Инфраструктура Android UI-тестов

Эмилия Кударева

Одноклассники

Какие проблемы мы хотим решить?

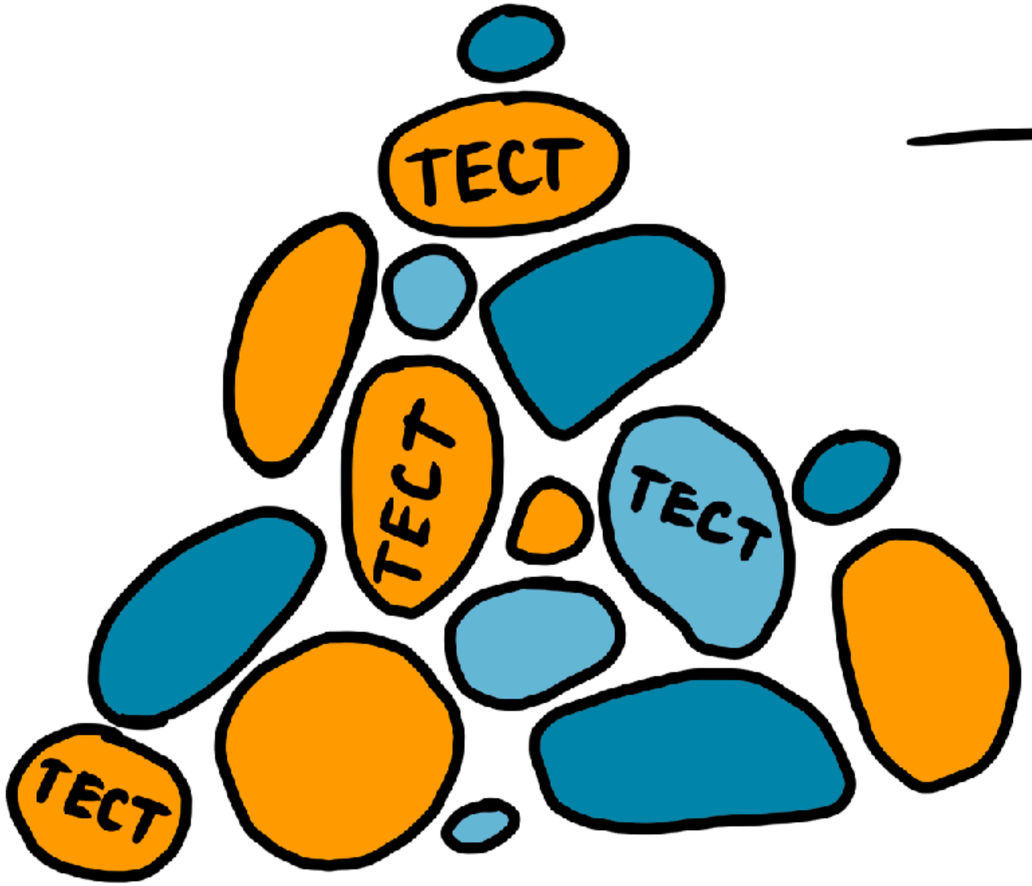
- Нестабильные автотесты
- Недоверие к результатам
- Сложно автоматически оценивать результаты

Что у нас есть?

