**Перший рівень**

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

| **Види тестування** | **Що перевіряється** | **Коли застосовується** | **Обмеження** | **Особливості** |
| --- | --- | --- | --- | --- |
| **Функціональне** | Чи відповідає наш продукт заздалегідь заданій поведінці; включає в себе аналіз специфікації функціональності компоненту або системи.  Швидкість роботи, ефективність, безпека, масштабованість, надійність, сумісність та інші. | На процесі комплексного тестування й іспитів, для визначення повноти реалізації функціональних задач і їхньої відповідності вхідним вимогам | -Поверхнева перевірка працездатності системи, без поглиблених перевірок роботи продукту  -Не може гарантувати 100% тестового покриття  -Потребує постійного оновлення та підтримки тестового випадку  -Може бути складним для автоматизації | -Орієнтовано на функціональність  -Орієнтовано на потреби користувача  -Результати перевірки легко вимірювані  -Може бути автоматизовано  -Використовує різні методи тестування  -Включає в себе тестування взаємодії між компонентами застосунку та іншими системами |
| **Нефункціональне** | Витримка системи при різних навантаженнях на неї  Здатність системи до відновлення, після збоїв  Продуктивність системи при стабільному потоці бізнес - користувачів | Перед тим, як піддавати ПЗ великим навантаженням    Коли очікується великий потік користувачів (чорні п’ятниці, розпродажі і т.д.)  Коли система має буде опрацьовувати більший об’єм даних  На процесі комплексного тестування | -Важкість виміру  -Обмеження доступу до середовища  -Недоступність деяких характеристик  -Суб’єктивність вимог | -Тестування реальних умов  -Вимірювання показників  -Тестування на етапі розробки  -Використання різних технік тестування |
| **Регресійне** | Всі функції ПЗ, які потерпіли зміни, проведені у самому додатку або середовищі  Може проводитися у функціональному та нефункціональному тестуванні | Коли якісь функції ПЗ, окремі коди, або модулі потерпіли зміни, або коли змінилося середовище | -Необхідність оновлення тестових сценаріїв  -Недостатнє охоплення тестів  -Погіршення результатів сценарію | -Потребує повторного тестування тест - кейсами, що були написані на початку розробки |
| **Димове** | Поверхнева оцінка працездатності системи на відповідність основним вимогам до функціоналу | На початку циклу тестування, для прийняття рішення про подальше тестування або виправлення наявних дефектів, до початку більш поглибленого тестування | -Низьке покриття тестування  -Неможливість виявити складні проблеми  -Низька точність  -Низька ефективність | -Мета тестування  -Час тестування  -Підготовлені сценарії тестування  -Перевірка базового функціоналу  -Можливість автоматизувати процес тестування  -Включає документацію і роботу за сценарієм |
| **Санітарне** | Працездатність системи, коли виправлена незначна помилка в системі або є невелика зміна у функціональності | Під час розробки ПЗ  Після випуску продукту на ринок | -Проводиться на збірках, які успішно пройшли димові випробування і перед циклом регресійного тестування | -Зазвичай виконується вручну  -Зберігає час в умовах, коли його недостатньо для максимально великого регресійного тестування  -Може бути автоматизоване |
| **Тестування збірки** | Готовність збірки до життєвого циклу тестування | У процесі розробки ПЗ | -Обмеженість варіантів збірки  -Відсутність доступу до певних компонентів системи  -Відсутність доступу до оточення  -Складність відтворення | -Валідація збірки  -Перевірка на відповідність вимогам  -Перевірка на стабільність  -Тестування на різних платформах та конфігураціях  -Тестування збірки у інтеграції з іншими системами  -Можливість автоматизації тестування |

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

Ретестинг проводиться для того, щоб довести, що минулі баги були виправлені та, наразі, більше не є багами. Або навпаки, не виправлені та потребують доробки знову.

А регресія дає можливість підтвердити або спростувати, що нові зміни коду або функціоналу не повинні давати побічні ефекти для існуючих функцій.

До того ж, регресія має під собою розуміння того, що ми маємо перевірити весь функціонал, подібний тому, до якого були внесені зміни (наприклад, якщо була додана нова кнопка - потрібно повторно перевірити всі, щоб зрозуміти, чи не мала впливу нова кнопка на старі, та на зовнішній вигляд системи).

