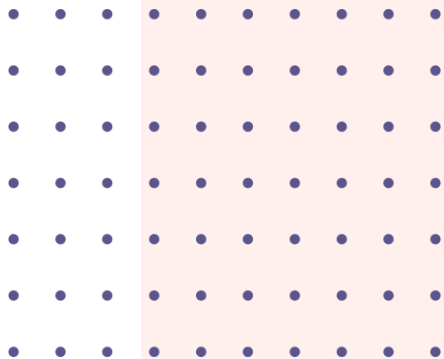


Урок 6. Рівні та види тестування

Структура заняття

1. Рівні тестування
2. Функціональні види тестування
3. Нефункціональні види тестування
4. Тестування, пов'язані зі змінами
5. Домашнє завдання



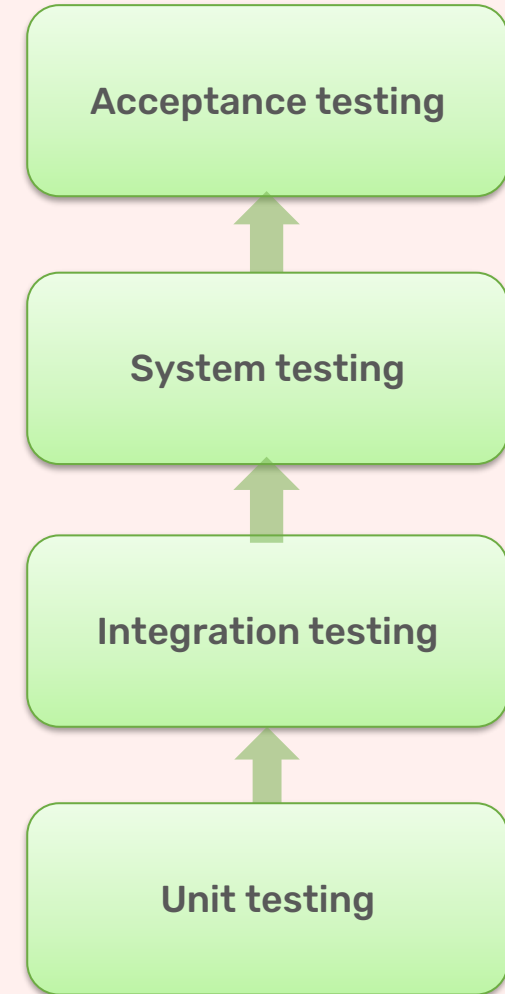
Рівні тестування

Приймальний тест - вид тестування, що проводиться на етапі здачі готового продукту (або готової частини продукту) замовнику. Метою приймального тестування є визначення готовності продукту, що досягається шляхом проходження тестових сценаріїв та випадків, які побудовані на основі специфікації вимог до ПЗ, що розробляється.

Системне тестування - це тестування ПЗ, що виконується на повній, інтегрованій системі, з метою перевірки відповідності системи вихідним вимогам, як функціональним, так і нефункціональним.

