

Тестування програмного забезпечення

Викладач: к.т.н. Шитікова Олена Вікторівна

Лекція 2

ВИДИ ТЕСТУВАННЯ

Види тестування

Існуючі на сьогодні методи тестування програмного забезпечення не дозволяють однозначно та повністю виявити всі дефекти й встановити коректність функціонування аналізованої програми, тому всі існуючі методи тестування діють в рамках формального процесу перевірки програмного забезпечення.

Такий процес формальної перевірки, або верифікації, може довести, що дефекти відсутні з точки зору використовуваного методу.



Види тестування

Тобто немає ніякої можливості точно встановити чи гарантувати відсутність дефектів у програмному продукті з урахуванням людського чинника, присутнього на всіх етапах життєвого циклу ПЗ.

Існує безліч підходів до вирішення завдання тестування та верифікації програмного забезпечення, але ефективне тестування складних програмних продуктів – це процес надзвичайно творчий, що не зводиться до слідування строгим та чітким процедурам або створення таких.

Види тестування

Якість програмного забезпечення можна визначити як сукупну характеристику досліджуваного ПЗ з урахуванням наступних складових:

- надійність;
- супроводжуваність;
- практичність;
- ефективність;
- мобільність;
- функціональність.

Склад та зміст документації, супутньої процесу тестування, визначається стандартом IEEE Std 829-2008.

Класифікація видів тестування

✓ за об'єктом тестування:

- функціональне тестування;
- тестування продуктивності:
 - ❖ навантажувальне тестування;
 - ❖ стрес-тестування;
 - ❖ тестування стабільності;
- конфігураційне тестування;
- юзабіліті-тестування;
- тестування інтерфейсу користувача;
- тестування безпеки;
- тестування локалізації;
- тестування сумісності;

Класифікація видів тестування

✓ *за знанням системи:*

- тестування чорного ящика;
- тестування білого ящика;
- тестування сірого ящика;

✓ *за ступенем автоматизації:*

- ручне тестування;
- автоматизоване тестування;
- напівавтоматизоване тестування;

Класифікація видів тестування

✓ *за часом проведення тестування:*

- альфа-тестування:
 - ❖ димове тестування (smoke testing);
 - ❖ тестування нової функції (new feature testing);
 - ❖ підтверджуюче тестування;
 - ❖ регресійне тестування;
 - ❖ приймальне тестування;
- бета-тестування;

Класифікація видів тестування

✓ *за ступенем ізольованості компонентів:*

- модульне тестування;
- інтеграційне тестування;
- системне тестування;

✓ *за ознакою позитивності сценаріїв:*

- позитивне тестування;
- негативне тестування;

✓ *за ступенем підготовленості до тестування:*

- тестування за документацією (формальне тестування);
- інтуїтивне тестування (ad hoc testing).

Рівні тестування ПЗ

1. Компонентне (модульне) тестування (Unit Testing or Component Testing) перевіряє функціональність та шукає дефекти у частинах додатку, які доступні й можуть бути протестовані окремо (модулі програм, об'єкти, класи, функції та ін.).

Зазвичай компонентне (модульне) тестування проводиться шляхом виклику коду, який необхідно перевірити, і за підтримки середовищ розробки, таких як фреймворки для модульного тестування або інструменти для налагодження. Усі знайдені дефекти, як правило, виправляються в коді без формального їх опису в системі менеджменту багів (Bug Tracking System).

Рівні тестування ПЗ

Один з найбільш ефективних підходів до компонентного (модульного) тестування – це **підготовка автоматизованих тестів** до початку основного кодування (розробки) ПЗ.

Це називається **«розробка від тестування»** (test-driven development) або **«підхід тестування на початку»** (test first approach).

При цьому підході створюються та інтегруються невеликі частини коду, навпроти яких запускаються тести, написані до початку кодування. Розробка ведеться до тих пір, поки всі тести не будуть успішно пройдені.



Рівні тестування ПЗ

2. Інтеграційне тестування (*Integration Testing*)

призначене для перевірки зв'язку між компонентами, а також взаємодії з різними частинами системи (операційною системою, обладнанням або зв'язком між різними системами).

Рівні інтеграційного тестування:

✓ компонентний інтеграційний рівень (Component Integration Testing);

✓ системний інтеграційний рівень (System Integration Testing).

