

**Новикова К.В.**  
*студентка кафедри інформаційних-комп'ютерних  
технологій та фундаментальних дисциплін  
Київський національний університет технологій і дизайну*

**Люта М.В.**  
*старший викладач кафедри інформаційних-комп'ютерних  
технологій та фундаментальних дисциплін  
Київський національний університет технологій і дизайну*

**Розломій І.О.**  
*аспірант кафедри інформаційних технологій  
Черкаський національний університет ім. Б. Хмельницького*

## **АНАЛІЗ МЕТОДІВ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

Тестування програмного забезпечення дуже важливий процес, який є однією з основних частин при розробці програмного забезпечення (ПЗ). Будь-яка компанія, яка нехтує цим процесом, перш за все, ризикує зруйнувати репутацію своєї компанії і торгової марки.

В даний час програмні продукти нерідко розробляються в малі терміни і при обмежених бюджетах проектів. На жаль, в такому поспіху розробники часто ігнорують необхідність забезпечення належної якості своїх продуктів, ставлячи тим самим користувачів до невинуватого ризику.

Основним аспектом, який доводить актуальність застосування тестування в процесі розробки програмного забезпечення, є мінімізація непередбачених витрат як розробника, так і споживача продукту. Ці витрати пов'язані з порушенням процесу розробки і застосування програмного продукту, викликаного необхідністю усунення знайдених в програмі помилок – дефектів. Дефекти, виявлені і усунені вже на ранній стадії розробки, обходяться розробнику і клієнтові в десятки і навіть сотні разів дешевше, ніж такі, які розкрилися вже в період комерційного використання продукту.

У даній роботі об'єктом дослідження є сучасні стратегії тестування, що застосовуються при розробці програмного забезпечення. Метою дослідження є аналіз існуючих підходів до тестування програмних продуктів і їх систематизація. Для досягнення поставленої мети були сформульовані і вирішені наступні завдання:

1. Вивчити існуючі підходи до тестування програмного забезпечення.
2. Виявити переваги і недоліки існуючих підходів.
3. Запропонувати єдину схему використання різних методів тестування в технологічному циклі розробки програмного забезпечення.

В даний час сформувалися дві протилежні парадигми поведінкове (тестування методом «чорного ящика») і структурне тестування (тестування методом «білого ящика»). Також можна виділити відокремлену стратегію тестування в реальних умовах.

Тестування методом «чорного ящика», також відоме як тестування, засноване на специфікації або тестування поведінки – техніка тестування, заснована на роботі виключно з зовнішніми інтерфейсами тестованої системи.

До методів поведінкового тестування відносяться тестування потоку управління, тестування потоків даних, тестування доменів, синтаксичне тестування, тестування систем з кінцевим числом станів, тестування циклів.

Метою цієї техніки є пошук помилок в таких категоріях:

- 1) неправильно реалізовані або відсутні функції;
- 2) помилки інтерфейсу;
- 3) помилки в структурах даних або організації доступу до зовнішніх баз даних;
- 4) помилки поведінки або недостатня продуктивність системи.

Таким чином, ми не маємо уявлення про структуру та внутрішній устрій системи. Потрібно концентруватися на тому, що програма робить, а не на тому, як вона це робить

Переваги:

1) тестування проводиться з позиції кінцевого користувача і може допомогти виявити неточності і прогрірччя в специфікації;

2) тестувальнику немає необхідності знати мови програмування і заглиблюватися в особливості реалізації програми;

3) тестування може проводитися фахівцями, незалежними від відділу розробки, що допомагає уникнути упередженого ставлення;

4) можна починати писати тест-кейси, як тільки готова специфікація.

Недоліки:

1) тестується тільки дуже обмежена кількість шляхів виконання програми;

2) без чіткої специфікації досить важко скласти ефективні тест-кейси;

3) деякі тести можуть виявитися надмірними, якщо вони вже були проведені розробником на рівні модульного тестування.

