

CI процессы в Android разработке Avito

Пинчук Сергей

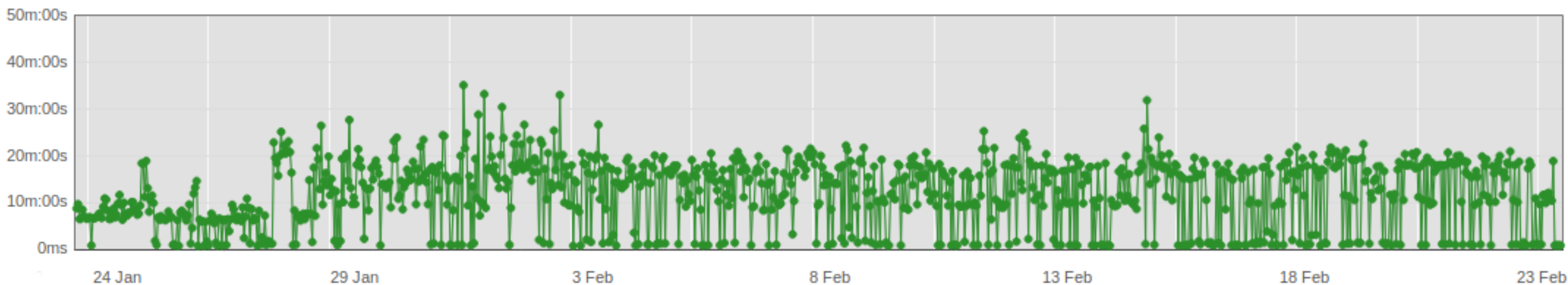
План

- Общие сведения о Avito Android
- Интеграция проверок на CI
- Обзор набора тестов и их оптимизации
- Планы на будущее
- Итоги
- Вопросы

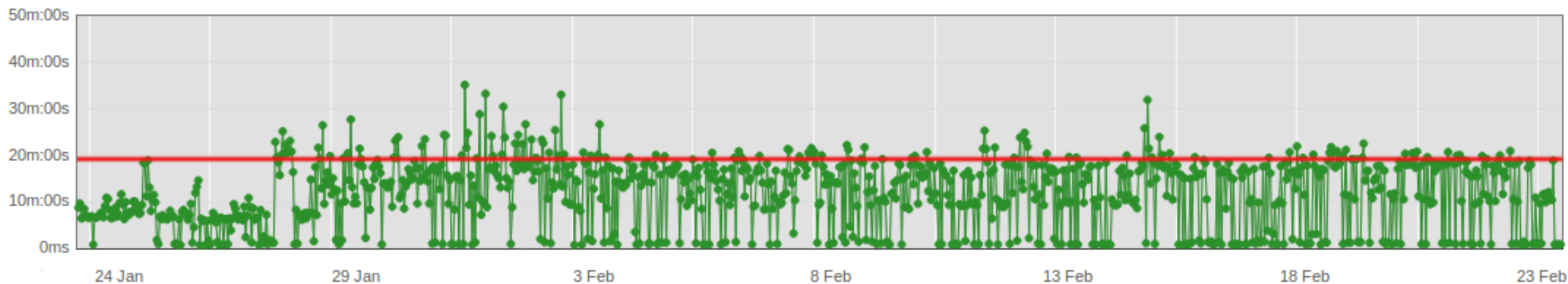
Avito Android CI это

- Более 11,000 unit-тестов
- Более 800 компонентных тестов
- Более 300 функциональных end-to-end тестов
- Порядка 50 прогонов в день
- Продолжительность 100 часов ежедневно

