA: Wiley-IEEE Press, 2019. – 447 p.
6. Giuliana Carullo Code Reviews 101 / USA: Leanpub, 2018. – 175 p. 7. Аналіз та опрацювання метрик якості програмного забезпечення на етапі проектування // Вісник ХНУ. – 2010, №1.
8. Модульне середовище для навчання MOODLE. Доступ до ресурсу: <https://msn.khnu.km.ua>.
9. Електронна бібліотека університету. Доступ до ресурсу: http://lib.khnu.km.ua/asp/php_f/page_lib.php.

Викладач: канд. техн. наук, доцент, Яшина О.М.

3. ПОЯСНЮВАЛЬНА ЗАПИСКА

Дисципліна «Управління проектами і якістю програмних систем» є однією із фахових дисциплін і займає провідне місце у підготовці фахівців освітнього рівня «магістр» за спеціальністю 121 «Інженерія програмного забезпечення» за освітньо-професійною програмою «Інженерія програмного забезпечення».

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 на характеристики, процеси оцінювання та процеси керування, необхідні для розробки конкурентоспроможного програмного забезпечення, які достатні для фахової діяльності та подальшого продовження освіти і самоосвіти в області обчислювальної техніки, інформаційних систем різного призначення.

Предмет дисципліни:

Предметом вивчення дисципліни є стандарти та характеристики процесу керування якістю програмних систем. Об'єктом вивчення є процес управління якістю програмних систем із врахуванням сучасних стандартів та методологій. **Завдання дисципліни:**

Надати студентам знання і практичні навички з управління проектами та якістю програмних систем, а також сформувати компетентності, необхідні для управління проектами та якістю. 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.

Відповідно до Стандарту вищої освіти дисципліна має забезпечити: **компетентності** – ЗК-3. Здатність проводити дослідження на відповідному рівні. ФК-1. Здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення. ФК-2. Здатність розробляти і реалізовувати наукові та/або прикладні проекти у сфері інженерії програмного забезпечення. ФК-5. Здатність розробляти, аналізувати та застосовувати специфікації, стандарти, правила і рекомендації в сфері інженерії програмного забезпечення. ФК-8. Здатність розробляти і координувати процеси, етапи та ітерації життєвого циклу програмного забезпечення на основі застосування сучасних моделей, методів та технологій розроблення програмного забезпечення. ФК-9. Здатність забезпечувати якість програмного забезпечення. ФК-10 Здатність виконувати проектування програмних систем та їх складових компонентів із використанням поглиблених знань з оптимізації, реінжинірингу програмних систем, управління проектами та якістю з вибором парадигм програмування та застосування машинного навчання.

програмні результати навчання: ПРН-1 Знати і застосовувати сучасні професійні стандарти і інші нормативно-правові документи з інженерії програмного забезпечення. ПРН-5 Розробляти, аналізувати, обґрунтовувати та систематизувати вимоги до програмного забезпечення. ПРН-6 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. ПРН-11 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. ПРН 12 Приймати

ефективні організаційно-управлінські рішення в умовах невизначеності та зміни вимог, порівнювати альтернативи, оцінювати ризики ПРН-16 Планувати, організувати та здійснювати тестування, верифікацію та валідацію програмного забезпечення. ПРН-18. 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.

4. СТРУКТУРА ЗАЛІКОВИХ КРЕДИТІВ ДИСЦИПЛІНИ

Назва теми	Кількість годин, відведених на:			
	Лекції	Лаб. роботи	Практичні роботи	СРС
Тема 1. Інженерія якості програмних систем.	4	4	4	24
Тема 2. Моделі та метрики якості програмних систем.	2	4	4	24
Тема 3. Управління проектами.	4	4	4	24
Тема 4. Методології та технології управління якістю програмних систем.	8	6	6	24
Разом	18	18	18	96

5. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

5.1. Зміст лекційного курсу

№ лекції	Перелік тем лекцій, їх анотації	Кількість годин
1	2	3
1	<p>Тема 1. Інженерія якості програмних систем. <i>Лекція 1. Концепція інженерії якості програмних систем.</i> Стандартизація розробки програмних систем: професійні стандарти та нормативно-правова документація з інженерії програмного забезпечення. Інфраструктура розробки. Архітектура процесів життєвого циклу. Парадигми програмування та якість. Інженерія процесів розробки. Інтеграція процесів життєвого циклу. Керування проектами. Аспекти визначення якості. Процес вимірювання при керуванні проектами. Взаємозв'язок понять у парадигмі якості. Літ.: [1] с.8-24.</p>	2
2	<p>Тема 1. Інженерія якості програмних систем. <i>Лекція 2. Методології та критерії підвищення якості програмних систем на всіх етапах життєвого циклу.</i> Застосування процесів контролю якості. Використання прогнозування при керуванні проектами. Керування ризиком в проекті. Вдосконалення процесів життєвого циклу. Підвищення зрілості організації. Керування якістю та впровадження системи якості. Літ.: [1] с.9-24, [3] с.17-24</p>	2
3	<p>Тема 2. Моделі та метрики якості програмних систем. <i>Лекція 3. Моделі та метрики якості програмних систем.</i> Метрика як основа виміру. Класифікація мір якості. Класифікація метрик якості. Узагальнена модель якості. Метрики в узагальненій моделі якості. Інші ієрархічні та неієрархічні моделі якості програмних систем. Байєсівський підхід до моделювання якості. Графічні моделі якості. Байєсівські мережі. Побудова метрик та моделей якості. Проектування метрик якості. Підготовка до використання метрик якості у вимірюваннях. Специфікація вимог до якості. Інші аспекти використання метрик. Модель QFD: компоненти та матриця. Методологія QFD для програмних систем. Методологія вимірювання в парадигмі: «Ціль-Питання-Міра». Принципи вимірювання та послідовність дій. Літ.: [1] с.276-326, [8]</p>	2
4	<p>Тема 3. Управління проектами. <i>Лекція 4. Управління проектами.</i> Основи управління проектами. Методичні основи планування проекту Техніко-економічне обґрунтування проекту. Контроль за виконанням проекту. Оцінка ефективності проектної діяльності. Основи проектного фінансування Організаційні форми управління проектами. Управління ресурсами проекту. Управління якістю проектів. Управління проектною командою.</p>	2

5	<p>Тема 3. Управління проектами. <i>Лекція 5. Управління ризиком проектів.</i> Проектна діяльність як сфера високого ризику. Елементи керування ризиком SEI. Таксономія ризику. Методологія оцінювання ризику. Безперервне керування ризиком. Колективне керування ризиком. Вимоги до процесу керування ризиком. Різні етапи керування ризиком. Функції, методи та засоби керування ризиком проекту. Ідентифікація ризику. Аналіз ризику. Планування, облік, контроль, регулювання ризику. Методи та засоби комунікації.</p>	2
	<p>Літ.: [1] с.169-189, [2] с.14-140.</p>	
6	<p>Тема 4. Методології та технології управління якістю програмних систем. <i>Лекція 6. Методології підвищення якості програмних систем.</i> Цикл керування якістю. Підхід до керування якістю за системою Ф. 2 Тейлора. Підхід до керування якістю в новій парадигмі. Статичне керування процесами. Загальне керування якістю програмних систем. Філософія TQM. Метод керування Е. Демінга.</p>	
	<p>Літ.: [3] с.25-92, [4, 5]</p>	
7.	<p>Тема 4. Методології та технології управління якістю програмних систем. <i>Лекція 7. Технології керування якістю програмних систем.</i> Гнучкі технологічні лінії. Методи покращення якості QIP та IDEAL. Фабрика досвіду. Уніфікований процес розробки RUP: характеристика та варіанти адаптації. Методологія підготовки програмних рішень MSF. 2 Модель процесів MSF, модель проектної групи MSF. Дисципліна керування ризиками, дисципліна керування проектами. Дисципліна керування професійною підготовкою. Процеси розробки в Agileметодологіях. Екстремальне програмування. Порівняння Agileметодологій. Порівняння методологій покращення процесів розробки. Літ.: [7] с.305-330.</p>	
8.	<p>Тема 4. Методології та технології управління якістю програмних систем. <i>Лекція 8. Методології оцінки якості програмних систем.</i> Технологічні модулі оцінювання. Оцінка інтегрального показника якості. Оцінювання процесів життєвого циклу. Еталонна модель оцінювання. Сумісна модель оцінювання. Вимоги до оцінювання. Етапи процесу оцінювання. Літ.: [7] с.305-330.</p>	2
9	<p>Тема 4. Методології та технології управління якістю програмних систем. <i>Лекція 9. Методології оцінювання зрілості організацій-розробників.</i> Моделі зрілості. Рівні зрілості процесу програмної інженерії по СММ. Методи оцінювання зрілості процесу по моделі СММ. Вибір організацій-виконавців програмних проєктів. Інтеграція моделей СММ. Сертифікація систем менеджменту якості. Стандарти для побудови та перевірки систем менеджменту якості. Ключові аспекти підвищення якості програмних систем. Узгодженість усіх процесів життєвого циклу програмних систем. Літ.: [7] с.355-370.</p>	2
	Разом	18

