Тестування та верифікація програмного забезпечення

Лекція №7

Тема: Забезпечення якості: спільне та відмінності процесів тестування, верифікації, валідації

ЖИТОМИРСЬКА ПОЛІТЕХНІКА
ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Лектор: асистент кафедри комп'ютерних наук Українець Микола Олександрович

Питання лекції

1. Тестування методами білої та чорної скрині.
2. Рівні тестування: модульний, інтеграційний, системний, валідаційний.
3. Розробка через тестування (Test-driven development).
4. Додаткові техніки верифікації та валідації: інспекція коду, перевірка на відповідність стандартам і вимогам, оцінювання зручності використання та користувацького досвіду, перевірка продуктивності та масштабованості.

Тестування методами білої та чорної скрині.

Тестування програмного забезпечення — процес перевірки відповідності заявлених до продукту вимог та реально реалізованої функціональності, що відбувається шляхом спостереження за його роботою в штучно створених ситуаціях та на обмеженому наборі тестів, вибраних визначеним чином.

Тестування vs. QA vs. QC

Тестування != Quality Assurance != Quality Control

Тестування vs. QA vs. QC

	Quality assurance	Quality control
Визначення	Сукупність дій, спрямованих на те, щоб переконатись, що якість дотримується на всіх етапах розробки	Сукупність дій, спрямованих на те, щоб переконатись, що кінцевий продукт відповідає поставленим вимогам
Мета	Мінімізація / недопущення появи дефектів на етапі розробки	Виявлення дефектів під час розробки і усунення їх до випуску продукту
Яке завдання вирішує?	Попередження дефектів шляхом покращення і дотримання процесів розробки	Пошук (і виправлення) дефектів у вже існуючому продукті
Як?	Запровадження системи управління якістю (QMS), а також періодичні аудити її ефективності	Пошук та виявлення джерела помилок для їх усунення, для подальшої відповідності узгодженим вимогам

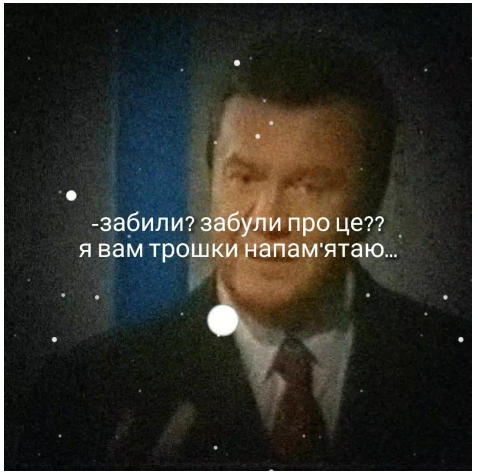
Тестування методами білої та чорної скрині.

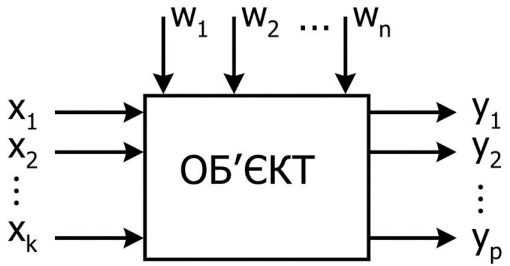
Black Box	White Box
базується на специфікаціях та формує тестові випадки від документації, не пов'язаної з внутрішньою структурою тестового об'єкта. Головною метою тестування чорної скриньки є перевірка поведінки системи відповідно до її специфікацій.	базується на структурі та формує тестові випадки на основі реалізації системи або внутрішньої структури (наприклад, коду, архітектури, робочих процесів та потоків даних). Головною метою тестування білої скриньки є покриття базової структури тестами до прийнятного рівня.
Black Box тестування — це підхід, при якому тестувальник перевіряє функціональність програмного забезпечення без знання його внутрішньої структури або коду. Основна увага приділяється вхідним і вихідним даним, відповідності системи вимогам. Типові методи: аналіз граничних значень, еквівалентне розбиття.	**White Box тестування —** це підхід, який передбачає повне розуміння внутрішньої логіки та структури коду. Тестування спрямоване на перевірку алгоритмів, умов, циклів і шляхів виконання. Типові методи: покриття операторів, гілок і шляхів.

Тестування методами білої та чорної скрині.

