**Beet Seed — відпрацюй навички на базовому рівні.**

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

|  | **Функціональне тестування** | **Нефункціональне тестування** | **Пов’язане зі змінами тестування** |
| --- | --- | --- | --- |
| **що перевіряється** | Відповідність функціональних вимог ПЗ його реальним характеристикам. Тестування всіх функцій системи для підтвердження, що кожна функція програми працює відповідно до документації.  Відповідаємо на питання «Чи працює система?» | Перевірка тих аспектів ПЗ, які можуть бути описані в документації, але не відносяться до функцій програмних продуктів. Перевірка відповідності властивостей додатку його нефункціональним вимогам  Відповідає на питання: «Як добре працює система?» | Перевірка працездатності функціоналу, що існує, та перевірка на відсутність сторонніх помилок після оновлення ПЗ.  Для підтвердження виправлення помилки та роботи даного функціоналу. |
| **коли застосовується** | Відбуває на всіх рівнях тестування: компонентному або модульному (Component/Unit testing), інтеграційному (Integration testing), системному (System testing) і приймальному (Acceptance testing);  Функціональне тестування може бути проведено відповідно до специфікації, а також і на основі бізнес-процесу, тобто відповідно до знань системи. | Проводиться на етапі системного тестування програмного забезпечення після завершення модульного та інтеграційного тестування.  Один з останніх етапів тестування програмного забезпечення перед тестуванням прийняття користувачем і випуском фінального продукту. | Після проведення необхідних змін в ПЗ, виправлення дефекту |
| **обмеження** | Існує ймовірність пропустити кілька помилок логіки програмного забезпечення під час перевірки функціоналу програми.  Якщо певні функції замовник не врахував, розробник зобов'язаний врахувати їх, якщо вони відповідають ПЗ та впливають на кінцевий результат. | Потребує багато ресурсів, часу та знань. Має бути гнучким і відповідати всім вимогам ринку. | Повинно проводитися кожного разу, коли розробники оновлюють програмне забезпечення або коли змінюється код.  Може бути дуже трудомістким. |
| **особливості** | Проводиться в умовах близьких до реальних.  В рамках тестування ми «копіюємо» безпосереднє використання системи  Це перевірка, наскільки успішно працює ПЗ та виконує всі ті функції, які були закладені в нього розробником | Дозволяє контролювати глобальні властивості:  Надійність (реакція системи на непередбачені ситуації).  Продуктивність (Працездатність системи під різними навантаженнями).  Зручність (Дослідження зручності роботи із додатком з точки зору користувача).  Масштабованість (Вимоги до горизонтального або вертикального масштабування додатку).  Безпека (Захищеність даних користувача).  Здатність до портування (Доступність до переносу додатку на різні платформи).  Та багато інших якостей та особливостей ПЗ. | Підвищує загальну якість продукту.  Вузькоспрямоване тестування. |

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

При тестуванні змін в системі дуже важливо зрозуміти різницю між поняттями регресійне тестування (Regression testing) та повторне тестування (Retesting).

**Регресійне тестування (Regression testing)** проводиться з метою перевірки працездатності функціоналу, що існує, та перевірки на відсутність сторонніх помилок після оновлення ПЗ, внесення правок, доповнень в систему. В рамках регресійного тестування тест-кейси, які були відмічені раніше як «Passed», повинні бути всі перевірені повторно. Всеохоплююче тестування - може бути дуже трудомістким.

**Повторне тестування (Retesting)** проводиться для підтвердження виправлення помилки та роботи даного функціоналу. В рамках повторного тестування перевіряються тест-кейси тільки зі статусом «Failed». Вузьконапревлене на перевірку виправленої помилки - вимагає менше часу. Не вимагає нових налаштувань середовища тестування.

**Beet Sprout — детальніше заглибся в практику.**

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

Функціональне і нефункціональне тестування – це два різних, але однаково важливих типи тестування програмного забезпечення, які разом використовуються для оцінки того, чи відповідає програмне забезпечення вимогам користувачів, викладеним у технічному завданні.

Якщо в рамках функціонального тестування ми відповідаємо на питання «Чи працює система?», то нефункціональне відповідає на питання: «Як добре працює система?». Тому можемо зробити висновок, що ці два види тестування взаємодоповнюються та всебічно перевіряють продукт.

Функціональне та нефункціональне тестування відбувається на різних етапах тестування програмного забезпечення, але обидва типи тестування зазвичай проводяться на етапі системного тестування. Як функціональне, так і нефункціональне тестування може допомогти нам зрозуміти, наскільки добре працює продукт і чи адекватно він виконує свою роботу.

