| 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 | Використовує статистичні методи для аналізу даних тестування та виявлення паттернів, які можуть свідчити про якість чи некоректну роботу системи | Основною ідеєю динамічного тестування є активне виконання програми та спостереження за її роботою | | Перевага №1 | Статистичні методи можуть допомогти зменшити кількість тест-кейсів, які потрібно виконати для досягнення прийнятного рівня впевненості у якості програмного продукту. Це дозволяє ефективніше використовувати ресурси, такі як час | Динамічні техніки надають можливість виконувати програму та перевіряти, чи поводиться вона так, як очікується. Це дозволяє ефективно виявляти дефекти та неправильності в роботі програмного продукту. | | Перевага №2 | Статистичні техніки дозволяють фокусуватися на тестуванні тих частин програми або функціональності, які мають найбільший потенціал для помилок або найважливіші з точки зору користувача | Динамічні техніки тестування дозволяють оцінювати програму в реальному або схожому до реального середовищі використання. Це допомагає ідентифікувати проблеми, які можуть виникнути лише при взаємодії програми з реальними умовами експлуатації. | | Перевага №3 (і т.д.) | Статистична техніка дозволяє використовувати аналіз даних для виявлення шаблонів, тенденцій і аномалій у тестових результатах. Це дозволяє швидко реагувати на зміни у якості програми та виявляти проблеми, які можуть залишитися непоміченими при інших методах тестування. | Для застосування динамічних технік тестування не обов'язково мати доступ до вихідного коду програми. Це дає можливість тестувати програми, для яких вихідний код не доступний або для яких важко зрозуміти весь код. | | Обмеження №1 | Для ефективного використання статистичних методів тестування може бути необхідний великий обсяг вхідних даних. Якщо вхідні дані обмежені або недоступні, статистичні методи можуть бути менш ефективними. | Зміни в коді або вимогах можуть вимагати постійного оновлення тестових сценаріїв. Це може бути часоємкою і трудомісткою задачею, особливо в ситуаціях, коли програма часто змінюється. | | Обмеження №2 | Деякі програмні системи можуть бути складними для моделювання статистичними методами через їх велику складність або непередбачувані взаємодії між компонентами. | Для досягнення повного покриття функціональності програми може вимагатися велика кількість тестів, що може бути важко реалізувати в обмежених умовах часу та ресурсів. | | Обмеження №3 (і т.д.) | Використання статистичних технік вимагає від команди тестування глибокого розуміння статистики. Якщо команда не має необхідних статистичних знань, це може призвести до неправильного використання або тлумачення результатів. | При реалізації динамічних тестів важко передбачити всі можливі варіанти введення, особливо коли програма використовується в реальних умовах. | | Висновок |  |  | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |