



Внедрение практики авто-тестирования FAST-Авто

Подход к реализации

• Предварительная подготовка

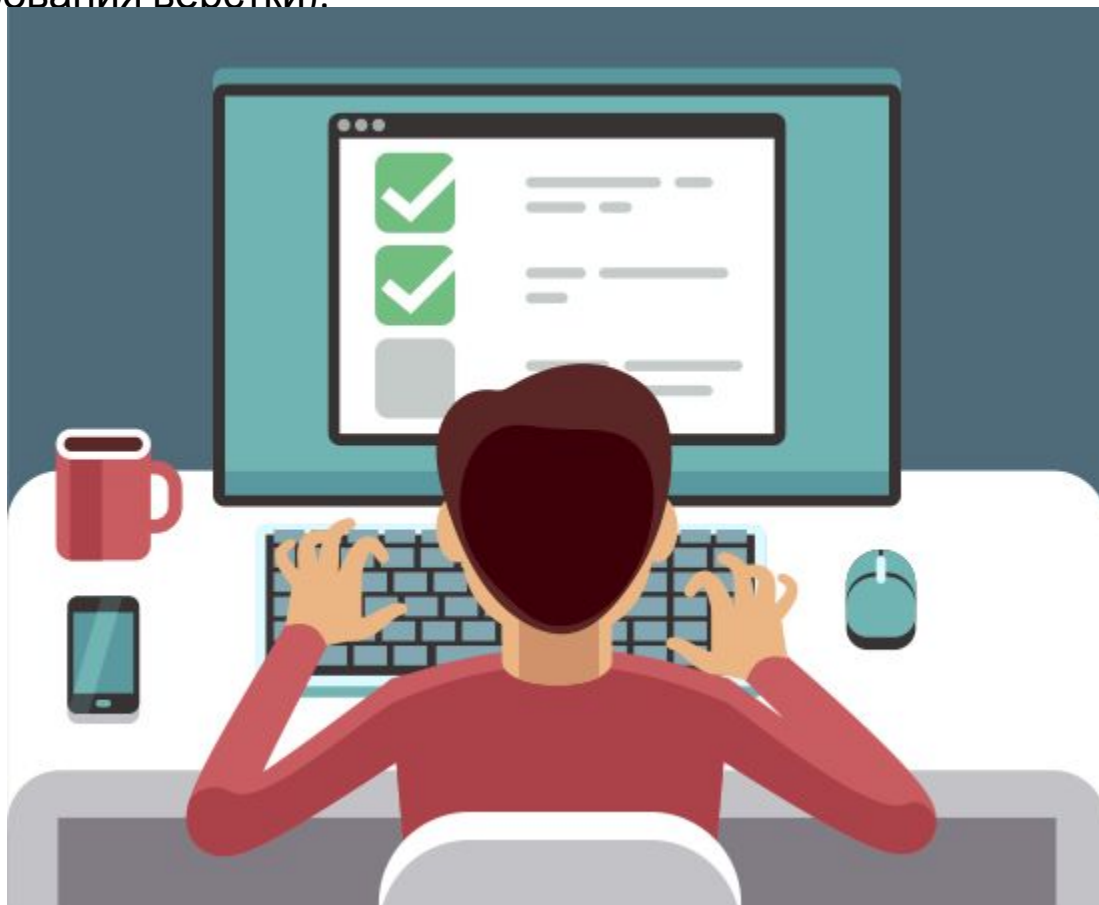
- Выбор команды + включая подбор курсов повышения квалификации
- Аналитика системы и процессов
- Согласование плана работ
- Подготовка инфраструктуры
- Разработка
- Анализ ошибок
- Разработка 😊
- Опытная эксплуатация и внедрение

Риск

- И Смена приоритетов компании (специалисты, вовлеченные в авто-тестирование, являются так же ручными тестировщиками, и могут привлекаться к другим проектам/задачам)

Преимущества внедрения

- **Улучшение качества процесса тестирования.** Достигается посредством исключения человеческого фактора, расширения набора тестовых данных и платформ, на которых производится UI тестирование (при единоразовом написании теста, мы можем его повторно воспроизводить в любых браузерах и в любом разрешении, для тестирования вёрстки).



Преимущества внедрения

- **Сокращение времени тестирования.** Скорость прохождения авто-тестов в сотни раз выше ручного тестирования.
- **Экономия человеческого ресурса.** Авто-тесты освобождают специалистов тестирования от рутины, в частности при наличии ежедневных релизов, сокращая затраты на регрессионное тестирование.