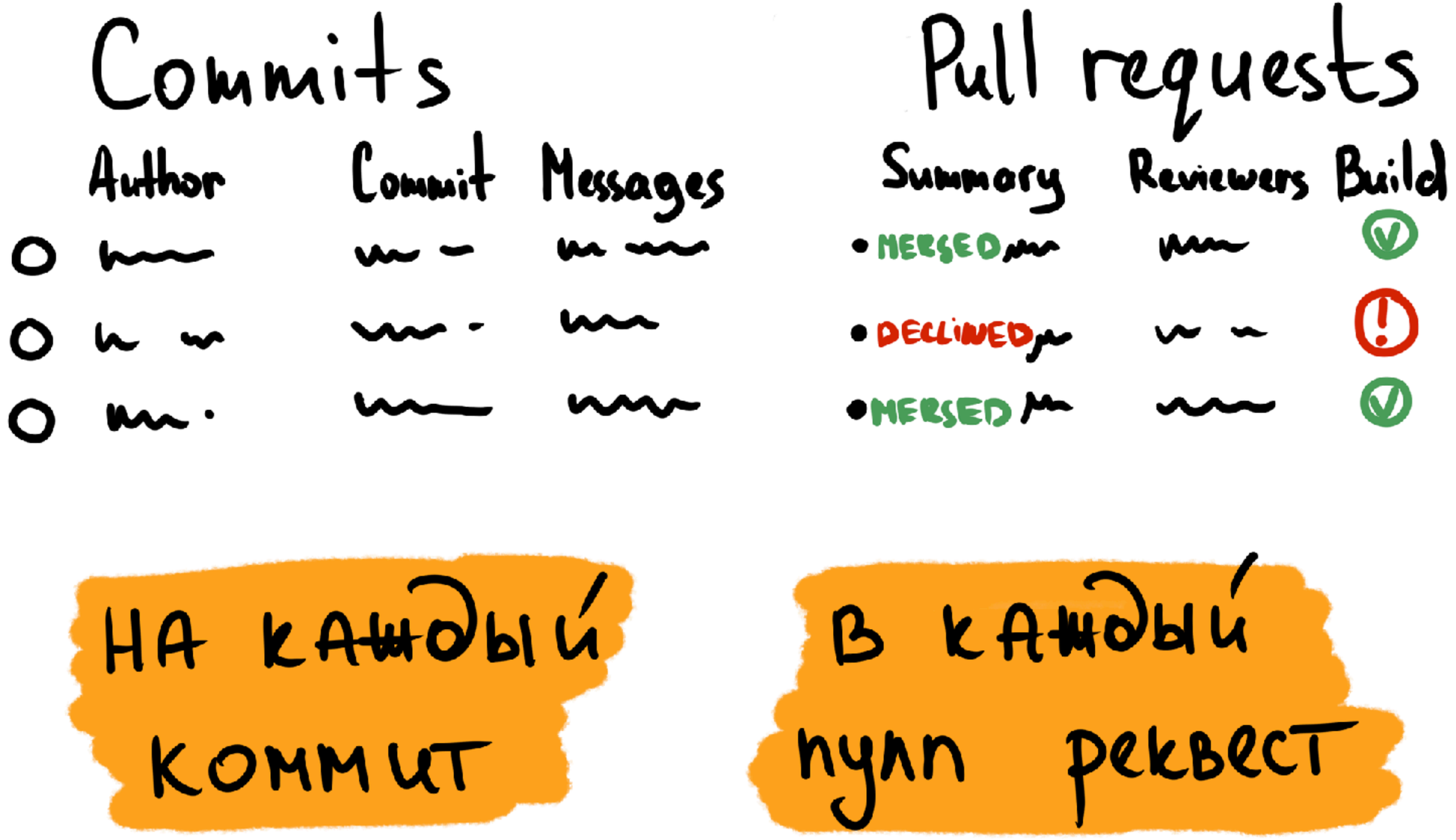
> 1.000 тестов

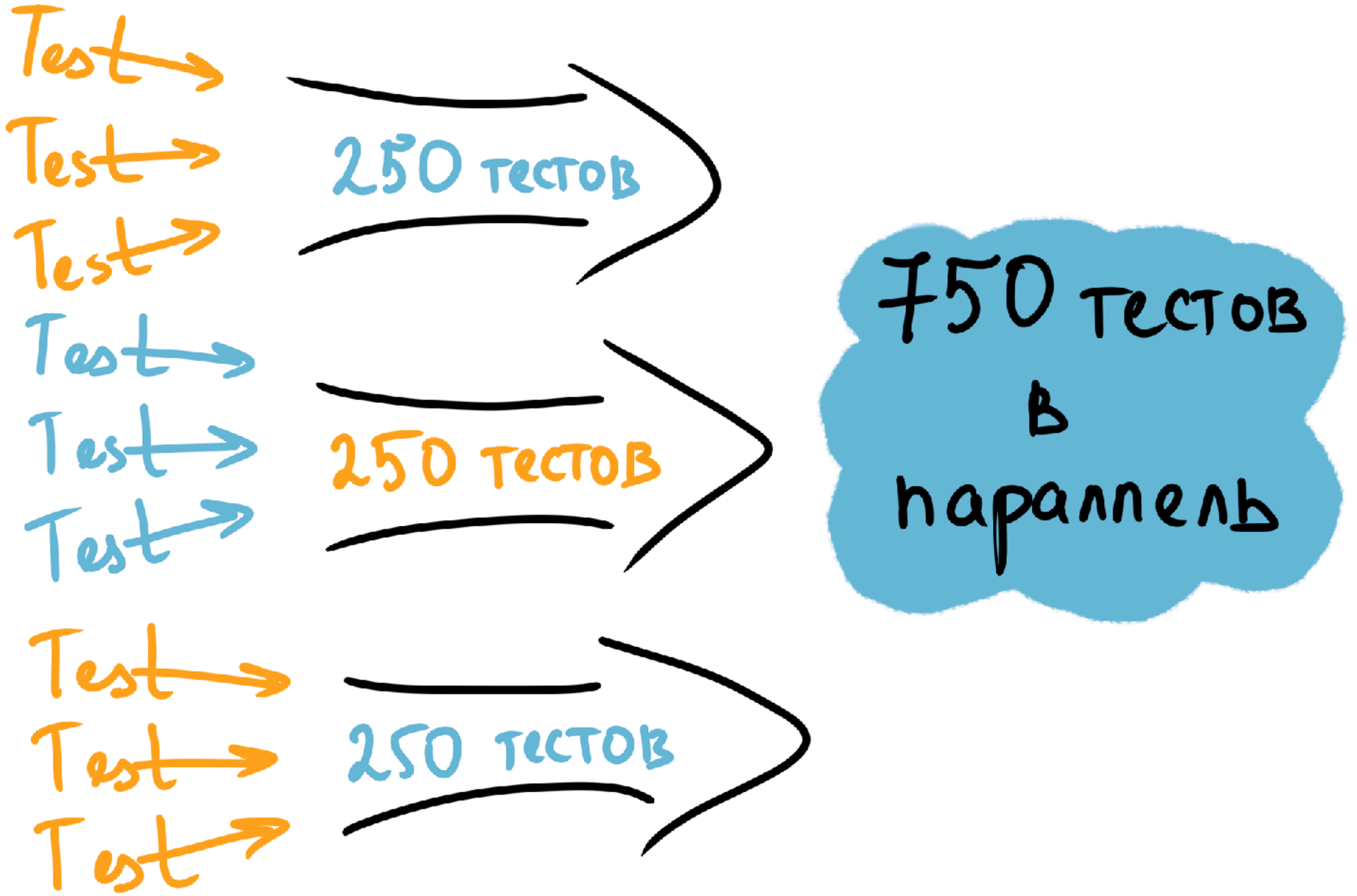
~15 min
сборка и unit'ы на TeamCity

~20 min
прогон UI тестов



~30 min
получение результатов


Commits
Author
Commit Messages
Pull requests
Summary
Reviewers
Build
MERGED
DECLINED
MERGED
НА КАЖДЫЙ КОММИТ
В КАЖДЫЙ ПУЛЛ РЕКВЕСТ


Test
Test
Test
250 тестов
Test
Test
Test
250 тестов
Test
Test
Test
250 тестов
750 тестов
в
параллель

План:

1. Запрет мерджа в пулл реквестах

2. Стабилизация тестов

1. Как устроен запрет мерджа в пулл реквестах?

Автоматическое разрешение мерджа зависит от:

- сборки приложения и тестов
- прогона unit тестов
- прогона UI тестов
- аппрувов от мейнтейнеров

Какие бывают результаты запуска UI тестов?

Пулл реквест

—

основная точка входа

Сообщение в пулл реквесте

1. Статистика по запущенным тестам
2. Ссылки на отчёты по тестам
3. Ссылки для перезапуска тестов
4. Ссылки на apk и сборку в TC
5. Ссылка на статистику по тестам

При разрешении мерджа:

- ✓ Количество упавших на ветке тестов по статусам:

  Стабильные 0

  Нестабильные 6

  Вечнопадающие 5

  ...

При запрете мерджа:

✗ Количество упавших на ветке тестов по статусам:

Стабильные 1

Нестабильные 10

Вечнопадающие 5

Упавшие на ветке

https://report/tests...

...

мы все тесты сломали

При запрете мерджа:

- ✗ Количество упавших на ветке тестов по статусам:
  - Стабильные 1
  - Нестабильные 10
  - Вечнопадающие 5
- Упавшие на ветке https://report/tests ...

