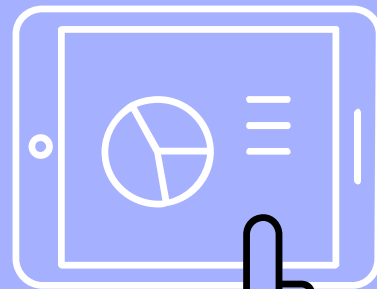
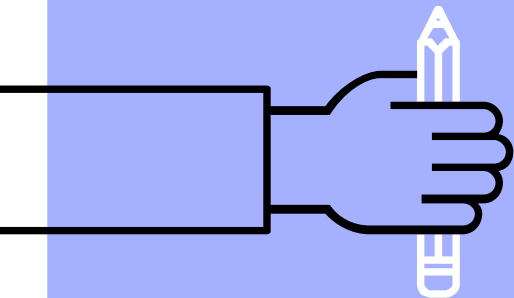
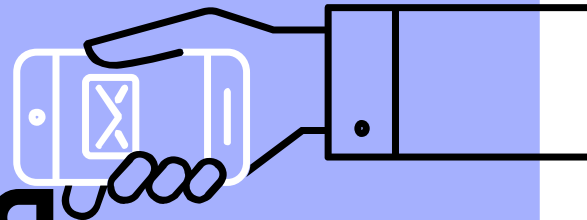
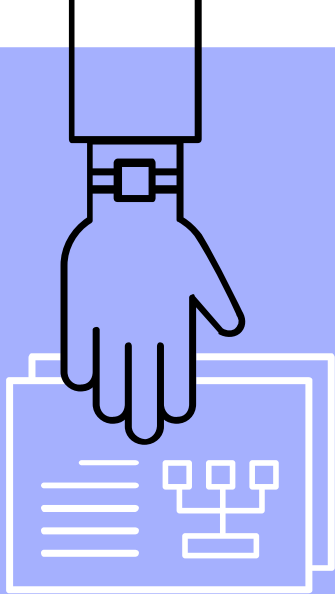


# ОРГАНИЗАЦИЯ ПРОЦЕССА ТЕСТИРОВАНИЯ С НУЛЯ



# ПРИВЕТ!

Меня зовут Артем!

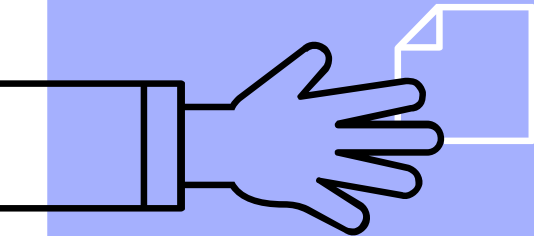
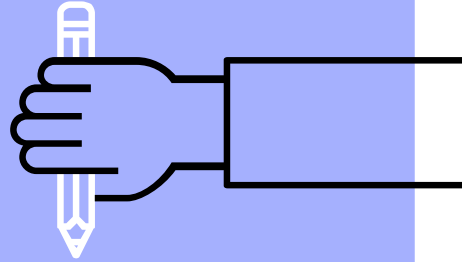
6 лет работаю в области QA,  
2 – в качестве тестировщика.

Веду Youtube-канал по теме  
QA (60K+ subs)



# ОСНОВНЫЕ ЭТАПЫ

Анализ > внедрение >  
рутина



“

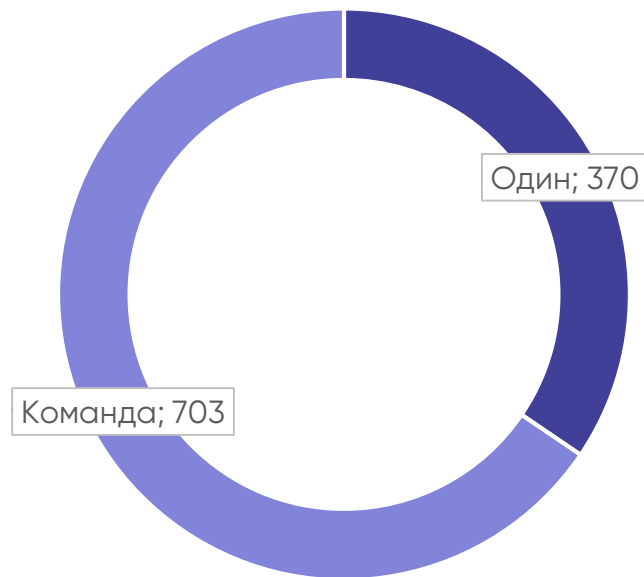
*Почему я должен  
делать вид, что мне  
хорошо, когда мне в  
действительности  
плохо?*



