

СУЧАСНІ ПІДХОДИ ТЕСТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

Вінницький національний технічний університет

Анотація

У роботі було розглянуто та проаналізовано сучасні методи тестування програмного забезпечення). Перелічено їх переваги та недоліки. Проведено аналіз областей використання кожного з методів.

Ключові слова: тестування, модульне тестування, Ad-hoc тестування, інтеграційне тестування.

Abstract

The paper considers and analyzes modern methods of software testing (software). Their advantages and disadvantages are listed. The analysis of areas of use of each of the methods is carried out.

Keywords: testing, modular testing, Ad-hoc testing, integration testing.

Тестування є стало невідомою частиною процесу розробки програмних продуктів (SDLC) і є одним із найбільш живих способів забезпечення їх якості. Причому під якістю в сфері розробки інформаційних систем мається на увазі не тільки стабільність програми але і чи користування цим продуктом зручне і зрозуміле.

Тестуванням називається процес виявлення в програми помилок та ефективного їх виправлення [1]. Під помилками або дефектами програми, маються на увазі погано написаний код при розробці програмного продукту, наслідком яких стає порушення ланцюжка очікуваний результат при запуску і роботі розроблюваного програмного продукту. Тобто по суті завданням фази тестування є виявлення фактів неспівпадання реальної поведінки додатків до вимог, що ни повинні бути забезпечені [2].

Основні методи тестування програмного забезпечення (ПЗ):

- модульне тестування [2];
- тестування Ad-Нос [3];
- інтеграційне тестування.

Модульне тестування. Модульне тестування- це тестування класів шляхом розробки тестових драйверів для перевірки функціональності, що входять в клас методів. При тестуванні класів тестовий драйвер створює один або більше число примірників тестованого класу і здійснює прогін тестових випадків. У пропонованому навчальному курсі розглянуті особливості тестування класів, пов'язані з їх властивостями: інкапсуляція, поліморфізм і успадкування.

Інкапсуляція. Інформацію про стан класу можна отримати тільки за допомогою вбудованих в нього методів, які повертають значення властивостей класу. У даному випадку не можна тестувати метод ізольовано, а необхідно розглядати його як частина класу. Наприклад, уявімо ієрархію класів, в якій метод визначений для суперкласу і успадковується кількома підкласами. Кожен підклас використовує цей метод в контексті його приватних властивостей і методів.

Цей контекст змінюється, тому даний метод треба тестувати в контексті кожного підкласу. При виклику поліморфного методу важко визначити, яка реалізація буде перевірятися. Тому слід розглянути виклик методу, як в базовому, так і в похідних класах [4].

Тестування Ad-Нос. Тестування виконується без попередньої підготовки до тестування продукту, без визначення очікуваних результатів, проектування тестових сценаріїв тощо. Воно не вимагає ніякої документації, планування, процесів, яких, як правило, слід дотримуватися при виконанні тестування.

Головна перевага, Ad-Нос тестування часто надає можливість віднайти складні для відтворення і важковловимі дефекти, які неможливо було б знайти, використовуючи стандартні сценарії перевірок.

Проводячи Ad-Hoc тестування тестувальник покладається на своє загальне уявлення про продукт, порівнюючи його зі схожими продуктами, з власного досвіду. Однак, при Ad-Hoc тестуванні є зміст володіти загальною інформацією про продукт, особливо якщо проект дуже складний і великий.

Найчастіше Ad-Hoc тестування виконується коли власник не оперує конкретними цілями, потребами, вимогами до свого програмного продукту. Звісно у нього не має навіть натяку на щось подібне до проектною документації [2].

Даний метод застосовується на наступних рівнях тестування ПЗ:

- модульне тестування - для тестування процесів всередині модуля;
- інтеграційне тестування - для тестування зв'язків між модулями;
- системне тестування - для тестування зв'язків між підсистемами.

Переваги методу:

- тестування може бути розпочато на більш ранній стадії при відсутності графічний інтерфейс;
- тестування більш ретельне, з можливістю охоплення більшості функцій.