Чорний ящик (чорна скринька) — це термін, який використовується у техніці й кібернетиці для позначення об'єкта чи системи, про принципи дії яких нічого невідомо, крім того, що певному вхідному сигналу відповідає певний вихідний сигнал.

Чорний ящик є важливим елементом також у плануванні експерименту та в науці загалом. Використовуючи це поняття тисячі вчених світу отримують змогу використовувати відповідний рівень знань при дослідженні об'єкта, необхідний для досягнення оптимального результату планування.



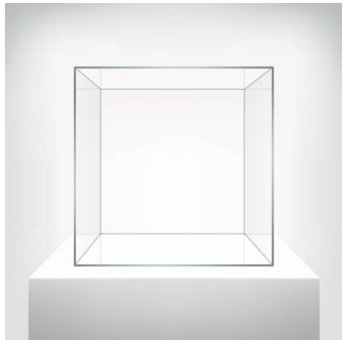


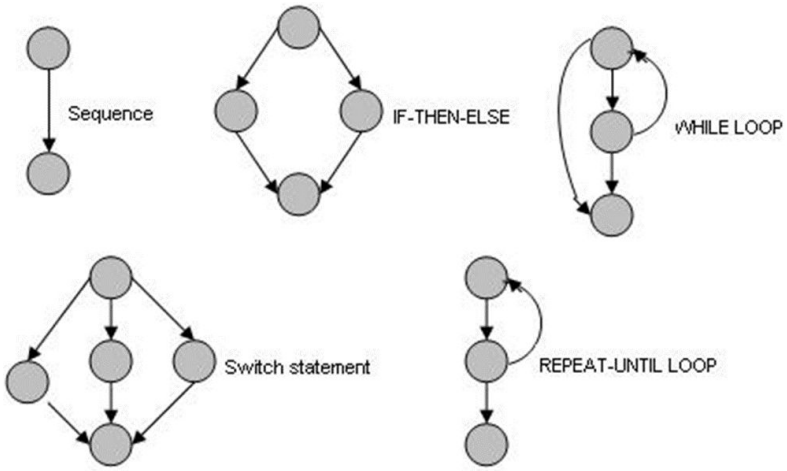
Тестування методами білої та чорної скрині.

Структурне тестування направлено на тестування структури системи або компонента. Цей вид тестування, як правило, відносять до тестування «білого» або «скляного», та «сірого» ящиків, оскільки ми перевіряємо, що відбувається всередині системи або додатка.

Методи структурного тестування:

- **Рядкове покриття (Statement Coverage)** – перевірка застосування усіх операторів в програмі на виконання (хоча б один раз).
- **Покриття рішення (Branch Coverage)** – перевіряє виконання всіх логічних розгалужень в програмі (кожна гілка if, else, switch, циклу повинна бути пройдена.);
- **Покриття умови (Condition Coverage)** – перевіряє виконання кожної частини складного логічного виразу з істинними та хибними значеннями (н-д: if (x > 0 && y > 0)).
- **Покриття шляху (Path Coverage)** – метод тестування, що перевіряє чи були пройдені усі можливі логічні шляхи виконання програми. Шлях — це послідовність рішень від початку програми до кінця.





Рівні тестування

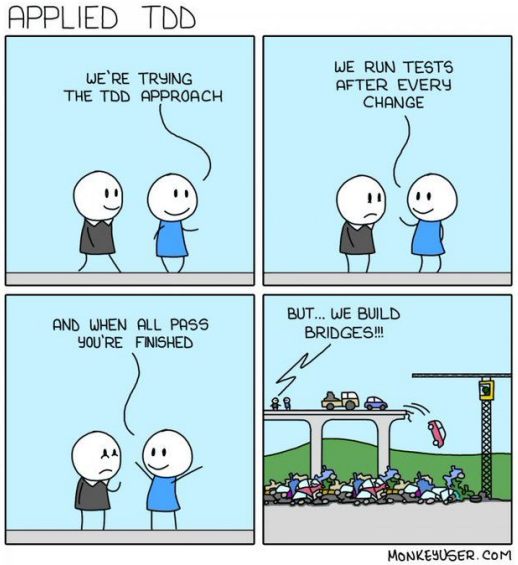
Згідно з силабусом ISTQB:

