

# Мониторим здоровье gradle-сборки KMP-проекта

Кирилл Старостин  
Senior KMP SE, X5 Tech



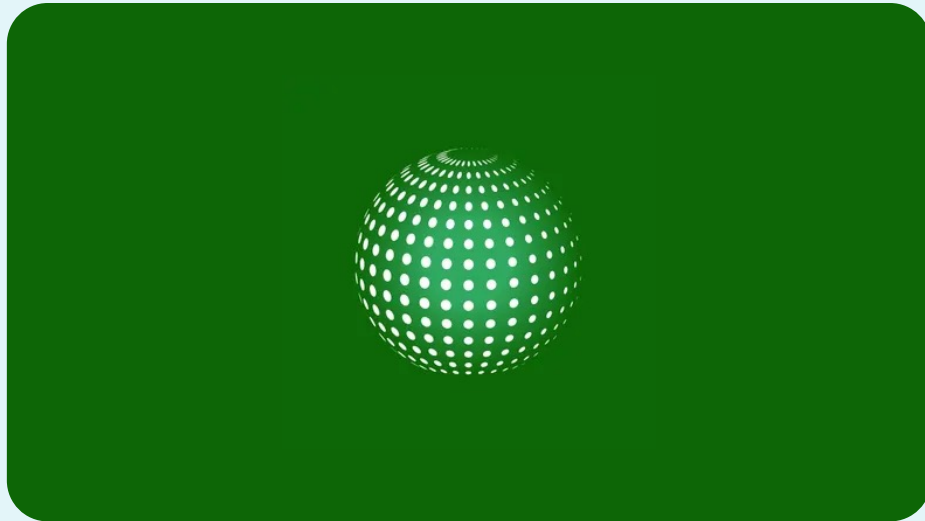
# Кто в проде на KMP?



▼ ⚠ app: finished 36 min, 14 sec, 336 ms

↓ Download info

⚠ Unused Kotlin Source Sets The K



# Наш контекст

**1 млн**  
строк кода

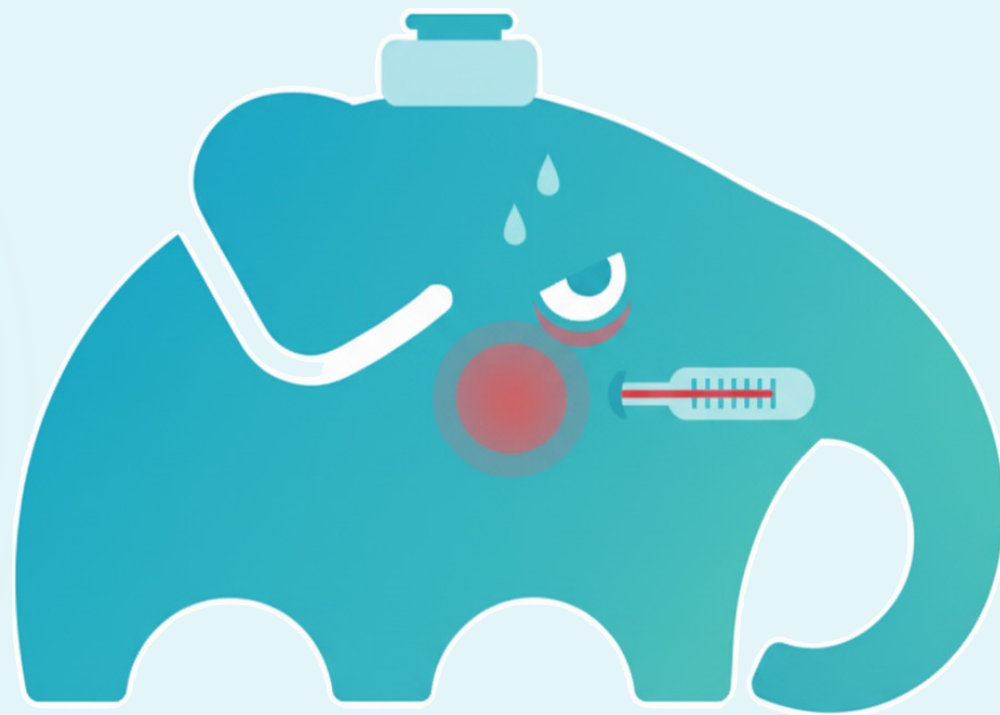
**20**  
инженеров

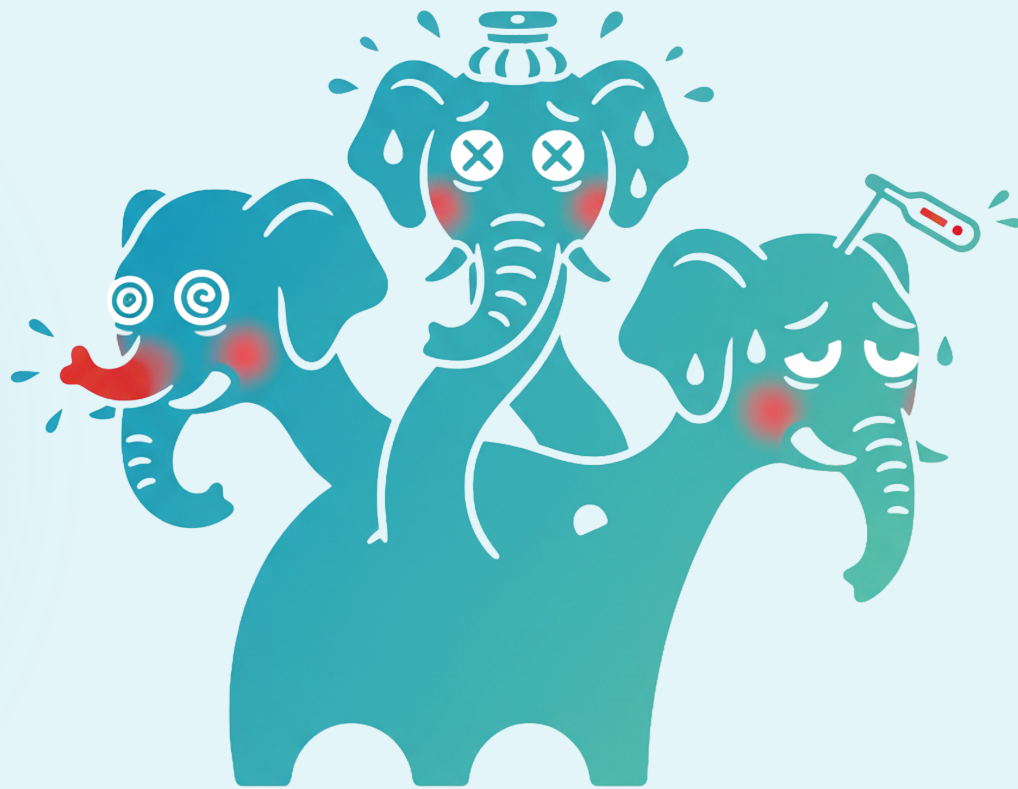
**13 мин**  
IDE sync

KMP CMP  
Android  
и iOS,  
99% reuse

Кэши  
включены

**3,5 мин**  
android debug





# Ускорились в 2 раза

Настроили мониторинг  
здоровья КМР сборки



# Ускорились в 2 раза

Настроили мониторинг  
здоровья КМР сборки

IDE sync 13 мин → 8 мин → ???



# Ускорились в 2 раза

Настроили мониторинг  
здоровья KMP сборки

IDE sync 13 мин → 8 мин → ???

Android debug 3,5 мин → 1,5 мин



# Экономим бизнесу миллионы

Сэкономили 7 мин синка,  
2 мин android debug



# Экономим бизнесу МИЛЛИОНЫ

Скостили **7 мин** синка,  
**2 мин** android debug

20 сборок, 5 синков в день =  
1ч 15 м разраб/день



# Экономим бизнесу МИЛЛИОНЫ

Скостили **7 мин** синка,  
**2 мин** android debug

20 сборок, 5 синков в день =  
1ч 15 м разраб/день

247 дней x 20 разрабов = 6850 ч/г



# Экономим бизнесу МИЛЛИОНЫ

Скостили **7 мин** синка,  
**2 мин** android debug

20 сборок, 5 синков в день =  
**1ч 15 м** разраб/день

247 дней x 20 разрабов = **6850 ч/г**

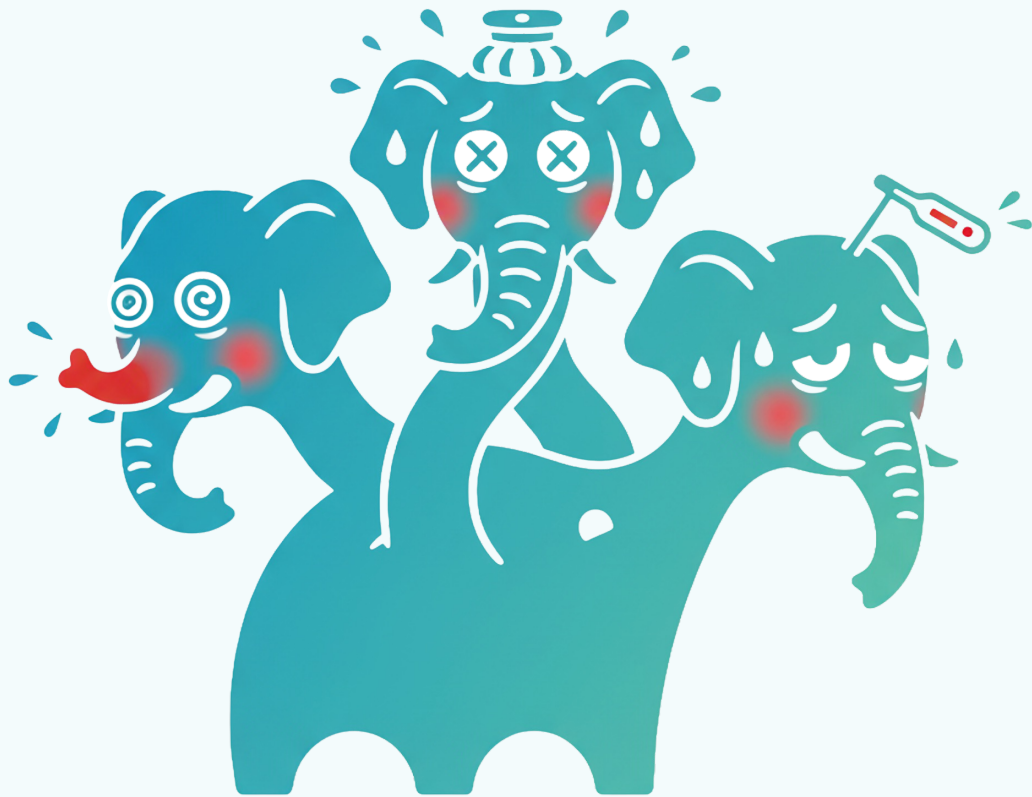
6850 x 2000 руб/ч = **13 700 000 ₽/г**

# О чём расскажем

- ✓ Как мониторить КМР-билды
- ✓ Проект-специфичные кейсы
- ✗ Магическая таблетка ускорения для любого проекта



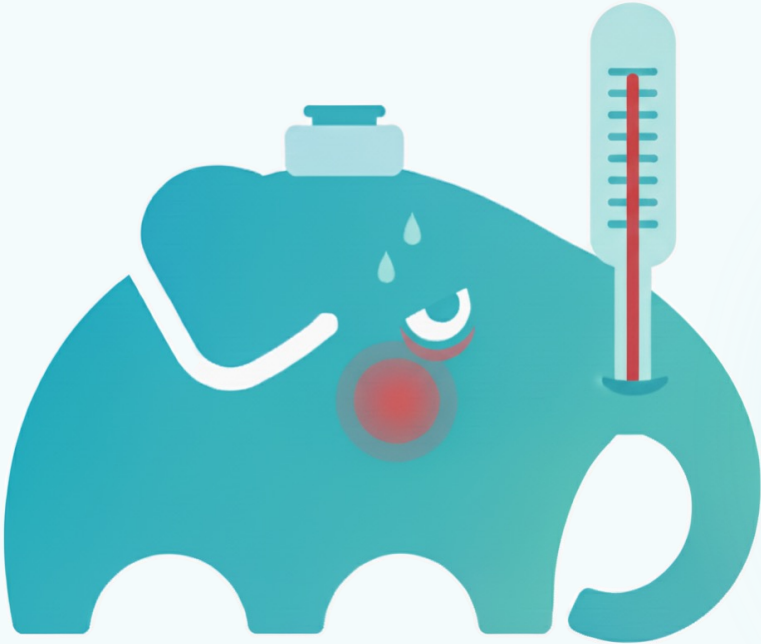
# Получи повышение

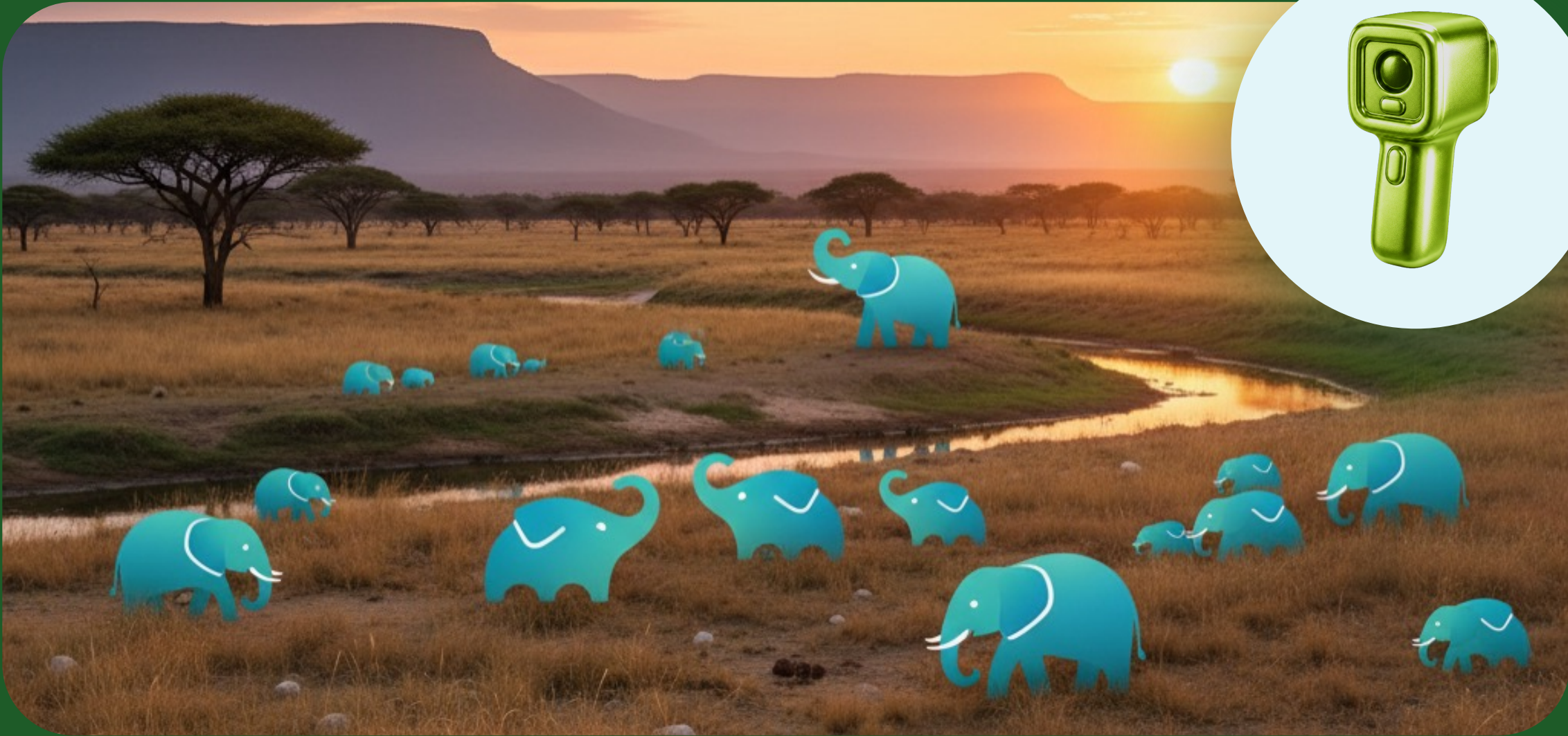




# Как измерять?

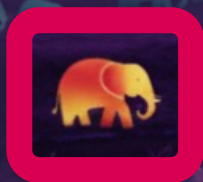
# Бенчмарк







# Метрики



Build Duration

90th Percentile Duration

6.60 mins

Average Build Duration

2.56 mins

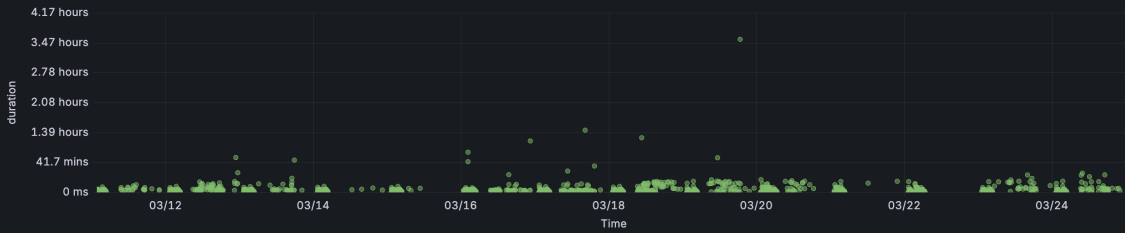
Median Build Duration

38.0 s

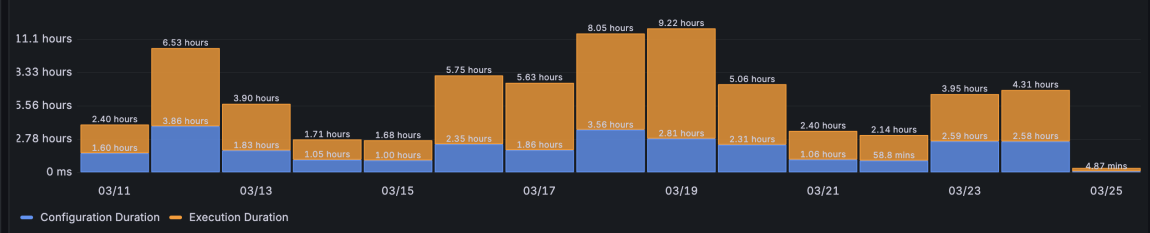
Total Build Duration

3.86 days

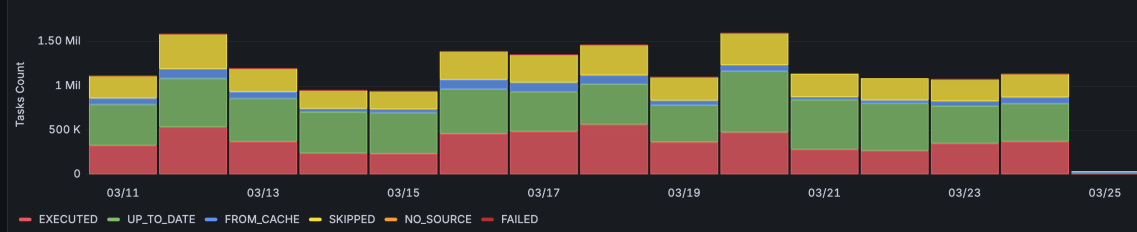
Build Duration Scatter Plot



Daily Build Configuration vs Execution Duration



Daily Task States Distribution



# Профайлер



**Бенчмарк**

**Профайлер**

**Метрики**

# Что измерять?

# KMP-сборка

## Android

Debug

Clean

QA

Incremental

Release

## iOS

Debug

Clean

Release

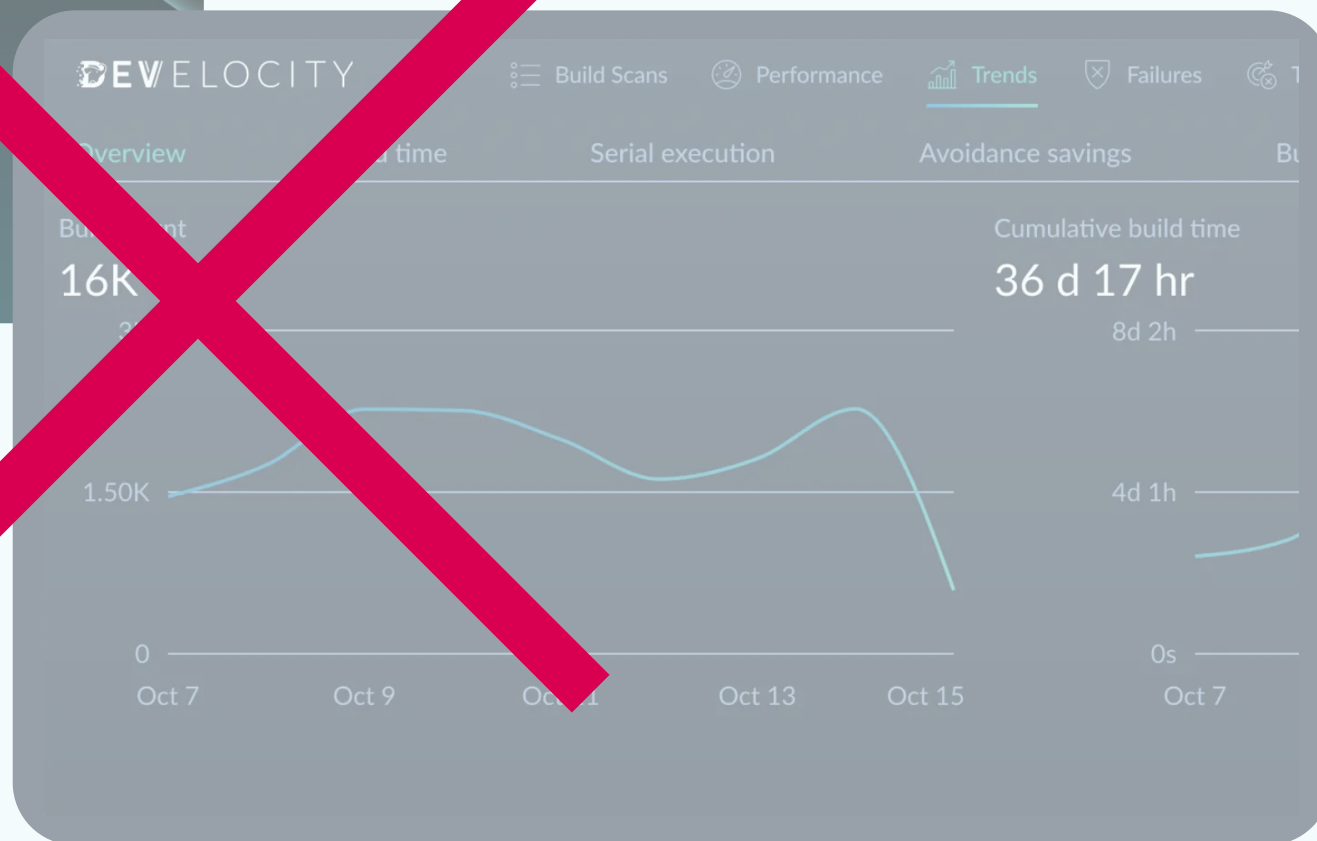
Incremental

Android Studio Sync

JVM

JS





# Как тогда измерять?

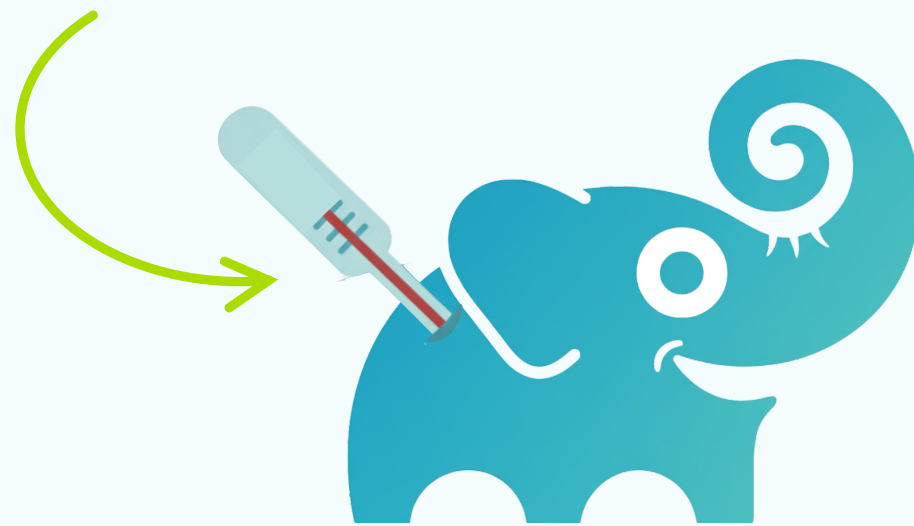
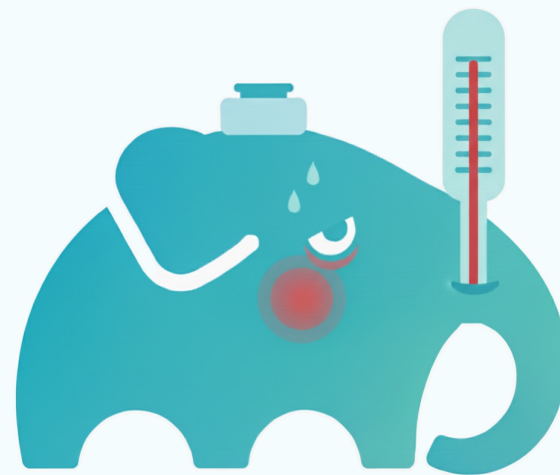
**Бенчмарк**

