:росток:Beet Seed — відпрацюй навички на базовому рівні.

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

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

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

*Відповіді до завдань запиши в Google Doc документ. Додай посилання на нього в LMS.*

1.

|  | Функціональне тестування | Нефункціональне тестування | Тестування змін |
| --- | --- | --- | --- |
| Що перевіряється | Перевіряються функції системи, її основні характеристики, а також чи відповідають вони заявленим вимогам і специфікаціям | Перевіряються нефункціональні вимоги до ПЗ, наприклад витривалість, безпека, зручність використання, продуктивність і т.д. | Назва говорить сама за себе, перевіряються внесені зміни і чи коректно вони функціонують, а також чи не впливають ці зміни на попередньо затверджені характеристики ПЗ чи сайту. |
| Коли застосовується | Під час інтеграційного виду тестування, перед нефункціональним тестуванням. Таке тестування необхідно застосовувати для верифікації функціоналу, тобто чи виконує програмне забезпечення свої функції вірно та відповідно до вимог. | Після функціонального тестування. В моменти, коли необхідно перевірити як добре працює система. | Коли були внесені будь-які зміни, наприклад новий функціонал, або нова локалізація, або на новому білді були виправлені старі дефекти і т.п. |
| Обмеження | Якщо необхідно перевірити взаємодію програми з реальними користувачами або інші аспекти, пов'язані з користувацьким досвідом, це може вимагати інших методів тестування. І звичайно якщо треба перевірити безпеку ПЗ, чи стійкість ПЗ, чи інші нефункціональні вимоги, то цей вид тестування не підходить. Але в будь-якому разі, це вид тестування, який найчастіше використовують. | Якщо головним об'єктом тестування є перевірка бізнес-логіки та функціональності програми (наприклад, правильність алгоритмів, обробка даних тощо), то нефункціональне тестування може бути менш важливим або взагалі не потрібним.  А також інколи деякі підвиди нефункціонального тестування вимагають або більшої експертизи (наприклад тестування безпеки), або ж більше ресурсів (наприклад якщо треба перевірити навантаження сайту з великою кількістю нових користувачів). | У разі відсутності змін, звичайно, подібне тестування не потрібне, але відсутність помилок майже неможлива, тож цей вид тестування може відбуватись абсолютно на будь-якій стадії розробки софта.  Також такий вид тестування на жаль не може бути автоматизованим, адже вимагає доволі багато часу для тестування не тільки нових функцій, а і перевірки попередніх тест-кейсів. |
| Особливості | 1. Кожна функція ПЗ чи сайту перевіряється шляхом надання відповідних тестових вхідних даних, тобто є певні очікування до вихідного результату, а також порівняння фактичного результату (під час тесту) з очікуваним результатом, прописаним до цього. 2. Таке тестування допомагає заводовольними потреби як користувачів, так і бізнесу, адже перевіряємо ми ПЗ з точки зору користувача, але дивимось, чи працює ПЗ відповідно до цілей, які переслідує в першу чергу бізнес. 3. Тестування проходить в умовах, близьких до “реальних”, тобто з таким тестуванням ми максимально наближаємось до User Experience. | 1. Такий спосіб тестування однозначно підвищує рівень продуктивності та надійності програмного забезпечення. 2. Основна мета нефункціонального тестування - це задовольнити потреби користувачів, тобто таке тестування майже повністю на стороні майбутніх юзерів. 3. Має багато способів перевірки навантаження, безпеки, надійності, зручності використання, тобто таке тестування є вичерпним і охоплює величезну кількість аспектів. | 1. Це тестування підвищує загальний рівень якості продукту, адже саме тестування змін дає остаточну впевненість, що продукт функціонує добре і є покращеною версією. 2. Це тестування, яке може стосуватись як функціонального, так і нефункціонального виду, бо зміни можуть бути абсолютно будь-які і вони потребують перевірки. 3. Має такі підвиди: смоук тестування (чи запускається програма взагалі), рестест (чи виправлені помилки і чи коректно працює), регресійне тестування (чи внесені узгоджені попередньо зміни і чи це не повпливало на інший функціонал), санітарне тестування (чи відповідає певний функціонал специфікації вимог). |

2. Різниця між регресією та ретестингом

**Регресійне тестування** - це перевірка нового, додаткового функціоналу ПЗ, а також його сумісність із тим функціоналом, що вже існував. Тобто основна задача такого тестування - це впевнитись, що нові фічі ПЗ добре функціонують і добре взаємодіють із старими.

**Ретестинг** - це перевірка того ж самого функціоналу ПЗ, але з виправленими в них помилками, що були знайдені раніше. Фактично кажучи, це перевірка тих тест-кейсів, котрі мали статус “Failed” і мали заведені на них баг репорти. І задача тестувальника під час такої перевірки - впевнитись, що багів на новому білді більше немає. Також, ретестинг має вищий пріоритет і має бути проведений до регресійного тестування.