

Беклог потомкам

Составил Апанякин Анатолий 2019-2020

- Стандартизация
 - Унифицировать ведение страниц тестеров (Приоритет - 1)
 - Разработать и масштабировать требованию по покрытию (Приоритет - 1)
 - Привести проекты к одной структуре папок и особенностей фреймворка (Приоритет - 2)
 - Определить DOD для всех АТ спецов. На выходе Jira задачи обязателен чеклист (Приоритет - 0)
 - Разработать и согласовать стандарт логгирования бизнес-событий в логах микросервисов (Приоритет - 3)
 - Научить команды заранее планировать и снимать метрики на бою (Приоритет - 3)
 - Соответствие пирамиде тестирования (автоконтроль)
 - Согласовать с саппортом приемку Автотестов (отчета по тестам на прелайве) (Приоритет - 4)
 - Общаться со специалистами синхронизируясь по звездной карте и проводить аессмент (Приоритет - 1)
 - Выработать стандарт покрытия юнит-тестами в разрезе front-mid-back совместно с их ЦК (Приоритет - ?)
 - Другое.
- Коммуникация
 - Школа тестировщиков
 - Набор и обучение джунов
 - Презентации внутри или внешне, публикация в опен сурс - как "акита"
 - Публикация сендбокс в опен сурс.
 - Хакатон тестеров или студентов
- Общее
 - Тимбилдинг
 - Встречи внутри направлений.
 - Проводить крутые опросы для выявления решений и лидеров ("как вы тестируете рест интеграцию")
 - Сделать резы автотестов прозрачнее для команды
 - Согласовать между ИТ ЦК и Бизнесом принцип "end to end responsibility"
 - Замерять и ускорять поставки
 - Масштабирование example mapping (+ 3 amigos)
 - Собрать в одном месте список/каталог целей.
 - Провести аудит-ревью аналитики.
- Техническое
 - Создание облачной платформы для написания тестов прямо в браузере
 - Разработка решения для полной автоматизации в POSTMAN
 - Докеризация тестирования
 - Свое решение по сбору отчетности
 - Провести аудит репозиториев на предмет структуры
 - Поднять культуру использования заглушек
 - Замутить коллагу с RPGшниками
 - Настроить автосборщик плана раскатки из Jenkins
 - Масштабировать SANDBOX-АТ-*
 - Всех тестиров обязательно отправлять на обучение ЕКУ. так как автоматизаторы нихера не шарят в нем

Что сделать в каком количестве. внутри объяснение зачем.

Стандартизация

Унифицировать ведение страниц тестеров (Приоритет - 1)

Проблема. заходя в проекты в конфе, трудно разобраться где лежат тестовые данные, какой состояния тестов, куда развивается проект, какое покрытие, кому обращаться. Желательно поставить КРП на соблюдение "гигиены" в этом разделе
Пример. Тестирование ИЗК

Разработать и масштабировать требованию по покрытию (Приоритет - 1)

Проблема. Специалисты АТ приходят со знаниями программирования, а знания систем и подходов к покрытию отсутствуют.

- тестируют GUI e2e, утопают в длинных/flaky тестах
- не тестируют ари
- тестируют ари. однако ари можно тестить по разному. кто то валидирует схему, кто то проверяет параметры ответов. кто то ловит логи..

Шаблон версии для требований: [Требования к покрытию.pdf](#)

Привести проекты к одной структуре папок и особенностей фреймворка (Приоритет - 2)

Проблема. Проекты на Ruby выглядят слишком по разному. нет стандарта использования библиотек. нет правил для код-ревью. нет соблюдаются SOLID/KISS принципы.

Первая попытка сделана. создан набор шаблонных проектов [Общие практики \(sandbox-at\)](#) - (10+ тестеров пишут по аналогичной схеме. + есть наработки в SFA)

+ провести аудит по проектам. кто как пишет.

