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

Р[егресивне тестування](https://tryqa.com/what-is-regression-testing-in-software/) проводиться для того, щоб переконатися, що існуюча функціональність працює нормально і немає побічних ефектів. Іншими словами, регресійне тестування перевіряє, чи не з'явилися нові [дефекти](https://tryqa.com/what-is-defect-or-bugs-or-faults-in-software-testing/) в раніше існуючій функціональності.

[Повторне тестування](https://tryqa.com/what-is-retesting/) проводиться під час [тестування програмного забезпечення](https://tryqa.com/what-is-a-software-testing/), щоб переконатися, що певний дефект був виправлений і функціональність працює належним чином.

Регресійне тестування проводиться для пройдених тест-кейсів

Повторне тестування проводиться тільки для невдалих тест-кейсів

Регресійне тестування перевіряє наявність несподіваних побічних ефектів

Повторне тестування гарантує, що початкова несправність була усунена

Основна відмінність між повторним і регресивним тестуванням полягає в тому, що повторне тестування призначене для перевірки на наявність відомих помилок і використовується для підтвердження того, що відповідна помилка була виправлена в цілому.

Регресійне тестування відрізняється тим, що воно шукає в додатку невідомі помилки, які могли виникнути через якусь впроваджену зміну.

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

|  | Що перевіряється | Коли застосовується | Обмеження | Особливості |
| --- | --- | --- | --- | --- |
| Функціонувальне тестування | чи відповідає кожна функція вимогам | Коли розробляюьбся нові функції сисстеми | Вхідними даними є бізнес вимоги | легко визначити вимоги |
|  | чи може користувач це зробити» або «чи працює ця конкретна функція». |  |  | частіше виконуєтья мануально |
|  | конкретну дію або функцію коду |  |  |  |
|  | чи правильно взаємодіють модулі та компоненти |  |  |  |
|  | чи працюють всі елементи інтерфейсу належним чином |  |  |  |
| Нефункціональне тестування | перевіряє систему в умовах стресу, оцінює здатність системи до регенераці | Після завершення функціонального тестування | базується на очікуванні клієнта | повинно проводитися кожного разу, коли розробники оновлюють ПЗ або коли змінюється код |
|  | успішну інсталяцію та налаштування а також оновлення чи видалення ПЗ |  |  | виконується автоматизовано |
|  | зручність і зрозумілість, привабливость для користувачів |  |  |  |
|  | безпеку системи |  |  |  |
| Тестування повязане із змінами | що після збірки коду (нового чи виправленого) ПЗ запускається та виконує основні важливі функції | Після внесення змін до системи | виконується лише при додаванні нової фічі | потребує частого оновлення тестових сценаріїв відповідно до змін в системі |
|  | що після внесення змін, проведених у самому додатку або середовищі функціональність працює, як і раніше |  |  |  |
|  | що дефект виправлений та змінений функціонал працює як треба. |  |  |  |
|  | визначення працездатності певних частин додатків після внесення змін |  |  |  |
|  |  |  |  |  |

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

Ми перевіряємо, чи працюють важливі функції

Це міні-та швидкий регресійний тест основних функціональних можливостей.

Завдання – перевірити, чи працює система в загальних рисах, без детальної перевірки.

Проводиться після випуску нової збірки або після значних змін коду

проводиться щоразу, коли розробляються нові функції ПЗ

Якщо ми не проведемо перевірку на ранніх стадіях, можемо виявити дефекти на пізніх тестах а це буде коштувати дорого

**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**

Хоча нефункціональне тестування і не перевіряє систему на виконання тих функцій, які вимагає від неї замовник, воно дозволяє контролювати більш глобальні властивості: безпеку, продуктивність, надійність Нефункціональне тестування може допомогти тестувальникам і розробникам покращити загальну продуктивність програмних додатків Нефункціональне тестування – найкращий спосіб зробити ваше програмне забезпечення більш зручним для користувача

**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: На**пиши 5 функціональних тест-кейсів, які перевіряли б роботу застосунку.

| id | Summary | Preconditions | Steps | Expected result | Priority |
| --- | --- | --- | --- | --- | --- |
| 1 | Завантаження фотографій котиків | Відкрити додаток | Натиснути копку Завантажити | Фотосвітлина котика зявиться в додатку | High |
|  |  |  | Вибрати фото котика з галереї |  |  |
| 2 | Неможливість завантаження | Відкрити додаток | Натиснути копку Завантажити | Система видасть помилку | High |
|  | фотографії інших тварин/людей/об’єктів |  | Вибрати фото людини з галереї |  |  |
| 3 | Додавання друзів | Відкрити додаток | Натиснути копку Додати друзів | Друг появиться в списку друзів | Normal |
|  |  |  | Вибрати із запропонованих уже зареєстрованих контактів |  |  |
|  |  |  | Натистути додати |  |  |
| 4 | Ставлення вподобайки | Відкрити додаток | Переглянути стрічку із світлини котиків | Сердечко під фото міняє колір на червоний | Normal |
|  |  |  | Натиснути іконку сердечко під вибраним фото | вподобайка поставлена |  |
| 5 | Залишення коментарів | Відкрити додаток | Переглянути стрічку із світлини котиків | Коментарій зявився під фото | Normal |
|  |  |  | Вибрати світлину |  |  |
|  |  |  | Натиснути коментувати |  |  |
|  |  |  |  |  |  |