

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

**ТЕСТУВАННЯ
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

**Методичні рекомендації
до самостійної роботи студентів
спеціальності 122 "Комп'ютерні науки"
першого (бакалаврського) рівня**

**Харків
ХНЕУ ім. С. Кузнеця
2022**

УДК 004.415.53(07.034)

Т38

Укладач Ю. І. Скорін

Затверджено на засіданні кафедри інформаційних систем.
Протокол № 1 від 27.08.2021 р.

Самостійне електронне текстове мережеве видання

Т38 **Тестування** програмного забезпечення [Електронний ресурс] : методичні рекомендації до самостійної роботи студентів спеціальності 122 "Комп'ютерні науки" першого (бакалаврського) рівня / уклад. Ю. І. Скорін. – Харків : ХНЕУ ім. С. Кузнеця, 2022. – 47 с.

Подано у систематизованому вигляді методичні рекомендації до самостійної роботи студентів згідно з тематичним планом навчальної дисципліни. Вміщено загальні положення щодо організації самостійної роботи студентів, методику підготовки до усного опитування, виконання практичних завдань, поточного тестового контролю.

Рекомендовано для студентів спеціальності 122 "Комп'ютерні науки" першого (бакалаврського) рівня.

УДК 004.415.53(07.034)

© Харківський національний економічний
університет імені Семена Кузнеця, 2022

Вступ

Сучасні економічні умови господарювання вимагають від фахівців, незалежно від їх спеціалізації, всебічного використання новітніх інформаційних технологій, комп'ютеризованих засобів збирання, оброблення та видачі необхідної інформації, створення на їх основі сучасного програмного забезпечення з метою підвищення якості та оперативності економічних розрахунків, ефективності процесу обґрунтування економічних рішень тощо.

Проєктування і створення програмного забезпечення вважаються досить молодими сферами діяльності, проте вони розвиваються швидше, ніж будь-які інші. Індустрія програмного забезпечення у даний час вважається однією з головних опор економічного зростання в багатьох країнах. Компанії, що виробляють програмне забезпечення та прагнуть до досягнення задоволеності клієнтів, часто зіштовхуються зі складними проблемами постачання високоякісних продуктів.

З того часу, як програмне забезпечення стало невід'ємною частиною нашого повсякденного життя, попит на нього значно збільшився. Сьогодні висока якість сприймається як обов'язковий компонент програмного забезпечення. Тому дуже важливо залучити групи контролю якості в процес планування і реалізації проєктів з самого початку. Проте до цих пір існують компанії, які вважають, що завдання забезпечення якості програмного забезпечення вирішується шляхом тестування наприкінці життєвого циклу розробки.

Крім того, постійно зростає поінформованість клієнтів і кінцевих користувачів про якість програмного забезпечення, яке вони купують. Застосунки або корпоративні системи, що демонструють низьку продуктивність або неякісну взаємодію з користувачами, будуть витіснені іншими, більш якісними продуктами. Сьогодні компаніям, що виробляють програмне забезпечення, необхідно дбати про якість своєї продукції як ніколи раніше.

У цьому контексті найактуальнішим стає питання забезпечення якості розробленого програмного продукту, а саме тестування, яке і є одним із найбільш ефективних способів підвищення якості розроблення програмного продукту, який входить до системи забезпечення якості програмного продукту.

Система якості повинна бути єдиним процесом, що проходить через весь життєвий цикл програмного продукту, гарантуючи, тим самим,

що якість формується в ході розроблення, а не раптом виявляється в кінці всього процесу. Необхідно наголошувати на попередженні появи дефектів, а не на їх виправленні після виникнення.

Якість програмного забезпечення – це сукупність властивостей, що обумовлюють його придатність задовольняти певні потреби відповідно до призначення. Вимоги до якості програмного забезпечення весь час підвищуються. Програми повинні бути надійними, зручними для роботи, простими для вивчення, а крім того користувач хоче мати гарантії, що він може довіряти певній програмі свої дані.

Реалізація вимог безпеки під час створення програмного забезпечення є однією зі складових загальної проблеми забезпечення його якості. Проблема якості програмного забезпечення має два аспекти, а саме, забезпечення і оцінювання якості.

Для забезпечення якості і надійності програм запропоновано безліч підходів, включаючи організаційні методи розроблення, різні технології і технологічні програмні засоби.

Якість програмного продукту характеризується набором властивостей, що визначають, наскільки продукт задовольняє вимоги зацікавлених сторін, а саме замовника продукту, спонсорів, кінцевого користувача, розробника та тестера продукту, інженера підтримки, співробітників відділів маркетингу, навчання та продажів. Кожен з учасників може мати різне уявлення про продукт і про те, наскільки він відповідає їх вимогам, тобто про те, наскільки висока якість продукту.

Таким чином, постановка завдання забезпечення якості програмного продукту переходить в завдання визначення зацікавлених осіб, їх критеріїв якості і – як наслідок – знаходження оптимального рішення, що задовольняє ці критерії.

Тестування є одним із найбільш усталених способів забезпечення якості розроблення програмного забезпечення та входить до набору ефективних засобів сучасної системи забезпечення якості програмного продукту. З технічного боку тестування полягає у виконанні програми на деякій множині вихідних даних і звірянні одержуваних результатів із заздалегідь відомими (еталонними) з метою встановлення відповідності різних властивостей і характеристик застосунка замовленим властивостям.

Як одна з основних фаз процесу розроблення програмного продукту, тестування характеризується досить великим внеском у сумарну трудомісткість розроблення продукту, унаслідок цього найбільший ефект

у зниженні трудомісткості може бути отриманий, насамперед, на фазі тестування.

Тому основні вклади в автоматизацію або генерацію коду слід здійснювати, насамперед, на цій фазі. Хоча в сучасному індустріальному програмуванні автоматизація тестування є досить поширеною практикою, водночас технологія верифікації вимог та специфікацій поки робить лише свої перші кроки. Завданням найближчої перспективи є рух у бік такого розподілу трудомісткості, щоб сумарна ціна виявлення більшості дефектів прагнула до мінімуму за рахунок виявлення переважного числа на найбільш ранніх фазах розроблення програмного продукту.

Навчальну дисципліну "Тестування програмного забезпечення" віднесено до групи освітньо-професійних дисциплін для студентів галузі знань 12 "Інформаційні технології" спеціальності 122 "Комп'ютерні науки" першого (бакалаврського) рівня. Вона є невід'ємною частиною циклу комп'ютерних дисциплін, необхідних фахівцям-аналітикам з:

- вивчення понять, принципів, методології та технологій тестування програмних продуктів (ПП);

- засвоєння основних понять та визначень з галузі тестування, критеріїв вибору тестів;

- огляду різновидів тестування;

- аналізу особливостей процесу й технології тестування;

- здобуття навичок застосування сучасних інформаційних технологій для аналізу та тестування інформаційних систем;

- створення звітної тестової документації.

Необхідним елементом успішного засвоєння навчального матеріалу дисципліни є самостійна робота студентів (СРС) з метою поглибленого вивчення окремих теоретичних положень навчальної дисципліни.

Практично всі види аудиторних навчальних занять, і навіть лекції, вимагають від студентів попередньої самостійної роботи. Крім того, у вищому навчальному закладі їм надається певна свобода вибору. Таким чином, набагато більше значення має не те, скільки із зазначених джерел вивчив студент, а наскільки глибоко він засвоїв вивчений матеріал.

Крім того, через нестачу часу не всі теми навчальної дисципліни докладно розглядаються на лабораторних заняттях. Деякі теми викладач не виносить навіть на лекцію, вказуючи лише, до яких джерел студенти повинні звернутися. Такий навчальний матеріал залишається для самостійного вивчення.

Тобто позааудиторна самостійна робота студента у ВНЗ не менш важлива, ніж обов'язкові навчальні заняття. Її успішність багато в чому визначається тим, наскільки вміло, раціонально сам студент зможе організувати свої індивідуальні заняття, наскільки регулярними і своєчасними вони будуть.

Самостійна робота студента – це форма організації навчального процесу, за якої заплановані завдання виконуються студентом самостійно під методичним керівництвом викладача [70].

Мета СРС – засвоєння в повному обсязі навчальної програми та формування у студентів загальних і професійних компетентностей, які відіграють суттєву роль у становленні майбутнього фахівця вищого рівня кваліфікації.

Самостійна робота студентів реалізується в різних видах. За виконання будь-яких видів самостійної роботи надається можливість отримати консультацію викладача.

Правильна організація СРС дає викладачеві можливість забезпечити вивчення найбільш складних питань програми з навчальної дисципліни.

Навчальний час, відведений для самостійної роботи студентів денної форми навчання, визначається навчальним планом навчальної дисципліни.

Під час самостійної роботи студент має бути активним учасником навчального процесу, навчитися свідомо ставитися до оволодіння теоретичними і практичними знаннями, вільно орієнтуватися в інформаційному просторі, нести індивідуальну відповідальність за якість власної професійної підготовки.

СРС містить:

опрацювання лекційного матеріалу;

доопрацювання лекційного матеріалу;

опрацювання та вивчення рекомендованої літератури, основних термінів та понять за темами навчальної дисципліни;

підготовку до лабораторних занять;

поглиблене опрацювання окремих лекційних тем або питань;

пошук (підбір) та огляд літературних джерел за заданою проблематикою навчальної дисципліни;

аналітичний розгляд наукової публікації;

контрольну перевірку студентами особистих знань за запитаннями для самодіагностики;

підготовку до контрольних робіт (тестування) та інших форм поточного контролю;

систематизацію вивченого матеріалу з метою підготовки до заліку.

Використання підходів промислового розроблення програмного забезпечення під час організації самостійної роботи студентів

Підготовка професіоналів для роботи у ІТ-компаніях потребує принципово інших методів, порівняно з традиційними, необхідність їх розроблення стає все більш гострою.