честно все

да вставай

дежурный, уже 2 ночи вставай

Разделение тестов на:

- стабильные
- нестабильные
- вечно падающие

Стабильность тестов считается:

- для n последних запусков

Стабильность тестов считается:

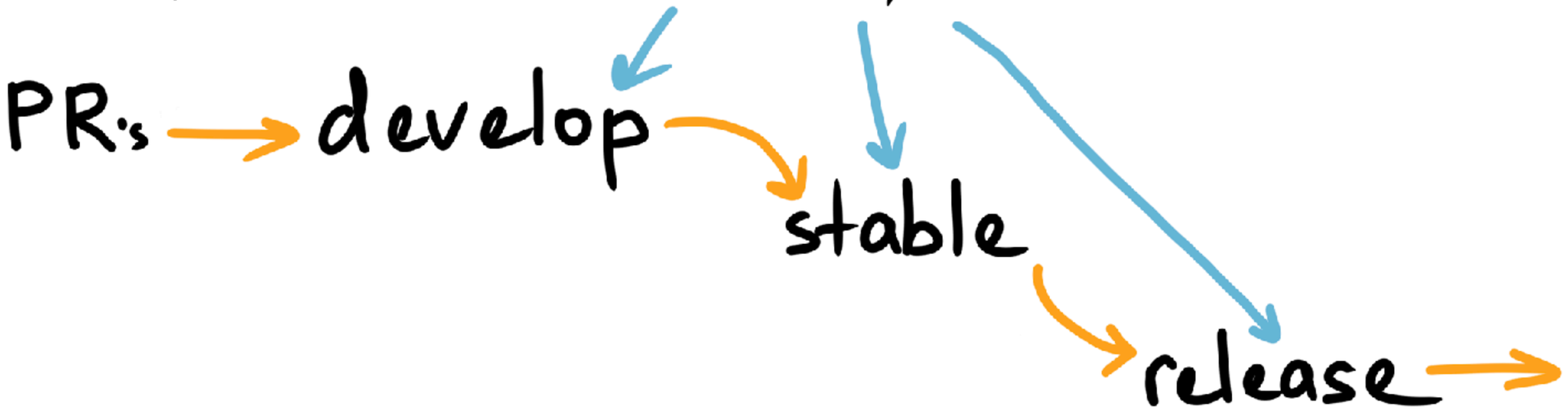
- для n последних запусков
- на доверенных джобах TeamCity

✗ ручные запуски с разными параметрами

✓ автоматические запуски с одинаковыми параметрами

Стабильность тестов считается:

- для n последних запусков
- на доверенных дюбах TeamCity
- **на основных ветках**



Стабильность тестов считается:

- для n последних запусков
- на доверенных джобах TeamCity
- на основных ветках
- исключаем инфраструктурные падения

Стабильность тестов считается:

- для n последних запусков
- на доверенных джобах TeamCity
- на основных ветках
- исключаем инфраструктурные падения
- считаем запуски и перезапуски

Стабильность тестов считается:

- для n последних запусков
- на доверенных джобах TeamCity
- на основных ветках
- исключаем инфраструктурные падения
- считаем запуски и перезапуски

+ Исключаем из падающих на ветке тесты с известными падениями

Выводим ссылки на отчёты по падающим тестам:

- стабильные
- топ нестабильных

* максимально выводим n тестов

При запрете мерджа:

ooo

Топ стабильных тестов, упавших на ветке:

- https://report/test
- https://~~~~

Топ нестабильных тестов, упавших на ветке

- https://~~~~
- https://~~~~

Смотрим статистику
по тестам

Что важно учитывать:

- Новые тесты ≠ стабильные
  - ↓
  - Ждём n запусков
- Тест начинает падать
  - ↓
  - Ждём наполнения истории
- Баг попал в основную ветку
  - ↓
  - Следим за падениями

Что мы делаем?