Рівні тестування ПЗ

Підходи до інтеграційного тестування:

✓ **знизу вгору** (Bottom Up Integration): усі низькорівневі модулі, процедури або функції збираються разом і потім тестуються. Після чого збирається наступний рівень модулів для проведення інтеграційного тестування.

Даний підхід вважається корисним, якщо всі або практично всі модулі розроблюваного рівня готові. Також даний підхід допомагає визначити за результатами тестування рівень готовності додатку;



Рівні тестування ПЗ

✓ **згори вниз** (Top Down Integration): спочатку тестуються всі високорівневі модулі й поступово, один за одним додаються низькорівневі. Усі модулі нижчого рівня симулюються заглушками з аналогічною функціональністю, потім у міру готовності вони замінюються реальними активними компонентами. Таким чином ми проводимо тестування згори вниз;

Рівні тестування ПЗ

✓ **великий вибух** («Big Bang» Integration): всі або практично всі розроблені модулі збираються разом у вигляді закінченої системи або її основної частини, а потім проводиться інтеграційне тестування.

Такий підхід застосовується для збереження часу. Однак, якщо тест-кейси та їх результати записані неправильно, то сам процес інтеграції дуже ускладниться, що стане перешкодою для команди тестування при досягненні основної мети інтеграційного тестування.



Рівні тестування ПЗ

3. Системне тестування (*System Testing*)

перевіряється інтегрована система на відповідність вимогам.

Основним завданням системного тестування є *перевірка як функціональних, так і не функціональних вимог у системі в цілому.*

При цьому виявляються дефекти, такі як неправильне використання ресурсів системи, непередбачені комбінації даних користувача рівня, несумісність з оточенням, непередбачені сценарії використання, відсутня або неправильна функціональність, незручність використання та ін.

Рівні тестування ПЗ

Підходи до системного тестування:

✓ на базі вимог (requirements based): для кожної вимоги пишуться тестові випадки (test cases), що перевіряють виконання даної вимоги;

✓ на базі випадків використання (use case based): на основі уявлення про способи використання продукту створюються випадки використання системи (Use Cases). За конкретним випадком використання можна визначити один або більше сценаріїв. Для перевірки кожного сценарію пишуться тест-кейси (Test Cases), які повинні бути протестовані.



Рівні тестування ПЗ

4. Приймальне тестування (*Acceptance Testing*)

– це формальний процес тестування, який перевіряє відповідність системи вимогам та **проводиться з метою:**

- визначення чи задовольняє система приймальні критерії;
- винесення рішення замовником або іншою уповноваженою особою чи приймається додаток.

Приймальне тестування виконується на підставі набору типових тестових випадків та сценаріїв, розроблених на підставі вимог до даного додатку.



Рівні тестування ПЗ

Рішення про проведення приймального тестування приймається у наступних випадках:

- продукт досягає необхідного рівня якості;
- замовник ознайомлений з планом приймальних робіт (product acceptance plan) або іншим документом, де описаний набір дій, пов'язаних з проведенням приймального тестування, дата проведення, відповідальні особи та ін.

Фаза приймального тестування триває доти, поки замовник не виносить рішення про відправлення додатку на доопрацювання або видачу додатку.

Рівні тестування ПЗ

5. Регресійне тестування

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

Регресійними можуть бути тести як функціональні, так і нефункціональні.



Рівні тестування ПЗ

Зазвичай для регресійного тестування використовуються тест-кейси, написані на ранніх стадіях розробки та тестування. Це дає гарантію того, що зміни у новій версії додатку не пошкодили вже існуючу функціональність.

Рекомендується робити автоматизацію регресійних тестів для прискорення подальшого процесу тестування та виявлення дефектів на ранніх стадіях розробки програмного забезпечення.



Рівні тестування ПЗ

Термін «Регресійне тестування», залежно від контексту використання, може мати різний зміст. Сем Канер описує три основних типи регресійного тестування:

- **регресія багів** (bug regression) – спроба довести, що виправлена помилка насправді не виправлена;
- **регресія старих багів** (old bugs regression) – спроба довести, що нещодавня зміна коду або даних зламала виправлення старих помилок, тобто старі баги знову стали відтворюватися;
- **регресія побічного ефекту** (side effect regression) – спроба довести, що нещодавня зміна коду або даних зламала інші частини додатку.