У той час як функціональне тестування перевіряє функції програмного забезпечення, нефункціональне тестування перевіряє інші аспекти, які можуть вплинути на продуктивність, ефективність, безпеку, надійність, стабільність тощо. Тому не раціонально перевіряти функцію без перевірки цих аспектів.

Функціональне тестування тестує функції програми і перевіряє, чи працюють вони так, як очікувалося. Нефункціональне тестування перевіряє інші аспекти програми, які впливають на задоволеність користувачів і якість програми - неможливо ізолювати функції від властивостей середовища та потреб кінцевих споживачів.

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

Метою **Smoke testing** є перевірка надійності і стабільності програмного забезпечення загалом, щоб дозволити подальше проведення ретельнішого тестування. Мінімальний набір тестів на явні помилки.

Короткий цикл тестів, що дозволяє відразу знаходити проблеми інтеграції; розкривати проблеми на ранній стадії; забезпечувати високий рівень впевненості в тому, що зміни в програмному забезпеченні не зробили несприятливої дії на основні області системи.

Здійснюється в тих випадках, коли QA отримують нову версію програми на тестування, при цьому вважаючи її відносно нестабільною. На цьому етапі необхідно переконатися, що надзвичайно важливі функції працюють згідно з очікуваннями. Концепція цього виду тестування полягає в тому, щоб виявити серйозні проблеми якомога раніше і повернути ПЗ на доопрацювання на ранньому етапі тестування. Це необхідно, щоб не поглиблюватися в складні тести і не витрачати час на явно неякісне програмне забезпечення.

Мотивом є перевірка стабільності нової зібраної версії продукту в цілому, тобто цей вид тестування не завжди доречно застосовувати, а можна одразу проводити **регресійне тестування (Regression testing).**

**Mighty Beet — різнобічно опануй тематику уроку.**

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

|  | **1. Login to your personal page** | | |
| --- | --- | --- | --- |
|  | Step Details | Test Data | Expected Results |
| 1 | Open the mobile application |  |  |
| 2 | Click on the button of your personal account |  | The login and password field opens |
| 3 | Enter your mobile phone number |  |  |
| 4 | Enter password |  | Successful login to your personal account |
|  |  |  |  |
|  | **2. Add a photo from the gallery** | | |
|  | **prerequisites: Open the application and log in to your personal page** | | |
|  | Step Details | Test Data | Expected Results |
| 1 | Click the + button on the main page |  | The New post page opens |
| 2 | Click on the Post button |  | Choose the type of publication |
| 3 | Click on the Gallery button |  | Open photos from the phone gallery |
| 4 | Select the photo you want to add |  |  |
| 5 | Click the Next button |  | A new post form opens |
| 6 | Click the Share button |  | Photo successfully published |
|  |  |  |  |
|  | **3. Add a photo to the archive** | | |
|  | **prerequisites: Open the application and log in to your personal page** | | |
|  | Step Details | Test Data | Expected Results |
| 1 | On the main page select the photo you want to send to the archive |  | Opens a photo |
| 2 | Press the ... |  | Opens the photo settings |
| 3 | Click the Archive button |  | Photo successfully archived |
|  |  |  |  |
|  | **4. Turn off photo comments** | | |
|  | **prerequisites: Open the application and log in to your personal page** | | |
|  | Step Details | Test Data | Expected Results |
| 1 | On the main page select the photo you want to send to the archive |  | Opens a photo |
| 2 | Press the ... |  | Opens the photo settings |
| 3 | Click the Turn off comments button |  | It is not possible to add comments to the photo |
|  |  |  |  |
|  | **5. Pin the photo** | | |
|  | **prerequisites: Open the application and log in to your personal page** | | |
|  | Step Details | Test Data | Expected Results |
| 1 | On the main page select the photo you want to send to the archive |  | Opens a photo |
| 2 | Press the ... |  | Opens the photo settings |
| 3 | Click the Pin to profile button |  | Photo successfully pinned. is displayed as the first photo |

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

* Система повинна бути захищена від несанкціонованого доступу. Тести безпеки оцінюють додаток та шукають вразливі місця в захисті програмного забезпечення. До них відносяться потенційні ризики безпеки, які можуть призвести до втрати даних або витоку конфіденційної інформації. Security testing – проводиться для відповіді на питання «Чи є додаток безпечним/захищеним чи ні?».
* Додаток повинен бути зручним та інтуїтивно зрозумілим у використанні.
* Кожна кнопка інтерфейсу має опис для підвищення рівня зрозумілості та простоти роботи за стосунку.

Usability testing – дослідження для визначення зручності використання ПЗ.

* Світлина додається менш ніж за 2 секунди. Перевірка швидкості роботи ПЗ або його окремих функцій за допомогою Performance testing.