

Методологія проектування тестів для перевірки програмного забезпечення

Олексій Лопуга, аспірант кафедри ІТ¹ (ORCID: 0000-0001-6397-2710), Світлана Білощицька, доктор технічних наук, професор кафедри ІТ¹ (ORCID: 0000-0002-0856-5474)

¹ Київський національний університет будівництва і архітектури, 03037, м. Київ, проспект Повітряних Сил, 31, Україна

АНОТАЦІЯ

У цій роботі подано короткий підсумок досліджень тестування програмного забезпечення. Він окреслює значення тестування та його роль, представляє основні методи та моделі інженерного тестування програмного забезпечення для обробки даних, а також пропонує структуру основного процесу тестування та метод його оцінки.

Ключові слова: програмна система, життєвий цикл.

1. ВСТУП

З 1992 по 2006 рр. Національна академія наук України досліджувала тестування програмного забезпечення з метою підвищення його якості [1]. Основними напрямками були аналіз і вдосконалення методів тестування, розробка нових моделей надійності та оптимізація процесу тестування. Результати досліджень сприяли створенню ефективніших методів, що підвищили якість і надійність програмного забезпечення. Академія зробила значний внесок у забезпечення відповідності програмного забезпечення функціональним та нефункціональним вимогам.

2. ПОСТАНОВКА ПРОБЛЕМИ

Проблема повного тестування великих програмних систем у межах обмеженого часу і бюджету є постійним викликом для розробників. Ефективне тестування потребує ретельно спланованого і добре організованого процесу для раціонального розподілу ресурсів і прийняття обґрунтованих рішень. Інженерія програмного забезпечення як дисципліна розробки спрямована на підвищення якості та ефективності програмних систем за допомогою сучасних моделей і методів, стандартизації процесів, вимірювання її оцінки, а також відповідних засобів підтримки.

Сучасний процесно-орієнтований підхід до розробки програмного забезпечення включає кілька важливих вимог щодо тестування програмних систем. Зокрема, це початок тестування на ранніх етапах життєвого циклу проекту замість його відкладання до завершальних стадій, вибір методів і стратегій тестування, що відповідають як об'єктам і рівням тестування, так і наявним ресурсам проекту [1]. Додатково, стандартизація процесу тестування допомагає узгодити його з вимогами програмної інженерії, орієнтованої на процеси, а встановлення кількісних критеріїв дозволяє чітко визначити, коли тестування можна вважати завершеним.

У проектах із обмеженими ресурсами особливого значення набуває управління ризиками, особливо стосовно дотримання встановлених термінів і бюджету тестування. Це охоплює не лише організацію та управління тестуванням, але й визначення обсягу тестових робіт та пріоритизацію цілей, враховуючи ризик збоїв програмного забезпечення у виробництві. У цьому контексті формування підходу до тестової інженерії як однієї з основних складових програмної інженерії є актуальним і надзвичайно важливим

завданням для досягнення високої якості програмних продуктів.

3. АНАЛІЗ ОСНОВНИХ ДОСЛІДЖЕНЬ І ПУБЛІКАЦІЙ

Сфера тестування програмного забезпечення значно еволюціонувала, вдосконалюючи методи та критерії оцінки їх ефективності, кількісні показники та інструменти для підтримки й аналізу тестових процесів. З 1990-х років тестування більше не розглядають як окремий завершальний етап; воно інтегрувалося в кожну стадію життєвого циклу розробки, ставши безперервною діяльністю. Однією з важливих моделей є V-модель, яка структурує тестування як дві взаємопов'язані частини: підготовка (ліва гілка V) і виконання тестів (права гілка V) [2]. На кожному етапі аналізуються розроблені артефакти, що дозволяє уточнювати цілі, сценарії тестування, вимоги, а також оцінювати необхідні ресурси. Це підвищує якість розробки і дає можливість оптимізувати такі критерії тестування, як тривалість, вартість і надійність.

Перехід до ітераційних моделей розробки програмного забезпечення дозволяє починати тестування на ранніх етапах, з регулярними випусками нових версій, що надає можливість краще оцінити систему. Стратегії тестування в ітераційних моделях враховують частоту та сценарії використання, а також формують профілі програмного забезпечення для підвищення його надійності. Цей підхід враховує категорії користувачів, основні функції ймовірності їхнього використання.

Випуск програмного забезпечення у оптимальний час залишається важливим завданням, оскільки час на тестування завжди обмежений. Ідеальний момент визначається факторами, такими як надійність, вартість і терміни випуску. Серед популярних підходів – максимізація надійності, мінімізація витрат та багатокритеріальні методи прийняття рішень, які враховують кілька факторів одночасно. У складних програмних системах, які часто розробляються окремими командами, виникає потреба в комбінаторній оптимізації, що включає врахування витрат, надійності та інших значущих характеристик.

Однак традиційні методи визначення часу випуску не завжди враховують такі важливі фактори, як задоволеність користувачів, продуктивність і безпека, через що багатокритеріальні методи прийняття рішень стають дедалі популярнішими. Вони допомагають знайти оптимальне

рішення, зважаючи різні аспекти і забезпечуючи якісний випуск програмного продукту.

