## Рівні та види тестування

**Beet Seed**

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

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

|  | **Функціональне тест.** | **Нефункціональне тест.** |
| --- | --- | --- |
| **Що перевіряється?** | Функціональні вимоги | Нефункціональні вимоги |
| **Коли застосовується?** | Застосовується у випадку коли тестувальнику необхідно перевірити головні функціональні аспекти програми, чи співпадає продукт заявленим раніше вимогам, очікуваному результату. | * Коли функціональне тестування перевіряє ЩО робить програма? Нефункц. перевіряє ЯК вона це робить? * Використовується задля перевірки таких критеріїв як: Продуктивність, зручність використання і т.д., покриває всі випадки які не покриває функціональне тестування. |
| **Обмеження** | * За виконання функц. тестування можна не помітити критичні та логічні помилки в системі * Може виникнути можливість виконнання надмірного тестування, через що процес може йти дуже повільно, а витрати підвищуватися * Функц тестування не гарантує готовність продукту для демонстрації клієнту | * Помилки при нефункц тестуванні можуть призвести до “завалу” продукту, якщо, наприклад, не виокремлюється оптимальне stress та load навантаження на систему * Продукт може не пройти валідацію потенційним користувачем через високий поріг використання софту. * Від нефункц тестування залежить діапазон потенційних користувачів через конфігураційне тестування |
| **Особливості** | * Продукування якісного продукту із витриманими вимогами * Запевнення в правильності праці функціоналу. * Запевнення в тому, що продукт працює як треба * За тестування функціоналу можливо попередити потенційні дефекти на початку розробки продукту * Запевнює у захисті продукції | * Запевнення в тому, що продукт є user-friendly * Зацікавлені люди зможуть побачити рівень перформансу софту * Підвищує якість досвіду використання для користувача * За нефункц тестування можна надати інформацію про те, як система буде оперувати після можливих сбоїв роботи |

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

Регресія проводиться для вже пройдених тест кейсів, коли ретестинг проводиться виключно для failed тест кейсів. Регресія проводиться на нових білдах задля виявлення потенційних побочних ефектів у корректно працюючому функціоналі, ретест - верифікація того, що оригінальна проблема була вирішена. Регресія проводиться завжди, а ретест виникає тільки через припущення помилки у коді, тощо. Регресія працює із загальною поведінкою та характеристиками продукту, коли ретест фокусується на чомусь конкретному.

**Mighty Beet**

*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 вважаю, що для забезпечення максимально якісного продукту необхідно проводити функціональне тестування згідно фнц вимог і нефнц тестування згідно нефнц вимог, тому що це є два необхідні один одному процеси, як інь та янь. Чи можливо провести ТІЛЬКИ фнц тестування над продуктом? Можливо, але що за “продукт” вийде на ринок та до рук користувачів можна лише здогадуватися.

Наприклад, у нас є інтернет магазин - коли тільки ми його використовуємо через адмінку - він функціонує, зайшло 10 відвідувачів - сайт лежить через невиконання нефункц тестування, або в нас є функціонуючі кнопки які потрібні для здійснення покупок або навігації у системі, але вони розкидані по всьому дисплею, чорного кольору із чорним текстом, що призводить до припинення потоку користувачів та покупців. Система функціонально сама по собі МОЖЕ робити, але ЯК вона це буде робити - думки та теорії вкрай негативні.

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

Нам мою думку, смоук тестування є критичним аспектом задля забезпечення якості продукту котрий повинен проводитись завжди. Це є практика тестування фундаментальних та основних елементів ПЗ або будь-яких застосунків. За смоук тестування можна виявити критичні та блокуючи помилки на самих ранніх етапах життя білду, коли система є найнестабільнейшою. Смоук тестування проводиться задля дуже швидкого фідбеку про рівень життя софту, так qa зможуть для себе пояснити усі проблемні та потенційно неприємні моменти у тестуванні подальшого життя продукту.

