Сергей, [12.12.20 19:06]

5.2 Тестування методом «чорного ящика»

При тестуванні методом «чорного ящика» тестувальник має доступ до ПЗ тільки через ті ж інтерфейси, що і замовник або користувач, або через зовнішні інтерфейси, що дозволяють іншого комп'ютера або іншому процесу підключитися до системи для тестування. Як правило, тестування «чорного ящика» ведеться з використанням специфікацій чи інших документів, що описують вимоги до системи. У даному вигляді тестування намагаються забезпечити покриття вимог і вхідних даних.

Тестування на основі стратегії чорного ящика можливе лише за наявності встановлених відкритих інтерфейсів, таких як інтерфейс користувача або програмний інтерфейс додатку (API). Якщо тестування на основі стратегії білого ящика досліджує внутрішню роботу програми, то методи тестування чорного ящика порівнюють поведінку додатки з відповідними вимогами. Крім того, ці методи зазвичай спрямовані на виявлення трьох основних видів помилок: функціональності, підтримуваної програмним продуктом; вироблених обчислень; допустимого діапазону або області дії значень даних, які можуть бути опрацьовані програмним продуктом. На цьому рівні тестувальники не досліджують внутрішню роботу компонентів програмного продукту, тим не менше вони перевіряються неявно.

13

Група тестування вивчає вхідні і вихідні дані програмного продукту. У цьому ракурсі тестування за допомогою методів чорного ящика розглядається як синонім

тестування на рівні системи, хоча методи чорного ящика можуть також застосовуватися під час модульного або компонентного тестування.

При тестуванні методами чорного ящика важлива участь користувачів, оскільки саме вони найкраще знають, яких результатів слід очікувати від бізнес-функцій. Ключем до успішного завершення системного тестування є коректність даних. Тому на фазі створення даних для тестування вкрай важливо, щоб кінцеві користувачі надали якомога більше вхідних даних.

Тестування за допомогою методів чорного ящика спрямоване на одержання множин вхідних даних, які найбільш повно перевіряють всі функціональні вимоги системи. Це не альтернатива тестуванню за методом білого ящика. Цей тип тестування націлений на пошук помилок, які відносяться до цілого ряду категорій, серед них:

v Невірна або пропущена функціональність

v Помилки інтерфейсу

v Проблеми зручності використання

v Методи тестування на основі Автоматизовані інструменти

v Помилки в структурах даних або помилки доступу до зовнішніх баз даних

v Проблеми зниження продуктивності та інші помилки продуктивності

v Помилки завантаження

v Помилки багатокористувацького доступу

v Помилки ініціалізації та завершення

v Проблеми збереження резервних копій і здатності до відновлення роботи

v Проблеми безпеки

v Методи тестування на основі стратегії чорного ящика

Рішення цих помилок може бути знайдено за допомогою таких методів тестування:

14

Еквівалентна розбиття. Повне тестування вхідних даних, як правило, нездійсненно. Тому слід проводити тестування з використанням підмножини вхідних даних.

При тестуванні помилок, пов'язаних з виходом за межі області допустимих значень, застосовують три основних типи еквівалентних класів: значення всередині межі діапазону, за межею діапазону і на кордоні. Виправдовує себе практика створення тестових процедур, які перевіряють граничні випадки плюс/мінус один щоб уникнути пропуску помилок «на одиницю більше» або «на одиницю менше». Крім розроблення тестових процедур, що використовують сильно структуровані класи еквівалентності, група тестування повинна провести дослідницьке тестування. Тестові процедури, при виконанні яких видаються очікувані результати, називаються правильними тестами. Тестові процедури, проведення яких має призвести до помилки, носять назву неправильних тестів.

Сергей, [12.12.20 19:06]

? Аналіз граничних значень. Аналіз граничних значень можна застосувати як на структурному, так і на функціональному рівні тестування. Межі визначають дані трьох типів: правильні, неправильні і лежать на границі. Тестування кордонів використовує значення, що лежать всередині або на кордоні (наприклад, крайні точки), і максимальні/мінімальні значення (наприклад, довжини полів). При такому дослідженні завжди повинні враховуватися значення на одиницю більше і менше граничного. При тестуванні за межами кордону використовується репрезентативний зразок даних, що виходять за межу, тобто невірні значення.

