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

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

|  | Функціональне тестування | Нефункціональне тестування | Тестування повязане зі змінами |
| --- | --- | --- | --- |
| Що перевіряється | аналіз функціональних характеристик додатка та перевірку на невідповідності між реальною поведінкою реалізованих функцій і очікуваною поведінкою відповідно до специфікації і бізнес-вимоги | включає тестування нефункціональних вимог системи, таких як продуктивність, безпека, масштабованість, зручність використання, надійність тощо | визначити, чи впливають зміни на існуючий функціонал системи |
| Коли застосовується | можуть проводитися на усіх рівнях тестування (компонентному, системному, приймальному) | вимірювання якості характеристик може бути менш об'єктивним і менш певним, ніж перевірка конкретних функцій. | На будь-якому рівні, де застосовувалися зміни |
| Обмеження | фокусується виключно на функціональності, а не на інших аспектах якості, таких як продуктивність, безпека або доступність | вимірювання якості характеристик може бути менш об'єктивним і менш певним, ніж перевірка конкретних функцій | Перш ніж провести тестування, слід переконатися, що існуючі функції працюють коректно (функціональне тестування) і що нефункціональні аспекти, такі як безпека і продуктивність, не порушуються. |
| Особливості | Функціональне тестування імітує фактичне використання системи | Нефункціональне тестування важливо для забезпечення того, що програмний продукт не тільки працює коректно, але й задовольняє потреби користувачів у відповідній якості та надійності. | допомагає забезпечити стабільність та якість продукту після внесення змін, знижує ризик помилок та допомагає зберегти задоволення користувачів |

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

Ретестинг нам показує, що проблема, на яку був заведений bug report, уже вирішена, її пофіксили. Регресія покаже, чи фікс цього багу не зламав взаємодію інтегрованих блоків у системі. При ретестингу виконують функціональне тестування, при регресії- це фенкціональне та нефункціональне тестування. Також ретестинг виконується без зміни середовища, без додавання нових функцій. Регресія може виконуватися при зміні середовища тестування, змінено функціонал, додані нові функції.

3. 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.

Функціональне тестування і нефункціональне тестування взаємопов'язані і доповнюють одне одного. І, хоча можливе проведення тільки функціонального тестування для деяких продуктів чи на деяких етапах розробки, в більшості випадків обидва види тестування є важливими для забезпечення якості програмного продукту.

Для дуже простих програм або інструментів, які мають мінімальну функціональність та не вимагають високої продуктивності, безпеки або інших нефункціональних характеристик, може бути можливим проведення тільки функціонального тестування

Для критичних застосунків, таких як медичне програмне забезпечення, фінансові системи або системи безпеки, нефункціональні аспекти, такі як безпека та надійність, є життєво важливими. Тут функціональне тестування не забезпечить належного рівня якості.

Також для систем, які мають високу продуктивність або обробляють великий обсяг даних (наприклад, веб-сервери або бази даних), нефункціональне тестування, таке як тестування продуктивності, є критичним для забезпечення ефективності роботи системи.

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

Димове (smoke) тестування - це вид тестування, який призначений для швидкого перевірки основної функціональності програмного продукту після внесення змін або після випуску нової версії. Його мета - переконатися, що основні функції продукту працюють без важливих помилок. Ось як я розумію необхідність та доречність димового тестування:

* відразу знаходяться проблеми інтеграції;
* тестування розкриває проблеми на ранній стадії;
* до випуску нової версії продукту важливо виконати димове тестування, щоб переконатися, що основні функції готові до використання користувачами;
* димове тестування спрощує процес перевірки функціональності, оскільки фокусується на основних аспектах продукту. Це дозволяє ефективно використовувати ресурси та прискорює випуск нових версій продукту.