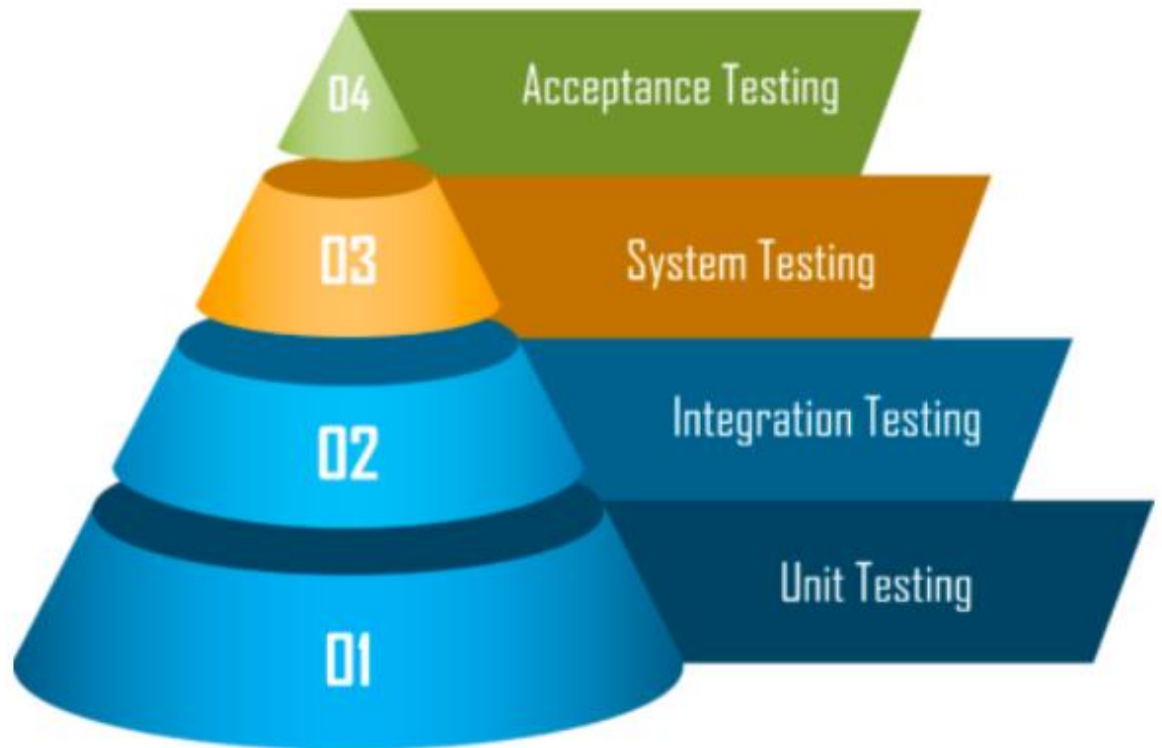
На етапі **інтеграційного тестування** кожен компонент або блок об'єднуються та тестуються разом. Ціль цього тестування - переконатися, що ПЗ або продукт працюють правильно, коли одиниця ПЗ об'єднана з іншими компонентами. Використовується виявлення помилок між взаємодіючими компонентами.

Модульне тестування - це процес тестування невеликих компонентів ПЗ чи продукту. Проводиться розробниками. Мета - протестувати кожен компонент ПЗ, щоб переконатися, чи воно нормально функціонує, чи воно працює за призначенням, чи відповідає воно зазначеним вимогам.



Піраміда тестування

Метафора, що означає групування динамічних тестів програмного забезпечення за різними рівнями. Вона також дає уявлення, яка кількість тестів має бути у кожній із цих груп. Основний принцип поділу рівнів - тест повинен бути на тому ж рівні, що і об'єкт, що тестується. У тесті вищого рівня ви не тестуєте всю умовну логіку та прикордонні випадки, які вже покриті тестами нижчого рівня.



