| 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 програмного забезпечення, при якій ПЗ тестується без запуску коду, являє собою процес або техніку, які виконуються для пошуку потенційних дефектів в програмному забезпеченні. | 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. | * Процес динамічного тестування добре налагоджений, додаток тестується з точки зору користувача, що підвищує якість ПЗ. | | Перевага №2 | * Відгуки, отримані в ході цього тестування, допомагають покращити функціонування процесу, що також допомагає команді уникнути подібних дефектів і багів. | * Процес динамічного тестування добре налагоджений, додаток тестується з точки зору користувача, що підвищує якість ПЗ. | | Перевага №3 (і т.д.) | * Підвищує інформованість про різні проблеми якості програмного забезпечення. | * Динамічне тестування може бути автоматизовано за допомогою спеціальних інструментів. | | Обмеження №1 | * Процес статичного тестування може займати багато часу, так як в основному він виконується вручну. | * Динамічне тестування зазвичай виконується після завершення кодування, і знайдені баги виявляються пізніше в життєвому циклі розробки. | | Обмеження №2 | * Перешкоджає виявленню вразливостей, представлених в середовищі виконання. | * Висока вартість проведення тестування. | | Обмеження №3 (і т.д.) |  | * Оскільки динамічне тестування являє собою складний процес, воно займає багато часу. | | Висновок | Незважаючи на те, що статичне тестування вимагає багато часу, все ж варто витратити час на запобігання появи дефектів на останніх етапах розробки продукту. | Завдяки безпосередньому виконанню тестів програмного забезпечення (перевірки функціональної поведінки, продуктивності, надійності та інших важливих аспектів) команда може перевірити і підтвердити якість і 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.   P.S. якщо я правильно зрозумів :)   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 |