

ДОСЛІДЖЕННЯ ВПЛИВУ ФУНКЦІОНАЛЬНИХ ВИМОГ НА ЯКІСТЬ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Запорізька державна інженерна академія, кафедра ПЗАС

Успіх процесу розробки сучасних програмних систем залежить від багатьох факторів. Для створення якісного програмного забезпечення, яке б повністю задовольняло потреби користувача, необхідно власне виявити проблеми клієнтів та забезпечити їх кваліфіковане вирішення.

Основним наслідком проблем з вимогами є переробка вже існуючих частин програмного продукту, на що, згідно з сучасними дослідженнями, витрачається від 30 до 50% загального бюджету розробки [2]. Неякісно протестована система може принести до 70% збитків компанії, при цьому ризики значно зростають на завершальних етапах розробки та на стадіях впровадження та супроводу [4].

Сучасні методи розробки програмних систем припускають впровадження процесу тестування ПЗ водночас з процесом виявлення вимог та розробкою програмного продукту по специфікації, що надана аналітиками.

Коли готова перша порція вимог, тестувальник вже може приступати до написання тестових випадків або сценаріїв. Якщо вимоги неякісні — суперечливі, двозначні, неповні тощо — тестувальник виявить це в процесі написання тестів, тому що методики, які він використовує, направлені на покриття всієї множини станів і переходів, вхідних даних та ситуацій відмови. Таким чином, відбувається побудова моделі майбутньої системи за наявними вимогами та складаються тести для перевірки цієї моделі. Якщо модель описана не повністю, або її частки не узгоджені між собою, цю модель неможливо буде перевірити, тож написання тестів дає зворотний зв'язок про якість набору функціональних вимог. З іншого боку, правильно і точно описані бізнес-вимоги також збагачують тестування у вигляді інформації про пріоритети. Проведення трасування від функціональних вимог до вимог вищих рівнів — користувальницьких та бізнес-вимог — дозволяє перевірити коректність побудови специфікацій та простежити відображення вимог на компоненти архітектури системи. Спілкуючись з аналітиками, тестувальники краще розуміють, які функції або характеристики якості системи є найбільш важливими, що дає можливість розподіляти зусилля з тестування відповідно до бізнес-пріоритетів. Тестування та вимоги до системи пов'язані між собою синергетичними відносинами, оскільки представляють собою взаємодоповнюючі погляди на систему [1].

Для формалізації функціональних вимог системи, які покращать розуміння вимог розробниками та тестувальниками, можна використовувати наступні методи:

- побудова варіантів використання;
- написання псевдокоду;
- скінчені автомати;
- графічні дерева рішень;
- моделі «сутність-зв'язок»;
- діаграми діяльності;
- таблиці рішень;
- об'єктно-орієнтовані моделі;
- схеми потоків даних [2,3].

Всі ці методи також можуть використовуватися для виявлення та узгодження вимог зацікавлених осіб проекту разом мозковим штурмом, ролевими іграми, прототипуванням та розкадровкою.

При аналізі функціональних вимог основну увагу слід приділяти визначенню основних критеріїв успішності проведених тестів. Задані у вимогах характеристики будуть базовими навантажувальними точками тестованої системи. Всі одержувані результати порівнюватимуться з ними для прийняття рішення про завершення або проведення подальшого регресійного тестування.

Оскільки процеси тестування та розробки знаходяться на одному ребрі трикутника час-гроші-якість, необхідно жорстко зв'язувати та синхронізувати ці два напрямки, інакше існує великий ризик того, що зсув плану розробки, який впливає на дату поставки версії в тестування, призведе до здвигу робіт по тестуванню, а це може стати причиною випуску неякісного програмного продукту. До того ж, вже на етапі видобуття вимог до системи необхідно чітко розставляти пріоритети вимог, які потім будуть перетворені на окремі функції, створюючи документ про образ та межі проекту, для з'ясування ризиків, які можуть виникнути при скороченні строків розробки або тестування.

Вимоги повинні використовуватися для оцінки масштабу проекту. Об'єм продукту можна визначити виходячи як з текстових вимог, так і моделей аналізу, прототипів та елементів користувацького інтерфейсу. Коректна оцінка часу та зусиль які знадобляться для розробки програмного продукту дозволить зосередитись на всіх етапах життєвого циклу, включаючи верифікацію, яка зазвичай страждає при скороченні строків розробки, що значно підвищить цінність розроблюваної системи. Вимоги є одним з артефактів життєвого циклу програмного продукту, тож вони також мають підлягати верифікації. Ця процедура дозволить виявити нечіткі вимоги, або вимоги, які не можна перевірити, а також ті, що були визначені недостатньо чітко для розробки. Неякісно виявлені або сформульовані вимоги неминуче призведуть до неточних розрахунків розміру, витрат і графіку. На розробку вимог середнього проекту необхідно витрачати близько 7% всіх ресурсів та 38,6% часу [1].

Успішне виявлення функціональних вимог на початку роботи з проектом не забезпечує успіху проекту, адже вимоги мають властивість змінюватися під час розробки програмного продукту. Необхідно здійснювати керування вимогами, яке полягає в керуванні змінами базової версії вимог, підтримці планів проекту актуальними у відповідності до змін вимог, контролі стану вимог та керуванні логічними зв'язками між вимогами та їх втіленням у артефактах проекту. Кожну зміну вимог в процесі розробки програмного продукту має бути відстежено та протестовано. Microsoft Visual Studio надає спеціальні можливості як для тестування (Team Foundation Server Test Manager), так і для документування та керування змінами вимог (Team Foundation Server Requirements Management), які можуть бути використані під час процесу розробки сучасних програмних продуктів.

Висновки:

Таким чином, розробка якісної специфікації вимог та її подальше використання для побудови тестів на основі потреб користувачів є основою вчасного випуску високоякісних програмних проектів. Для покращення підтримки співпраці аналітиків та тестувальників рекомендується використання автоматизованих засобів з підтримкою бази вимог та можливістю їх відстеження та трасування в архітектуру та артефакти системи та подальші приймальні випробування готового продукту.

Література

1. Вигерс К. Разработка требований к программному обеспечению. /К. Вигерс —М.: Издательско -торговый дом «Русская Редакция», 2004. —576с.: ил.
2. Леффингуелл Д. Принципы работы с требованиями к программному обеспечению. /Д. Леффингуелл, Д. Уидриг — М.: ИД “Вильямс”, 2002—448с.
3. Мацяшек Л. Анализ требований и проектирование систем. Разработка информационных систем. /Л.Мацяшек — М.:Вильямс,2002.-428 с.
4. Тамре Л. Введение в тестирование программного обеспечения.: Пер. с англ./Л.Тамре — М.: Издательский дом «Вильямс», 2003. - 368 с.