**Перший рівень — відпрацюй навички на базовому рівні.**

1. Склади порівняльну таблицю функціонального, нефункціонального і пов’язаного зі змінами видів тестування.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Функциональное** | **Нефункциональное** | **Изменений** |
| **Что проверяется** | Все функции системы на соответствие документации | Проверяются характеристики системы | Определенный компонент и/или системы целиком |
| **Когда используется** | На каждом уровне тестирования от компонентного до приемочного | На каждом уровне тестирования от компонентного до приемочного | После внесения изменений в систему (багфиксов, изменений требований и тд) |
| **Ограничения** | Не проверяются характеристики системы | Не проверяются функции системы | - |
| **Особенности** | Включает UI-testing; security testing; compatibility testing etc.  Может быть использован подход на базе требований и подход на базе | Включает performance; installation; localization; internationalization; configuration; usability; failover&recovery; stability/reliability testing etc. | Включает smoke; sanity; regression; retest. |

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

Регрессия – это выфвление дефектов в уже протестированных частях системы, а ретестинг – проверка пофикшенных багов. Регрессия покрывает весь имеющийся функционал, а ретест подразумевает конкретную точечную проверку. Регрессию удобно выполнять с помощью автотестов, а ретест практичнее проверять вручную. Регрессия проводится в основном после обновления системы, а ретест – после того, как исправлен конкретный баг и эта правка донесена в систему.

**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.

|  |  |
| --- | --- |
| Я считаю, что грамотно – проводить тестирование как функциональное, так и нефункциональное. Представим, что мы проверили системы функциональным тестированием и все прошло успешно. Функции работают в соответствии и документацией, и система выпускается в продакшн. Но без нефункциональных тестов она может как минимум быть не настолько эффективной, как могла бы быть. А, возможно, и вовсе не сможет существовать и функционировать. Например, мобильное приложение без проведенного installation test может оказаться невозможным для скачивания и установки на телефоны и планшеты. Интернет-магазин без Usability test может быть настолько неудобным для покупателей, что они будут уходить на другую более интуитивную площадку, не совершив здесь покупку. А сайт прогноза погоды не сможет достаточно быстро выдавать таблицу температур, когда в 7 утра на него зайдет максимальное количество посетителей, потому что не был проведен performance test. |  |

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

Smoke test – это проверка самых критичных функциональностей системы. Он проводится, как правило, после обновления системы. Он необходим для того, чтобы в краткие сроки дать ответ на вопрос и жизнеспособноости новой версии системы. Если в ходе smoke определяются критические ошибки, то может быть принято решение об откате системы к предыдущей работавшей версии. При положительном прохождении smoke test делается вывод, что самые важные функции системы работают корректно, и можно углубляться в более детальное тестирование. Всегда ли уместен smoke test? Считаю, что нет. Неразумно и слишком затратно, например, было бы проводить smoke после каждой новой фичи, внедренной в среде разработки.