Домашнє завдання. Lesson 6

Завдання 1

1.1 Склади порівняльну таблицю функціонального , нефункціонального і пов’язаного зі змінами видів тестування. Порівняння має містит такі блоки:

* що перевіряється;
* коли застосовується;
* обмеження;
* особливості.

|  | Функціональне тестування | Нефункціональне | Тестування пов’язане зі змінами |
| --- | --- | --- | --- |
| Що перевіряє | -Зовнішню поведінку системи, -тільки одну функцію.  -чи програмне забезпечення чи програми працюють належним чином без помилок. | наскільки добре працює програмне забезпечення чи програми, а не те, чи вони взагалі працюють.  -перевіряється безпека, надійність,живучість ,доступність, юзабіліті, масштабованість, ефективність, портативність, гнучкість | для підтвердження працездатності програми,  правильності здійсненого виправлення або дефекту |
| Коли застосовується | на етапі комплексного тестування і випробувань для визначення повноти реалізації функціональних завдань і їх відповідності вихідним вимогам. | на етапі системного тестування програмного забезпечення після завершення модульного та інтеграційного тестування | необхідно проводити після установки програмного забезпечення. |
| Обмеження | -можливість упущення логічних помилок у програмному забезпеченні;  -ймовірність надмірного тестування | -може бути дуже повторюваним, що не тільки забирає час, але й втомлює тестувальників  -досить дороге у використанні. | в залежності від виду тестування пов’язаного зі змінами, не завжди вдається охопити весь обсяг функціоналу, і тому помилку буде знайдено невчасно |
| Ообливості | імітує фактичне використання системи | -описує тести, необхідні для визначення характеристик програмного забезпечення, які можуть бути виміряні різними величинами. У цілому, це тестування того, «як» [система](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) працює.  - оцінює основні критерії користувача, такі як [продуктивність](https://www.zaptest.com/what-is-performance-testing-deep-dive-into-the-types-practices-tools-challenges-and-more) і зручність використання, а також перевіряє, чи працює програмне забезпечення так, як очікується, за межами його базової функціональності. | скоротити час, на пошук дрібних помилок, адже коли знайти більш масштабну, при її виправленні можуть прибратися і більш не такі вагомі, плюс це додаткова перевірка виправлених |
|  |  |  |  |

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

Ретестінг- перевіряє , програму чи систему після виправленого дефеку.Тобто при тестуванні знайшли дефект його виправили за допомогою ретестінгу протестували, все добре. А Регресія спрямована на перевірку всіх функцій продукту. Тобто зробили спочатку ретестінг виправили дефект. Далі подумали з якими ще функціями в програмі може бути по’язаний виправлений дефект і за допомогою регресивного тестування протестили чи виправлений дефект не змінив роботу всієї в цілому програми, що наявна раніше функціональність працює як і раніше.Якщо регресійне тестування не виявило дефектів то все добре.

Завдання 2

2 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.

На мою думку потрібно проводити обидва види тестування, тобто функціональне і нефункціональне. Так як функціональні види тестів розглядають зовнішню поведінку системи і базуються на функціях і особливостях системи.Зазвичай необхідно проводитит функціональне тестування перед нефункціональним, оскільки неможливо перевірити надійність або продуктивність функцій , які взагалі не працюють. А нефункціональні види тестування тестують в цілому “як” система працює.Нефункціональне тестування необхідне, оскільки воно оцінює основні критерії користувача, такі як продуктивність і зручність використання, а також перевіряє, чи працює програмне забезпечення так як очікується, за межами його базової функціональності.Нефункціональне тестування спрямоване на перевірку факторів, які впливають на зручність використання продукту, його надійність, ремонтопридатність , портативність та ефективність. Нефункціональне також зменшує ризик і витрати, пов’язані з випуском продукції на ринок, максимізуючи шанси команди випустити задовільний продукт.

Отже, зваживши вище перечислене нефункціональне тестування допомагає командам тестувальників краще зрозуміти програмне забезпечення, над яким вони працюють, але також може надати корисні знання , які допоможуть тестувальникам краще зрозуміти майбутні збірки.Без нефункціонального тестування команди тестувальників не змогли б перевірити , чи дійсно програмне забезпечення відповідає вимогам , викладеним у плані розробки програмного забезпечення.

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

Димове тестування означає мінімальний набір тестів на явні помилки. Цей тест зазвичай виконується як програмістом так і тестувальником. Smoke-тестування або перевірка збірки проводиться для того, щоб до запуску продукту переконатися, що все працює стабільно і відповідає вимогам замовника. Воно проводиться під час кожної нової збірки. У димового тестування багато переваг: воно допомагає помітити дефекти на ранньому етапі, підвищити якість системи та економить час команди.На мою думку Smok тестування є завжди доречним, так як програму, яка не пройшла такий тест немає сенсу передавати на глибше тестування.

