

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**

Затверджено на засіданні кафедри інформаційних систем

Самостійне електронне текстове мережеве видання

Укладач Ю. І. Скорін

Тестування програмного забезпечення : методичні рекомендації до самостійної роботи студентів спеціальності 122 "Комп'ютерні науки" першого (бакалаврського) рівня : [Електронне видання] / уклад. Ю. І. Скорін. – Харків : ХНЕУ ім. С. Кузнеця, – 48 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.

© Харківський національний економічний
університет імені Семена Кузнеця

Вступ

Сьогодення характеризується тим, що сучасні економічні умови господарювання вимагають від фахівців, незалежно від їх спеціалізації, всебічного використання новітніх інформаційних технологій, комп'ютеризованих засобів збору, обробки 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.

Проектування і створення програмного забезпечення вважаються досить молодими сферами діяльності, проте вони ростуть швидше, ніж будь-які інші. Індустрія програмного забезпечення в даний час вважається однією з головних опор економічного зростання в багатьох країнах. Компанії, що виробляють програмне забезпечення, прагнучі до досягнення задоволеності клієнтів, часто стикаються зі складними проблемами поставки високоякісних продуктів.

З тих пір 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 проектів 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, які 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 розробки програмного продукту, який входить до системи забезпечення якості програмного продукту.

Система якості повинна являти собою єдиний процес, що проходить через весь життєвий цикл програмного продукту, гарантуючи, тим самим, що якість формується в ході розробки, а не раптом виявляється в кінці всього процесу. Упор необхідно робити на попередження появи дефектів, а не на їх виправлення після виникнення.

Якість програмного забезпечення це сукупність властивостей, які 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, крім того, користувач хоче мати гарантії, що він може довіряти певній програмі свої дані.

Реалізація вимог безпеки під час створення програмного забезпечення є однією із складових частин 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, таких, як замовник продукту, спонсор, кінцевий користувач, розробники та тестери продукту, інженери підтримки, співробітники 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 задачу визначення зацікавлених осіб, їх критеріїв якості і потім знаходження оптимального рішення, що задовольняє цим критеріям.

Тестування є одним із найбільш усталених способів забезпечення якості розробки програмного забезпечення та входить в набір ефективних засобів сучасної системи забезпечення якості програмного продукту. З технічної точки зору 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 додатка замовленим властивостям.

Як одна з основних фаз процесу розробки 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 розробки 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 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 включає [74]:

- опрацювання лекційного матеріалу;
- доопрацювання лекційного матеріалу;
- опрацювання та вивчення рекомендованої літератури, основних термінів та понять за темами дисципліни;
- підготовку до лабораторних занять;
- поглиблене опрацювання окремих лекційних тем або питань;
- пошук (підбір) та огляд літературних джерел за заданою проблематикою 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 промислової розробки програмного забезпечення під час організації самостійної роботи студентів

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

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

У даному підході застосовуються активні форми навчання, а саме імітується процес розробки програмного забезпечення, прийнятий в ІТ галузі, тим самим процес навчання наближається до умов реального ринку праці [5].

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.

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

1. Автор. У завдання автора входить формулювання завдання для розробки студентом.

2. Модератор. У завдання модератора входить призначення завдання на виконавців, контроль процесу роботи над завданням і винесення остаточного рішення про результат її 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 задачу без зауважень. Це означає, що інспектору абсолютно зрозумілі всі програмні рішення, код написаний правильно, текст програми забезпечений всіма необхідними коментарями;

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

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

5. Тестер. У завдання тестера входить 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 задачі.

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 якої необхідно написати програму, і передає її кодувальнику;

модератор призначає ролі;

кодувальник пише програму і передає її інспектору;
інспектор аналізує програмний код і виносить 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 з усунення зауважень до винесення рішення "Прийняти без зауважень";

тестер розробляє сценарій і перевіряє якість програмного продукту;
модератор виносить остаточне рішення про якість програмного продукту і оцінює результати роботи.

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

Таблиця 1

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

Роль	Учасник	Функція
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 і 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 варіанту традиційної форми процесу навчання.

Таблиця 2

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

Роль	Учасник	Функція
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 і 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 пропонуваної технології навчання.

**Розподіл ролей між учасниками освітнього процесу,
пропонована технологія навчання з використанням методів про-
грамної інженерії (варіант 1)**

Роль	Учасник	Функція
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 і 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 виносить оцінку кожному учаснику.