Время выполнения unit-тестов



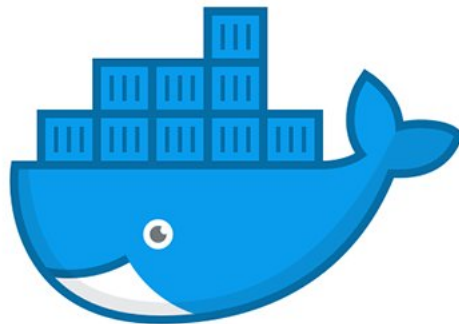
Время выполнения unit-тестов



Стек технологий

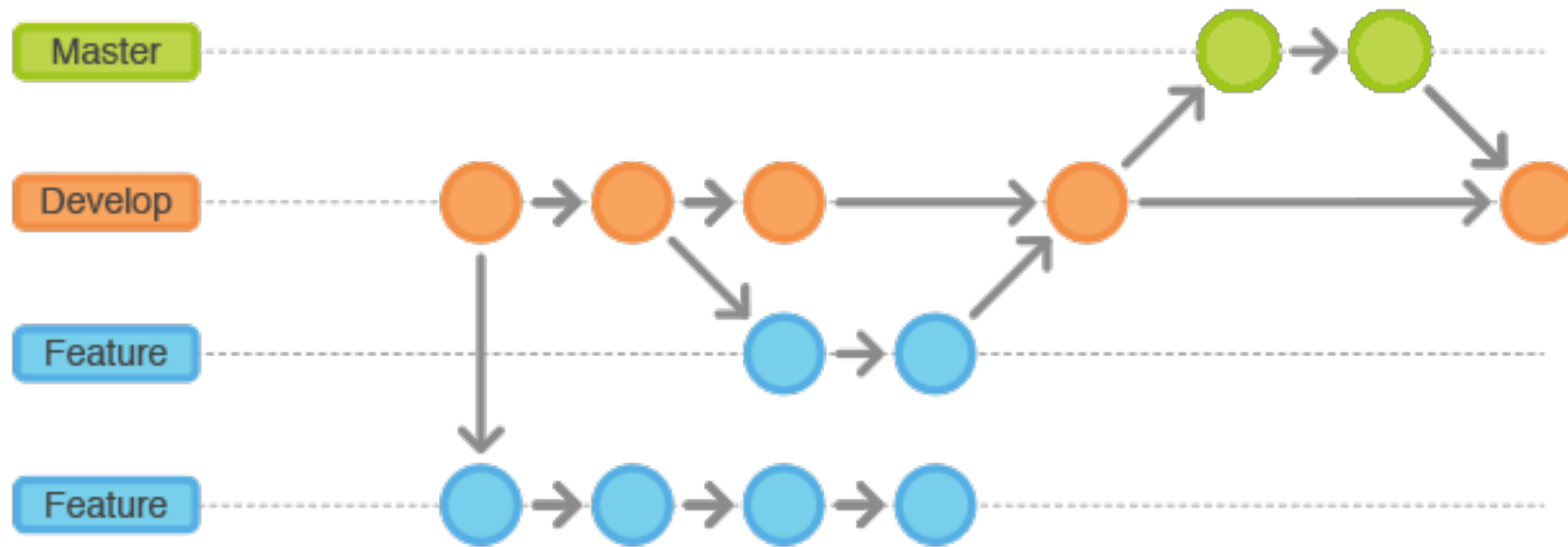


Bitbucket

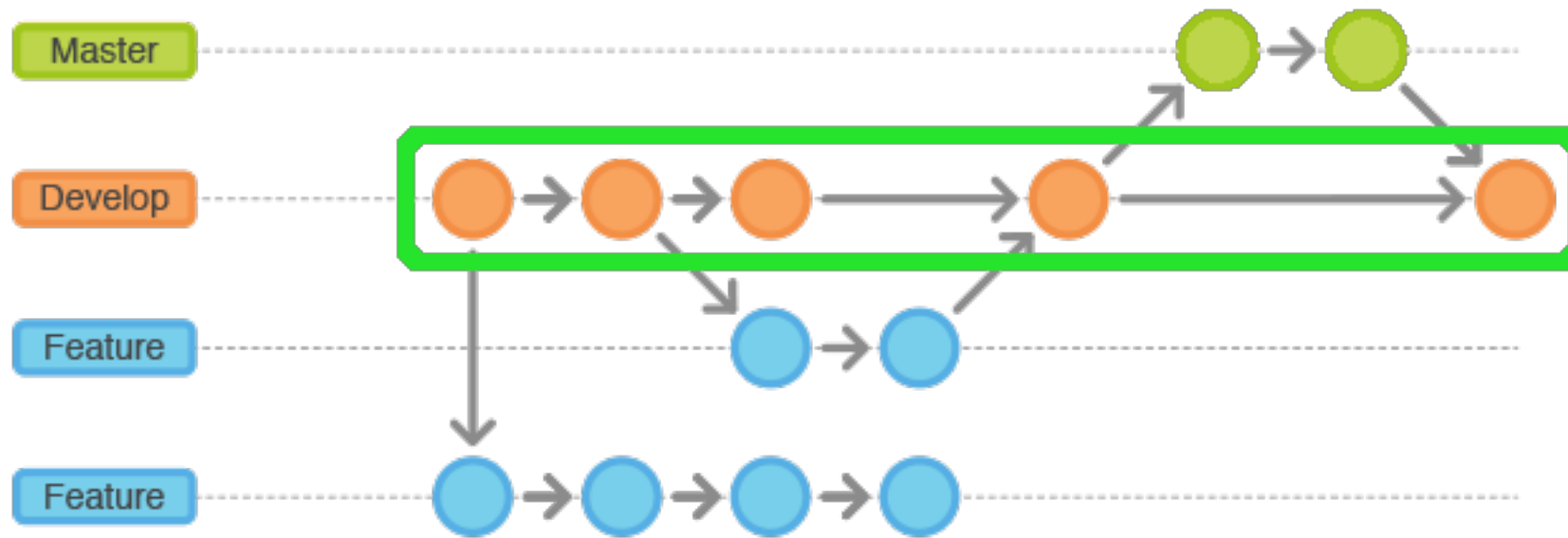