5.2. Зміст лабораторних занять

№ з/п	Тема лабораторного заняття	Кількість годин
1	Лабораторна робота №1. Частина 1 Оптимізація програмних систем із використанням метрик Літ.: [1, 2, 3,5]	2
2	Лабораторна робота №1. Частина 2 Оптимізація програмних систем із використанням метрик Літ.: [1, 2, 3,5]	2
3	Лабораторна робота №2. Частина 1 Розробка засобів оцінки складності програмних систем на основі окремих метрик розміру програм Літ.: [1, 2, 3,5]	2
4	Лабораторна робота №2. Частина 2 Розробка програм оцінки складності програмних систем на основі окремих метрик розміру програм Літ.: [1, 2, 3,5]	2
5	Лабораторна робота №3. Частина 1 Розробка засобів оцінки складності програмних систем на основі окремих метрик складності потоку Літ.: [1, 2, 3,5,7]	2
6	Лабораторна робота №3. Частина 2 Розробка засобів оцінки складності програмних систем на основі окремих метрик складності потоку Літ.: [1, 2, 3,5,7]	2
7	Лабораторна робота №4. Частина 1 Розробка засобів оцінки складності програмних систем на основі окремих метрик складності програм Літ.: [1, 2, 3,5,7]	2
8	Лабораторна робота №4. Частина 2 Розробка засобів оцінки складності програмних систем на основі окремих метрик складності програм Літ.: [1, 2, 3,5,7]	2
9	Лабораторна робота №4. Частина 3 Розробка засобів оцінки складності програмних систем на основі окремих метрик складності програм Літ.: [1, 2, 3,5,7]	2
Разом за семестр		18

5.3. Зміст практичних занять

№ з/п	Тема лабораторного заняття	Кількість годин
-------	----------------------------	-----------------

1	Практична робота №1. Частина 1 Метрики цикломатичної складності по Маккейбу Літ.: [1, 2, 3,5,7]	2
2	Практична робота №1. Частина 2 Метрики цикломатичної складності по Маккейбу Літ.: [1, 2, 3,5,7]	2
3	Практична робота №2. Частина 1 Метрика Джилба Літ.: [1, 2, 3,5,7]	2
4	Практична робота №2. Частина 2 Метрика Джилба Літ.: [1, 2, 3,5,7]	2
5	Практична робота №3. Частина 1 Метрика Холстеда Літ.: [1, 2, 3,5,7]	2
6	Практична робота №3. Частина 2 Метрика Холстеда Літ.: [1, 2, 3,5,7]	2
7	Практична робота №4. Частина 1 Метрика Чена Літ.: [1, 2, 3,5,7]	2
8	Практична робота №4. Частина 2 Метрика Чена Літ.: [1, 2, 3,5,7]	2
9	Підсумкове заняття	2
	Разом за семестр	18

5.4 Зміст самостійної роботи

Самостійна робота студентів *денної* форми навчання полягає у систематичному опрацюванні програмного матеріалу, підготовці до виконання і захисту лабораторних робіт, тестування з теоретичного матеріалу, виконанні індивідуальних завдань, в тому числі курсового проекту, тощо.

Номер тижня	Зміст самостійної роботи	Кількість годин
1	2	3
1	Опрацювання лекційного матеріалу, підготовка до виконання лабораторної роботи № 1.	2
2	Опрацювання лекційного матеріалу Підготовка до захисту лабораторної роботи №1	2
3	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи №2. Опрацювання довідкової літератури.	2
4	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи № 2	2