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

**Розподіл ролей між учасниками освітнього процесу,
пропонована технологія навчання з використанням методів про-
грамної інженерії (варіант 2)**

Роль	Учасник	Функція
Автор	Студент 1	Формулювання завдання
Модератор	Викладач	Визначення ролі кожного зі студентів, розподіл за- вдань 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 і її тестування

У даному варіанті роль автора виконують студенти.

Застосування описаної технології в процесі вивчення цієї дисципліни, а саме під час організації самостійної роботи студентів дозволяє підвищити ефективність навчального процесу з точки зору набуття студентами компетенцій в області тестування програмних продуктів.

Крім цього, її використання дозволяє підвищити мотивацію студентів до вивчення дисципліни, познайомити їх з прийомами колективної роботи, які застосовуються в професійному середовищі і забезпечити прозорість виставлення оцінок з дисципліни [23].

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	Література
1	2	4	5
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 розробки програмного продукту"	Презентація результатів, захист лабораторної роботи	Основна: [2; 13; 25]. Додаткова: [27 – 34]
<i>Тема 3.</i> Класифікація видів тестування	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]

Закінчення табл. 2.1

1	2	4	5
Модуль 2. Організація процесу тестування програмного забезпечення			
Тема 4. Особливості індустріального тестування	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]
Тема 5. Регресійне тестування	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]
Тема 6. Проблеми якості програмного забезпечення	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

Змістовний модуль 1.

Основи тестування програмного забезпечення

Тема 1. Тестування як спосіб забезпечення якості програмного продукту

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

1.2. Мотивація до вивчення технологій тестування. Історія розвитку тестування 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. Розробка тестів. Аналіз тестових випадків. Виконання тестових випадків. Оцінка результатів виконання програми на тестах. Етапи тестування. Керуючий 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 чорного ящика. Тестування білого ящика.

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. Псевдоотладка і мутаційні тестування.

Змістовний 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 розробки тестів. Тестування специфікацій. Тестування сценаріїв. Ручна розробка тестів. Генерація тестів. Виконання тестів.

4.3. Документація и супровід тестів. Тестові процедури. Опис тестів. Тестові процедури. Тестовий звіт. Оцінка 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 при відборі. Можливості повторного використання тестів. Класифікація вибірових 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 розробки програмного забезпечення та їх використання.

6.2. Методики забезпечення якості програмного забезпечення в рамках усього життєвого циклу розробки. 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 і оволодівати навичками творчої роботи.

Запис лекцій – складний вид вузівської аудиторної роботи. 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 по можливості власними формулюваннями.

Бажано здійснювати запис на одній сторінці, а наступну залишати для опрацювання навчального матеріалу самостійно. Конспект лекції краще поділяти 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 [70]:

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;

встановіть 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, суть 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 самотійно 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 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.

Самотійна 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 знайомство 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 [70].

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 і 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 розробки 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 і автоматизованого тестування.
23. Документація и 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 розробки програмного забезпечення та їх використання.
5. Методики забезпечення якості програмного забезпечення в рамках усього життєвого циклу розробки.
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 розробки 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 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 1-го до 2-х) за самостійну підготовку до практичних занять студент може отримати за умови якісного виконання самих завдань, а також за підготовку есе за матеріалами, які не надійшли 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 викладання думок за темою у вільній формі, тобто есе це літературна форма, невеликий прозовий текст, 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 передачі 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 стандарти 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 обробки 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 розробки 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 вузівського рівня.

Успіх і якість виконання письмових робіт обумовлені обізнаністю студента в навчальному плані (за курсами навчання), у змісті освітніх програм з дисциплін, його вміння доцільно планувати й організовувати 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 дисципліни.

Для формування завдань використана закрита 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, такі, як:

роздавання 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. Перелік питань тесту

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

- a) керівник групи тестування;
- b) тест аналітик;
- c) тест розробник;
- d) виконавець тестів.

2. Вкажіть хто з членів групи тестування відповідає за розробку тестових специфікацій та аналіз результатів тестування:

- a) керівник групи тестування;
- b) тест аналітик;
- c) тест розробник;
- d) виконавець тестів.

3. Виберіть правильне твердження:

- a) методи тестування описуються у Плані тестування (Test plan);
- b) методи тестування описуються в Тестовій специфікації (Test case specification);
- c) методи тестування описуються в Тестових процедурах (Test procedure specification).

