

Становление процесса автоматизированного тестирования в интернет-магазине О'КЕЙ



Кугач Анастасия
Рексофт 2016

Кратко о проекте

- Заказчик – О'КЕЙ
- Интернет-магазин
- Платформа IBM WebSphere Commerce



Цели и задачи

Цели:

- Уменьшение трудозатрат на регрессионное тестирование
- Оптимизация использования имеющихся ресурсов
- Уменьшение времени на релиз

Задачи:

- Разработка собственной утилиты сборки
- Поиск ресурсов для автоматизации тестирования
- Подготовка тестов

С чего начать?

- Ресурсы
 - Люди
 - Бюджет

Бюджет – обоснование необходимости автоматизации тестирования

- экономия времени на регрессионное тестирование
- сравнительно небольшая стоимость поддержки автотестов по сравнению с ручным тестированием
- окупаемость процесса автоматизации за предсказуемое время
- более строгий контроль качества разработки

С чего начать?

- Ресурсы
 - Люди
 - Бюджет
 - Тестовое окружение

Тестовое окружение – основные факторы

- железо (если требуется)
- инструменты автоматизации
- стоимость лицензий специализированного ПО и операционных систем

С чего начать?

- Ресурсы
 - Люди
 - Бюджет
 - Тестовое окружение

- Тесты
 - База регрессионных тестов

А кому и зачем это нужно?

- Команде тестирования
- Команде разработки
- Руководителю проекта
- Заказчику

ЗАЧЕМ?

Предсказуемое время для получения точного результата



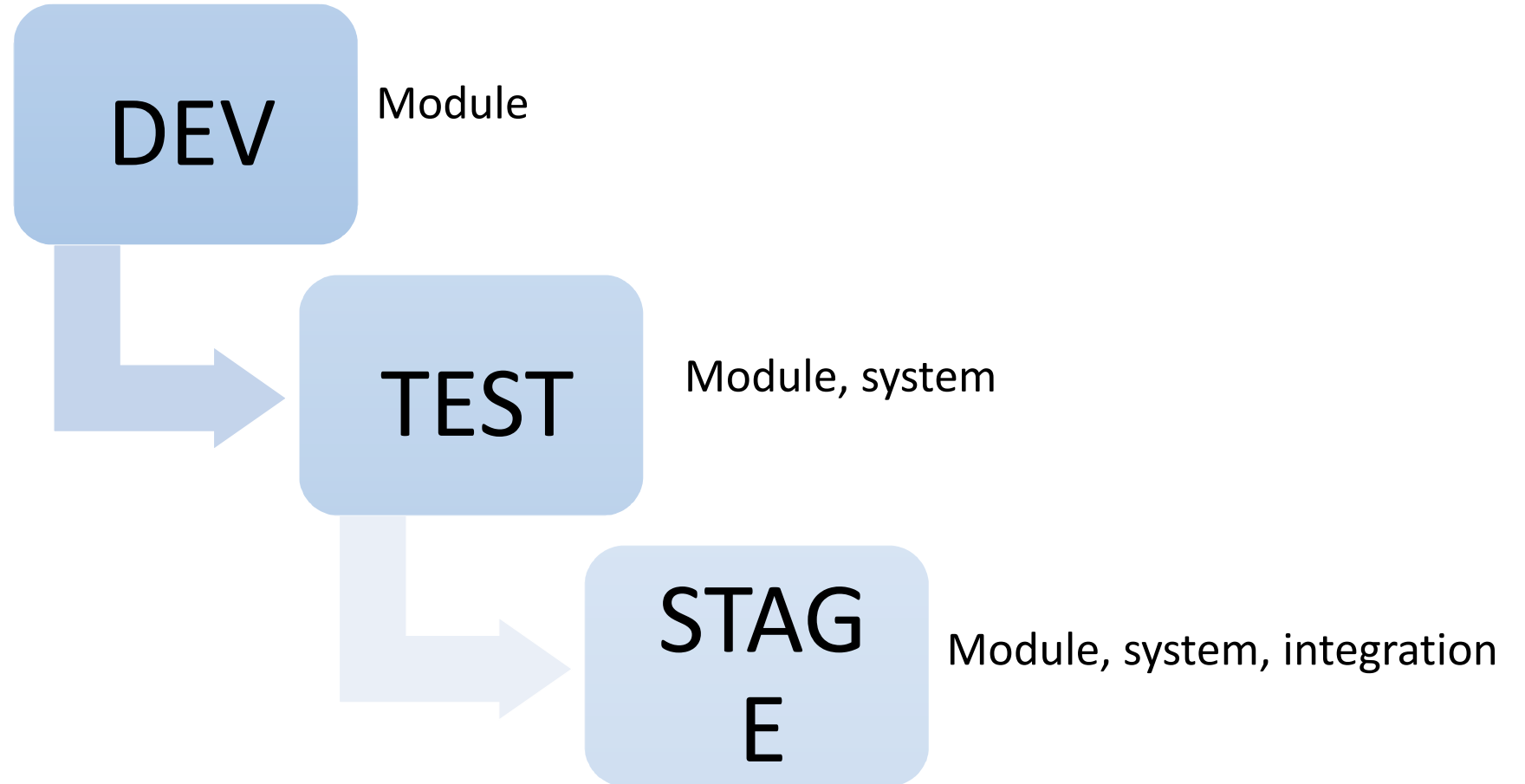
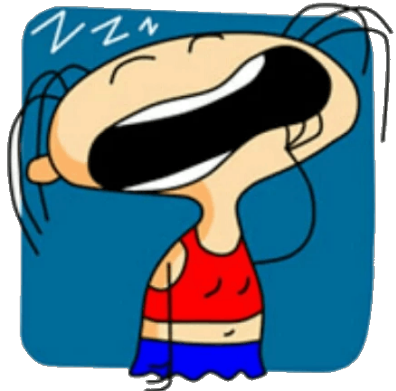
Технические особенности проекта, на которые нельзя закрыть глаза