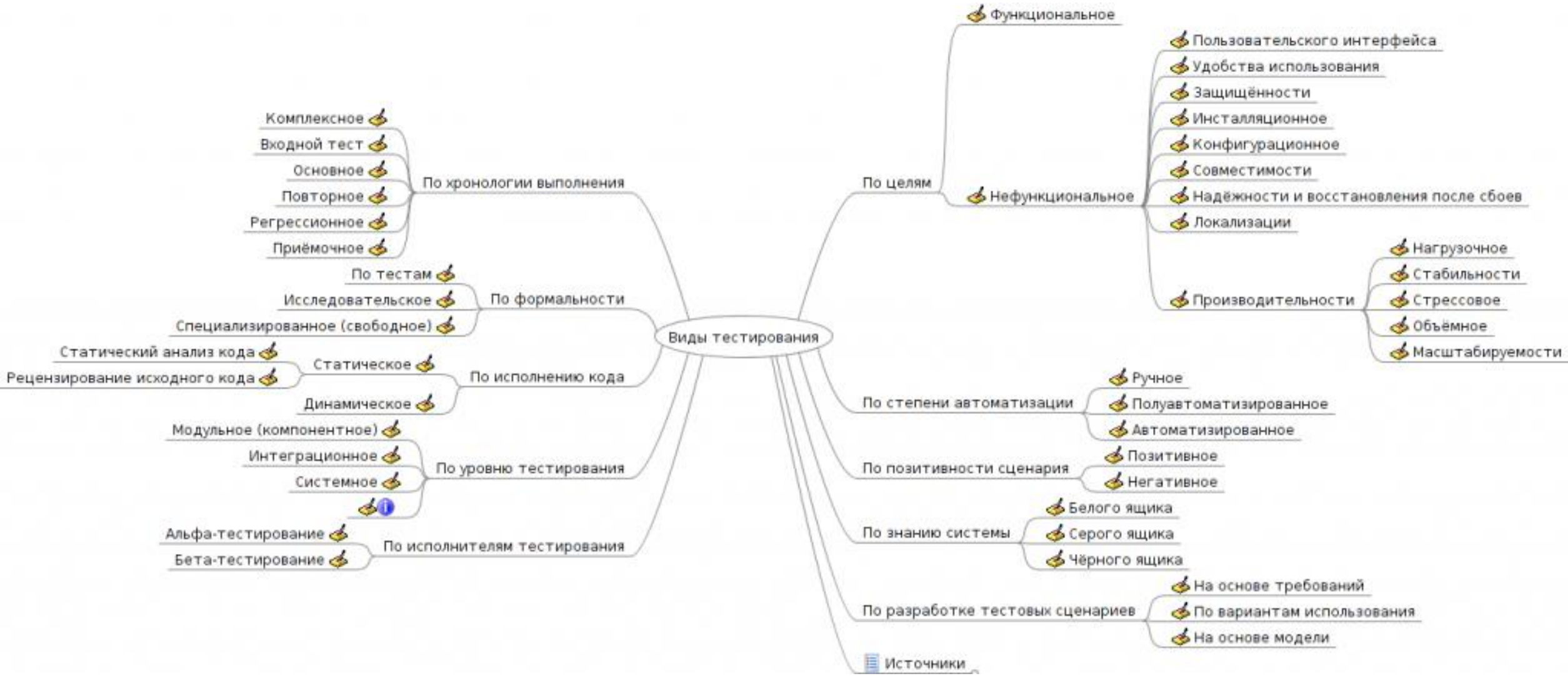
Піраміда тестування



Практична робота

- 4 групи (згідно з кількістю рівнів)
- обрати предмет (наприклад, комп'ютер чи автомобіль)
- кожна група пояснює, що саме буде тестувати і яким чином на своєму рівні тестування.







Функціональні види тестування

Функціональні тести базуються на функціях та особливостях, а також взаємодії з іншими системами, і можуть бути представлені на всіх рівнях тестування: компонентному або модульному (Component/Unit testing), інтеграційному (Integration testing), системному (System testing) та приймальному (Acceptance testing) .

Функціональні види тестування розглядають зовнішню поведінку системи.

Функціональні види тестування



Функціональне тестування (Functional testing)

розглядає заздалегідь зазначену поведінку і полягає в аналізі специфікації компонента чи системи загалом, тобто перевіряється коректність роботи функціональності програми.



Методика тестування на основі вимог (Requirement-based testing)

у цьому типі тестування вимоги пріоритизовані залежно від критеріїв ризику (що важливіше та більш критичне наразі) – і відповідно визначається пріоритет виконання тестів



Методика тестування на основі бізнес-процесів або сценаріїв використання (Business-process-based testing)

у цьому типі тестування описуються сценарії, пов'язані з повсякденним бізнес-використанням системи. На основі цих сценаріїв формуються тест-кейси.

Функціональні види тестування



Тестування користувацького інтерфейсу (GUI testing)

функціональна перевірка інтерфейсу на відповідність вимогам — розмір, шрифт, колір, послідовна поведінка.



Тестування безпеки (Security and Access Control Testing)

стратегія тестування, що використовується для перевірки безпеки системи, а також для аналізу ризиків, пов'язаних із забезпеченням цілісного підходу до захисту додатків, атак хакерів, вірусів, несанкціонованого доступу до конфіденційних даних.



Тестування взаємодії (Interoperability Testing)

перевіряє здатність програми взаємодіяти з одним і більше компонентами або системами і включає тестування сумісності (compatibility testing) і інтеграційне тестування (integration testing).



Нефункціональні види тестування

Описує тести, необхідні для визначення характеристик програмного забезпечення, які можна виміряти різними величинами. Загалом це тестування того, як система працює.



Нефункціональне тестування

Усі види тестування продуктивності:

- Навантажувальне тестування (Performance and Load Testing)
- Стресове тестування (Stress Testing)
- Тестування стабільності чи надійності (Stability / Reliability Testing)
- Об'ємне тестування (Volume Testing)

