LVL1

|  | Що перевіряється | Коли застосовується | Обмеження | Особливості |
| --- | --- | --- | --- | --- |
| Функціональний вид | Виявлення невідповідностей між реальною поведінкою реалізованих функцій і очікуваною поведінкою відповідно до специфікації і вимог | Використовується на процесі комплексного тестування для визначення повноти реалізації функціональних задач і їхньої відповідності вимогам | Оскільки функціональне тестування це детальний процес, його виконання може займати багато часу. | Дає гарантії того, що проєкт відповідає вимогам користувача.  Знижує ризики та фінансові затрати на розробку продукту. |
| Нефункціональний вид | Перевіряє зокрема надійність, продуктивність, зручність та безпеку програмного забезпечення | Використовується для перевірки готовності системи по нефункціональним параметрам, таких як продуктивність, зручність використання, безпека | Кожен раз, коли оновлюється ПЗ, потрібно знову проводити нефункціональне тестування. | Забезпечує високий рівень безпеки.  Гарантує здатність системи до навантажень.  Підвищує ефективність системи. |
| Пов’язаний зі змінами | Перевіряє підтвердження того факту, що після проведення необхідних змін, таких як виправлення бага або дефекта, проблема була дійсно вирішена | Використовується для визначення працездатності певної частини програми, після змін, зроблених в ній | Потрібно проводити після кожної зміни в коді.  Якщо цей процес не автоматизований, забирає багато часу. | Підвищує якість продукту.  Гарантує, що виправлені помилки та дефекти не повторяться. |

Регресія vs Ретестинг

1. Ретестинг не обов’язково має бути частиною процесу тестування, якщо помилка не знайдена і не виправлена, тоді як регресійне тестування завжди є частиною процесу тестування і проводиться кожен раз, коли в код вносять зміни.
2. Регресійне тестування часто досліджує великі частини програми і тому є більш трудомістким, натомість ретестинг досліджує лише певний дефект і тому займає менше часу.
3. Регресійне тестування виконується для пройдених тестових кейсів, тоді як ретестинг виконується лише для провальних тестових кейсів.
4. Регресійне тестування можливе з використанням автоматизації, ретестинг виконується тільки вручну.
5. Ретестинг включає верифікацію дефектів, регресійне тестування - ні.

LVL2

На мою думку не проводити нефункціональне тестування можливо, в цілому, наприклад, в тих випадках, коли у нас не вистачає працівників або ми не встигаємо провести нефункціональне тестування до релізу. В такому випадку краще провести якісне функціональне тестування, щоб зрозуміти, що наш продукт відповідає всім вимогам користувачів, але, без проведення нефункціонального тестування наш продукт буде неякісним в питаннях безпеки, продуктивності, юзабіліті, надійності. Тому нібито вище написане суперечить одне одному, як висновок, хочу сказати, що в контексті “чи можливо взагалі” не проводити нефункціональне тестування я б відповів, що можливо, але в кінцевому результаті, у нас вийде продукт, який не корректно працюватиме при першому навантаженні, на різних операційних системах або користувачам буде незручно ним користуватись, тому випускати такий продукт немає ніякого сенсу.

Smoke-тестування - це тип тестування, який включає в себе набір тетсів, спрямованих на перевірку роботи найбільш важливих, критичних функцій системи. В результаті такого тестування ми бачимо, чи дійсно ми зробили те, що хотіли. Наприклад, якщо ми працювали над кошиком для інтернет-магазину, відправили цей модуль на тестування, тестувальники проводять smoke-тестування і в процесі виявляється, що функція додавання товару до кошика не працює, а це є основним функціоналом, то в застосуванні інших видів тестування не має сенсу. Я вважаю, що проведення smoke-тестування є завжди доречним, тому що ми бачимо чи має сенс використовувати той чи інший модуль в нашому проекті.