Определить DOD для всех AT спецов. На выходе Jira задачи обязаны иметь чеклист (Приоритет - 0)

Проблема. Спецы попавшие в команду выпадают из вида, начинают жить по разным workflow команды. В итоге как происходит ручное тестирование не прозрачно ни руководству, ни саппорту, ни команде.

+ = пополнение тестовой модели.

Разработать и согласовать стандарт логгирования бизнес-событий в логах микросервисов (Приоритет - 3)

Проблема #1. Когда ловим ошибку на бою. слишком тяжело ее локализовать из-за неинформативных логов

Проблема #2. Когда тестируем приложение руками или полу-автоматически уходит слишком времени из за неинформативности логов.

П.С. Если в логах будут прописаны подробные флаги бизнес событий, даже ручники смогут делать вызовы-кучей, и после просто валидировать появление нужных флагов в логах.

Научить команды заранее планировать и снимать метрики на бою (Приоритет - 3)

Проблема #1. Подходы девопса или эджайла подразумевают что все что мы делаем обвешивается метриками по пользователям. "нравится ли им", "пользуются ли".

Проблема #2. Саппорт должен мониторить количество запросов на ваше приложение в моменте, плюс мониторить непоявление ошибок.

Соответствие пирамиде тестирования (автоконтроль)

Согласовать с саппортом приемку Автотестов (отчета по тестам на прелайве) (Приоритет - 4)

Проблема. Много стендов. Нет прозрачности. Нет ответственности.

Стоит попробовать сделать процесс приемки результатов тестирования обязательным, а не просто DOD

Общаться со специалистами синхронизируясь по звездной карте и проводить ассессмент (Приоритет - 1)

Проблема. Все спецы разного уровня. нет синхронизации. Сложно отслеживать рост спецов. нет идей о чем с ними разговаривать на 1-1

Пример общения со спецами Зп стрим 2019/2020: [Потребность в специалистах.xlsx](#)

АКТИВНОСТИ СОТРУДНИКА	Швецов	Ширяев	Чурсов	Горшков	Городенков	Исаев
поддержка старых сценариев	+	-	+	+	+	+
написание новых	+	+	+	+	+	+
нагрузочное тестирование при каждом внедрении / разовом?	-	-	-	-	+	-
ручное тестирование фич	+	+	+	+	-	+
Выявление неисправных компонентов продукта	+	+	+	+	+	+
Заведение задач	+	+	+	+	+	+
пополнение модели если есть	+	-	-	-	+	-
Внедрение ЕФ	+	+	+	-	-	-
Внедрение АЗОН	-	-	-	-	-	-
Внедрение ЕО	-	-	-	-	-	-
Внедрение WS	-	-	-	-	-	-
Внедрение CLM/ОС	-	-	-	-	-	-
заявки	-	-	-	-	-	-
Свой Job в Jenkins	+	-	-	-	-	-
настройка - балансировка / devops / settings	+	-	-	-	+(под НТ)	+
рефакторинг тесткода	+	+	+	+	+	+
наставничество	+	Хочу	-	Фронты	-	-
поддержка стендов	+	-	+	-	-	+
метрики-мониторинга, для контроля ошибок в бою	-	-	-	-	-	-
метрики-бизнеса, для подтверждения гипотез	-	-	-	+	-	-
участие в реализации архитектуры и логики приложения	+	+	+	+	-	+
Ревью тестеров	+	-	-	-	-	-
Ревью разработ	-	-	+	-	-	-
Тестирование ЕФ	+	+	+	2 пит.реп	3 пит.реп	5 пит.реп
Тестирование WS	+	+	2	+	+	-
Тестирование LEGACY_AZON	-	-	-	+	-	-
Тестирование EQ	-	-	-	+	-	-
Тестирование CLM/ОС	-	-	+	-	-	-
РФЛ	БТР	АльфаЗавры	Заявка на ЗП	НАЧИСЛЕНИЯ, КОР	ОНБОРДИНГ	

Выработать стандарт покрытия юнит-тестами в разрезе front-mid-back совместно с их ЦК (Приоритет - ?)