4. Вкажіть, які документи використовуються для розробки Плану тестування (Test plan):

- a) технічне завдання;
- b) тестові специфікації (test case specification);
- c) тестові процедури (Test procedure specification);
- d) підсумковий звіт тестування (Test summary report).

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

- a) структурні критерії;
- b) функціональні критерії;
- c) критерії стохастичного тестування;
- d) мутаційні критерії?

6. На якому етапі тестування застосовуються структурні критерії:

- a) регресійне тестування;
- b) модульне тестування;
- c) інтеграційне тестування;
- d) системне тестування?

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

- a) структурні критерії;
- b) функціональні критерії;
- c) критерії стохастичного тестування;
- d) мутаційні критерії?

8. Яка модель тестування використовується у процесі функціонального тестування:

- a) чорний ящик;
- b) білий ящик?

9. Яка модель тестування використовується у ході модульного тестування:

- a) чорний ящик;
- b) білий ящик?

10. Яка модель тестування використовується у процесі інтеграційного тестування:

- a) чорний ящик;
- b) білий ящик?

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

- a) системним тестуванням;
- b) модульним тестуванням;
- c) інтеграційним тестуванням;
- d) тестуванням інтерфейсів;
- e) стрес-тестуванням;
- f) навантажувальним тестуванням.

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

- a) системне тестування;
- b) функціональне тестування;
- c) GUI-тестування;
- d) UNIT-тестування;
- e) стрес-тестування;
- f) навантажувальне тестування?

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

- a) системне тестування;
- b) функціональне тестування;
- c) GUI-тестування;
- d) UNIT-тестування;
- e) стрес-тестування;
- f) навантажувальне тестування?

14. Для перевірки працездатності окремих елементів системи використовується:

- a) системне тестування;
- b) функціональне тестування;
- c) GUI-тестування;
- d) UNIT-тестування;
- e) стрес-тестування;

f) навантажувальне тестування.

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

- a) системне тестування;
- b) функціональне тестування;
- c) GUI-тестування;
- d) UNIT-тестування;
- e) стрес-тестування;
- f) навантажувальне тестування?

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

- a) системне тестування;
- b) функціональне тестування;
- c) GUI-тестування;
- d) UNIT-тестування;
- e) стрес-тестування;
- f) навантажувальне тестування.

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

- a) системне тестування;
- b) функціональне тестування;
- c) GUI-тестування;
- d) UNIT-тестування;
- e) стрес-тестування;
- f) навантажувальне тестування?

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

- a) це є помилкою;
- b) це не є помилкою.

19. У модальному вікні розташована кнопка "Застосувати":

- a) це є помилкою;

b) це не є помилкою.

20. Чи є помилкою наявність піктограм на застосованих кнопках:

a) так;

b) ні.

21. У вікні реєстрації користувача при виборі в полі "LOGIN" імені user зникає поле "PASSWORD":

a) це є помилкою;

b) це не є помилкою.

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

a) це є помилкою;

b) це не є помилкою.

23. Чи є обмеження на ширину поля зі списком:

a) так;

b) ні?

24. Список містить більше 7 елементів, при цьому список не відсортований:

a) це є помилкою;

b) це не є помилкою.

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

a) це є помилкою;

b) це не є помилкою.

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

a) це є помилкою;

b) це не є помилкою.

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

- a) це є помилкою роботи додатка;
- b) додаток працює правильно.

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

- a) це є помилкою роботи додатка;
- b) додаток працює правильно.

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

- a) це є помилкою роботи додатка;
- b) додаток працює правильно.

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

- a) це є помилкою роботи додатка;
- b) додаток працює правильно.

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

- a) керівник групи тестування;
- b) тест аналітик;
- c) тест розробник;
- d) виконавець тестів.

32. Вкажіть, хто з членів групи тестування відповідає за розробку тестової документації:

- a) керівник групи тестування;
- b) тест аналітик;
- c) тест розробник;
- d) виконавець тестів.

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

- a) керівник групи тестування;
- b) тест аналітик;
- c) тест розробник;
- d) виконавець тестів.

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

- a) керівник групи тестування;
- b) тест аналітик;
- c) тест розробник;
- d) виконавець тестів.

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

- a) керівник групи тестування;
- b) тест аналітик.