разделяем тесты по категориям

+

отслеживаем их стабильность

+

игнорируем известные и инфраструктурные падения

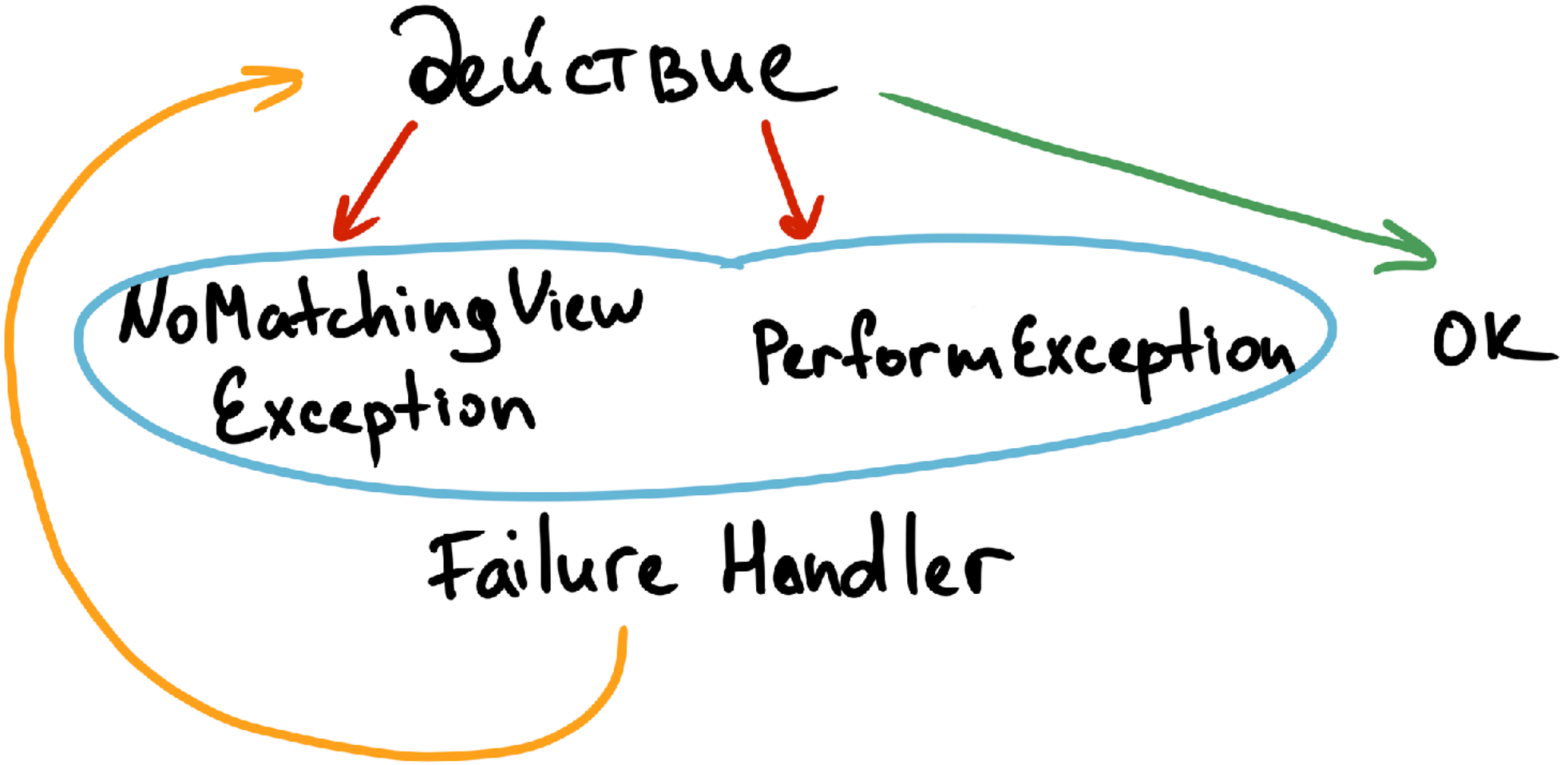
=

АВТОМАТИЧЕСКИ ЗАПРЕЩАЕМ МЕРДЖ

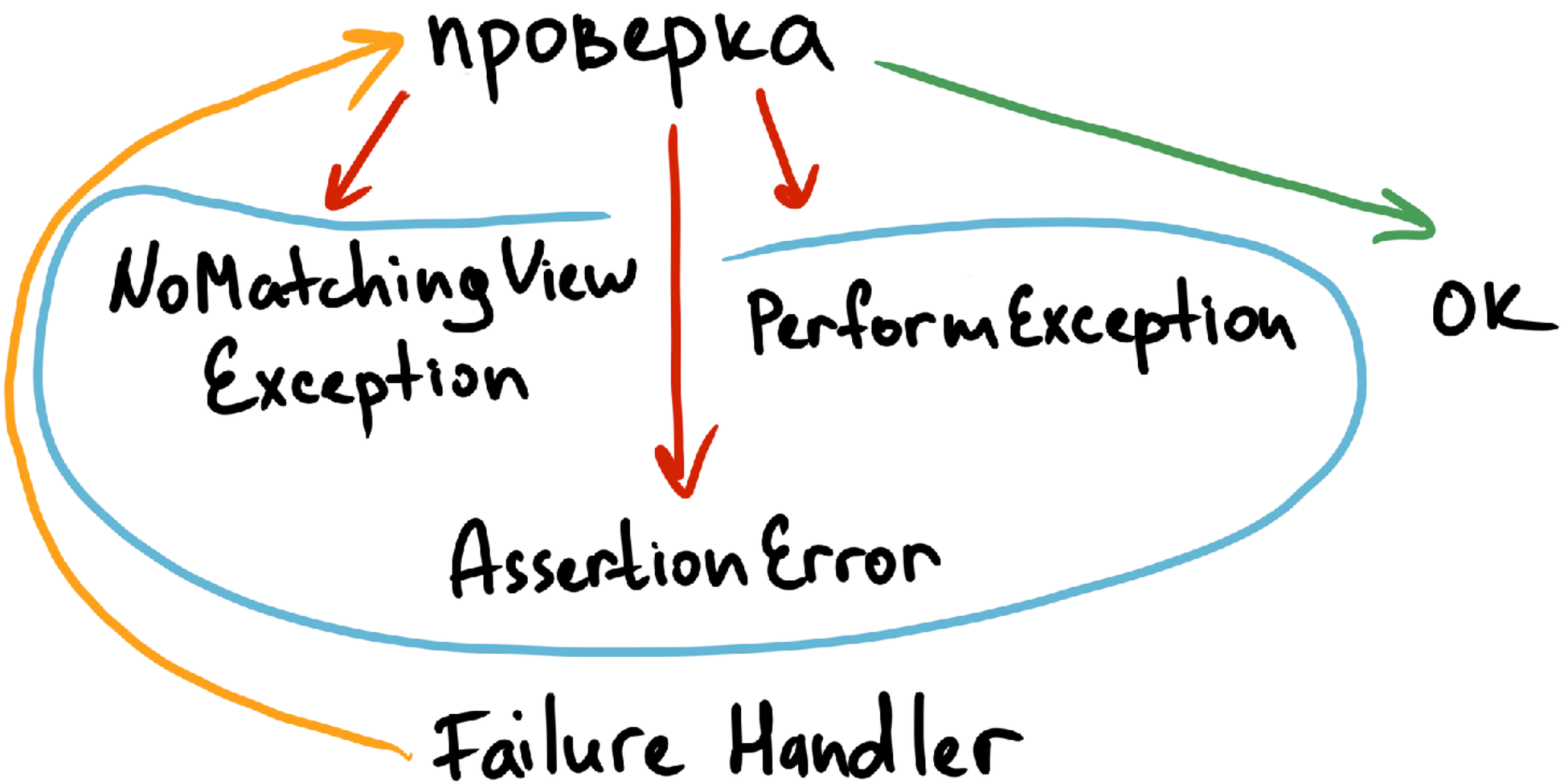
2. Как мы делаем тесты стабильнее?

