

УДК 004.415 (045)

Руда О. А. к.т.н.,
Моденов Ю.Б.

МЕТОДИ ТА МОДЕЛІ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Національний авіаційний університет

У роботі описані основні результати досліджень процесу тестування програмного забезпечення. Зазначено роль тестування, наведено основні методи та моделі тестування ПЗ. Визначено основні етапи процесу тестування

Вступ

Швидкий розвиток програмного забезпечення (ПЗ), комп'ютерної техніки, створення нових та вдосконалення існуючих технологій побудови програмних продуктів, розширення спектра використання автоматизованих систем у сучасному світі зумовлюють паралельний розвиток усіх складових процесу побудови та впровадження ПЗ. Підвищення складності та багатокomпонентності сучасних програмних проектів вимагають спеціалізованого підходу під час створення та застосування. Сьогодні техніка, автоматизовані системи, ПЗ повинні досягати високого рівня надійності, що дозволяє їм ставати ефективним інструментом у світі нових технологій та цілей. Будь-яка неполадка може мати серйозні наслідки та втрати, що неприпустимо у сучасному світі конкуренції. І саме тестування є одним із методів забезпечення якості продукту.

Тестування - це можливий спосіб оцінки якості програмного забезпечення в термінах знайдених дефектів, як для функціональних вимог, так і для нефункціональних вимог і характеристик програмного забезпечення (наприклад, надійність, практичність, ефективність, супроводжуваність і переносимість)[5].

Загальний погляд на тестування програмного забезпечення останні роки активно еволюціонував, стаючи все більш конструктивним, прагматичним і наближеним до реалій сучасних проектів розробки програмних систем (ПС).

Постановка проблеми

З огляду на неможливість вичерпного тестування великих програмних систем, а також обмеження щодо часу та вартості тестування, перед розробниками завжди поставали проблеми впорядкування дій з тестування у вигляді зв'язаного процесу задля раціонального розподілу ресурсів, а також прийняття обґрунтованих рішень щодо початку та завершення тестування програмної системи.

Виділення задач тестування, що вирішуються у рамках різних процесів ЖЦ, в єдиний процес дозволяє завчасно створити середовище і визначити ресурси тестування, встановити обсяги і терміни тестування в рамках плану проекту ПС, а також визначити завдання інших процесів, які необхідні для ефективного виконання процесу тестування і забезпечення якості ПС.

Процес включає наступні кроки: створення групи тестування; аналіз ризику; визначення цілей тестування; розробка плану тестування; розробка тестів; автономне та інтеграційне тестування; тестування ПЗ; системне тестування; аналіз результатів тестування; регресійне тестування.

Отже, поява окремого опису процесу тестування є дуже важливим кроком, насамперед, тим, що в стандарті ISO / IEC 12207 задачі тестування «розмиті» по багатьом процесам ЖЦ, що не дозволяє чітко визначити обов'язки групи тестування на всіх стадіях розробки ПС та визначити трудомісткість підготовки та проведення тестування на всіх рівнях.

Аналіз останніх досліджень та публікацій

Перші програмні системи розроблялися в рамках програм наукових досліджень або програм для потреб міністерств оборони. Тестування таких продуктів проводилося строго формалізовано із записом всіх тестових процедур, тестових даних, отриманих результатів [1]. Тестування виділялося в окремий процес, який починався після завершення кодування, але при цьому, як правило, виконувалося тим же персоналом. Еволюція поглядів на тестування програмного забезпечення зображена в таблиці 1.

Новий погляд на тестування знайшов відображення у V-подібній моделі (V-моделі) тестування, яка систематизує всі види пов'язаних з ним дій. V- модель упорядковує два послідовні підпроцеси тестування: підготовку до його здійснення на всіх рівнях випробувань (ліва гілка V-моделі) і власне здійснення тестування при випробуваннях всіх рівнів (права гілка V-моделі).

