**QA – Homework 5**

**Ilya Nechytailo**

Рівні та види тестування.

**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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вид тестування | Що перевіряється | Коли застосовується | Обмеження | Особливості |
| Функціональне | Виявлення невідповідності між реальною поведінкою реалізованих функцій і очікуваною поведінкою відповідно до специфікації і вимог | Можуть бути представлені на всіх рівнях тестування | Обмежується перевіркою на баги та невідповідності в роботі лише функціоналу програмного продукту | Нас цікавить лише результат роботи системи, а не її проміжні стани |
| Нефункціональне | Перевіряється масштабованість та надійність проекту  Перевірка продуктивності, зручності, надійності, безпеки | Найчастіше нефункціональне тестування проводиться після функціонального із використання засобів автоматизації, оскільки ручне проведення тестів може бути інколи проблематичним | Обмежуються перевіркою нефункціональних вимог та властивостей програмного забезпечення, але на нефункц. тестування покладена більш значуща роль в сенсі того, що від якості проведеного нефункц.тестування залежить подальше враження користувача, яке вплине на популярність та рейтинг компанії-розробника | Вони дають інформацію про безпеку, справність та надійність системи. Головною особливістю, я б, мабуть, виокремив, що нефункціональне тестування має більш ширшу роль в плані цілого проекту та забезпечує якість безперебійної роботи системи. |
| Пов’язаний зі змінами | ПЗ повинно бути перевірено для підтвердження того факту, що попередньо виявлена проблема була коректно вирішена.  Може перевірятись в цілому усе ПЗ, може бути перевірка якогось локального дефекту, тощо | Застосовується найчастіше після проведення певних змін в програмному забезпеченні, такі як: виправлення бага/дефекта | Обмежуються перевіркою конкретних багів, нюансів, після внесення змін. Головною суттю та обмеженням даного виду тестування є перевірка коректності функціювання програмного забезпечення після внесення поправок. | Смоук-тестінг – найпростіші тести, без проходження яких – далі рухатись нема сенсу, бо вони є базовими та необхідними для подальшого тестування.  Санітарне тестування – зосередження уваги на предметі, місці, де були дефекти, там і перевіряємо.  Згідно з ISTQB – Smoke test = Sanitary test – це синоніми. |

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

***Регресія*** *– це поняття, яке позначає вид тестування функціональності ПЗ, після внесення змін на фазі системного тестування. Це вид тестування, для того, щоб зрозуміти, що продукт працює нормально з новими функціями.*

*В свою чергу,* ***ретестинг*** *– це поняття, яке характеризує, процес перевірки конкретних тестів, у яких було виявлено помилки, іншими словами, це перевірка баг-фікса.*

*Таким чином, головною різницею цих двох понять є те, що ретестинг – це перевірка конкретної помилки, чи проблеми, на яку був створений баг-репорт, а регресійне тестування проводиться – коли до наявної програми додаються нові функції, змінюється середовища тестування, додані якісь виправлення в системі.*

**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.

*Щодо відповіді на це питання – моя думка залишається нейтральною, що я маю на увазі. З одного боку, я вважаю, що в тестуванні можливо все, найулюбленіша наша фраза, яка нас завжди рятує «Все залежить від проекту». Функціональне та нефункціональне тестування в цілому, я б чисто гіпотетично міг назвати самостійними один від одного незалежними компонентами. Якщо, наприклад, проект обмежений ресурсами, часом, кваліфікацією тестувальників, вони залучені на пізніх етапах, аби тільки перевірити функціональність ПЗ, то чисто в теорії, я вважаю, що проведення тільки функціонального тестування можливе.*

*Питання інше, особисто від мене – Чи буде це ефективно та гарантувати найвищий ступінь якості продукту – вважаю, що Ні. Бо, до проекту висуваються різні вимоги, а саме, в цьому випадку нам важливо розуміти, що є функціональні вимоги та нефункціональні, і кожен з видів вимог потребує перевірки згідно з відповідним типом тестування, аби надати більш якісну та ширшу картину якості продукту та готовності до релізу.*

*Тому, на мою думку, підсумовуючи вищезазначене, можу припустити, що зустрічаються випадки на проектах, коли проводиться тільки функціональне тестування продукту, але, я вважаю, що лише поєднання усіх видів тестування одночасно може надати так би мовити «найвищий» ступінь якості продукту відповідно до специфікації та очікуванням замовника та майбутніх користувачів.*

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

*Аби відповісти на це питання, пропоную, більш детально ознайомитись з поняттям Димового тестування, а саме: це певний мінімальний набір тестів, на якісь певні явні помилки. Цей тест зазвичай може виконуватись самим програмістом. Важливо розуміти, що програму, яка не пройшла цей вид тестування, нема сенсу передавати на більш глибоке тестування.*

*Необхідність цього тестування, на мою думку, каже само за себе, не пройшовши найбазовіших моментів – рухатись далі нема сенсу.*

*Чи завжди воно є доречним? В контексті процесу тестування та підготовки до тестування – вважаю, що так. Поясню чому. По-перше, димове тестування за своєю суттю є дуже легким та швидким не глибоким процесом, тому його виконання не потребує багато ресурсів. В той же час, ми повинні пам’ятати про вартість дефекту на різних етапах, смоук тестінг – це свого роду підготовчий крок для початку повноцінного тестування, тому помітивши певний дефект або помилку під час цього тестування – ми знижуємо автоматично вартість знайденого дефекту, час, витрачений на тестування ПЗ, інші виробничі ресурси.*

*Окрім, цього димове тестування можна застосувати під час впровадження та інтеграції нових функцій з існуючою збіркою ПЗ, таким чином перевіряємо критично важливі функції при змінах, аби рухатись далі.*

*Тому, вважаю, що цей вид тестування є доречним у використанні.*

**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 зазначені у файлі Excel - **Homework\_5\_Test cases, доступ надано за посиланням.**

3. Напиши, які нефункціональні вимоги ти хотів/ла б  застосувати для продукту твого стартапу.   
Опиши перевірки, які б їх перевіряли (3-5 прикладів).

Вимоги до продукту:

1. Застосунок повинен відкриватись не більше ніж 2с на телефонах з операційною системою iOS.

* Перевірка – використовуючи усі вищезазначені у вимогах умови середовища, зроблю вручну тестування додатку за, перевірю швидкість його откліку на відкриття.

1. Застосунок повинен ідентифікувати світлини без котиків та не допускати їх до завантаження.

* Перевірка – Цю вимогу я перевірю шляхом додавання світлини до системи, на якій не буде зображений котик. Очікувана реакція програма за допомогою впровадженого штучного інтелекту повинна відобразити помилку з надписом – «Переконайтесь, що Ви обрали, саме котика»

1. Кнопка «Додати світлину» повинна бути кольору # FFA07A.

* Перевірка – перевірити колір кнопки можна за допомогою коду, який написаний та в якому зазначається вказаний rgb-код.

1. Світлини розміром більше 20мб не можуть бути додані до системи.

* Перевірка – тестування шляхом спроби завантажити світлину розміром 21мб.