1 рівень

1.

| Вид тестування | Що перевіряється | Коли застосовується | Обмеження | Особливості |
| --- | --- | --- | --- | --- |
| Функціональне | Чи працює система згідно до вимог | На всіх етапах, особливо при перевірці функціоналу | Не включає нефункціональні аспекти типу безпеки та продуктивності | Може виконуватись вручну або автоматизовано, перевіряє чи відповідає система вимогам |
| Нефункціональне | Продуктивність, стабільність, зручність, безпека | Після функціонального тестування, перед релізом | Не перевіряє, чи виконує система правильно дії | Потребує спеціальних інструментів, фокус на користувачі і стабільності |
| Пов’язане зі змінами | Чи не зламалось щось після змін | Після оновлень, виправлень або змін у коді | Не виявляє нових багів поза зоною змін, потребує оновлення тестів | Забезпечує стабільність вже існуючого функціоналу |

2.

Ретестинг - це тестування, яке перевіряє чи дійсно був виправлений конкретний дефект. І його головне завдання - перевірити чи не з’явився той самий дефект знову.

Регресійне тестування - це тестування спрямоване на перевірку частини програмного забезпечення, яке було раніше протестоване з метою забезпечення, що внесені зміни не викликали помилок десь у іншій частині функціоналу, яка раніше працювала без помилок. Його обсяг ширший ніж у ретестинга.

2 рівень

1. Взагалі, так, функціональне тестування може бути проведене без нефункціонального в випадках, коли продукт тільки формується, і головна задача це перевірити чи працює наприклад певна кнопка або продукт зроблений виключно для тесту.

Але без нефункціонального тестування ми не можемо впевнитись, що система буде працювати стабільно, без підвисань та незручностей, або взагалі не перестане працювати при роботі декількох сотен пристроїв.

Тому, можна сказати, що функціональне тестування можливе без нефункціонального, але для того щоб продукт був якісний і повноцінний, то обов'язково потрібна перевірка нефункціональних вимог.

2. Мета димового тестуванння - швидко визначити чи стабільна базова функціональність продукту щоб уникнути витрату часу на більш детальне тестування, якщо є критичні збої. Особливо це доречно у проектах з частими оновленнями. І навпаки димове тестування може бути зайвим, якщо проект вже стабільний, і зміни відбуваються мінімальні, або це невеликий проект з мінімальною функціональністю.

3 рівень

1. [Урок 6. Test Case.xlsx](https://docs.google.com/spreadsheets/d/1kh6AZ_8hYGfrNwJTiPc27UZt6hdRfFvO/edit?usp=sharing&ouid=104240034735240506994&rtpof=true&sd=true)

2.

1) Завантаження стрічки фотографій застосунку не довше ніж за 2 секунди. (можна виміряти час відкриття головного екрану, використати декілька пристроїв та емуляторів для перевірки результатів залежно від пристрою).

2) Застосунок має коректно працювати на iOS та Android. (можна запустити застосунок на реальних власних пристроях та використати емулятори для перевірки).

3) Застосунок має стабільно обробляти не менше ніж 1000 користувачів одночасно. (можна змоделювати вхід необхідну кількість одночасних користувачів методом навантажувального тестування, та відстежувати результати).

4) Перевірити зручність користування застосунком (додаванням фотографій, друзів) (можна провести тест із 5 новими користувачами, не знайомі з застосунком, які нададуть відгук про складність використання та поради щодо покращення).