kubernetes

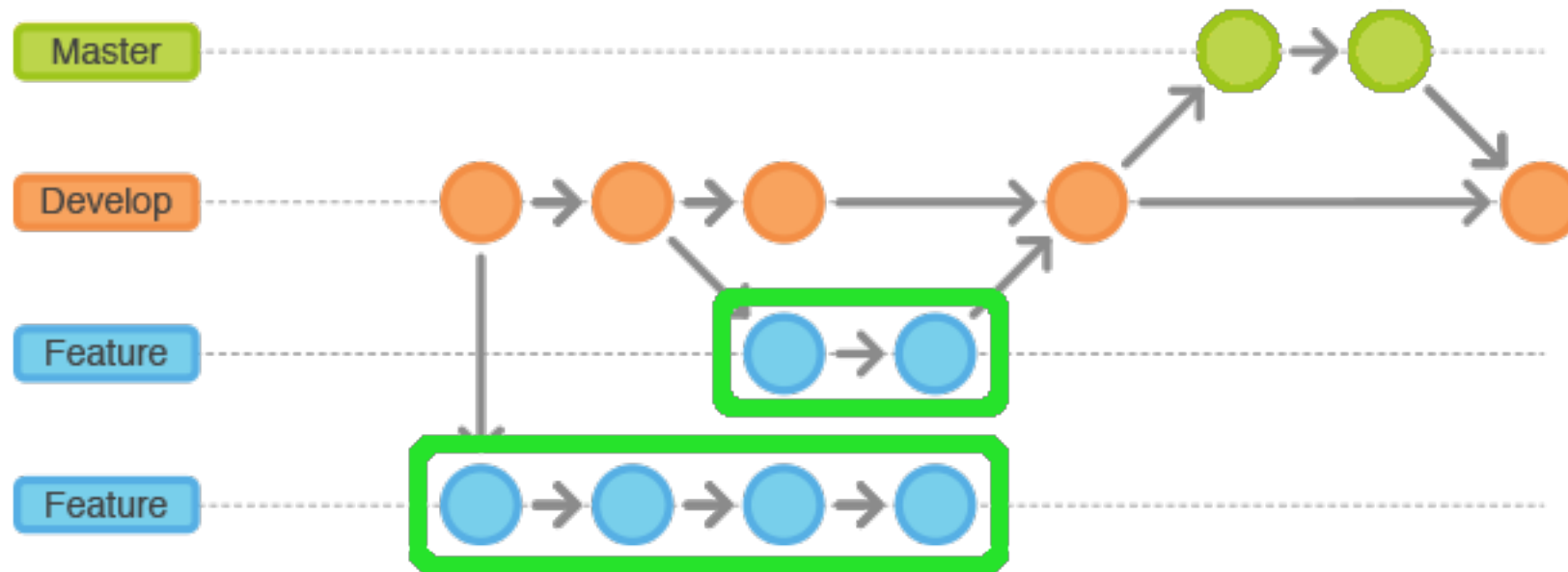
Git flow



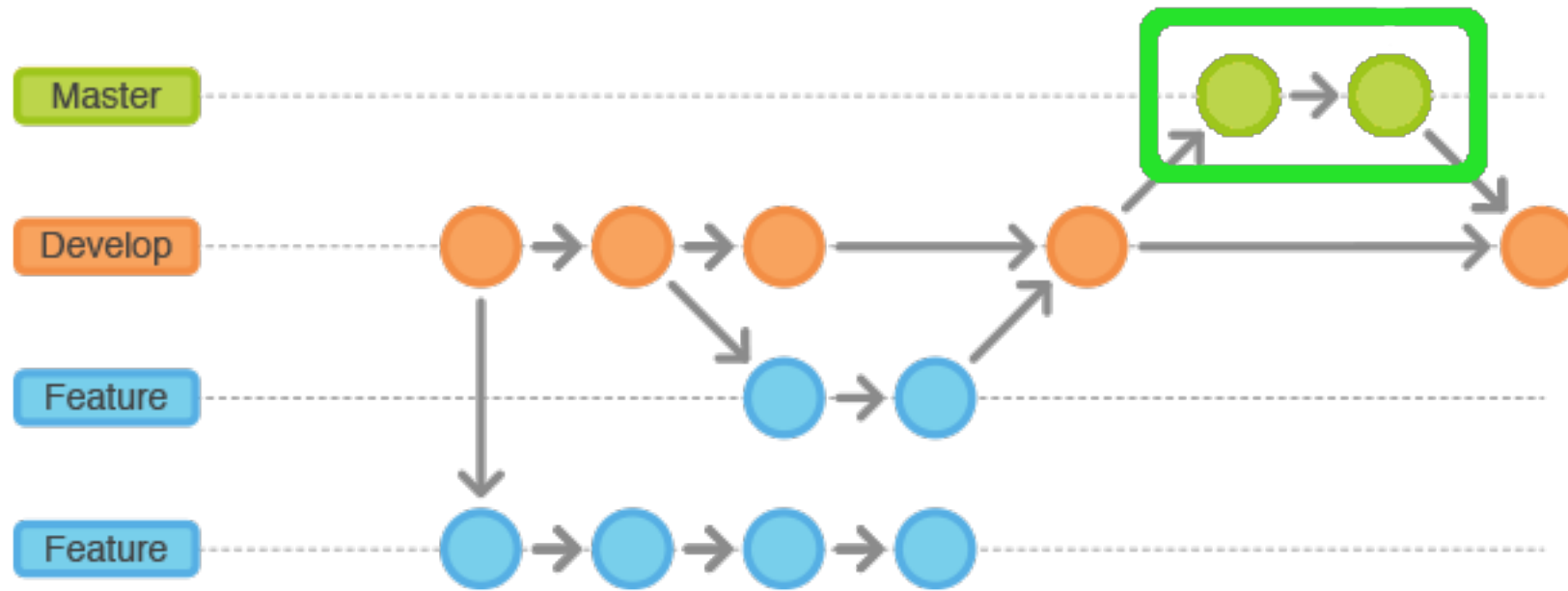
Git flow



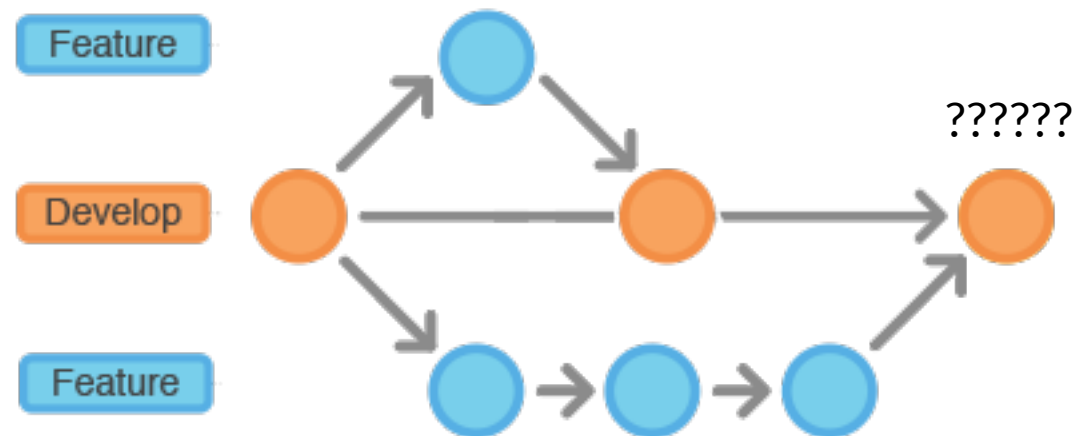
Git flow