Тестування методом «білого ящика» – метод тестування програмного забезпечення, який передбачає, що внутрішня структура / пристрій / реалізація системи відомі тестувальнику. Ми вибираємо вхідні значення, ґрунтуючись на знанні коду, який буде їх обробляти. Точно так само ми знаємо, яким повинен бути результат цієї обробки. Знання всіх особливостей програми, що тестується і її реалізації – обов'язкові для цієї техніки. Тестування «білого ящика» – поглиблення під внутрішній пристрій системи, за межі її зовнішніх інтерфейсів.

Основними методами структурного тестування є покриття операторів програми, покриття гілок програми, покриття умов.

Переваги:

1) тестування може проводитися на ранніх етапах: немає необхідності чекати створення призначеного для користувача інтерфейсу;

2) можна провести більш ретельне тестування, з покриттям великої кількості шляхів виконання програми;

3) допомагає в оптимізації коду.

Недоліки:

1) так як знання коду і внутрішньої структури – це необхідна умова, то для здійснення даного виду тестування необхідний досвідчений тестер, а це, в свою чергу, збільшує витрати на програмне забезпечення;

2) при використанні автоматизації тестування на цьому рівні, підтримка тестових скриптів може виявитися досить накладною, якщо програма часто змінюється;

3) майже неможливо заглянути в кожен шматок коду, щоб з'ясувати приховані помилки, які можуть створити проблеми, що призводять до збою програми.

Порівняння «чорного» та «білого» методів тестування

Критерій	Black-Box testing	White-Box testing
Рівні, до яких може бути застосована техніка	- системне тестування - приймальне тестування	- юніт-тестування - інтеграційне тестування
Виконавець	тестувальники	розробники
Знання програмування	не потрібно	необхідно
Знання реалізації	не потрібно	необхідно
Основа для тест-кейсів	специфікація, вимоги	проектна документація

В результаті аналізу сучасних методів тестування програмного забезпечення було встановлено, що не існує універсального засобу позбавлення від помилок програмного забезпечення. Кожен з розглянутих методів орієнтований на виявлення свого, певного класу дефектів. Тому для забезпечення прийнятної якості програмних продуктів доцільно застосовувати комбінацію розглянутих методів.

На етапі помодульної розробки найбільш ефективною представляється стратегія структурного тестування. Тестування «прозорого ящика» має розглядатися як частина процесу програмування. Саме програмісту слід окремо тестувати кожен модуль програми після його написання, оскільки він краще за інших знається на власному код і володіє засобами його налагодження.

На етапі інтеграції окремих модулів в єдину систему доцільно використовувати стратегію поведінкового тестування, що дозволяє охопити всю систему цілком. При цьому таке тестування має виконуватися не програмістами, а фахівцями з тестування. Це ефективно, тому що фахівці з тестування не мають відношення до написання вихідного коду програми. Вони не знають, як програма працює, вони лише знають те, як вона повинна працювати, і можуть повністю сконцентруватися на пошуку помилок

невідповідності вимогам. Крім того, такий підхід вирішує психологічну проблему, пов'язану з підсвідомою упевненістю програміста в правильності свого коду.

У роботі були розглянуті основні стратегії, що застосовуються при тестуванні сучасного програмного забезпечення. Щоб отримати найвищу якість і виявити максимальну кількість дефектів, тестування необхідно проводити, спираючись на існуючі методи.

В даний час ще не розроблені універсальні підходи в тестуванні програмного забезпечення. В ході дослідження було встановлено, що кожен з існуючих методів орієнтований на виявлення певного типу помилок. Для забезпечення прийнятної якості ПЗ доцільно застосовувати комплексні підходи до тестування, впроваджуючи роботи з тестування на різних етапах технологічного циклу розробки програмного забезпечення.

#### Література:

1. Тамре Л. Введение в тестирование программного обеспечения. – М.: Вильямс, 2003. – 359 с.
2. Бейзер Б. Тестирование «черного ящика». Технология функционального тестирования программного обеспечения. – СПб.: Питер, 2004. – 318 с.
3. Савин Р. Тестирование Dot Ком, или Пособие по жесткому обращению с багами в интернет-стартапах. – М.: Дело, 2007. – 312 с.
4. Василенко Н.В. Модели оценки надежности программного обеспечения / Н.В. Василенко, В.А. Макаров // Вестник НГУ. – 2004. – №28. – С. 126–132.