

Тестування, верифікація та валідація програмного забезпечення

Лекція 5

Тема: Автоматизація тестування

Питання лекції

1. Цілі та завдання автоматизації тестування. Рівні автоматизації тестування.
2. Оцінка доцільності автоматизації тестування. Життєвий цикл автоматизації тестування.
3. Інструменти та фреймворки для автоматизації тестування.

Цілі та завдання автоматизації тестування

Автоматизоване тестування програмного забезпечення — частина процесу тестування на етапі контролю якості в процесі розробки програмного забезпечення. Воно використовує програмні засоби для виконання тестів і перевірки результатів виконання, що допомагає скоротити час тестування і спростити його процес.

Мета:

- Зменшення часу і вартості регресійного тестування.
- Підвищення повторюваності й точності тестів.
- Можливість швидкого запуску великої кількості тестів.

Види автоматизації тестування:

- Автоматизація тестування коду (Code-driven testing)
- Автоматизація тестування графічного інтерфейсу користувача (Graphical user interface testing)
- Автоматизація тестування API



Цілі та завдання автоматизації тестування.

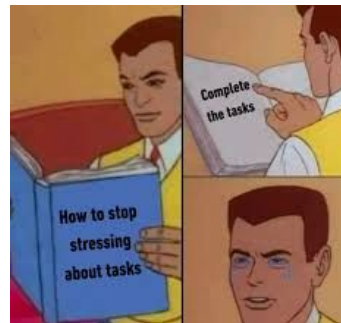
Цілі автоматизації

1. Скорочення часу на виконання тестів
2. Забезпечення стабільності та повторюваності перевірок
3. Розширення покриття тестами
4. Інтеграція в процес безперервної розробки (CI/CD)
5. Оптимізація витрат у довгостроковій перспективі



Завдання автоматизації

1. Автоматизація рутинних і часто повторюваних сценаріїв
2. Забезпечення швидкого зворотного зв'язку для розробників
3. Підтримка стабільної регресії
4. Моделювання навантаження і performance-тестів
5. Уніфікація тестових сценаріїв
6. Зменшення ручної рутини для тестувальників