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 дисципліни?
6. Вивчення яких навчальних дисциплін забезпечує дана дисципліна?
7. Наведіть складові якості програмного забезпечення.
8. Охарактеризуйте стандарт ISO.

9. Вкажіть, хто з членів групи тестування відповідає за розробку тестових специфікацій та аналіз результатів тестування?

10. Вкажіть, 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 оцінці 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 розробки 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 при проходженні 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 розробки програмного забезпечення.
4. Визначте особливості тестування на етапі планування.
5. Визначте особливості тестування на етапі проектування.
6. У чому полягає автоматизація тестування?
7. Вкажіть, хто з членів групи тестування відповідає за розробку 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 віднести тестування інтерфейсів (екранів, кнопок тощо)?

Тема 4. Особливості індустріального тестування

1. Обґрунтуйте вибір критеріїв якості тестування.
2. Охарактеризуйте статичне і динамічне тестування. В 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 розробка тестових сценаріїв та процедур?
3. У чому полягає виконання тестів?
4. Назвіть компоненти плану тестування.
5. Визначте загальне призначення тестового плану.
6. Вкажіть, 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 розробки плану тестування?
10. Які документи використовуються для розробки тестових специфікацій?
11. Який документ служить 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 оцінки швидкості виявлення помилок.
15. Поясніть поняття програмної помилки.
16. Визначте категорії програмних помилок.
17. Наведіть приклади помилки користувальницького інтерфейсу.
18. Як здійснюється обробка помилок?
19. Визначте помилки, що пов'язані з обробкою граничних умов.
20. Визначте помилки передачі даних.
21. Визначте помилки обробки даних.
22. У чому полягає перенавантаження?
23. Вкажіть, 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 відносять перевірку правильності роботи обробки 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 розробки 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. В 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 і граничні умови.

18. У чому полягає тестування функціональності?

19. У чому полягає тестування переходів між станами?

20. У чому полягає тестування випробування навантажень?

6. Рекомендована література

6.1. Основна

1. Бейзер Б. Тестирование чёрного ящика. Технология функционального тестирования / Б. Бейзер. – Санкт-Петербург, 2004. – 221 с.
2. Бек К. Экстремальное программирование: планирование / К. Бек. – Санкт-Петербург : Питер, 2003. – 143 с.
3. Богданов Д. В. Стандартизация процессов обеспечения качества программного обеспечения / Д. В. Богданов, В. А. Путилов, В. В. Фильчаков. – Апатиты : КФ ПетрГУ, 1988. – 152 с.
4. Брауде Э. Технология разработки программного обеспечения / Э. Брауде. – Санкт-Петербург : Питер, 2004. – 410 с.
5. Глухих М.И. Программная инженерия. Обеспечение качества программных средств методами статического анализа. Учебное пособие / М.И. Глухих. – СПб: Изд-во Политехн. ун-та. 2011, 150 с.
6. Гребенюк В. М. Оценка целесообразности внедрения автоматизированного тестирования. Институт Государственного управления, права и инновационных технологий (ИГУПИТ) / В. М. Гребенюк // Интернет-журнал «НАУКОВЕДЕНИЕ» №1 2013.
7. Гриняк В. М. Использование методов программной инженерии в процессе обучения дисциплинам типа "Программирование" / В. М. Гриняк, Н. Л. Слугина // Современные проблемы науки и образования, 2013. – № 2.
8. Дастин Элфрид. Автоматизированное тестирование программного обеспечения / Элфрид Дастин, Джефф Рэшка, Джон Пол. – Москва : Изд. "ЛОРИ", 2003. – 567 с.
9. Джефф Рэшка. Автоматизированное тестирование программного обеспечения. Внедрение, управление, эксплуатация / Рэшка Джефф. – Издательство Лори, 2012. – 578 с.
10. Калбертсон Роберт. Быстрое тестирование / Роберт Калбертсон, Крис Браун, Гэри Кобб. – Москва : Вильямс, 2002. – 374 с.
11. Канер Сэм. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений / Сэм Канер, Джек Фолк, Енг Кек Нгуен. – Киев : ДиаСофт, 2001. – 544 с.
12. Ковальов А. И. Управление проектом по созданию интернет-сайта / А. И. Ковальов. – Москва : Альпина, 2004. – 338 с.

