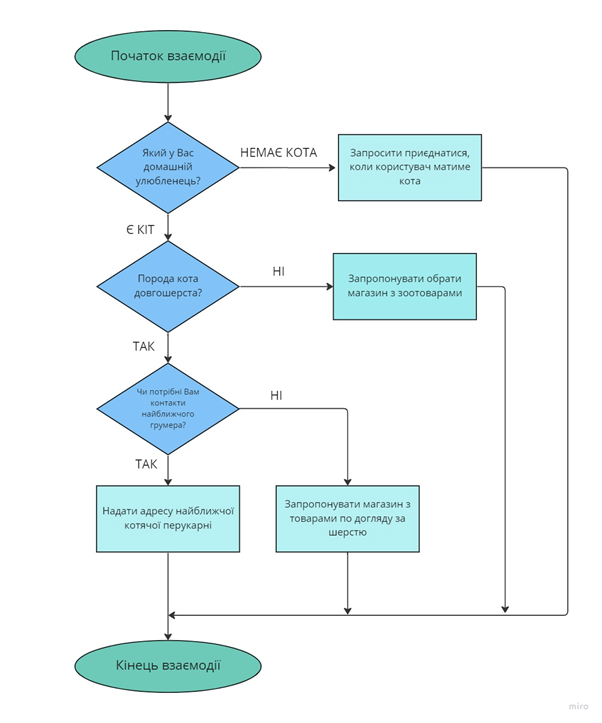
| Must have рівень:   1. Зроби порівняння статичних та динамічних технік тестування. Наведи переваги та можливі обмеження при використанні кожної з них.  |  | Статистична техніка тестування | Динамічна техніка тестування | | --- | --- | --- | | Основна інформація | Не вимагає запуску програми. Має на меті пошук потенційних дефектів в програмному забезпеченні. А також усуває помилки в різних супровідних документах | Виконується шляхом фактичного використання програми. Має на меті визначити чи працює функціональність програми, так я очікується. | | Перевага №1 | Знижує витрати на виправлення багів, які знаходяться на ранніх етапах циклу розробки | Забезпечує високу якість ПЗ | | Перевага №2 | Підвищує інформованість про проблеми якості ПЗ | Виявляє складні помилки, які можуть бути пропущені на етапі перевірки коду | | Перевага №3 | Покращує обмін важливою інформацією між членами команди | Може бути автоматизоване | | Перевага №4 | Ідентифікує баги, які складно виявити при динамічному тестуванні | Ретельне вивчення функціоналу програмного продукту | | Перевага №5 | Попереджує дефекти дизайну або кодування |  | | Обмеження №1 | Вимагає великої кількості часу через переважно ручне тестування | Вимагає великої кількості часу для проведення | | Обмеження №2 | Не виявляє дефекти пов’язані з існуванням продукту в реальному середовищі | Дороговартісний процес | | Обмеження №3 |  | Баги знаходяться вже в процесі реального життєвого циклу розробки продукту | | Висновок | Є важливим, оскільки допомагає запобігти дефектам на останніх етапах розробки, хоч і є досить тривалим по часу | Завдяки безпосередньому виконанню тестів програмного забезпечення команда може перевірити та 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.  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? |



Statement coverage = 4

Decision coverage = 4

Отже, потрібно 4 тести для 100% покриття коду