Для вирішення зазначеної проблеми доцільно використовувати у ході організації самостійної роботи студентів підходи, які застосовані в промисловому розробленні програмного забезпечення та асоційовані на сьогодні з поняттям "тестування програмних продуктів".

У даному підході застосовуються активні форми навчання, а саме імітується процес розроблення програмного забезпечення, прийнятий у ІТ-галузі, тим самим процес навчання наближається до умов реального ринку праці.

Для опису даної моделі процесу навчання необхідно ввести такі поняття.

1. Учасник процесу навчання. Учасниками процесу навчання є студент і викладач.

2. Роль. Під роллю розуміється сукупність дій і функцій, що виконуються учасниками процесу. Студент і викладач у процесі навчання з використанням методів програмної інженерії можуть перебувати в різних ролях, причому одна і та ж роль може виконуватися як студентом, так і викладачем.

3. Робочий продукт. Результат роботи учасника процесу в тій чи іншій ролі.

Ураховуючи, що основним об'єктом у даній моделі є завдання щодо створення програмного продукту, над чим йде колективна робота, необхідно ввести такі ролі учасників цього колективного освітнього процесу:

1. Автор. До обов'язків автора належить формулювання завдання для розроблення студентом.

2. Модератор. До обов'язків модератора належить призначення завдання для виконавців, контроль процесу роботи над завданням і винесення остаточного рішення про результат його вирішення.

3. Кодувальник. Завдання кодувальника – написання програми для вирішення призначеного йому завдання.

4. Інспектор. Інспектор займається аналізом коду розробленої програми на предмет її коректності та якості кодування, адекватності застосовуваних програмних рішень. Інспектор виносить один з трьох можливих вердиктів:

а) прийняти завдання без зауважень. Це означає, що інспектору абсолютно зрозумілі всі програмні рішення, код написаний правильно, текст програми забезпечений всіма необхідними коментарями;

б) прийняти завдання за умови усунення зауважень. Це означає, що програма в цілому написана правильно, однак деякі моменти залишаються незрозумілими або некоректними. Інспектор формулює свої зауваження і після їх усунення кодувальником перевіряє завдання знову. Процес усунення зауважень триває до тих пір, поки інспектор не винесе рішення "Прийняти без зауважень";

в) відхилити завдання. Це означає, що програма не виконує необхідних функцій або кількість зауважень настільки велика, що вона не може бути прийнята без суттєвого доопрацювання.

5. Тестер. До завдання тестера належить розроблення тесту, який дозволить перевірити якість написання програми шляхом пошуку в ній помилок, і власне перевірка. Робота тестера починається після того, як інспектор виніс вердикт "Прийняти без зауважень". У цьому разі тестер не повинен аналізувати код програми, він лише повинен запускати її на сценаріях, адекватних умові завдання.

Робочими продуктами розглянутої моделі процесу навчання є:

у автора – формулювання завдання;

у модератора – оцінювання роботи виконавців;

у кодувальника – код програми (текст);

у інспектора – зауваження і набір рекомендацій з доопрацювання коду;

у тестера – сценарій тестів.

Послідовність виконання ролей у процесі роботи над програмою:

автор формулює завдання, для вирішення якого необхідно написати програму, передає її кодувальнику;

модератор призначає ролі;

кодувальник пише програму і передає її інспектору;

інспектор аналізує програмний код і ухвалює своє рішення.

Якщо код приймається без зауважень, то програмний продукт передається тестеру. Якщо у інспектора є зауваження, він повідомляє їх кодувальнику.

Далі відбувається взаємодія кодувальника з інспектором щодо усунення зауважень до винесення рішення "Прийняти без зауважень";
 тестер розробляє сценарій і перевіряє якість програмного продукту;
 модератор ухвалює остаточне рішення про якість програмного продукту й оцінює результати роботи.

Можливо кілька варіантів розподілу ролей між учасниками процесу. У табл. 1 і 2 показано розподіл у рамках традиційних форм процесу навчання.

Таблиця 1

**Розподіл ролей між учасниками освітнього процесу
 за умови традиційної форми навчання (варіант 1)**

Ролі	Учасники	Функції
Автор	Викладач	Формулювання завдання
Модератор	Викладач	Розподіл завдань між студентами (видача завдань)
Кодувальник	Студент	Написання програмного коду
Інспектор	Викладач	Перевірка коду програми, оцінювання роботи студента
Тестер	Викладач	Запуск програми та її тестування, оцінювання роботи студента

У першому традиційному варіанті ролі автора, модератора, інспектора та тестера виконує викладач, а студент виконує лише роль кодувальника.

Застосування такого підходу на сьогодні виправдане лише для навчання студентів непрофільних, не пов'язаних безпосередньо з інформаційними технологіями, напрямів підготовки студентів.

У табл. 2 показано розподіл ролей для другого варіанта традиційної форми процесу навчання.

**Розподіл ролей між учасниками освітнього процесу
за умови традиційної форми навчання (варіант 2)**

Ролі	Учасники	Функції
Автор	Викладач	Формулювання завдання
Модератор	Викладач	Розподіл завдань між студентами (видача завдань)
Кодувальник	Викладач	Написання програмного коду, оцінювання роботи студента
Інспектор	Викладач	Перевірка коду програми, оцінювання роботи студента
Тестер	Студент	Запуск програми та її тестування

У другому традиційному варіанті ролі автора, модератора, інспектора та кодувальника виконує викладач, а студент виконує лише роль тестера.

Застосування такого підходу на сьогодні виправдане лише для навчання студентів непрофільних, не пов'язаних безпосередньо з інформаційними технологіями, напрямів підготовки студентів.

У табл. 3 показано розподіл ролей для першого варіанта запропонованої технології навчання.

**Розподіл ролей між учасниками освітнього процесу,
запропонована технологія навчання з використанням методів
програмної інженерії (варіант 1)**

Ролі	Учасники	Функції
Автор	Викладач	Формулювання завдання
Модератор	Викладач	Розподіл завдань між студентами, визначення ролі кожного зі студентів, оцінювання роботи студента
Кодувальник	Студент 1	Написання програмного коду
Інспектор	Студент 2	Перевірка коду програми, формулювання зауважень з поліпшення і доопрацювання
Тестер	Студент 3	Розроблення сценаріїв тестування, запуск програми та її тестування

У варіанті, що подано в табл. 3, викладач виконує ролі автора і модератора, а ролі кодувальника, інспектора та тестера виконують студенти.

Викладач формулює завдання на ту чи іншу тему, розподіляє ролі між студентами, контролює процес їх роботи й оцінює результат.

Студент-кодувальник пише програмний код і передає його студенту-інспектору.

Інспектор аналізує поданий код, формулює свої зауваження та (у разі необхідності) повертає його кодувальнику на доопрацювання. За умови відсутності зауважень програма передається студенту-тестеру.

Тестер розробляє сценарій тесту і "проганяє" через нього програму з формулюванням виведення типу "пройдено або не пройдено" за кожним пунктом сценарію.

Після закінчення роботи тестера викладач ставить оцінку кожному учаснику.

У тому випадку, якщо для групи студентів реалізується поглиблене вивчення програмування, можливий другий варіант розподілу ролей учасників (табл. 4).

Таблиця 4

**Розподіл ролей між учасниками освітнього процесу,
запропонована технологія навчання з використанням методів
програмної інженерії (варіант 2)**

Ролі	Учасники	Функції
Автор	Студент 1	Формулювання завдання
Модератор	Викладач	Визначення ролі кожного зі студентів, розподіл завдань між студентами, оцінювання роботи студента
Кодувальник	Студент 2	Написання програмного коду
Інспектор	Студент 3	Перевірка коду програми, формулювання зауважень з поліпшення і доопрацювання
Тестер	Студент 4	Розроблення сценаріїв тестування, запуск програми та її тестування

У даному варіанті роль автора виконують студенти.

Застосування описаної технології в процесі вивчення цієї навчальної дисципліни, а саме під час організації самостійної роботи студентів

дозволяє підвищити ефективність навчального процесу з точки зору набуття студентами компетентностей у сфері тестування програмних продуктів.

Крім цього, її використання дозволяє підвищити мотивацію студентів до вивчення навчальної дисципліни, ознайомити їх з прийомами колективної роботи, які застосовуються у професійному середовищі і забезпечити прозорість виставлення оцінок з навчальної дисципліни.

Графік самостійної роботи студентів

Необхідним елементом успішного засвоєння матеріалу навчальної дисципліни є самостійна робота студентів з вітчизняною та закордонною спеціальною економічною літературою, нормативними актами з питань державного регулювання економіки, статистичними матеріалами.

Основні види самостійної роботи, які запропоновані студентам для засвоєння теоретичних знань з навчальної дисципліни, наведені в табл. 5.

Таблиця 5

Завдання для самостійної роботи студентів та форми її контролю

Назва теми	Зміст самостійної роботи студентів	Форми контролю СРС	Рекомендована література
1	2	3	4
Змістовий модуль 1. Основи тестування програмного забезпечення			
<i>Тема 1.</i> Тестування як спосіб забезпечення якості програмного продукту	Вивчення лекційного матеріалу; огляд теоретичного матеріалу та підготовка доповіді "Поняття забезпечення якості програмного забезпечення"	Презентація результатів, захист лабораторної роботи	Основна: [2; 5; 7; 18]. Додаткова: [27 – 29]
<i>Тема 2.</i> Основні поняття тестування програмного забезпечення	Вивчення лекційного матеріалу; підготовка до лабораторного заняття, огляд теоретичного матеріалу та підготовка доповіді "Особливості тестування на етапі розроблення програмного продукту"	Презентація результатів, захист лабораторної роботи	Основна: [2; 13; 25]. Додаткова: [27 – 34]

1	2	3	4
Змістовий модуль 2. Організація процесу тестування програмного забезпечення			
<i>Тема 3. Класифікація видів тестування</i>	Вивчення лекційного матеріалу, огляд теоретичного матеріалу та підготовка доповіді "Класифікація видів тестування"	Презентація результатів, захист лабораторної роботи	Основна: [2; 3; 9; 26]. Додаткова: [27 – 30]
<i>Тема 4. Особливості індустріального тестування</i>	Вивчення лекційного матеріалу, підготовка до лабораторного заняття, огляд теоретичного матеріалу та підготовка доповіді "Особливості індустріального тестування"	Презентація результатів, захист лабораторної роботи	Основна: [2; 9; 12; 25]. Додаткова: [27 – 30]
<i>Тема 5. Регресійне тестування</i>	Вивчення лекційного матеріалу, підготовка до лабораторного заняття, підготовка до контрольної роботи, огляд теоретичного матеріалу та підготовка доповіді "Регресійне тестування"	Презентація результатів, захист лабораторної роботи, контрольна робота	Основна: [9; 12; 26]. Додаткова: [27 – 29]
<i>Тема 6. Проблеми якості програмного забезпечення</i>	Вивчення лекційного матеріалу, огляд теоретичного матеріалу та підготовка доповіді "Основні проблеми якості програмного забезпечення"	Презентація результатів, захист лабораторної роботи	Основна: [1; 3; 5; 7; 23]. Додаткова: [29 – 32]

1. Програма навчальної дисципліни

Змістовий модуль 1. Основи тестування програмного забезпечення

Тема 1. Тестування як спосіб забезпечення якості програмного продукту

1.1. Вступ. Основні теми лекційного курсу. Основні теми практикуму.

1.2. Мотивація до вивчення технологій тестування. Історія розвитку тестування програмного забезпечення. Якість програмного забезпечення. Якість програмного забезпечення в контексті міжнародних стандартів.

Тема 2. Основні поняття тестування програмного забезпечення

2.1. Концепція тестування. Основна термінологія з галузі тестування програмного забезпечення. Організація тестування програмного забезпечення. Специфікація програми.

2.2. Розроблення тестів. Аналіз тестових випадків. Виконання тестових випадків. Оцінювання результатів виконання програми на тестах. Етапи тестування. Керувальний граф програми. Основні проблеми тестування.

Тема 3. Класифікація видів тестування

3.1. За об'єктом тестування. Функціональне тестування. Тестування продуктивності. Тестування навантаження. Стрес-тестування. Тестування стабільності. Конфігураційне тестування. Тестування зручності використання. Тестування інтерфейсу користувача. Тестування безпеки. Тестування локалізації. Тестування сумісності.

3.2. За повнотою інформації про об'єкт тестування. Тестування чорної скриньки. Тестування білої скриньки.

3.3. За ступенем автоматизації процесу тестування. Ручне тестування. Автоматизоване тестування.

3.4. За ступенем ізольованості компонентів. Компонентне (модульне) тестування. Інтеграційне тестування. Системне тестування.

3.5. За часом проведення тестування. Альфа-тестування. Бета-тестування. Реліз.

3.6. За стратегією тестування. Тестування частин проти тестування цілого. Псевдоналаштування і мутаційні тестування.

Змістовий модуль 2. Організація процесу тестування програмного забезпечення

Тема 4. Особливості індустріального тестування

4.1. Індустріальний підхід. Особливості індустріального тестування.

4.2. Процес тестування. Фази процесу тестування. Тестовий цикл. Планування тестування. Тестовий план.

4.3. Підходи до розроблення тестів. Тестування специфікацій. Тестування сценаріїв. Ручне розроблення тестів. Генерація тестів. Виконання тестів.

4.4. Документація і супровід тестів. Тестові процедури. Опис тестів. Тестові процедури. Тестовий звіт. Оцінювання якості тестів. Тестові метрики. Огляди тестів і стратегії.

Тема 5. Регресійне тестування

5.1. Регресійне тестування. Цілі і завдання регресійного тестування. Види регресійного тестування.

5.2. Метод відбору тестів. Обґрунтування коректності методу відбору тестів. Класифікація тестів під час відбору. Можливості повторного використання тестів. Класифікація вибіркового методів.

Тема 6. Проблеми якості програмного забезпечення

6.1. Проблеми якості програмного забезпечення та практичні рекомендації. Якість програмного забезпечення як необхідність. Наскрізне забезпечення якості програмного забезпечення. Стандарти розроблення програмного забезпечення та їх використання.

6.2. Методики забезпечення якості програмного забезпечення в рамках усього життєвого циклу розроблення. Аналіз вимог. Аналіз і наскрізний контроль коду. Сесійне тестування. Тестування, засноване на ризиках.

2. Методичні рекомендації щодо підготовки до лекційних занять, роботи з рекомендованою літературою

Головна мета в період підготовки до лекційних занять – навчитися методам самостійного розумової праці, свідомо розвивати свої творчі здібності й оволодівати навичками творчої роботи.

Запис лекцій – складний вид аудиторної роботи студента. Уважне слухання і конспектування лекцій передбачає інтенсивну розумову діяльність студента. Короткі записи лекцій, їх конспектування допомагає засвоїти навчальний матеріал.

Конспект є корисним тоді, коли записана основна інформація і виконано це самим студентом. Не треба прагнути записати дослівно всю лекцію. Таке "конспектування" приносить більше шкоди, ніж користі. Запис лекцій рекомендується вести, за можливістю власними формулюваннями.

Бажано здійснювати запис на одній сторінці, а наступну залишати для опрацювання навчального матеріалу самостійно. Конспект лекції краще розподіляти на пункти, параграфи тощо.

Цьому більшою мірою будуть сприяти пункти плану лекції, що запропоновані викладачем.

Але не менш важливим є вивчення конспекту лекцій, його доповнення. Працюючи з конспектом лекцій, завжди необхідно використовувати не тільки підручник, але й ту літературу, яку додатково рекомендував лектор. Саме така серйозна, клопітка робота з лекційним матеріалом дозволить глибоко оволодіти знаннями.

Алгоритм виконання завдання зі збагачення і доповнення лекційного матеріалу в ході самостійної роботи:

усвідомте і конкретизуйте для себе мету завдання, наприклад, доопрацювати тему лекції, підготуватися до контрольної роботи, з'ясувати підстави неоднозначних підходів різних авторів до одного і того ж явища, поняття тощо;

відтворіть основні положення прочитаної лекції викладачем з досліджуваного питання;

установіть зв'язок між змістом лекції і проблемою, яка була розглянута у процесі вивчення нового літературного джерела;

прочитайте літературне джерело, усвідомте його зміст, звернувши увагу на закладені в ньому світоглядні ідеї, сутність і значення;

з'ясуйте методологічну сутність, педагогічний зміст роботи, мету і завдання, запропоновані методи їх досягнення, можливий результат;

складіть план матеріалу, з яким працюєте, і відповідно з ним конспектуйте рекомендований літературним джерелом матеріал;

конспектуючи літературне джерело, зобразіть у вигляді схем, діаграм і таблиць деякі його положення, факти, статистичні дані, що допоможе краще засвоїти досліджуваний матеріал;

підготуйте відповіді на поставлені викладачем запитання, усно або письмово відтворіть основні положення та факти;

запишіть питання, які хотілося б запропонувати для обговорення в групі, або які викликають сумніви.

Вміння самостійно працювати з літературою – ще одне з найважливіших умінь, що повинен набути студент під час навчання у ВНЗ.

Дійсно, на сьогодні накопичена людством така величезна кількість інформації, яка до того ж постійно збільшується з кожним днем, що людина не в змозі засвоїти її в повному обсязі навіть у межах однієї області знання.

Тому набагато важливіше навчитися орієнтуватися в цьому морі найрізноманітніших відомостей, швидко знаходити джерела необхідної інформації, тобто перш за все знати, що, де і як шукати.

Самостійна робота з підручниками, навчальними посібниками, науковою, довідковою і популярною літературою, матеріалами періодичних видань та інтернету, статистичними даними є ефективним методом отримання знань, дозволяє значно активізувати процес оволодіння інформацією, сприяє глибокому засвоєнню матеріалу, що вивчається, формує у студентів своє ставлення до конкретної проблеми.

Глибшому розкриттю питань сприяє ознайомлення з додатковою літературою, рекомендованою викладачем з кожної теми практичного заняття, що дозволяє студентам проявити свою індивідуальність в рамках виступу на даних заняттях, виявити широкий спектр думок з досліджуваної проблеми.

2.1. Перелік тем презентацій

Змістовий модуль 1

Основи тестування програмного забезпечення

Тема 1. Тестування як спосіб забезпечення якості програмного продукту

1. Історична довідка про розвиток інформаційних технологій забезпечення якості програмного забезпечення шляхом тестування.
2. Тестування як спосіб забезпечення якості програмного продукту.
3. Складові якості програмного забезпечення: надійність, супровід, практичність, ефективність, мобільність, функціональність.
4. Набір вимог до управління якістю.

5. Вимоги до менеджменту компанії.
6. Вимоги до контролю продукції.
7. Характеристики якості програмного продукту.
8. Методи визначення показників якості.
9. Метрики програмного продукту та їх складові.
10. Стандарт якості ISO – світовий стандарти якості ПП.

Тема 2. Основні поняття тестування програмного забезпечення

1. Якість програмного забезпечення.
2. Програмний продукт.
3. Показники якості програмного забезпечення.
4. Життєвий цикл програмного продукту.
5. Концепція тестування.
6. Основна термінологія.
7. Організація тестування.
8. Специфікація програми.
9. Розроблення тестів.
10. Аналіз тестових випадків.
11. Виконання тестових випадків.
12. Оцінювання результатів виконання програми на тестах.
13. Три етапи тестування.

Тема 3. Класифікація видів тестування

1. Класифікація видів тестування.
2. За об'єктом тестування.
3. За повнотою інформації про об'єкт тестування.
4. За ступенем автоматизації процесу тестування.
5. За ступенем ізольованості компонентів.
6. За часом проведення тестування.
7. За рівнем тестування.
8. За стратегією тестування.
9. За ознакою позитивності сценаріїв.
10. За ступенем підготовленості до тестування.
11. Порівняльний аналіз ручного й автоматизованого тестування.
12. Інструментарій для автоматизованого тестування.
13. Статичне тестування.

