

Manual QA course

Lecture 6. Уровни тестирования программного обеспечения. Часть 2

Дорофеев Максим

Уровни тестирования

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

Системное тестирование (System Testing)

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 системы в той или иной среде, во время тестирования рекомендуется использовать окружение максимально приближенное к тому, на которое будет установлен продукт после выдачи

Системное тестирование (System Testing)

Цель системного тестирования – изучить функциональность системы на этапах сборки каждой версии продукта, а также на выпуске ПО, в виде альфа и бета – тестирования.

Системное тестирование (System Testing)

Когда проводится:

- Модульное и интеграционное тестирование успешно завершено;
- Разработка ПО в соответствии с требованиями спецификаций завершена;
- Создана соответствующая тестовая среда для проведения системного тестирования;

Системное тестирование (System Testing). Виды

- Функциональное тестирование;
- Тестирование производительности;
- Нагрузочное или стресс тестирование;
- Тестирование конфигурации;
- Тестирование безопасности;
- Тестирование на отказ и восстановление;
- Тестирование удобства использования.

Системное тестирование (System Testing)

Подходы 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 (**Use Cases**). По конкретному случаю
использования можно определить один или более сценариев. На
проверку каждого сценария пишутся тест кейсы (**test cases**), которые
должны быть протестированы

Системное тестирование (System Testing)

- Создается Тест – план;
- Создаются тест и use – кейсы;
- Производится прогон авто – тестов;
- Затем ручное тестирование;
- Формируется отчет об ошибках;

Приемочное тестирование или Приемосдаточное испытание (Acceptance Testing)

Формальный процесс тестирования, который проверяет соответствие системы требованиям и проводится с целью:

- Определения удовлетворяет ли система приемочным критериям;
- Вынесения решения заказчиком или другим уполномоченным лицом принимается приложение или нет

Приемочное тестирование или Приемочное испытание (Acceptance Testing)

Приемочное тестирование выполняется на основании набора типичных тестовых случаев и сценариев, разработанных на основании требований к данному приложению

Приемочное тестирование или Приемосдаточное испытание (Acceptance Testing)

Решение о проведении приемочного тестирования принимается, когда:

продукт достиг необходимого уровня качества;
заказчик ознакомлен с Планом Приемочных Работ (Product Acceptance Plan (**ссылка на шаблон в приложении**)) или иным документом, где описан набор действий, связанных с проведением приемочного тестирования, дата проведения, ответственные и т. д.

Приемочное тестирование или Приемосдаточное испытание (Acceptance Testing)

Фаза приемочного тестирования длится до тех пор, пока заказчик не выносит решение об отправлении приложения на доработку или выдаче приложения

Интеграционное тестирование (Integration Testing)

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

Интеграционное тестирование (Integration Testing)

Уровни интеграционного тестирования:

Компонентный интеграционный уровень
(**Component Integration testing**)

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

Интеграционное тестирование (Integration Testing)

Component Integration testing - проверяется взаимодействие между компонентами системы после проведения компонентного тестирования

Интеграционное тестирование (Integration Testing)

System Integration Testing - проверяется взаимодействие между разными системами после проведения системного тестирования

Интеграционное тестирование (Integration Testing)

Подходы к интеграционному тестированию:

- Снизу вверх (Bottom Up Integration)
- Сверху вниз (Top Down Integration)
- Большой взрыв ("Big Bang" Integration)

Интеграционное тестирование (Integration Testing)

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

Интеграционное тестирование (Integration Testing)

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

Интеграционное тестирование (Integration Testing)

"**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

Интеграционное тестирование (Integration Testing)



Самый главный навык QA

Искусство поиска информации!!

Искусство поиска информации

- Не ждите универсальных рецептов
- Помните, что поисковая машина – это робот
- Определите зону поиска и ищите сразу там
- Искать по конкретным сайтам эффективнее
- Не ленитесь разбираться в настройках

Искусство поиска информации

- По одной фразе легко найти весь текст. Но не всегда
- Поиск в интернете – это еще и про общение

Искусство поиска информации

- Поиск по кэшу Google index of /<слово, фраза>
- Поиск при помощи знаков «+», «-» и «~»
- Что бы найти точную фразу, ее необходимо взять в «»
- Поиск необходимого расширения файлов filetype:<расширение файла>
- Поиск с помощью intitle:, inurl;

Вопросы и ответы



ССЫЛКИ

https://en.wikipedia.org/wiki/Integration_testing#Bottom_Up

https://en.wikipedia.org/wiki/Integration_testing#Big_Bang

https://en.wikipedia.org/wiki/Integration_testing#Top-down_and_Bottom-up

<https://meduza.io/cards/kak-pravilno-guglit>

<https://rootsector.blogspot.com/2014/12/google-hacking-google.html>

http://www.protesting.ru/documentation/product_acceptance_plan_rup.zip