**Профайлер**

**Метрики**



Надёжно  
оцениваем  
тренд  
В ОДИН КЛИК



# Ручные замеры

Бенчмарк

## Android debug clean

- no cache
- cache

## Android debug incremental

- no change
- change, config cache
- no config cache

## IDE Sync

—

**3 раза** → мало, шум, муторно

**10 раз** → долго, невыносимо

# А как хотим?

Бенчмарк

1 клик

30 мин – 1 час

Надёжный  
вердикт:  
хуже или лучше

Автопрогон

1

Автопрогон

1

Стабильное  
окружение

2

# Бенчмарк

Бенчмарк

Автопрогон

1

Стабильное  
окружение

2

Стандартный  
формат

3

gradle / **gradle-profiler** Public

<> Code Issues 95 Pull requests 20

★ Star 1.5k

## About

A tool for gathering profiling and benchmarking information for Gradle builds

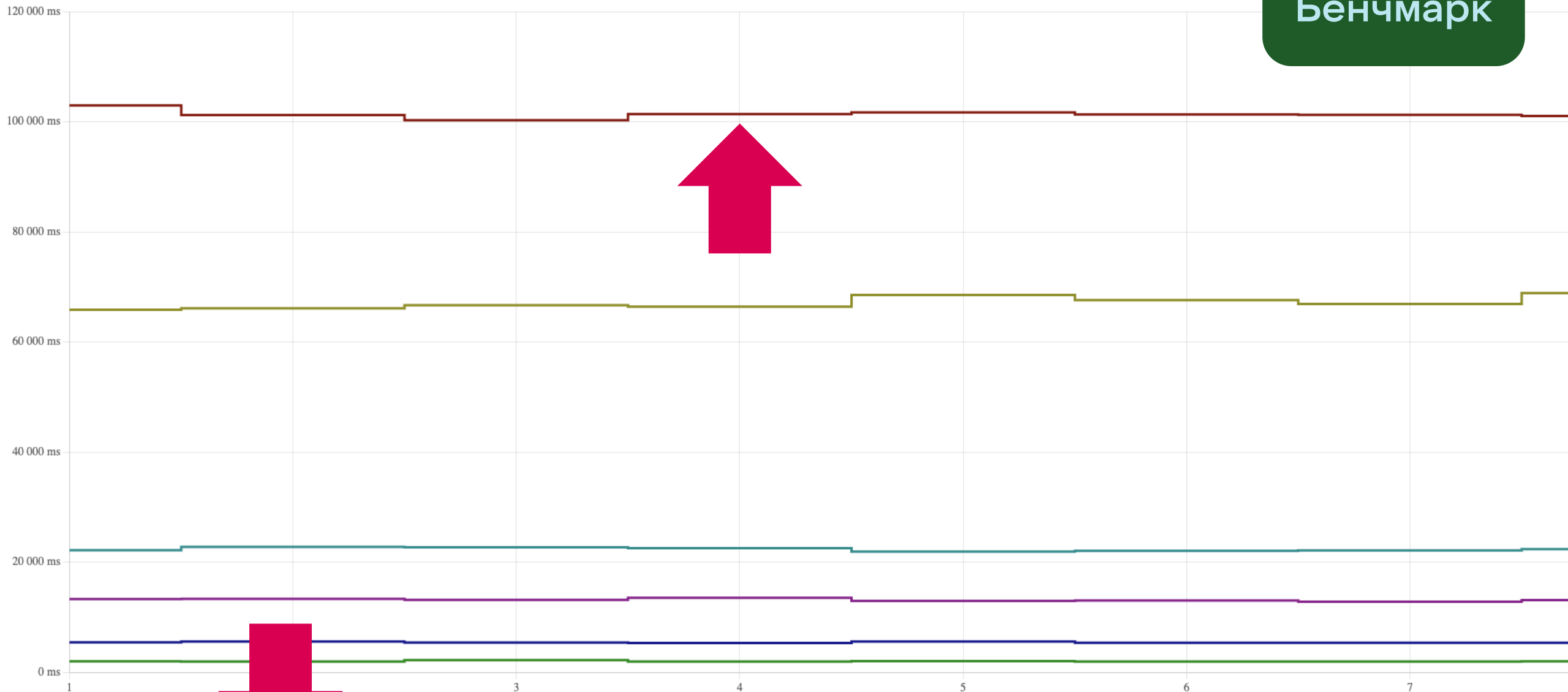
[gradle-bt](#)

[gradle-bt-core-execution](#)

```
11 android_debug_fast = [  
12     "android_debug_clean_real_worst_baseline",  
13     "android_debug_clean_both_caches",  
14     "android_debug_incremental_no_change_best_baseline",  
15     "android_debug_incremental_no_change_config_cache_disabled",  
16     "android_debug_incremental_5files_abi_10files_nonabi",  
17     "minimal_pure_config_time",  
18 ]
```

```
37 android_debug_incremental_no_change_config_cache_disabled {
38     tasks = ["app:assembleDebug"]
39     gradle-args = [
40         "--build-cache",
41         "--no-configuration-cache",
42         "--parallel",
43     ]
44     daemon = warm
45     warm-ups = 6
46     iterations = 10
47 }
```

```
gradle-profiler --benchmark \  
--project-dir . \  
--gradle-user-home ${PATH_FOR_GRADLE}/ \  
--output-dir ${PATH_FOR_REPORTS} \  
--group ci_android_debug_fast \  
--scenario-file platform/plugins/build-metrics/
```



Historical Sorted Select all Clear all Show all Show selected only

Scenario	Baseline	Sample	Mean	Min	P25	Median	P75	Max	Std.dev	Iterations
+ android_debug_clean_real_worst_baseline	<input type="checkbox"/>	total execution time	101 527,88 ms	100 325,09 ms	101 132,47 ms	101 315,41 ms	101 658,51 ms	103 016,61 ms	769,88 ms	101 636,47 ms 101 739,19 ms 102 492,95 ms
+ android_debug_clean_both_caches	<input type="checkbox"/>	total execution time	67 267,16 ms	65 874,84 ms	66 230,25 ms	66 824,96 ms	68 341,86 ms	69 250,24 ms	1 184,92 ms	66 638,43 ms 67 245,66 ms 69 044,67 ms

# Сравнение двух прогонов

Бенчмарк

Прогон ДО

benchmark\_1.csv

Изменение

Прогон ПОСЛЕ

benchmark\_2.csv

## Allow comparison of 2 different executions of gradle profiler #399

Open



yogurtearl opened on Jan 29, 2022

```
gradle-profiler --compare yesterday_benchmark.csv today_benchmark.csv
```



asodja on Feb 2, 2022 · edited by asodja

Edits ▾

Member



That would be a nice feature indeed. I doubt that we will work on it now since there are other priorities in gradle/gradle atm. But you can provide a PR if you are interested.

# Кастомный скрипт

Coefficient of Variation



stable?



visible?



different?



how much?



# Кастомный скрипт

Coefficient of Variation

Signal-to-noise ratio

Бенчмарк



stable?



visible?



different?



how much?



# Кастомный скрипт

Coefficient of Variation

Signal-to-noise ratio

Mann-Whitney U test



stable?



visible?



different?



how much?



# Кастомный скрипт

Coefficient of Variation

Signal-to-noise ratio

Mann-Whitney U test

Bootstrap Confidence Intervals



stable?



visible?



different?



how much?



```
/compare_two_benchmarks_trend.py 1_ANTIVIR_TESTS_CORE_DISABLED/benchmark.csv 0_ANTIVIR_TESTS_ALL_ENABLED/benchmark.csv
```

## TREND ANALYSIS RESULTS:

Scenario	Pts	Base CV	New CV	SNR	Base Med	New Med	Med Diff
android_debug_clean_both_caches	10/10	✓ 0.4%	✓ 0.5%	✓✓ 63.19	1m 34s	1m 10s	-24.9% (-23.40 s)
android_debug_clean_real_worst_baseline	10/10	✓ 1.0%	✓ 3.9%	✓✓ 14.30	3m 14s	2m 4s	-36.1% (-1m 10s)
android_debug_incremental_5files_abi_10files_nonabi	10/10	✓ 1.7%	✓ 3.2%	△△ 0.36	8.79 s	8.69 s	-1.2% (-0.10 s)
android_debug_incremental_no_change_best_baseline	10/10	✓ 2.7%	✓ 1.9%	✓✓ 49.51	4.67 s	2.38 s	-49.0% (-2.28 s)
android_debug_incremental_no_change_config_cache_disabled	10/10	✓ 2.7%	✓ 2.6%	✓✓ 34.81	34.06 s	17.85 s	-47.6% (-16.21 s)
minimal_pure_config_time	10/10	✓ 3.8%	✓ 1.9%	✓✓ 57.99	24.73 s	11.72 s	-52.6% (-13.00 s)

## Med Confidence Interval

## Mann-Whitney U

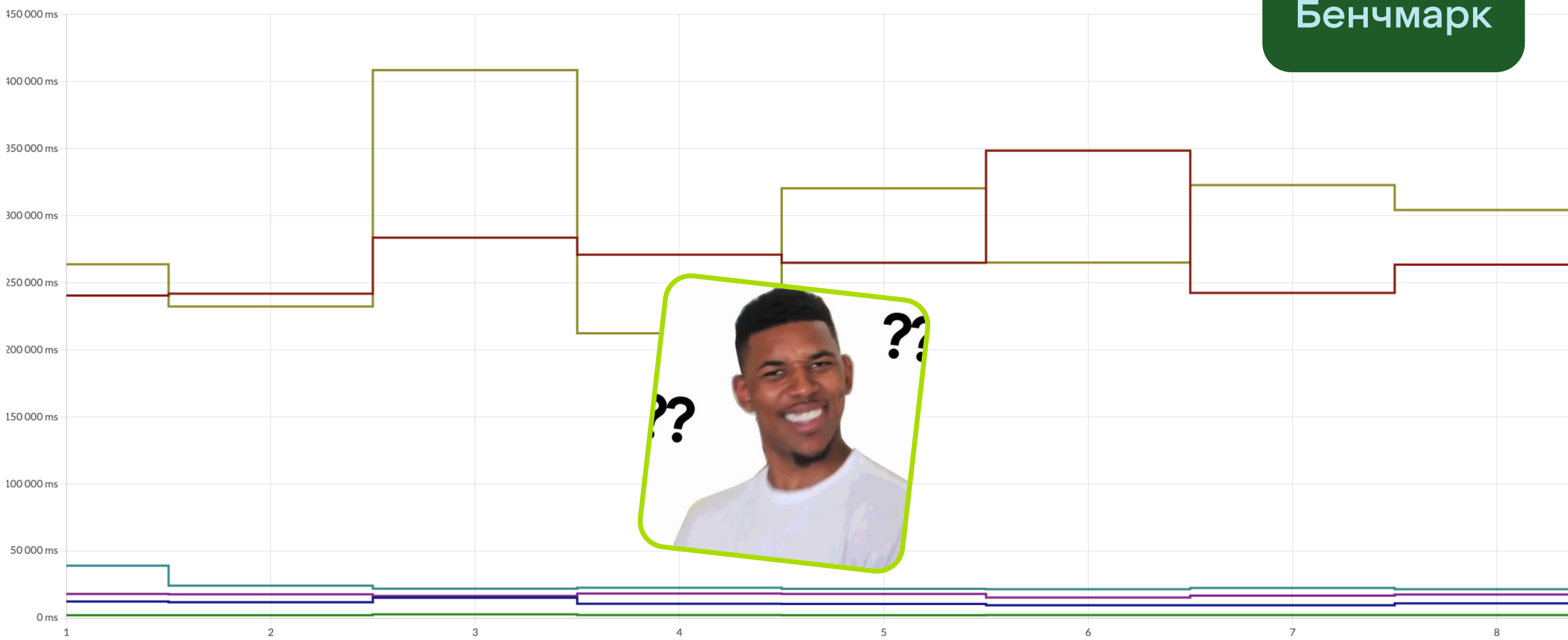
-25.3%, -24.4% (-23.84 s, -22.92 s)	HIGHLY SIGNIFICANT (p=0.0002)
-37.0%, -33.9% (-1m 12s, -1m 5s)	HIGHLY SIGNIFICANT (p=0.0002)
-3.2%, +2.0% (-0.28 s, 0.18 s)	NOT SIGNIFICANT (p=0.5205)
-51.0%, -45.9% (-2.38 s, -2.14 s)	HIGHLY SIGNIFICANT (p=0.0002)
-51.0%, -45.2% (-17.39 s, -15.40 s)	HIGHLY SIGNIFICANT (p=0.0002)
-54.6%, -51.8% (-13.50 s, -12.82 s)	HIGHLY SIGNIFICANT (p=0.0002)

## DECISION

✓ IMPROVE: 24.9% faster (24.4%, 25.3%)
✓ IMPROVE: 36.1% faster (33.9%, 37.0%)
△ UNCLEAR: weak signal (SNR 0.36)
✓ IMPROVE: 49.0% faster (45.9%, 51.0%)
✓ IMPROVE: 47.6% faster (45.2%, 51.0%)
✓ IMPROVE: 52.6% faster (51.8%, 54.6%)



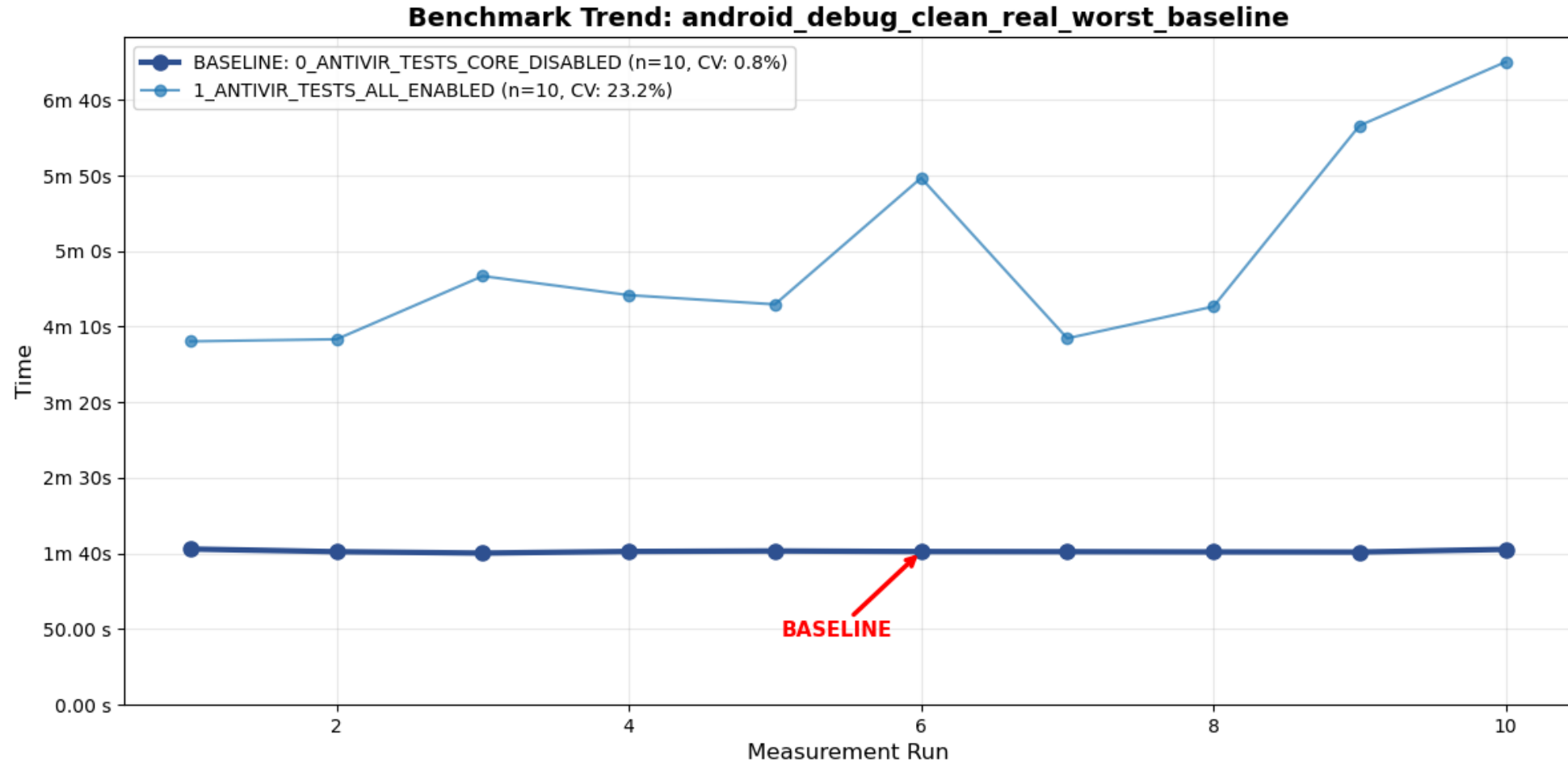
(с) «Как мы ускорили сборку селлера», Habr, Ozon, 2025

