| Must have рівень:   1. Зроби порівняння статичних та динамічних технік тестування. Наведи переваги та можливі обмеження при використанні кожної з них.  |  | Статистична техніка тестування | Динамічна техніка тестування | | --- | --- | --- | | Основна інформація | Перевірка будь-яких робочих документів, створених у процесі розробки ПЗ.  Етап валідації ПЗ | Перевірка шляхом фактичного використання програми і визначення того, чи працює функціональність так, як очікується.  Етап верифікації ПЗ | | Перевага №1 | Запобігаються помилки на ранній стадії розробки. | Процес динамічного тестування добре налагоджений, додаток тестується з точки зору користувача, що підвищує якість ПЗ. | | Перевага №2 | Скорочується час розробки. | Виявляються складні помилки, які могли вислизнути на етапі код рев'ю. | | Перевага №3 | Менші витрати на тестування та час тестування. | Динамічне тестування може бути автоматизовано за допомогою спеціальних інструментів. | | Перевага №4 | Огляди не тільки допомагають знайти помилки, але й знайти відсутні вимоги, недоліки дизайну та поганий код. | Динамічне тестування може застосовуватися до будь-якої програми | | Перевага №5 | Через покращення дизайну та коду — підвищення продуктивності розробки. | . | | Обмеження №1 | Процес може займати багато часу. | Оскільки динамічне тестування являє собою складний процес, воно займає багато часу. | | Обмеження №2 | Перевірка коду та документації не дає повного уявлення про дизайн програми, і головне його юзабіліті. | Висока вартість проведення тестування. | | Обмеження №3 |  | Динамічне тестування зазвичай виконується після завершення кодування, і знайдені баги виявляються пізніше в життєвому циклі розробки. | | Висновок | Це “теоретична” робота з усією документацією проєкта з метою пошуку помилок.  Статичне тестування не є альтернативою динамічному тестуванню, оскільки в обох випадках знайдуться різні помилки. | Це практична робота з системою з метою пошуку та усунення багів. | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Середній рівень:   1. Виконай завдання попереднього рівня. 2. Наступне твердження стосується покриття рішень:   *Коли код має одну ‘IF” умову, не має циклів (LOOP) або перемикачів (CASE), будь-який тест, який ми виконаємо, дасть результат 50% покриття рішень (decision coverage).*  Яке твердження є коректним?   * 1. Коректно. Будь-який тест кейс надає 100% покриття тверджень, таким чином покриває 50% рішень.   2. **Коректно. Результат будь-якого тесту умови IF буде або правдими, або ні.**   3. Некоректно. Один тест може гарантувати 25% перевірки рішень в цьому випадку.   4. Некоректно, бо занадто загальне твердження. Ми не можемо знати, чи є воно коректним, бо це залежить від тестованого ПЗ.  1. Є псевдокод: Switch PC on -> Start MS Word -> IF MS Word starts THEN -> Write a poem -> Close MS Word.   Скільки тест кейсів знадобиться, щоб перевірити його функціонал?   * 1. **1 – для покриття операторів, 2 – для покриття рішень**   2. 1 – для покриття операторів, 1 – для покриття рішень   3. 2 – для покриття операторів, 2 – для покриття рішень   4. 2 – для покриття операторів, 1 – для покриття рішень  1. Скільки потрібно тестів для перевірки тверджень коду:     1. 2    2. **1**    3. 3    4. 4 |
| Програма максимум:   1. Виконай завдання двох попередніх рівнів. 2. Продовжуємо розвивати стартап для застосунку, який дозволяє обмінюватися фотографіями котиків.   Є алгоритм:  *Запитай, якого улюбленця має користувач.*  *Якщо користувач відповість, що має кота, то запитай, яка порода його улюбленця: «короткошерста чи довгошерста?»*  *Якщо клієнт відповість «довгошерста», то запитай: «ви бажаєте отримати контакти найближчого грумера?»*  *Якщо клієнт відповість «так», то скажи: «Надайте адресу найближчої котячої перукарні»*  *Інакше*  *Скажи: «Запропонуй магазин з товарами по догляду за шерстю»*  *Закінчити*  *Інакше*  *Скажи «Запропонуй обрати магазин із зоотоварами»*  *Закінчити*  *Якщо клієнт не має кота*  *Скажи “Коли вирішите завести улюбленця – приходьте”*  *Закінчити*  Завдання:   1. Намалюй схему алгоритму (в інструменті на вибір, наприклад, у вбудованому Google Docs редакторі, [figjam](https://www.figma.com/figjam/) чи [miro](https://miro.com/)) 2. Який потрібен мінімальний набір тест-кейсів, щоб переконатися, що всі запитання були поставлені, всі комбінації були пройдені та всі відповіді були отримані? |

Згідно моїх розрахунків, для перевірки такої блок-схеми треба 7 тест-кейсів. Але у самій блок-схемі не впевнена, ще зі школи з ними не надто дружила))