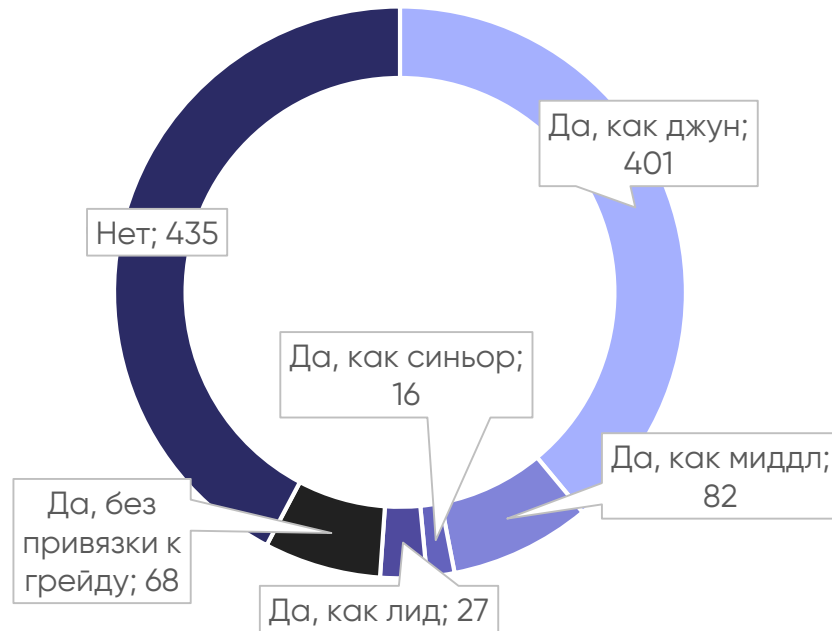
(с) Бараш



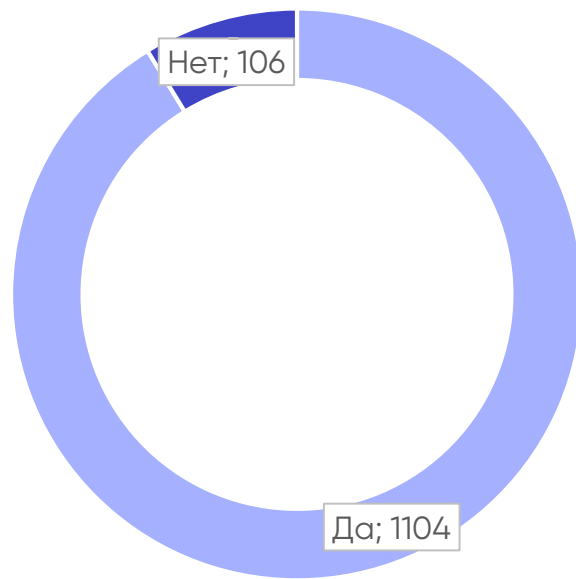
Вы начинали работу в качестве одного  
тестировщика?



Вы участвовали в  
налаживании процессов?



Вам было бы интересно налаживать процессы?



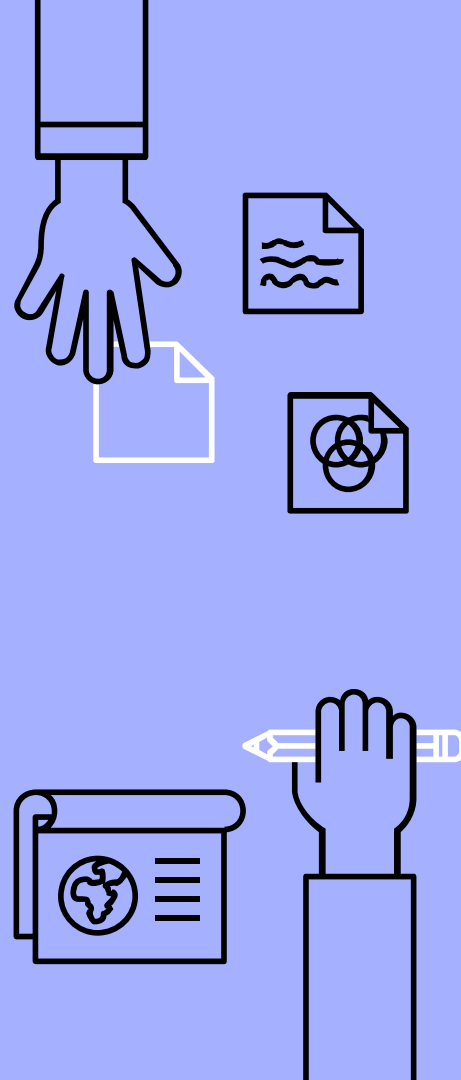
# КОГДА ВАМ НУЖНО ПОДУМАТЬ О ПРОЦЕССАХ?

## **Проект начинается с нуля**

Частый кейс, при котором вас определяют на новый проект и ставят задачу настроить «мифические» процессы тестирования

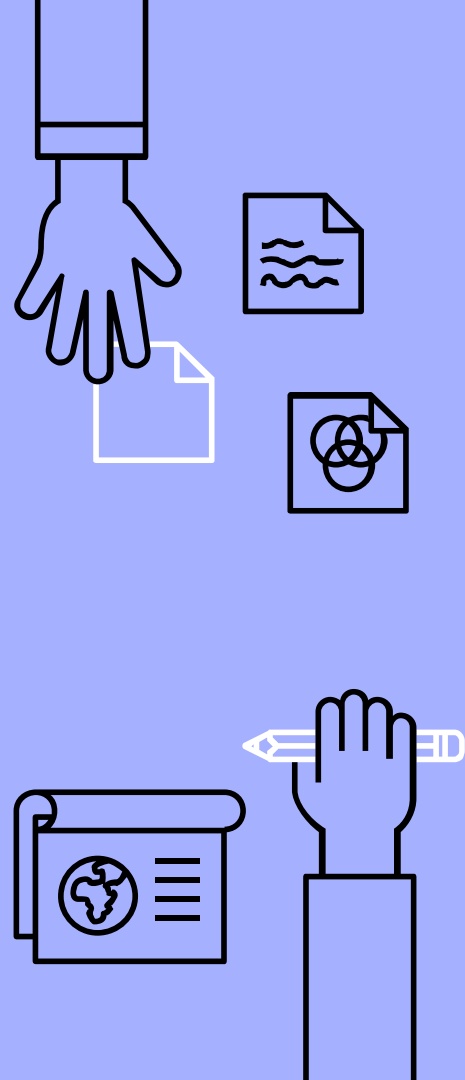
## **Проект существует давно**

Проект существует давно, однако тестировщика либо не было в команде, либо существующая до этого модель разработки не позволяла наладить процесс





# ВЫ ОДИН ТЕСТИРОВЩИК НА ПРОЕКТЕ



# ROADMAP



# ROADMAP

СОЗДАНИЕ  
ТЕСТОВОЙ  
ДОКУМЕНТАЦИИ

7

ФОРМИРОВАНИЕ  
ОТЧЕТНОСТИ

9

ВОЗВРАЩЕНИЕ  
К ШАГУ 6

11

ПРОЦЕСС  
ТЕСТИРОВАНИЯ

8

РЕТРОСПЕКТИВА

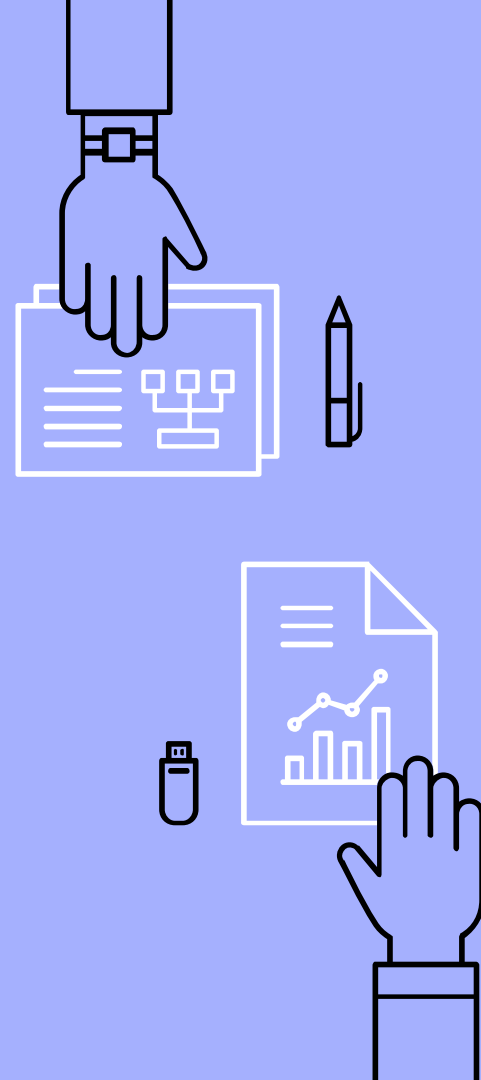
10

СЧАСТЛИВЫЙ  
ЗАКАЗЧИК И  
КОМАНДА



# ОНБОРДИНГ

- ▶ Оптимальное время на прохождение – **2-4 недели**
- ▶ **Рекомендация по формату:** чек-лист с дедлайнами
- ▶ **Наполнение:** гайдлайны по настройке окружения, ссылки на документацию
- ▶ **Nice to have:** контрольные точки с новобранцами



# СЛОЖНОСТИ

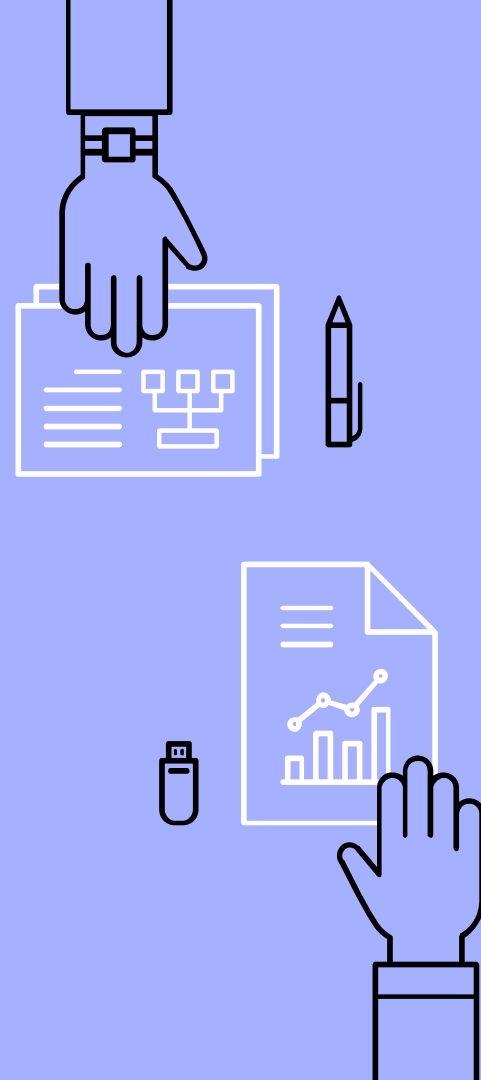
У нас нет на это времени

- ▶ **Решение:** Оценка трудозатрат, оценка эффективности нового сотрудника, обратная связь от новобранца, анализ рисков
- ▶ Подготовка презентации с вашим анализом менеджменту/заказчику