13. Котляров В. П. Основы тестирования программного обеспечения : учеб. пособ. / В. П. Котляров, Т. В. Коликова. – Москва : Интернет-Университет Информационных Технологий. БИНОМ. Лаборатория знаний, 2006. – 285 с.
14. Криспин Лайза. Гибкое тестирование: практическое руководство для тестировщиков ПО и гибких команд / Лайза Криспин, Джанет Грегори. – Москва : Вильямс, 2010. – 464 с.
15. Липаев В. В. Выбор и оценивание характеристик качества программных средств: Методы и стандарты / В. В. Липаев; Моск. госуд. технологич. ин-т "Станкин". – Москва : Синтег, 2001. – 224 с.
16. Липаев В. В. Качество программных средств / В. В. Липаев. – Москва : Янус-К, 2012. – 399 с.
17. Липаев В. В. Программная инженерия. Методологические основы / В. В. Липаев. – Москва : ТЕИ, 2006. – 608 с.
18. Липаев В. В. Сертификация программных средств : учебник / В. В. Липаев. – Москва : СИНТЕГ, 2010. – 344 с.
19. Липаев В. В. Тестирование программ / В. В. Липаев. – Москва : Радио и связь, 1986. – 296 с.
20. Липаев В. В. Управление разработкой ПС: методы, стандарты, технология / В. В. Липаев. – Москва : Фин. и Ст., 1993. – 160 с.
21. Майерс Г. Искусство тестирования программ / Г. Майерс. – Москва : Финансы и статистика, 1982. – 176 с.
22. Макгрегор Джон. Тестирование объектно-ориентированного программного обеспечения / Джон Макгрегор, Дэвид Сайкс. – Киев : ООО "ТИД ДС", 2002. – 432 с.
23. Рекс Блек. Ключевые процессы тестирования / Блек Рекс. – Москва : Издательство Лори, 2014. - 544 с.
24. Основы инженерии качества программных систем / Ф. И. Андон, Г. И. Коваль Т. М., Коротун и др.; НАН Украины. Ин-т прогр. систем. – Киев : Изд. дом "Академпериодика", 2002. – 502 с.
25. Савин Р. Тестирование Дот Ком или Пособие по жестокому обращению с багами в интернет-стартапах / Р. Савин. – Москва : Дело, 2007. – 312 с.
26. Синицын С. В. Верификация программного обеспечения / С. В. Синицын, Н. Ю. Налютин. – Москва : БИНОМ, 2008. – 368 с.
27. Слугина Н. Л. Повышение уровня подготовки кадров в области современных информационных технологий на базе центра компетенций

/ Н. Л. Слугина, В. М. Гриняк // Современные проблемы науки и образования. – 2012. – № 5. – с. 205.

28. Соммервилл Иан. Инженерия программного обеспечения / Иан Соммервилл. – 6-е изд. – Москва : Вильямс, 2002. – 642 с.

29. Тамре Л. Введение в тестирование программного обеспечения / Л. Тамре. – Москва : Издательский дом "Вильямс", 2003. – 368 с.

30. Уиттакер Д. Как тестирует Google / Д. Уиттакер, Д. Арбон, Д. Каролло. – Санкт-Петербург : Издательский дом "Питер", 2014. – 320 с.

6.2. Додаткова

31. Боэм Б. Характеристики качества программного обеспечения / Б. Боэм, Дж. Браун, Х. Каспар. – Москва : Мир, 1981. – 208 с.

32. Бураков В. В. Методика оценки качества программных средств / В. В. Бураков // Известия вузов. Приборостроение. – 2008. – Т. 51, № 1. – С. 13–18.

33. Гацко Г. Тестирование ПО как один из элементов системы качества / Г. Гацко // Открытые системы. – № 06. – 1998.

34. Гласс Р. Сопровождение программного обеспечения / Р. Гласс; под ред. Ю. А. Чернышева. – Москва : Мир, 1983. – 158 с.

35. Долгов Ю. А. Количественная оценка некоторых характеристик надежности программного обеспечения / Ю. А. Долгов, Т. Г. Данилина // Радіоелектронні і комп'ютерні системи. – №7 (19). – 2006. – с. 152–155.

36. Долгов Ю. А. Статистическое моделирование : учебник для вузов / Ю. А. Долгов. – изд. 2-е, доп. – Тирасполь : Полиграфист, 2011. – 352 с.

37. Домаратский А. Н. Управление улучшением стандартного процесса и качеством программных изделий / А. Н. Домаратский // Программные продукты и системы. – № 4. – 1998. – с. 20–24.