2.1 Внутри кода тестов

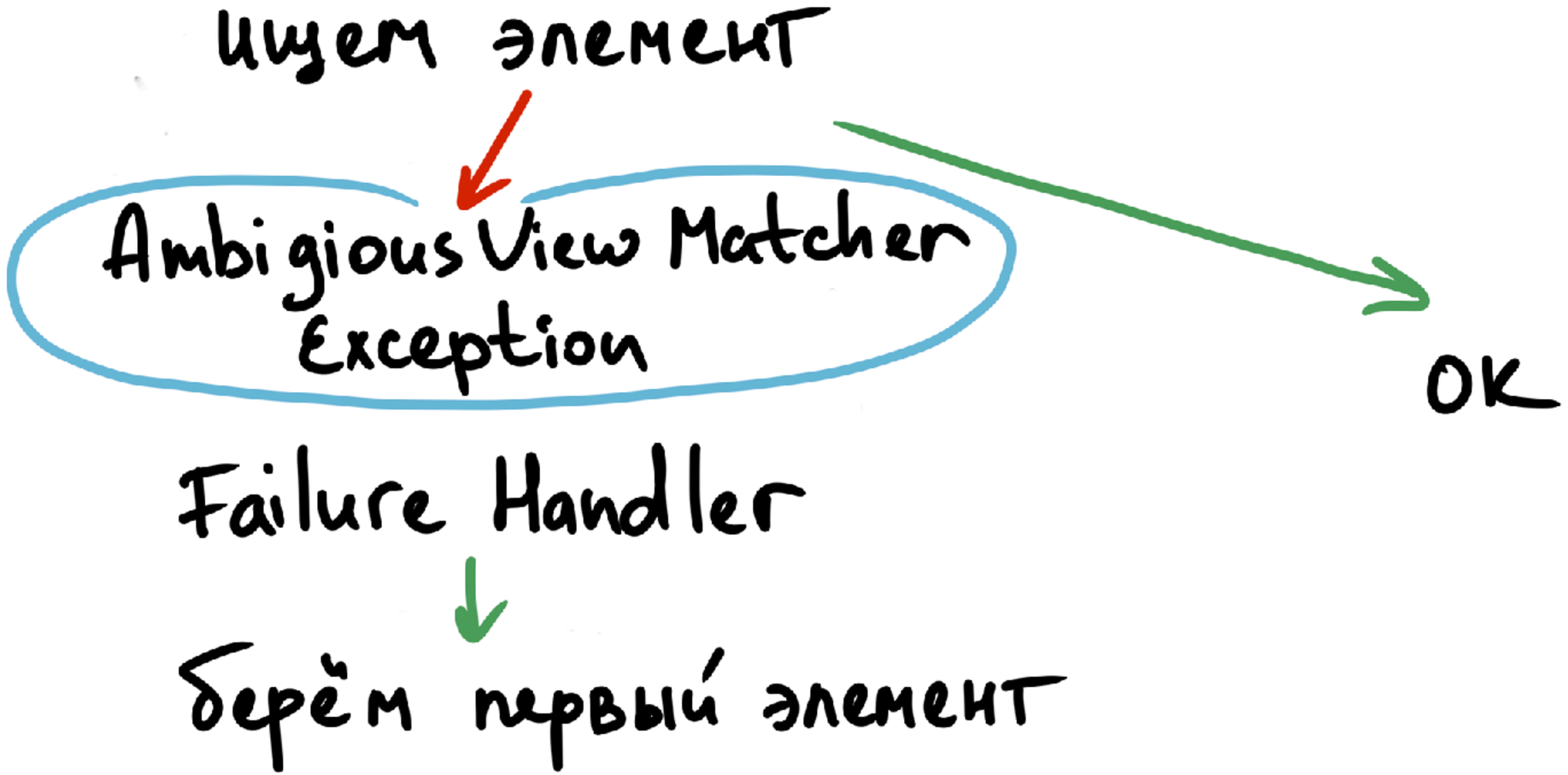
Ретраим действия



Ретраим проверки



Работаем с повторяющимися элементами



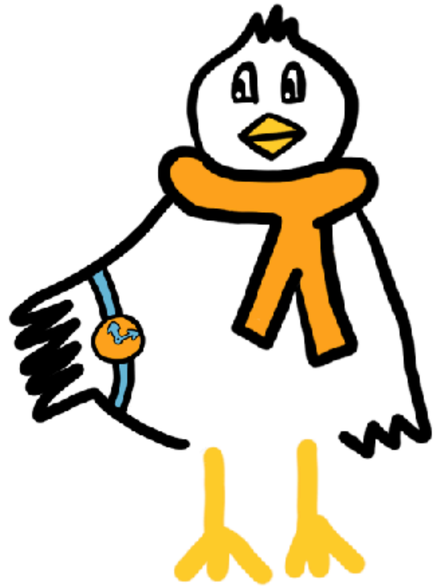
Используем ожидания

IdlingResource

- IdlingViewShown
- IdlingViewNotShown
- IdlingFirstViewShown
- IdlingViewCount
- IdlingWaitingForActivity

\+

паттерн LoadableComponent



Уходим от UI

- API методы
- Быстрая навигация = deep links
- Быстрый логин

Что мы сделали внутри кода тестов?