Недоліки методу:

- оскільки тести можуть бути дуже складними, потрібні висококваліфіковані тестувальники з глибокими знаннями в області програмування і реалізації;
- обслуговування тестового сценарію може бути витратним, якщо реалізація змінюється занадто часто;
- оскільки цей метод тестування тісно пов'язаний з тестованим додатком, інструменти для обслуговування кожного виду платформи можуть бути недоступні.

Як впливає з опису метод БС різко контрастує з методом ЧС [1, 4].

Інтеграційне тестування. Інтеграційне тестування - тестування за принципом «білого ящика» правильності взаємодії класів, що входять в різні модулі. Тестувальник цілісності повинен бути фахівцем, як в області розробки програмних продуктів, так і в області тестування. Взаємодія об'єктів являє собою запит одного об'єкта на виконання іншим об'єктом однієї з операцій одержувача і всіх видів обробки, необхідних для завершення цього запиту [2].

Можливі такі способи взаємодії класів:

- общедоступная операція має параметри об'єктного типу; -общедоступная операція повертає значення об'єктного типу;
- метод одного класу створює екземпляр іншого класу як частину своєї реалізації;
- метод одного класу посилається на глобальний примірник іншого класу.

Виділити набір методів, за допомогою яких здійснюється взаємодія класів необхідно, використовуючи діаграму класів, взаємодії і кооперації, розроблені на етапі проектування ПО.

Найбільш часто використовують два напрямки інтеграції об'єктно систем: тестування, засноване на потоках, і тестування, засноване на використанні [1]. Для першого напрямку об'єктом інтеграції є набір класів, що обслуговує одиничний введення даних в систему. Іншими словами, засоби обслуговування кожного потоку інтегрують і тестують окремо. Згідно другого - спочатку інтегрують і тестують незалежні класи.

Висновок

У даній роботі було розглянуто та проаналізовано найбільш актуальні методи тестування ПЗ. Детально проаналізовано як їх переваги так і недоліки. Проаналізовано сфери використання кожного з методів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. The Art of Software Testing / Glenford J. Myers, Revised and Updated by Tom Badgett, Todd M. Thomas, Corey Sandler. - 2nd ed. - Hoboken, New Jersey.: John Wiley & Sons, Inc., 2004 - 234 p.
2. Бейзер Б. Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем. – СПб.: Питер, 2004. – 318 с.: ил.
3. Степанченко И. В. Методы тестирования программного обеспечения: Учеб. пособие / ВолгГТУ, Волгоград, 2006. – 74 с.
4. Maydaniuk V. P. Increasing the Speed of Fractal Image Compression Using Two-Dimensional Approximating Transformations / V. P. Maydaniuk, I. R. Arseniuk, O. O. Lishchuk // Journal of Engineering Sciences. – Sumy : Sumy State University, 2019. – Volume 6, Issue 1. – P. E16 – E20.

Куш Ярослав Юрійович , аспірант групи 122-20а, кафедри комп'ютерних наук ВНТУ, Вінницький національний технічний університет, м. Вінниця, e-mail: kush1ki14b@gmail.com.

Іванчук Ярослав Володимирович – д-р техн. наук, доцент, професор кафедри комп'ютерних наук, Вінницький національний технічний університет, Вінниця, e-mail: ivanchuck@ukr.net.

Озеранський Володимир Сергійович, к.т.н., старший викладач кафедри комп'ютерних наук ВНТУ, Вінницький національний технічний університет, м. Вінниця, e-mail: ozeransky@urk.net.

Куш Ярослав Юрійович - postgraduate of group 122-20a of the Computer Sciences Chair, Vinnytsia National Technical University, Vinnytsia, e-mail: kush1ki14b@gmail.com.

Ivanchuk Yaroslav V. — Dr. Sc. (Eng.), Professor of the Computer Science Department, Vinnytsia National Technical University, Vinnytsia, e-mail: ivanchuck@ukr.net.

Ozeranskyj Volodymyr S. - lecturer of the Computer Sciences Chair, Vinnytsia National Technical University, Vinnytsia, e-mail: ozeransky@urk.net.