38. Иванова Г. С. Технология программирования : учеб. для вузов / Г. С. Иванова. – Москва : МГТУ им. Н. Э. Баумана, 2002. – 320 с.

39. Калбертсон Роберт. Быстрое тестирование / Роберт Калбертсон, Крис Браун, Гэри Кобб. – Москва : Вильямс, 2002. – 384 с.

40. Канер Сэм. Тестирование программного обеспечения / Сэм Канер. – Киев : Изд. "ДиаСофт", 2000. – 318 с.

41. Канер Сэм. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений / Сэм Канер, Джек Фолк, Енг Кек Нгуен. – Киев : Изд-во "ДиаСофт", 2001. – 544 с.
42. Калинина Л. Ю. Оценка качества программных продуктов / Л. Ю. Калинина // Качество. Инновации. Образование. – 2006. – № 4. – С. 52–55.
43. Кантор Марри. Управление программными проектами. Практическое руководство по разработке успешного программного обеспечения / Марри Кантор. – Москва : Вильямс, 2002. – 176 с.
44. Коваленко И. И. Оценка качества программных продуктов с использованием теории Демпстера-Шейфера / И. И. Коваленко, А. В. Швед // Наукові праці ЧДУ – Комп'ютерні технології. – 2011. – Т. 160. – № 148. – С. 22–26.
45. Когаловский М. Р. Энциклопедия технологий баз данных / М. Р. Когаловский. – Москва : Финансы и статистика, 2005. – 800 с.
46. Коул Дж. Принципы тестирования ПО / Дж. Коул, Т. Горэм // Открытые системы. – № 2. – 1998. – С. 34–37.
47. Котляров В.П. Основы тестирования программного обеспечения / В.П. Котляров. - М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2015. – 956 с.
48. Кречетов Н. Технологии оценки качества программных продуктов / Н. Кречетов // Computer Week. – № 25. – 1996. – С. 27–32, 42.
49. Криспин Лайза. Гибкое тестирование: практическое руководство для тестировщиков ПО и гибких команд / Лайза Криспин, Джанет Грегори. – Москва : Вильямс, 2010. – 464 с.
50. Луиза Тамре. Введение в тестирование программного обеспечения. Руководство / Тамре Луиза. - М.: Диалектика / Вильямс, 2018. – 600 с.
51. Майерс Гленфорд. Искусство тестирования программ / Гленфорд Майерс, Том Баджетт, Кори Сандлер. – 3-е издание. – Москва : Диалектика, 2012. – 272 с.
52. Макгрегор Джон. Тестирование объектно-ориентированного программного обеспечения. Практическое пособие / Джон Макгрегор, Девид Сайкс. – Киев : ООО "ТИД ДС", 2002. – 432 с.
53. Менчер Э. М. Обобщенная функция полезности / Э. М. Менчер // Радионуклеиды и тонирующие излучения в исследованиях по виноградарству. – Кишинев : Штиинца, 1983. – с. 104–118.

54. Рэшка Джефф. Тестирование программного обеспечения / Джефф Рэшка, Элфрид Дастин, Джон Пол. – Москва : Лори, 2012. – 568 с.

55. Скорін Ю. І. Методика кількісної оцінки ефективності роботи інженера із забезпечення якості програмного забезпечення / Ю. І. Скорін, О. В. Щербаков, Є. С. Луценко // Системи обробки інформації. Збірник наукових праць. – Вип. 4 (102). – том 1. Інформаційні технології та захист інформації. – Харків : ХУПС ім. І. Кожедуба. – 2012. – С. 81–84.

56. Скорин Ю. И. Модуль баг-трекинга при разработке программного обеспечения / Ю. И. Скорин, А. А. Пивень, В. В. Стадник // Системи обробки інформації : Збірник наукових праць. III Міжнародна науково-практична конференція "Проблеми і перспективи розвитку ІТ-індустрії". – Харків : ХУПС ім. І. Кожедуба. – 2011. – Вип. 7(97). –111 с.

57. Скорин Ю. И. Тестирование программного обеспечения / Ю. И. Скорин, А. А. Пивень // Системи обробки інформації : Збірник наукових праць. – Вип. 4 (102). – том 1. Інформаційні технології та захист інформації. Харків : ХУПС ім. І. Кожедуба. – 2012. – С. 56–58.