Сокращение времени тестирования на примере регресса типового продукта КАСКО:

Кол-во тестов для выполнения задачи : **25 кейсов**

Время ручного выполнения теста (среднее) : **5 минут**

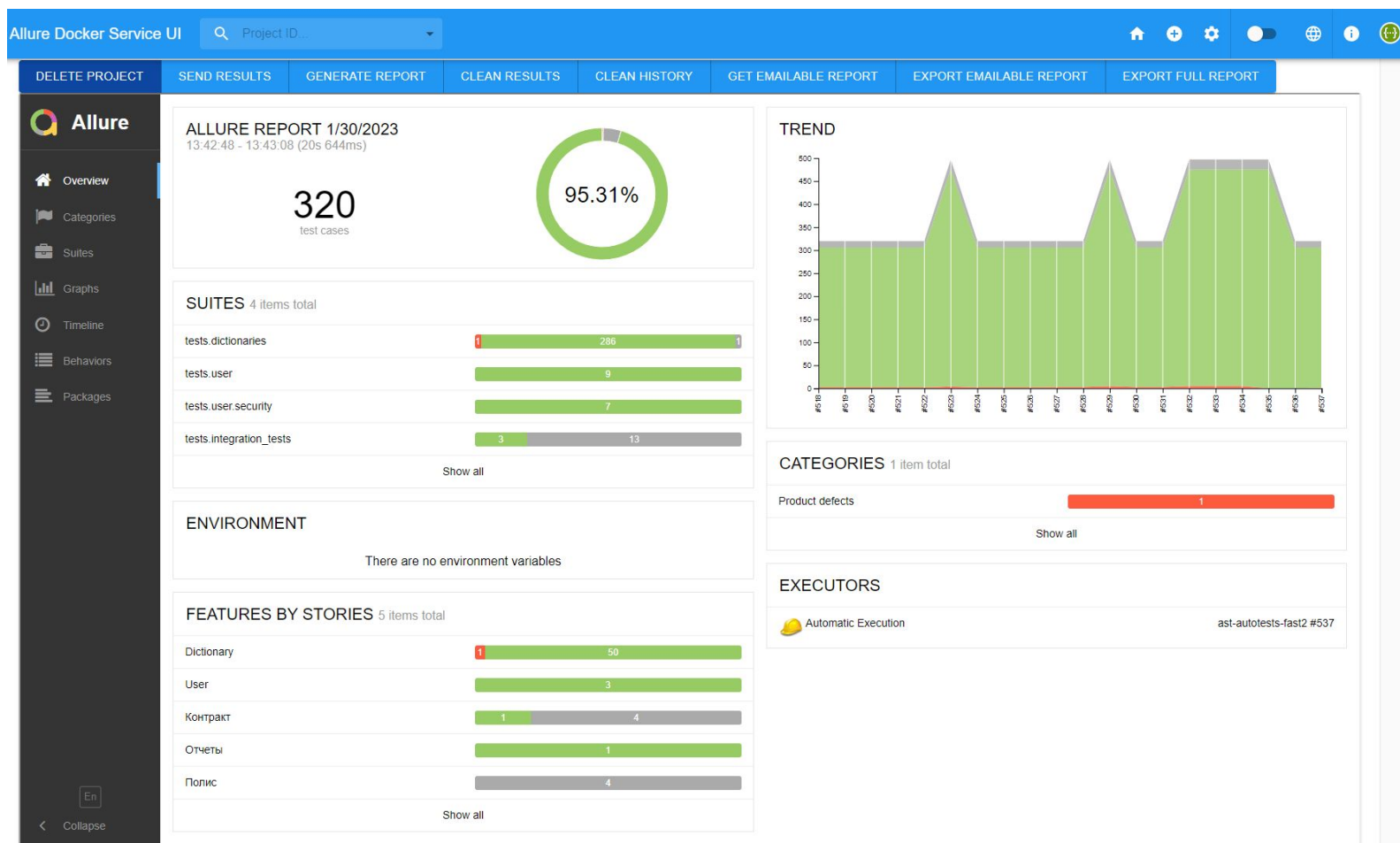
Общее время ручного тестирования : **125 минут**