- Платформа IBM WebSphere Commerce
- Не имеет своей системы сборки
- Различие окружений dev, test, stage, prod

Технические особенности проекта, на которые нельзя закрыть глаза

- Разные ветки разработки - разные окружения – разные конфигурационные файлы
- Различие в идентификаторах публикуемых магазинов (особенность WSC)
- Внутренние идентификаторы доступности товаров в SOLR – различается от окружений

Технические особенности проекта, на которые нельзя закрыть глаза



Выгода

- Исходные данные:
 - 6 чел/дней - **ручное** регрессионное тестирование версии (T_{manual})
 - 132 чел/дня – разработка утилиты сборки и автоматизация (T_{auto})
 - 20 – 23 новые версии в год (C)
 - Бесплатный инструмент Selenium/Selenium Grid
- Расчет: $T_{manual} * C > T_{auto} * C$
- Вывод: срок окупаемости автотестирования: 1-1,5 года

Масштабируемость

- Увеличение числа nodes, hubs, scripts
 - **Test script** - программа, использующая библиотеку Selenium для выполнения тестов;
 - **NODE** - запускает браузеры и выполняет действия запрограммированные в **Test script**'е;
 - **HUB** - выполняет роль сервера, который получает запросы от **Test script**'а и направляет их на выполнение в **NODE**'ы.
- Различные конфигурации тестовых систем (ОС + версия браузера)

Что мы имеем

Универсальный процесс

- Установка current PROD версии на тестовое окружение
- Сборка и установка изменений на тестовое окружение
- Загрузка тестовых данных (каталоги, цены, изображения, etc.)
- Запуск тестовых скриптов из Selenium Grid
- Сбор и публикация результатов

Время проведения - ночь

Выводы

- Высокая скорость проверки
- Удобное масштабирование окружений и поддержка тестов
- Сокращение трудозатрат на регресс
- 100 % результат после исполнения тестов
- Плюстик в карму 😊

Спасибо за внимание

Вопросы?



Анастасия Кугач

Старший инженер по тестированию

Рексофт