

# CI/CD не работает, *и я знаю почему*

Карта точек отказа мобильного конвейера

Вадим Некипелов, Head of Mobile в Туту

2026

**tutu**

# Дисклеймер

В докладе не будет *рубилова по техничке*, но, надеюсь, что будет интересно. Моя задача сегодня, чтобы вы *расслабились и получили удовольствие* от презентации

# Ошибки в workflow

**Ошибки в workflow**

**Очереди на CI/CD**

**Ошибки в workflow**

**Очереди на CI/CD**

**Статистика сбоев**

**Ошибки в workflow**

**Очереди на CI/CD**

**Статистика сбоев**

**Цена простоя**

# Вызываем технический десант

Core-команда  
и DevOps



# Немного о Туту

**50** мобильных разработчиков,

**10** мобильных нативных приложений

**2-х** недельные релизы во все популярные сторы



# Наш конфиг в Туту

В качестве железа  
используем:

10 Mac Studio Self Host

GitHub + GitHub Actions



“The power of *the sun*,  
in the palm of my  
hand”



“Is it *too much* to ask  
for both?”



“I can do this *all day*”



“There was *an idea*...”



**“Whatever it takes!”**



**“Avengers, assemble”**



“Simple spells but  
quite *unbreakable*”





# Аналогия проблемы

**Раньше:** «*Один чан*  
на все случаи жизни»»



**Инструмент:**

Огромная кастрюля на 4 конфорки

**Проблема:**

Чтобы сварить яйцо, греем 50 литров воды. Если пригорело — выливаем всё и начинаем с нуля