Діаграми при чинно-наслідкових зв'язків. Складання діаграм причинно-наслідкових зв'язків - це метод, що дає чітке уявлення про логічних умовах і відповідних діях. Метод передбачає чотири етапи. Перший етап полягає в складанні переліку причин (умов введення) і наслідків (дій) для модуля й у присвоєнні ідентифікатора кожному модулю. На другому етапі розробляється діаграма причинно-наслідкових зв'язків. На третьому етапі діаграма перетворюється в таблицю рішень. Четвертий етап включає в себе встановлення причин і наслідків в процесі читання специфікації функцій. Кожній причини і слідству присвоюється власний ідентифікатор. Причини перераховуються в стовпчику з лівого боку аркуша паперу, а слідства - з правого. Потім причини і наслідки з'єднуються лініями так, щоб були відображені наявні між ними відповідності. На діаграмі проставляються булеві вирази, які об'єднують дві або більше причин, пов'язаних зі слідством. Далі правила таблиці рішень перетворюються в тестові процедури.

15

Системне тестування. Термін «системне тестування» часто вживається до?? до синонім «тестування за допомогою методів чорного ящика», оскільки під час системного тестування група тестування розглядає в основному «зовнішня поведінка» додатка.

Системне тестування включає в себе кілька підтипів тестування, у тому числі функціональне, регресійне, безпеки, перевантажень, продуктивності, зручності використання, випадкове, цілісності даних, перетворення даних, збереження резервних копій і здатності до відновлення, готовності до роботи, приймально-здавальні випробування і альфа/бета тестування.

Функціональне тестування. Функціональне тестування перевіряє системне додаток відносно функціональних вимог з метою виявлення невідповідності вимогам кінцевого користувача. Для більшості програм тестування програмного продукту даний метод тестування є головним. Його основне завдання - оцінка того, чи працює додаток відповідно до ставляться.

Регресійне тестування. Сенс проведення тестування полягає у виявленні дефектів, їх документуванні та відстеженні аж до усунення. Тестувальник повинен бути впевнений в тому, що заходи, прийняті для усунення знайдених помилок, які не породять у свою чергу нових помилок в інших областях системи. Регресійне тестування дозволяє з'ясувати, чи не з'явилися які-небудь помилки в результаті ліквідації вже виявлених помилок. Саме для регресійного тестування застосування інструментів автоматизованого тестування дає найбільшу віддачу. Всі створені раніше скрипти можна використовувати знову для підтвердження того, що в результаті змін, внесених при усуненні помилки, чи не з'явилися нові дефекти. Ця мета легко досяжна, оскільки скрипти можна виконувати без ручного втручання і використовувати стільки разів, скільки необхідно для виявлення помилок.

? Тестування безпеки. Тестування безпеки включає в себе перевірку роботи механізмів доступу до системи і до даних. Для цього придумують тестові процедури, які намагаються подолати захист системи. Тестувальник перевіряє ступінь безпеки й обмеження доступу, виявляючи таким чином відповідність встановленим вимогам до безпеки і всім застосовуваним правилам з безпеки системи.

Сергей, [12.12.20 19:06]

? Тестування перевантажень. При тестуванні перевантажень виконується перевірка системи без врахування обмежень архітектури з цілі виявлення технічних обмежень системи. Ці тести проводяться на піку обробки транзакцій і при безперервному завантаженні великого обсягу даних. Тестування перевантажень вимірює пропускну спроможність системи та її еластичність (resiliency) на всіх апаратних платформах.

16

Цей метод передбачає одночасне звернення з боку багатьох користувачів до певних функцій системи, причому деякі вводять значення, що виходять за межі норми. Від системи потрібна обробка величезної кількості даних або виконання великого числа функціональних запитів протягом короткого періоду часу.

? Тестування продуктивності. Тести продуктивності перевіряють, чи задовольняє системне додаток вимогам по продуктивності. Застосовуючи тестування продуктивності, можна заміряти і скласти звіти за такими показниками, як швидкість передачі вхідних і вихідних даних, загальне число дій по введенню і виведенню даних, середній час, що витрачається базою даних на відгук на запит, і інтенсивність використання центрального процесора. Як правило, для автоматичної перевірки ступеня продуктивності, проведеної в рамках тестування продуктивності, використовуються ті ж інструменти, що і при тестуванні перевантажень.

? Тестування зручності використання. Тести зручності використання спрямовані на підтвердження простоти застосування системи і того, що користувальницький інтерфейс виглядає привабливо. Такі тести враховують людський фактор у роботі системи. Тестувальника потрібно оцінити програму з точки зору кінцевого користувача.

lign="justify">.

3 Тестування методом «сірого ящика»

При тестуванні методом «сірого ящика» розробник тесту має доступ до вихідного коду, але при безпосередньому виконанні тестів доступ до коду, як правило, не потрібно. Тестування проводиться так само, як і в методі «чорного ящика», однак для побудови тестів використовуються знання про внутрішній устрій програми.