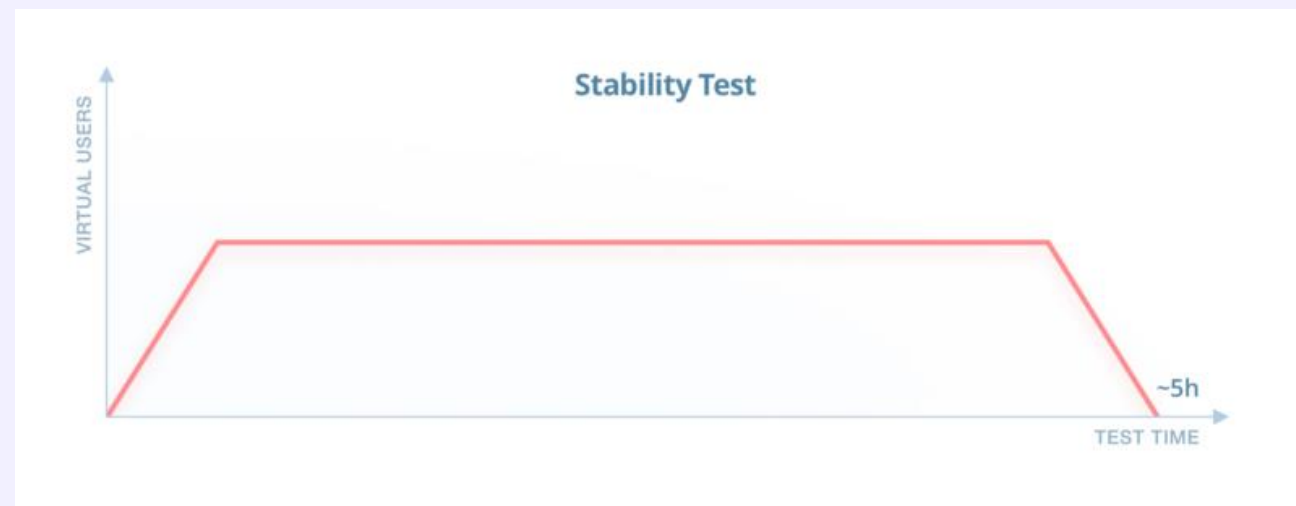
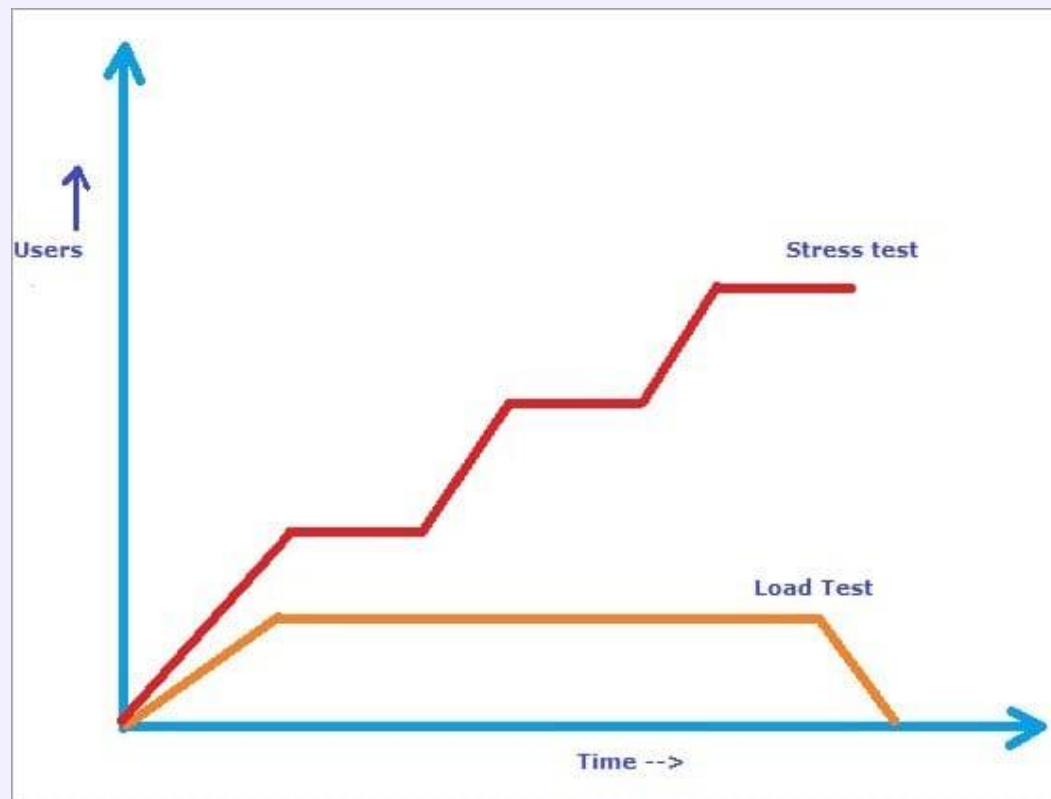
Тестування установки (Installation testing)

Тестування зручності користування (Usability Testing)

Тестування на відмову та відновлення (Failover and Recovery Testing)

Конфігураційне тестування (Configuration Testing)

Тестування безпеки (Security and Access Control Testing)





Тестування, пов'язане зі змінами

Після проведення необхідних змін, таких як виправлення багу/дефекту, ПЗ має бути перевірене для підтвердження того факту, що проблема була дійсно вирішена.

