

Лабораторна робота № 7

Тема: Тестування ергономіки програмного забезпечення.

Мета: Ознайомитись з методами тестування ергономіки, порівняти різні методи між собою.

Теоретичні відомості

Частина 1 - Вступ. Види тестування програмного забезпечення

У залежності від переслідуваних цілей види тестування можна умовно розділити на наступні типи:

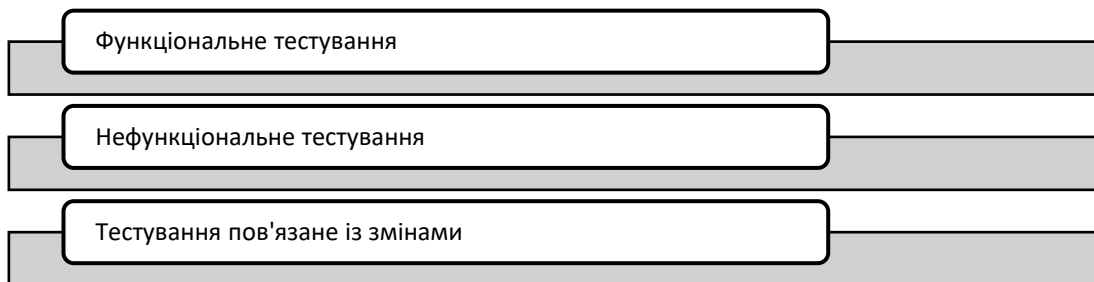


Рис. 1 Види тестування програмного забезпечення

Функціональне тестування – один з основних видів незалежного тестування програмного забезпечення, спрямований на перевірку реалізованості функціональних вимог. Інакше кажучи, фахівці з функціонального тестування визначають, чи вирішує розроблене ПЗ завдання, заради якого воно було створено, чи задовольняє воно потреби замовника / користувача.

Нефункціональне тестування описує тести, необхідні для визначення характеристик програмного забезпечення, які можуть бути виміряні різними величинами. У цілому, це тестування того, як система працює

Частина 2 - Функціональні тести

Функціональні тести базуються на функціях та особливостях, а також на взаємодії з іншими системами, і можуть бути представлені на всіх рівнях

тестування: компонентному або модульному (Component/Unit testing), інтеграційному (Integration testing), системному (System testing) і приймальному (Acceptance testing). Функціональні види тестування розглядають зовнішню поведінку системи.

Одні з найпоширеніших видів функціональних тестів:

- Функціональне тестування (Functional testing).
- 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 програми, атак хакерів, вірусів, несанкціонованого доступу до конфіденційних даних. Тестування безпеки може виконуватися як автоматизовано так і в ручну, включаючи перевірку як позитивних, так і негативних тестових випадків.

Частина 3 - Нефункціональне тестування

Основні види нефункціональних тестів:

- **тестування навантаження** (Performance and Load Testing) - визначення масштабованості додатків під навантаженням, при цьому відбувається: вимір часу виконання вибраних операцій за певних інтенсивностей виконання цих операцій; визначення кількості користувачів, що одночасно працюють з додатком; визначення меж прийнятної продуктивності при збільшенні навантаження (при збільшенні інтенсивності виконання цих операцій); дослідження продуктивності при високих, граничних, стресових навантаженнях;
- **стресове тестування** (Stress Testing) дозволяє перевірити наскільки додаток і система в цілому працездатні в умовах стресу і також оцінити здатність системи до регенерації, тобто до повернення до нормального стану після припинення впливу стресу. Стресом у даному контексті може бути підвищення інтенсивності виконання операцій до дуже високих значень або аварійна зміна конфігурації сервера. Також одним із завдань при стресовому тестуванні може бути оцінка деградації продуктивності, таким чином цілі стресового тестування можуть перетинатися з цілями тестування продуктивності;