6.3. Інформаційні ресурси

58. Автоматизация тестирования [Электронный ресурс]. – Режим доступа : <http://www.performance-lab.ru/avtomatizacija-testirovanija>.

59. Автоматизированное тестирование [Электронный ресурс]. – Режим доступа: <https://gist.github.com/codedokode/a455bde7d0748c0a351a>.

60. Библиотека MSDN. Источник информации для разработчиков, использующих средства, продукты, технологии и службы корпорации Майкрософт [Электронный ресурс]. – Режим доступа : <http://msdn.-microsoft.com>.

61. Виды Тестирования [Электронный ресурс]. – Режим доступа : <http://www.protesting.ru/testing/types/sanity.html>.

62. Геннадий Алпаев. Учебник по TestComplete [Электронный ресурс]. – Режим доступа : <http://tctutorial.ru/>.

63. ИНТУИТ. Интернет университет информационных технологий [Электронный ресурс]. – Режим доступа : <http://intuit.ru>.

64. Качество программного обеспечения [Электронный ресурс]. – Режим доступа : <http://openquality.ru/main>.

65. КиберФорум. Форум начинающих и профессиональных программистов, системных администраторов, администраторов баз данных [Электронный ресурс]. – Режим доступа : <http://cyberforum.ru>.

66. Клуб программистов [Электронный ресурс]. – Режим доступа : <http://www.cyberguru.ru>.

67. Коломейченко А.С. Информационные технологии: учеб. пособие / А.С. Коломейченко, Н.В. Польшакова, О.В. Чеха. – Санкт-Петербург : Лань. [Электронный ресурс]. – Режим доступа: <https://e.lanbook.com/book/101862>.

68. Липаев В. В. Документирование сложных программных средств / В. В. Липаев [Электронный ресурс]. – Режим доступа : <http://www.computer-museum.ru/books/lipaev/>.

69. Липаев В. В. Надежность и функциональная безопасность комплексов программ реального времени / В. В. Липаев [Электронный ресурс]. – Режим доступа : <http://www.computer-museum.ru/books/lipaev/>.

70. Липаев В. В. Программная инженерия. Методологические основы. Предисловие к курсу / В. В. Липаев [Электронный ресурс]. – Режим доступа : <http://citforum.ru/SE/lipaev/preface.shtml>.

71. Липаев В. В. Программная инженерия в жизненном цикле программных средств / В. В. Липаев [Электронный ресурс]. – Режим доступа : <http://citforum.ru/SE/lipaev/>.

72. Липаев В. В. Проектирование и производство сложных заказных программных продуктов / В. В. Липаев [Электронный ресурс]. – Режим доступа : <http://www.computer-museum.ru/books/lipaev/>.

73. Липаев В. В. Сопровождение и управление конфигурацией сложных программных средств / В. В. Липаев [Электронный ресурс]. – Режим доступа : <http://www.computer-museum.ru/books/lipaev/>.

74. Липаев В. В. Стандарты, регламентирующие жизненный цикл сложных программных комплексов / В. В. Липаев [Электронный ресурс]. – Режим доступа : <http://www.pcweek.ru/year1998/N24/CP1251/Reviews/-chapt1.htm>.

75. Липаев В. В. Тестирование компонентов и комплексов программ / В. В. Липаев [Электронный ресурс]. – Режим доступа : <http://www.computer-museum.ru/books/lipaev/>.

76. Липаев В. В. Экономика производства программных продуктов / В. В. Липаев [Электронный ресурс]. – Режим доступа : <http://www.computer-museum.ru/books/lipaev/>.