5	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи № 3. Опрацювання довідкової літератури.	2
6	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи № 3. Опрацювання довідкової літератури.	2
7	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи № 4. Опрацювання довідкової літератури.	2
8	Опрацювання лекційного матеріалу та довідкової літератури. Підготовка до ТК1.	4
9	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи № 4. Опрацювання довідкової літератури.	8
10	Опрацювання лекційного матеріалу. Підготовка до виконання практичної роботи № 1. Опрацювання довідкової літератури.	8
11	Опрацювання лекційного матеріалу. Підготовка до виконання практичної роботи №2. Опрацювання довідкової літератури.	8
12	Опрацювання лекційного матеріалу. Підготовка до захисту практичної роботи № 1. Опрацювання довідкової літератури.	8
13	Опрацювання лекційного матеріалу. Підготовка до захисту практичної роботи № 2. Опрацювання довідкової літератури.	8
14	Опрацювання лекційного матеріалу. Підготовка до виконання практичної роботи № 3. Опрацювання довідкової літератури.	8
15	Опрацювання лекційного матеріалу. Підготовка до захисту практичної роботи №3. Опрацювання довідкової літератури.	8
16	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи №4. Опрацювання довідкової літератури.	8
17	Опрацювання лекційного матеріалу. Підготовка до захисту практичної роботи №4. Підготовка до здачі ТК2.	8
Разом:		96

6. МЕТОДИ НАВЧАННЯ

Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних методів: методи проблемного викладання і візуалізації пояснювально-ілюстративні, частковопошукові, дослідницькі, самостійна робота. Всі заняття проводяться з використанням інформаційних технологій і мають за мету – формування системного базового представлення, належного рівня знань, умінь та навичок студентів в області управління проектами та якістю програмних систем, що об'єднує стандарти на характеристики, процеси оцінювання та процеси керування, необхідні для розробки конкурентоспроможного

програмного забезпечення, які достатні для фахової діяльності та подальшого продовження освіти і самоосвіти в області інженерії програмного забезпечення.

7. ФОРМИ І МЕТОДИ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Поточний контроль здійснюється під час лекційних та лабораторних занять, а також у дні проведення контрольних заходів, встановлених робочим планом дисципліни. Семестровий контроль проводиться у формі іспиту. При цьому при виведенні остаточної оцінки враховуються результати поточного контролю та письмового іспиту.

Використовуються наступні методи поточного контролю:

- усне опитування студентів перед допуском до лабораторного заняття;
- допуск до захисту лабораторних робіт;
- захист лабораторних робіт;
- тестування та перевірка працездатності створеного у ході виконання лабораторної роботи програмного, технічного та іншого забезпечення;

Підсумкова оцінка виставляється з урахуванням коефіцієнтів вагомості.

Студент, який не допущений до захисту або не захистив лабораторні роботи, вважається невстигаючим.

Процес оцінювання підготовленості студента можна розділити на етапи:

Перший етап оцінювання направлений на визначення знань інформаційного мінімуму. Якщо студент твердо засвоїв визначену навчальним планом суму формальних знань, то це означає, що він вміє використати їх при вирішенні різних питань при проектуванні виробів із шкіри, вміє розширити їх.

Перед вивченням дисципліни, як правило, проводиться вхідний контроль знань з дисциплін, що їй передують і забезпечують. При цьому необхідно встановити рівні та критерії сформованості знань щодо змісту навчальних елементів. Такими рівнями є:

Ознайомчо-орієнтовний (ОО) – особа має орієнтовне уявлення щодо понять, які вивчаються, здатна: відтворювати формулювання визначень поняття мобільних платформ; орієнтуватись в методиках розробки програмного продукту; знати теоретичні основи створення додатків для мобільних платформ.

Понятійно-аналітичний (ПА) – особа має чітке уявлення щодо навчального об'єкту, здатна здійснювати смислове виділення, пояснення вибору конструктивних рішень щодо створення програмного продукту, методик їх проектування, їх обґрунтування на основі аналізу існуючих. Може чітко визначити структуру програм при їх проектуванні в залежності від їх використання, тобто здатна перенести раніше засвоєнні знання на типові ситуації.

Продуктивно-синтетичний (ПС) – особа має глибоке розуміння щодо навчального об'єкту, здатна здійснювати синтез, генерувати нові ідеї та уявлення, переносити раніше засвоєнні знання на нетипові, нестандартні ситуації. Тобто на цьому рівні студент повинен на основі теоретичних знань вміти проектувати різні види програмного продукту за найбільш поширеними методиками, вносити свої пропозиції щодо удосконалення способів проектування та одержання додатків для мобільних платформ; створювати конкурентоспроможні програмні додатки на світовому ринку.

При оцінюванні знань студентів використовуються різні засоби контролю, зокрема:

-допуск до виконання лабораторної роботи здійснюється на її початку усним опитуванням студента;

-якість виконання лабораторної роботи оцінюється шляхом перевірки виконаної лабораторної роботи у робочому файлі та в робочому зошиті;

-набуття теоретичних знань і практичних навичок перевіряється шляхом захисту кожної лабораторної роботи;

-засвоєння теоретичного матеріалу перевіряється тестовим контролем.

