**1.**

|  | Функціональне | Нефункціональне | Пов’язане зі змінами |
| --- | --- | --- | --- |
| Що перевіряється | Функціонал чи його частинка.  Чи виконує програма функції,які на неї покладені згідно функціональних вимог | Наскільки стабільно працює програма, її навантаженість, як працює в стресовій ситуації, чи може сама відновитися після якогось крешу,як працює програма на різних пристроях, наскільки зручна вона для користувача | Перевіряється по-перше, те, що змінилося, де був дефект. І, по-друге, той функціонал, який міг бути пов'язаний з виправленою частинкою. Чи все знову працює коректно. |
| Коли застосовується | На етапі кодування продукту і на всіх етапах STLC | на всіх етапах тестування продукту | Після виправлення дефектів і перед продакшином |
| Обмеження | пріоритетність завдань, наприклад, спочатку тестуємо те, що не працює частинка основного функціоналу. Тут нам важливо щоб програма виконувала функції, які на неї покладені | пріоритетність, вибираємо на нашу думку найважливіші або ті, що ще не перевірялися. | перевіряємо не все, а те, що найшвидше могло поламатися, якщо ми обмежені в часі |
| Особливості | Відповідає на питання: Що робить продукт? | Відповідає на питання: Як саме працює продукт? Наскільки продуктивно, красиво, зручно. | Завжди пов”язане зі змінами. |

Маю питання. Чи правильно я зрозуміла, що мається на увазі під пунктом”Обмеження”? Дякую.

**2.Різниця між регресією та ретестингом.**

Ретестинг- це повторне тестування, це ми знайшли дефект, виправили і перевірили цей кусочок функціоналу чи вже немає цього дефекту. Немає. Ок.

А Регресія- це ми після ретестингу думаєм, що цей ретестинг міг зачепити, може щось почало працювати некоректно. Аналізуєм які поля найбільше пов'язані і їх перевіряєм. І на всяк випадок, перевіряєм чи працює весь функціонал так само як до того (тільки без дефекту,який ми виправили).