Лабораторная работа 10 Уровни тестирования программного обеспечения

Цель работы

Исследовать уровни тестирования программного обеспечения.

Пояснения к работе

Краткие теоретические сведения

Тестирование на разных уровнях производится на протяжении всего жизненного цикла разработки и сопровождения программного обеспечения. Уровень тестирования определяет то, **над чем** производятся тесты: над отдельным модулем, группой модулей или системой, в целом. Проведение тестирования на всех уровнях системы - это залог успешной реализации и сдачи проекта.

Модульное тестирование

Модульное тестирование (unit testing) — процесс, позволяющий проверить на корректность отдельные модули исходного кода программы.

Цель модульного тестирования — изолировать отдельные части программы и показать, что по отдельности эти части работоспособны.

Идея состоит в том, чтобы писать **тесты** для каждой нетривиальной **функции** или **метода.** Это позволяет достаточно быстро проверить, не привело ли очередное изменение кода к **регрессии**, то есть к появлению ошибок в уже оттестированных местах программы, а также облегчает обнаружение и устранение таких ошибок.

Этот тип тестирования обычно выполняется **программистами**.

Модульное тестирование проверяет функциональность и ищет дефекты в частях приложения, которые доступны и могут быть протестированы по отдельности (**модули программ, объекты, классы, функции и т.д.**). Обычно компонентное (модульное) тестирование проводится вызывая код, который необходимо проверить и при поддержке сред разработки, таких как фреймворки (frameworks - каркасы) для модульного тестирования или инструменты для отладки. Все найденные дефекты, как правило исправляются в коде без формального их описания в системе менеджмента багов (Bug Tracking System).

Один из наиболее эффективных подходов к компонентному (модульному) тестированию - это **подготовка автоматизированных тестов** до начала основного кодирования (разработки) программного обеспечения. Это называется разработка от тестирования (**test-driven development**) или подход тестирования вначале (**test first approach**). При этом подходе создаются и интегрируются небольшие куски кода, напротив которых запускаются тесты, написанные до начала кодирования. Разработка ведется до тех пор пока все тесты не будут успешными.

Интеграционное тестирование

Интеграционное тестирование (Integration testing) – тестирование, при котором отдельные программные **модули объединяются** и **тестируются в группе**.

Обычно интеграционное тестирование проводится после модульного тестирования и предшествует системному тестированию.

Целью интеграционного тестирования является проверка соответствия проектируемых единиц функциональным, приёмным требованиям и требованиям надежности. Тестирование этих проектируемых единиц выполняется через их интерфейс, с использованием тестирования «**чёрного ящика**».

Интеграционное тестирование предназначено для проверки связи между компонентами, а также взаимодействия с различными частями системы (операционной системой, оборудованием либо связи между различными системами).

Уровни интеграционного тестирования:

- **Компонентный интеграционный уровень** (*Component Integrationtesting*)

Проверяется взаимодействие между компонентами системы после проведения компонентного тестирования.

- **Системный интеграционный уровень** (*System Integration Testing*)

Проверяется взаимодействие между разными системами после проведения системного тестирования.

Подходы к интеграционному тестированию:

- **Снизу вверх** (*Bottom Up Integration*)

Все низкоуровневые модули, процедуры или функции собираются воедино и затем тестируются. После чего собирается следующий уровень модулей для проведения интеграционного тестирования. Данный подход считается полезным, если все или практически все модули, разрабатываемого уровня, готовы. Также данный подход помогает определить по результатам тестирования уровень готовности приложения.

- **Сверху вниз** (*Top Down Integration*)

Вначале тестируются все высокоуровневые модули, и постепенно один за другим добавляются низкоуровневые. Все модули более низкого уровня симулируются заглушками с аналогичной функциональностью, затем по мере готовности они заменяются реальными активными компонентами. Таким образом мы проводим тестирование сверху вниз.

- **Большой взрыв** (*"Big Bang" Integration*)

Все или практически все разработанные модули собираются вместе в виде законченной системы или ее основной части, и затем проводится интеграционное тестирование. Такой подход очень хорош для сохранения времени. Однако если тест кейсы и их результаты записаны не верно, то сам

процесс интеграции сильно осложнится, что станет преградой для команды тестирования при достижении основной цели интеграционного тестирования.

Системное тестирование

Системное тестирование — это тестирование, выполняемое на полной, **интегрированной системе**, с целью проверки соответствия системы **исходным требованиям**. Системное тестирование относится к методам тестирования **чёрного ящика**, и, тем самым, не требует знаний о внутреннем устройстве системы.

Основной **задачей** системного тестирования является **проверка** как **функциональных**, так и **не функциональных требований** к системе **в целом**. При этом выявляются дефекты, такие как:

• неверное использование ресурсов системы,

• непредусмотренные комбинации данных пользовательского уровня,

• несовместимость с окружением,

• непредусмотренные сценарии использования,

• отсутствующая или неверная функциональность,

• неудобство использования и т.д.

Основной задачей системного тестирования является **проверка как функциональных, так и не функциональных требований в системе в целом**. При этом выявляются дефекты, такие как неверное использование ресурсов системы, непредусмотренные комбинации данных пользовательского уровня, несовместимость с окружением, непредусмотренные сценарии использования, отсутствующая или неверная функциональность, неудобство использования и т.д. Для минимизации рисков, связанных с особенностями поведения в системы в той или иной среде, **во время тестирования рекомендуется использовать окружение максимально приближенное к тому, на которое будет установлен продукт после выдачи**.

Можно выделить два подхода к системному тестированию:

• **на базе требований** (*requirements based*)

Для каждого требования пишутся тестовые случаи (test cases), проверяющие выполнение данного требования.

• **на базе случаев использования** (*use case based*)

На основе представления о способах использования продукта создаются случаи использования системы (**UseCases**). По конкретному случаю использования можно определить один или более сценариев. На проверку каждого сценария пишутся тест кейсы (test cases), которые должны быть протестированы.

Задание

Изучить уровни тестирования на примере одного из разработанных ранее приложений: модульный, интеграционный и системный уровни. Отчет должен содержать план тестирования известного приложения, краткие выводы.

Содержание отчета

Контрольные вопросы

1. Является ли требование частью функциональной спецификации?
2. Для чего предназначено интеграционное тестирование?

3. На каких уровнях разработки программного обеспечения произовдится тестирование?
4. Какое программное окружение рекомендуется использовать на уровне системного тестирования?

Литература

2. «Инженерия программного обеспечения», Соммервилл И.
3. Основы Программной Инженерии (по SWEBOK) http://swebok.sorlik.ru/
4. Майерс Г. Искусство тестирования программ.-М.:Финансы и статистика ,1982.-178с.