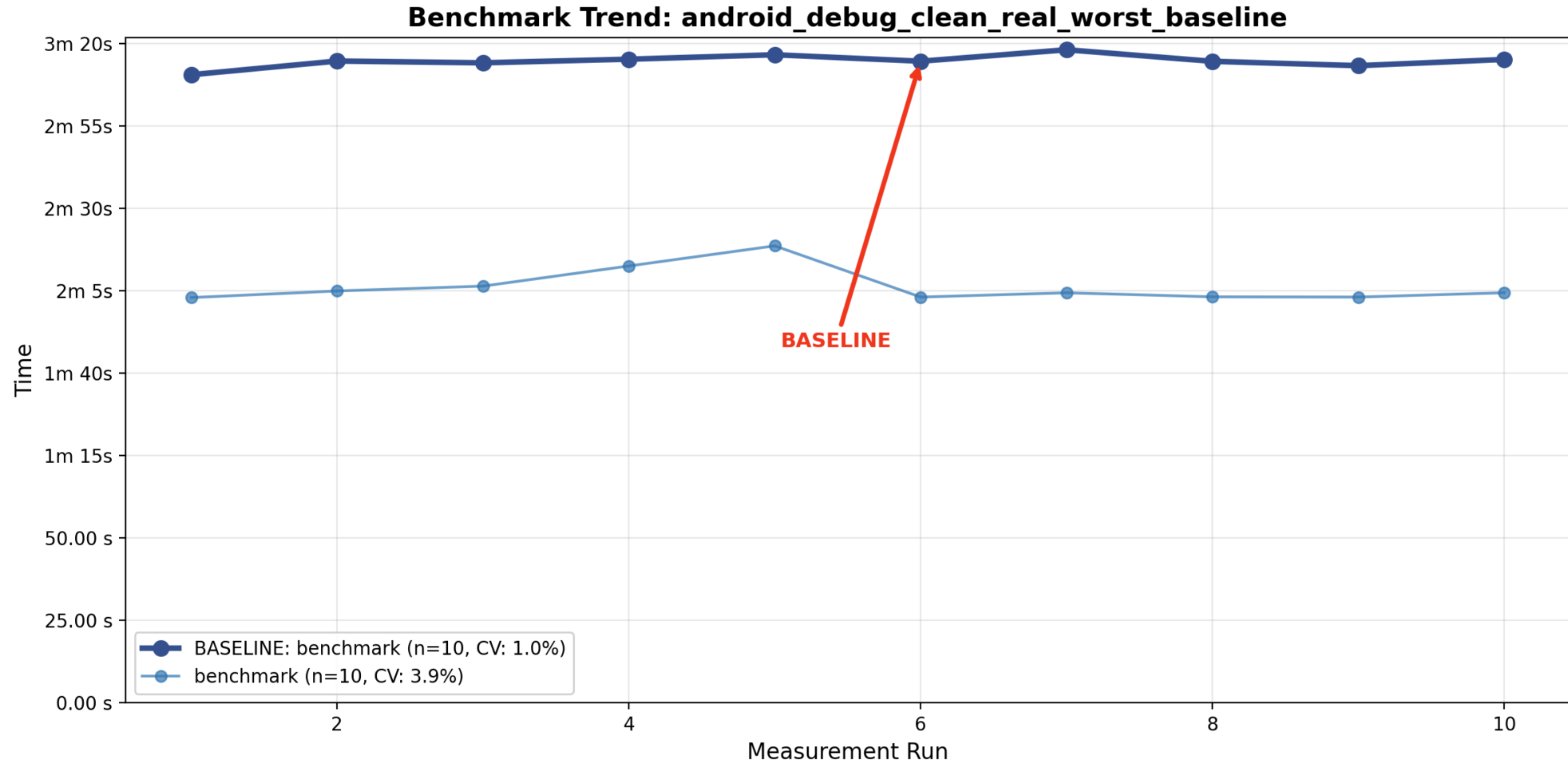


Historical Sorted Select all Clear all Show all Show selected only

# ✓ Антивирус, 2х–3х скорость

Бенчмарк





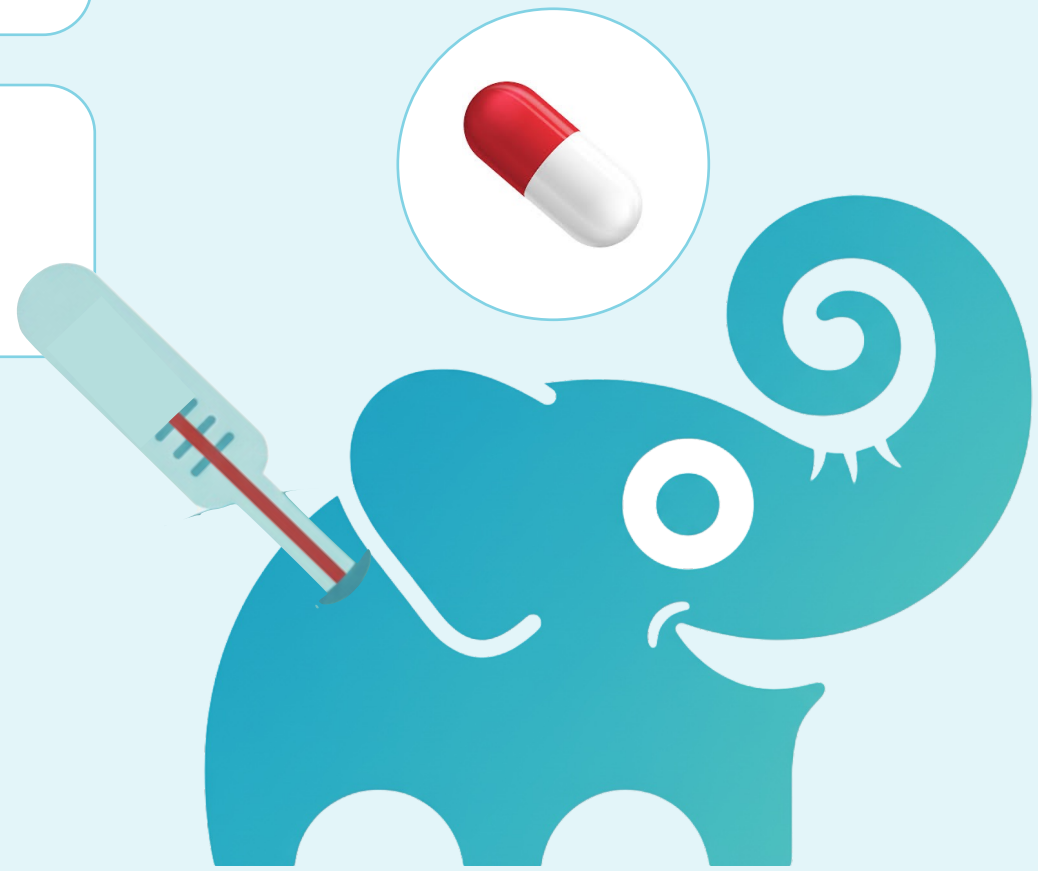
```
4  androidStudioSync {
5      title = "Android Studio Sync"
6      android-studio-sync {
7          studio-jvm-args = ["-Xms256m", "-Xmx24G"]
8      }
9      warm-ups = 3
10     iterations = 5
11 }
```

# Итого

Бенчмарк

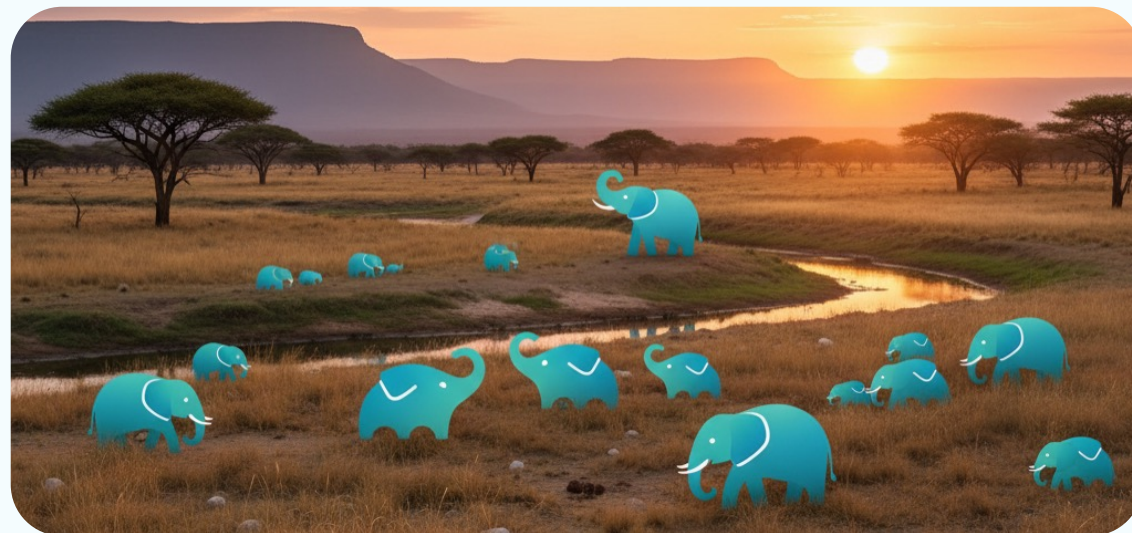
✓ 1 клик =  
все измерения

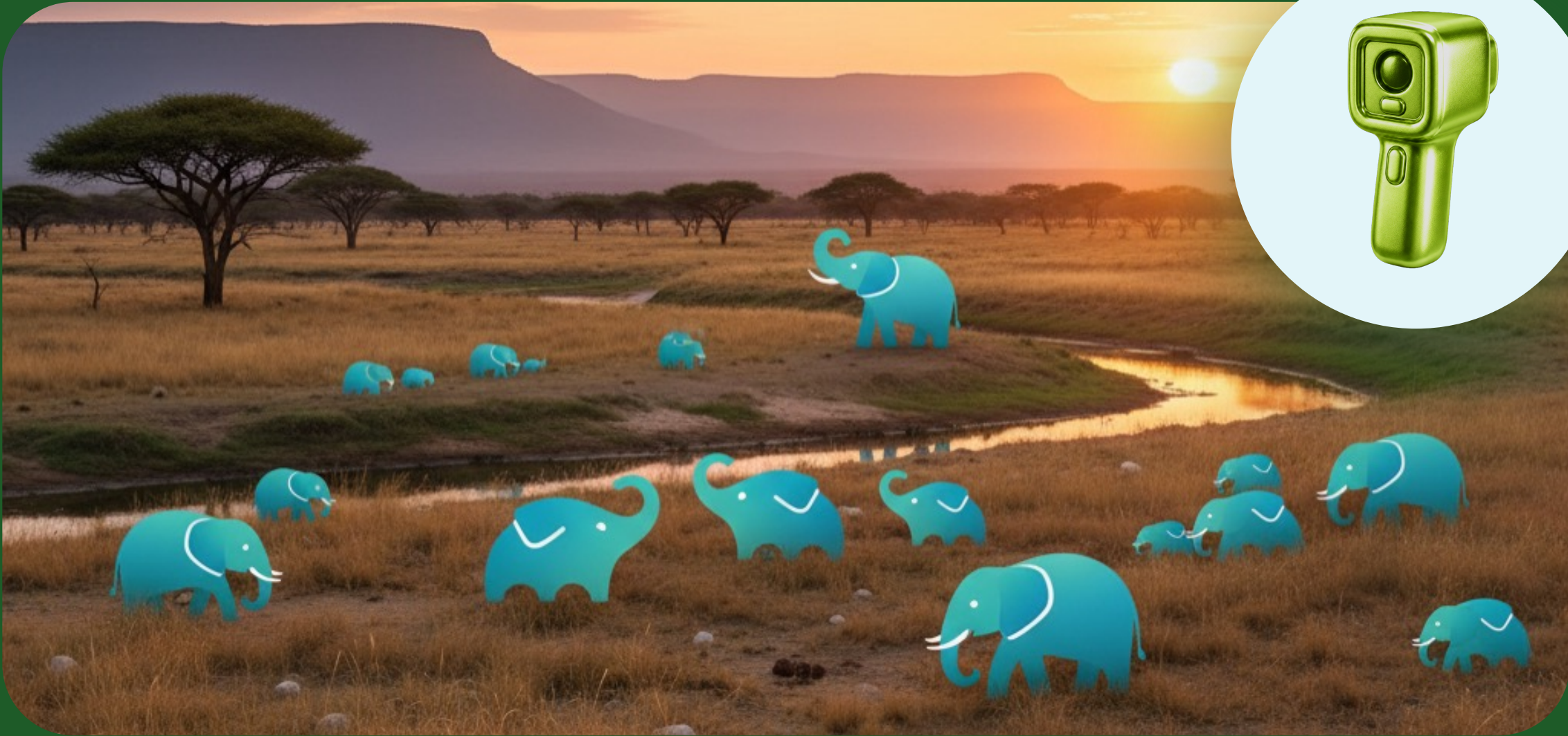
✓ 2 клик =  
надёжная оценка тренда



2

Собираем  
среднее  
по больнице







# Что хотим?

## Авто-сбор:

- метрик билдов с машин разработов
- с CI

Легко  
визуализировать

Масштабируется

# Avito Build Metrics?



Avito Android




## Warning



This plugin is internal and not recommended to use.  
See alternatives: [Gradle Enterprise](#), [Talaiot](#)



(с) «Как мониторить здоровье вашей Gradle-сборки»,  
Habr, Боиштян (Авито), 2023

# T-Банк Build Metrics?

Метрики

**B** **build-metrics-plugin** 

 master  build-metrics-plugin


 **Project import generated by Copybara.**   
Copybara authored 3 months ago

**Project information**

An extensible Gradle plugin that designed to collect information of your Gradle build

---

Версия **none**

 **Star** **3**

# Talaiot?

Метрики

 **cdsap / Talaiot** Public

 **Code**  **Issues** 22  **Pull requests** 12

## About

Simple and extensible plugin to track task times in your Gradle Project.

kotlin

gradle

kotlin-library

**Releases** 46

 **v2.1.0** Latest  
on Nov 15, 2025

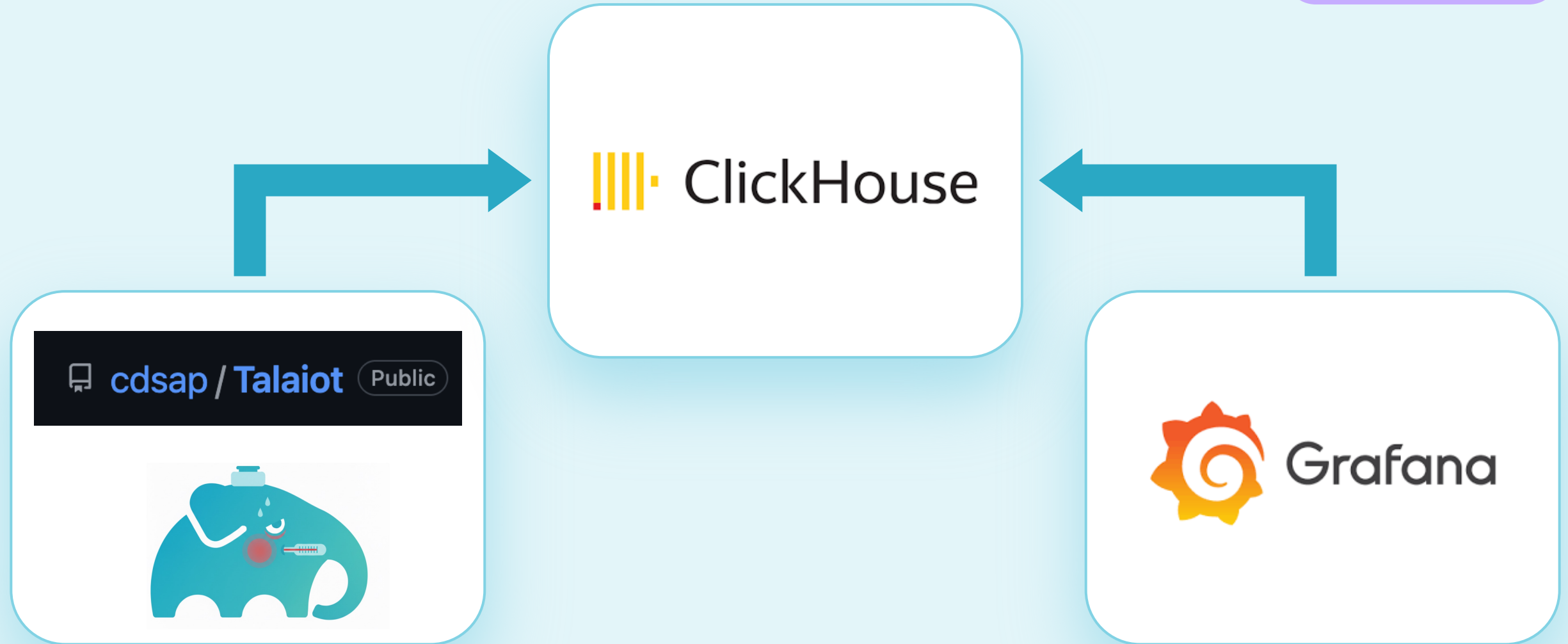
+ 45 releases



**Star** 617

# Talaiot

Метрики



# Подключаем Talaiot

```
val talaiotExtension = project.extensions.findByType(  
    type = TalaiotPluginExtension::class.java,  
)  
talaiotExtension?.apply {  
    filterBuilds()  
    enableMetrics(project, extension)  
    enablePublishers(extension, project)  
    initLogger(extension)  
}
```

# Настраиваем ClickHouse

Метрики

Кастомный  
публишер  
Talait

1

# Настраиваем ClickHouse

Метрики

Кастомный  
публишер  
Talaiot

1

Podman +  
AI-assistance

2

# Настраиваем ClickHouse

Метрики

Кастомный  
публишер  
Talaiot

1

Podman +  
AI-assistance

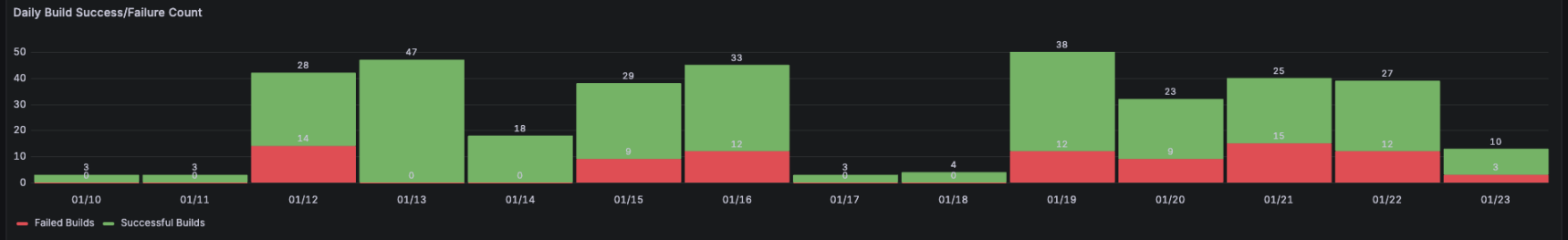
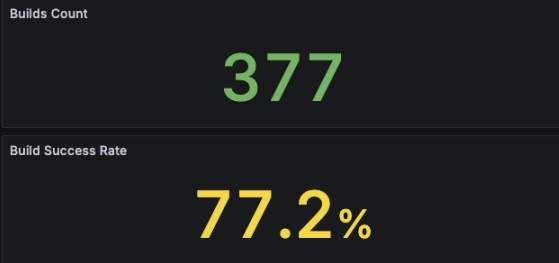
2

DevOps

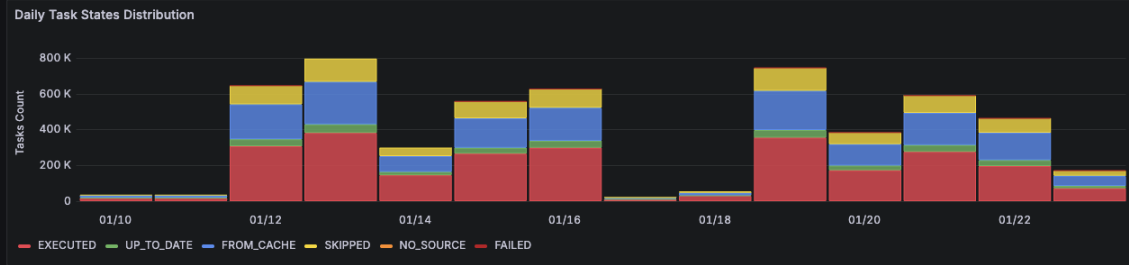
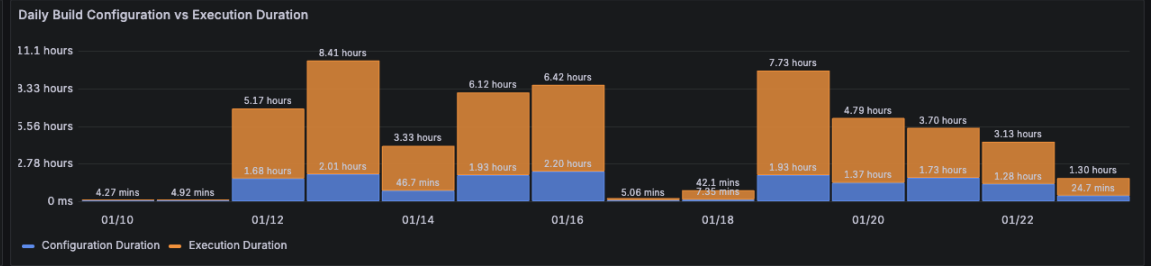
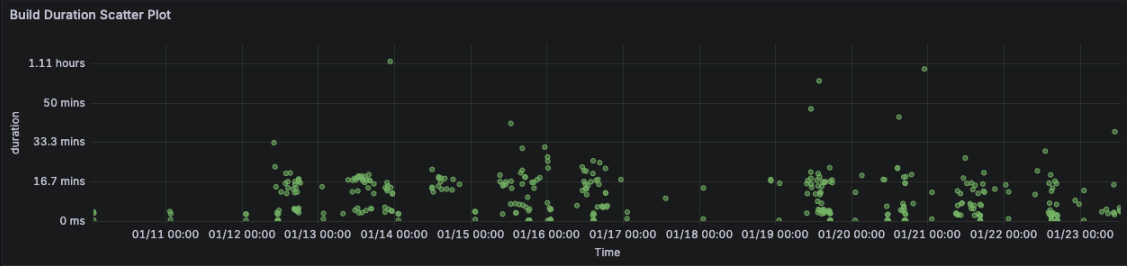
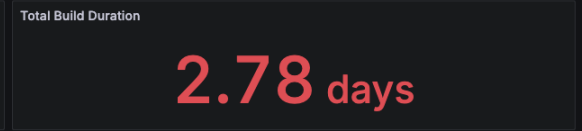
3

Root Project All Runner ci Build Type All

## Builds Count & Success



## Build Duration



> Build Cache (3 panels)

Build ID `b4c73258-5e9b-42d6-88e3-7`

[Back to Build Tree](#)

## Build Summary

Build ID	Timestamp	Status	Duration	Config Cache
<code>b4c73258-5e9b-42d6-88e3-7ce9c6cc030e</code>	2026-03-25 03:29:49	✓	Total Duration <b>7.33 mins</b>	✗

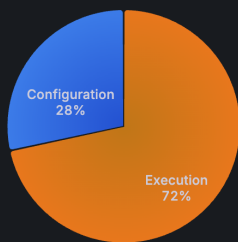
Root Project	<b>Aura</b>	Requested Tasks	<b>:app:androidApp:assemble</b>
--------------	-------------	-----------------	---------------------------------

> Environment Details *(5 panels)*

> Gradle Configuration *(12 panels)*

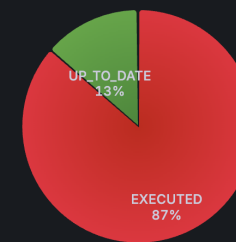
## Duration Breakdown

### Duration Split

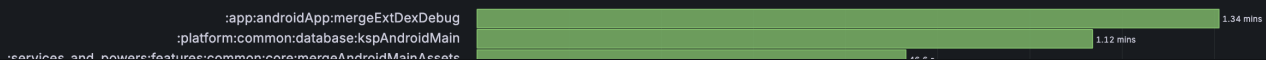


	Value	Percent
Execution	5.25 mins	72%
Configuration	2.08 mins	28%

### Actionable Tasks



### Top Slowest Specific Tasks

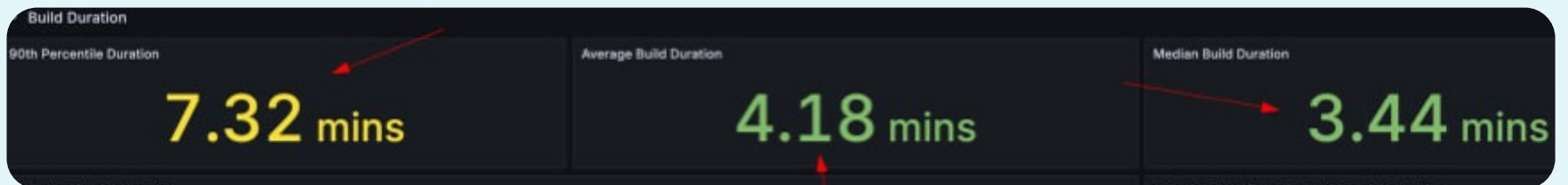
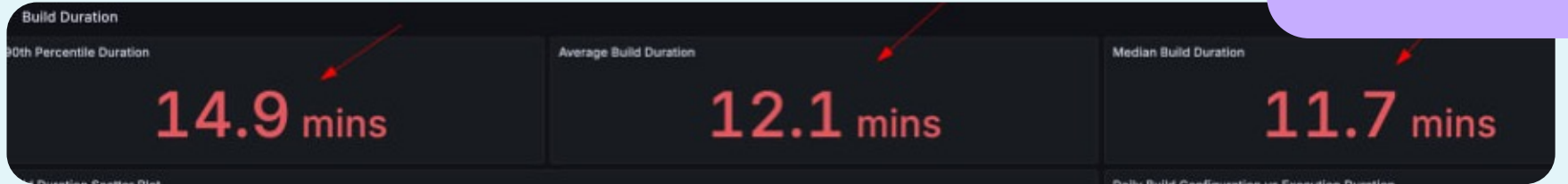


### Full Task States Breakdown

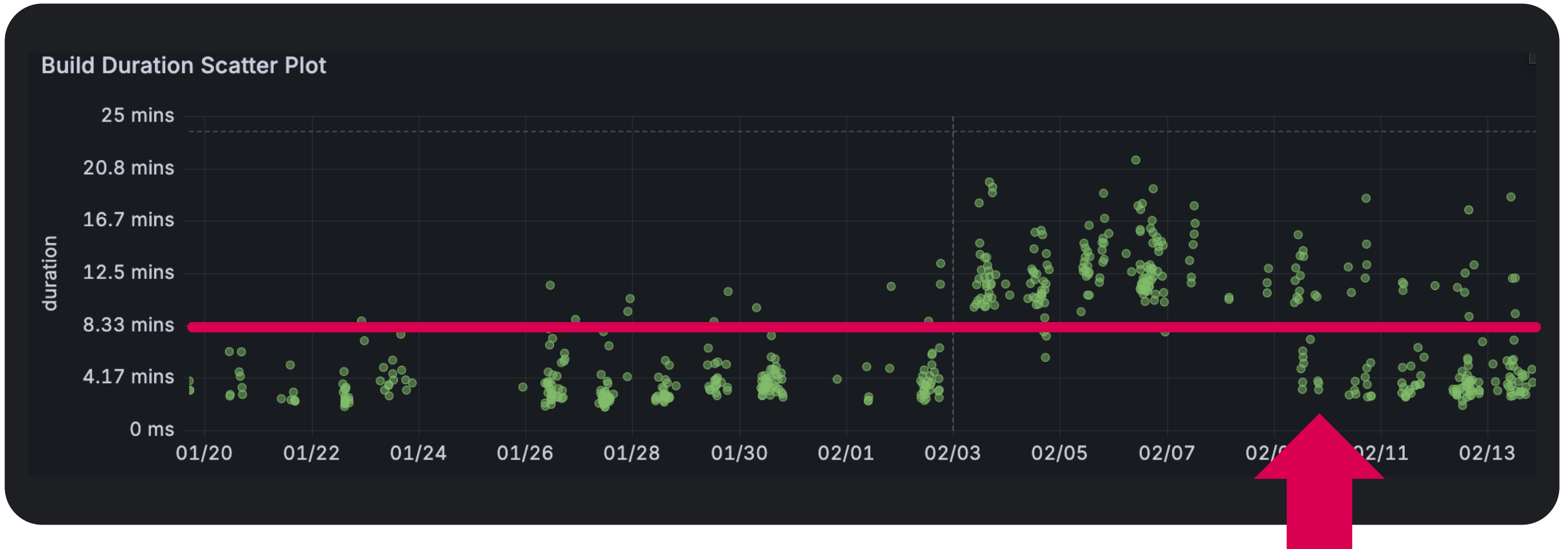


# Дело №2

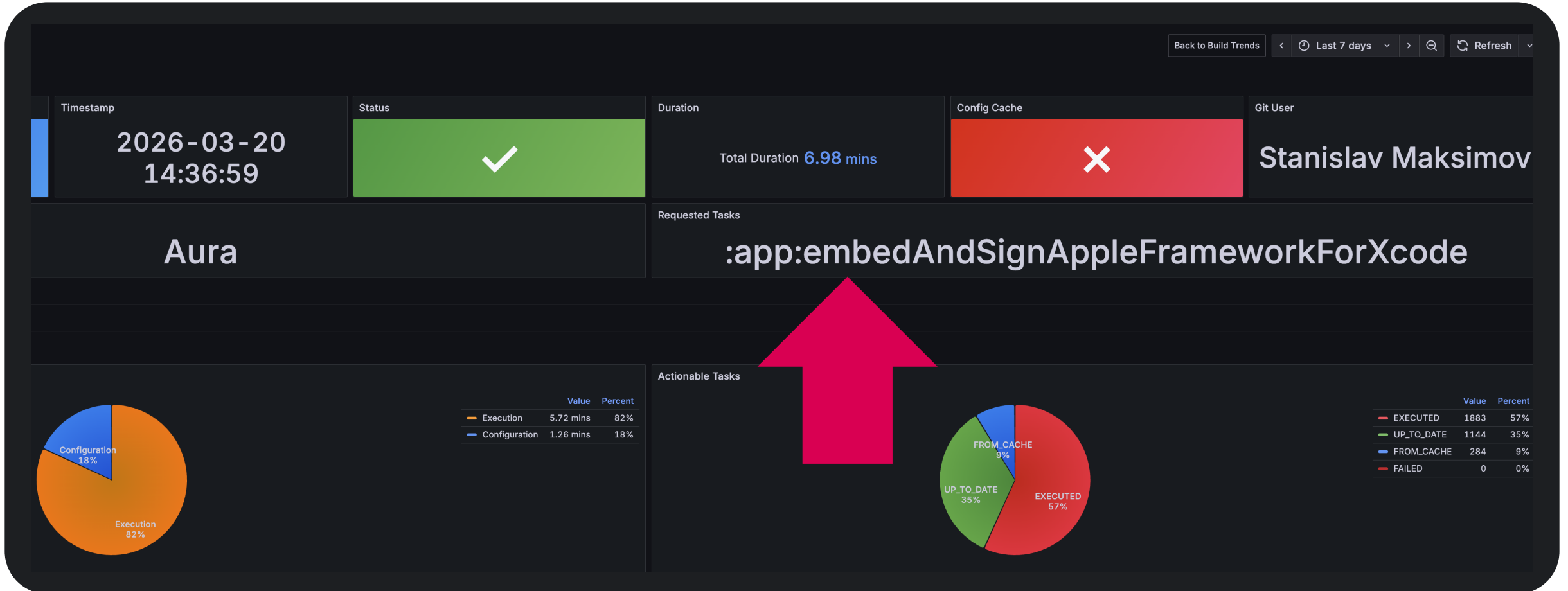
Метрики

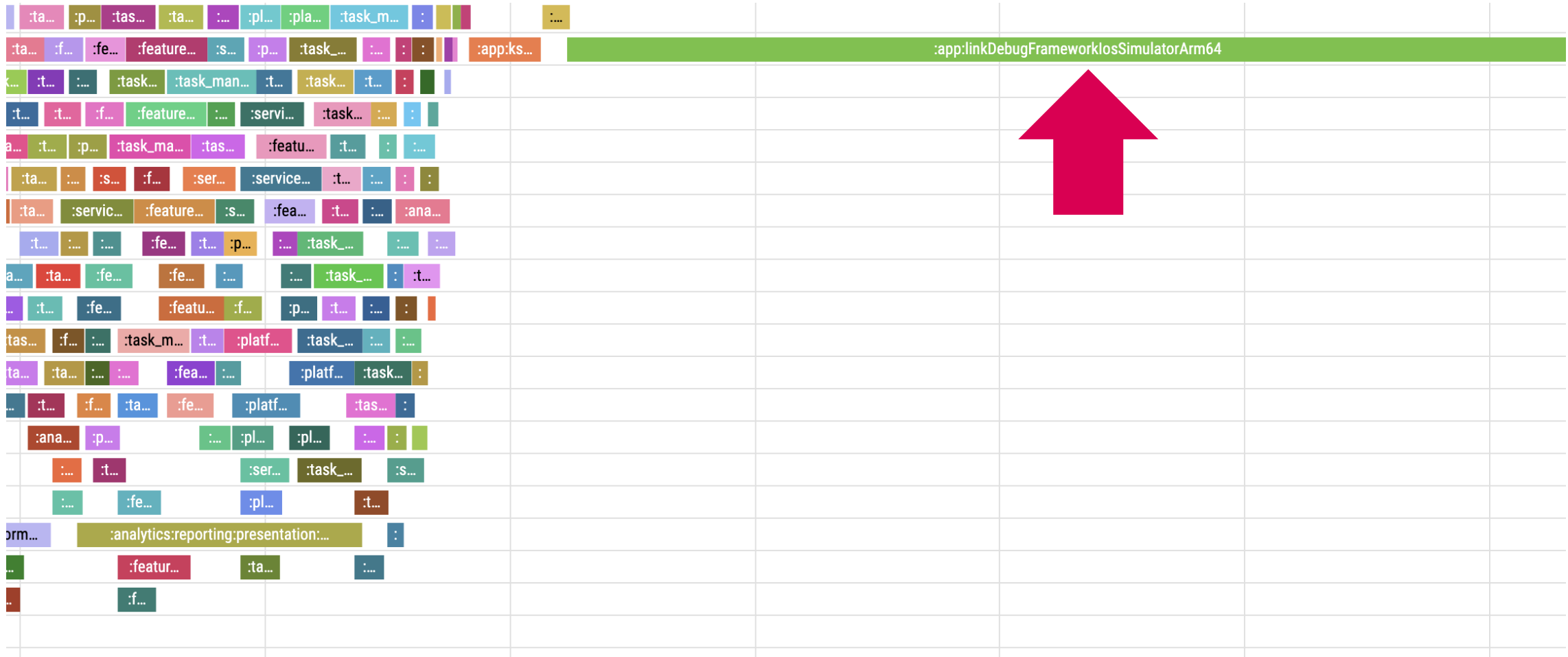


# Отвал кэшей на CI



# iOS-билды?





# IDE Sync?

Pure gradle (последняя задача prepareKotlinForIdealImport)	Pure IDE part	x	Talaitot time	IDE CLI time	IDE Full UI time
3m 40s	6m 12s	x	6m 10s	8m 17s	9m 52s
2m 48s	6m 6s	x	6m 6s	7m 20s	8m 54s
3m 5s	5m 35s	x	5m 35s	7m 14s	8m 40s
3m 5s	6m 5s	x	5m 49s	7m 29s	9m 10s
3m 15s	6m 46s	x	6m 30s	8m 23s	10m 1s

# Советы по деплою

Метрики

ClickHouse

1

# Советы по деплою

ClickHouse

1

Local  
Podman

2

# Советы по деплою

Метрики

ClickHouse

1

Local  
Podman

2

AI-assisted  
engineering

3

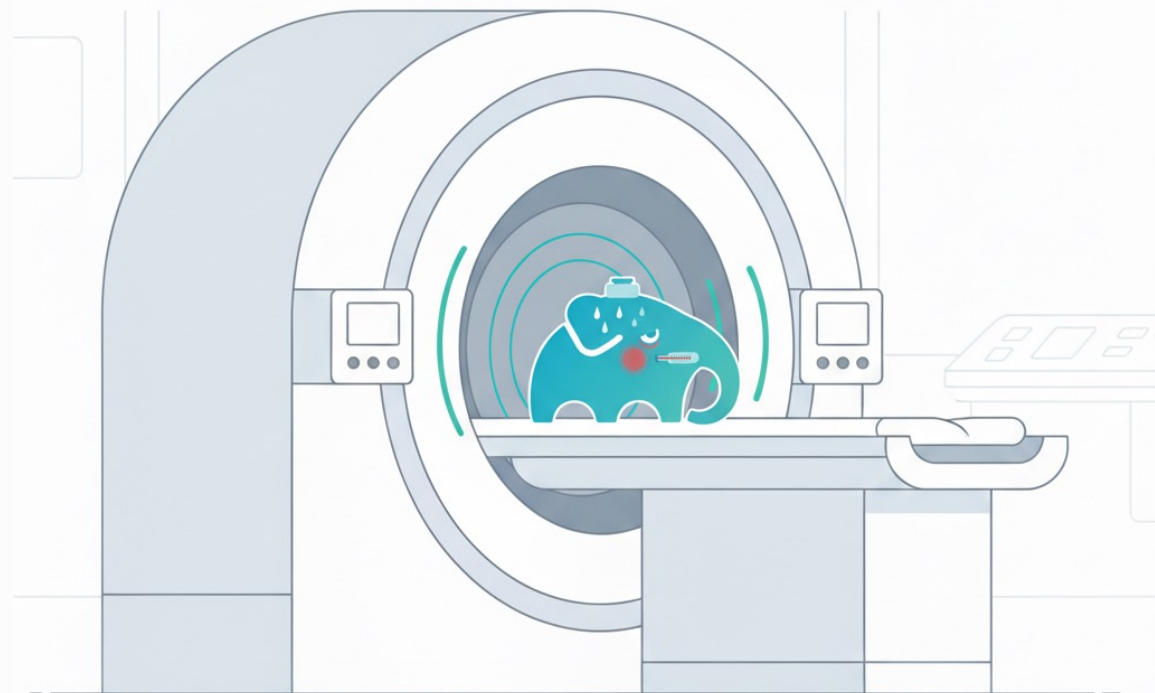
# Итого

- ✓ Авто-сбор метрик с машин и CI
- ✓ Гибкая визуализация
- ✓ Легко масштабируется



3

Ставим  
ТОЧНЫЙ  
ДИАГНОЗ



# Что хотим?

Граф gradle  
tasks билда

В порядке  
вызова, с учётом  
параллелизации

CPU / RAM / GC?

# Крутые профайлеры

Профайлер

Gradle-profiler  
Chrome Trace

~~Async Profile~~

Java Flight  
Recorder

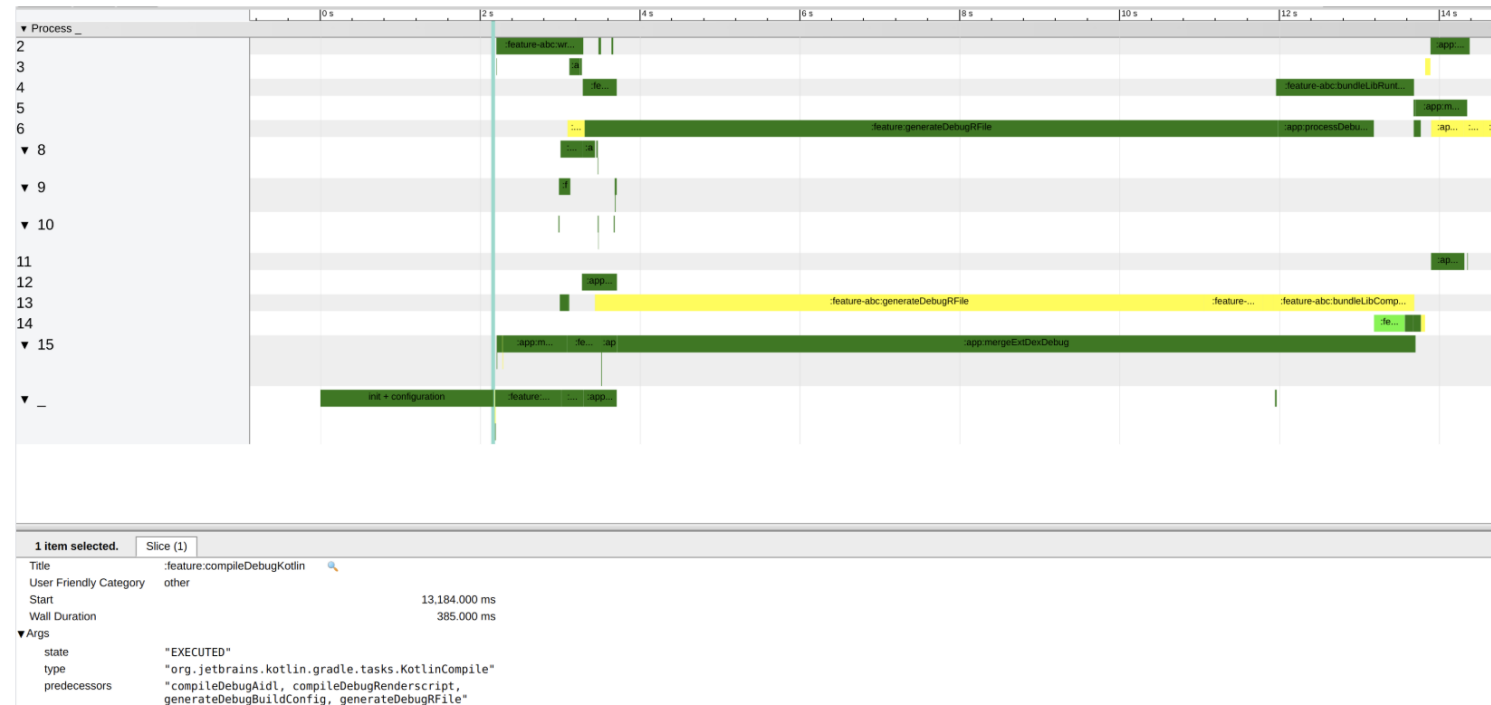
~~JProfiler~~

~~YourKit~~

## Build trace Gradle plugin

This plugin is a primitive analog of [Gradle build scan](#). Use it if you can't use a build scan for any reason.

This plugin collects tasks execution time in a trace event format.

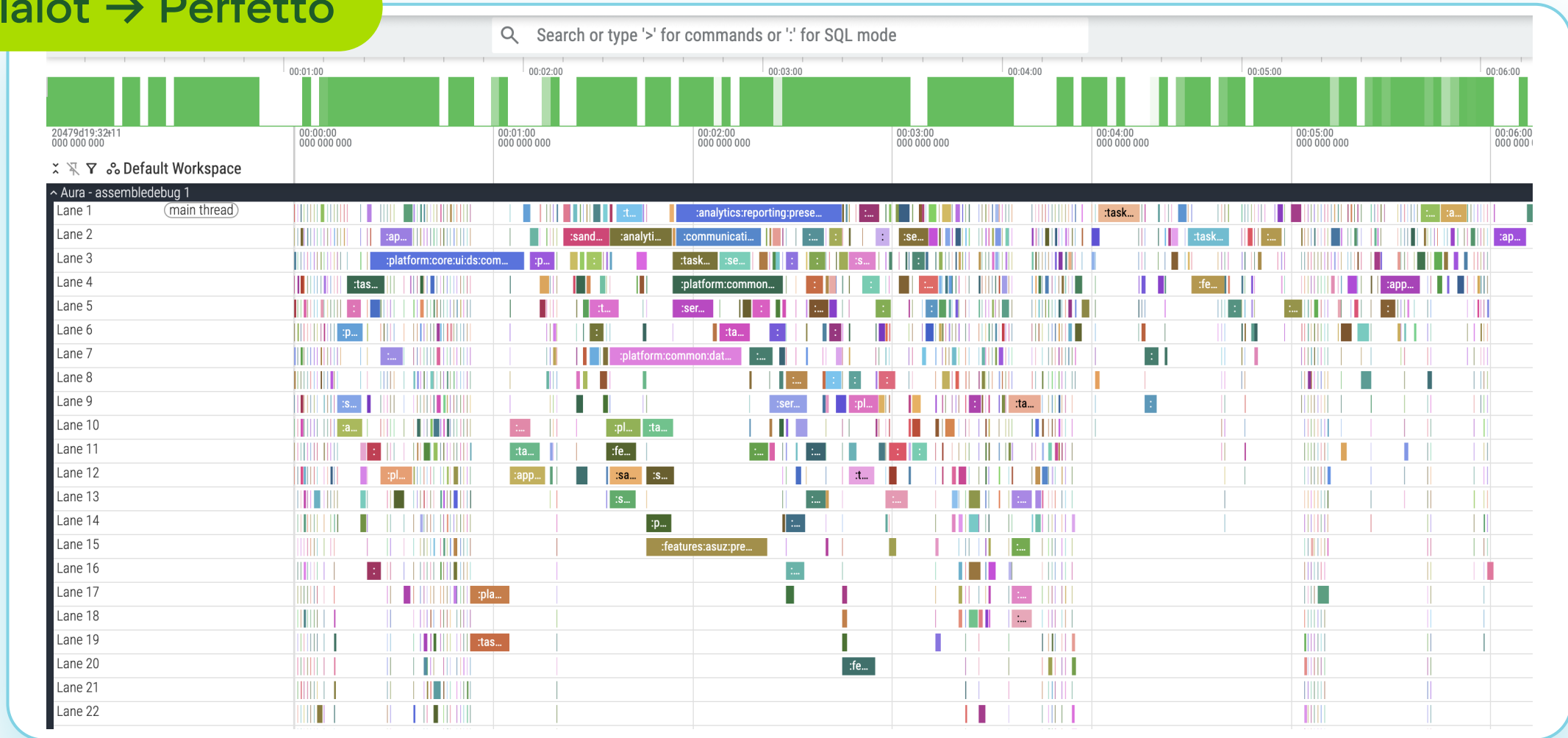


(c) Avito Build Trace

# 20-80 профайлинга

Профайлер

Talait → Perfetto



# Talaitot → Perfetto

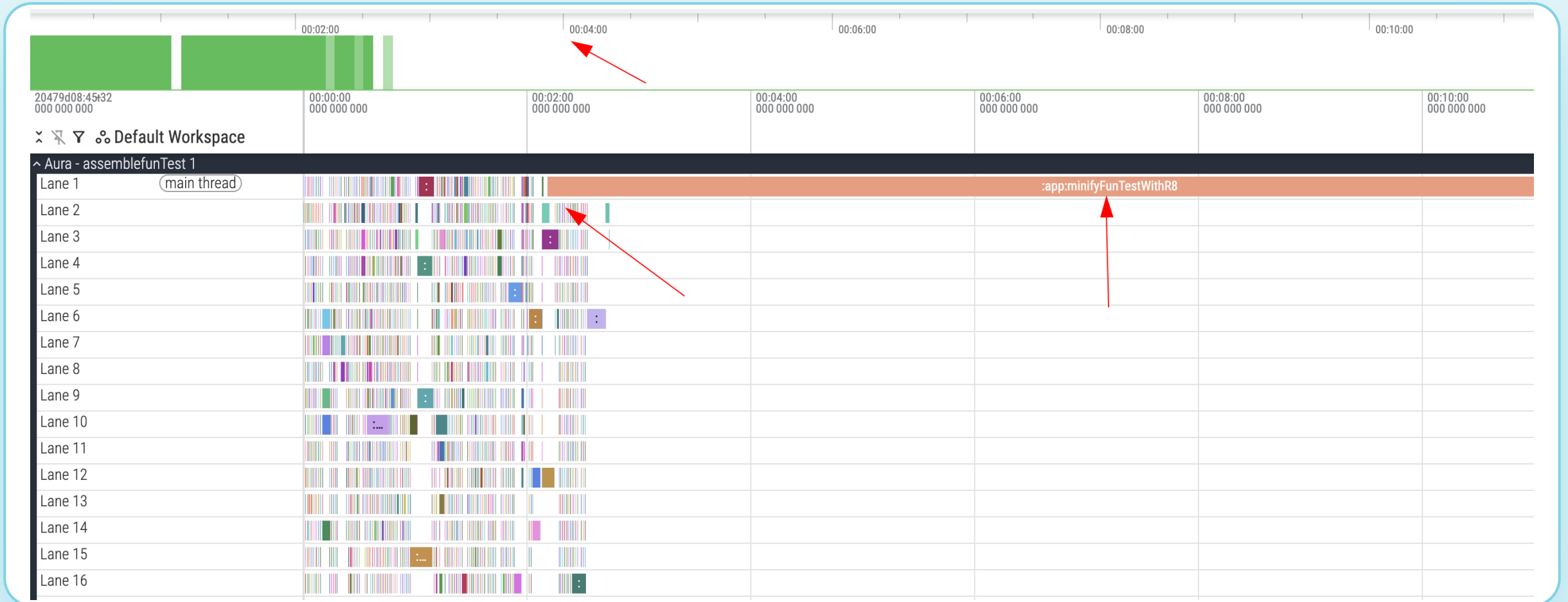
Профайлер

Use Perfetto to view as a timeline:

```
./platform/plugins/build-metrics/tools/profiling/0_talaitot_to_perfetto.py 93944580-ad52-49ff-b647-82a2945f10e4 --min-duration 10
```