Кроки з підготовки до тестування розподіляються за етапами розроблення програмної системи, що передують випробуванням. На кожному кроці підготовки здійснюється аналіз артефактів відповідного етапу розробки для означення цілей, об'єктів, сценаріїв і ресурсів тестування, адекватних рівню випробувань. Такий аналіз дає могутній побічний ефект. Це з одного боку - поліпшення якості аналізованих напрацювань через усунення їх некоректності, двозначності, несумісності і. т.п. (що фактично є задачами верифікації ПС), а з іншого - уточнення множини функцій, вимог і сценаріїв функціонування ПС при моделюванні тестових ситуацій і оптимізації критеріїв завершення тестування (за тривалістю, вартістю, надійністю тощо).

Таблиця 1. Еволюція поглядів на тестування

Період	Опис
1960-ті	Увага приділялося «вичерпному» тестуванню, яке повинно проводитися з використанням усіх шляхів у коді або всіх можливих вхідних даних.
1970-ті	Тестування ПО позначалося як

	«процес, спрямований на демонстрацію коректності продукту» або як «діяльність по підтвердженню правильності роботи ПЗ».
1980-ті	Тестування розширилося таким поняттям, як попередження дефектів.
1990-ті	В поняття «тестування» стали включати планування, проектування, створення, підтримку і виконання тестів і тестових оточень
2000-ні	В поняття тестування було додано поняття «оптимізація бізнес-технологій»

З поширенням ітераційних моделей ЖЦ відбулося зміщення виконання тестування на ранні стадії ЖЦ, оскільки ітераційні моделі з їх періодичними випусками працездатних версій, дозволяють розпочати тестування вже за готовності до тестування першої версії. Тестування кожної наступної версії охоплює також попередні, що забезпечує більш ретельне тестування всієї системи[6].

Крім безпосередньої оцінки поточного рівня ефективності того чи іншого процесу є й більш цікава задача - підвищення рівня ефективності або, як прийнято говорити, рівня зрілості процесів. Коли базові проблеми планування і проведення робіт по тестуванню в інтеграції з процесом розробки програмного забезпечення вирішені, виникає задача пошуку оптимальних організаційних і процедурних схем виконання робіт. Відповівши на питання «хто» і «коли», доводиться шукати відповіді на питання «як, яким чином», «як вимірювати результати», «як контролювати», «як працювати ефективніше», а також «як управляти і розвивати процес, базуючись на отриманих даних і досвіді». Виділяють шість моделей зрілості процесів тестування:

- Testability Maturity Model (TMM);
- Software Testing Maturity Model (TMMSW);
- Test Process Improvement (TPI);
- Test Organization Maturity (TOM);
- Testing Assessment Program (TAM);
- Proposed Evaluation and Test SW-CMM Key Process Areas (SW-CMM KPA).

Основні результати досліджень

1. Методи тестування програмного забезпечення.

Достатня кількість підходів і методів тестування ускладнює їх однозначну класифікацію.

Традиційна класифікація заснована на поділі методів тестування на дві категорії - «чорний ящик» (функціональне) і «білий» (структурний) - і враховує тільки два підходи до проектування тестів.

«Білий ящик» (тестування білого ящика) це процес тестування ПЗ з можливим доступом до коду програми, що тестується, а «чорний ящик» дозволяє проводити тестування тільки через загальнодоступний користувацький інтерфейс програми.

Техніка чорного ящика дозволяє тестувати ПЗ через інтерфейс програми, доступний звичайному користувачеві. Модуль для тесту може емулювати натискання клавіш або кліки мишкою в тестованому ПЗ і дозволяє спостерігати, чи все проходить правильно і чи збігаються ці маніпуляції з натисканням клавіш і натискань мишки в реальності.

Є ще й третій варіант, іменованний «сірим ящиком». При цьому методі тестувальник так само як і в тестуванні білого ящика, має доступ до коду програми, але в процесі тесту цього доступу не потребує [2].

Кожний метод надає відповідний критерій покриття набором тестів вхідного простору. Вибір найбільш ефективних методів тестування за певних умов та на різних рівнях тестування є складною проблемою та пов'язаний з аналізом ризиків відмов програмної системи. Оскільки застосовність методів є однією з задач, ви-

рішуваних при побудові базового процесу тестування, проаналізовано та систематизовано основні методи, які враховують специфіку типів програмних систем[3].