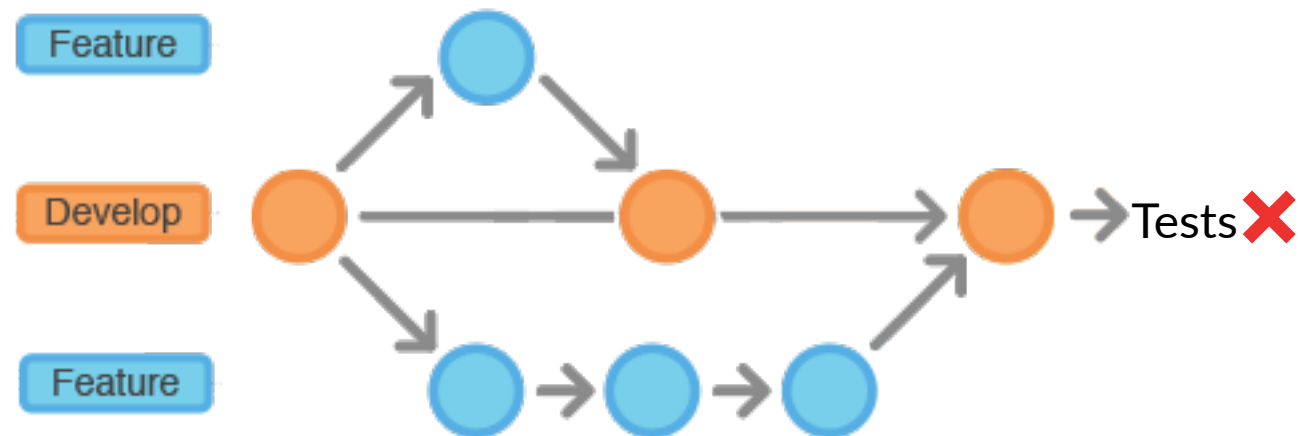
Git flow



Интеграция тестов на CI



Интеграция тестов на CI



Інтеграція тестов на CI

 
in avito-android-dev

Я сломал девелоп, чиню



11 replies

 
Табличку повесил?