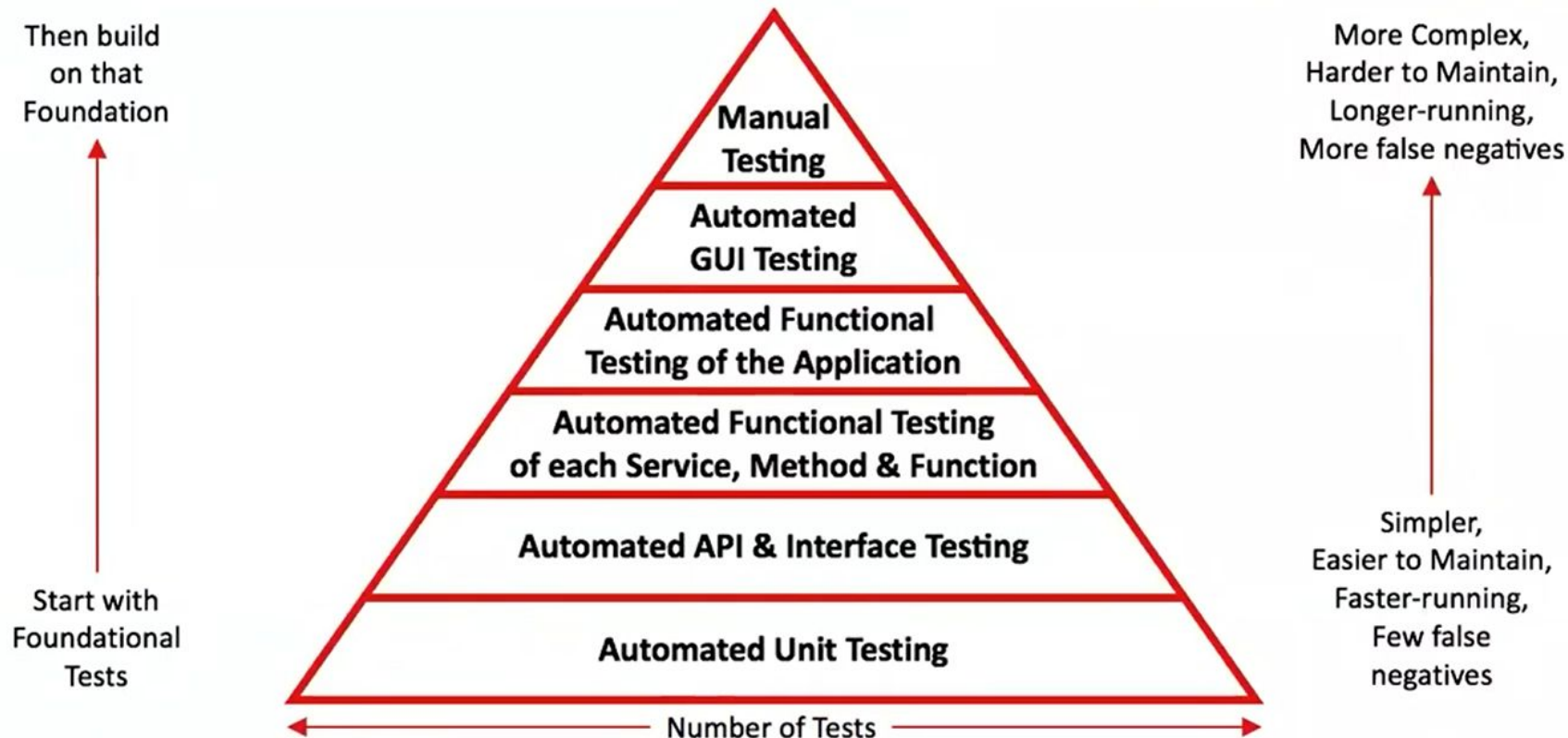


Цілі та завдання автоматизації тестування

Чому автоматизація важлива:

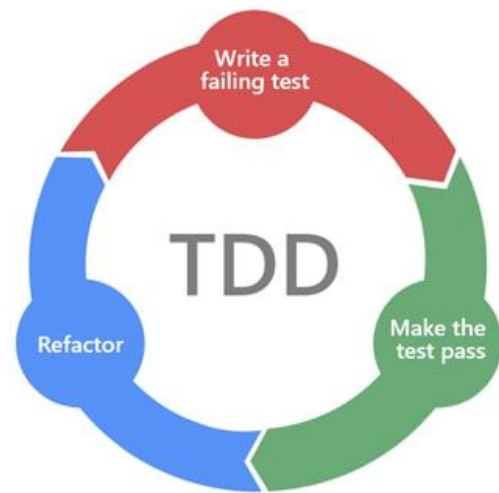
- Ручне тестування усіх робочих процесів, усіх полів, усіх негативних сценаріїв вимагає багато часу та грошей.
- Автоматизація не вимагає втручання людини. Ви можете запустити автоматичний тест без нагляду (наприклад вночі).
- Автоматизація збільшує швидкість виконання тесту.
- Автоматизація допомагає збільшити покриття тестами (Test Coverage).
- Ручне тестування може бути нудним а, отже, веде до випадкових помилок.
- Доволі складно протестувати мультимовні застосунки вручну.

Рівні автоматизації тестування



Рівні автоматизації тестування

Test Driven Development — це процес написання коду, який включає написання автоматизованого тест-кейсу на рівні **unit**-тестування, який не проходить, потім написання достатньої кількості коду для успішного проходження тесту, рефакторинг як тестового коду, так і продакшн-коду, а потім повторення з іншим новим тестовим випадку.



Рівні автоматизації тестування

Автоматизація тестування API — це процес використання програмного забезпечення для автоматичної перевірки програмних інтерфейсів додатків (API). Замість ручного тестування, створюються скрипти, які надсилають запити до API, отримують відповіді та порівнюють їх з очікуваними результатами. Це дозволяє перевірити функціональність, надійність, продуктивність та безпеку на рівні бізнес-логіки, ще до створення графічного інтерфейсу.