14. Динамічне тестування.
15. Ручне тестування.
16. Автоматизоване тестування.
17. Комбінований вид тестування.
18. Альфа-тестування.
19. Бетта-тестування.
20. Тестування релізу.

Змістовий модуль 2

Організація процесу тестування програмного забезпечення

Тема 4. Особливості індустріального тестування

1. Індустріальний підхід.
2. Особливості індустріального тестування.
3. Процес індустріального тестування.
4. Фази процесу індустріального тестування.
5. Тестовий цикл.
6. Планування тестування.
7. Тестовий план.
8. Типи тестування.
9. Тестування продуктивності.
10. Тестування на відповідність стандартам.
11. Тестування сумісності з іншими програмно-апаратними комплексами.
12. Тестування роботи з оточенням.
13. Тестування роботи на конкретній платформі.
14. Підходи до розроблення тестів.
15. Тестування специфікацій.
16. Тестування сценаріїв.
17. Ручне розроблення тестів.
18. Генерація тестів.
19. Виконання тестів.
20. Ручне тестування.
21. Автоматизоване тестування.
22. Порівняння ручного й автоматизованого тестування.
23. Документація і супровід тестів.
24. Тестові процедури.

25. Опис тестів.
26. Тестовий звіт.
27. Оцінювання якості тестів.
28. Тестові метрики.
29. Огляди тестів і стратегії.
30. Тестування стратегією "скляної скриньки".
31. Тестування стратегією "чорної скриньки".
32. Призначення та структура плану тестування.
33. Компоненти плану тестування.
34. Підсумковий звіт тестування.
35. Методи планування тестування документації.

Тема 5. Регресійне тестування

1. Модульне тестування.
2. Інтеграційне тестування.
3. Системне тестування.
4. Регресійне тестування.
5. Комбінування рівнів тестування.
6. Цілі і завдання регресійного тестування.
7. Види регресійного тестування.
8. Обґрунтування коректності методу відбору тестів.
9. Класифікація тестів під час відбору.
10. Можливості повторного використання тестів.
11. Класифікація вибірових методів.
12. Регресійне тестування об'єктно-орієнтованих програм.
13. Зменшення обсягу програми, що тестується.
14. Методи впорядкування тестів.
15. Доцільність відбору тестів.
16. Функції передбачення доцільності.

Тема 6. Проблеми якості програмного забезпечення

1. Проблеми якості програмного забезпечення.
2. Якість програмного забезпечення як необхідність.
3. Наскрізне забезпечення якості програмного забезпечення.
4. Стандарти розроблення програмного забезпечення та їх використання.
5. Методики забезпечення якості програмного забезпечення в рамках усього життєвого циклу розроблення.

6. Аналіз вимог.
7. Аналіз і наскрізний контроль коду.
8. Сесійне тестування.
9. Тестування, засноване на ризиках.
10. Питання прогнозування помилок.
11. Критерії щодо тестування вимог.
12. Проблеми якості програмного забезпечення.
13. Якість програмного забезпечення як необхідність.
14. Наскрізне забезпечення якості програмного забезпечення.
15. Стандарти розроблення програмного забезпечення.
16. Методики забезпечення якості програмного забезпечення в рамках усього життєвого циклу програмного продукту.

3. Методичні рекомендації щодо підготовки до виконання практичних завдань

Практичні види роботи становлять ряд завдань з навчальної дисципліни для самостійного виконання під час підготовки до занять та безпосередньо під час занять. У середньому виконання практичного завдання залежно від складності обраної теми та особливостей організації студентом своєї самостійної роботи становить від 30 до 90 хвилин.

Практичні завдання виконуються на заняттях. Оцінка за виконання практичних завдань враховується в роботі на заняттях. Додатковий бал (від одного до двох) за самостійну підготовку до практичних занять студент може отримати за умови якісного виконання самих завдань, а також за підготовку есе за матеріалами, які не ввійшли до лекційного курсу.

У процесі підготовки до виконання практичних завдань студенту необхідно опрацювати теоретичний матеріал з досліджуваної теми, наведений у лекційному курсі, методичні рекомендації до виконання практичних робіт, виконати приклади практичних завдань, що містяться в робочій навчальній програмі, підготувати есе за заданою темою.

Есе за заданою темою – це викладення думок за темою у вільній формі, тобто есе є літературною формою, невеликим прозовим текстом, що виражає індивідуальну точку зору автора. Жанр есе передбачає свободу творчості.

Головне у есе – особистість автора, його думки, почуття, ставлення до світу. Це головне завдання твору. Однак слід пам'ятати, що незважаючи на свободу творчості, писати в жанрі есе нелегко, оскільки потрібно знайти оригінальну ідею (навіть на традиційному матеріалі), нестандартний погляд на будь-яку проблему.

Стиль есе відрізняється образністю та парадоксальністю. Для передання особистісного сприйняття, освоєння світу автор залучає численні приклади, проводить паралелі, підбирає аналогії, використовує всілякі асоціації.

Специфіка жанру есе полягає в тому, що якщо ви хочете сказати щось своє, нове, нестандартне, то жанр есе – це ваш жанр.

Основною відмінною рисою есе є його стислість, і хоча обсяг може коливатися від однієї до п'ятдесяти сторінок, зазвичай воно займає від десяти до двадцяти сторінок. Есе має певну внутрішню єдність, яку зазвичай важко зберегти у великих прозових текстах, що виражають авторське бачення.

Написані есе студенти здають викладачеві на перевірку, основною формою якої є рецензування. Можуть застосовуватися й інші форми перевірки, наприклад, взаєморецензування робіт студентами або захист есе в формі усної доповіді на практичних видах занять [73].

3.1. Перелік тем есе

Змістовий модуль 1

Основи тестування програмного забезпечення

Тема 1. Тестування як спосіб забезпечення якості програмного продукту

1. Складові якості програмного забезпечення: надійність, супровід, практичність, ефективність, мобільність, функціональність.
2. Стандарт якості ISO – світовий стандарт якості.
3. Історична довідка про розвиток інформаційних технологій забезпечення якості програмного забезпечення шляхом тестування.
4. Мінімальний набір вимог до управління якістю.
5. Вимоги до менеджменту компанії.
6. Вимоги до контролю продукції.
7. Характеристики якості програмного продукту.

8. Методи визначення показників якості.
9. Метрики програмного продукту та їх складові.

Тема 2. Основні поняття тестування програмного забезпечення

1. Склад групи тестування.
2. Права й обов'язки членів групи тестування.
3. Критерії вибору тестів.

Тема 3. Класифікація видів тестування

1. Класифікація тестів за рівнем автоматизації.
2. Порівняльний аналіз ручного і автоматизованого тестування.
3. Інструментарій для автоматизованого тестування.

Змістовий модуль 2

Організація процесу тестування програмного забезпечення

Тема 4. Особливості індустріального тестування

1. Стратегії тестування.
2. Тестування стратегією "скляної скриньки".
3. Тестування стратегією "чорної скриньки".
4. Призначення та структура плану тестування.
5. Компоненти плану тестування.
6. Підсумковий звіт тестування.
7. Методи планування тестування документації.

Тема 5. Регресійне тестування

1. Методи та засоби оброблення програмних помилок.
2. Перенавантаження.
3. Вплив причини "занадто велика кількість всіх можливих комбінацій вхідних даних".
4. Вплив причини "занадто велика кількість усіх можливих послідовностей виконання коду програми".

Тема 6. Проблеми якості програмного забезпечення

1. Питання прогнозування помилок.
2. Критерії щодо тестування вимог.
3. Проблеми якості програмного забезпечення.

4. Якість програмного забезпечення як необхідність.
5. Наскрізне забезпечення якості програмного забезпечення.
6. Стандарти розроблення програмного забезпечення.
7. Методики забезпечення якості програмного забезпечення в рамках усього життєвого циклу програмного продукту.

4. Методичні рекомендації щодо підготовки до поточного тестового контролю (контрольної роботи)

Своєчасно і якісно виконані письмові роботи є дієвою і конструктивною формою зв'язку індивідуальності студента з викладачем. Зміст роботи дозволяє виявити ступінь розуміння і засвоєння студентом навчального матеріалу, його вмінь самотійно працювати в різних аспектах освітнього процесу.

Успіх і якість виконання письмових робіт обумовлені обізнаністю студента в навчальному плані (за курсами навчання), у змісті освітніх програм з навчальних дисциплін, його вміння доцільно планувати й організувати самотійну роботу, а так само від наявності ясних, чітких рекомендацій і вимог педагогів, що висуваються до змісту виконуваної студентом роботи.

Письмова робота розглядається як провідна навчальна діяльність студента, що дає можливість творчо, індивідуально й оригінально розкрити і реалізувати потенціал майбутнього.

Успішне виконання тестового контролю є необхідною умовою підсумкової позитивної оцінки відповідно до рейтингової системи навчання.

Тестові завдання підготовлені на основі лекційного матеріалу, підручників та навчальних посібників з даної навчальної дисципліни.

Виконання тестових завдань надає студентам можливість самотійно контролювати рівень своїх знань, виявляти "прогалини" в знаннях і вживати заходів щодо їх ліквідації.

Форма викладу тестових завдань дозволяє закріпити і відновити в пам'яті засвоєний матеріал.

Запропоновані тестові завдання охоплюють вузлові питання теоретичних і практичних основ з цієї навчальної дисципліни.

Для формування завдань використана замкнена форма. У студента є можливість вибору правильної відповіді або кількох правильних відповідей із запропонованих варіантів.

Для виконання тестових завдань студенти повинні вивчити лекційний матеріал з теми; відповідні розділи підручників, навчальних посібників та інших літературних джерел.

Контрольні тестові завдання виконуються студентами на лабораторних заняттях.

Пробні тестові завдання містяться в робочій навчальній програмі дисципліни.

З ними доцільно ознайомитися під час підготовки до контрольного тестування.

Порядок проведення поточного оцінювання знань студентів полягає у такому. Поточний тестовий контроль (контрольна робота) проводиться два рази за семестр.

Однією з основних і безперечних його переваг є мінімум часових витрат на отримання надійних підсумків контролю.

У ході тестування використовуються електронні варіанти, які дозволяють отримати результати практично відразу після завершення тесту.

Тестування виконує три основні взаємопов'язані функції: діагностичну, навчальну і виховну.

Діагностична функція полягає у виявленні рівня знань, вмінь та навичок студента. Це основна функція тестування.

Навчальна функція тестування полягає в мотивуванні студента до активізації роботи із засвоєння навчального матеріалу.

З метою посилення навчальної функції тестування можуть бути використані додаткові заходи стимулювання студентів, а саме:

надання викладачем приблизного переліку питань для самостійної підготовки;

наявність у самому тесті підказок, спільне опрацювання результатів тесту.

Виховна функція проявляється в періодичності тестового контролю. Це дисциплінує, організовує і спрямовує діяльність студентів, допомагає виявити й усунути "прогалини" в знаннях, формує прагнення розвинути свої здібності.

Тест передбачає запитання одиничного і множинного вибору щодо перевірки знань основних категорій навчальної дисципліни.

Максимальна кількість балів за контрольну роботу становить 10.

Розподіл балів має такий вигляд: якщо студент дав правильну відповідь на:

0 – 5 % запитань, то він отримує 1 бал;

6 – 15 % – 2 бали;

16 – 25 % – 3 бали;

26 – 35 % – 4 бали;

36 – 45 % – 5 балів;

46 – 55 % – 6 балів;

56 – 65 % – 7 балів;

66 – 75 % – 8 балів;

76 – 85 % – 9 балів;

86 – 95 % – 10 балів.

4.1. Тести

1. Укажіть, хто з членів групи тестування відповідає за організацію процесу тестування:

а) керівник групи тестування;

б) тест-аналітик;

в) тест-розробник;

г) виконавець тестів.

2. Укажіть, хто з членів групи тестування відповідає за розроблення тестових специфікацій та аналіз результатів тестування:

а) керівник групи тестування;

б) тест-аналітик;

в) тест-розробник;

г) виконавець тестів.

3. Оберіть правильне твердження:

а) методи тестування описуються у плані тестування (*test plan*);

б) методи тестування описуються в тестовій специфікації (*test case specification*);

в) методи тестування описуються в тестових процедурах (*test procedure specification*).

4. Укажіть, які документи використовуються для розроблення плану тестування (*test plan*):

- а) технічне завдання;
- б) тестові специфікації (*test case specification*);
- в) тестові процедури (*test procedure specification*);
- г) підсумковий звіт тестування (*test summary report*).

5. Який із критеріїв тестування використовує інформацію про структуру програми:

- а) структурні критерії;
- б) функціональні критерії;
- в) критерії стохастичного тестування;
- г) мутаційні критерії?

6. На якому етапі тестування застосовуються структурні критерії:

- а) регресійне тестування;
- б) модульне тестування;
- в) інтеграційне тестування;
- г) системне тестування?

7. Який із критеріїв тестування забезпечує контроль ступеня виконання вимог замовника в програмному продукті:

- а) структурні критерії;
- б) функціональні критерії;
- в) критерії стохастичного тестування;
- г) мутаційні критерії?

8. Яка модель тестування використовується у процесі функціонального тестування:

- а) чорна скринька;
- б) біла скринька?

9. Яка модель тестування використовується у ході модульного тестування:

- а) чорна скринька;
- б) біла скринька?

10. Яка модель тестування використовується у процесі інтеграційного тестування:

- а) чорна скринька;
- б) біла скринька?

11. Виявлення помилок, пов'язаних з роботою системи, в цілому виконується таким способом:

- а) системним тестуванням;
- б) модульним тестуванням;
- в) інтеграційним тестуванням;
- г) тестуванням інтерфейсів;
- ґ) стрес-тестуванням;
- д) навантажувальним тестуванням.

12. До якого класу тестування можна зарахувати тестування інтерфейсів (екранів, кнопок тощо):

- а) системне тестування;
- б) функціональне тестування;
- в) GUI-тестування;
- г) UNIT-тестування;
- ґ) стрес-тестування;
- д) навантажувальне тестування?

13. До якого класу тестів належить перевірка працездатності роботи системи:

- а) системне тестування;
- б) функціональне тестування;
- в) GUI-тестування;
- г) UNIT-тестування;
- ґ) стрес-тестування;
- д) навантажувальне тестування?

14. Для перевірки працездатності окремих елементів системи використовується:

- а) системне тестування;
- б) функціональне тестування;
- в) GUI-тестування;
- г) UNIT-тестування;

- г) стрес-тестування;
- д) навантажувальне тестування.

15. Яке тестування необхідно виконати для перевірки працездатності програми в екстремальних умовах:

- а) системне тестування;
- б) функціональне тестування;
- в) GUI-тестування;
- г) UNIT-тестування;
- г) стрес-тестування;
- д) навантажувальне тестування?

16. Тестування характеристик програмного продукту, пов'язаних зі швидкістю оброблення даних, – це:

- а) системне тестування;
- б) функціональне тестування;
- в) GUI-тестування;
- г) UNIT-тестування;
- г) стрес-тестування;
- д) навантажувальне тестування.

17. Яке тестування слід виконувати у разі випуску нового релізу продукту:

- а) системне тестування;
- б) функціональне тестування;
- в) GUI-тестування;
- г) UNIT-тестування;
- г) стрес-тестування;
- д) навантажувальне тестування?

18. У вікні введення пароля розташована кнопка підтвердження введення пароля "Реєстрація":

- а) це є помилкою;
- б) це не є помилкою.

19. У модальному вікні розташована кнопка "Застосувати":

- а) це є помилкою;
- б) це не є помилкою.

20. Чи є помилкою наявність піктограм на застосованих кнопках:

- а) так;
- б) ні.

21. У вікні реєстрації користувача під час вибору в полі "*LOGIN*" імені *user* зникає поле "*PASSWORD*":

- а) це є помилкою;
- б) це не є помилкою.

22. У процесі введення даних у полі введення відсутня підказка:

- а) це є помилкою;
- б) це не є помилкою.

23. Чи є обмеження на ширину поля зі списком:

- а) так;
- б) ні?

24. Список містить більше семи елементів, при цьому список не відсортований:

- а) це є помилкою;
- б) це не є помилкою.

25. Форма містить кілька радіо-кнопок, при цьому ні одна з кнопок не вибрана:

- а) це є помилкою;
- б) це не є помилкою.

26. Список містить більше 50 елементів. У цьому випадку список відсортований за частотою використання елементів:

- а) це є помилкою;
- б) це не є помилкою.

27. Тестується реєстраційна форма. Після введення всіх даних натиснути кнопку підтвердження введення і бачити повідомлення про неправильне введення даних у полі "Дата народження":

- а) це є помилкою роботи застосунка;
- б) застосунок працює правильно.

28. Тестується реєстраційна форма. Після введення чисел у полі "Прізвище" відображається модальне вікно з повідомленням про помилку:

- а) це є помилкою роботи застосунка;
- б) застосунок працює правильно.

29. Тестується реєстраційна форма. Після введення останнього значення в поле введення форми натиснути клавішу ENTER. Застосунок почав оброблення форми:

- а) це є помилкою роботи застосунка;
- б) застосунок працює правильно.

30. Тестується головне меню застосунка. У разі натиснення пункту меню "Довідники" відкривається форма довідників без заголовка:

- а) це є помилкою роботи застосунка;
- б) застосунок працює правильно.

31. Укажіть, хто з членів групи тестування відповідає за організацію процесу тестування:

- а) керівник групи тестування;
- б) тест-аналітик;
- в) тест-розробник;
- г) виконавець тестів.

32. Укажіть, хто з членів групи тестування відповідає за розроблення тестової документації:

- а) керівник групи тестування;
- б) тест-аналітик;
- в) тест-розробник;
- г) виконавець тестів.

33. Укажіть, хто з членів групи тестування відповідає за розроблення автоматизованих тестів, установку і супроводження тестового оточення:

- а) керівник групи тестування;
- б) тест-аналітик;
- в) тест-розробник;
- г) виконавець тестів.

34. Укажіть, хто з членів групи тестування відповідає за виконання тестів і реєстрацію помилок:

- а) керівник групи тестування;
- б) тест-аналітик;
- в) тест-розробник;
- г) виконавець тестів.

35. Укажіть, хто з членів групи тестування відповідає за роботу з документацією:

- а) керівник групи тестування;
- б) тест-аналітик;
- в) тест-розробник;
- г) виконавець тестів.

5. Контрольні запитання для самодіагностики

Змістовий модуль 1

Основи тестування програмного забезпечення

Тема 1. Тестування як спосіб забезпечення якості програмного продукту

1. Назвіть шляхи подальшого розвитку методів та засобів тестування в Україні.
2. У чому полягає багаторівнева модель якості програмного забезпечення?
3. Охарактеризуйте поняття якості програмного забезпечення.
4. У чому полягають мета та завдання навчальної дисципліни?
5. На знанні яких навчальних дисциплін базується опанування цієї навчальної дисципліни?
6. Вивчення яких навчальних дисциплін забезпечує дана навчальна дисципліна?
7. Наведіть складові якості програмного забезпечення.
8. Охарактеризуйте стандарт ISO.
9. Укажіть, хто з членів групи тестування відповідає за розроблення тестових специфікацій та аналіз результатів тестування?

10. Укажіть, хто з членів групи тестування відповідає за організацію процесу тестування?

11. Який із критеріїв тестування вказується в описі вимог до програмного продукту?