- Димове тестування (Smoke Testing)
- Регресійне тестування (Regression Testing)
- Тестування збірки (Build Verification Test)
- Санітарне тестування або перевірка узгодженості/справності (Sanity Testing)



Тестування збірки (Build Verification Test)

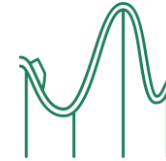
Тестування спрямоване на визначення відповідності випущеної версії, критеріям якості початку тестування. Вглиб воно може проникати далі, залежно від вимог до якості випущеної версії. За цілями є аналогом smoke testing



Димове тестування (Smoke Testing)

Поверхнєве тестування найбільш важливих модулів додатка на предмет можливості виконання необхідних завдань та наявності критичних і блокуючих дефектів, що швидко знаходять.

У разі відсутності таких дефектів – додаток передається на повний цикл тестування.



Санітарне тестування (Sanity Testing)

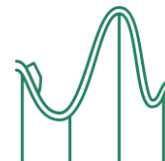
вужькоспрямоване тестування для доказу того, що конкретна функція працює відповідно до заявлених у специфікації вимог.

Є підмножиною регресійного тестування. Використовується для визначення працездатності певної частини програми після змін зроблених у ній чи навколишньому середовищі. Зазвичай виконується вручну.



Регресійне тестування (Regression Testing)

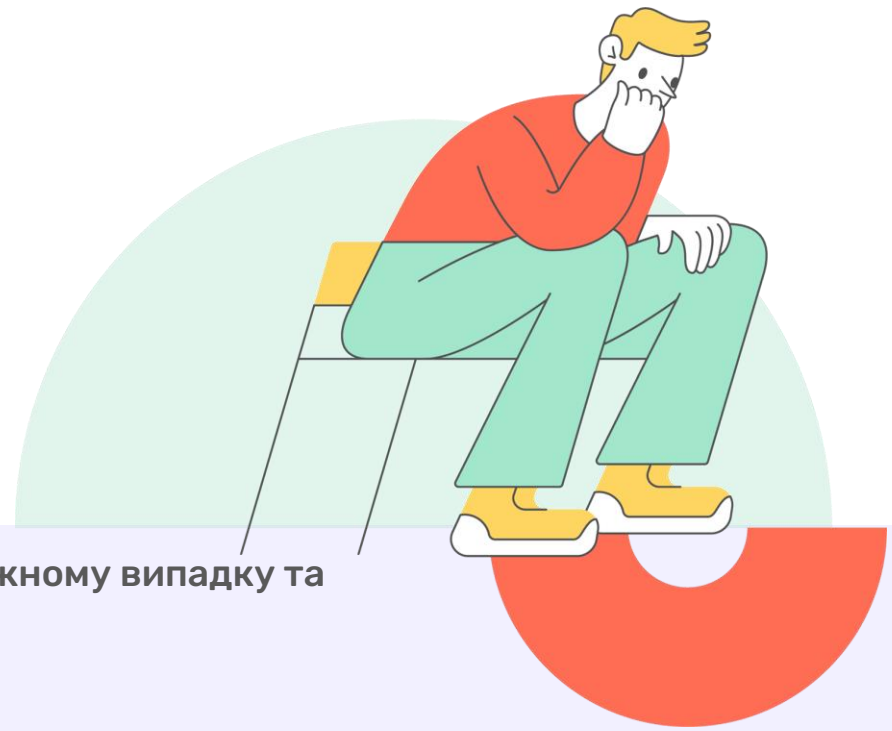
Тестування спрямоване на перевірку змін, зроблених у додатку або навколишньому середовищі (лагодження дефекту, злиття коду, міграція на іншу операційну систему, базу даних, веб-сервер або сервер програми), для підтвердження того факту, що існуюча раніше функціональність працює як і раніше.



Повторне тестування (Re-test)

тестування, яке перевіряє, що дефект виправлений та змінений функціонал працює як треба

Задача



Опишіть, які види тестування ви будете використовувати у кожному випадку та чому. У чому різниця між вибраними підходами?

Ви тестуєте продукт, для якого:

- Раз на два дні Вам надходить нова версія (з новим функціоналом)
- Щодня правиться один баг і оновлення потрапляють у Ваш продукт
- Приблизно раз на тиждень тім лід Вам каже: “Глянь-ка швидко, як зараз працює основний функціонал системи. Можна поверхово, маєш 10 хвилин”

