**1.1.**

Склади порівняльну таблицю функціонального, нефункціонального і пов’язаного зі змінами видів тестування.

|  | Функціональне | Нефункціональне | Регресійне |
| --- | --- | --- | --- |
| Що перевіряється | Коректність виконуваних операцій, з точки зору сценаріїв використання системи.  Перевіряється поведінка системи. | Глобальні властивості: безпека, ефективність, продуктивність, надійність та інше.  Перевіряється як працює система. | Відсутність або наявність помилок, які з'явилися при додаванні в існуючий білд нової частини програми або при виправленні інших багів. |
| Коли застосовується | В юніт-тестуваннях, інтеграційних і приймальних тестуваннях | В тестуваннях продуктивності, навантажувальному тестуванні та стрес-тестуванні, тобто в системному тестуванні. | Робиться в кожній новій версії продукту,  повторно виконуються всі тести, які були написані та проведені раніше. |
| Обмеження | Не перевіряє на продуктивність, ефективність | Не перевіряє систему на виконання тих функцій, які вимагає від неї замовник. | Регресійне тестування, в основному, не покриває весь додаток, а тільки ті ділянки, які зачепили зміни. |
| Особливості | Основним його завданням є підтвердження того, що програмний продукт, який розробляється, володіє усім необхідним замовнику функціоналом.  Імітує фактичне використання системи.  Функціональне тестування робиться перед нефункціональним тестуванням. | Має чіткі параметри або дані, які можна виміряти.  Допомогає оптимізувати налаштування продукту. | Активно використовується в екстремальній розробці.  Рекомендується проводити 3-5 разів, тому часто використовують засоби автоматизації тестування. |

**1.2.**

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

Регресійним тестуванням перевіряється, чи не зламалось щось в системі з додаванням нового функціоналу або виправленням іншого багу. Тест-кейси можна отримати з функціональних вимог або специфікацій. Перевіряється тільки та частина системи, в якій були зміни.

Ретест перевіряє, що виявлені дефекти виправлені, при тому, для ретесту використовується той самий тест-кейс, який до цього виявив цей баг. Перевіряються раніше виявлені баги, не залежно від того, де вони були знайдені.



**2.2.**

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.

В цілому система буде дієздатною після перевірки функціональним тестуванням. Але не думаю, що для продукту був би корисним такий підхід. Нефункціональні вимоги базуються на продуктивності, надійності, зручності, ремонтопридатності тощо. Ці всі фактори - про якість. Можливо, в тих проєктах, які не потребують подальшої масштабованості, якихось модифікацій, і треба, щоб воно “просто працювало з, умовно, однією тією функцією, що є” цей підхід буде ок. Але я не можу придумати, якщо чесно, приклад такої системи.

Якщо ж для більшості комерційних проєктів, орієнтованих на користувачів, якщо вони розраховані на подальше масштабування, на можливі зміни, покращення, то, думаю, без перевірки нефункціональних вимог не обійтись.

**2.3.**

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

Smoke test виконуються при оновленні системи, він доречний в тому разі, якщо це оновлення не вважається стабільним. Перевіряється чи працює критично важливий функціонал системи. Таке тестування дозволяє заощадити час і гроші. Якщо є якісь серйозні проблеми з системою - його легше відправити на доопрацювання, а не витрачати час на довгі тести системи, яка не працює.

**3.2.**

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.

[Test-suite KittyApp](https://docs.google.com/spreadsheets/d/18XoP4reXBnjeReSCaxSy5N_zbhxIOPcbIth1qCmNP-Y/edit?usp=sharing)

**3.3.**

Напиши, які нефункціональні вимоги ти хотів/ла б застосувати для продукту твого стартапу.

Опиши перевірки, які б їх перевіряли (3-5 прикладів).