Проблема. Разработчики варятся в своих проблемах, тесты пишут без особых знаний. Хотя тестов на их уровне должно быть в разы больше чем везде.

Желательно научиться трассировать юниты на требования.

Другое.

развивайте. я устал

стандарт. упрощение входа

стандарт. упрощение кода

стандарт. код во всех репах одинаков

стандарт. тестирование на проекте

Зефира нет. Разобранные способы ведение тест доки в альфе. [тест кейсы азон.pdf](#)

Разобранный процесс работы тестировщика в альфабанке. [Работа тестировщика..pdf](#)

Коммуникация

Школа тестировщиков

Проблема. На рынке гуляют узкоспециализированные мидлы, времени на введение в дела банка уходит много. рубистов мало. джависты дорогие. Теорию тестирования (на практике) применяют на 5%.

Вывод. Надо организовать Пайплайн (CJM/Воронку) для выращивания специалистов.

Набор и обучение джуунов

Тесно связано со школой тестировщиков. Учить все равно надо. только тут сразу на проектах. а в школе это отдельное месячное обучение.

Презентации внутри или внешне, публикация в опен сорс - как "акита"

Проблема. мало PR для альфаIT. у меня вся лента в соцсетях в "рекламе" посмотрите как тестируют у нас, приходите на конференцию.

Публикация сендбокс в опен сорс.

Создание бренда. мотивация спецов заработать очки в пространстве WWW

Хакатон тестеров или студентов

Java и JS Айфоны разыгрывают на конференциях.

- можно соревноваться кто больше багов найдет на вымышленном сайте
- можно давать задание накодить автотестов на скорость.
- можно ..? any ideas?

Общее

Тимбилдинг

Боулинг и только боулинг. только вопрос: "Где деньги лебовски?"



Встречи внутри направлений.

Поможет избежать закрытости. Главное без микроменеджмента.

Проводить крутые опросы для выявления решений и лидеров ("как вы тестируете рест интеграцию")

Проблема. Я пишу вопрос в чат на 100 чел. высказывается один. остальные видят что ответ есть (даже если он расходится с мнением) и игнорят его.

Есть идея. выделить несколько тем которые интересуют нас или спецов. и делать условно-анонимный опрос. Ответы опубликовываются на всех на следующий день.

вместо того чтобы получить один ответ от умника получаем 5 ответов. Анализируем их одновременно и не предвзято.

Сделать резы автотестов прозрачнее для команды

Проблема. Раз 15 уже слышал "мы не понимаем что делает автотестер"

Надо бороться с этим. повышать прозрачность

Согласовать между ИТ ЦК и Бизнесом принцип "end to end responsibility"

Без него вся разработка будет медленная и сложная

Замерять и ускорять поставки

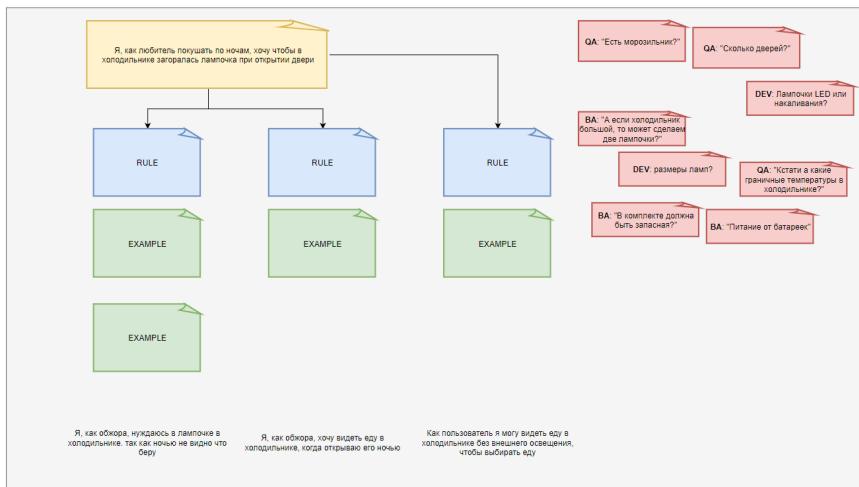
Проблема. Команды копят задачи в поставках. Регресс растет. Тестирование удлиняется и усложняется.

