Разработка web-приложений

Тестирование

Цыгулин Алексей Александрович к.т.н.

Что такое тестирование

проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом.

- анализ и планирование;
- разработка тестовых сценариев;
- оценка критериев окончания тестирования;
- отчеты;
- рецензирование документации (в том числе и исходного кода);
- проведение статического анализа.

Верификация и валидация

- **Верификация (verification)** – это процесс оценки системы или её компонентов с целью определения того, удовлетворяют ли результаты текущего этапа разработки условиям, сформированным в начале этого этапа. То есть, выполняются ли задачи, цели и сроки по разработке продукта.
- **Валидация (validation)** – это определение соответствия разрабатываемого ПО ожиданиям и потребностям пользователя, требованиям к системе.

Верификация и валидация

Верификация:

– Педали есть? – Есть.
– Седло есть? – Есть.
– Цепь есть? – Есть.
– Все есть…? – Да, все.

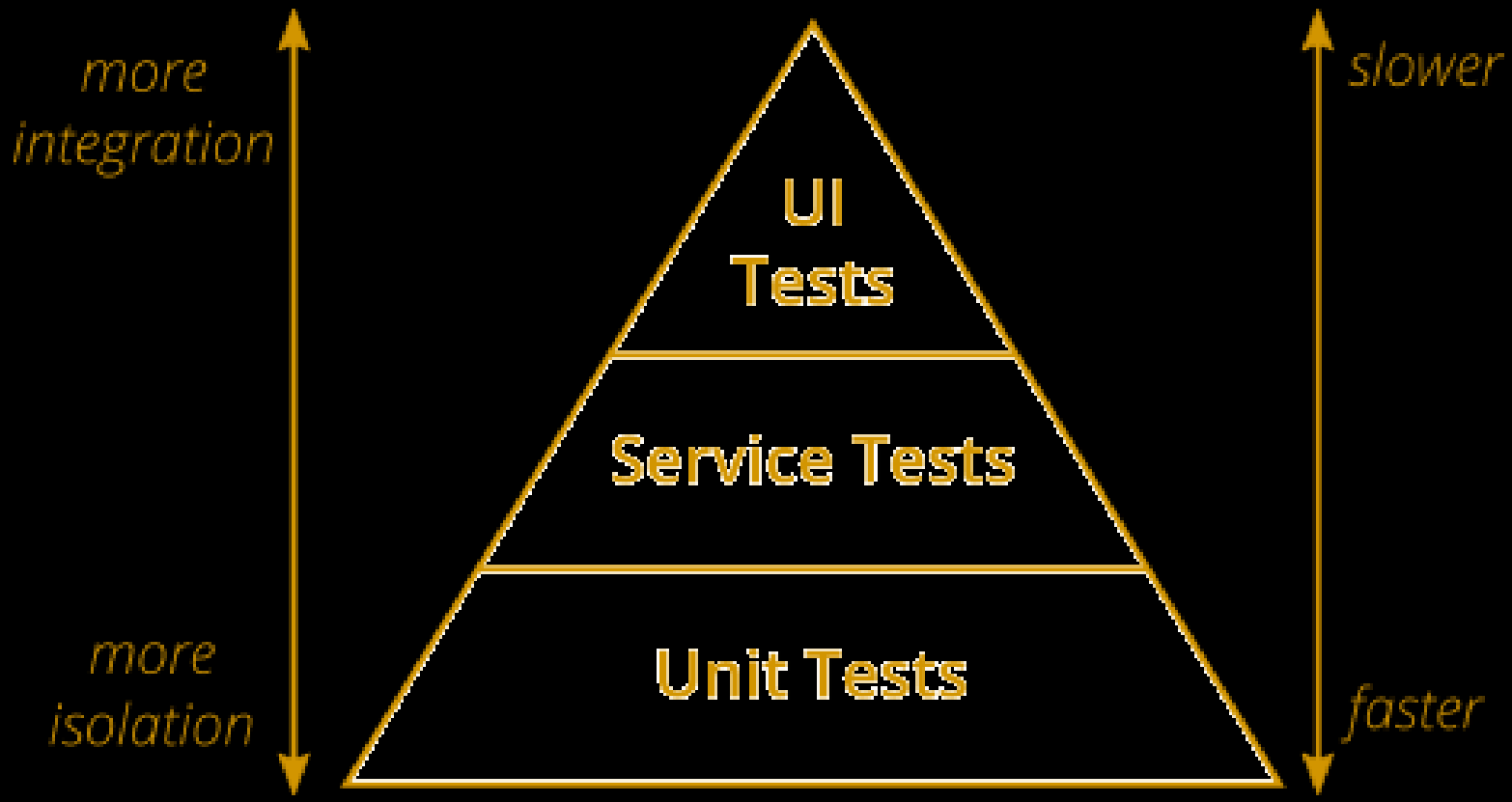
Валидация:

– Едет? – Не едет. Эффекта от того, что у нас есть – нет.

