| 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 | Тестування ПЗ відбувається без запуску коду, знаходить потенційні дефекти у ПЗ а також у супутніх документах програмного забезпечення. Включає у себе аналіз коду, написаного розробниками. | Ця техніка тестування виконується безпосередньо шляхом використання програми і визнанням того, чи працюють її функції згідно з очікуванням. Включає у себе тестування ПЗ у реальному часі, шляхом вивчення поведінки програми. | | Перевага №1 | Виявляє баги на ранніх етапах тестування, знижує ресурсозатратність | Відповідає найвищим стандартам, так як перевіряє всю функціональність програми | | Перевага №2 | Скорочує зусилля з пошуку та аналізу помилок, підвищує продуктивність | Якість ПЗ підвищується, тому що процесс тестування виконується з точки зору користувача. | | Перевага №3 (і т.д.) | Покращує обмін інформацією та ґрунтується на відгуках, які у подальшому можуть покращити уникненню нових багів і дефектів | Виявлення помилок, не помічених під час перевірки коду та можливість автоматизації тестування | | Обмеження №1 | Займає багато часу | Також займає багато часу, з приводу тестування усього функціоналу ПЗ | | Обмеження №2 | Не виявляє недоліки у середовищі застосування | Ресурсозатратне тестування | | Обмеження №3 (і т.д.) | Тестувальнику потрібно знати будову та механіки розробки коду, що ускладнює його працю | Баги можуть бути виявлені вже у життєвому циклі ПЗ, після кодування, що буде ускаладнювати процеси усунення | | Висновок | Статистичне тестування ідеально підходить для початкової перевірки ПЗ, знижує ресурсозатратність та уникає появу подальших помилок пов'язаних із кодом, але на ці механики тестування витрачається більше часу, та тестувальник, що виконує аналіз, потрібен бути компетентен у розумінні будови коду. | Динамічне тестування є дуже важливим, тому що тестує фактично як працює ПЗ, особливо з точки зору користувача, перевіряє якість ПЗ, функціонал, надійність, поведінку, тощо. Ця механіка витрачає більше ресурсів, але є невід’ємною частиною тестування, тому що, напряму від неї залежить верифікація та валідація продукта. | |  |  |  | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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/)) <https://www.figma.com/file/KFFgOUdPdwSUHKJmaFRzTq/Untitled?node-id=0%3A1&t=uMWSRilX191tsGSq-0> 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 тест-кейсів - 4 |