

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

Порівняння має містити такі блоки:

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

Види тестування	Що перевіряється	Коли застосовується	Обмеження	Особливості
Функціональне	Відповідність функціональних вимог програмного забезпечення його реальним характеристикам	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.	Не враховує інші фактори, такі як продуктивність продукту, тобто швидкість реагування, пропускну здатність, які є важливими в тестуванні перед випуском продукту	Тестування зосереджено на специфікаціях системи, операційної системи, браузерів тощо. Забезпечує випуск високоякісного продукту, який відповідає вимогам замовника, який має всі функції, що працюють відповідно до вимог замовника.

Нефункціональне	Оцінює роботу системи на відповідність набору вимог, таких як, продуктивність, безпека, сумісність та стабільність.	Тестування продуктивності: вимірювання часу відгуку сервера, обробки запитів та часу завантаження сторінок. Тестування безпеки: перевірка на вразливості, атаки на відмову в обслуговуванні, перехоплення даних. Тестування сумісності: перевірка роботи програмного забезпечення на різних платформах, браузерах та пристроях.	Нефункціональне тестування проводиться на етапі системного тестування ПЗ після завершення модульного та інтеграційного тестування. Необхідно проводити функціональне тестування перед нефункціональним, оскільки неможливо перевірити надійність або продуктивність функцій, які взагалі не працюють.	Нефункціональне тестування в тестуванні програмного забезпечення повинно проводитися кожного разу, коли розробники оновлюють програмне забезпечення або коли змінюється код. Це означає, що нефункціональне тестування може бути дуже повторюваним, що не тільки забирає час, але й втомлює тестувальників, також це впливає на зростання вартості продукту.
Види тестування, пов'язаного зі змінами	Після проведення необхідних змін, таких як виправлення бага / дефекту, програмне забезпечення повинне бути перетестоване для підтвердження того факту, що проблема була дійсно вирішена.	Регресійне тестування спрямоване на перевірку змін, зроблених в додатку або середовищі (лагодження дефекту, злиття коду, міграція на іншу операційну систему, базу даних, веб сервер або сервер додатки), для підтвердження того факту, що	Цей вид тестування дуже часто має обмеження в часі і може бути проведено поверхнево	Види тестування, які необхідно проводити після установки програмного забезпечення, для підтвердження працездатності програми або правильності здійсненого виправлення дефекту: Регресійне тестування; Димове тестування;

		існуюча раніше функціональність працює як і раніше. Димове тестування, виконується для підтвердження того, що після збірки коду (нового чи виправленого) програмне забезпечення запускається та виконує основні важливі функції. Повторне тестування перевіряє, що дефект виправлений та змінений функціонал працює, як треба. Тестування збірки направлене на визначення, чи відповідає випущена версія критеріям якості для початку тестування.		Санітарне тестування або перевірка узгодженості; Тестування збірки
--	--	---	--	--

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

Ретестинг застосовується для перевірки якості будь-якої конкретної функції, компонента чи модуля програми, якщо в цих частинах були виявлені баги. Мета регресійного тестування полягає в тому, щоб перевірити, чи нові зміни коду не впливають негативно на існуючі розроблені та протестовані функції програми, та зменшення кількості багів у системі на момент релізу. Під час виконання регресійного тестування виконуються як функціональні, так і нефункціональні тести.

Регресія застосовується коли до наявної програми додається нова функція, якщо на етапі тестування було виявлено багато критичних помилок, додані виправлення, які

зачіпають більшу частину функціональності системи, інтерфейс програми було змінено для покращення взаємодії з користувачем.

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.

Проведення тільки функціонального тестування, без перевірки нефункціональних вимог на мою думку неможливо. Під час тестування системи тестувальники проводитимуть як функціональне, так і нефункціональне тестування, починаючи з функціонального тестування.

Після того, як тестувальники встановили, що програмне забезпечення функціонує як очікувалося, вони проводять нефункціональне тестування, щоб оцінити, чи відповідає воно також нефункціональним параметрам.

Зазвичай необхідно проводити функціональне тестування перед нефункціональним, оскільки неможливо перевірити надійність або продуктивність функцій, які взагалі не працюють.

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

Димове тестування використовується для виявлення, в основному, явних помилок, які можуть виникнути на найперших етапах роботи програмного забезпечення. Димове тестування можна визначити, як деякий короткий цикл тестів, який здійснюється після виходу чергової версії системи. Виконується це для перевірки роботи основного функціоналу розроблюваної програмної системи. Після успішного проходження димного тестування, система, що розробляється, відправляється на наступні цикли більш серйозних видів тестів.

Цей вид тестування доречний, коли є потрібність поверхневої перевірки всіх модулів програми на предмет працездатності і наявності швидкого знаходження критичних і блокуючих дефектів.

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.

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

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

Назва	Вимога
Мовна локалізація застосунку	Застосунок повинен бути локалізований такими мовами: ua, eng. Мова застосунку за замовчуванням - українська.
Сумісність застосунку	Застосунок повинен підтримувати пристрої Android, що працюють на версіях 9,10,11,12,13 та iOS на версіях ОС: 3.6, 3.3, 3.4, 4.3, 2.3
Надійність системи застосунку	Система повинна працювати без збоїв у 95 відсотках випадків використання протягом місяця
Масштабованість системи застосунку	Система повинна бути достатньо масштабованою, щоб підтримувати 500 000 відвідувань одночасно та зберігати оптимальну продуктивність
Ремонтопридатність системи застосунку	Середній час відновлення системи після збою не повинен перевищувати 10 хвилин