**Рівні та види тестування**

**Перший рівень** — 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).

|  |  |  |
| --- | --- | --- |
| **Функціональне** | **Нефункціональне** | **Пов’язане зі змінами** |
| Перевіряємо основний функціонал, який має працювати. Головні функції продукту. Наприклад, якщо це мобільний застосунок для обміну світлинами, то чи завантажується він на телефон, чи можна в ньому зареєструватися, чи можна завантажувати світлини в додаток, чи можна їх відправляти іншим користувачам (можливо не прямо всі функції перечислив, але сенс такий). | Тут вже перевіряємо взаємодію продукту із навколишнім середовищем (це якщо прям загально сказати). Ми перевіряємо, на тому ж прикладі застосунку, що якщо кількість користувачів буде більша, ніж очікується на перших етапах користування, як себе буде вести застосунок. Якщо змінимо базу даних, як це вплине на роботу застосунку. Скільки часу займає загрузити світлину, відправити її іншому користувачу. | Тут ми перевіряємо вже роботу нашого продукту загалом при внесенні змін. Якщо наприклад зробили оновлення, і додали ще якийсь функціонал у наш продукт. Якщо в нашому додатку із обміну світлинами ми додали функціонал, що на наші світлини можуть ставити лайки або писати коментарі. |
| Застосовується перш за все при модульному тестуванні. І в інших рівнях тестування теж, але при модульному мені здається найбільше. | Застосовується при інтеграційному тестуванні більше за все. Ну в інших рівнях теж присутнє, просто тут більше за все | Застосовується вже більше при системному тестуванні. |
| Ми можемо перевірити одну із основних функції програми, але не можемо перевірити взаємодію однієї функції із іншою. Наприклад, на сторінці Розетка ми б не могли при такому тестування перевірити взаємодію «Додати товар у кошик» та «Оплатити його» разом, так як це різні функції, і тестування кожного проводиться окремо. | В цьому виді тестування ми перевіряємо вплив якогось фактору на працездатність продукту, а не правильність спрацювання окремого функціонала. | Цей вид тестування не є автономним. Завжди є доповненням і продовженням функціонального і нефункціонального. |
| Із особливостей, це тестування однієї конкретної  функції. | Особливість цього виду, що ми можемо протестувати, як витримає наш продукт певні навантаження. | В продукт внесені зміни. |

2. **Різниця між регресією і ретестингом.**

Ретестинг, це коли ми перевіряємо функціонал, який перед цим вже перевіряли, і виявили в ньому баг, і відправили на виправлення. Регресія - це перевірка того, що і так працювало, щоб переконатися, що зміни в суміжніх частинах не зламали його. Якщо на прикладах, то нам дали перевірити, чи працює функція оплати карткою VISA на сайті, ми перевірили, і виявилось, що не працює, і відправили на виправлення багу. Коли ми будемо другий раз перевіряти після виправлення, це вже буде ретест. А регресія, це коли у ту ж функцію оплати на сайті додали можливість сплачувати крім картки VISA ще й MASTERCARD. І така перевірка вже буде регресія, бо внесли зміни і ми перевіряємо вже зміни при оплаті карткою на сайті.

**Другий рівень** — детальніше заглибся в практику.

1. Виконай завдання попереднього рівня.

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.

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

1.      На мою думку тільки одного функціонального тестування недостатньо, тому що нам важливо, як буде працювати продукт при навантаженні. Наприклад, якщо ми розробляємо програмне забезпечення для банку. Для нас вкрай важливо, скільки користувачів одночасно може використовувати дане ПЗ, бо якщо при великому навантаженні ПЗ ляже, то це принесе великі збитки для банку. Навіть із боку не лише користувацького це важливо, а й зі сторони бізнесу. Обидві сторони зазнають негативний вплив, користувачі будуть засмучені несправністю ПЗ, а власник понесе збитки.

2.      Smoke тестування – це корисна і навіть важлива річ. Я вважаю, що в ній багато плюсів, найголовніші із них  – це економія часу тестувальників, якщо ми не можемо перевірити власне основний функціонал, то навіщо витрачати час на розроблення тест плану того, що не можна протестувати. І це економія фінансів замовника, тому що він не витрачає фінанси на оплату роботи, яку неможливо зараз виконати.