**Другий рівень**

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

* Так, для продукту можливе проведення тільки функціонального тестування, коли наш продукт має обмежений список функцій.

За допомогою функціонального тестування ми маємо змогу перевірити, чи відповідає продукт функціональним вимогам, чи відповідає інтерфейс заданим вимогам, та чи коректно працює задана послідовна поведінка. Також, на етапі функціонального тестування ми маємо змогу перевірити захищеність системи та даних, котрі вона містить, швидкість та ефективність роботи системи.

Тобто, якщо ми маємо простий сайт/блог, зі збором даних користувачів тільки через форму коментування (поля “ім’я”, “пошта”, “коментар”), ми можемо провести тільки функціональне тестування.

* Проведення функціонального та нефункціонального тестування дає нашому продукту більшу захищеність, більш поглиблену перевірку відповідності системи до заданих бізнесом та потенційним користувачем, вимог, стресостійкість, та вміння відновлюватися самостійно, після збоїв. До того ж, продукт, протестований функціонально та нефункціонально має бути більш витривалим до великої кількості користувачів, запитів, бути менш вразливим до хакерських атак та краще захищати та обробляти інформацію, яка зберігається у його системі.

Якщо ми маємо більш складну систему, якщо ми щось продаємо на нашому сайті та націлені на збільшення трафіку нашого сайту, або якщо наш продукт має багато важливих завдань, та має бути тестований за всіма рівнями, тільки функціональним тестуванням ми обмежитися не можемо.

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

Я розумію димове тестування як необхідність поверхневої перевірки функціональної складової системи, яка дає можливість зрозуміти, чи готовий продукт до подальшого, більш обширного тестування, та чи відповідає він загальним потребам бізнесу. Або продукт ще має бути в розробці чи виправлений.

**Третій рівень**

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

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

1. Перевірити, чи може користувач, який має особистий кабінет у нашому застосунку, завантажувати фото, на яких немає котиків. Для тестувальника це буде пріоритетна перевірка, тому що обмін фотографіями саме котиків - основна ціль додатку.
2. Перевірити роботу кнопку “Додати зображення котика” та впевнитися, що під кнопкою “Додати зображення котика” є інформація про те, що “користувач може завантажувати тільки фотографії котика/котиків”.
3. Перевірити, чи може зареєстрований користувач нашого застосунку, після перехіду на сторінку іншого користувача, додати його в друзі, чи є в нас така кнопка, та чи реагує система на її натискання.
4. Перевірити поведінку системи нашого застосунку, якщо користувач вибере більше 10 фото котиків для публікації. Тут, також, можна додати ще одну перевірку реакції системи, на додавання одного або декількох фото не котиків, серед фото котиків.
5. Перевірити, чи може зареєстрований користувач залишати коментар під іншими постами.

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

Додаток має коректно працювати, навіть якщо користувач користується ним цілодобово.

Додаток має встановлюватися, видалятися та оновлюватися успішно.

Додаток має бути зручним для користувачів.

Система додатку має працювати злагоджено, за умов великої кількості запитів від користувачів додатку.

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

1. **Тестування встановлення.** Перевіряємо, чи встановлюється додаток на різних версіях iOS та Android, чи може користувач видалити додаток, чи може користувач зупинити встановлення додатку, та чи видаляється додаток з телефону користувача, якщо завантаження було припинено, або застосунок було видалено з телефону, та чи коректно працюватиме застосунок, після оновлення.
2. **Тестування зручності використання.** Перевірити, спираючись на особистий досвід, чи буде потенційному користувачеві зручно користуватися додатком. Чи відповідає додаток потребам та очікуванням користувача. Чи може користувач користуватися додатком, інтуїтивно розуміючи, де та що має знаходитися у інтерфейсі.
3. **Стресове тестування.** Перевірити роботу застосунку, коли 10000 користувачів одночасно додають фото до свого особистого кабінету, завантажуючи їх у систему. Або навантажити систему та якщо трапиться збій - перевірити здатність до регенерації системи, та збереженості даних.
4. **Тестування стабільності або надійності.** Перевірити роботу застосунку, за умов постійного, але стабільного навантаження.
5. **Навантажувальне тестування.** Перевірити автоматизованим тестуванням роботу застосунку, при імітований роботі певної кількості бізнес - користувачів.