



1. Склади порівняльну таблицю функціонального, нефункціонального і пов’язаного зі змінами видів тестування.

Порівняння має містити такі блоки:

- що перевіряється;
- коли застосовується;
- обмеження;
- особливості

Вид тестування	Що перевіряється	Коли застосовується	Обмеження	Особливості
Функціональне	Перевірка форм на сайті: валідація даних, обов’язкові поля, правильність повідомлень про помилки. Тестування навігації: перехід між сторінками, гіперпосилання, кнопки, меню та інші елементи керування. Тестування функціоналу пошуку: алгоритми фільтрації, сортування, пагінація.	Функціональне тестування проводиться на основі вимог бізнесу.	Здійснюється вручну. Приклад: метод тестування чорної скриньки.	Функціональне тестування перевіряє, чи справжній результат працює відповідно до очікуваного результату.
Нефункціональне	Тестування продуктивності: вимірювання часу відгуку сервера, обробки запитів та часу завантаження сторінок. Тестування безпеки: перевірка на вразливості, атаки на відмову в обслуговуванні (DDoS), SQL-ін’єкції, перехоплення даних. Тестування сумісності: перевірка роботи	Нефункціональне тестування проводиться на основі очікувань замовника та вимог до продуктивності.	Більш доцільно тестувати за допомогою автоматизованих інструментів.	Нефункціональне тестування стосується аспектів програмного забезпечення, які не пов’язані з конкретними функціями чи операціями

	програмного забезпечення на різних платформах, браузерах та пристроях.			
--	---	--	--	--

2. Поясни, в чому різниця між регресією та ретестингом (5 речень).

Ретестинг - це процес перевірки вже перевіреного багу. Тестувальник перевіряє за розробником, чи дійсно він виправив баг. Повторне тестування виконується з тими самими даними та тим самим середовищем, але з новою збіркою.

Повторне тестування має вищий пріоритет та має бути виконано до регресійного.

Регресія - це тестування спрямоване на виявлення помилок у вже протестованих ділянках початкового коду. Одна з головних цілей регресійного тестування - це визначити, чи впливає зміна в одній частині програмного забезпечення на його інші частини.



Beet Sprout

3. Як ти вважаєш, чи можливе для продукту проведення тільки функціонального тестування, без перевірки нефункціональних вимог?

- Якщо так – в яких випадках?
- Якщо ні – чому?
- Обґрунтуй свою відповідь.

Виконання нефункціонального тестування під час виконання функціонального тесту є більш вигідним і додасть більше переваг загальній якості програмного забезпечення. Це дозволить виявити помилки продуктивності (обмежені тестовим середовищем та залежностями), а отже, зменшить ситуації припущення про функціональні проблеми.

Нефункціональне тестування необхідне, оскільки воно оцінює основні критерії користувача, такі як продуктивність і зручність використання, а також перевіряє, чи працює програмне забезпечення так, як очікується, за межами його базової функціональності.

4. Як ти розумієш необхідність проведення smoke (димового) тестування? Чи завжди воно є доречним?

Димові тести здійснюються в тих випадках, коли тестувальники отримують нову версію (білд) програми на тестування, при цьому вважаючи її відносно нестабільною. На

цьому етапі необхідно переконатися, що надзвичайно важливі функції AUT (Application Under Test) працюють згідно з очікуваннями. Такий вид тестування потрібен для того, щоб виявити серйозні проблеми якомога раніше і відхилити цей білд або повернути його на доопрацювання на ранньому етапі тестування.

Метою димового тестування є перевірка надійності і стабільності програмного забезпечення загалом, щоб дозволити подальше проведення ретельнішого тестування.

Якщо продукт не пройшов тест на дим, це означає, що початкова збірка має суттєві недоліки, які необхідно усунути до початку подальших випробувань.

Таке тестування димом відбувається щоразу, коли до збірки програмного забезпечення додаються нові функції.

Якщо ми не проведемо димовий тест на цих етапах, ви можете виявити серйозні дефекти на більш пізніх етапах тестування функціональності, а це може вплинути на дату випуску нової збірки або призвести до більш серйозних порушень.



Mighty Beet

5. Ти – засновник/ця стартапу і плануєш випустити на ринок мобільний застосунок для обміну світлинами котиків для iOS та Android пристроїв.

Користувачі можуть завантажувати фотографії котиків. Але не можуть завантажувати фотографії інших тварин/людей/об'єктів. Користувачі можуть додавати друзів, ставити "вподобайки", залишати коментарі.

Завдання: Напиши 5 функціональних тест-кейсів, які перевіряли б роботу застосунку.

Таблиця CatMobile

6. Напиши, які нефункціональні вимоги ти хотів/ла б застосувати для продукту твого стартапу.

Опиши перевірки, які б їх перевіряли (3-5 прикладів).

1. Мобільний додаток повинен бути захищений від несанкціонованого доступу.

- ввести в поле пароль декілька комбінацій паролю та залогінитися

- ввести некоректний емейл та залогінитися

2. Мобільний додаток повинен бути доступним, коли це необхідно.

- відкрити мобільний додаток з різних пристроїв iOS, Android

-- відкрити мобільний додаток з мобільного інтернету або Wi-Fi

3. Мобільний додаток повинен бути простим в обслуговуванні та оновленні.

- встановити оновлення на мобільний додаток

- звернутися до служби підтримки клієнтів

4. Додаток повинен працювати на платформах IOS та Android.

- відкрити мобільний додаток з різних пристроїв iOS,Android

5.Додаток повинен бути простим у використанні та зрозумілим.

- перевірити чи всі сторінки, кнопки, інтуїтивно зрозумілі та клікабельні

- перевірити кольорову гаму інтерфейсу

- перевірити шрифт-перевірити текст на граматичні помилки

- перевірити швидкість переходу між сторінками

- перевірити якість фото, що публікується