Тема 2. Основні поняття тестування програмного забезпечення

1. Назвіть функціональні фази тестування.

2. Охарактеризуйте особливості індустріального підходу в оцінюванні якості програмного продукту засобами тестування.

3. Класифікація тестів за об'єктом тестування: функціональне тестування та тестування продуктивності.

4. Назвіть напрями тестування продуктивності: тестування навантаження, стрес-тестування, тестування стабільності, конфігураційне тестування.

5. Назвіть склад групи тестування, охарактеризуйте права й обов'язки членів групи тестування.

6. Визначте місце тестування на етапах розроблення програмного продукту.

7. Назвіть критерії та методи системи менеджменту якості.

8. Визначте критерії вибору тестів.

9. Охарактеризуйте модульне та інтеграційне тестування.

10. На якому етапі тестування застосовуються структурні критерії?

11. Яка умова використовується під час проходження тестування кожної команди не менше одного разу?

12. Які умови використовуються під час тестування процесу проходження кожного шляху?

13. Який з критеріїв тестування забезпечує контроль ступеня виконання вимог замовника в програмному продукті?

Тема 3. Класифікація видів тестування

1. У чому полягає структура тестового набору для автоматичного тестування?

2. У чому полягає структура інструментального середовища автоматизації тестування?

3. Назвіть типи тестів та їх роль у процесі розроблення програмного забезпечення.

4. Визначте особливості тестування на етапі планування.
5. Визначте особливості тестування на етапі проєктування.
6. У чому полягає автоматизація тестування?
7. Укажіть, хто з членів групи тестування відповідає за розроблення автоматизованих тестів, установку і супроводження тестового оточення?
8. Яка модель тестування використовується у ході функціонального тестування?
9. Яка модель тестування використовується у ході модульного тестування?
10. Яка модель тестування використовується під час інтеграційного тестування?
11. До якого класу тестування можна зарахувати тестування інтерфейсів (екранів, кнопок тощо)?

Тема 4. Особливості індустріального тестування

1. Обґрунтуйте вибір критеріїв якості тестування.
2. Охарактеризуйте статичне і динамічне тестування. У чому полягає їх різниця?
3. Особливості тестування "скляної скриньки" на стадії кодування.
4. Дайте визначення регресійному тестуванню.
5. Особливості тестування "чорної скриньки".
6. Як здійснюється забезпечення якості програмного продукту засобами тестування?
7. Який із критеріїв тестування використовує інформацію про структуру програми?

Змістовий модуль 2

Організація процесу тестування програмного забезпечення

Тема 5. Регресійне тестування

1. Охарактеризуйте регресійне тестування.
2. У чому полягає розроблення тестових сценаріїв та процедур?
3. У чому полягає виконання тестів?
4. Назвіть компоненти плану тестування.
5. Визначте загальне призначення тестового плану.

6. Укажіть, хто з членів групи тестування відповідає за виконання тестів і реєстрацію помилок?
7. Охарактеризуйте процес документування тестових матеріалів.
8. Назвіть види звітів про виконання тестування.
9. Які документи використовуються для розроблення плану тестування?
10. Які документи використовуються для розроблення тестових специфікацій?
11. Який документ слугує для документування результатів виконання плану тестування?
12. Які існують документи щодо процесу тестування програмних продуктів?
13. Охарактеризуйте життєвий цикл помилки.
14. Назвіть особливості застосування різних критеріїв оцінювання швидкості виявлення помилок.
15. Поясніть поняття програмної помилки.
16. Визначте категорії програмних помилок.
17. Наведіть приклади помилки користувальницького інтерфейсу.
18. Як здійснюється оброблення помилок?
19. Визначте помилки, що пов'язані з обробленням граничних умов.
20. Визначте помилки передавання даних.
21. Визначте помилки оброблення даних.
22. У чому полягає перенавантаження?
23. Укажіть, хто з членів групи тестування відповідає за виконання тестів і реєстрацію помилок?
24. До якого класу тестів належить перевірка правильності роботи системи?
25. До якого класу тестів зараховують перевірку правильності роботи оброблення даних?

Тема 6. Проблеми якості програмного забезпечення

1. Назвіть проблеми якості програмного забезпечення.
2. Охарактеризуйте якість програмного забезпечення як необхідність.
3. У чому полягає наскрізне забезпечення якості програмного забезпечення.
4. Охарактеризуйте стандарти розроблення програмного забезпечення.

5. Назвіть методики забезпечення якості програмного забезпечення в рамках усього життєвого циклу програмного продукту.
6. У чому полягає сесійне тестування?
7. Охарактеризуйте тестування, що засноване на ризиках.
8. У чому полягає особливість тестування на основі специфікацій та сценаріїв?
9. Охарактеризуйте регресійне тестування.
10. Охарактеризуйте альфа-тестування.
11. Охарактеризуйте бета-тестування.
12. У чому полягає сутність релізу?
13. У чому полягає процес документування тестових процедур для ручних та автоматизованих тестів?
14. Назвіть основне призначення систем відстеження дефектів.
15. Визначте різновиди систем баг-трекінгу.
16. Наведіть структуру звіту про проблему.
17. Охарактеризуйте класи еквівалентності та граничні умови.
18. У чому полягає тестування функціональності?
19. У чому полягає тестування переходів між станами?
20. У чому полягає тестування випробування навантажень?

Рекомендована література

1. Авраменко А. С. Тестування програмного забезпечення : навч. посіб. / А. С. Авраменко, В. С. Авраменко, Г. В. Косенюк. – Черкаси : ЧНУ імені Богдана Хмельницького, 2018. – 284 с.
2. Дяченко М. П. Методичні матеріали щодо забезпечення самостійної роботи студентів з дисципліни "Методи та засоби тестування програмного забезпечення" (для освітньо-кваліфікаційного рівня "магістр") / М.П. Дяченко. – Київ : МАУП, 2018. – 35 с.
3. Коротенко Г. М. Методичні вказівки до практичних робіт по курсу "Тестування та верифікація ПЗ" / Г. М. Коротенко, Л. М. Коротенко, О. С. Шевцова; М-во освіти і науки України, Нац. техн. ун-т "Дніпровська політехніка". – Дніпро : НТУ "ДП", – 2020. – 62 с.
4. Крепич С. Я. Якість програмного забезпечення та тестування: базовий курс. Навчальний посібник для бакалаврів галузі знань 12 "Інформаційні технології" спеціальності 121 "Інженерія програмного забезпечення" / С. Я. Крепич, І. Я. Співак. – Тернопіль : ФОП Паляниця В. А., – 2020. – 478 с.
5. Ревенчук І. А. Методичні вказівки до лабораторних робіт з дисципліни "Якість та тестування програмного забезпечення" для студентів усіх форм навчання напряму 7.05010302 – "Інженерія програмного забезпечення" (ІПЗ) / І. А. Ревенчук, Т. С. Ткачова. – Харків : ХНУРЕ, 2018. – 41 с.
6. Скорін Ю. І. Якість програмного забезпечення та тестування [Електронний ресурс] : робоча програма для студентів спеціальності 121 "Інженерія програмного забезпечення" першого (бакалаврського) рівня / уклад. Ю. І. Скорін. – Харків : ХНЕУ ім. С. Кузнеця, 2019. – 11 с.
7. Скорін Ю. І. Якість програмного забезпечення та тестування [Електронний ресурс] : методичні рекомендації до самостійної роботи студентів спеціальності 121 "Інженерія програмного забезпечення" першого (бакалаврського) рівня / уклад. Ю. І. Скорін. – Харків : ХНЕУ ім. С. Кузнеця, 2019. – 48 с.
8. Скорін Ю. І. Технології тестування програмного забезпечення [Електронний ресурс] : робоча програма навчальної дисципліни для студентів спеціальності 122 "Комп'ютерні науки" першого (бакалаврського) рівня / уклад. Ю. І. Скорін. – Харків : ХНЕУ ім. С. Кузнеця, 2021. – 12 с.
9. Skorin Y. Introduction of electronic test tools with the aim of increasing the effectiveness of the knowledge test / Y. Skorin, O. Shcherbakov // Development Management. – Volume 4. – 2018. – Issue 4. – Pp. 14–25.

10. Аналіз якості програмного забезпечення [Електронний ресурс]. – Режим доступу : <http://oldconf.neasmo.org.ua/node/430>.
11. Аналіз якості програмного забезпечення [Електронний ресурс]. – Режим доступу : https://pidru4niki.com/1194121347734/informatika/analiz_yakosti_programnogo_zabezpechennya.
12. Гуманітарії в ІТ. Чому тестувальник – ідеальна професія для лінгвіста? [Електронний ресурс]. – Режим доступу : <https://training.epam.ua/#!/News/332?lang=ua>.
13. Дідковська М. В. Тестування: критерії та методи [Електронний ресурс] / М. В. Дідковська, Ю. О. Тимошенко. – Режим доступу : <https://www.quality-assurance-group.com/book/testuvannya-kriteriyi-ta-metodi>.
14. Дідковська М. В. Тестування. Основні визначення, аксіоми та принципи. Текст лекцій. Частина I [Електронний ресурс] / М. В. Дідковська, Ю. О. Тимошенко. – Режим доступу : <https://www.quality-assurance-group.com/book/testuvannya-osnovni-viznachennya-aksiomi-ta-printsipi-tekst-lektsij-chastina-i>.
15. Дідковська М. В. Текст лекцій до курсу "Технології розробки і тестування програм" [Електронний ресурс] / М. В. Дідковська. – Режим доступу : <http://mmsa.kpi.ua/disciplines/sac/14c>.
16. Добірка корисних подкастів для ІТ-фахівців [Електронний ресурс]. – Режим доступу : <https://training.epam.ua/#!/News/399?lang=ua>.
17. Захаркін О. З гуманітарія в інженери з контролю якості [Електронний ресурс] / О. Захаркін. – Режим доступу : <https://training.epam.ua/#!/-News/445?lang=ua>.
18. Крамаренко Я. Вступ до розробки програмного забезпечення [Електронний ресурс] / Я. Крамаренко. – Режим доступу : <https://ain.ua/2018/08/08/-beslatnaya-kniga-po-programmirvaniyu-na-ukrainskom>.
19. Львівська Школа тестування програмного забезпечення [Електронний ресурс]. – Режим доступу : <https://web.archive.org/web/20130620151529/http://www.qa-school.lviv.ua>.
20. Незнахіна В. Знайомство новачків QA, інсталяція та кастомізація [Електронний ресурс] / В. Незнахіна. – Режим доступу : <https://www.quality-assurance-group.com/book/os-linux>.
21. Незнахіна В. Основи продуктивності системи [Електронний ресурс] / В. Незнахіна. – Режим доступу : <https://www.quality-assurance-group.com/book/prezentatsiya-valentina-neznahina-osnovi-produktivnosti-sistemi>.