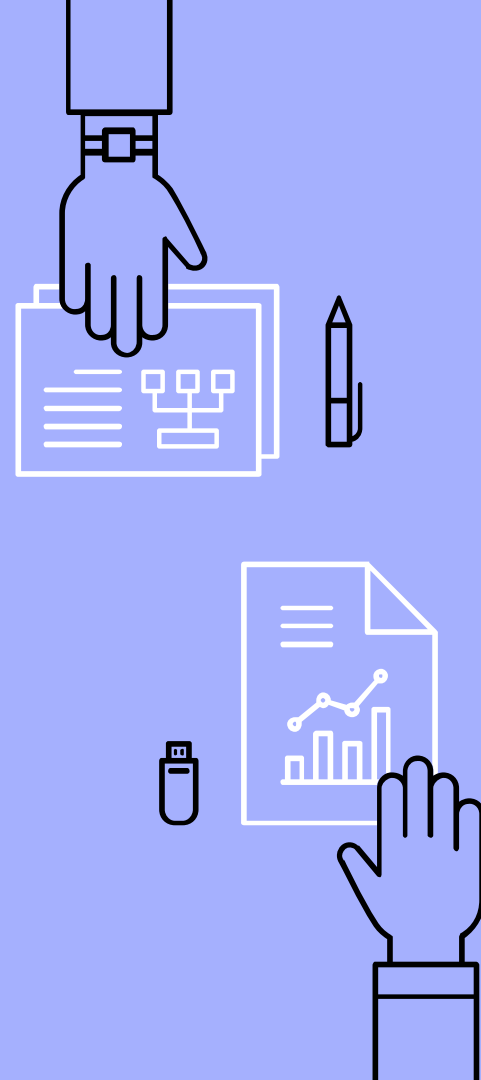
# АНАЛИЗ ТЕКУЩЕЙ СИТУАЦИИ

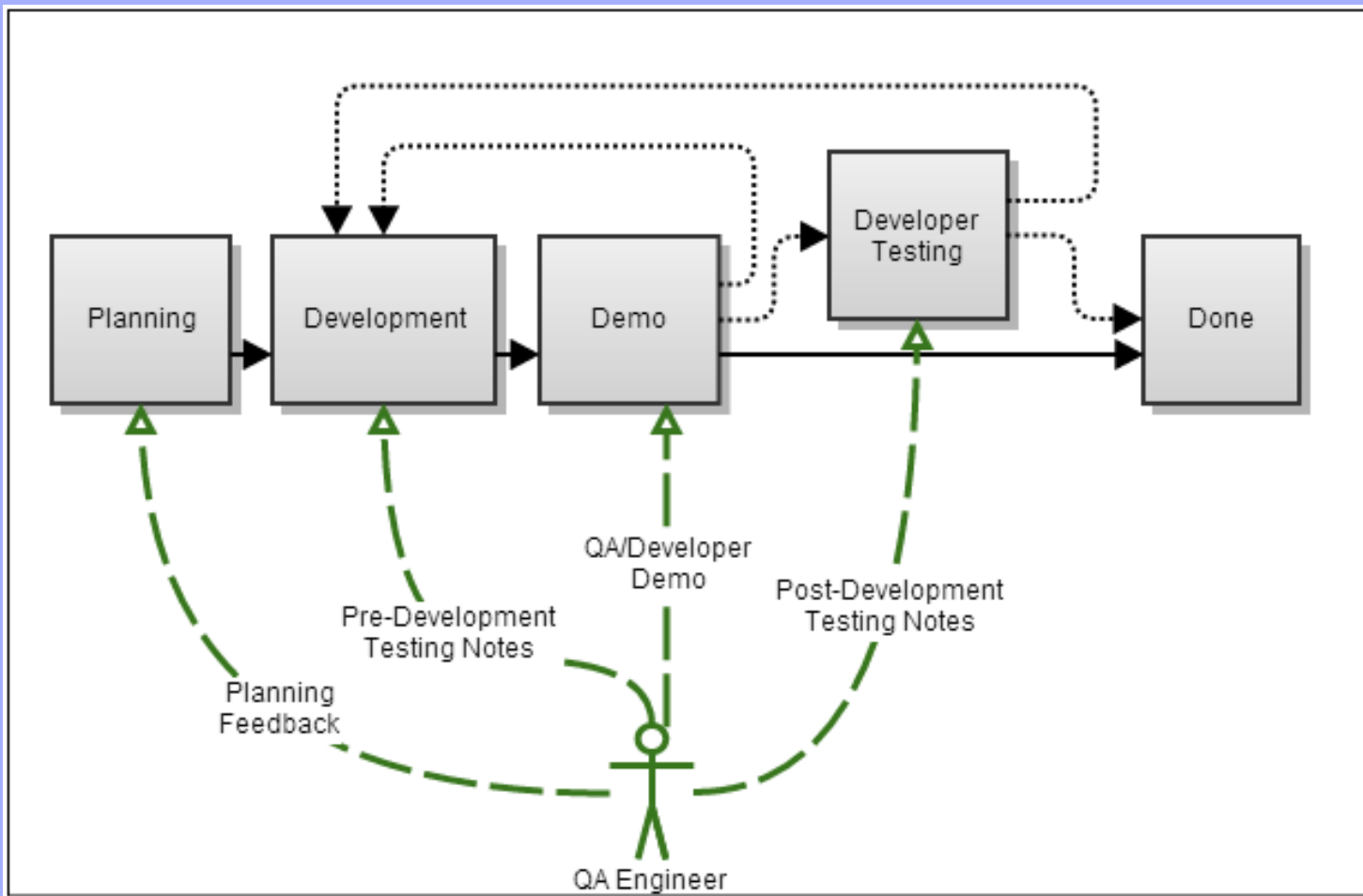
- ▶ Изучение продукта, ad-hoc тестирование
- ▶ Коммуникация с командой (установочный митинг)
- ▶ Коммуникация с заказчиком (one2one)
- ▶ Изучение существующей документации и процесса



# РАЗРАБОТКА QA-FLOW В SDLC

- ▶ Анализ текущей багтрекинговой системы и статусов в ней
- ▶ Подготовка схемы с участием тестировщика на каждом из этапов разработки
- ▶ Презентация схемы команде
- ▶ Внедрение в процесс
- ▶ **Зачем это нужно?** Прозрачность и понимание всей командой процесса тестирования
- ▶ **Nice to have:** календарь релизов







# СОЗДАНИЕ ТЕСТ-ПЛАНА

- ▶ Это руководство для процесса тестирования

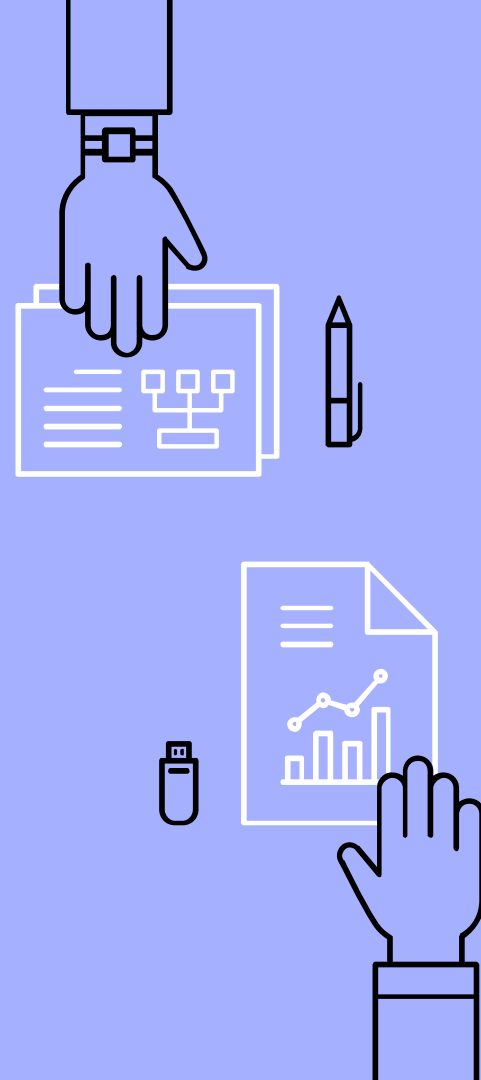
- ▶ **Что должен включать в себя хороший тест-план?**