# Дело №4. Долгий CI



# Дело №4. Долгий CI

Профайлер

ДО: assembleFunTest



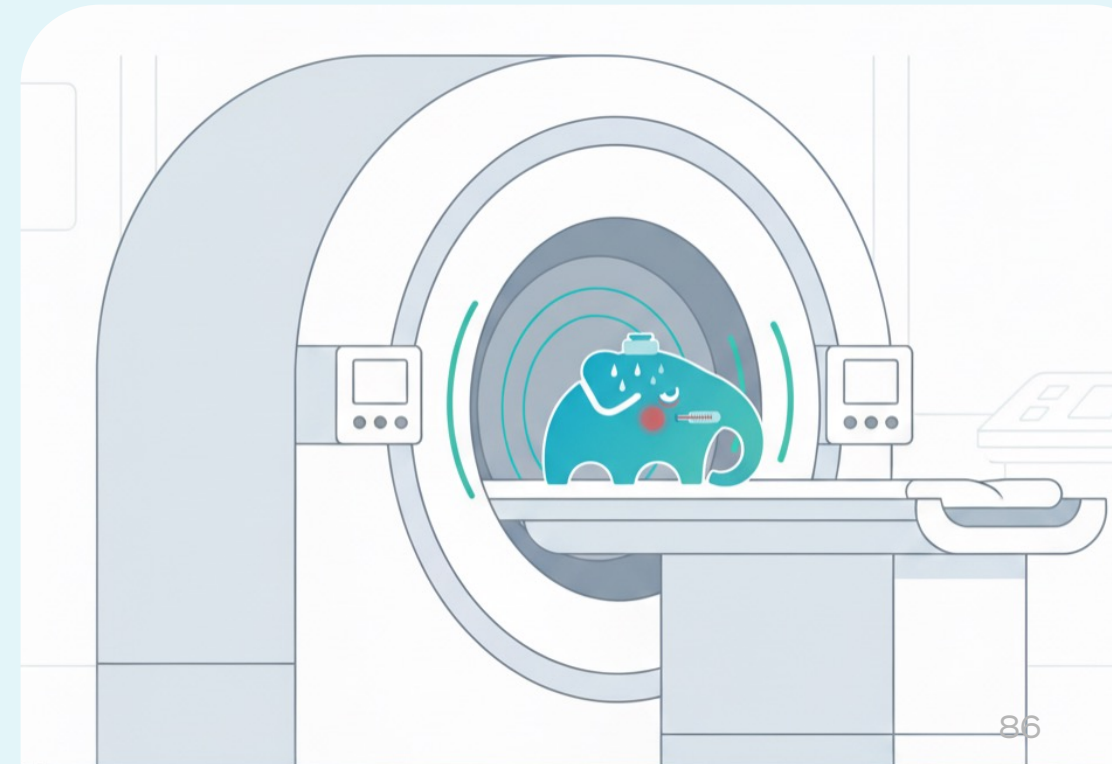
ПОСЛЕ: assembleDebug



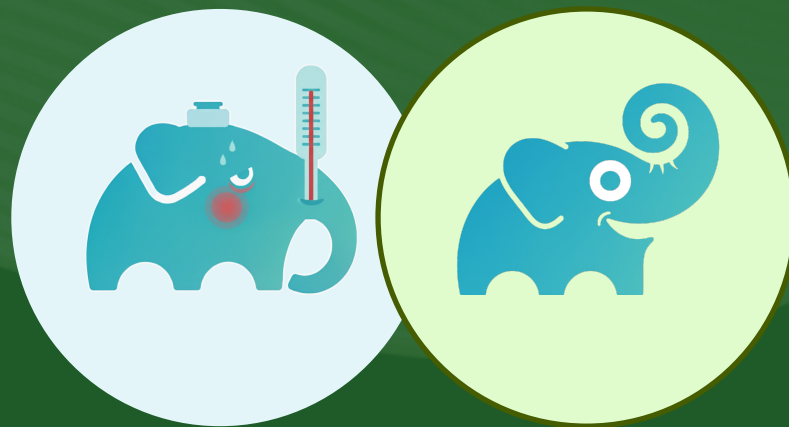
# Итого

✓ Детальные таски  
любого билда

✓ Легко найти  
боттлнеки



# Результат



# Чего добились

Оценка  
тренда  
в 1 клик

1

# Чего добились

Оценка  
тренда  
в 1 клик

1

Sanity check:  
сбор живых  
метрик

2

# Чего добились

Оценка  
тренда  
в 1 клик

1

Sanity check:  
сбор живых  
метрик

2

Детальная  
диагностика

3

# Вид решения

gradle-profiler

1

# Вид решения

**gradle-profiler**

1

**Talait +  
ClickHouse +  
Grafana**

2

# Вид решения

**gradle-profiler**

1

**Talait +  
ClickHouse +  
Grafana**

2

**Perfetto**

3

# Чего добились



# Чего добились



Обосновали вынос вложенного проекта, ускорились в 2 раза



# Чего добились

✓ Обосновали вынос вложенного проекта, ускорились в 2 раза

✓ Отловили замедление от антивируса в 2-3 раза



# Чего добились

✓ Обосновали вынос вложенного проекта, ускорились в 2 раза

✓ Отловили замедление от антивира в 2-3 раза

✓ Отловили отвал кэшей



# Чего добились

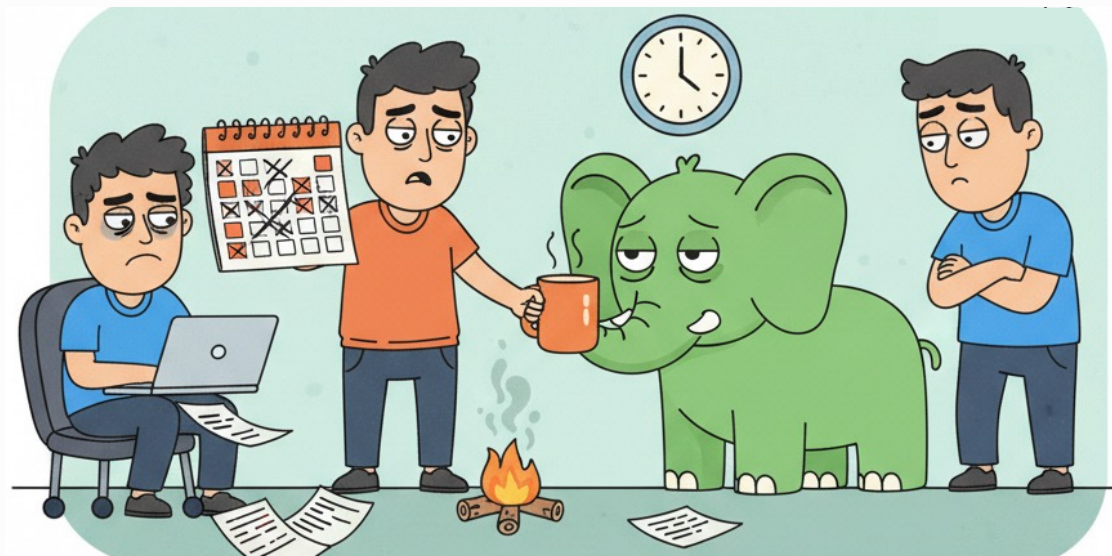
✓ Обосновали вынос вложенного проекта, ускорились в 2 раза

✓ Отловили замедление от антивируса в 2-3 раза

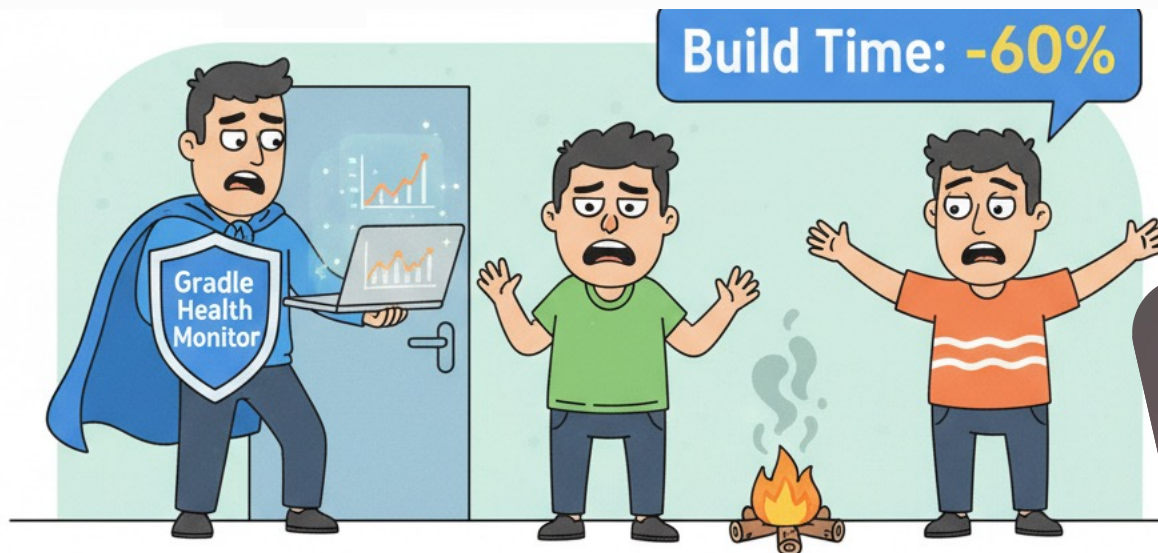
✓ Отловили отвал кэшей

✓ Форсировали переход на AGP 9, ускорились на 30%

До:



После:



Что предстоит  
решить?



# Android Studio

Otter 3 Feature Drop

2025.2.3



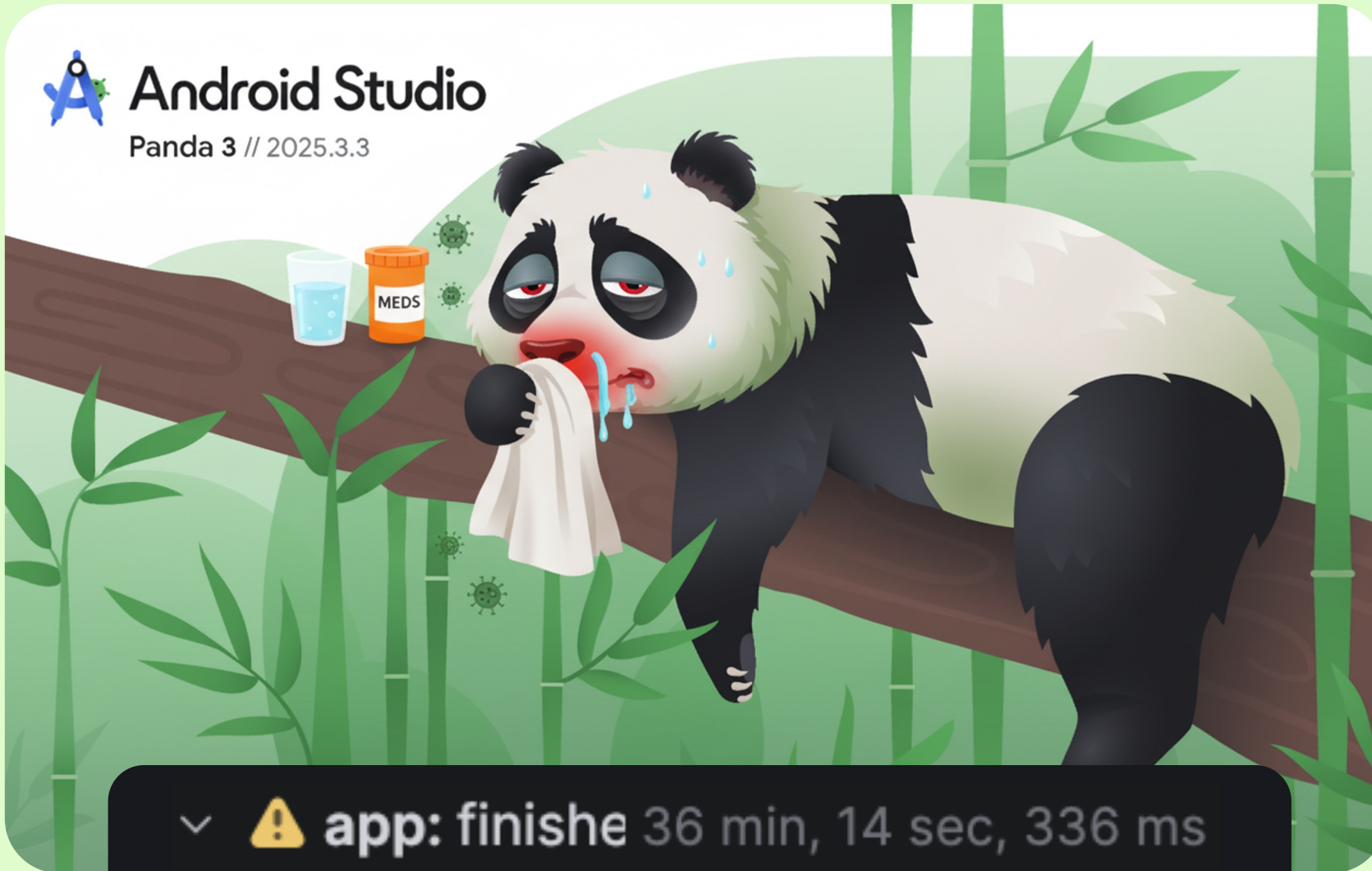
Powered by the IntelliJ® Platform

8 min, 24 sec, 940 ms



# Android Studio

Panda 3 // 2025.3.3



⚠ app: finishe 36 min, 14 sec, 336 ms

⬇ Download info

```
user@KL02594CJH-MBP app % ./local_scripts/kmp_targets/kmp-target-manager.sh android-fast
```

```
✓ Изменено файлов: 8, создано: 0
```

```
Конфигурация local.properties в репозиториях:
```

```
=====
app                ✓ android ✗ ios ✓ jvm ✗ web
platform          ✓ android ✗ ios ✓ jvm ✗ web
analytics         ✓ android ✗ ios ✓ jvm ✗ web
task manager      ✓ android ✗ ios ✓ jvm ✗ web
services and powers ✓ android ✗ ios ✓ jvm ✗ web
communication information ✓ android ✗ ios ✓ jvm ✗ web
mgk               ✓ android ✗ ios ✓ jvm ✗ web
sandbox           ✓ android ✗ ios ✓ jvm ✗ web
=====
```

```
Легенда: ✓ = включено, ✗ = отключено
```

🔥 Включен android-таргет + JVM (для корректных импортов)

⚠ Не забудьте перед коммитом вернуть в 'default' или хоть 'android-ios', проверить ios

# Фиче-реквесты

Автопрогоны  
бенчмарка  
на CI

А

RAM, CPU, GC

Б

Оценка  
здоровья  
билда от AI

В

Как быстро  
начать мониторить?

# Value-план на 3 дня

01

Настроить бенчмарк  
для android debug билда

02

Сварганить скрипт оценки тренда



# Value-план на 2 недели

1-3

дни

Настроить  
бенчмарк

4-7

дни

Подключить  
Talait

8-14

дни

Развернуть  
ClickHouse,  
настроить  
Grafana



# Главные ИТОГИ

Настроили мониторинг  
здоровья КМР сборки



# Главные ИТОГИ

Настроили мониторинг  
здоровья КМР сборки

IDE sync 13 мин → 8 мин → ???



# Главные ИТОГИ

Настроили мониторинг  
здоровья КМР сборки

IDE sync 13 мин → 8 мин → ???

Android debug 3,5 мин → 1,5 мин



# Главные ИТОГИ

Настроили мониторинг  
здоровья КМР сборки

IDE sync **13 мин** → **8 мин** → ???

Android debug **3,5 мин** → **1,5 мин**

Подключили 13 проектов



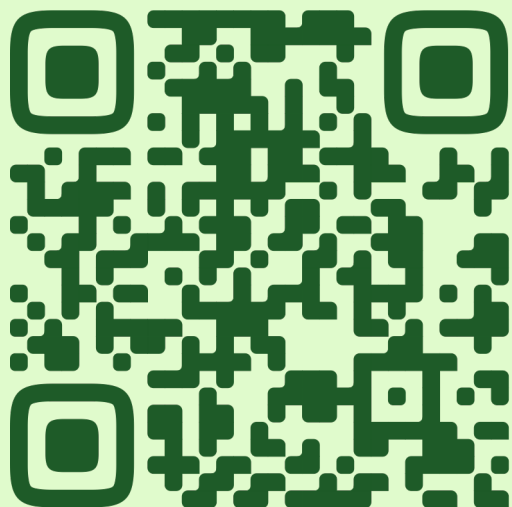
# Экономим бизнесу МИЛЛИОНЫ

Скостили **7 мин** синка,  
**2 мин** android debug

20 сборок, 5 синков в день =  
**1ч 15 м** разраб/день

247 дней x 20 разрабов = **6850 ч/г**

6850 x 2000 руб/ч = **13 700 000 ₽/г**



tg: @keystarrjob

Кирилл Старостин  
Senior KMP SE, X5 Tech

Настроили мониторинг  
здоровья KMP сборки

IDE sync 13 мин → 8 мин → ???

Android debug 3,5 мин → 1,5 мин

Подключили 13 проектов