**Результат:**

Повара злятся, кухня в дыму, заказы опаздывают, клиенты голодные

# Ключевые проблемы

Ближе к релизу накапливались очереди, разработчики ждали прогон PR по часу

Ошибки на CI/CD были непрозрачные, и над их решениями системно никто не работал

Пересборки на регрессе долгие, выбивались из графика релиза

**Нужны новые мощности?**

# Проблема количества и качества

Раннеров много — дорого

Раннеров мало — очереди

# Цена «красного» билда

**CI/CD — критическая часть продукта**

**Прямая корреляция:**  
стабильность пайплайна <-> Delivery Time

**Цена простоя**

# Цена проста

## Дано:

**10 часов** в месяц  
простоя одного повара

**1500 рублей**  
час работы повара

**50 поваров**



# Цена проста

## Дано:

**10 часов** в месяц  
простоя одного повара

**1500 рублей**  
час работы повара

**50 поваров**

## Расчет:

$$1 \quad 1500 * 10 = 15.000$$



# Цена простоя

## Дано:

**10 часов** в месяц  
простой одного повара

**1500** рублей  
час работы повара

**50** поваров

## Расчет:

**1**  $1500 * 10 = 15.000$

**2**  $15000 * 50 = 750.000$



# Цена простоя

## Дано:

**10 часов** в месяц  
простой одного повара

**1500** рублей  
час работы повара

**50** поваров

## Расчет:

**1**  $1500 * 10 = 15.000$

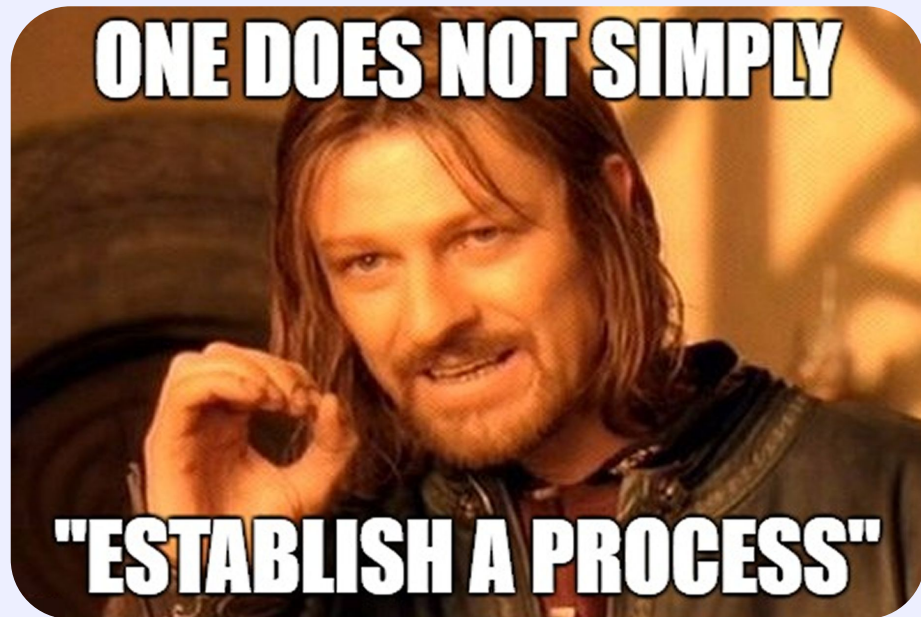
**2**  $15000 * 50 = 750.000$

**3**  $750000 * 12 = 9.000.000$





# Через что решили технически делать проект

- Взяли за основу API GitHub
- В качестве клиента взяли  
**Grafana** **Kibana**
- И немного фантазии




**A few moments later...**

# Метрики. Шаг 1: Занятость раннеров

Статус билдеров 				
Номер билдера 	Статус билдера	Пайплайн	Репозиторий	Доступность би
mac-builder-01	Занят	MCORE-469: Сбор и отправка метрики количества соединений	android-core	online
mac-builder-02	Занят	MCORE-469: Сбор и отправка метрики количества соединений	android-core	online
mac-builder-03	Занят	ETRAIN-7518: добавлены deviations data модели и репозиторий	android-core	online
mac-builder-04	Занят	ETRAIN-7518: добавлены deviations data модели и репозиторий	android-core	online
mac-builder-05	Занят	MCORE-2114: Управление ФФ на примере ЛК	ios-core	online
mac-builder-06	Занят	MCORE-2146: Релиз 3.165 в мастер	ios-core	online
mac-builder-07	Занят	HOTELS-13102: Снекбар на карте отелей	ios-core	online
mac-builder-08	Занят	MCORE-469: Сбор и отправка метрики количества соединений	android-core	online
mac-builder-09	Занят	Auto bump iOS	devops-monorepo	online
mac-builder-10	Занят	MCORE-469: Сбор и отправка метрики количества соединений	android-core	online

# Метрики: Список очередей

Очереди 			
Статус	Пайплайн	Джоба	Репозиторий
В очереди	<a href="#">MCORE-1687: Бекпорт java.time.Duration методов</a>	Unit tests	android-core
В очереди	<a href="#">FINTECH-1601: Реализован Compose UI для пополнения ЛК через pull top-up</a>	Run danger-kotlin	android-core
В очереди	<a href="#">MCORE-1687: Backport some java.time.Duration methods</a>	Run security scan	android-core
В очереди	<a href="#">CA-466: [Price Alert Avia] Добавлена шторка для включения пушей в настройках</a>	Run security scan	ios-core

