| 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 | є методикою тестування програмного забезпечення, при якій ПЗ тестується без запуску коду. Воно являє собою процес або техніку, які виконуються для пошуку потенційних дефектів в програмному забезпеченні. Це також процес виявлення і усунення помилок і дефектів в різних супровідних документах, таких як специфікації вимог до ПЗ. | тип тестування, який перевіряє функціональність програми, коли код виконується. Динамічне тестування виконується шляхом фактичного використання програми і визначення того, чи працює функціональність так, як очікується. | | Перевага №1 | Знижує вартість фіксу знайдених багів, оскільки виявляє баги на ранніх етапах циклу розробки програмного забезпечення. | Це ретельне дослідження, яке розглядає всю функціональність програми, тому якість відповідає найвищим стандартам | | Перевага №2 | Відгуки, отримані в ході цього тестування, допомагають покращити функціонування процесу, що також допомагає команді уникнути подібних дефектів та багів. | Додаток тестується з точки зору користувача, що підвищує якість ПЗ | | Перевага №3 | Підвищує інформованість про різні проблеми якості програмного забезпечення | Виявлення складних помилок, які могли вислизнути на етапі код ревью | | Перевага №4 | Покращує обмін критичної і важливої інформації між членами команди | Може бути автоматизовано за допомогою спеціальних інструментів. | | Обмеження №1 | Процес статичного тестування може займати багато часу, так як в основному він виконується вручну | Являє собою складний процес, тому займає багато часу і дорого вартує | | Обмеження №2 | Перешкоджає виявленню вразливостей, представлених в середовищі виконання | Зазвичай виконується після завершення кодування, і знайдені баги виявляються пізніше в життєвому циклі розробки | | Висновок | Являється дуже важливим етапом, бо допомагає запобігти появі дефектів, завдяки комплексному тестуванню на ранніх етапах розробки ПЗ | Хоч і виконується на останніх етапах розробки ПЗ, але завдяки безпосередньому виконанню тестів програмного забезпечення (перевірки функціональної поведінки, продуктивності, надійності та інших важливих аспектів) команда може перевірити і підтвердити якість та ефективність ПЗ | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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? |

Щоб переконатися, що всі запитання були поставлені, всі комбінації були пройдені та всі відповіді були отримані потрібно 6 тест-кейсів, таким чином перевіряються всі варіанти умов( True & False) та кожен варіант відповідей.