4. ОСНОВНІ РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

Методи тестування є важливою частиною процесу, що визначає його ефективність. Існує два основних підходи: тестування "чорного ящика", коли код недоступний, і "білого ящика", коли код доступний. Кожен метод має власні критерії покриття вхідного простору тестів. Вибір методу залежить від конкретних умов і рівнів тестування, а також передбачає аналіз ризиків збоїв системи. Методика підбирається з урахуванням типу програмної системи.

Через обмежену доступність автоматизованих інструментів, ітераційне прототипування часто використовується для швидкого створення перших версій програмних систем. Це дозволяє враховувати потенційні ризики на етапі планування тестування.

Ризик-орієнтована стратегія тестування ґрунтується на аналізі можливих відмов компонентів системи та формуванні критеріїв для оптимізації тестування. Ця методика дозволяє визначити, які компоненти потребують більш тривалого тестування, розподіляючи час і ресурси відповідно до ризиків. Метод враховує як структуру програмної системи, так і її передбачуване використання, що дозволяє мінімізувати ризики збоїв у роботі системи.

Базовий процес тестування: процесно-орієнтований підхід до розробки програмного забезпечення активно використовується для врахування ризиків системних збоїв і покращення процесу тестування через моделі та методи. Цей підхід інтегрується в життєвий цикл розробки, допомагаючи визначити оптимальний час для тестування компонентів і зменшити кількість збоїв [3].

Процес тестування складається з 10 етапів (рис. 1) підготовка, виконання та оцінка результатів. На етапі підготовки аналізуються робочі продукти, визначаються цілі, об'єкти, сценарії та ресурси тестування, що документується в планах випробувань. Під час виконання реєструються результати тестів і повідомляються дефекти. Завершальний етап включає оцінку результатів і вдосконалення для майбутніх тестувань.

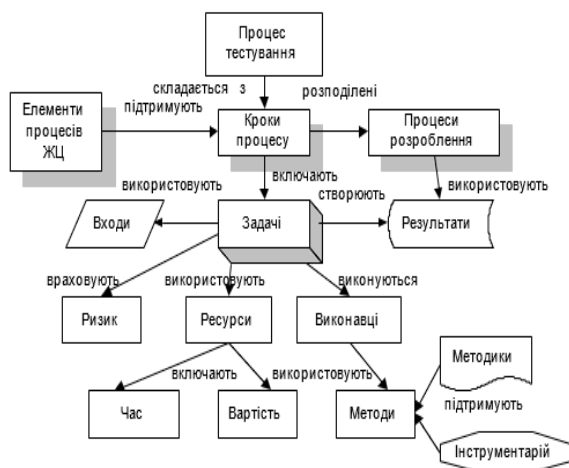


Рис. 1. Модель базового процесу тестування

Процес охоплює розподіл завдань, професійні вимоги, стандарти документування, метрики, методи тестування, критерії початку і завершення завдань, а також перехід до

наступного етапу. Для документування процесу розроблено шаблони для кожного етапу.

Моделі та методи вдосконалення процесу тестування: аналіз і розробка моделей оцінки процесу тестування спрямовані на його вдосконалення. Розроблено модель зрілості процесу тестування на основі п'ятирівневої моделі ТММ [2], яка включає цілі зрілості, проміжні цілі, а також дії та відповідальність за впровадження моделі [4].

Методика оцінки зрілості базується на моделі ТММ і включає опитування експертів проєкту для визначення поточного стану тестування і напрямів його вдосконалення.

5. ВИСНОВОК

Дослідження надали ключові висновки щодо фундаментального процесу тестування програмного забезпечення, а також методів і моделей, розроблених для його ефективного реалізації. Ці висновки сприяють глибшому розумінню тестових технік та практик, що використовуються в індустрії.

Результати дослідження були перевірені в рамках кількох проєктів розробки програмного забезпечення, що, в свою чергу, призвело до підвищення загальної ефективності, зменшення кількості дефектів у фінальному продукті та значного зниження витрат на виправлення помилок. Такі позитивні зміни стали можливими завдяки інтеграції процесів тестування та розробки, що забезпечило більш злагоджену роботу команди. Організована діяльність команди тестування, в поєднанні з використанням запропонованих моделей і методів, сприяла кращому плануванню та управлінню всім процесом тестування, що в свою чергу підвищило загальну якість розробленого програмного забезпечення.

В даний час дослідження в сфері інженерії тестування активно зосереджуються на розробці нових методів, що відповідають вимогам систем програмного забезпечення наступного покоління. Це охоплює інтегровані компоненти, веб-додатки, веб-сервіси та сімейства програмних продуктів, які постійно еволюціонують у відповідь на зміни в технологічному середовищі.

Кінцевою метою таких досліджень є вдосконалення інтегрованого середовища розробки, використовуючи інноваційні методи тестування систем, які створюються за допомогою генеративного програмування. Це дозволить не тільки підвищити якість та надійність програмних продуктів, але й забезпечити їх адаптацію до швидко змінюваних вимог ринку та технологій.

Список літератури

- [1] Коротун Т.М. "Тестирование информационных систем с учетом риска", 2000, р. 385 – 392.
- [2] Коротун Т.М. "Совершенствование процесса тестирования программного обеспечения", 1998, р. 59 – 64.
- [3] С.В. Цюцюра, О.В. Криворучко, М.І. Цюцюра, "Практика застосування ключових показників ефективності та впровадження системи цільового управління", 2012, р. 76.
- [4] ОВ Криворучко, СВ Цюцюра, "Основи експертних систем: Навч. посіб.", 2006, р.5.