**Mighty Beet**

| *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.  Завдння: Напиши 5 функціональних тест-кейсів, які перевіряли б роботу застосунку.*  1.   | ID | FTC 01 | | --- | --- | | Summary | Testing uploading valid format and file size cat photos | | Test data | Catphoto1.JPG(2Mb), Catphoto2.JPG(5Mb) | | Preconditions | Log-in into existing app account | | Test Steps | 1. Open homepage  2. Click “+” button in the headers left corner  3. Click “Upload” button  4. Select valid cat photos from your device (Catphoto1.JPG, Catphoto2.JPG)  5. Click “⇒” button | | ER | Users cat photos should appear in live feed |   2.   | ID | FTC 02 | | --- | --- | | Summary | Testing users sharing option of other social media accounts: Instagram | | Test data |  | | Preconditions | Log-in into existing app account | | Test Steps | 1. Click “My profile” button 2. Click “Options” button in the upper-right corner 3. Click “Edit Profile” button 4. Click “Share socials”  5. Select Instagram  6. Link existing Instagram account  7. Click “Done” button | | ER | An active link to Instagram account should appear in users profile |   3.   | ID | FTC 03 | | --- | --- | | Summary | Testing “App tutorial” function | | Test data |  | | Preconditions | Log-in into existing app account | | Test Steps | 1. Open KittyCats app  2. Click “Options” button  3. Select “Tutorial”  4. Complete “Navigation” tutorial  5. Complete “Describe your account” tutorial  6. Complete “Live feed” tutorial  7. Complete “Interactions” tutorial  8. Complete “Other functions” tutorial  9. Click “Done” button | | ER | User should receive a notification with a message: “Congratulations! You’ve completed the tutorial” |   4.   | ID | FTC 04 | | --- | --- | | Summary | Testing like button () in the live feed section under posts with photos | | Test data |  | | Preconditions | Log-in into existing app account | | Test Steps | 1. Click “live feed” button  2. Click under ANY post with a photo | | ER | The button should react with animation turning itself red |   5.   | ID | FTC 05 | | --- | --- | | Summary | Testing “Add Friend” function from users phone book | | Test data |  | | Preconditions | Log-in into existing app account | | Test Steps | 1. Click “My profile” 2. Click “Friends” in upper-middle part of the app  3. Click “Add Friends” button  4. Select “Phone book” option  5. Select friends numbers  6. Click “Add” button | | ER | Added users should be seen in “My friends” segment of users account |   *3. Напиши, які нефункціональні вимоги ти хотів/ла б застосувати для продукту твого стартапу.  Опиши перевірки, які б їх перевіряли (3-5 прикладів).* |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

1. Під час логіну до власного аккаунта користувач повинен мати можливість двоетапної верифікації за рахунок відбитка пальця.
2. Згідно дизайну окремі UI елементи мають бути такого коду: жовтий - **#F4dc14**, блакитний - **#1c74cc**
3. Застосунок має підтримуватися ОС Android 12

1.

| ID | TC1 |
| --- | --- |
| Summary | Checking user log-in fingerprint Two-Factor Authentication |
| Test data | login: q123123213213Q pass: Q123!QQ |
| Preconditions | 1.Create an account in application  2. Set up 2FA |
| Test Steps | 1. Open App  2. Click “Log-in‘  3. Enter valid login  4. Enter valid password  5. Click “Continue” button  6. Put a finger which was registered on 2FA on a fingerprint scanner of the device |
| ER | User should receive a message: “Welcome! You’ve successfully logged in” |

2.

| ID | TC2 |
| --- | --- |
| Summary | Check “blue” UI elements of a search bar |
| Test data | uicolors.pdf |
| Preconditions |  |
| Test Steps | 1. Open the app  2. Click “Search” button  3. Check color of “Search” button |
| ER | Background value is set to **#1c74cc** |

3.

| ID | TC3 |
| --- | --- |
| Summary | Checking apps compatibility with Android 12 |
| Test data | Xiaomi Redmi Note 11. Android 12 |
| Preconditions | Download and install the app |
| Test Steps | 1. Open the app |
| ER | The app should execute, UI elements be seen, can be interacted with |