## **Рівень 1**

| **Види тестування** | **Коли застосовується** | **Обмеження** | **0собливості** |
| --- | --- | --- | --- |
| **Функціональне тестування** | Функціональне тестування може бути проведено відповідно до специфікації, а також і на основі бізнес-процесу, тобто відповідно до знань системи. | В рамках функціонального тестування ми відповідаємо на питання «Чи працює система? Це тестування спрямоване на тестування всіх функцій системи для підтвердження, що кожна функція програми працює відповідно до документації. | 1. Підготовка тестових даних відповідно до описаної документації;  виходячи з бізнес-вимог.  2. Як частина функціонального тестування;  3. Отримання результатів на основі специфікації;  4.Проходження тест-кейсів;  5. Аналіз фактичних та очікуваних результатів. |
| **Нефункціональне тестування** | Нефункціональне тестування направлено на перевірку тих аспектів ПЗ, які можуть бути описані в документації, але не відносяться до функцій програмних продуктів. | Нефункціональне відповідає на питання: «Як добре працює система?». Нефункціональне тестування направлено на перевірку тих аспектів ПЗ, які можуть бути описані в документації, але не відносяться до функцій програмних продуктів | Види нефункціонального тестування:  1) тестування навантаження;  2) стресове тестування;  3) тестування стабільності або надійності;  4) об’ємне тестування (Volume Testing)  5) тестування установки (Installation testing)  6)тестування зручності користування (Usability Testing) 7) тестування на відмову і відновлення (Failover and Recovery Testing)  8) конфігураційне тестування (Configuration Testing) |
| **Тестування змін** | Повторне – виконання тестових скриптів, за допомогою яких тестувальник виявив баги під час останнього запуску ПЗ. Така перевірка здійснюється для підтвердження того, що тепер з ПЗ все добре.  Тестування збирання – спрямоване на перевірку відповідності версії ПЗ критеріям якості для того, щоб почати її тестування. По факту, це перевірка того, чи готова поточна версія ПЗ до її перевірки.  Санітарне – спрямоване на те, щоб довести, що конкретна функція працює відповідно до заявлених у специфікації вимог. | Цей вид тестування передбачає перевірку змін у ПЗ після виявлення та усунення багів, будь-яких змін в технічному середовищі чи банально – в замовленні користувача. | Особливість полягає в тому що після проведення необхідних змін, таких як виправлення бага / дефекту, програмне забезпечення повинне бути протестоване для підтвердження того факту, що проблема була дійсно вирішена. |

**2. Різниця між регресією та ретестингом**

Регресійне тестування виконується тільки при додаванні нової фічі (додаткова функціональність ПЗ) або істотній зміні функціоналу системи, а ретест виконується в тому ж оточенні й з тими ж даними, але на новому білді. Регрес можна проводити паралельно з повторним тестуванням. Повторне тестування має вищий пріоритет та має бути виконано до регресійного.

Тест-кейси в регресійному тестуванню можуть бути автоматизовані, а тест-кейси в повторному не можуть бути автоматизовані. В рамках регресійного тестування тест-кейси, які були відмічені раніше як «Passed», повинні бути перевірені повторно, а В рамках повторного тестування (ретест) перевіряються тест-кейси тільки зі статусом «Failed».

**РІВЕНЬ 2**

1. Я вважаю що неможливо проведення тільки функціонального тестування, без перевірки нефункціональних вимог. Нефункціональні вимоги відіграють не менш важливу роль, ніж функціональні. Вони дають інформацію про безпеку, справність та надійність системи, зручність системи, навантаження системи, допомагають з'ясувати, наскільки складно ПЗ можна перенести в інше середовище. Як функціональне, так і нефункціональне тестування потрібно, аби ПЗ відповідало всім вимогам користувача. Функціональні тести перевіряють, чи відповідає ПЗ визначеним вимогам, а нефункціональне тестування з'ясовує, наскільки добре працює система загалом. Нефункціональне тестування оцінює інші аспекти системи: перевірку продуктивності, зручності, надійності тощо. Обидва типи тестування однаково важливі, адже вони допомагають краще зрозуміти різні особливості системи. Найважливіша мета таких тестувань — забезпечити кінцевих користувачів якісним програмним продуктом.

**2. Димове тестування** (Smoke testing) також є аналогом тестування складання (Build Verification Testing) – це тип тестування програмного забезпечення, який включає невичерпний набір тестів, спрямованих на перевірку роботи найбільш важливих, критичних функцій в системі. Результат цього тестування використовується для визначення того, чи досить стабільна збірка, щоб продовжити подальше тестування. Димові тести здійснюються в тих випадках, коли тестувальники отримують нову версію (білд) програми на тестування, при цьому вважаючи її відносно нестабільною. На цьому етапі необхідно переконатися, що надзвичайно важливі функції AUT (Application Under Test) працюють згідно з очікуваннями.

Концепція цього виду тестування полягає в тому, щоб виявити серйозні проблеми якомога раніше і відхилити цей білд або повернути його на доопрацювання на ранньому етапі тестування. Це необхідно, щоб не поглиблюватися в складні тести і не витрачати час на явно неякісне програмне забезпечення. Метою димового тестування є перевірка надійності і стабільності програмного забезпечення загалом, щоб дозволити подальше проведення більш ретельного тестування

**Рівень 3**

**1. Функціональні тест- кейси мобільного застосунку " Пухнастики":**

1. Перевірка завантаження валідних даних (світлину котика) на мобільний застосунок “ Пухнастик”

2. Перевірка видалення світлини котика з мобільного застосунку “Пухнастик”

3. Перевірка завантаження невалідних даних (фотографії собак) на мобільний застосунок “ Пухнастик”

4. Перевірка можливості залишати коментарі під світлинами котиків у мобільному застосунку "Пухнастик”.

5. Перевірка можливості поставити вподобайки на світлину котика іншому користувачу в мобільному застосунку "Пухнастик”.

**2. Нефункціональні вимоги для мобільного застосунку "Пухнастик”:**

1. Перевірка інсталяції мобільного застосунку "Пухнастик” на телефон.

2. Перевірка зручності розташування кнопки “подобається” під фотографією котика.

3. Перевірити чи можливо додаток видалити і встановити заново.