

## Перший рівень — відпрацюй навички на базовому рівні.

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

Порівняння має містити такі блоки:

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

Функціональне тестування	Нефункціональне тестування	Тестування змін
Що перевіряється: функціональні елементи системи	Що перевіряється: як ці елементи функціонують (характеристики)	Що перевіряється: зміни (збої) внаслідок проведення фіксів в інших ділянках коду.
Коли застосовується: після написання коду.	Коли застосовується: після створення функціоналу.	Коли застосовується: коли внесені зміни в інші ділянки коду.
Обмеження: Вимоги мають бути завершені та уточнені.	Обмеження: може проводитися після закінчення функціонального тестування, та після уточнення вимог.	Обмеження: виконується після смюк тестінгу
Особливості: перевіряю наявність всіх елементів системи для подальшої перевірки на комфортність роботи.	Особливості: перевіряю наскільки комфортно користуватися системою, її рентабельність.	Особливості: свідчить про те, що функціональність, яка мала б нормально працювати працює або ж не працює нормально.

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

Різниця між регресією і ретестингом полягає в тому, що регресія перевіряє функціональність, до якої не вносились зміни, тобто функціональність, яка мала б працювати після внесення змін. Ретестинг – це confirmation тестінг, він перевіряє безпосередньо, чи внесені зміни до функціональності, в якій був знайдений баг. Приклад: ми знайшли баг в модулі авторизації (нам не висвічує, що такий логін вже зареєстрований на сайті, ми це пофіксували, перевірили, вже висвічує), після цього ми маємо перевірити, чи внаслідок цього фіксу приходить клієнту смс на телефон при авторизації (якщо є 2-хфакторна), або перевірити, чи зможе клієнт змінити пароль з висиланням листа на пошту.

**Другий рівень** — детальніше заглибся в практику.

1. 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.

Моя точка зору така, потрібно проводити обидва види. В функціональному тестуванні ми перевіряємо, які функції ПО реалізовані, і наскільки правильно (відповідно до вимог, чи загальних стандартів функціонування вони реалізовані), нефункціональне тестування демонструє, наскільки коректно ці функції працюють, тобто оцінюється, як ПО працює. Потрібно завжди робити ці 2 види. Нефункціональне тестування взагалі думаю неможливе без реалізації та перевірки певних функцій функціональним тестуванням. А функціональне також немає сенсу без нефункціонального, якщо наші функції будуть реалізовані, але працюватимуть некоректно (повільно, не витримуватимуть навантаження, не зможуть відновитися, чи інсталюватися тощо).

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

Воно завжди є доречним, якщо здійснюємо ми його після випуску нового білда (версії продукту). Перевіряємо ключову (критично важливу функціональність). Це дає нам змогу впевнитись, що ключові тести пройдені, і можемо переходити до регресії, санітарного тестування і любого іншого виду. Інженерною мовою потрібно впевнитись, чи після запуску з установки не пішов дим. Наприклад, ми перейшли на новий сервер для того, щоб більше юзерів могли працювати з більшим масивом даних. Нам потрібно впевнитися, чи відкривається наш сайт, чи можна в ньому авторизуватись, і чи видно ключові пункти меню.

Для тестування змін, чи дрібнішої функціональності смоук тестінг не потрібно проводити (це буде трата часу).

**Третій рівень** — різнобічно опануй тематику уроку.

1. 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.

**Тестування безпеки** – перевіряю, чи можливо зламати профіль користувача, щоб викачати звідти зображення;

**Тестування взаємодії** – перевіряю, чи взаємодіє додаток з Android, iOS, чи з'єднується він по технології Bluetooth з іншим девайсом.

**Тестування зміни** – чи можливо переглядати картинки в full hd форматі та викачувати їх (якщо до того було 720p)

**Смоук тестінг** – після оновлення додатку, чи відкривається він, чи можемо авторизуватися.

**Sanity testing** – після регресії перевіряємо детально конкретну функцію (чи відпрацьовує модуль підписки на розсилку для категорії улюблених картинок).

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

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

**Load тестінг** – виявити, скільки користувачі одночасно та з яким об'ємом інформації можуть працювати з додатком;

**Usability** - наскільки їм зручно працювати з інтерфейсом додатку (активізація в пам'яті, правильність виконання задач, ефективність виконання задач, емоційна реакція);

**Тестування установки** – чи установлюється наш додаток на більшість сучасних мобільних телефонів з останніми версіями ОС та мінімальними версіями.

**Тестування локалізації** - перевіряємо правильність перекладу системних повідомлень чи помилок і всіх розділів додатку, щоб переконатися, що він підтримує багатомовний інтерфейс.

**Тестування інтернаціоналізації** - створення в додатку можливості підтримки елементів, які неможливо локалізувати звичайним чином (вертикальний текст азійських країн, читання з права на ліво арабських країн, спеціальні валютні позначення і т.д.).