# Промежуточный результат

1

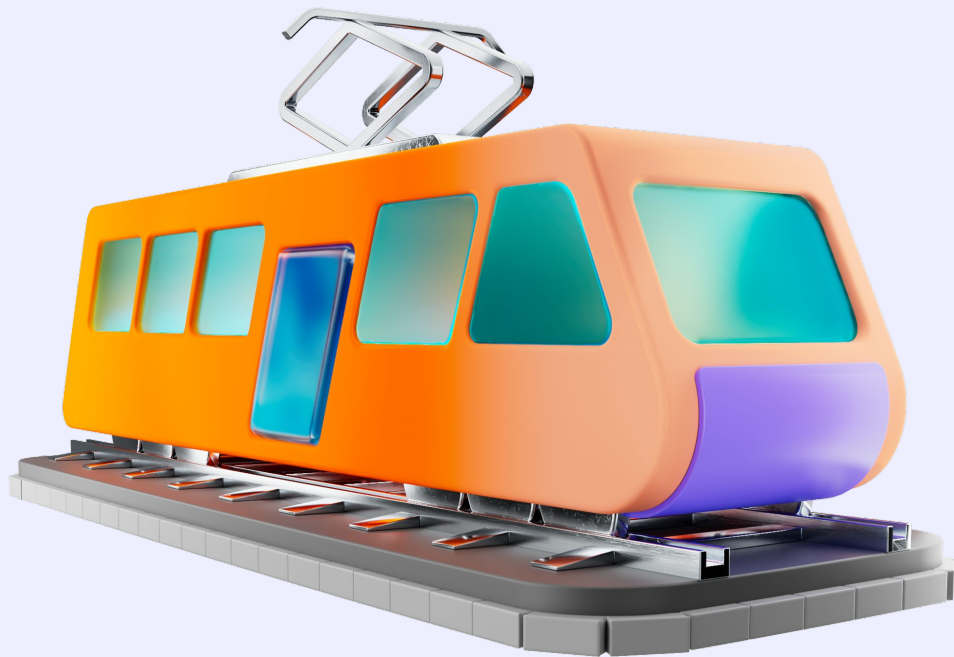
Разработчики стали сознательнее подходить к деплою и вставать в очередь заранее

2

Core-команда перестала быть точкой правды и единственными информаторами состояния CI/CD

3

Релизный поезд стал стартовать по расписанию в х3 стабильнее (без опозданий)



# А что дальше?

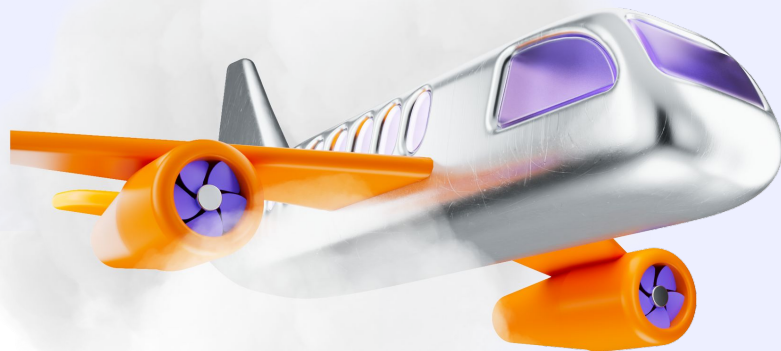
## Вопросы:

1

А как часто  
возникают  
очереди?

2

А точно ли  
проблема в очередях  
и количестве машин?



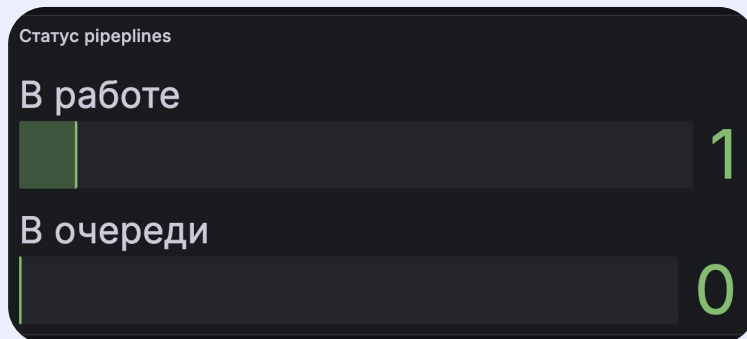
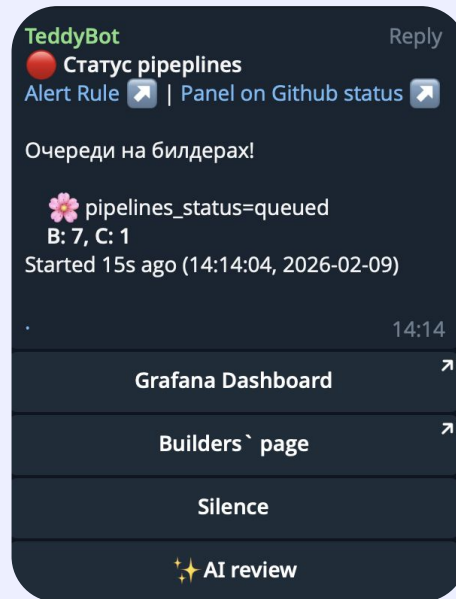
# Автоматизация. Шаг 1