Цель. Мотивировать команды уменьшать поставки. тем самым будет легче тестить, точнее планировать и т.п

Масштабирование example mapping (+ 3 amigos)

Рекомендуется как упражнение для тестеров и для всей команды.

Три-амиго (Example Mapping).pdf



Собрать в одном месте список/каталог целей.

Поможет спецам и их лицам быстрее ориентироваться в постановке. кому то книжку прочитать, кому то в менеджерство пойти, кому то ядро линух изучить. главное чтоб банку было полезно.

Провести аудит-ревью аналитики.

Сделать выводы. если окажется что вся аналитика написана по-разному. не читаемо, тяжело преобразуемо в тесты. то тогда обсудить с ЦК А что надо что то менять и приводить к стандарту.

Техническое

Создание облачной платформы для написания тестов прямо в браузере

Проблема. Я ленивый тестер. Я редактирую код прям в битбакете. А еще я хочу скачать чужой проект и потыкать его а там другая версия руби или другие библиотеки.

А еще я не хочу быть привязанным к рабочему месту и к окружению

Для всего этого всего существует VS CODE в докере. Я пересобрал его для ruby python js и java.

<https://hub.docker.com/r/hoochiecoo/vscode-ruby-ide>

<https://hub.docker.com/r/hoochiecoo/vscode-python3-ide>

<https://hub.docker.com/r/hoochiecoo/vscode-nodejs-ide>

<https://hub.docker.com/r/hoochiecoo/vscode-java-ide>

Берешь поднимаешь его хоть на мезосе хоть на кубернетесе хоть на acloud/ и дебажишь код прям в браузере.

ЛЯПОТА.

Разработка решения для полной автоматизации в POSTMAN

Проблема. postman инструмент доступны ФТшникам, в нем есть автоматизация но она основана только на HTTP запросах.

1. можно ждать когда они прикрутят плагины позволяющие лазить в БДшки или еще куда.
 2. А можно самим написать мини адаптеры которые можно запихнуть в докер и делать простые проверки вызовов-апи- проверка в скеле. либо получать данные со скеля и вызывать апи и проверять ответ.
- В таком варианте ФТшники становятся АФТшниками без особых знаний кода.

Докеризация тестирования

У меня нет аргумента я устал писать беклог. Докеризация хорошо удобно няшко.

Свое решение по сбору отчетности

UI - графана

DB - mysql mongo influx

Collector - скрипт в конце прогона, слушать трафик браузеров.

П.С. Удобно тем, что не надо особо кодить. отправляй резы в базу. Дашборд собирается на графане.

Провести аудит репозиториев на предмет структуры

Проблема. Все собирают фреймворки по разному. Разные либы, разные способы извращений с кукумбером (это прям отдельную статью можно написать)

На основе полученных данных генерируем: sandbox, общие практики, новые решения.

Поднять культуру использования заглушек

Wiremock in docker. - тут еще и прокся внутри можно пропускать часть запросов на шину, а часть отвечать заглушкой. Apache tomcat.

Знают единицы. рекомендаций мало.

Замутить коллабу с RPGшниками

Коршики устроили революцию. Теперь у них есть свой jenkins с cucumber тестами которые лежат рядом с RPG кодом. Прям вызывают call pgm и проверяют таблички.

Настроить автосборщик плана раскатки из Jenkins

UFR Jenkins Pipeline мощный движок. Можно упростить процесс внедрения и научить дженкинс создавать правильный отчет для внедрения. избавить от рутины тестеров и аналитиков.

