

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКА НАЦІОНАЛЬНА АКАДЕМІЯ ЗВ'ЯЗКУ ім. О.С. ПОПОВА

Кафедра комп'ютерних наук

“ЗАТВЕРДЖУЮ”
Зав.каф. комп'ютерних наук

_____ 2019 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
Якість програмного забезпечення та тестування

Спеціальність: 122 Комп'ютерні науки
(шифр і назва спеціальності)

факультет, відділення **Інфокомунікацій та програмної інженерії**

Одеса
2019 рік

Робоча програма Якість програмного забезпечення та тестування для студентів
за спеціальністю підготовки бакалавра 122 Комп'ютерні науки

Розробники програми: ст.викл. Северин М.В., к.т.н., доц. Єгошина Г.А.

Робоча програма затверджена на засіданні кафедри Комп'ютерних наук

Протокол від “___” _____ 2019 року № __

Завідувач кафедри Комп'ютерних наук

_____ (_____)

“___” _____ 20__ року

1. Опис навчальної дисципліни

Найменування показників	Галузь знань, напрям підготовки, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни
		денна форма навчання
Кількість кредитів – 3	Галузь знань <u>12 Інформаційні технології</u> (шифр і назва)	Вибіркова
	Спеціальність підготовки <u>122 Комп'ютерні науки</u> (шифр і назва)	
Семестрів – 1	Спеціалізація (професійне спрямування): не передбачено	Рік підготовки:
Змістових модулів – 2		4-й
Індивідуальне науково-дослідне завдання _____ (назва)		Семестр
Загальна кількість годин 1-го семестру – 90		4.1
		Лекції
Тижневих годин для денної форми навчання: аудиторних – 3 самостійної роботи студента – 3,4	Освітньо-кваліфікаційний рівень: <i>бакалавр</i>	14 год.
		Практичні, семінарські
		14 год.
		Лабораторні
		14 год.
		Самостійна робота
		48 год.
Індивідуальні завдання:	Не передбачено	
	Вид контролю: залік	

2. Анотація дисципліни

Програма вивчення навчальної дисципліни *«Якість програмного забезпечення та тестування»* складена відповідно до освітньо-професійної програми підготовки бакалаврів спеціальності *122 Комп'ютерні науки*.

Тестування виконує важливу роль в розробці якісного програмного забезпечення. Жорстка конкуренція на сучасному ринку програмного забезпечення призводить до необхідності максимально швидко поставляти програмні продукти на ринок. Проблема безперервної підтримки потрібного рівня якості і, в той же час, дотримання календарного плану призводить до створення нових стратегій тестування, нерозривно пов'язаних з процесом розробки.

Курс присвячено одному з найбільш важливих та нетривіальних аспектів в процесі створення складних програмних систем. Детально розглядається широкий спектр питань: від організації процесу тестування до тестування проекту та коду.

QA (Software Testing and Quality Assurance) або тестувальник - це фахівець із забезпечення якості програмного забезпечення. Без ретельного тестування неможливо досягти високої якості програмного продукту - ось чому QA-фахівці дуже затребувані в ІТ-компаніях. Програма курсу розроблена таким чином, що студент отримує всі необхідні знання і практичні навички для початку своєї кар'єри тестувальника. Курс дозволить вивчити основи, які є «must have» для всіх тестувальників, незалежно від сфери тестування і продукту, який належить тестувати.

3. Мета та завдання навчальної дисципліни

Метою викладання дисципліни *«Якість програмного забезпечення та тестування»* є ознайомлення студентів з існуючими способами контролю якості розробки програмного забезпечення з позицій тестування, навчання студентів сучасним методикам та інструментальним засобам для організації та проведення тестування програмного забезпечення.

Курс забезпечує засвоєння основних понять та визначень в галузі тестування, критеріїв вибору тестів, огляд різновидів тестування, аналіз особливостей процесу й технології індустріального тестування, набуття навичок застосування сучасних інформаційних технологій для аналізу та тестування інформаційних систем, створення звітної тестової документації.

Набуті студентами знання можуть бути використані під час виконання кваліфікаційної роботи.