Завдання 3

3.1. 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.

Test Case Description: Перевірка роботи додатку “ Обмін світлинами котиків”

| **Test Scenario:** | **Додати фото котика** |  |
| --- | --- | --- |
| **Step #** | **Step Detalis** | **Expected Results** |
| **1 Позитивний тест-кейс** | -відкрити застосунок “ Котики”  -на головному екрані натиснути вкладку меню  - натиснути додати фото  -вибрати наприклад “ загрузити фото з галереї  -вибрати фото котика, яке потрібно додати  -натиснути опублікувати фото | відкрилося вікно з написом “ загрузити фото з галереї”або “ зробити фото”  фото опубліковане |
| **2 Негативний тест-кейс** | -відкрити застосунок “ Котики”  -на головному екрані натиснути вкладку меню  - натиснути додати фото  -вибрати наприклад “ загрузити фото з галереї  -вибрати фото котика, яке потрібно додати  -натиснути опублікувати фото | -  -відкрилося вікно з написом “ загрузити фото з галереї”або “ зробити фото”  -користувач вибрав фото ще одного улюбленця песика  -можливо додати тільки фото котика, зробіть інший вибір |
| **Test Scenario:** | **Поставити вподобайку** |  |
| **Step #** | **Step Detalis** | **Expected Results** |
| **3 Позитивний тест-кейс** | -відкрити додаток котики  -на головній сторінці відкрити вкладку фото  -відкрити фото під яким необхідно поставити вподобайку  -під фото натиснути на сердечко, що означає вподобайка | вподобайка поставлена |
| **Test Scenario:** | **Додати коментар** |  |
| **Step #** | **Step Detalis** | **Expected Results** |
| **4. Позитивний тест-кейс** | -відкрити додаток котики  -натиснути на фото під яким буде коментар  -натиснути значок коментаря  - натиснути в строчці з мигаючим курсором.  - написати текст коментаря  -натиснути відправити | коментар відправлений |
| **Test Scenario:** | **Додати друга** |  |
| **Step #** | **Step Detalis** | **Expected Results** |
| **5. Позитивний тет-кейс** | * -відкрити додаток котики * - на головному екрані натиснути вкладку меню * -серед запропонованого вибрати “ знайти друзів” * - натиснути на лупу * - в пошуковій стрічці написати нікнейм друга якого необхідно додати * -вибираємо небхідного друга * -натискаємо “ додати друга” | -випаде список людей з подібним нікнеймом  -запит надіслано |
|  |  |  |

Test Case Description: Checking the operation of the Cat Photo Exchange app

| **Test Scenario:** | **Add a foto of the cat** |  |
| --- | --- | --- |
| **Step #** | **Step Detalis** | **Expected Results** |
| **1** | -open the SEALs application  -On the main screen, click the menu tab  - click add photo  -select for example " upload photo from gallery  -select the cat photo you want to add  -tap publish photo | a window opens  with the words "upload a photo from the gallery" or "take a photo"  the photo is published |
| **2** | -open the SEALs application  -On the main screen, click the menu tab  - click add photo  -select for example " upload photo from gallery  -select the cat photo you want to add  -tap publish photo | -a window opens with the words "upload a photo from the gallery" or "take a photo"  -the user has selected a photo of another pet dog  -it is possible to add only a photo of a cat, make another choice |
| **Test Scenario:** | **Like this page** |  |
| **Step #** | **Step Detalis** | **Expected Results** |
| **3** | -opens the application seals  -On the main page, open the photo tab  -open the photo under which you want to like  -under the photo, click on the heart, which means like | like it or dislike it |
| **Test Scenario:** | **Add a comment** |  |
| **Step #** | **Step Detalis** | **Expected Results** |
| **4.** | -open the application seals  -click on the photo under which the comment will be  -click the comment icon  - click on the line with the flashing cursor.  - write the text of the comment  -click send | comment sent |

| **Test Scenario:** | **Add a friend** |  |
| --- | --- | --- |
| **Step #** | **Step Detalis** | **Expected Results** |
| **5.** | * -open the SEALs application * - on the main screen, click the menu tab * -Select "find friends" from the menu that appears * - click on the magnifying glass * - in the search bar, write the nickname of the friend you want to add * -select the friend you want to add * -click "add friend" | -a list of people with a similar nickname will be displayed  -request sent |

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

1. Додаток повинен витримувати навантаження 100 тис. користувачів.
2. Система працює тільки у Windows і Linux.
3. Без доступу до інтернету продукт працює офлайн
4. При збої в програмі система відновлюється через 30 сек.
5. На головній сторінці вкладки мають вигляд квадратів синього кольору.