## Настраиваем *трешхолд*:

**1** Если в течение 5 минут в очереди более 3 запросов

**2** Отправляем алерт-нотификацию

**3** Составляем статистику за отчетный период времени



# Промежуточный результат

1

Команда девопсов лучше понимает и осознает проблему нехватки ресурсов

2

Пики нагрузки перед каждым релизом

3

Пики нагрузки при запуске тестов



# Что дальше?

## Вопросы:

1

**Купить еще  
пул машин?**

2

**Оптимизировать  
нагрузку на машины?  
Что на них запускается?**



# Три всадника апокалипсиса



## DEV

Ошибки кода,  
тестов, линтеров



## CORE

Инфраструктурный  
ад (YAML, конфиги)



## EXTERNAL

Внешние факторы  
(Сеть, API, санкции)



# Зачем делить ответственность

**Разные уровни = разные способы лечения**

**Прозрачность для менеджмента ресурсов**

**Четкие зоны ответственности команд**

# Let's

# do it!

А давайте тестово  
соберем *все логи*  
*отказов за день*  
в наших репозиториях?



# АД в логах

```
1905 67 10.5M 67 7312k 0 0 25285 0 0:07:16 0:04:56 0:02:20 36061
1906 68 10.5M 68 7343k 0 0 25254 0 0:07:16 0:04:57 0:02:19 31882
1907 68 10.5M 68 7343k 0 0 25219 0 0:07:17 0:04:58 0:02:19 29133
1908 68 10.5M 68 7391k 0 0 25304 0 0:07:16 0:04:59 0:02:17 31500
1909 68 10.5M 68 7423k 0 0 25335 0 0:07:15 0:05:00 0:02:15 33041
1910 curl: (28) Operation timed out after 300054 milliseconds with 7602149 out of 11032811 bytes received
1911 Error: Process completed with exit code 28.
```

# АД в логах

```
1905 67 10.5M 67 7312k 0 0 25285 0 0:07:16 0:04:56 0:02:20 36061
1906 68 10.5M 68 7343k 0 0 25254 0 0:07:16 0:04:57 0:02:19 31882
1907 69 10.5M 69 7312k 0 0 25210 0 0:07:17 0:04:58 0:02:18 33100
84
85 FAILURE: Build failed with an exception.
86
87 * What went wrong:
88 Execution failed for task ':tutu_landing:testDebugUnitTest'.
89 > org.gradle.api.internal.tasks.testing.junit.result.TestFailure
90
```

# АД в логах

```
1905 67 10.5M 67 7312k 0 0 25285 0 0:07:16 0:04:56 0:02:20 36061
1906 68 10.5M 68 7343k 0 0 25254 0 0:07:16 0:04:57 0:02:19 31882
1907 69 10.5M 69 7312k 0 0 25210 0 0:07:17 0:04:58 0:02:18 33100
```

```
84
```

```
85 FAILURE: Build failed with an exception.
```

```
1. RouteDetailsViewTests.testRouteDetailsViewWithMockData()
```

```
Target : AviaFlightDetailsSnapshotTests
```

```
Сообщение: failed - Snapshot does not match reference.
```

```
Файл : Users/dev/actions-runner/_work/ios-core/ios-
```

```
core/Avia/AviaFlightDetailsSolution/AviaFlightDetailsSnapshotTests/Route/RouteDetailsViewSna
```

# АД в логах

```
1905 67 10.5M 67 7312k 0 0 25285 0 0:07:16 0:04:56 0:02:20 36061
1906 68 10.5M 68 7343k 0 0 25254 0 0:07:16 0:04:57 0:02:19 31882
1907 69 10.5M 69 7312k 0 0 25210 0 0:07:17 0:04:58 0:02:18 33100
```

84

85 FAILURE: Build failed with an exception.

1. RouteDetailsVewTests.testRouteDetailsViewWithMockData()

2. SegmentViewSnapshotTests.testSegmentViewWithMockData()

Target : AviaFlightDetailsSnapshotTests

Сообщение: failed – Snapshot does not match reference.

