

Домашняя работа

Вопрос 1

Опишите своими словами, как вы понимаете уровни тестирования. Что необходимо указать в ответе - какие бывают виды, кто чаще всего выполняет тесты на конкретном уровне, приведите пример одного вида тестирования, которое можно выполнять на каждом из уровней.

Ответ: Классификация тестирования по уровням была придумана для того, чтобы у тестировщиков было понимание, что им необходимо тестировать: часть приложения, сразу несколько частей объединенных в группу, систему или же готовое приложение. То есть уровень тестирования определяет, то над чем проводятся тесты.

Существует 4 уровня тестирования:

1. Модульное тестирование (Unit testing)

Unit testing 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 модульное тестирование проводится разработчиками приложения, вызывая код. Все найденные дефекты, как правило исправляются в коде без формального их описания в системе менеджмента багов (Bug Tracking System). Часто данный вид тестирования реализуется с использованием специальных технологий и инструментальных средств автоматизации тестирования, значительно упрощающих и ускоряющих разработку соответствующих тест-кейсов.

Результатом тестирования и верификации отдельных модулей, составляющих программную систему, является заключение о том, что эти модули являются внутренне непротиворечивыми и соответствуют требованиям.

В модульном тестировании применяется динамическое, функциональное тестирование по методу белого ящика.

2. Интеграционное тестирование (integration testing) или его еще называют сборочным тестированием (тестирование сборки) предназначено для проверки связи между компонентами в сборке, а также взаимодействия с различными частями системы.

В качестве входных данных использует группы, состоящие из модулей, над которыми было проведено модульное тестирование.

Применяется функциональное, нефункциональное тестирование, метод белого ящика, smoke test и другие.

Тестирование проводится в графической среде, тестирование проводят разработчики и тестировщики.

3. Системное тестирование (System testing) направлено на тестирование приложения или системы целиком на соответствие функциональным и нефункциональным требованиям спецификации.

Во время тестирования рекомендуется использовать окружение максимально приближенное к тому, на которое будет установлен продукт после выдачи. Тестирование проводится тестировщиками, методом черного ящика (не требует знаний о внутреннем устройстве системы). Применяется функциональное, нефункциональное тестирование, альфа-тестирование, применение test cases на

базе требований и test cases на основе сценариев (use cases), негативное и позитивное тестирование и др.

4. Приемочное тестирование (Acceptance testing) - это комплексное тестирование, проводится перед выводом продукта в промышленную среду. Решение о проведении приемочного тестирования принимается, когда продукт достиг необходимого уровня качества и заказчик ознакомлен с планом приемочных работ (Product Acceptance Plan) или иным документом, где описан набор действий, связанных с проведением приемочного тестирования, дата проведения, ответственные и т.д. Приемочное тестирование выполняется на основании набора типичных тестовых случаев и сценариев, разработанных на основании требований к данному продукту.

На данном уровне завершенное приложение (система) тестируется тестировщиками, заказчиком, конечными пользователями с целью определения соответствия системы “Требованиям Заказчика” и готовности системы к внедрению. Приемосдаточные испытания оформляют процесс передачи продукта от разработчика заказчику. Приемочное тестирование длится до тех пор, пока заказчик не выносит решение об отправлении приложения на доработку или принятия решения о том, что продукт готов к внедрению.

В зависимости от особенностей продукта и от требований заказчика они могут проводиться в различной форме: например в виде альфа- или бета-тестирования на промышленном окружении, или выполнения тест-кейсов на тестовом окружении, максимально приближенном к промышленному.

На всех уровнях тестирования можно применить функциональное тестирование, так как функциональность - это главное для чего разрабатывается продукт, а соответственно функциональность приложения должна соответствовать требованиям заказчика, которые описаны в спецификации. При этом на каждом уровне происходит тестирование функциональности с разной детализацией - от небольшой функции (на этапе юнит тестирования) до большой функциональности (включающей выполнение нескольких функций) на следующих уровнях.

Вопрос 2

Что такое регрессионное тестирование? В каких случаях его необходимо выполнять? Какие особенности есть у этого вида тестирования?

Ответ:

Регрессионное тестирование - это вид тестирования направленный на проверку работоспособности ранее разработанной функциональности программы и/или ее окружения, для подтверждения того, что предыдущая версия программы работает с новыми изменениями так же, как и раньше

Регрессионное тестирование необходимо выполнять после исправления ошибок или добавления нового кода.

Особенность тестирования сводится к тому, что перед выпуском очередной версии программу пропускают через набор тестовых сценариев, подготовленных для предыдущей версии.

Вопрос 3

Что такое юнит-тесты? В чем их преимущества? Почему в большинстве случаев нельзя ограничиться только юнит-тестами?

Ответ:

Юнит-тесты - это тест, предназначенный для проверки работоспособности небольшой части программы (модуля). Преимущество unit-test в том, что каждую функциональность можно протестировать отдельно от всей программы с использованием всевозможных входящих данных и внешних условий (с использованием заглушек). Unit-test проверяет функциональность отдельной части программы, но не проверяет взаимодействие различных частей.

Вопрос 4

Предположим, вам предлагают протестировать какой-то проект (программу, сайт). Чем вы будете руководствоваться, выбирая конкретные виды тестирования для этого проекта?

Ответ: При выборе вида тестирования необходимо руководствоваться предоставленными требованиями (их наличием, полнотой) и информацией о внутреннем устройстве системы (из каких модулей состоит, как эти модули взаимодействуют между собой, библиотеки, базы данных и др.)
Например: при тестировании сайта интернет-магазина был применено свободное (интуитивное) тестирование (ad hoc testing), так как не были предоставлены требования.

Вопрос 5

Что такое тестирование черного ящика (Black Box Testing) и чем оно отличается от тестирования белого ящика (White box testing)? Приведите пример вида тестирования, которое невозможно при тестировании черного ящика.

Ответ:

Тестирование черного ящика - это метод тестирования проводимый без знаний как реализована функциональность, а по тест-кейсам написанными на базе требований. Тестирующий знает какой должен быть результат при определенном наборе данных, которые подаются на вход, и в процессе тестирования проверяет соответствие фактического результата заявленному в требованиях. В отличие от тестирования белого ящика тестирующий не вникает в процесс выполнения функции, только сравнивает фактический результат с ожидаемым.

Вид тестирования, который невозможен при тестировании черного ящика: Unit-test