22. Основні характеристики якості програмного забезпечення [Електронний ресурс]. – Режим доступу : <https://www.youtube.com-/watch?v=fH7P4bUf7Xg>.

23. Подоба В. Веб-розробка з Python та Django для початківців. Практичний посібник по веб-розробці з допомогою мови програмування Python та фреймворку Django [Електронний ресурс] / В. Подоба. – Режим доступу : <https://www.quality-assurance-group.com/book/shpargalka-po-git-vid-vitaliya-podobi>.

24. Про тестування та тестувальників без міфів та стереотипів [Електронний ресурс]. – Режим доступу : <https://training.epam.ua/#!/News/318?lang=ua>.

25. Робоча програма навчальної дисципліни "Якість програмного забезпечення та тестування" [Електронний ресурс]. – Режим доступу : <https://ukd.edu.ua/sites/default/files/2020-03/YAkist-programnogo-zabezpechennya-ta-testuvannya.pdf>.

26. Таємниці успішного навчання від керівників найпопулярніших напрямів в EPAM [Електронний ресурс]. – Режим доступу : <https://training.epam.ua/#!/News/415?lang=ua>.

27. Тестування та якість програмного забезпечення [Електронний ресурс]. – Режим доступу : https://kpi-fict-ip32.github.io/Blog/s07/-software_quality.html.

28. Тестуємо професію тестувальника [Електронний ресурс]. – Режим доступу : https://onlineitea.com.ua/testing_m?utm_campaign=-qa-marathon&utm_source=google&utm_medium=QAUA01Keys08&gclid=Cj0KCQjwwY-LBhD6ARIsACvT72NVjT6GJWeEoboVghFdJster2X6bZC_siaXIOMRBQrSy9JW5awDVSoaApHEEALw_wcB.

29. Чим займаються різні спеціалісти на IT-проєктах: ролі, задачі та компетенції [Електронний ресурс]. – Режим доступу : <https://training.epam.ua/#!/News/267?lang=ua>.

30. Шлях тестувальника: з чого починати вивчення автоматизації [Електронний ресурс]. – Режим доступу : <https://training.epam.ua/#!/News/383?lang=ua>.

31. Що таке автоматизоване тестування, або чому я обираю автоматизацію? [Електронний ресурс]. – Режим доступу : <https://training.epam.ua/#!/News/384?lang=ua>.

32. Які бувають IT-компанії? [Електронний ресурс]. – Режим доступу : <https://training.epam.ua/#!/News/193?lang=ua>.

33. Якість програмного забезпечення [Електронний ресурс]. – Режим доступу : <https://infopedia.su/4x32c3.html>.
34. Якість програмного забезпечення та тестування [Електронний ресурс]. – Режим доступу : <http://ksuonline.kspu.edu/course/view.php?id=245>.
35. Якість програмного забезпечення та тестування [Електронний ресурс]. – Режим доступу : <https://zntu.edu.ua/yakist-programnogo-zabezpechennya-ta-testuvannya>.
36. Яку мову обрати для старту в QA Automation? [Електронний ресурс]. – Режим доступу : <https://training.epam.ua/#!/News/437?lang=ua>.
37. Agarwal B. B. Software Engineering [Electronic resource] / B. B. Agarwal, M. Gupta, S. P. Tayal. – Access mode : https://books.google.com.ua/books?id=CDWRq0-B9e5kC&hl=ru&source=gbs_similarbooks.
38. Aggarwal K. K. Software Engineering [Electronic resource] / K. K. Aggarwal. – Access mode : https://books.google.com.ua/books?id=dx2C9Z-kez5YC&hl=ru&source=gbs_similarbooks.
39. API, хмари та продуктивність: три сучасних тренди в автоматизації тестування [Електронний ресурс]. – Режим доступу : <https://training.epam.ua/#!/News/461?lang=ua>.
40. Automated Testing Meetup у Гродно [Електронний ресурс]. – Режим доступу : <https://training.epam.ua/#!/News/183?lang=ua>.
41. Automated Testing Tools – Automated Testing Tools [Electronic resource]. – Access mode : https://www.froglogic.com/squish/?gaid=644-934-3215&gclid=Cj0KCQjwkIGKBhCxAARIs-AINMiolbtibYPioNgJ2arX7yD1yCS-iJkRXFhE60JWcg-YMJqmPwo81WPLLQaAgPxEALw_wcB.
42. Axelrod A. Complete Guide to Test Automation: Techniques, Practices, and Patterns for Building and Maintaining Effective Software Projects [Electronic resource] / A. Axelrod. – Access mode : <https://www.amazon.com/-Complete-Guide-Test-Automation-Maintaining-ebook/dp/B07FKGVQP6>.
43. Bath G. The Software Test Engineer's Handbook, 2nd Edition: A Study Guide for the ISTQB Test Analyst and Technical Test Analyst Advanced Level Certificates [Electronic resource] / G. Bath. – Access mode : https://www.amazon.com/Software-Test-Engineers-Handbook-2nd-ebook/dp/B00VB46NWS/ref=sr_1_1?crid=29T27G1RCKZO4&keywords=the+software+test+engineer%27s+handbook&qid=1563031646&s=digital-text&sprefix=%E2%80%A2%09The+Software+-Test+Engineer%27s+-Handbook%2Cdigital-text%2C231&sr=1-1.

44. Bath G. Improving the Test Process: Implementing Improvement and Change - A Study Guide for the ISTQB Expert Level Module (Rocky Nook Computing) [Electronic resource] / G. Bath. – Access mode : https://www.amazon.com/gp/product/B00VB46L7A?notRedirectToSDP=1&ref_=db_s_mng_calw_1&storeType=ebooks.

45. Bath G. Software Testing Foundations, 4th Edition: A Study Guide for the Certified Tester Exam (Rocky Nook Computing) [Electronic resource] / G. Bath. – Access mode : https://www.amazon.com/gp/product/B00VB46K5S-?notRedirectToSDP=1&ref_=db_s_mng_calw_2&storeType=ebooks.

46. Beizer B. Black-Box Testing: Techniques for Functional Testing of Software and Systems [Electronic resource] / B. Beizer. – Access mode : <https://www.amazon.com/Black-Box-Testing-Boris-Beizer/dp/0471120944>.

47. Bhosale S. S. Software Engineering [Electronic resource] / S. S. Bhosale, V. I. Pujari, A. V. Patil. – Access mode : https://books.google.com.ua/books?id=NG1ZDwAAQBAJ&hl=ru&source=gbs_similarbooks.

48. Black R. Pragmatic Software Testing: Becoming an Effective and Efficient Test Professional [Electronic resource] / R. Black. – Access mode : <https://www.amazon.com/Pragmatic-Software-Testing-Effective-Professional-/dp/0470127902>.

49. Blundell P. Learning Android Application Testing [Electronic resource] / P. Blundell, D. T. Milano. – Access mode : https://www.amazon.co.jp/s?i=englishbooks&rh=p_27%3ADiego+Torres+Milano&s=relevancerank&language=en&text=Diego+Torres+Milano&ref=-dp_byline_sr_book_1.

50. Burnstein I. Practical Software Testing. Testing [Electronic resource] / I. Burnstein. – Access mode : <https://www.springer.com/gp/-book/9780387-951317>.

51. Copeland L. A Practitioner's Guide to Software Test Design Illustrated Edition [Electronic resource] / L. Copeland. – Access mode : <https://www.amazon.com/Practitioners-Guide-Software-Test-Design/dp/-158053791X>.

52. Copeland L. A Practitioner's Guide to Software Test Design (Artech House Computing Library) [Electronic resource] / L. Copeland. – Access mode : https://www.amazon.com/Practitioners-Software-Design-Computing-Library-ebook/dp/B001GS7030/ref=sr_1_1?crid=1F191GCZO775V-&keywords=a+practitioner%27s+guide+to+software+test+design&qid=1563031628&s=digitaltext&prefix=%E2%80%A2%09A+Practitioner%27s+Guide+to+Software+Test+Design%2Cdigital-text%2C901&sr=1-1.

53. Crispin L. Agile Testing: A Practical Guide for Testers and Agile Teams (Addison-Wesley Signature Series (Cohn)) [Electronic resource] / L. Crispin. – Access mode : <https://www.amazon.in/Agile-Testing-Practical-Testers-Signature/dp/0321534468>.

54. Culbertson R. Rapid Testing [Electronic resource] / R. Culbertson. – Access mode : <https://www.amazon.com/Rapid-Testing-Robert-Culbertson/dp/0130912948>.

55. Dosch W. Software Engineering Research and Applications [Electronic resource] / W. Dosch, Y. Lee Roger, Wu C. – Access mode : https://books.google.com.ua/books?id=-vRhGjEFuJXEC&hl=ru&source=gbs_-similarbooks.

56. Dustin E. Automated Software Testing: Introduction, Management, and Performance: Introduction, Management, and Performance [Electronic resource] / E. Dustin. – Access mode : https://books.google.com.ua/books/about/Automated_Software_Testing.html?id=kl2H0G6EFf0C&redir_esc=y.

