**Перший рівень**

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 тестування** | **Нефункціональне тестування** | **Тестування, пов'язане зі змінами** |
| --- | --- | --- | --- |
| **Що перевіряється?** | Перевірка відповідності вимог до функціоналу його реальним функціям; перевіряємо те, що система робить. | Надійність; продуктивність; зручність використання; масштабованість; безпека; локалізація; конфігураційне тестування; доступності; тестування на відмову або відновлення. | Тестування відповідності випущеної версії білда критеріям якості (тестування білда/збірки); поверхневе тестування продукту/частини продукту на предмет працездатності і наявність швидкого знаходження критичних і дефектів, що блокують (димове тестування); чи конкретна функція працює згідно заявленим у специфікації вимогам (санітарне тестування); чи існуюча функціональність працює так, як раніше (регресійне тестування); ретест. |
| **Коли застосовується?** | На всіх етапах, починаючи з вимог, та на всіх рівнях (модульне, інтеграційне, системне та прийомочне). | Застосовується на тих етапах, де продукт/частину продукту вже зроблено та на всіх рівнях. | Застосовується на тих етапах, де продукт/частину продукту вже зроблено та на всіх рівнях. |
| **Обмеження** | - | - | - |
| **Особливості** | Тестуємо тільки функціонал. | Тестуємо як працює система, наскільки добре вона працює. | Проводиться тільки після змін. |

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

Ретестинг - це тестування окремого модуля або певного сегмента після того, як він *виправлений*, при виявленні помилок з попереднього тесту. Тут важливо відмітити, що повторне тестування не стосується впливу виправлення на інші компоненти в системі.

Процедура регресійного тестування призначена для виявлення нових помилок або «регресів» у існуючих функціональних та нефункціональних сферах програмної системи після внесення змін, нових фічів, оновлень. Його основна мета – перевірити, чи *впливають зміни* на інші компоненти програмного забезпечення та чи вносять вони нові помилки в систему.

**Другий рівень**

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

Ні. Навіть якщо в специфікації не описано нефункціональні вимоги, їх варто перевіряти, оскільки вони впливають на якість продукту. Наприклад, сайт може мати чудовий функціонал, але повільно завантажується чи не гарантує безпеку даних своїх користувачів.

* **Якщо так – в яких випадках?**

Ймовірно, тоді коли бюджет обмежений і замовник хоче тільки перевірку функціоналу. Наприклад, команда робить не продукт “під ключ” (повна розробка ПЗ, сайту, додатку і тд), а займається розробкою чи зміною певної частини/модуля продукту. Відповідно, відсутність нефункціональноі перевірки не впливає на якість продукту в цілому.

* **Якщо ні – чому?**

Читала, що бувають випадки, коли такі вимоги не описують. Однак, перевірка деяких нефункціональних вимог проводиться по замовчуванню, оскільки це впливає на якість продукту. Повторюсь, що на мою думку це залежить від того працює команда на продуктом повністю чи над окремим модулем.

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

Димове тестування спрямоване на поверхневу перевірку всіх модулів продукту на предмет працездатності, швидкого знаходження критичних дефектів та дефектів, що блокують. За результатами такого тестування робиться висновок про те, чи приймається продукт/окремий модуль в тестування, експлуатацію або на постачання замовникові.

Димове тестування є доречним завжди, оскільки таким чином використовуються більш раціонально ресурси, особливо часові. Наприклад, не витрачається час на тестування. Якщо продукт має блокуючий дефект і його в будь-якому випадку відправляють на виправлення дефекту, тоді доведеться робити додатково ретест та регресійне тестування.

**Третій рівень**

**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.**

**Вимоги до функціоналу:**

**Користувачі можуть завантажувати фотографії котиків.**

**Але не можуть завантажувати фотографії інших тварин/людей/об’єктів.**

**Користувачі можуть додавати друзів, ставити “вподобайки”, залишати коментарі.**

Файл з тест-кейсами: <https://docs.google.com/spreadsheets/d/1Ewci5ZB7Dy5BsUf8NDx0YCuHLhGjFQs2/edit?usp=sharing&ouid=103076507960248822347&rtpof=true&sd=true>

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

Нефункціональні вимоги:

* надійність;
* продуктивність;
* зручність використання;
* масштабованість;
* адаптивність для роботи з різними браузерами.

Опис перевірок:

* Надійність - (Stability / Reliability Testing, перевірка працездатності додатка при тривалому (багатогодинному) тестуванні із середнім рівнем навантаження).
* Продуктивність (Stress Testing, Stability / Reliability Testing та Load vs Performance Testing).
* зручність використання (Usability Testing, встановлення ступеня зручності використання, навчання, зрозумілості і привабливості для користувачів).