Завданнями вивчення навчальної дисципліни є теоретична та практична підготовка майбутніх фахівців з таких питань:

- термінологія та визначення в галузі тестування, фази й технологія тестування, основні проблеми і завдання тестування;
- структурні та функціональні критерії вибору тестів, оцінки покриття проекту;
- модульне, інтеграційне, системне, регресійне тестування, автоматизація тестування, витрати тестування;

- планування тестування, підходи до розробки тестів, особливості ручної розробки й генерації тестів, автоматизація тестового циклу;
- документування тестування, огляди й метрики, аналіз методів відбору тестів, оцінка їх ефективності.

В результаті успішного засвоєння навчальної дисципліни студент матиме змогу продемонструвати такі результати навчання:

знання:

- основні поняття забезпечення якості програмних продуктів;
- особливості індустріального підходу в оцінці якості програмного продукту засобами тестування;
- основні функціональні фази тестування і основні завдання і проблеми тестування;
- критерії вибору тестів, оцінки покриття проекту;
- різновиди тестування;
- методику планування тестування, підходи до розробки тестів, особливості ручної розробки й генерації тестів;
- структуру тестового набору для автоматизованого тестування;
- структуру документів "Тестовий план" (Test Plan), "Тестова процедура" (Test Case), "Тестовий звіт" (Summary test report);
- методику описування тестових наборів (Test Suite) та тестових звітів (Bug Report).

вміння:

- аналізувати особливості процесу й технології тестування;
- планувати тестування;
- обирати кінцевий набір тестів;
- забезпечувати розробку тестів з урахуванням особливостей ручної розробки й генерації тестів і з використанням засобів автоматизації тестового циклу;
- виконувати тестування на основі специфікацій та сценаріїв;
- розробляти звітну тестову документацію.

Вивчення навчальної дисципліни передбачає формування та розвиток у студентів **компетентностей**:

загальних:

- здатність до абстрактного мислення, аналізу та синтезу
- здатність застосовувати знання у практичних ситуаціях;
- знання та розуміння предметної області та розуміння професійної діяльності;
- здатність вчитися й оволодівати сучасними знаннями;
- здатність до пошуку, оброблення та аналізу інформації з різних джерел
- здатність генерувати нові ідеї (креативність);
- здатність працювати в команді;
- здатність приймати обґрунтовані рішення;

- здатність оцінювати та забезпечувати якість виконуваних робіт.

фахових:

- здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління;
- здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника;
- здатність забезпечити організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності функціонування операційних систем і системного програмного забезпечення;
- здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника;
- здатність забезпечувати якість комп'ютерних систем та оцінювати їх показники якості з використанням відповідних моделей та засобів на всіх етапах розробки.

Результати навчання даної дисципліни деталізують такі **програмні результати навчання:**

- застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук;
- проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій;
- розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук;
- володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт) ;
- забезпечувати ефективне управління якістю комп'ютерних систем на основі використання відповідних методологій, моделей, та інструментальних засобів.

На вивчення навчальної дисципліни відводиться 90 годин / 3 кредити ECTS.

4. Інформаційний обсяг навчальної дисципліни

Програма навчальної дисципліни складається з таких змістових модулів:

№ з/п	Назва теми
Якість та верифікація програмного забезпечення	
1	Основні поняття й визначення у галузі тестування програмних продуктів. Поняття якості програмного забезпечення. Складові якості програмного забезпечення: надійність, супровід, практичність, ефективність, мобільність, функціональність. Світові стандарти якості програмних продуктів. Метрики оцінки якості на рівні функціонального програмування.
2	Моделі життєвого циклу розробки ПЗ. Задачі валідації та верифікації. Ролі, обов'язки та етапи процедури верифікації. Огляд методів верифікації. Перелік звітної документації.
Тестування програмного забезпечення	
3	Основи тестування програмного забезпечення. Класифікація тестів за об'єктом тестування: функціональне тестування, та тестування продуктивності. Напрями тестування продуктивності: тестування навантаження, стрес-тестування, тестування стабільності, конфігураційне тестування. Критерії вибору тестів. Модульне та інтеграційне тестування. Склад групи тестування, права і обов'язки членів групи тестування.
4	Визначення Test Plan, призначення, рекомендації з написання Test Plan. Види тест планів. Критерії початку і закінчення тестування. Тест дизайн. Визначення, етапи, ролі. Тестовий випадок (Test Case): визначення, структура, види тестових випадків. Баг Репорт (Bug Report). Серйозність і пріоритет дефекту. Основне призначення систем відстеження дефектів. Різновиди систем багтрекінга.
5	Функціональне тестування. Еквівалентну розбиття. Аналіз граничних значень. Метод всіх пар. Таблиці прийняття рішень.
6	Тестування методами «білого ящика». Метод покриття операторів. Метод покриття рішень. Метод покриття умов. Покриття рішень та умов. Метод комбінаторного покриття умов.
7	Тестування локалізації

5. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин					
	денна форма					
	усього	у тому числі				
л		п	лаб	інд	с.р	
<i>1</i>	2	3	4	5	6	7
Змістовий модуль 1. Якість та верифікація програмного забезпечення						
Основні поняття й визначення у галузі тестування програмних продуктів. Поняття якості програмного забезпечення. Складові якості програмного забезпечення: надійність, супровід, практичність, ефективність, мобільність, функціональність. Світові стандарти якості програмних продуктів. Метрики оцінки якості на рівні функціонального програмування.	12	2	2	2	-	8
Моделі життєвого циклу розробки ПЗ. Задачі валідації та верифікації. Ролі, обов'язки та етапи процедури верифікації. Огляд методів верифікації. Перелік звітної документації.	12	2	2	2	-	10
<i>Разом за змістовним модулем 1</i>	<i>30</i>	<i>4</i>	<i>4</i>	<i>4</i>	<i>-</i>	<i>18</i>
Змістовий модуль 2. Тестування програмного забезпечення						
Основи тестування програмного забезпечення. Класифікація тестів за об'єктом тестування: функціональне тестування, та тестування продуктивності. Напрями тестування продуктивності: тестування навантаження, стрес-тестування, тестування стабільності, конфігураційне тестування. Критерії вибору тестів. Модульне та інтеграційне тестування. Склад групи тестування, права і обов'язки членів групи тестування.	12	2	2	2	-	6
Визначення Test Plan, призначення, рекомендації з написання Test Plan. Види тест планів. Критерії початку і закінчення тестування. Тест дизайн. Визначення, етапи, ролі. Тестовий випадок (Test Case): визначення, структура, види тестових випадків. Баг Репорт (Bug Report). Серйозність і пріоритет дефекту. Основне призначення систем відстеження дефектів. Різновиди систем багтрекінга.	12	2	2	2	-	6
Функціональне тестування. Еквівалентну розбиття. Аналіз граничних значень. Метод всіх пар. Таблиці прийняття рішень.	12	2	2	2	-	6
Тестування методами «білого ящика». Метод покриття операторів. Метод покриття рішень. Метод покриття умов. Покриття рішень та умов. Метод комбінаторного покриття умов.	12	2	4	2	-	6
Тестування локалізації	12	2	-	2	-	6
<i>Разом за змістовним модулем 2</i>	<i>60</i>	<i>10</i>	<i>10</i>	<i>10</i>	<i>-</i>	<i>30</i>
УСЬОГО	90	14	14	14	-	48

6. Теми практичних занять

№ з/п	Назва теми	Кількість годин
1	Побудова діаграм залежності якості програмного продукту від часу розробки	2
2	Метрик для оцінки якості верифікації	2
3	Класичний підхід до інтеграційного тестування. Висхідне тестування, нисхідне, тестування за методом сендвіча. Критерії й метрики інтеграційного тестування. Створення тестів на основі UML діаграм варіантів використання.	2
4	Тест-план. Розділи тест плану. Тест дизайн. Визначення, етапи, ролі. Тестовий випадок (Test Case): визначення, структура, види тестових випадків.	2
5	Функціональне тестування. Еквівалентну розбиття. Аналіз граничних значень.	2
6	Тестування потоків керування програми. Критерій покриття операторів (C0). Критерій покриття рішень (C1). Критерій покриття шляхів	2
7	Граничне тестування циклу. Внутрішнє тестування циклу. Простий критерій покриття умов. Критерій покриття умов/рішень. Модифікований критерій покриття умов/рішень. Комбінаторний критерій покриття умов/рішень.	2
Разом		14

7. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Аналіз метрик якості та надійності ПЗ. Розробка метричної моделі якості. Використання метрик для оцінки якості програмного коду.	2
2	Валідація та верифікація ПЗ. Формування циклу попередження дефектів в ІТ-компанії.	2
3	Використання модульного тестування: функцій. Використання модульного тестування: класів.	2
4	Баг Репорт (Bug Report). Системи багтрекінга.	2
5	Функціональне тестування. Метод всіх пар. Таблиці прийняття рішень.	2
6	Тестування потоків даних програми. Критерії “all-defs”, “all p-uses”, “all c-uses”, “all c-uses / some p-uses”, “all p-uses / some c-uses”, “all uses”, “all du-paths”.	2
7	Тестування локалізації	2
Разом		14