Масштабировать SANDBOX-AT-*

Цель. Выделить из git-репозиториев **популярные и простые** подходы к тестированию, для дальнейшего **переиспользования** но вичками и унификации подходов

Подход sandbox-* **подразумевает**

- Что в проекте. содержитя **минимальное** число гемов/файлов/кукумбер-шагов.
 - С помощью одной рецы **решается** способ работы с **одной** сущностью (протокол/БД/какой-то подход)
 - Нет связки на **реальные данные** в рамках шагов, только шаги работы с сущностью.
- Файлк с шагами можно взять и просто скопировать в любой другой проект и сразу начать пользоваться. (В папку **base_step s/**)

На выходе. для своего проекта можно собрать базовый набор. например: rest+sql, UI+rest, UI+mongo, ws+kafka.

82.01 sandbox-at-test	Базовый проект для быстрого построения архитектурных тестов
82.02 sandbox-at-sql	Базовый проект для быстрого построения sql-тестов ... DBизолированных
82.03 sandbox-at-net	Базовый проект для генерирования WSDL-Справок
82.04 sandbox-at-UI-only-gherkin	Базовый проект для аналитиков и диктов. Быстро начать писать работающие тесты
82.05 sandbox-at-UI-page-object	Базовый проект для page-object-pattern с примерами
82.06 sandbox-at-UI-сравнение	
gem_chayns	(SharedDB) темы для различения общих конфигурационных данных для нескольких проек...
gem_pubsub_rack	клиент для интеграции в Rack через проект сервер - pubsubhubbub
sandbox-at-ALL-IN	Базовый проект где есть все - > 1000+ половинчи
sandbox-at-архив_зап	для генерирования на архиве
sandbox-at-as400	TODO: add cucumber-like syntax (Eukache)
sandbox-at-автомат	TODO: дополнить проект include matrix между java/python/из проекта с тестами по правилам
sandbox-at-ff	TODO:DELETE: USE AT-FF. Вспомогательный проект для быстрого построения тестов на FF
sandbox-at-изображ	Сравнение изображений
sandbox-at-яким	для прорыва отчёта в движущее
sandbox-at-кафа	TODD: Базовый проект для быстрого построения тестов СКАФА
sandbox-at-шар	
sandbox-at-mongo	Подключение к MongoDB и создание запросов к ней
sandbox-at-mongodb_yml	TODO:Работы: открыть таблицы по Yaml
sandbox-at-sharek-yml	В данном листинге описан пример создания и выпада в отдельный проект каких-то общих на...
sandbox-at-стаб-маппинг	TODO:Работы: залогиниться в стартере базовый проект с настроенным профилем залогиненого д...
sandbox-at-ленинг	example of usage http://confluence.mesmerizingsoft.net/display/UF44L/UTestKit#
зарегистрироваться	подключениико удаленному браузеру, кликните, добавлен через VNC

Всех тестиров обязателю отправлять на обучение ЕКУ. так как автоматизаторы нихера не шарят в нем

Файлы.

Файл	Изменено
PDF Файл Требования к покрытию.pdf	27.10.2020 by U_M0WQ9
Microsoft Excel таблицы Потребность в специалистах.xlsx	27.10.2020 by U_M0WQ9
PNG-файл image2020-10-27_20-54-42.png	27.10.2020 by U_M0WQ9
PDF Файл Работа тестировщика..pdf	27.10.2020 by U_M0WQ9
файл JPEG Story-Splitting-Flowchart-RUS.jpg	27.10.2020 by U_M0WQ9
PDF Файл тест кейсы азон.pdf	27.10.2020 by U_M0WQ9
PNG-файл image2020-10-27_21-59-2.png	27.10.2020 by U_M0WQ9
PNG-файл image2020-10-27_22-27-31.png	27.10.2020 by U_M0WQ9
PDF Файл Три-амиго (Example Mapping).pdf	27.10.2020 by U_M0WQ9
файл JPEG 12_STRAIKOV.JPG	28.10.2020 by U_M0WQ9

Перетащите для загрузки или [обзор файлов](#)

 [Скачать Все](#)