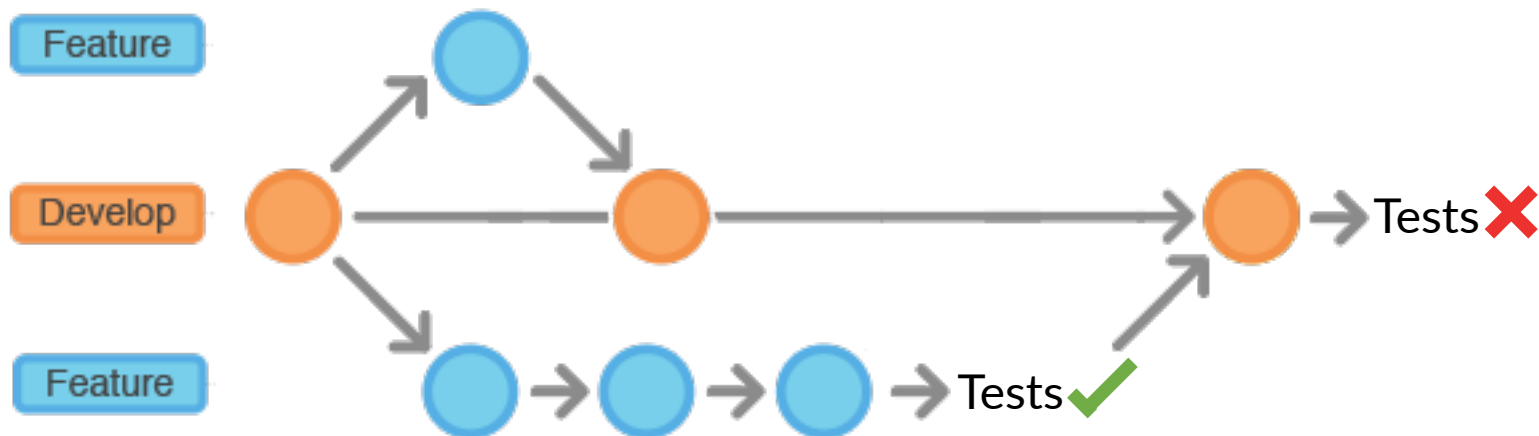
 
😊 Заслужил право))