77. Марченко Е. Что такое качество программного обеспечения? / Е. Марченко [Электронный ресурс]. – Режим доступа : <http://www.silicontaiga.ru/home.asp?artId=2862>.
78. Основные положения тестирования [Электронный ресурс]. – Режим доступа : <https://habrahabr.ru/post/110307/>.
79. Оценка эффективности автоматизации тестирования [Электронный ресурс]. – Режим доступа : <http://a1qa.ru/blog/otsenka-effektivnosti-avtomatizatsii-testirovaniya/>.
80. Панкратов Вячеслав. Разработка критериев анализа систем автоматизации тестирования [Электронный ресурс]. – Режим доступа : <http://www.it4business.ru/>.
81. Портал об автоматизированном тестировании ПО [Электронный ресурс]. – Режим доступа : <http://automated-testing.info>.
82. Портал специалистов по тестированию и обеспечению качества ПО [Электронный ресурс]. – Режим доступа : <http://software-testing.ru>.
83. Про тестинг [Электронный ресурс]. – Режим доступа : <http://www.protesting.ru>.
84. Самостоятельная работа студентов [Электронный ресурс]. – Режим доступа : <http://www.studfiles.ru/preview/3495098/>.
85. Сертификация программного обеспечения (ПО) [Электронный ресурс]. – Режим доступа : <http://www.nspru.ru/sertsoftware/>.
86. Силласте Г. Г. Самостоятельная работа студентов: методические рекомендации / Г. Г. Силласте [Электронный ресурс]. – Режим доступа : https://docviewer.yandex.ua/?url=http%3A%2F%2Fmeggush.ru%2Fsites%2Fdefault%2Ffiles%2Fmetod.rekom_sam.rab_dlya_studentov.pdf&name=metod.rekom_sam.rab_dlya_studentov.pdf&lang=ru&c=583593fd0dce&page=3.
87. Таганов А.И. Основы идентификации, анализа и мониторинга проектных рисков качества программных изделий в условиях нечеткости учеб.-метод. пособие. – Электрон. дан. – Москва : Горячая линия Телеком. [Электронный ресурс]. – Режим доступа : <https://e.lanbook.com/book/5244>.
88. Тестирование и качество ПО [Электронный ресурс]. – Режим доступа : <http://software-testing.ru>.
89. Тестирование программного обеспечения [Электронный ресурс]. – Режим доступа : [https://ru.wikipedia.org/wiki/test%](https://ru.wikipedia.org/wiki/test%20).
90. Технология разработки программного обеспечения : Учеб. пос. / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Виснадул. Под ред.

проф. Л.Г. Гагариной. – Москва : ИД ФОРУМ: НИЦ Инфра-М [Электронный ресурс]. – Режим доступа : <http://znanium.com/catalog.php?bookinfo=389963>.

91. Что такое Конфигурационное тестирование [Электронный ресурс]. – Режим доступа : <http://software-testing.org/testing/chto-takoe-konfiguracionnoe-testirovanie-configuration-testing.html>.

6.4. Методичне забезпечення

92. Робоча програма навчальної дисципліни "Технології тестування програмних продуктів" для студентів за напрямом підготовки 6.050101 "Комп'ютерні науки" усіх форм навчання. Навчальне видання / уклад. Ю.І. Скорін. – Харків : Вид. ХНЕУ ім. С. Кузнеця, 2012. – 48 с.

93. Робоча програма навчальної дисципліни "Технології тестування програмних продуктів" для студентів напряму підготовки "Комп'ютерні науки" всіх форм навчання. Навчальне видання / уклад. Ю.І. Скорін. – Харків : Вид. ХНЕУ, 2013. – 40 с.

94. Робоча програма навчальної дисципліни "Технології створення програмних продуктів" для студентів напряму підготовки 6.050101 "Комп'ютерні науки" денної форми навчання [Електронне видання] / уклад. Скорін Ю.І. – Харків : ХНЕУ ім. С. Кузнеця, 2016. – 73 с.

95. Технології тестування програмних продуктів [Електронне видання] : методичні рекомендації до самостійної роботи студентів спеціальності 122 "Комп'ютерні науки та інформаційні технології" першого (бакалаврського) рівня / уклад. Скорін Ю.І. – Харків : ХНЕУ ім. С. Кузнеця, 2016. – 35 с.

96. 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.

97. 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.

Зміст

Вступ.....	3
1. Програма навчальної дисципліни.....	15
2. Методичні рекомендації щодо підготовки до лекційних занять, роботи з рекомендованою літературою.....	17
2.1. Перелік тем презентацій.....	24
3. Методичні рекомендації щодо підготовки до виконання практичних завдань	23
3.1. Перелік тем есе.....	24
4. Методичні рекомендації щодо підготовки до поточного тестового контролю (контрольної роботи).....	26
4.1. Перелік питань тесту.....	28
5. Контрольні запитання для самодіагностики	34
6. Рекомендована література.....	39
6.1. Основна	39
6.2. Додаткова	41
6.3. Інформаційні ресурси.....	43
6.4. Методичне забезпечення	46

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 *Ушакова І.О.*

Редактор

Коректор

План 2022 р. Поз.

Обсяг 48 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.*