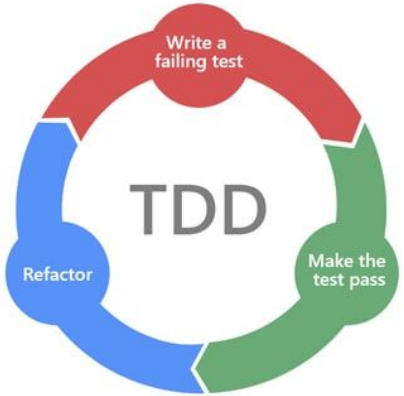
*Деякі ресурси виділяють в окремий рівень, деякі ні

Test-driven development

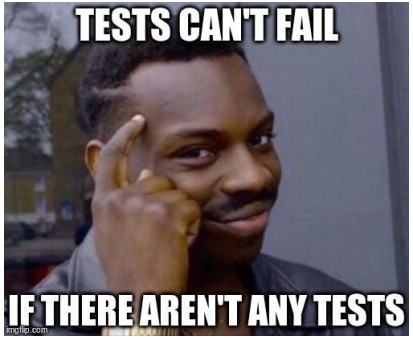
Test-Driven Development (TDD) — це підхід до розробки програмного забезпечення, при якому тести створюються до написання основного коду. Розробка ведеться через формулювання вимог у вигляді тестів, що визначають очікувану поведінку системи.

Основна ідея: код пишеться ***лише*** для того, щоб пройти тест.





Test-driven development



1. Формування сценаріїв
Визначити варіанти поведінки нової функціональності:
основний випадок;
граничні та помилкові ситуації;
2. Написання тесту
Створити автоматизований тест для одного сценарію, який описує очікуваний результат.
3. Запуск тестів (Red)
виконати всі тести;
новий тест має впасти як підтвердження необхідності реалізації.
4. Реалізація (Green)
написати найпростіший можливий код, щоб пройти тест;
допускається тимчасово неідеальне рішення (hardcode).
5. Перевірка
запустити всі тести і переконатися, що всі проходять; у разі помилок — виправити мінімальними змінами.
6. Рефакторинг (Blue)
Покращити код без зміни поведінки: усунути дублювання; покращити назви; спростити структуру; розбити складні методи. **Після кожної зміни запускати тести.**
7. Повторення циклу

- перейти до наступного сценарію;
- повторити кроки 2–6;
- працювати малими ітераціями.

Додаткові техніки верифікації та валідації

Згідно з [IEEE-STD-610]:

Верифікація - процес оцінювання програмного забезпечення для визначення того, чи відповідають продукти певного етапу розробки умовам, встановленим на початку цього етапу.

Валідація - процес оцінювання програмного забезпечення під час або в кінці процесу розробки, щоб визначити, чи відповідає воно заданим вимогам

Те саме, але ~~людською мовою~~ простіше:

Верифікація - перевірка того, чи ми **створюємо продукт правильно** відповідно до вимог, специфікацій тощо

Валідація - перевірка того, чи створюємо ми **“правильний продукт”** з точки зору користувача, чи він вирішує його проблеми, так як йому потрібно

Верифікація vs. Валідація

	Валідація	Верифікація
Визначення	Процес визначення чи правильно ми створюємо продукт	Процес визначення чи правильний продукт ми створюємо
Як?	Перевірка відповідності специфікаціям, вимогам, технічній документації	Перевірка відповідності очікуванням користувача та реальним потребам
Орієнтована на	процес і проміжні результати	кінцевий продукт
Приклади засобів	code review, unit-тести, статичний аналіз	beta-тестування, user acceptance testing

Додаткові техніки верифікації та валідації

Техніки верифікації

- **Інспекція коду**
  - Аналіз вихідного коду без виконання:
    - пошук логічних помилок;
    - перевірка структури та стилю;
    - відповідність стандартам написання коду.

- **Перевірка відповідності стандартам і вимогам**
  - Контроль відповідності системи:
    - технічним стандартам;
    - архітектурним рішенням;
    - специфікаціям і документації.

Додаткові техніки верифікації та валідації

Техніки валідації

- **Оцінювання зручності використання (Usability & UX)**
  - Перевірка взаємодії користувача із системою:
    - зрозумілість інтерфейсу;
    - зручність виконання задач;
    - задоволеність користувачів.
- **Перевірка продуктивності та масштабованості**
  - Оцінка роботи системи в реальних умовах:
    - швидкодія та стабільність;
    - поведінка під навантаженням;
    - здатність масштабуватись.

Дякую за увагу!