| Must have рівень:   1. Зроби порівняння статичних та динамічних технік тестування. Наведи переваги та можливі обмеження при використанні кожної з них.  |  | Статична техніка тестування | Динамічна техніка тестування | | --- | --- | --- | | Основна інформація | Верифікація програмного коду, документації, та ін. документів. Автоматизоване тестування коду | Валідація ПЗ, аналіз поведінки програми згідно специфікацій та очікувань кінцевого користувача | | Перевага №1 | Виявлення дефектів на ранніх етапах (до запуску ПЗ) | Перевірка роботи ПЗ в реальних умовах (наприкл., продакшн) | | Перевага №2 | Зниження вартості розробки та скорочення часу тестування | Можливість виконати перевірку продуктивності ПЗ | | Перевага №3 | Віднаходження багів, які неможливо виявити під час динамічного тестування : ) | Віднаходження багів, які неможливо виявити під час статичного тестування : ) | | Перевага №4 | Коректування дизайну як розробки, так і тестування, за необхідності | Різноманітність методів тестування (White Box, системне, функц.…) | | Обмеження №1 | Потребує глибокого занурення у проєкт та великої кіль-ті тех. знань | Переважна більшість методів не підлягає автоматизації | | Обмеження №2 | Не дозволяє оцінити поведінку системи під час виконання | Більша вартість та кіль-ть часу на тестування через необхідність створення тестового середовища | | Обмеження №3 | Результати можуть бути неочевидними для нефахівців (наприклад, замовника) | Залежність від умов реального використання або тестового середовища | | Обмеження №4 | Аналіз проблем взаємодії компонентів системи неможлива | Дефекти, що були знайдені за доп. цієї техніки, складніше виправляти | | Висновок | Обидва техніки є критично важливими для використання у тестуванні ПЗ, але як і завжди, кожний інструмент ефективний тільки у відповідному випадку: статичне перед релізом, а динамічне вже після : ) | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 (тому що перевіряємо тільки істинні та неістинні умови)**   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? |