Время выполнения одного авто-теста (среднее): **0.5 минут**

Общее время автоматического тестирования : **12,5 минут**

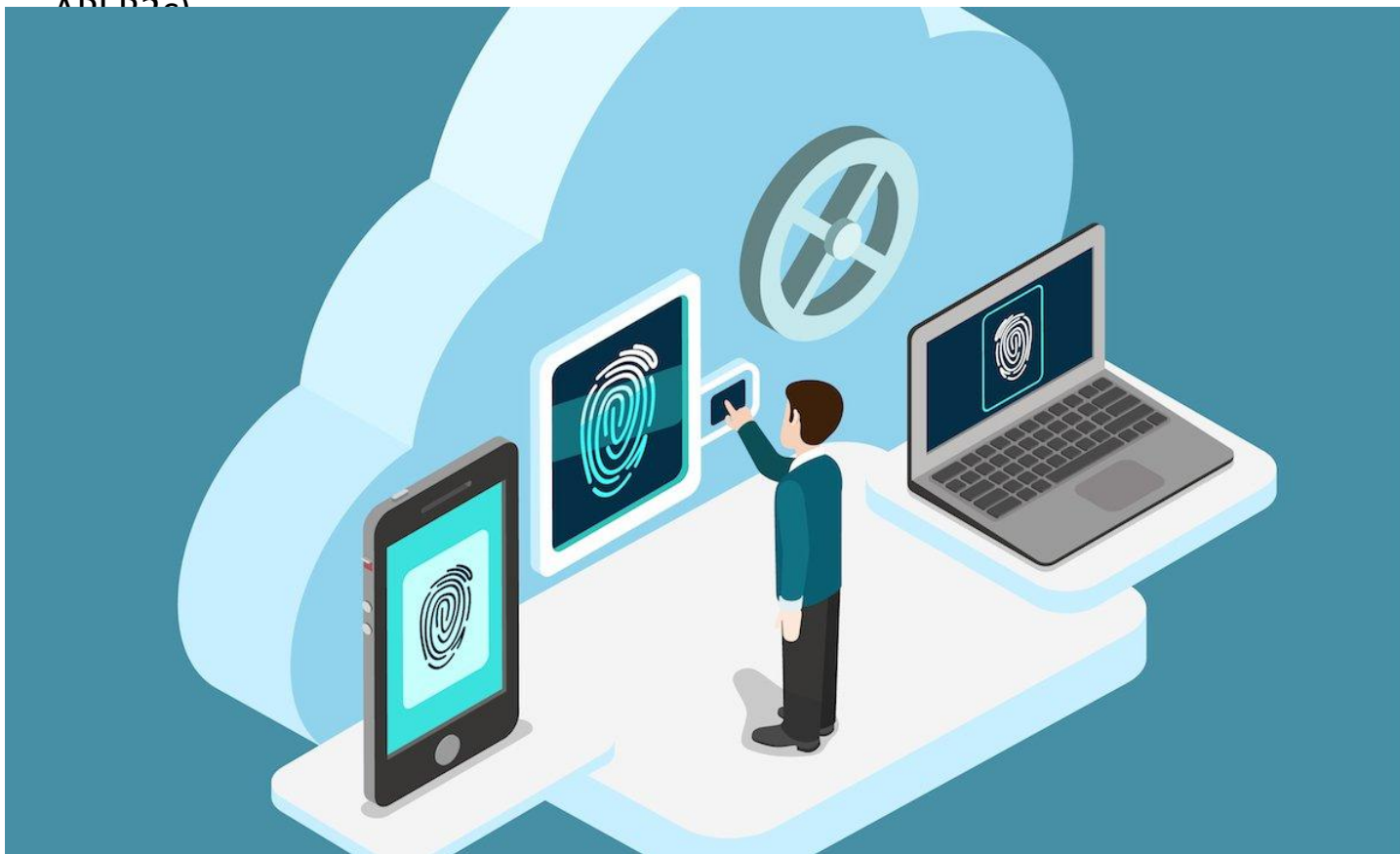
Преимущества внедрения

- **Доступная отчетность.** По итогам тестирования автоматически генерируется визуальный отчет, отражающий текущее состояние системы и историю прохождения тестов, позволяя отслеживать тренды.