1. Введение
2. Объем работы
3. Критерии входа и выхода
4. Оценка рисков
5. Ресурсы
6. Тестовая документация
7. Тестовая стратегия (методы, виды, уровни) – можно ограничиться только ей на начальных этапах
8. Сроки тестирования



# НАСТРОЙКА ТЕСТОВОГО ОКРУЖЕНИЯ

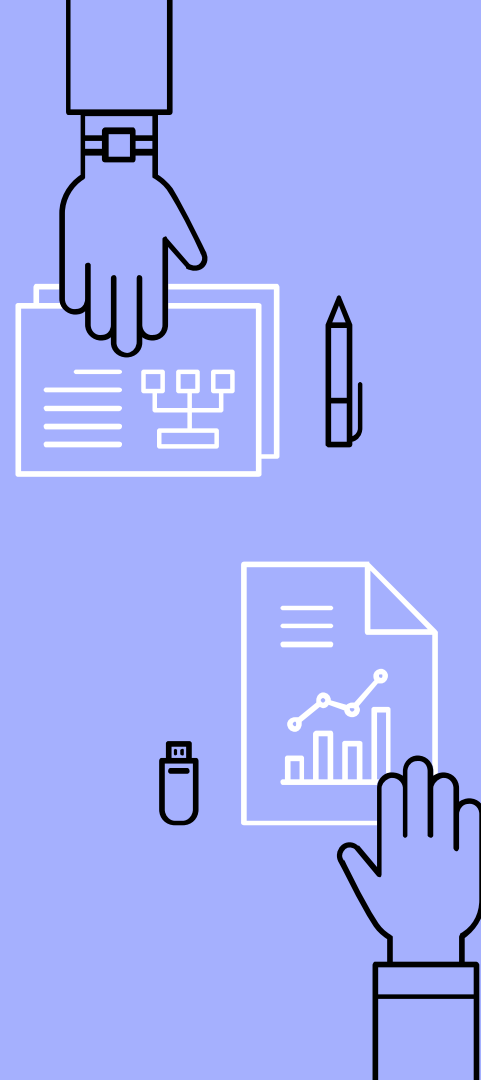
- ▶ Анализ продукта и текущего CI/CD процесса
- ▶ Выбор багтрекинговой системы
- ▶ Выбор системы по управлению тестовой документацией
- ▶ Участие в выборе и разработке тестовых стендов



# СЛОЖНОСТИ

## Отказ в закупке ПО

- ▶ **Решение:** анализ текущего рынка, использование триальных версий приложений для их оценки, выбор наиболее оптимальных вариантов по принципу **цена – решение нужд проекта (интеграция, прослеживаемость, отчетность)**
- ▶ Подготовка презентации с вашим анализом менеджменту/заказчику



# СЛОЖНОСТИ

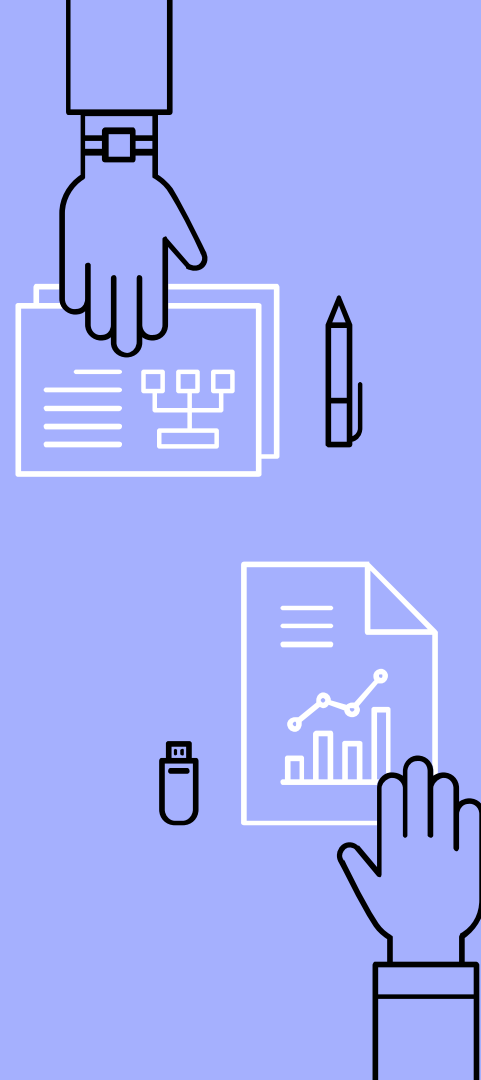
Отказ в создании дополнительных стендов

- ▶ **Решение:** оценка трудозатрат, подсчет критических дефектов на проде
- ▶ Подготовка отчета по результатам вашего анализа



# АНАЛИЗ ТРЕБОВАНИЙ

- ▶ Коммуникация с РО/ВА
- ▶ Анализ текущего решения для хранения требований
- ▶ Анализ текущей формы требований
- ▶ Рекомендации по улучшению
- ▶ Обязательный учет этой фазы при планировании каждого спринта