Більш детальна класифікація методів тестування, заснована на підходах до проектування тестів зображена на рис. 1.

2. Процес тестування.

Модель процесу тестування зображена на рис.2.

На кожному кроці підготовки здійснюється аналіз робочих продуктів відповідних процесу розробки (вхідних для даного кроку процесу тестування) для визначення цілей, об'єктів, сценаріїв і ресурсів тестування, адекватних рівню тестування.

Результати виконання кроків підготовки тестування фіксуються в планах тестування. На кожному кроці виконання здійснюється фіксація результатів виконання тестів і їх порівняння з очікуваними результатами.

Результати тестування на кожному рівні аналізуються для того, щоб визначити поточний стан ПС і прийняти рішення про достатність тестування на даному рівні.

Кожен крок процесу складається з набору вирішуваних завдань.

Базовий процес враховує: розподіл обов'язків між учасниками процесу, вимоги до професійної підготовки виконавців процесу, стандарти для представлення документів, метрики процесу, застосовні методи для вирішення задач тестування, критерії початку та завершення задач і переходу до наступного кроку процесу. Для документування процесу тестування розроблено шаблони документів, пов'язані з кожним кроком процесу[6].

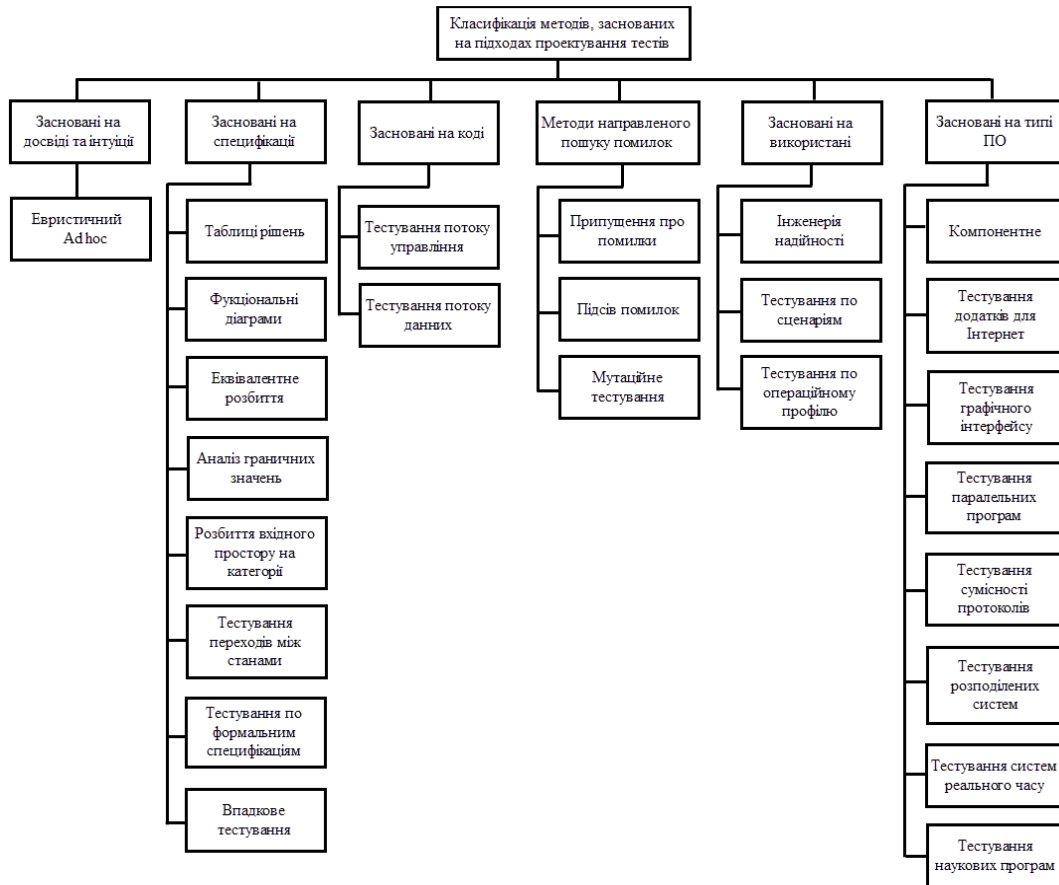


Рис.1. Класифікація методів, заснованих на підходах проектування тестів

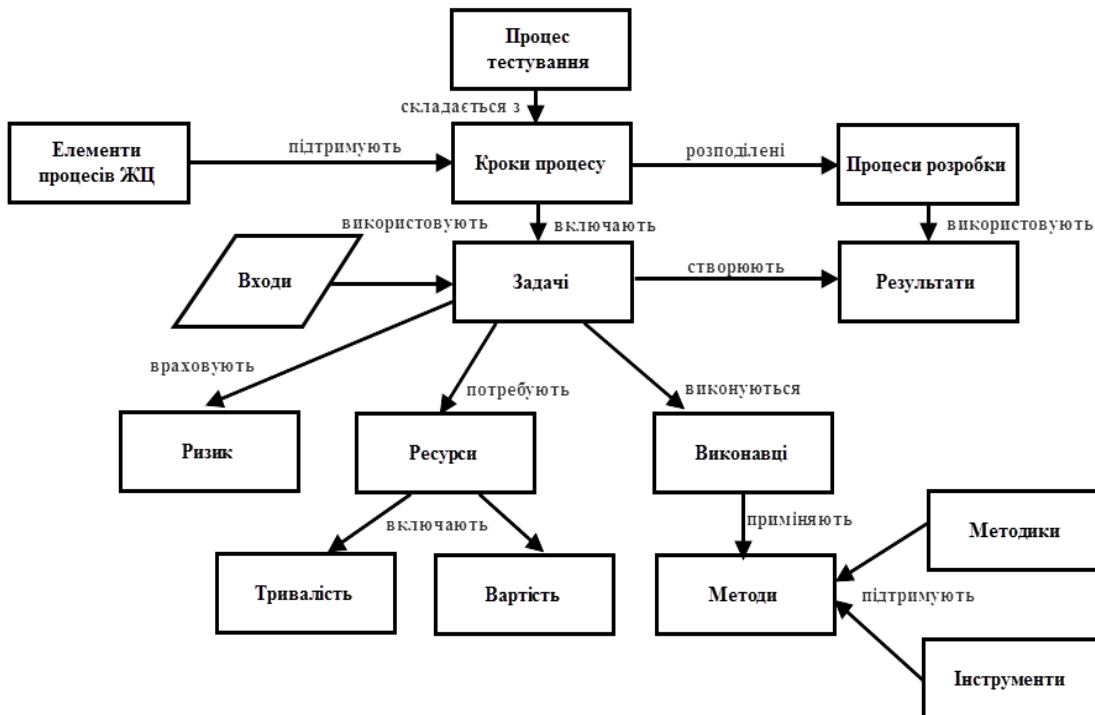


Рис. 2. Модель процесу тестування

3. Методи вдосконалення процесу тестування.

Одним з напрямів досліджень інженерії тестування, виконуваних з метою визначення складу задач базового процесу тестування та шляхів його вдосконалення, є аналіз та розробка відповідних моделей оцінювання процесу тестування. ТММ - модель зрілості тестування (Test Maturity Model). П'ятиступінчаста структура поліпшення процесу тестування, пов'язана з моделлю зрілості процесів програмного забезпечення (СММ / СММІ) і описує ключові елементи ефективного процесу тестування.

Рівень 1 - хаотичний. Процес тестування програмного забезпечення має хаотичний характер, що відрізняє більшість початківців компаній. Процес тестування не визначений як виділена активність і не відділений від процесу налагодження коду. Тестування виконується за фактом створення коду і побудови або складання системи.