 
давно сломался? а то я полчасу назад стянул дев

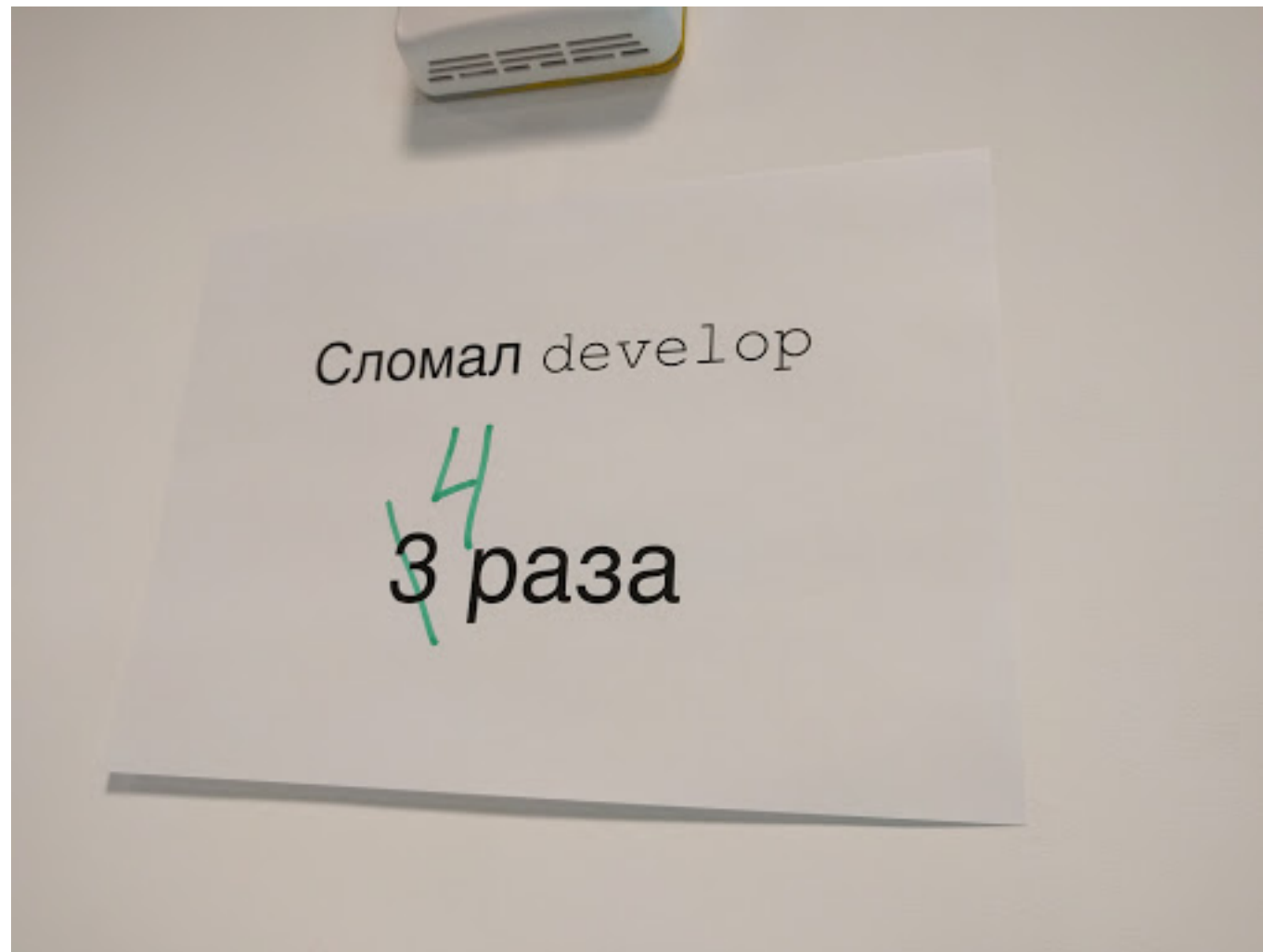
 
час назад



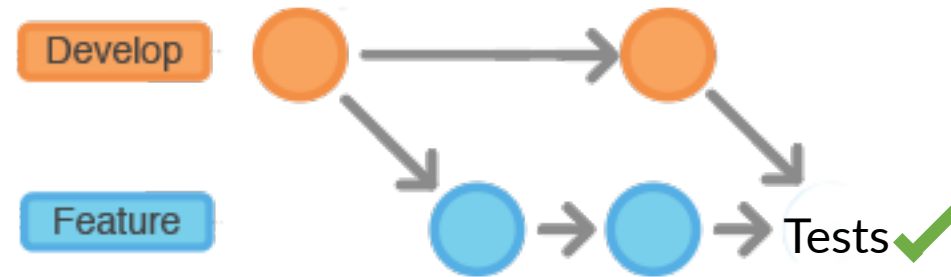
Интеграция тестов на CI



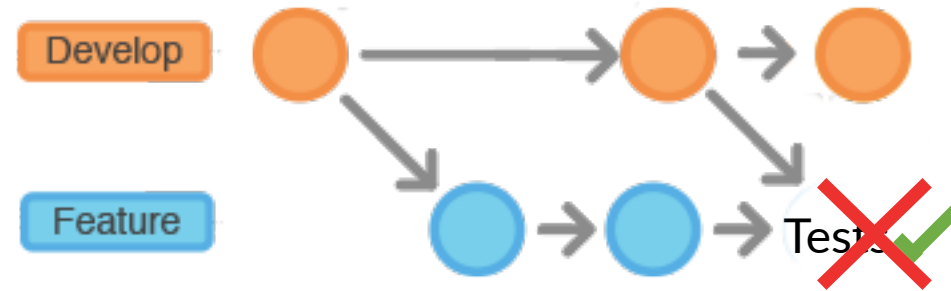
Интеграция тестов на CI



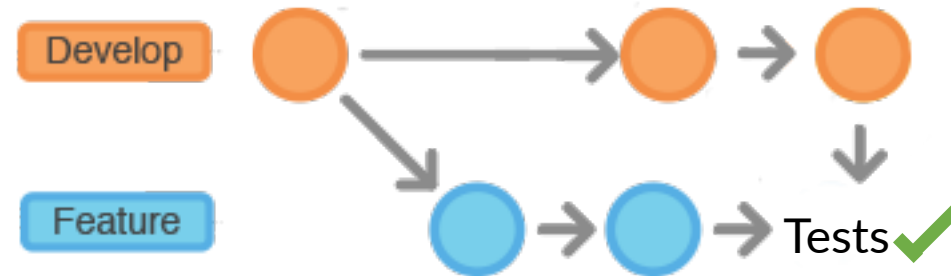
git merge origin/develop



git merge origin/develop



git merge origin/develop



Pull request flow в Avito

- Билды запускаются при открытии Pull Request и каждом последующем коммите
- Перед каждым тестом `git merge origin/develop`
- “Контрольный” запуск тестов перед merge в develop

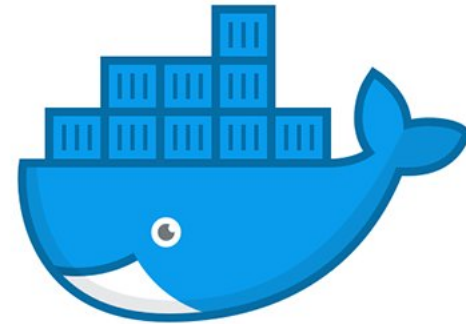
Причины хранить CI скрипты в VCS

- Версионирование
- Синхронизация изменений
- Тестирование изменений на ветке
- Локальная воспроизводимость