Ключові переваги автоматизації тестування API:

- Раннє виявлення дефектів
- Підвищення ефективності та швидкості
- Покращене тестове покриття
- Інтеграція з CI/CD
- Незалежність від мови та платформи

Основні аспекти, що перевіряються:

- Функціональність
- Безпека
- Продуктивність
- Надійність



Рівні автоматизації тестування

Автоматизація UI тестування — це процес використання спеціалізованих інструментів для автоматичної перевірки функціональностей шляхом імітації дій реального користувача з графічним інтерфейсом додатку (вебсайту, мобільного додатку). Програмні скрипти автоматично перевіряють, чи всі візуальні елементи, такі як кнопки, меню, форми та посилання, виглядають і функціонують коректно.



Рівні автоматизації тестування

Performance-тестування (або тестування продуктивності) — це процес визначення швидкості, надійності, масштабованості та стабільності системи під певним навантаженням. Автоматизація цього процесу дозволяє симулювати тисячі одночасних користувачів та збирати точні метрики, що неможливо зробити вручну.

Основна мета: знайти та усунути "вузькі місця" в системі, щоб забезпечити її стабільну роботу та позитивний користувацький досвід навіть при пікових навантаженнях.

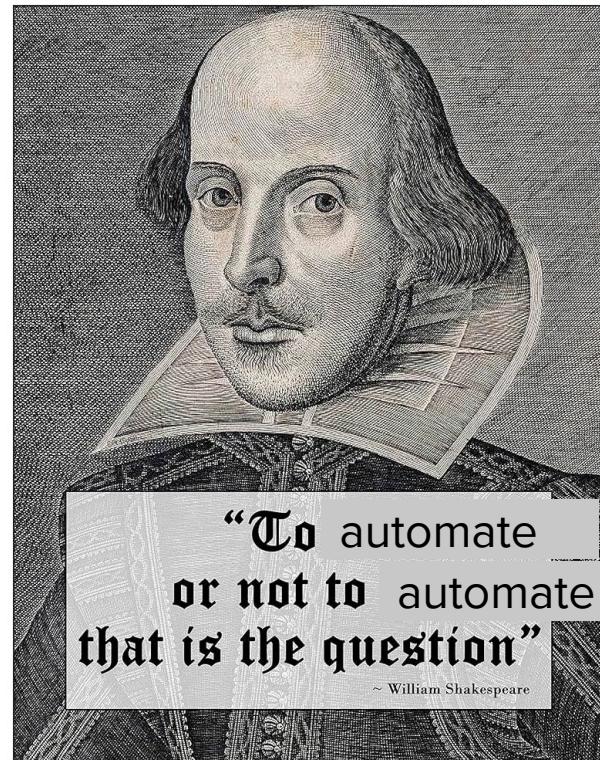
Основні види Performance-тестів:

- Навантажувальне тестування (Load Testing)
- Стрес-тестування (Stress Testing)
- Тестування на витривалість (Endurance/Soak Testing)
- Тестування масштабованості (Scalability Testing)

Оцінка доцільності автоматизації тестування

Визначення того, які тести підходять для ручного чи автоматизованого тестування, має вирішальне значення для розробки успішної стратегії забезпечення якості. Якщо під час роботи доводиться часто стикатися з такими типами тестів, слід розглянути можливість їхньої автоматизації:

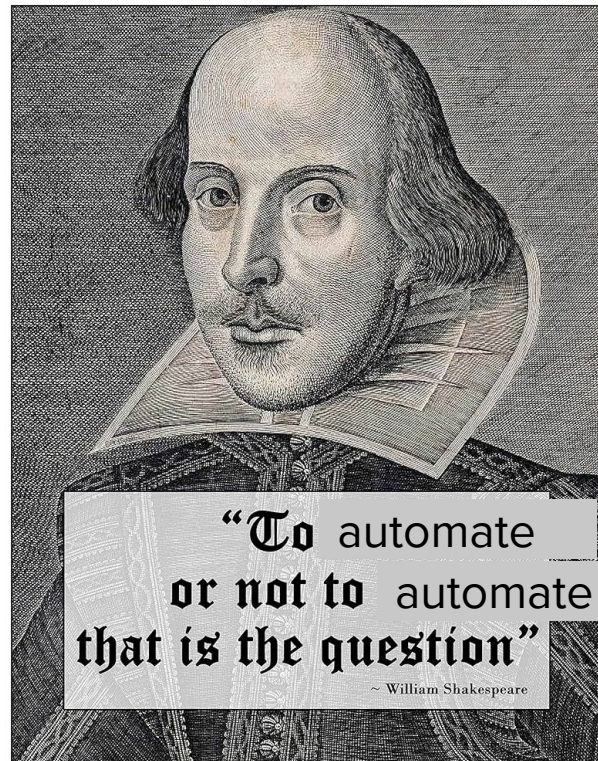
- Тестові випадки, що повторюються з часом.
- Тестові випадки, що потребують багато часу.
- Тестові випадки зі стабільними, незмінними вимогами.
- Тестові випадки, схильні до людських помилок.
- Тестові випадки зі значним простом між кроками.



