| 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 | Це підхід до виявлення помилок у програмному забезпеченні без запуску коду. Замість виконання програми, перевіряються артефакти, такі як вимоги, дизайн, документація, або сам код (рев’ю, аналіз, верифікація). Статичне тестування починається на ранніх етапах життєвого циклу і є відповідно , частиною процесу верифікації. | Є частиною процесу валідації програмного забезпечення. Динамічне тестування - це тестування із запуском коду виконання, включає тестування ПЗ в режимі реального часу шляхом надання вхідних даних і вивчення результату поведінки ПЗ. | | Перевага №1 | Допомагає покращити читабельність та підтримуваність коду. | Дозволяє виявити помилки виконання, які неможливо помітити статично. | | Перевага №2 | Дає змогу виявити молки на ранньому етапі, цим самими зменшує витратність | Дає змогу перевірити функціональність, продуктивність, безпеку та зручність. | | Перевага №3 (і т.д.) | Підходить для перевірки вимог, тест-кейсів, покращує обмін критичної і важливої інформації між членами команди. | Тестується з точки зору реального користувача, що підвищує якість продукту | | Обмеження №1 | Обмеження у часі ( може бути затратно по часу ) , якщо немає чітких критеріїв. | Найдені баги можуть бути пізно виявленні, так як динамічне тестування проходить після написання коду. | | Обмеження №2 | Не виявляє всі типи помилок, наприклад продуктивності. | Може бути ресурсозатратною ( вимагає багато часу і витрат ) | | Обмеження №3 (і т.д.) | Потребує досвідчених фахівців | Не гарантує повного покриття: можна пропустити частини логіки, якщо не написано достатньо тест-кейсів. | | Висновок | Являється важливою технікою у процесі. Це як вичитати інструкцію до лодки, на якій маєш плисти, не гарантує що пройде все гладко, але зменшує шанс що щось піде не так | Являється важливою технікою у процесі. Це як не просто прочитати інструкцію до кавоварки, а перевірити як вона включається, як заливається вода тобто протестити її (перевірити чи працює у реальному житті) | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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? |

Мінімальний набір повинен складатись з 4-х тест-кейсів

1. Користувач має кота- довгошерстна порода- хоче контакти
2. Користувач має кота- довгошерстна - не хоче контакти глумера
3. Користувач має кота- короткошерстна
4. Користувач немає кота

Схема нижче