Преимущества внедрения

- **Дополнительные возможности.** Авто-тесты помогают в организации тестирования безопасности проекта (доступность методов и справочных значений API в разрезе нетиповых действий пользователя, уровня доступа пользователя и тд; позволяют организовать эффективный процесс тестирования систем без интерфейса (прим. API для дилеров, API B2B)





FAST2		Виджеты	API V1	API B2C
Backend	Frontend			
- Невозможность проводить полный регресс системы после каждого релиза - Огромное количество-тест кейсов - Дефицит специалистов внутри команды - Высокая стоимость специалистов	- Отсутствие возможности проводить полный регресс системы после каждого релиза - Медленный ручной труд в разрезе количества тест кейсов и задач - Наличие человеческого фактора в разрезе количества кейсов и задач	- Напрямую зависят от релизов fast - Не учитываются в разработке той или иной фичи fast - Не тестируются после релизов (баги ловят реальные пользователи) - Часто ломаются - Типовой монотонный регресс всех виджетов = большой ресурс ручного тестировщика	- Напрямую зависят от релизов fast - Не учитываются в разработке той или иной фичи fast - Не тестируются после релизов (баги ловят реальные пользователи) - Проект не имеет интерфейса для ручного тестирования (требуется высококвалифицированные специалисты) - Для тестирования требуется знание специфического ПО - Дефицит специалистов внутри команды - Высокая стоимость специалистов	- Проект не имеет интерфейса для ручного тестирования (требуется высококвалифицированные специалисты) - Огромное количество-тест кейсов - Медленный ручной труд в разрезе количества тест кейсов - Для тестирования требуется знание специфического ПО - Дефицит специалистов внутри команды - Высокая стоимость специалистов

Язык программирования - **Python**

Библиотеки:

- **Pytest** – тест runner
- **Requests** – API тесты
- **Playwright** – UI тесты
- **Allure** – формирование отчетности
- **TMS** (Test Management System) –система управления наборами тестов, тестовыми примерами, планами тестирования, тестовыми запусками и документацией

Планируется внедрить:

Schemathesis - тестирование на основе свойств для схем API

Swagger coverage py. - анализ покрытия API-тестами на основе OAS (Swagger).

Первичное внедрение в API B2C

- Произведены попытки автоматизации тестирования средствами Postman (сложно, неудобно, в перспективе – дорого)
- Разработан фреймворк, упрощающий написание тестов и проведение проверок;
- Разработана эмуляция расчета Альфы ОСАГО для повышения тестируемости системы;
- Разработаны E2E тесты;
- Внедрено построение отчетов (Allure)
- Покрыты авто-тестами 100% методов





FAST API

Что сделано (4 квартал 2022)

- Разработан фреймворк, упрощающий написание тестов и проведение проверок;
- Покрыты авто-тестами ~30% методов (около 350 тестов)
- Тесты интегрированы в CI/CD
- Внедрено построение отчетов (Allure)

Что предстоит сделать (2-4 квартал 2023)

- Составить высокоприоритетные тест-кейсы
- 100% покрытие методов авто-тестами
- Тестирование Swagger'a
- При наличии ресурса– составить и автоматизировать менее приоритетные тест-кейсы

API V1

Что сделано (4 квартал 2022)

- Составлен тест-план и высокоприоритетные тест-кейсы
- Разработан отдельный фреймворк, упрощающий написание тестов и проведение проверок
- Покрыто авто-тестами ~10% методов
- Внедрено построение отчетов (Allure)
- Внедрена TMS, добавлены тест-кейсы в отчетность Allure

Что предстоит сделать (1 квартал 2023)

- Покрыть авто-тестами ~40% методов
- Разработать систему хранения тестовых данных и генерация запросов по ним
- Закончить генерацию высокоприоритетных тест-кейсов
- Составить тест-кейсы популярных продуктов
- Составить интеграционные тест-кейсы
- Автоматизировать описанные выше тест-кейсы
- При наличии ресурса– составить и

FAST UI

Что сделано (4 квартал 2022)

- Определен стек
- Определены области применения (виджеты + фронт система)

Что предстоит сделать (2-4 квартал 2023)

- Составление недостающих и актуализация имеющихся тест-кейсов
- Определить объем UI тестов
- Внедрить Смоук тесты
- Тестирование популярных продуктов
- Выявление популярных сценариев и реализация по ним E2E тестов

Команда проекта



Роль	Сотрудник
Руководитель проекта	Верещагин Сергей
Team Lead	Щербаков Павел
QA-автоматизатор	Королёва Наталья
Devops – разработчик (частично)	Новиков Максим



Backlog

- Определить регламент актуализации тест-кейсов
- Внедрение практики автоматизации тестов по итогам реализации задач
- Внедрение практики проведения регрессионного тестирования релиз кандидатов и коррекция CI/CD
- Внедрение использования TMS в процессы ручного тестирования
- Покрытие системы юзкейсами