CO

Файл : Users/dev/actions-runner/\_work/ios-core/ios-

core/Avia/AviaFlightDetailsSolution/AviaFlightDetailsSnapshotTests/Segment/SegmentViewSn

# АД в логах

```
1905 67 10.5M 67 7312k 0 0 25285 0 0:07:16 0:04:56 0:02:20 36061
1906 68 10.5M 68 7343k 0 0 25254 0 0:07:16 0:04:57 0:02:19 31882
1907 69 10.5M 69 7312k 0 0 25285 0 0:07:17 0:04:56 0:02:19 33182
```

```
84
```

```
85 FAILURE: Build failed with an exception.
```

```
1. RouteDetailsViewTests.testRouteDetailsViewWithMockData()
```

```
2. SegmentViewSnapshotTests.testSegmentViewWithMockData()
```

```
Target : AviaFlightDetailsSnapshotTests
```

```
272 FAILURE: Build failed with an exception.
```

```
273
```

```
274 * What went wrong:
```

```
275 Execution failed for task ':etrains:etrains_app:processPlayMarketQaGoogleServices'.
```

```
276 > File google-services.json is missing.
```

```
277 The Google Services Plugin cannot function without it.
```

```
278 Searched locations: /Users/dev/actions-runner/_work/android-core/android-core/etrains/etrains_app/src/pla
```

# АД в логах

```
1905 67 10.5M 67 7312k 0 0 25285 0 0:07:16 0:04:56 0:02:20 36061
1906 68 10.5M 68 7343k 0 0 25254 0 0:07:16 0:04:57 0:02:19 31882
1907 69 10.5M 69 7312k 0 0 25210 0 0:07:17 0:04:58 0:02:18 33182
```

84

85 FAILURE: Build failed with an exception.

1. RouteDetailsViewTests.testRouteDetailsViewWithMockData()

2. SegmentViewSnapshotTests.testSegmentViewWithMockData()

Target : AviaFlightDetailsSnapshotTests

272 FAILURE: Build failed with an exception.

273

27 281 ! There are some linter warnings/errors. Please, fix it and try again !

27 282 /Users/dev/actions-runner/\_work/ios-core/ios-core/TransportApp/TransportApp/Coordinators/PTTCoordinator

27 283 ⚠ Line 141: Line should be 120 characters or less; currently it has 136 characters (line\_length)

27 284 /Users/dev/actions-runner/\_work/ios-core/ios-

27 core/TransportApp/TransportApp/SharedDependencies/DependencyFactories/ServiceDependenciesFactory.swift

285 ⚠ Line 32: Type body should span 250 lines or less excluding comments and whitespace: currently span

286 **Error:** Process completed with exit code 2.

# Промежуточный результат

мобильный девопс  
хочет написать  
заявление  
на увольнение



# AI-driven аналитика

**Регулярки больше не катят**

**LLM превращает тысячи строк логов  
в понятный топ-5**

**Автоматическая классификация ошибок**



# Автоматизация. Шаг 2

Успешность Workflow в мобильных репозиториях 

mob...	mo...	m.	mobile_con...	mo...	mobile_context.reason (allValues)
89%	131	17	ios-core	dev	1. Ошибка связана с отсутствием поддерживаемых платформ для сборки схемы в xcodebuild, из-за чего не за
91%	161	13	android-core	dev	1. Ошибка произошла при выполнении unit тестов модуля train_app — KSP завершился с ошибкой PROCESSING_I
98%	131	17	ios-core	external	1. Ошибка сетевого соединения: проблема с TLS handshake при попытке соединения с внешним сервером, поэ
98%	161	13	android-core	core	1. Ошибка возникла из-за отсутствия аннотированного AppComponent, необходимого для генерации кода KSP, г
99%	131	17	ios-core	core	Не удалось выполнить git push из-за защиты ветки master: требуется выполнение status check "run_pull_request

# Автоматизация. Шаг 2

## Успешность Workflow в мобильных репозиториях

mob...	mo...	m.	mobile_con...	mo...	mobile_context.reason (allValues)
89%	131	17	ios-core	dev	1. Ошибка связана с отсутствием поддерживаемых платформ для сборки схемы в xcodebuild, из-за чего не за
91%	161	13	android-core	dev	1. Ошибка произошла при выполнении unit тестов модуля train_app — KSP завершился с ошибкой PROCESSING_I
98%	131	17	ios-core	external	1. Ошибка сетевого соединения: проблема с TLS handshake при попытке соединения с внешним сервером, поэт
98%	161	13	android-core	core	1. Ошибка возникла из-за отсутствия аннотированного AppComponent, необходимого для генерации кода KSP, г
99%	131	17	ios-core	core	Не удалось выполнить git push из-за защиты ветки master: требуется выполнение status check "run_pull_request

1. Ошибка связана с отсутствием поддерживаемых платформ для сборки схемы в xcodebuild, из-за чего не запускаются unit/UI тесты TransportApp (они не могут старт овать из-за проблем с конфигурацией схемы или симулятора).
2. Истек таймаут выполнения действия 'Sync Dangerfile', связанного с Danger и проверками pull request.
3. Произошёл таймаут выполнения действия Danger, связанный с ошибкой выполнения тестов или инструментов проверки кода.
4. Упали unit/UI snapshot тесты AviaApp (`testRouteDetailsViewWithMockData`, `testSegmentViewWithMockData`) из-за несовпадения снимка с эталонным изображением (Snapshot does not match reference).
5. Упали unit тесты Etrain из-за таймаута ожидания и несоответствия ожидаемого и фактического результата (асинхронные wait timeout и XCTAssertEqual).

# Инсайт

**1** Пробуйте картографировать  
результаты

**2** Не изобретайте велосипед

**3** Ошибаться  
нормально



# Что дальше?

1

**Понимаем,  
что есть фронт  
работ**

2

**Хотим выстроить  
метрику успешности  
нашего конвейера**



# Метрики успеха – SLA

**Порог  
стабильности**



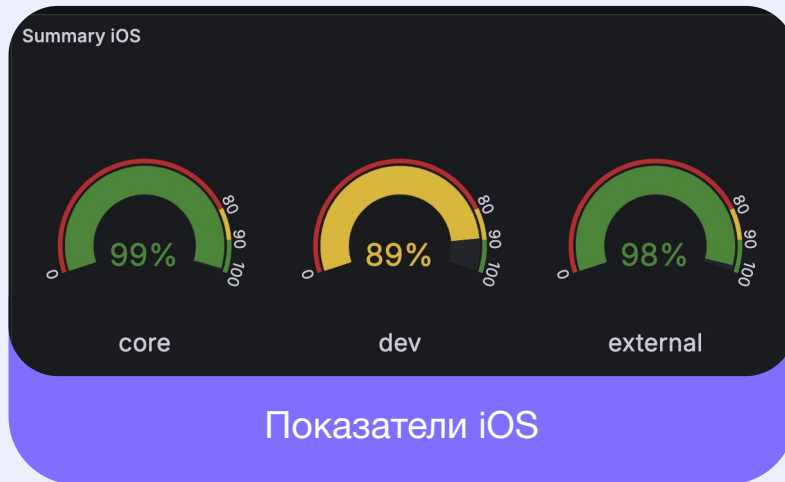
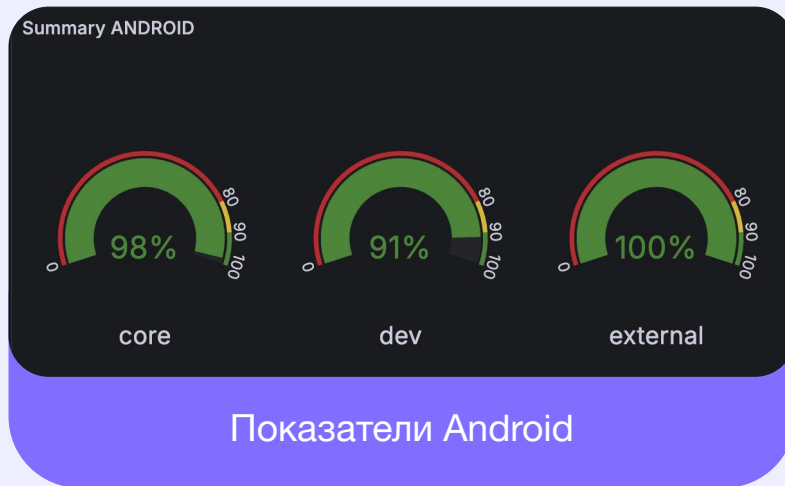
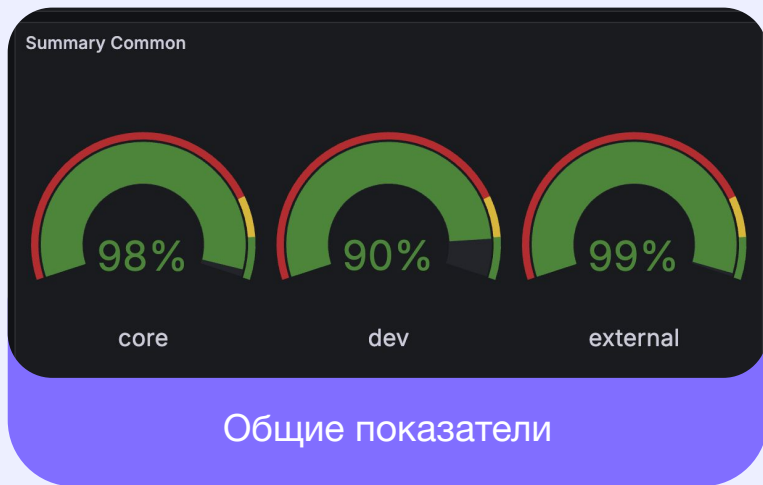
**Почему нам нужно 95%?**