8. Самостійна робота

№ з/п	Види роботи	Кількість годин
1	Проробка лекцій	20
2	Вивчення додаткового матеріалу до лекцій	10
3	Підготовка до практичних занять	-
4	Підготовка до лабораторних робіт	20
Разом:		50

9. Методи навчання

Мультимедійні лекції, демонстрування, викладання-пояснення, практичні заняття та лабораторні роботи, метод проблемно-орієнтовного навчання, самостійне спостереження, запитання-бесіда, ілюстрування, виконання індивідуальних завдань.

10. Методи контролю

Контроль рівня виконавських умінь для перевірки практичної підготовки, усне опитування, опитування під час презентації та захисту індивідуальних завдань.

11. Розподіл балів, які отримують студенти

Поточне тестування та самостійна робота		Сума
ЗМ1	ЗМ2	
40	60	100

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	A	відмінно	зараховано
82-89	B	добре	
74-81	C		
64-73	D	задовільно	
60-63	E		
35-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

12. Методичне, технічне й програмне забезпечення /обладнання

Презентації лекцій, комплекс навчально-методичного забезпечення дисципліни. В якості засобів для розробки UML діаграм рекомендується StarUml (<http://staruml.io/>), для формування тест-кейсів – TestRail.

11. Рекомендована література

1. Брауде Э. Дж. Технология разработки программного обеспечения . – СПб.: Питер, 2004. – 655 с.:ил.
2. A. Abran, J W. Moore, Guide to the Software Engineering Body of Knowledge. – 2004, - <http://www.computer.org/portal/web/swebok/html/contents>.
3. Myers G.J. The Art Of Software Testing. N.Y. John Wiley & Sons, Inc. 2004. – 254 p.
4. ДСТУ 2844-94. Програмні засоби ЕОМ. Забезпечення якості. Терміни та визначення.
5. ДСТУ 2850-94. Програмні засоби ЕОМ. Показники та методи оцінювання.
6. ISO 9126:2001, Software engineering -- Product quality
7. Д. Стотлемайер . Тестирование Web-приложений : Средства и методы для автоматизированного и ручного тестирования программного обеспечения Web-сайтов; пер. с англ. А.Хахалин. – М. : КУДИЦ-ОБРАЗ, 2003. – 240с.

8. Блэк Р. Ключевые процессы тестирования: Планирование, подготовка, проведение, совершенствование. – Минск.: Лори, 2006. – 544с.
9. Винниченко И.В. Автоматизация процессов тестирования. – СПб.: Питер, 2005. – 208с.
10. Дастин Э., Рэшка Дж., Пол Дж. Автоматизированное тестирование программного обеспечения. Внедрение, управление и эксплуатация : пер. с англ. - М. : Лори , 2003. - 567 с.
11. Канер С., Фолк Дж., Енг Кек Нгуен. Тестирование программного обеспечения: Фундаментальные концепции менеджмента бизнес-приложений. – Киев:ДиаСофт, 2001. - 544 с.
12. Бек К. Экстремальное программирование: разработка через тестирование: пер. с англ. - СПб. : Питер , 2003. - 224 с.
13. Б. Бейзер. Тестирование черного ящика. СПб: Питер, 2004.
14. Соммервилл И. Инженерия программного обеспечения, 6-е изд.: Пер. с англ. – М.: Вильямс, 2002. – 624 с.: ил.
15. Шафер Д., Фарелл Р., Шафер А. Управление программными проектами: достижение оптимального качества при минимуме затрат.: Пер. с англ. – М.: Издательский дом «Вильямс», 2003. - 1136 с.

12. Інформаційні ресурси

- <https://www.softwaretestinghelp.com/resources/>
- <https://www.softwaretestinghelp.com/resources/>
- <http://googletesting.blogspot.com/>
- <https://www.youtube.com/user/shindevijaykr/playlists>
- <https://www.softwaretestinghelp.com/practical-software-testing-new-free-ebook-download/>
- <http://www.softwaretestinghelp.org/software-testing-help-ebook-new/>
- <https://www.softwaretestinghelp.com/atlassian-jira-tutorial-1/>
- <https://www.softwaretestinghelp.com/free-online-software-testing-qa-training-course/>