Оцінка доцільності автоматизації тестування

Варто впроваджувати автоматизацію тестування коли:

- автоматизація тестів економічно вигідніша за ручне тестування
- тестування є повторюваним, наприклад, регресійне тестування
- тестування вручну починає займати все більше і більше часу
- виконання тест-кейсів стане більш точним
- тестування проводиться часто
- потрібно виконувати різні тести одночасно



ЖИТТЄВИЙ ЦИКЛ АВТОМАТИЗАЦІЇ ТЕСТУВАННЯ

Життєвий цикл автоматизації тестування — це процес, який складається з шести основних етапів для забезпечення ефективного та якісного тестування програмного забезпечення.

1. Визначення обсягу автоматизації (Feasibility Analysis)

Мета: Вирішити, які саме частини програми та які тестові сценарії доцільно автоматизувати.

2. Вибір інструментів для автоматизації (Tool Selection)

Мета: Підібрати правильні інструменти, які відповідають технологіям проекту, бюджету та навичкам команди.

3. Планування, дизайн та стратегія (Test Planning, Design, and Strategy)

Мета: Створити детальну "дорожню карту" для всього процесу тестування.

4. Налаштування тестового середовища (Test Environment Setup)

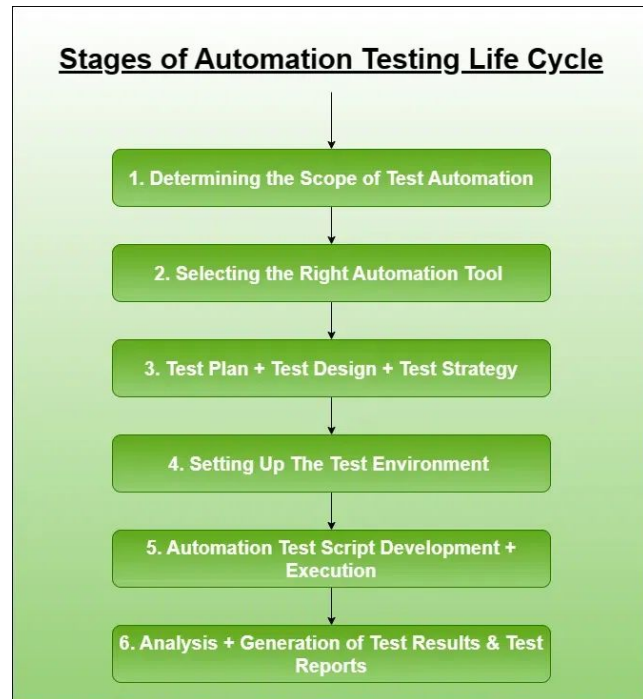
Мета: Підготувати стабільне та ізольоване середовище, що максимально наближене до реальних умов експлуатації продукту.

5. Розробка та виконання тестових скриптів (Script Development and Execution)

Мета: Написати код, який буде виконувати тестові сценарії, та запустити його.

6. Аналіз результатів та звітність (Analysis and Reporting)

Мета: Оцінити результати виконання тестів та надати звіт зацікавленим сторонам.



Інструменти та фреймворки для автоматизації тестування

Для **unit**: JUnit, TestNG, NUnit, xUnit, pytest.

Для **UI**: Selenium, Cypress, Playwright.

Для **API**: Postman/Newman, RestAssured, pytest + requests.

Для **performance**: JMeter, Gatling, Locust.

Для **CI/CD**: Jenkins, GitHub Actions, GitLab CI.