Оцінка, яка виставляється за лабораторне заняття, складається з таких елементів: усне опитування студентів перед допуском до виконання лабораторної роботи; знання теоретичного матеріалу з теми; якість оформлення результатів лабораторної роботи у зошиті та в робочому файлі; вміння студента обґрунтовувати прийняті рішення та робити висновки; своєчасний захист лабораторної роботи.

Термін захисту лабораторної роботи вважається своєчасним, якщо студент захистив її на наступному після виконання роботи занятті. За несвоєчасний захист лабораторної роботи (без поважної причини) захист лабораторної роботи оцінюється у три бали.

Пропущене з поважної причини лабораторне заняття студент повинен відпрацювати в лабораторіях кафедри в установленій викладачем термін.

Кожний вид роботи оцінюється за чотирибальною шкалою. Підсумкова оцінка з дисципліни виставляється як середньозважена з усіх видів робіт. Вагові коефіцієнти (ВК) видів робіт дисципліни приведено в таблицях 1-2.

Таблиця 1 - 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

Оцінка роботи студента за традиційною системою

Оцінка *"відмінно"* виставляється студентові, який вільно володіє матеріалом, чітко, грамотно, логічно, повно і послідовно висвітлює питання, демонструє знання не лише лекційного матеріалу, але й результати самостійного опрацювання літератури, вміння висвітлювати власні міркування та робити висновки.

Найвища оцінка ставиться студенту, який показав всебічне та глибоке знання навчально-програмного матеріалу, розбирається в поняттях, принципах та методах аналізу вимог до ПЗ, а також: який засвоїв основну і знайомий з додатковою літературою, засвоїв основні поняття дисципліни; дав повні і точні відповіді на всі запитання з поясненнями і висновками; проявив творчі здібності в розумінні, викладенні і використанні навчального матеріалу.

Оцінка *"добре"* виставляється студентові, який добре володіє навчальним матеріалом, грамотно і логічно викладає його в усній чи письмовій формі, допускаючи при цьому не більше одного-двох недоліків чи незначних помилок, Обмежується викладом основних теоретичних положень, не володіє матеріалом додаткової літератури, хоча загалом має тверді знання.

"Задовільно" отримує студент, що показав знання основного матеріалу в обсязі, необхідному для подальшого навчання (основні поняття, принципи, закономірності), а також: знайомий з основною літературою; при відповідях на запитання відчуває себе невпевнено, допускає незначні помилки; при допусканні погрішностей у відповідях володіє знаннями для їх виправлення під керівництвом викладача.

Оцінка *"незадовільно"* виставляється студентові, який демонструє не зовсім повні знання курсу навчальної дисципліни чи її окремої теми, допускає при викладанні навчального матеріалу суттєві помилки, не орієнтується у лекційному матеріалі та матеріалі основної і додаткової літератури.

"Незадовільно" ставиться студентові, який має пробіли в знаннях основного матеріалу, а також: допустив принципові помилки, не розуміє основних концептуальних положень дисципліни; на запитання не дає задовільних відповідей, не справляється з рішенням задач; не може продовжувати навчання без додаткових занять з дисципліни.

Якщо студент отримав негативну оцінку, то він має перездати її в установленому порядку, але обов'язково до терміну наступного контролю. У випадку, коли студент не виконав індивідуальний план з дисципліни у заплановані терміни без поважних причин, то під час відпрацювання заборгованості при позитивній відповіді йому виставляється оцінка „задовільно”.

Підсумкова семестрова оцінка за національною шкалою і шкалою ЄКТС встановлюється в автоматизованому режимі після внесення усіх оцінок до електронного журналу. Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС наведені у табл. 2.

Залік виставляється, якщо середньозважений бал, який отримав студент з дисципліни, знаходиться в межах від 3,00 до 5,00 балів. При цьому за вітчизняною шкалою ставиться «зараховано», а за шкалою ЄКТС – буквене позначення оцінки, що відповідає набраній студентом кількості балів.

Перехід від вітчизняної шкали оцінювання до європейської (ECTS) представлено у таблиці 2:

Таблиця 2 - Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ECTS	Бали	Вітчизняна система	
A	4,75-5,00	5	ВІДМІННО - глибоке і повне опанування навчального матеріалу і виявлення відповідних умінь та навичок
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 двоматрьома 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

8. ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ РЕЗУЛЬТАТІВ НАВЧАННЯ

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 9126-2, 9126-3, 9126-4? 9. Перерахувати та охарактеризувати фактори якості програмних систем.