- **тестування стабільності або надійності (Stability / Reliability Testing)** - перевірка працездатності програми при тривалому (багатодинному) тестуванні з середнім рівнем навантаження. Час виконання операцій може грати в даному виді тестування другорядну роль. При цьому на перше місце виходить відсутність витоків пам'яті, перезапущів серверів під навантаженням й інші аспекти, які впливають саме на стабільність роботи;
- **об'ємне тестування (Volume Testing)** - отримання оцінки продуктивності при збільшенні обсягів даних у базі даних програми, при цьому відбувається: вимір часу виконання вибраних операцій за певних інтенсивностей виконання цих операцій; може проводитися визначення кількості користувачів, що одночасно працюють з додатком;
- **тестування установки (Installation testing)** спрямоване на перевірку успішної інсталяції та настройки, а також оновлення або видалення програмного забезпечення. На даний момент найбільш поширена установка ПЗ за допомогою інсталяторів (спеціальних програм, які самі по собі так само потребують належного тестування). У реальних умовах інсталяторів може не бути. У цьому випадку доведеться самотійно виконувати установку програмного забезпечення, використовуючи документацію у вигляді інструкцій або readme файлів, де крок за кроком описано всі необхідні дії та перевірки;
- **тестування зручності користування (Usability Testing)** - це метод тестування, спрямований на встановлення ступеня зручності використання, навченості, зрозумілості та привабливості для користувачів розроблюваного продукту в контексті заданих умов.

Практичні завдання

Завдання 1

Ознайомтеся з видами тестування та їхньою короткою характеристикою поданою у **Частині 1**. Опишіть, які, на вашу думку, недоліки та переваги функціонально тестування?

Завдання 2

Розгляньте приклади подані у **Частині 2**. Наведіть свій приклад виду функціонального тестування. Коротко опишіть його.

Завдання 3

Прочитайте відомості подані у **Частині 3** про нефункціональне тестування. Проведіть пошукову роботу та опишіть два з запропонованих видів нефункціонального тестування:

1. Димове тестування (Smoke Testing).
2. Регресійне тестування (Regression Testing)
3. Тестування збірки (Build Verification Test)
4. Санітарне тестування або перевірка узгодженості / справності (Sanity Testing)
5. Конфігураційне тестування (Configuration Testing)

Завдання 4

Сформулюйте висновок до даної лабораторної роботи.

Лабораторна робота № 7

Приклад виконання

Завдання 1

Ознайомтеся з видами тестування та їхньою короткою характеристикою поданою у **Частині 1**. Опишіть, які, на вашу думку, недоліки та переваги функціонально тестування?

Переваги	Недоліки
імітує фактичне використання системи;	можливість упущення логічних помилок в програмному забезпеченні;
	ймовірність надмірного тестування;

Завдання 2

Розгляньте приклади подані у **Частині 2**. Наведіть свій приклад виду функціонального тестування. Коротко опишіть його.

Тестування взаємодії (Interoperability Testing) - це функціональне тестування, що перевіряє здатність програми взаємодіяти з одним і більше компонентами або системами і включає в себе тестування сумісності (compatibility testing) та інтеграційне тестування (integration testing).

Завдання 3

Прочитайте відомості подані у **Частині 3** про нефункціональне тестування. Проведіть пошукову роботу та опишіть два з запропонованих видів нефункціонального тестування:

1. Димове тестування (Smoke Testing).
2. Регресійне тестування (Regression Testing)
3. Тестування збірки (Build Verification Test)

4. Санітарне тестування або перевірка узгодженості / справності (Sanity Testing)
5. Конфігураційне тестування (Configuration Testing)

Опишемо Димове тестування (Smoke Testing) та Регресійне тестування (Regression Testing):

- *Димове тестування (Smoke Testing).*

Поняття димове тестування пішло з інженерного середовища. При введенні в експлуатацію нового “заліза” вважалося, що тестування пройшло вдало, якщо з установки не пішов дим. В області ж тестування програмного забезпечення, воно спрямоване на поверхневу перевірку всіх модулів програми на предмет працездатності та наявності швидко встановлюваних критичних і блокуючих дефектів. За результатами димового тестування робиться висновок про те, приймається чи ні встановлена версія програмного забезпечення на тестування, експлуатацію або на постачання замовнику.

- *Регресійне тестування (Regression Testing)*

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

Висновок: У даній лабораторній роботі ми ознайомились з видами тестування, детально розглянули функціональне та нефункціональне тестування, а також їх підвиди.