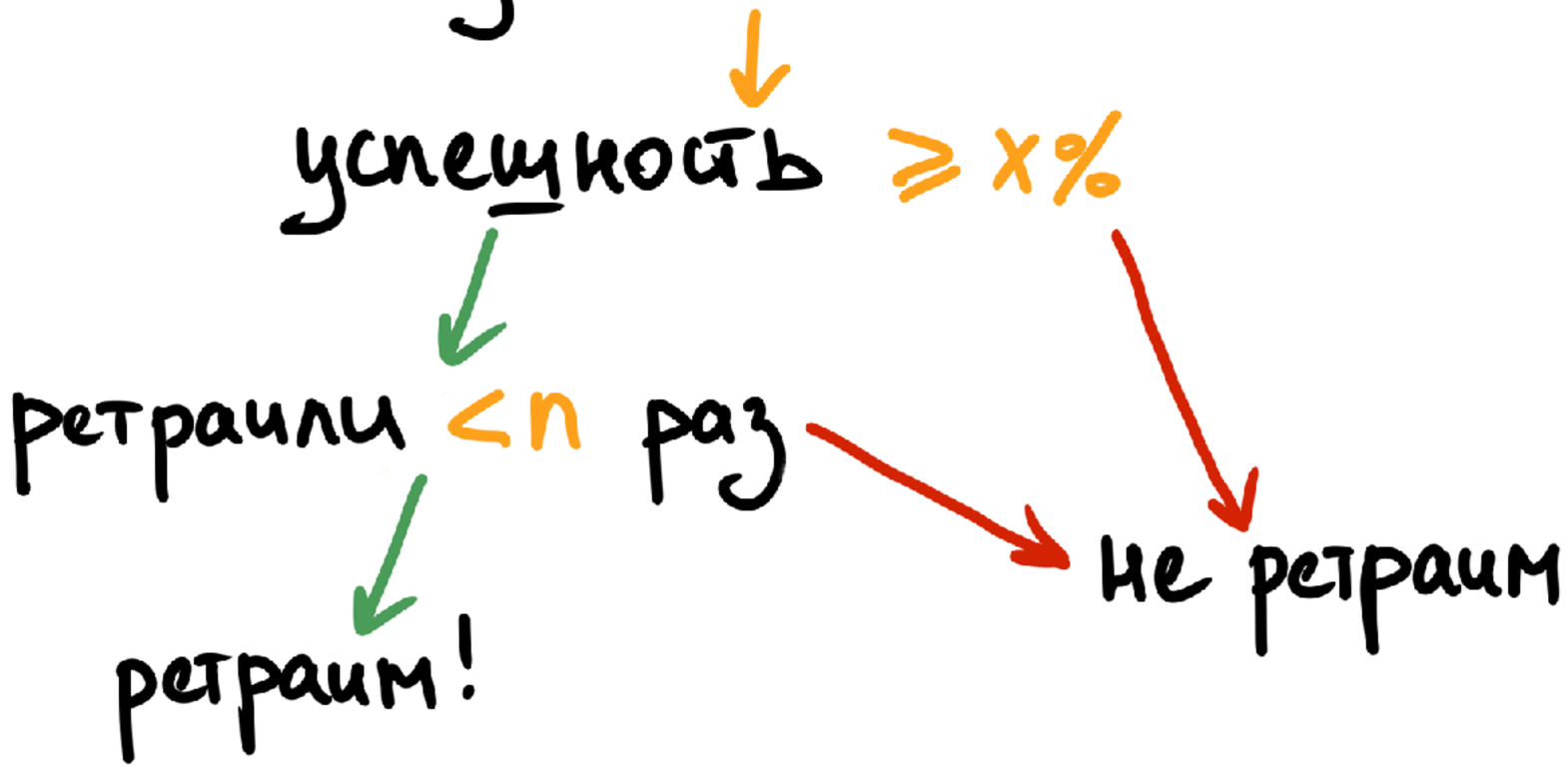
- ретраим действия и проверки
- работаем с повторяющимися элементами
- используем ожидания
- заменяем UI шаги

2.2 Внутри Runner'a

Ретраим действия Runner'а

- Получения эмулятора
- Установки приложения и тестов
- Скачивания данных с эмулятора

Ретраим тесты



* ограничиваем max ретраев на запуск

Используем таймауты

- ограничиваем время прогона теста
- останавливаем зависшие тесты

Что мы сделали внутри Runner'а?

- ретраим действия Runner'а
- ретраим тесты
- ограничиваем время

2.3 Внутри пулл реквеста

Из сообщения в PR'е

- перезапустить только упавшие тесты
- перезапустить все тесты

Замораживаем фича тоггл

- Для всех тестов
- Для отдельных тестов через аннотации

Где и как мы делаем тесты стабильнее?

- в коде
- в раннере
- в пулл реквесте
- с фича тоггл

3. Как мы запускаем тесты?

Параллелизируем тесты

- временные боты

  тест атомарен относительно себя

- статические боты на чтение
- статические боты на редактирование

  отдельные прогоны + последовательные запуски

Добавляем аннотации для запуска

- тип устройства: смартфон / планшет
- тип теста
- имя пакета, класса, метода
- теги
- имя автора

Настраиваем запуск через TeamCity

- среда для запуска
- Strict Mode
- Дополнительные логи
- ...

Что мы сделали для запуска тестов?

- запараллелили всё, что можно
- навесили аннотации под разную логику
- настроили запуск с кастомными параметрами

2018	2020		2022
~300	~700	всего тестов	>1.000
~2%	~6%	падающих	~1%
15-20 min	15 min	время запуска	~20 min
X	X	ретраи действий	✓
X	X	ретраи шагов	✓
X	X	ретраи тестов	✓
X	✓	запрет мерджа	✓ new

Время для вопросов

Иду писать автотесты после Heisenbug'а

Эмилия
Кусарева

kutcareva@gmail.com

https://t.me/addstickers/ducklingAutomator