57. EPAM University Online: Java, .NET, QA, DevOps и Front-End відтепер доступні з будь-якого куточку планети програміста [Електронний ресурс]. – Режим доступу : <https://training.epam.ua/#!/News/329?lang=ua>.

58. Foster E. Software Engineering [Electronic resource] / E. Foster. – Access mode : https://books.google.com.ua/books?id=RWMnCgAAQBAJ&hl=ru&source=gbs_similarbooks.

59. Graham D. Foundations of Software Testing: ISTQB Certification Revised Edition [Electronic resource] / D. Graham. – Access mode : <https://www.amazon.com/Foundations-Software-Testing-ISTQB-Certification-/dp/1844809897>.

60. Guide to the Software Engineering Body of Knowledge [Electronic resource]. – Access mode : <https://web.archive.org/web/20130509164903/http://www.computer.org/portal/web/swebok-/html/copyright>.

61. Hamilton T. Software Testing Techniques with Test Case Design Examples [Electronic resource] / T. Hamilton. – Access mode : <https://www.guru99.com/software-testing-techniques.html>.

62. Hendrickson E. Explore It!: Reduce Risk and Increase Confidence with Exploratory Testing Paperback [Electronic resource] / E. Hendrickson. – Access mode : <https://www.amazon.in/Explore-ElisabethHendrickson/dp/1937785025>.

63. Homes B. Fundamentals of Software Testing [Electronic resource] / B. Homes. – Access mode : <https://www.amazon.com/Fundamentals-Software-Testing-Bernard-Hom%C3%A8s/dp/1848213247>.

64. James K. L. Software Engineering [Electronic resource] / K. L. James. – Access mode : https://books.google.com.ua/books?id=dZAp-WJIIBT8C&hl=ru&source=gbs_similarbooks.

65. Kaner C. Lessons Learned in Software Testing: A Context-Driven Approach [Electronic resource] / C. Kaner. – Access mode : <https://www.amazon.com/Lessons-Learned-Software-Testing-Context-Driven-ebook/dp/B07MGKRDBY>.

66. Kaner C. Lessons Learned in Software Testing: A Context-Driven Approach [Electronic resource] / C. Kaner. – Access mode : <https://www.amazon.com/Lessons-Learned-Software-Testing-Context-Driven/dp/0471081124>.

67. Kaczanowski T. Practical Unit Testing with JUnit and Mockito [Electronic resource] / T. Kaczanowski. – Access mode : <https://www.amazon.com/Practical-Unit-Testing-JUnit-Mockito/dp/8393489393>.

68. Kassem A. S. Software Engineering [Electronic resource] / A. S. Kassem. – Access mode : https://books.google.com.ua/books?id=N69KPjBEWygC&hl=ru&source=gbs_similarbooks.

69. Knott D. Hands-On Mobile App Testing: A Guide for Mobile Testers and Anyone Involved in the Mobile App Business [Electronic resource] / D. Knott. – Access mode : <https://www.amazon.com/Hands-Mobile-App-Testing-Involved/dp/0134191714>.

70. Lee R. Y. Software Engineering: A Hands-On Approach [Electronic resource] / R. Y. Lee. – Access mode : https://books.google.com.ua/books?id=-zdBEAAAQBAJ&hl=ru&source=gbs_-similarbooks.

71. Mathew S. Software Engineering [Electronic resource] / S. Mathew. – Access mode : https://books.google.com.ua/books?id=-arXKB7jho4oC&hl=ru&source=gbs_book_similarbooks.

72. Meszaros G. xUnit Test Patterns: Refactoring Test Code [Electronic resource] / G. Meszaros. – Access mode : <https://www.amazon.com/xUnit-Test-Patterns-Refactoring-Code/dp/0131495054>.

73. Myers G. J. The Art of Software Testing [Electronic resource] / G. J. Myers. – Access mode : https://www.amazon.com/Art-Software-Testing-GlenfordMyers/dp/1118031962/ref=sr_1_1?crid=-3T53HAB5LVCV&keywords=art+of+software+testing&qid=1563031549&s=gateway&prefix=-Art+of+software+t%2Caps%2C221&sr=8-1.

74. Nguyen H. Q. Testing Applications on the Web: Test Planning for Internet-Based Systems [Electronic resource] / H. Q. Nguyen. – Access mode : https://books.google.com.ua/books?id=nBqLi_dIBR0C&hl=ru&source=gbs_book_similarbooks.

75. Page A. How We Test Software at Microsoft [Electronic resource] / A. Page. – Access mode : <https://www.amazon.com/How-We-Test-Software-Microsoft/dp/0735624259>.

76. Page A. How We Test Software at Microsoft [Electronic resource] / A. Page. – Access mode : <https://www.amazon.com/How-We-Test-Software-Microsoft/dp/0735624259/ref=sr11?crd=1ZNIS8XLJBG9D&keywords=how+we+test+software+at+microsoft&qid=1563031492&s=gateway&sprefix=How+we+tes%2Caps%2C223&sr=8-1>.

77. Paperback P. Digital Logic and Computer Design [Electronic resource] / P. Paperback. – Access mode : <https://www.amazon.com/Digital-Logic-Computer-Design-Morris/dp/933254252X>.

78. QA Automation: тестування чи програмування? [Електронний ресурс]. – Режим доступу : <https://training.epam.ua/#!/News/343?lang=ua>.

79. QA Club Kiev - Киевское сообщество тестировщиков [Электронный ресурс]. – Режим доступа : <http://www.qaclubkiev.com/2015/03/q-u-ality.html>.

80. Rajendra P. Software Engineering [Electronic resource] / P. Rajendra. – Access mode : https://books.google.com.ua/books?id=-li5QDAAQBAJ&hl=ru&source=gbs_similarbooks.

81. Rajput S. Software Engineering [Electronic resource] / S. Rajput. – Access mode : https://books.google.com.ua/books?id=ee5JDwAAQBAJ&hl=ru&source=gbs-_similarbooks.

82. Riley T. Beautiful Testing: Leading Professionals Reveal How They Improve Soft-ware (Theory in Practice) [Electronic resource] / T. Riley. – Access mode : <https://www.amazon.com/Beautiful-Testing-Professionals-Software-Practice/dp/0596159811>.

83. SAP Development training [Електронний ресурс]. – Режим доступу : <https://training.epam.ua/#!/News/177?lang=ua>.

84. Stottlemeyer D. Automated Web Testing Toolkit: Expert Methods for Testing and Managing Web Applications [Electronic resource] / D. Stottlemeyer. – Access mode : https://books.google.com.ua/books?id=Ur5QAAAAMAAJ&hl=ru&source=gbs_similarbooks.

85. Software Testing and Analysis: Process, Principles, and Techniques [Electronic resource]. – Access mode : <https://ix.cs.uoregon.edu/~michal/book>.

86. Software Testing and Quality Assurance: Theory and Practice [Electronic resource]. – Access mode : <https://www.wiley.com/en-us/Software+Testing+and+Quality+Assurance%3A-+Theory+and+Practice-p-9781118211632>.

87. Software Testing: підготовка до відбіркових етапів [Електронний ресурс]. – Режим доступу : <https://training.epam.ua/#!/News/444?lang=ua>.
88. Software Testing: Матеріали для підготовки [Електронний ресурс]. – Режим доступу : <https://training.epam.ua/#!/News/97?lang=ua>.
89. Software VS Hardware: що зараз в тренді і за ким майбутнє? [Електронний ресурс]. – Режим доступу : <https://training.epam.ua/#!/News/312?lang=ua>.
90. Sundar D. Software Engineering [Electronic resource] / D. Sundar. – Access mode : https://books.google.com.ua/books?id=kU_goyAq6hAC&hl=ru&source=gbs_similarbooks.
91. Standard for Software Component Testing [Electronic resource]. – Access mode : <http://www.testingstandards.co.uk/Component%20Testing.pdf>.
92. Test Automation в університетах від EPAM University Ambassador [Електронний ресурс]. – Режим доступу : <https://training.epam.ua/#!/News/264?lang=ua>.
93. Whittaker J. A. Exploratory Software Testing: Tips, Tricks, Tours, and Techniques to Guide Test Design [Electronic resource] / J. A. Whittaker. – Access mode : <https://www.amazon.com/Exploratory-Software-Testing-Tricks-Techniques/dp/0321636414>.
94. Whittaker J. How to Break Software: A Practical Guide to Testing [Electronic resource] / J. Whittaker. – Access mode : https://www.amazon.ca/s?i=stripbooks&rh=p_27%3AJames+Whittaker&s=relevancerank&text=James+Whittaker&ref=dp_byline_sr_book_1.

Зміст

Вступ.....	3
1. Програма навчальної дисципліни	14
2. Методичні рекомендації щодо підготовки до лекційних занять, роботи з рекомендованою літературою.....	15
2.1. Перелік тем презентацій.....	17
3. Методичні рекомендації щодо підготовки до виконання практичних завдань	21
3.1. Перелік тем есе.....	22
4. Методичні рекомендації щодо підготовки до поточного тестового контролю (контрольної роботи).....	24
4.1. Тести.....	26
5. Контрольні запитання для самодіагностики	32
Рекомендована література.....	37

НАВЧАЛЬНЕ ВИДАННЯ

ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

**Методичні рекомендації
до самостійної роботи студентів
спеціальності 122 "Комп'ютерні науки"
першого (бакалаврського) рівня**

Самостійне електронне текстове мережеве видання

Укладач **Скорін Юрій Іванович**

Відповідальний за видання *І. О. Ушакова*

Редактор *В. О. Дмитрієва*

Коректор *В. Ю. Труш*

План 2022 р. Поз. 91 ЕВ. Обсяг 47 с.

Видавець і виготовлювач – ХНЕУ ім. С. Кузнеця, 61166, м. Харків, просп. Науки, 9-А

*Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру
ДК № 4853 від 20.02.2015 р.*