Інструменти та фреймворки для автоматизації тестування

xUnit.net — це безкоштовний інструмент з відкритим кодом для спільноти модульного тестування C#, F# та Visual Basic. xUnit.net версії 3 підтримує .NET 8.0 або пізнішої версії, а також .NET Framework 4.7.2 або пізнішої версії.

```
namespace SampleProject;

public class UnitTest1
{
    public static int Add(int x, int y) =>
        x + y;

    [Fact]
    public void Good() =>
        Assert.Equal(4, Add(2, 2));

    [Fact]
    public void Bad() =>
        Assert.Equal(5, Add(2, 2));
}
```



Інструменти та фреймворки для автоматизації тестування

Selenium – це набір інструментів з відкритим вихідним кодом для автоматизації роботи браузерів, що використовується переважно для тестування веб-додатків. Він підтримує різні мови програмування, такі як Java, Python, C# та інші, дозволяючи створювати автоматизовані тести для веб-сторінок та виконувати рутинні завдання в браузері.



Selenium WebDriver

If you want to create robust, browser-based regression automation suites and tests, scale and distribute scripts across many environments, then you want to use Selenium WebDriver, a collection of language specific bindings to drive a browser - the way it is meant to be driven.



Selenium IDE

If you want to create quick bug reproduction scripts, create scripts to aid in automation-aided exploratory testing, then you want to use Selenium IDE; a Chrome, Firefox and Edge add-on that will do simple record-and-playback of interactions with the browser.



Selenium Grid

If you want to scale by distributing and running tests on several machines and manage multiple environments from a central point, making it easy to run the tests against a vast combination of browsers/OS, then you want to use Selenium Grid.

Інструменти та фреймворки для автоматизації тестування

[Apache JMeter](#) — це програмне забезпечення з відкритим вихідним кодом, призначені для навантажувального тестування функціональної поведінки та вимірювання продуктивності. Спочатку вона була розроблена для тестування веб-застосунків, але з того часу розширилася до інших тестових функцій.

Apache JMeter можна використовувати для тестування продуктивності як статичних, так і динамічних ресурсів, а також динамічних веб-застосунків.

Його можна використовувати для імітації високого навантаження на сервер, групу серверів, мережу або об'єкт для перевірки його міцності або для аналізу загальної продуктивності за різних типів навантаження.

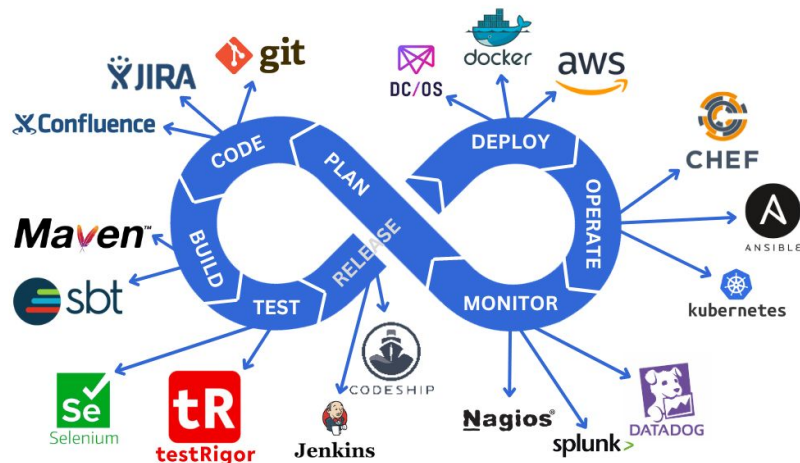


Інструменти та фреймворки для автоматизації тестування

CI/CD (Continuous Integration/Continuous Delivery або Continuous Deployment) — це набір практик у розробці програмного забезпечення, що автоматизують процеси злиття коду, його тестування та розгортання для прискорення та підвищення надійності випуску оновлень.

CI (безперервна інтеграція) означає часте об'єднання коду від різних розробників у спільний репозиторій з автоматичним запуском тестів для виявлення помилок.

CD (безперервна доставка або розгортання) завершує процес, автоматично готуючи код до релізу (доставка) або відправляючи його на продакшн (розгортання), що забезпечує швидше та якісніше постачання продукту користувачам.



Дякую за увагу!