Настройка окружения



SALTSTACK ANSIBLE VS



puppet

Плюсы использования Docker

- Стабильность и повторяемость окружения
- Локальный запуск CI билдов
- Централизованное и контролируемое обновление
- Версионирование

Docker образы

- Builder
- Emulator
- DevOps

Builder Dockerfile

```
RUN wget https://dl.google.com/android/repository/sdk-tools-linux-3859397.zip -O android-sdk.zip
```

```
RUN unzip android-sdk.zip -d /opt/android-sdk
```

```
COPY android-sdk-packages /android-sdk-packages
```

```
RUN yes | sdkmanager --package_file=/android-sdk-packages
```

```
RUN yes | sdkmanager --licenses
```

Builder Dockerfile

```
RUN wget https://dl.google.com/android/repository/sdk-tools-linux-3859397.zip -O android-sdk.zip
```

```
RUN unzip android-sdk.zip -d /opt/android-sdk
```

```
COPY android-sdk-packages /android-sdk-packages
```

```
RUN yes | sdkmanager --package_file=/android-sdk-packages
```

```
RUN yes | sdkmanager --licenses
```

Builder Dockerfile

```
RUN wget https://dl.google.com/android/repository/sdk-tools-linux-3859397.zip -O android-sdk.zip
```

```
RUN unzip android-sdk.zip -d /opt/android-sdk
```

```
COPY android-sdk-packages /android-sdk-packages
```

```
RUN yes | sdkmanager --package_file=/android-sdk-packages
```

```
RUN yes | sdkmanager --licenses
```

Builder Dockerfile

```
RUN wget https://dl.google.com/android/repository/sdk-tools-linux-3859397.zip -O android-sdk.zip
```

```
RUN unzip android-sdk.zip -d /opt/android-sdk
```

```
COPY android-sdk-packages /android-sdk-packages
```

```
RUN yes | sdkmanager --package_file=/android-sdk-packages
```

```
RUN yes | sdkmanager --licenses
```

Emulator Dockerfile

```
RUN echo "no" | \  
    avdmanager create avd \  
    -name emulator_19 \  
    -package "system-images;android-19;google_apis;x86" \  
    -abi google_apis/x86
```

```
RUN mksdcard -l e 512M /sdcard.img
```

```
COPY hardware/config_19.ini \  
    $HOME/.android/avd/emulator_19.avd/config.ini
```

Emulator Dockerfile

```
RUN echo "no" | \  
    avdmanager create avd \  
    -name emulator_19 \  
    -package "system-images;android-19;google_apis;x86" \  
    -abi google_apis/x86
```

```
RUN mksdcard -l e 512M /sdcard.img
```

```
COPY hardware/config_19.ini \  
    $HOME/.android/avd/emulator_19.avd/config.ini
```

Emulator Dockerfile

```
RUN echo "no" | \  
    avdmanager create avd \  
    -name emulator_19 \  
    -package "system-images;android-19;google_apis;x86" \  
    -abi google_apis/x86
```

```
RUN mksdcard -l e 512M /sdcard.img
```

```
COPY hardware/config_19.ini \  
    $HOME/.android/avd/emulator_19.avd/config.ini
```

gradle.properties

```
org.gradle.jvmargs=-Xmx8192m
```

```
org.gradle.daemon=false
```

```
org.gradle.caching=true
```

```
org.gradle.parallel=true*
```

```
org.gradle.workers.max=8
```

```
*broken in Kotlin 1.2.20 issue KT-22513
```