# СЛОЖНОСТИ

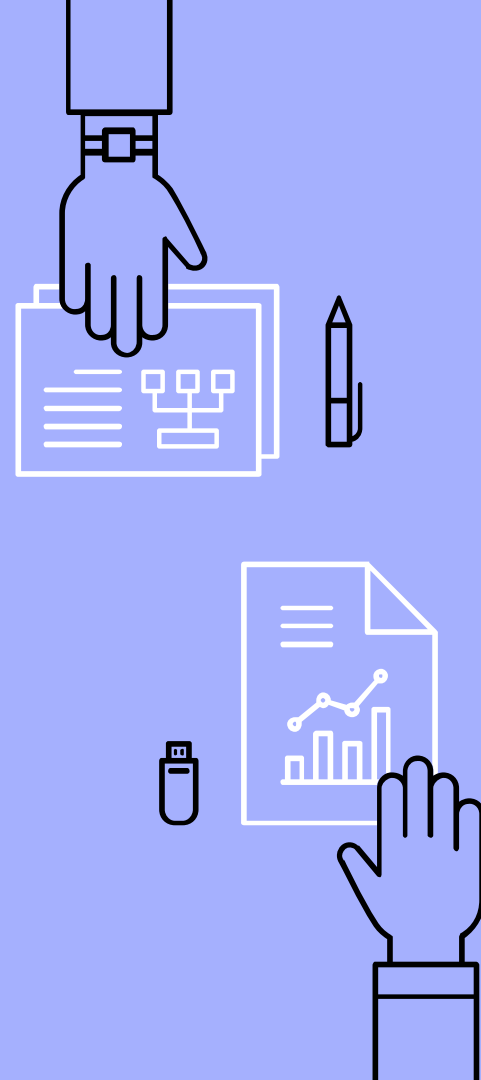
Требования оформляются в виде отдельного предложения или скриншота

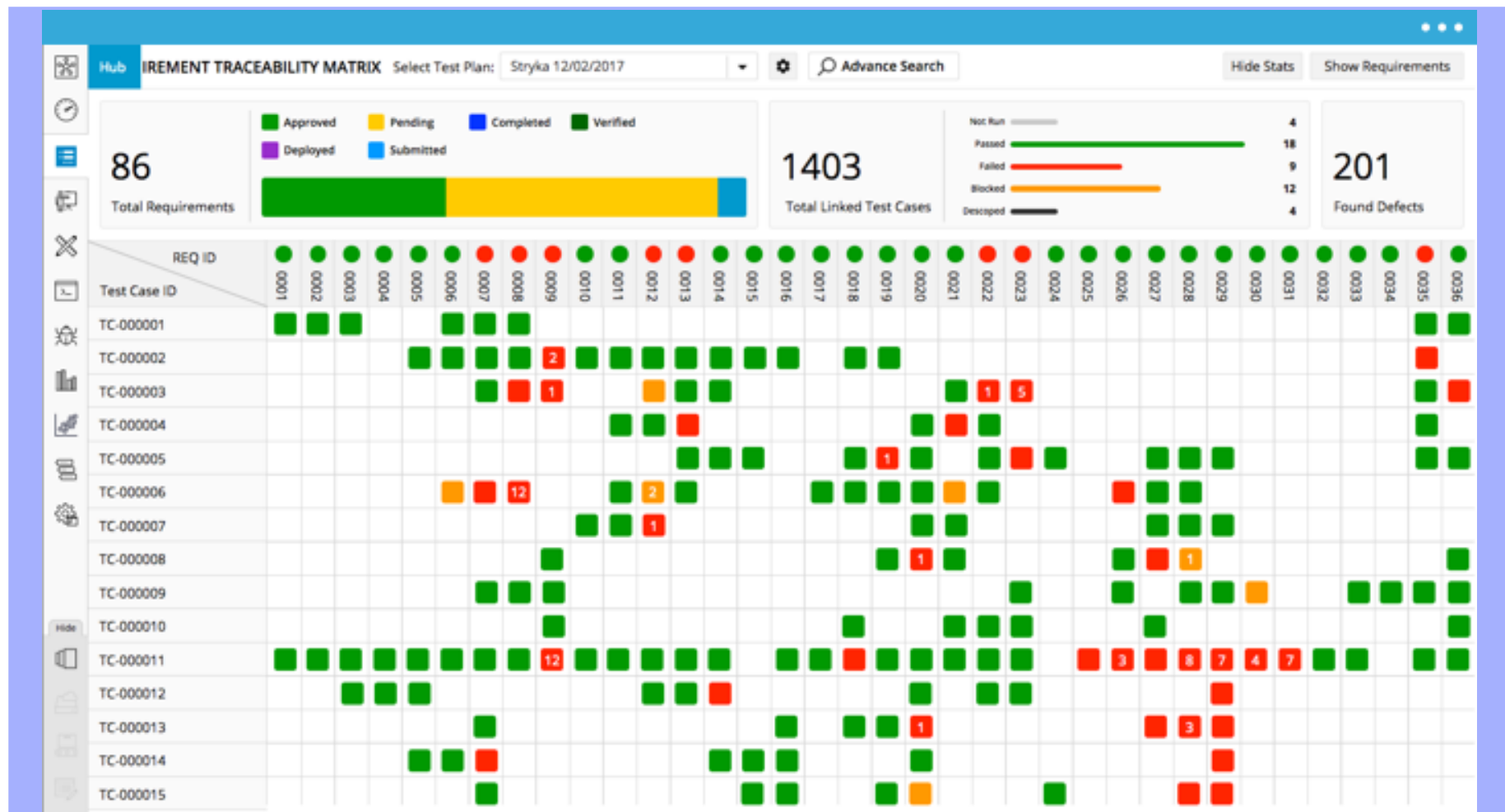
- ▶ **Решение:** анализ требований, отправка требований на доработку, анализ существующих багов, сбор обратной связи от команды, любые изменения должны оформляться новым тикетом и не попадать в бэклог спринта
- ▶ **FEEDBACK:** указание на неполноту требований, что в дальнейшем приведет к неполноте тестирования и различной трактовке со стороны DEV/QA, дополнительным багам



# СОЗДАНИЕ ТЕСТОВОЙ ДОКУМЕНТАЦИИ

- ▶ Использование на проекте чек-листов и/или тест-кейсов
- 1. Сложность проекта
- 2. Длительность проекта
- 3. Вид тестирования
- 4. Текучка кадров
- ▶ Разработка минимального количества тестовых наборов: smoke, NFT, regression, API, automation
- ▶ **NICE TO HAVE:** матрица трассировки, ревью тест-кейсов



