УДК 004.415.53

В.О. Волошина
Я.А. Кулик

ДОСЛІДЖЕННЯ СТРАТЕГІЙ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Вінницький національний технічний університет

Анотація

У роботі було розглянуто та досліджено стратегії тестування програмного забезпечення. Виявлено їх переваги та недоліки. Проаналізовано сфери використання кожної із стратегій.

Ключові слова: *тестування ПЗ, тестування чорної скриньки, тестування білої скриньки, функціональне тестування, структурне тестування.*

Abstract

The paper considers and investigates software testing strategies. Their advantages and disadvantages are revealed. The areas of use of each of the strategies are analyzed.

Keywords: *software testing, black box testing, white box testing, functional testing, structural testing.*

Підвищення якості програмного забезпечення є актуальним завданням розробки. Для її вирішення створено безліч методів та інструментів, застосування яких стало можливим завдяки потужності обчислювальних засобів, що постійно зростає. Сьогодні висока якість програмного забезпечення сприймається як обов'язковий компонент у сфері інформаційних технологій. Дуже важливо залучати засоби та методи контролю якості у процес планування та реалізації проектів із самого початку. В даний час реальні програмні продукти найчастіше розробляються в стислий термін та з обмеженим бюджетом. На жаль, за таких умов розробники програмного забезпечення часто ігнорують необхідність контролю та підтримки належної якості продукту, що розробляється, наражаючи цим кінцевих користувачів на невиправданий ризик.

Основним аспектом, що доводить необхідність застосування тестування разом із процесом розробки програмного забезпечення, є мінімізація витрат як розробника, так і споживача продукту. Такі витрати пов'язані з порушенням процесу розробки та застосування програмного продукту, викликаного необхідністю усунення знайдених у програмі помилок чи дефектів. Дефекти, виявлені і усунені на ранній стадії розробки, обходяться розробнику і клієнту набагато дешевше, ніж такі, що виявлені вже в період комерційного використання програмного продукту. Більше того, тестування дозволяє вести аналітичний збір інформації про вже вчинені в процесі розробки дефекти. Своєчасне забезпечення подібною інформацією розробників та керівників проектів істотно знижує ризик повторення дефектів, що в кінцевому результаті позитивно позначається на якості програмного продукту.

У цій роботі об'єктом дослідження є сучасні стратегії тестування, що застосовуються для розробки програмного забезпечення. Метою є аналіз існуючих методів тестування та їх систематизація.

Тестування програмного забезпечення є процесом пошуку помилок у реалізації програми. Очевидно, хотілося б організувати тестування таким чином, щоб виявити всі можливі помилки та дефекти у програмі. Для того щоб стверджувати, що програма повністю без дефектів необхідно підготувати всі можливі набори вхідних даних (включаючи некоректні), виконати програму на всіх можливих перестановках та варіаціях вхідних даних, проаналізувати всі вихідні дані та встановити, що кожен тестовий вихідний набір відповідає правильному.

Для реалізації ідеального та повного тестування потрібно задіювати великий обсяг ресурсів. Не важко зрозуміти, що навіть для невеликих програм кількість тестових випадків може обчислюватися значною кількістю комбінацій. Це доводить, що повне тестування програмного забезпечення, коли перевіряються всі можливі послідовності виконання програми, неможливе, відповідно неможливо виявити дефекти в реалізації програми. Інтуїтивний, побудований випадково

процес тестування не дасть належного результату. Тому тестування має базуватися на деякій підмножині всіляких тестових сценаріїв, які є вимушеною необхідністю. Процес тестування повинен виконуватися не спонтанно, а на підставі конкретної стратегії, якої слід дотримуватися під час тестування. Це свідчить про те, що дослідження сучасних методів тестування, дозволяють проводити його системно.

Перед розробкою стратегії тестування, необхідно зібрати якомога більше інформації про вимоги до програмного забезпечення і оцінити ризики. Стратегія тестування повинна враховувати безліч факторів та визначати, що і коли мають виконувати розробники та тестувальники. Мета стратегії тестування — мінімізація ризиків, беручи до уваги терміни, бюджет, ліміт ресурсів та інші нюанси розробки програмного забезпечення.

В даний час сформувалися дві протилежні одна одній парадигми тестування - функціональне (метод чорної скриньки) та структурне (метод білої скриньки).

Стратегія тестування методом «чорної скриньки» тривалий час залишалася основним способом тестування. У цьому методі тестування програмі подаються деякі дані на вхід і перевіряться результати, сподіваючись знайти невідповідності. При цьому те як працює програма вважається несуттєвим. Важливо відзначити, що за такого підходу обов'язково необхідно мати специфікацію програми для того, щоб було з чим порівнювати отримані в кінці тестування результати.