**Система алертинга**



**Как не проспать деградацию?**

# Автоматизация. Шаг 3



# Автоматизация. Шаг 4

Время жизни пулл-реквестов в мобильных репозиториях [↗](#)

mobile_context...	mobile_con...	mobile_context.q...	Время реран...	время жизни...	время чеков(...	кол-...	время жизни ПР (тотал в...
ios-foundation	02.02	Q1 2026	1.32	12.24	1.32	1	12.24
ios-core	02.02	Q1 2026	2.51	24.00	20.00	7	168.00
design-system-tokens	02.02	Q1 2026	0.00	219.12	0.00	1	219.12
android-design	02.02	Q1 2026	0.29	2.40	0.29	1	2.40
android-core	02.02	Q1 2026	1.37	24.54	6.28	4	99.37
ios-features	03.02	Q1 2026	0.24	1.12	0.47	2	2.24

# Промежуточные итоги

Инженеры знают  
о *загруженности машин*

Инженеры знают о популярных  
*проблемах на конвейере*

Инженеры и бизнес смотрят  
на *SLA успешности*

Инженеры составляют  
*roadmap работ*

# Конечный вид продукта

## Успешность Workflow в мобильных репозиториях [↗](#)

mob...	mobile_co...	m.	mobile_context...	mo...	mobile_context.reason (allValues)	mobile_cont
89%	131	17	ios-core	dev	1. Ошибка связана с отсутствием поддерживаемых платформ для сборки схемы в xcode	02.04
91%	161	13	android-core	dev	1. Ошибка произошла при выполнении unit тестов модуля train_app — KSP завершился с	02.04
98%	131	17	ios-core	external	1. Ошибка сетевого соединения: проблема с TLS handshake при попытке соединения с в	02.04
98%	161	13	android-core	core	1. Ошибка возникла из-за отсутствия аннотированного AppComponent, необходимого дл	02.04
99%	131	17	ios-core	core	Не удалось выполнить git push из-за защиты ветки master: требуется выполнение status	02.04

## Время жизни пулл-реквестов в мобильных репозиториях [↗](#)

mobile_context...	mo...	mobile_...	Время р...	врем...	время...	кол-...	время жизни ПР (тотал в часах)
ios-core	02.04	Q2 2026	1.44	26.15	5.11	3	78.46
devops-монорепо	02.04	Q2 2026	0.02	0.18	0.02	1	0.18