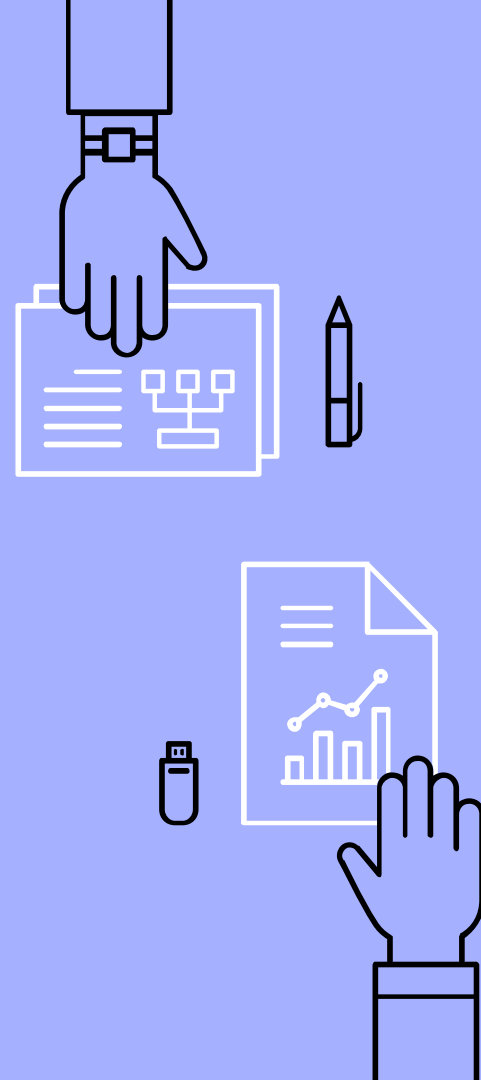




# СЛОЖНОСТИ

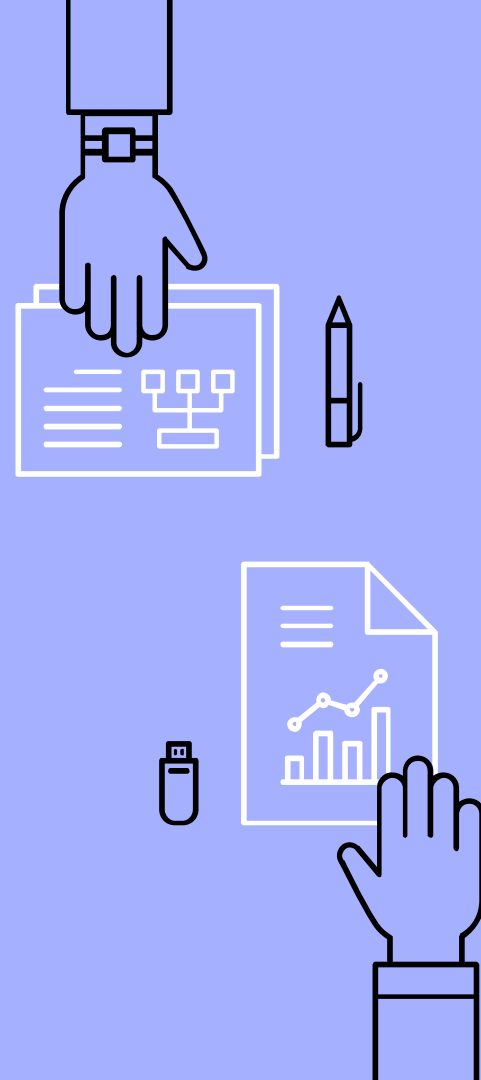
Нет времени на написание тестовой документации

- ▶ **Решение:** объяснение менеджменту важности сохранения тестовых артефактов в системе
- ▶ Smoke – быстрое обнаружение критичных багов
- ▶ NFT – обеспечение полноты тестирования
- ▶ Regression – анализ влияния, интеграционное тестирование
- ▶ API – тестирование на ранних этапах разработки
- ▶ Все тест-кейсы – обучение, полнота тестирования, источник требований, отслеживание изменений, отчетность