Важность автоматизации (тестов)

1. Сборка
2. Тестирование
3. Подготовка инфраструктуры
4. Развертывание
5. тестирование

Пирамида тестов



Оригинальная пирамида тестов Майка Кона

Пирамида тестов

два принципа:

Писать тесты разной детализации.

Чем выше уровень, тем меньше тестов.

инструменты и библиотеки

- JUnit: для запуска тестов
- Mockito: для зависимостей имитаций
- Wiremock: для заглушек внешних сервисов
- Pact: для написания CDC-тестов
- Selenium: для написания сквозных тестов UI
- REST-assured: для написания сквозных тестов REST API

Юнит-тесты

- Что такое юнит?

- Имитации (mocks) и заглушки (stubs)

Структура теста

- Настройка тестовых данных.
- Вызов тестируемого метода.
- Проверка, что возвращаются ожидаемые результаты.

Тесты времени исполнения

- Контроль входных параметров
- Проверка на ошибки при вызове системных функций
- Контроль времени исполнения

Интеграционные тесты

- Запуск базы данных.
- Подключение приложения к БД.
- Запуск функции в коде, которая записывает данные в БД.
- Проверка, что ожидаемые данные записаны в базу путём их чтения из БД.

Тесты совместимости

- Совместимость с операционной системой
- Совместимость с мобильными терминалами
- Совместимость с браузерами
- Совместимость с различными периферийными устройствами
  - Мониторы и проекторы
  - Графические ускорители
  - Подсистемы ввода вывода
  - ….

Контрактные тесты

- Команда потребителя пишет автоматизированные тесты со всеми ожиданиями со стороны потребителей.
- Они публикуют тесты для команды поставщика.
- Команда поставщика непрерывно запускает CDC-тесты и следит за ними.
- Команды немедленно вступают в переговоры, когда CDC-тесты ломаются.

Тесты UI

Тесты UI проверяют правильность работы пользовательского интерфейса приложения. Действия пользователя должны инициировать правильные события, данные должны представляться пользователю, состояние UI должно изменяться ожидаемым образом.

Цельность вёрстки веб-приложения.

Приёмочные тесты

Чем выше вы поднимаетесь в пирамиде тестов, тем больше вероятность возникновения вопросов

Тестирование производительности.

- Большое количество пользователей
- Частые обращения
- Обращения большого размера
- Одновременные обращения

- Замер времени отклика
- Замер пропускной способности
- Графики времени реакции

Тестирование безопасности

- проверка возможности доступа защищенных страниц без авторизации,
- проверка того, что все сессии заканчивают свою работу после прекращения пользовательской активности (закрытия приложения),
- проверка SSL-протокола приложения,
- проверка того, что файлы с ограниченным доступом невозможно загрузить без предварительной авторизации.

Тестирование безопасности

- Безопасная передача данных;
- Аутентификация;
- Управление сеансом;
- Авторизация;
- Криптография;
- Проверка данных;
- Отказ в обслуживании (Denial of Service);
- Специфические функциональные тесты;
- Обработка ошибок.

Особенности тестирования web-приложений

1. Технологические отличия.
   - Структурные отличия.
   - Отличия режимов работы.
   - Отличия формирования интерфейса.
2. Отличия работы с сетью.
3. Разница в количестве пользователей.
4. Особенности среды функционирования.
   - Отличия запуска и остановки, инсталляции и деинсталляции.

Заключение

- Тестовый код так же важен, как и код в продакшне. Уделите ему столько же заботы и внимания. «Это всего лишь тест» — не оправдание для небрежного кода.
- Проверяйте только одно условие в каждом тесте. Тогда тесты останутся лаконичными и понятными.
- Правило трёх A (arrange, act, assert) или триада «дано, когда, тогда» — хорошая мнемоника, чтобы поддерживать хорошую структуру тестов.
- Читаемость имеет значение. Не злоупотребляйте DRY (правило «не повторяйся»). Повторение хорошо, если улучшает читаемость. Попробуйте найти баланс между кодом DRY и DAMP (DAMP — Descriptive And Meaningful Phrases, содержательные и осмысленные фразы).