10. Описати схему структурної моделі якості Боема.
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 валідації 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.

9. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Навчальний процес з дисципліни повністю і в достатній кількості забезпечений необхідною навчально-методичною літературою в модульному середовищі.

10. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

1. Daniel Galin Software Quality Concepts and Practice / USA.: Wiley-IEEE Press, 2018. – 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. - Бібліогр.: с. 161-162.
3. Мірських Г.О. Комбіновані методи визначення вагових коефіцієнтів в задачах оптимізації та оцінювання якості об'єктів [Електронний ресурс] /Г.О. Мірських, Ю.Ю. Реутська // Вісник Національного технічного університету України «Київський політехнічний інститут». Серія: Радіотехніка. Радіоапаратобудування . – 2011. – Вип. 47. – С. 199-211.
4. Коцовський В.М. Супровід програмних систем: Методичний посібник для студентів спеціальності «Інженерія програмного забезпечення» / В.М. Коцовський. – Ужгород: Видавництво УжНУ «Говерла», 2016. – 52с.
5. Claude Y. LaporteAlain Software Quality Assurance, First Edition/ USA: Wiley-IEEE Press, 2017. – 596 p.
6. Mark A. Levin, Ted T. Kalal, Jonathan Rodin Software Quality Analysis Techniques / USA: WileyIEEE Press, 2019. – 447 p.
7. Giuliana Carullo Code Reviews 101 / USA: Leanpub, 2018. – 175 p.
8. Steven Shaffer A Brief Introduction to Software Development and Quality Assurance Management / USA: S. C. Shaffer Publications, 2013. – 25 p.
9. Youness Abdelaziz, Habib Hicham Comparative Study of Software Quality Models / IJCSI International Journal of Computer Science Issues, 2016. – 155 p.
10. Ronald Cummings-John, Owais Peer Leading Quality: How Great Leaders Deliver High-Quality Software and Accelerate Growth / USA: ROI Press, 2019. – 134 p.
11. Anirban Basu Software Quality Assurance, Testing and Metrics/ USA: PHI Learning Pvt. Ltd., 2015. – 288 p.
12. ASQ Quality Press TQM: Introduction to and Overview of Total Quality Management / USA: ASQ Quality Press, 2013. – 22 p.
13. Gerald M. Weinberg CHANGE: Planned & Unplanned / USA: Weinberg & Weinberg, 2012. – 193 p.
14. Програмна інженерія. Якість продукту. Частина 1. Модель якості (ISO/IEC 9126-1:2001, IDT): ДСТУ ISO/IEC 9126-1:2013 – [Чинний від 2014-07-01]. – К.: МІНЕКОНОМРОЗВИТКУ України, 2014. – 20 с. –(Національний стандарт України).
15. О. Перевозчикова Інформаційні системи і структури даних. – К. : К-М Академія. – 2011. – 288 с.
16. Д. Сідоров, І. Казак Основи інженерних розрахунків на ПЕОМ. Програмування алгоритмічною мовою Фортран. – В: ЦНЛ. – 186 с.

17. Говорущенко Т. О. Методологія оцінювання достатності інформації для визначення якості програмного забезпечення: монографія. Хмельницький: Хмельницький національний університет, 2017. 310 с.

Додаткова

18. Бевз О. М. Проектування програмних засобів систем управління: навчальний посібник / О. М. Бевз, В. М. Папінов, Ю. А. Скидан. – Вінниця: ВНТУ, 2010. – 125 с.
19. Лаврищева Е. М. Software Engineering комп'ютерних систем. Парадигми, технології та CASEзасоби програмування. – Київ: Наукова думка, 2013. – 283 с.
20. Говорущенко Т. О. Аналіз, дослідження та оцінювання програмних систем: навчальний посібник. – Хмельницький: Хмельницький національний університет, 2019. 358 с.

10. ІНФОРМАЦІЙНІ РЕСУРСИ

1. Модульне середовище для навчання. Доступ до ресурсу: <https://msn.khnu.km.ua/course/view.php>.
2. Електронна бібліотека ХНУ. Доступ до ресурсу: http://lib.khnu.km.ua/asp/php_f/p1age_lib.php.
3. Репозитарій ХНУ. Доступ до ресурсу: <http://elar.khnu.km.ua/jspui/?locale=uk>.