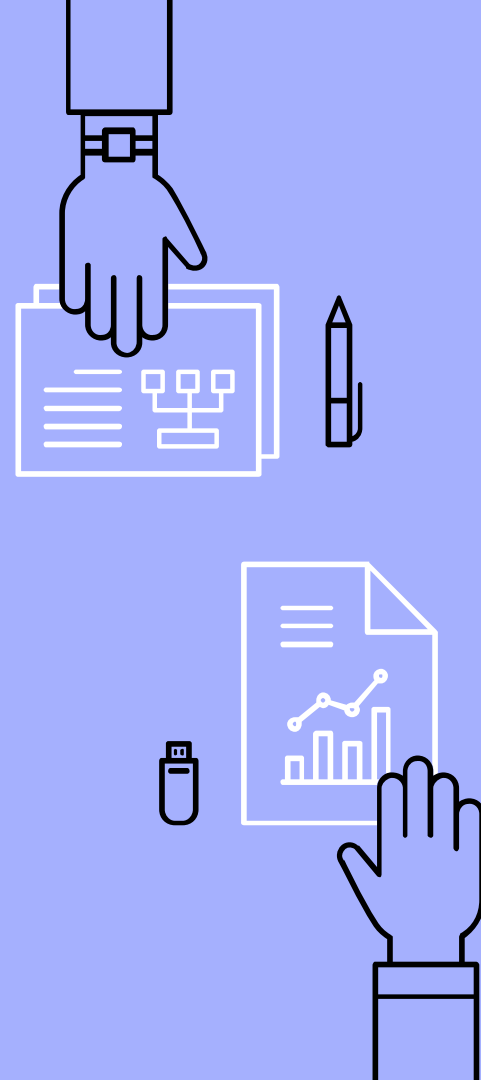
# ПРОЦЕСС ТЕСТИРОВАНИЯ

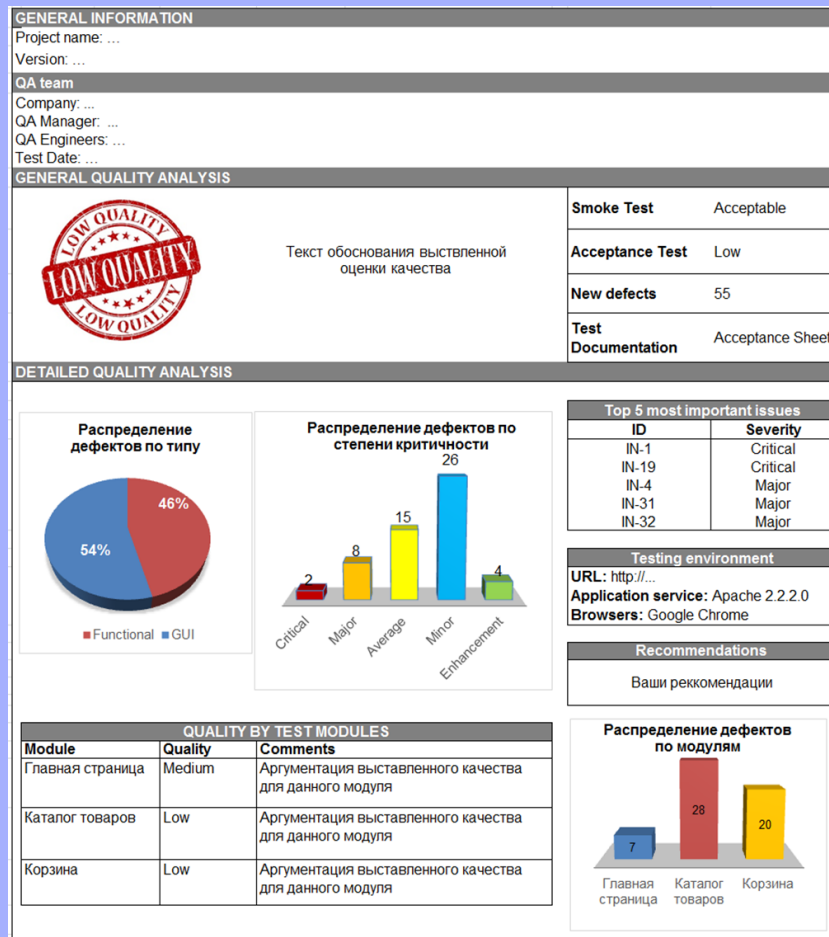
- ▶ Процесс тестирования на всех этапах разработки
- ▶ Эстимация задач должна проводиться с учетом времени на тестирование
- ▶ Непрохождение SMOKE – остановка тестирования
- ▶ Все баг-репорты должны быть в системе, никаких устных договоренностей с разработчиком
- ▶ Четкое и ясное понимание QA flow всей командой
- ▶ STAGE env только для тестировщиков
- ▶ Новый билд только после фикса всех критичных дефектов



# ФОРМИРОВАНИЕ ОТЧЕТНОСТИ

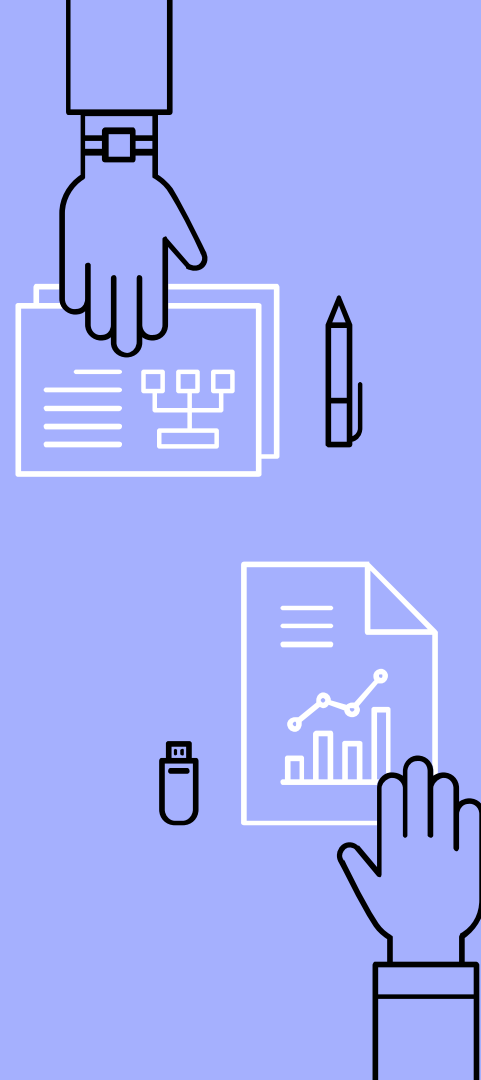
- ▶ Отчет по результатам тестирования должен быть подготовлен в конце итерации и/или перед релизом
- ▶ ОБЯЗАТЕЛЬНО должен содержать решение команды тестирования о возможности к выпуску продукта в релиз
- ▶ Метрики определяются для каждой конкретной команды. Самые распространённые:
  1. Общая оценка качества продукта
  2. Количество обнаруженных багов
  3. Количество написанных тест-кейсов
  4. Диаграммы по PRIORITY обнаруженных багов
  5. Отслеживание изменения метрик в динамике
- ▶ Рекомендации по улучшении/риски





# РЕТРОСПЕКТИВА

- ▶ Даже если вы не работаете по SCRUM, советую инициировать данный митинг для всей команды
- ▶ Основная цель ретроспективы это получение обратной связи по прохождению спринта и разработка корректирующих действий после
- ▶ Обязательно поделитесь с командой: что прошло хорошо и что требует улучшения
- ▶ Выберите несколько наиболее приоритетных задач для следующего спринта и улучшите существующий процесс
- ▶ На следующей ретро поделитесь результатами по исполнению Action Items



# THANKS!

## Any questions?

You can find me at:

**Telegram:** @artsiom\_qa

**Mail:** qa.rusau@gmail.com

