**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

Функціональний вид тестування:

**-Що перевіряється**: перевіряється заздалегідь задана поведінка програми в певному модулі у випадку Unit тестування або програми в цілому у випадку з системним тестуванням.

Також на останньому рівні Acceptance testing- перевіряється чи відповідає тестування вимогам заданим в acceptance criteria.

**-Коли застосовується**: коли потрібно функціонально перевірити ПЗ, тобто коли потрібно перевірити певний модуль (unit testing), взаємодію декількох модулів у роботі (integration testing) або перевірити загалом роботу усієї програми з усіма юніт тестами System testing)

**-Обмеження:**

Обмеженням функціонального типу тестування є те, що нас цікавить не внутрішня реалізація, а саме отриманий результат.

**Особливості** функціонального тестування полягають у тому, щоб переконатися, що весь функціонал ПЗ працює саме так, як зазначено в технічному завданні.

Нефункціональний тип тестування:

**-Що перевіряється**: нефункціональне тестування оцінює інші аспекти системи: перевірку продуктивності, зручності, надійності тощо

**-Коли застосовується:** Нефункціональне тестування застосовується у випадку коли потрібно перевірити інформацію про безпеку, справність, надійність системи та її юзабіліті.

**-Обмеження** нефункціонального тестування полягає в тому, що ми обмежуємось перевіркою тільки властивостей, які не відповідають функціональності системи.

**-Особливості** Нефункціональне тестування це той тип тестування який відповідає на питання «як?», тобто, наприклад, як програма поведе себе при навантаженні? Як працює локалізація ПЗ? Як програма працює на різних системах? Як користувач взаємодію з зовнішнім інтерфейсом?

Пов‘язане зі змінами тестування:

**-Що перевіряється:** Перевіряється робота ПЗ після внесення змін до певного функціоналу і чи це не впливає на роботу уже готових ф-цій.

**-Коли застосовується:**

Уже з назви виду тестування, можна зробити висновок, що він застосовується при умові коли у ПЗ були внесені певні зміни які потребують тестування.

**-Особливості** тестування пов‘язаного зі змінами полягає в тому що, ми перевіряємо роботу певних модулів або ж усією програми тільки коли були внесені якісь зміни в ПЗ, тобто ми маємо пересвідчитись, що ці зміни ніяк не вплинули на вже існуючий функціонал програми.

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

Регресія - це більш ширший вид тестування. Він полягає в тому що ми перевіряємо всі ф-ції ПЗ після внесення змін, проведених у самому додатку або у середовищі( виправлення дефекту, зміна у базі даних і тд.)

Ретестинг ж полягає у тому, що ми тестуємо безпосередньо якийсь певний дефект і перевіряємо чи змінений функціонал працює як треба. Наприклад: ми знайшли баг у якомусь модулі, пишемо на нього баг-репорт, Розробник відправляє нам виправлений дефект і ми проводимо ретестінг цього дефекту.

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

1. Виконай завдання попереднього рівня.

2. 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.

Я думаю, що фунціональне і нефункціональне тестування взаємопов‘язані між собою. Обидва типи тестування однаково важливі, адже вони допомагають краще зрозуміти різні особливості системи. Найважливіша мета таких тестувань — забезпечити кінцевих користувачів якісним програмним продуктом.

Наприклад провівши усі рівні функціонального тестування, нам потрібно буде перевірити UI та UX по відношенню до користувача, нам 100% потрібно буде провести Perfomance тестування, яке точно буде включати load i stress тестування, для того щоб перевірити як ПЗ поведе себе в різних ситуаціях та в різних сценаріях роботи.

Це дозволить нам зрозуміти на скільки ПЗ є юзабельним для кінцевих користувачів.

Тому, я вважаю що обидва види тестування є важливими, вони доповнюють один одного.

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

Smoke або ж димове тестування проводиться у випадку, коли проводяться певні зміни у програмі. Воно пооводиться поверхнево але разом з тим перевіряє на роботоспроможність найважливіший функціонал ПЗ.

Воно є доречним коли в програмі вносилисьпевні зміни і ми хочемо переконатись, що після цих змін все працює як і має. На відмінну від Regression тестування, яке також проаодиться після певних змін, але тут ми вже перевіряємо чи пвсля цих проведених змін не з‘явились якісь нові помилки.

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

1. Виконай завдання двох попередніх рівнів.

2. 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.

[Тест Кейси ДЗ 5](https://docs.google.com/spreadsheets/d/1CF7vjHHagSfwHVvDhn3UYObmMk5OAA0NPYdur160fjg/edit#gid=0)

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

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

У процесі тестування всі нефункціоналі види тестування є важливими, я б застосував такі нефункціональні вимоги:

-тестування продуктивності

-тестування встановлення

- UI i UX тестування

-конфігураційне тестування

Під час тестування продуктивності я б перевірив load i stress тестування, щоб розуміти як додаток себе поведе при умові коли додатком користується одночасно 10тис чоловік і 100тис;

Також протестував би як програма працює при середній завантаженості системи.

Під час тестування встановлення перконався б у тому що додаток легко видалити з пристрою (телефону, ПК) і так само легко встановити без додаткових файлів.

Перевірив би взаємодію інтерфейсу та зручність користування звичайному юзеру.

100% випустив би Бета версію програми, щоб всі бажаючі користувачі могли поюзати програму в реальних умовах і внести свої побажання, якщо такі будуть. + це буде класний досвід користувачу відчути себе в ролі тестувальника та одним з перших завантажити додаток.

Під час конфігураційного тестування перевірили б взаємодію додатку з різними операційними системами( ios, android, windows, macos, chrome os).

Та різними браузерами(safari, chrome, mozilla, opera)