Вирізняють такі особливості методів функціонального тестування:

- Тестування системи загалом, включаючи окремі модулі та інтерфейси між ними.
- Тестування без знання вихідного коду, що дозволяє прискорити процес тестування та зробити його більш об'єктивним.
- Тести базуються на специфікації та не залежать від вихідного коду.
- Зручність автоматизації та регресійного тестування на базі тестів, побудованих на основі стратегії «чорної скриньки».
- Деякі ситуації не будуть перевірені внаслідок того, що тести покривають основну функціональність.

Метод тестування «чорної скриньки» досі є найпоширенішим у повсякденній практиці, але в нього є певний ряд недоліків, наприклад, неможливо знайти помилки, що взаємознищуються, і деякі помилки виникають досить рідко (наприклад помилки роботи з пам'яттю) і тому їх важко знайти і відтворити.

Стратегія структурного тестування передбачає створення тестів з урахуванням структури системи та її реалізації. Такий підхід іноді називають тестуванням методом «білої скриньки», щоб відрізняти його від тестування методом «чорної скриньки». Структурне тестування називають тестуванням шляхом покриття логіки.

Як правило, структурне тестування застосовується до відносно невеликих програмних елементів, наприклад до підпрограм або методів, асоційованих з об'єктами. За такого підходу аналізується програмний код і для отримання тестових даних використовуються знання структури компоненту. Набір тестових вхідних даних визначається виходячи із структури програми, наявності та взаємного розташування в коді певних конструкцій мови програмування – циклів, умовних операторів. Основна мета такого тестового набору вхідних даних – забезпечити необхідне покриття програмного коду. Основними методами структурного тестування є покриття операторів програми, гілок програми, умов.

Вирізняють такі особливості структурного тестування:

- Мінімальна вартість усунення помилки завдяки локалізації помилки всередині програмного модуля.
- Тести, побудовані на базі вихідного коду, забезпечать його повне покриття.
- Тести залежать від програмного коду і тестувальник змушений актуалізувати стан тестів за змінами програми.
- Спеціаліст з тестування повинен розбиратися в коді, що перевіряється.
- Складність тестування в цілому.

Висновок

В результаті дослідження сучасних методів тестування програмного забезпечення було встановлено, що не існує універсального засобу для позбавлення програмного забезпечення помилок. Кожен із розглянутих методів орієнтований на виявлення певних класів дефектів. Тому задля досягнення певної якості програмного продукту буде розумно застосовувати комбінацію розглянутих методів.

На етапі розробки найефективніше представляється стратегія структурного тестування. Тестування «білої скриньки» сприймається як частина програмування. Програмістові слід окремо тестувати кожен модуль програми після його написання, оскільки він краще за інших розбирається у власному коді.

На етапі інтеграції окремих модулів у єдину систему доцільно використати стратегію функціонального тестування. Таке тестування в жодному разі не повинно виконуватись програмістами, а спеціалістами з тестування. Це ефективно тому що вони не знають, як працює програма, вони тільки знають, як вона повинна працювати відповідно до специфікації і можуть повністю сконцентруватися на пошуку помилок невідповідності вимогам. Зрештою, тільки успішно пройшовши всі тести, програмний продукт може бути підданий тестуванню в реальних умовах.

У роботі було розглянуто основні стратегії, що застосовуються під час тестування програмного забезпечення. Встановлено, що вичерпне тестування неможливе через велике поєднання всіляких вхідних даних. Щоб досягти найкращої якості та виявити максимальну кількість помилок, тестування необхідно проводити, спираючись на існуючі методи. Для забезпечення високого рівня якості програмного забезпечення доцільно застосовувати комплексні підходи до тестування.

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 чорного 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.

Волошина Валерія Олегівна, студентка групи 1ІСТ-19б, кафедри автоматизації та інтелектуальних інформаційних технологій, Вінницький національний технічний університет, м. Вінниця, e-mail: 0972633723lera@gmail.com.

Кулик Ярослав Анатолійович, доцент кафедри автоматизації та інтелектуальних інформаційних технологій, Вінницький національний технічний університет, м. Вінниця, email: kulyk.y.a@vntu.edu.ua.

Voloshyna Valeria Olehivna, student of group 1IST-19b of the Automation and Intelligent Information Technologies Chair, Vinnytsia National Technical University, Vinnytsia, e-mail: 0972633723lera@gmail.com.

Kulyk Yaroslav Anatoliyovych, Associate Professor of the Automation and Intelligent Information Technologies Chair, Vinnytsia National Technical University, Vinnytsia, email: kulyk.y.a@vntu.edu.ua.