

Кондратюк Є.А.

*Студент 2 курсу центру післядипломної освіти
Науковий керівник - Вакалюк Т. А.,
кандидат пед. наук, доцент
Житомирський державний університет імені Івана Франка
м. Житомир*

МЕТОДИ ТЕСТУВАННЯ ПРОГРАМНИХ СИСТЕМ

Так вже склалось історично, що процес розробки програмного забезпечення (ПЗ) не може існувати без його тестування. Ще в 50-60-х роках минулого століття процес тестування став відокремлюватись від розробки ПЗ. Тоді народилась концепція виключного тестування (exhaustive testing) – перевірка усіх можливих шляхів виконання коду зі всіма можливими вхідними даними, однак вже згодом виявилось що це не можливо, так як кількість можливих шляхів і вхідних даних дуже велика.

Так існувало до середини 70-х років, коли народились дві фундаментальні ідеї тестування, яке спочатку розглядалось як процес доведення працездатності програми в деяких заданих умовах (positive testing), а потім навпаки, як процес доведення непрацездатності програми в деяких заданих умовах (negative testing). Ця внутрішня суперечність не тільки не зникла з часом, але й в подальшому зарекомендувала себе як дві взаємодоповнюючі цілі тестування.

З 80-х років тестування почали застосовувати на протязі всього циклу розробки ПЗ, а з 90-х відбувся перехід від тестування як такого до більш всеосяжний процесу, який називається «забезпечення якості (quality assurance)» [1].

Таким чином, тестування програмного забезпечення – це оцінка розроблюваного програмного забезпечення продукту, для перевірки його можливостей, здібностей і відповідність очікуваним результатам. На сьогоднішній день розрізняють такі методи в області тестування і забезпечення якості [7]:

1. Методом чорного ящика (black box testing) – без будь-яких знань внутрішньої роботи системи.
2. Методом білого ящика (white box testing) – враховуючи внутрішнє функціонування і логіку роботи коду.

3. Методом сірого ящика (grey box testing) – щось середнє між першим і другим, де тестер володіє лише загальними знаннями даного продукту, необхідними для виконання тесту [2].

Також є класифікація **за об'єктом тестування:**

- Функціональне тестування (functional testing)
- Тестування інтерфейсу користувача (UI testing)
- Тестування локалізації (localization testing)
- Тестування швидкості і надійності (load / stress / performance testing)
- Тестування безпеки (security testing)
- Тестування досвіду користувача (usability testing)
- Тестування сумісності (compatibility testing)

За суб'єктами тестування:

- Альфа-тестувальник (alpha tester)
- Бета-тестувальник (beta tester)

За часом проведення тестування:

- До передачі користувачу - альфа-тестування (alpha testing)
- Тест приймання (smoke test, sanity test або confidence test)
- Зміст тестування нових функціональностей (new feature testing)
- Регресивне тестування (regression testing)
- Тест здачі (acceptance of certification test)
- Після передачі користувачу - бета-тестування (beta testing) [2]

За критерієм сценаріїв:

- Позитивне тестування (positive testing)
- Негативне тестування (negative testing)

По ступеню ізольованості тестованих компонентів:

- Компонентне тестування (component testing)
- Інтеграційне тестування (integration testing)
- Системне (або енд-ту-енд) тестування (system or end-to-end testing)

За ступенем автоматизованості :

- Ручне тестування (manual testing)
- Автоматизоване тестування (automated testing)
- Змішане / напівавтоматизоване тестування (semi automated testing) [2], [8].

Методика тестування програмних систем може бути представлена у виді розвернутої спіралі (див. рис. 1) [6].



Рис. 1. Процесс тестування у виді еволюційної розгорнутої спіралі

Спочатку виконується тестування елементів(модулів), що перевіряє результати етапу кодування ПЗ. На другому кроці – тестуванні інтеграції, орієнтоване на виявлення помилок етапу проектування ПЗ. На третьому звороті спіралі проводиться тестування правильності, що перевіряє коректність етапу аналізу вимог до ПЗ. На заключному витку спіралі проводиться системне тестування, що виявляє дефекти етапу системного аналізу ПЗ.

Охарактеризуємо кожен крок процесу тестування:

1. Тестування елементів. Ціль – індивідуальна перевірка кожного модуля. Використовуються способи тестування «білого ящика».

2. Тестування інтеграції. Ціль – тестування зборки модулів у програмну систему. В основному використовують способи тестування «чорного ящика».

3. Тестування правильності. Ціль – перевірити реалізацію у програмній системі усіх функціональних та поведінкових вимог, а також вимоги ефективності. Використовуються виключно способи тестування «чорного ящика».

4. Системне тестування. Ціль – перевірка правильності об'єднання і взаємодії всіх елементів комп'ютерної системи, реалізації всіх системних функцій. [6]

Організація процесу тестування у виді еволюційної розгорнутої спіралі, забезпечує максимальну ефективність пошуку помилок.

Таким чином, кінцевою метою тестування програмних додатків є отримання якісного програмного продукту для інформаційної системи при максимізації результативності капіталовкладень в тестування (або максимізації кількості помилок, які виявляються одним тестом). [2] Для цього потрібно розглядати внутрішню структуру програми і робити деякі обґрунтовані припущення про наявність помилок і їх угрупованні в різних частинах або модулях програми. При цьому бажано кваліфікувати групи помилок: помилки, допущені досвідченими програмістами, помилки, допущені програмістами низької кваліфікації, або помилки, які являються наслідком слабо опрацьованих загальних ідей. Раннє виявлення таких груп помилок збільшує ефективність процесу тестування. наприклад, якщо в якій-небудь частині програми виявлено більше помилок, ніж в інших, то на

Її тестування повинні бути спрямовані додаткові зусилля психологічного, економічного або технічного характеру.

Список використаних джерел та літератури:

1. Куликов С. Тестирование программного обеспечения. Базовый курс. – ЕРАМ Systems. – 2015-2016. – С. 5-7.
2. Савін Р. Тестування або Посібник по жорсткому поводженню з багами в інтернет-стартапи. – Видавництво «Дело». – Москва, 2007. – С. 131-169.
3. Моделі менеджмента при розробці програмних продуктів [Електронний ресурс]. – Режим доступу до ресурсу: <http://staratel.com/iso/InfTech/DesignPO/index.html> . – Назва з екрану;
4. Автоматизоване тестування Веб- додатків [Електронний ресурс]. – Режим доступу до ресурсу: <http://www.developers.org.ua> . – Назва з екрану;
5. IDE-Метрики якості програмного забезпечення [Електронний ресурс]. – Режим доступу до ресурсу: <http://www.pmpofy.ru/content/rus/67/672-article.asp> . – Назва з екрану;
6. Методика тестирования программных систем [Електронний ресурс]. – Режим доступу до ресурсу: http://studopedia.ru/6_130074_metodika-testirovaniya-programmnih-sistem.html. – Назва з екрану;
7. Вакалюк Т. А. Підготовка майбутніх учителів інформатики до тестування програмного забезпечення / Т. А. Вакалюк // Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку: матеріали Всеукраїнської науково-практичної Internet-конференції. – Черкаси, 2013. – С. 275-277;
8. Вакалюк Т. А. Технології тестування програм : навчально-методичний посібник для студентів фізико-математичного факультету / Тетяна Анатоліївна Вакалюк. – Житомир : Вид-во ЖДУ, 2013. – 96 с.