Рівень 2 - фаза розробки. Тестування програмного забезпечення відділене від кодування і виділяється як наступна фаза. Головна мета тестування - показати, що додаток відповідає вимогам. Є базові підходи і практики тестування.

Рівень 3 - інтегрований. Процес тестування інтегрований в життєвий цикл розробки програмного забезпечення. Цілі тестування базуються на вимогах. Є організація тестування, а саме тестування виділено в професійну діяльність.

Рівень 4 - управління і вимірювання. Тестування є вимірюваним та контролюваним процесом. Процеси критичних оглядів (review) проектних артефактів (тестові плани і сценарії, повідомлення про помилки, підсумкові звіти про стан версії і т.д.) відносяться до тестових активностей. Продукт тестується на відповідність таким метрикам якості, як надійність, зручність, супроводжуємость. Тестові сценарії записані, зберігаються в системі управління тестуванням і можуть бути багаторазово використані разом з тестовими наборами даних. Виявлені дефек-

ти не тільки фіксуються, але й аналізуються за формальними ознаками. Проводиться оцінка якості програмних продуктів.

Рівень 5 - оптимізація процесу, запобігання помилок і контроль якості. Тестування є визначеним і керованим процесом. Вартість тестування нарівні з показниками ефективності може бути визначена. Тестування як процес піддається змінам, які однозначно позитивно на нього впливають. Впроваджені і використовуються практики запобігання помилок і контролю якості. Автоматизоване тестування застосовується як основний підхід у тестуванні. Проектування тестів, аналіз отриманих результатів, обробка описів помилок, а також метрик, пов'язаних з тестуванням, здійснюється за допомогою відповідних інструментальних засобів. Широко поширений підхід повторного використання процесних практик[4].

Всі перераховані рівні зрілості, крім першого, включають цілі розвитку, які, в свою чергу, містять підцілі, тобто дозволяють оперувати не тільки високорівневими завданнями менеджменту якості процесу розробки програмного забезпечення, але і формулювати оперативні завдання для всіх виконавців в проекті.

Висновки та перспективи подальших досліджень

Швидкий розвиток програмного забезпечення (ПЗ), комп'ютерної техніки, створення нових та вдосконалення існуючих технологій побудови програмних продуктів, розширення спектра використання автоматизованих систем у сучасному світі зумовлюють паралельний розвиток усіх складових процесу побудови та впровадження ПЗ.

Найвища мета тестування - це забезпечення якості: накопичення інформації, яка, повернувшись до програміста, допоможе уникнути йому минулих помилок і покращити якість програмного забезпечення в майбутньому.

Підвищення ефективності виконання задач, зменшити кількість помилок можливо за рахунок:

– залучення тестування з самого початку роботи над проектом

– розподіл обов'язків між учасниками процесу тестування

– використання запропонованих методів та моделі тестування для покращення планування та керування процесом.

На теперішній час дослідження тестування програмного забезпечення спрямоване на розробку нових методів та процесів тестування для програмних систем нової генерації - інтегрованих компонентів ПС різних типів, Web-застосувань, Web-сервісів, сімейств програмних продуктів.

Список літератури

1. Майерс Г. Искусство тестирования программ.-М: Финансы и статистика, 1982.
2. Белый ящик», «черный ящик» в тестировании ПО[Електронний ресурс] - <http://blog.beitnelly4u.com/testing-qa/white-box-black-box-testing-qa.html>
3. Андон Ф.И., Коваль Г.И., Коротун Т.М., Лавринцева Е.М., Суслов В.Ю. Основы инженерии качества программных систем 2-е изд., перераб. и доп. - К.: Академперіодика, 2007. – 672 с.
4. Модели зрелости процесса тестирования ПО// «Открытые системы», № 02, 2007
5. Бейзер Б. Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем. - СПб.:Питер, 2004. - 318с. - ISBN 5-94723-698-2
6. Д.Коул, Т. Горэм, М. МакДональд, Р. Спарджеон. Принципы тестирования ПО. //Открытые системы, №2, 1998 с. 60-63.