Билды на PR

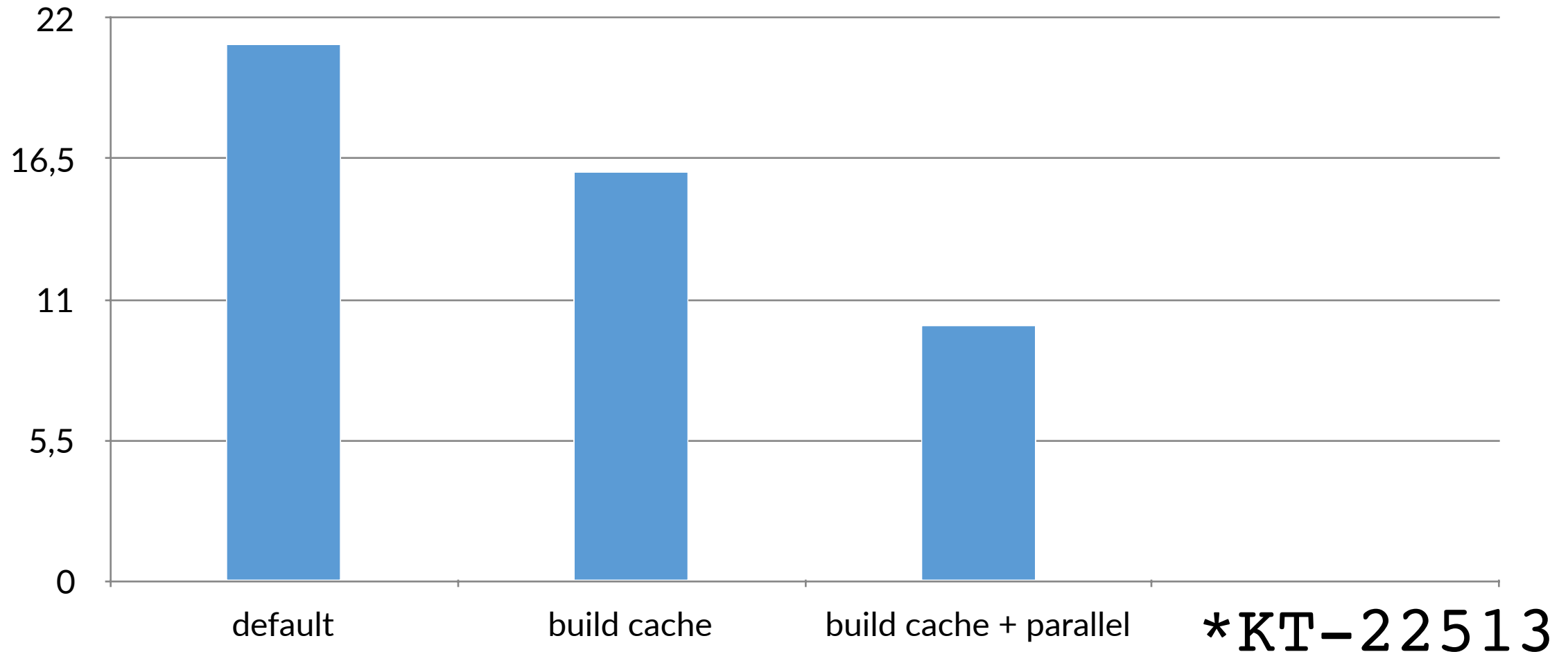
- Lint
- Build release
- Unit tests
- UI tests

Android lint

- он есть (:

Build release

Gradle build duration time, minutes

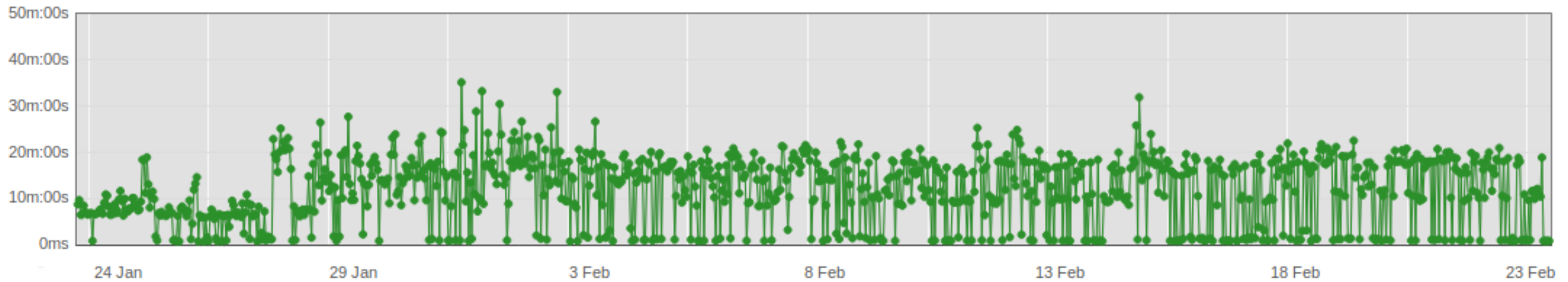


*KT-22513

Unit tests

- ✓ Tests passed: 10686, ignored: 43 | ▾ (22m:15s)
- ✓ Tests passed: 11125, ignored: 43 | ▾ (24m:29s)
- ✓ Tests passed: 11186, ignored: 69 | ▾ (21m:18s)
- ✓ Tests passed: 3 | ▾ (56s)
- ✓ Tests passed: 11099, ignored: 43 | ▾ (19m:47s)
- ✓ Tests passed: 5856, ignored: 37 | ▾ (14m:26s)
- ✓ Tests passed: 4845, ignored: 11 | ▾ (12m:23s)
- ✓ Tests passed: 10621, ignored: 43 | ▾ (21m:33s)
- ✓ Tests passed: 3 | ▾ (1m:05s)
- ✓ Tests passed: 3 | ▾ (1m:02s)

Unit tests



UI tests

- Не запускаем автотесты на железных устройствах
- Отказались от Firebase
- Тестируем на AVD эмуляторах

Цели на будущее

- impact-анализ в UI тестах
- gradle remote cache

Итоги

- Тестирование на CI это всегда компромисс между полнотой проверок и затрачиваемыми ресурсами
- Не делать карго-культ из CI/CD практик
- Пробовать что-то новое

Спасибо!