## Summary Common



core




dev




external


# Конечный вид продукта

Daily report			
Дата	Категория	%	Суммаризация
2026-04-02	dev	90%	1. Ошибки компиляции и несовместимости кода (Kotlin/Swift, устаревшие опции компилятора, отсутствующие или несовместимые зависимости,
2026-04-02	core	98%	1. Отсутствие аннотированного AppComponent вызвало сбой генерации кода KSP и провал сборки AndroidTest.
2026-04-02	external	98%	1. Проблема с TLS handshake при соединении с внешним сервером, из-за чего загрузка не началась. 2. Сетевой таймаут при загрузке файла через curl, что

# Конечный вид продукта

Weekly report			
Дата	Категория	%	Суммаризация
01.04–05.04	common	96%	<ol style="list-style-type: none"><li>1. Массовые ошибки запуска unit/UI тестов связаны с некор</li><li>2. Компиляция модулей на Kotlin и Swift часто завершалась</li><li>3. Действие Checkout регулярно завершалось по таймауту п</li></ol>
30.03–31.03	common	96%	<ol style="list-style-type: none"><li>1. Наиболее часто встречаются ошибки компиляции и несо</li><li>2. Массово проявляются падения unit, UI и интеграционных</li><li>3. Таймауты и сбои при выполнении CI/CD pipeline (Checkout)</li></ol>
16.03–22.03	common	85%	<ol style="list-style-type: none"><li>1. Массовые ошибки компиляции и кода (Kotlin, Swift) в ключ</li><li>2. Многочисленные падения unit/UI/smoke тестов (бронь, оп</li><li>3. Провал линтинга и автоматизированных проверок (Dange</li></ol>

# Конечный вид продукта

Monthly report			
Дата	Категория	%	Суммаризация
апрель	common	96%	<ol style="list-style-type: none"><li>1. Массовые ошибки запуска unit/UI тестов из-за некорректной</li><li>2. Регулярные таймауты и сбои при загрузке репозитория и в</li><li>3. Компиляция модулей на Kotlin и Swift завершается с ошибка</li></ol>
март	common	91%	<ol style="list-style-type: none"><li>1. Массовые ошибки компиляции, несовместимость и конфлик</li><li>2. Многочисленные падения unit, UI, end-to-end, snapshot и инте</li><li>3. Регулярные сбои статических и автоматических проверок (D</li></ol>
февраль	common	94%	<ol style="list-style-type: none"><li>1. Массовые ошибки компиляции Kotlin/Swift (неразрешённые</li><li>2. Массовые падения unit и UI тестов (включая MessengerCoreT</li><li>3. Частые провалы автоматизированных код- и pull-request пр</li></ol>

# Конечный вид продукта

Quarterly report			
Дата	Категория	%	Суммаризация
Q2 2026	common	96%	<ol style="list-style-type: none"><li>1. Массовые ошибки запуска unit/UI тестов из-за некорректной конфигурации</li><li>2. Регулярные таймауты и сбои при загрузке репозитория и в процессе тестирования</li><li>3. Компиляция модулей на Kotlin и Swift завершается с ошибками</li></ol>
Q1 2026	common	89%	<ol style="list-style-type: none"><li>1. Массовые ошибки компиляции и конфликты зависимостей</li><li>2. Многочисленные падения unit, UI, e2e и интеграционных автотестов</li><li>3. Массовые сбои и нарушения статических и автоматических проверок</li></ol>
Q4 2025	common	76%	<ol style="list-style-type: none"><li>1. Массовые падения unit и UI тестов (включая snapshot-тесты)</li><li>2. Ошибки компиляции и генерации кода (Kotlin, Swift, kapt, Dagger)</li><li>3. Провалы автоматических проверок качества кода и линтинга</li></ol>

# Решение в аналогии

**Сейчас:**  
*«Профессиональный  
цех»*



**Специализация:**

Каждому блюду — своя посуда

**Контроль:**

Датчики температуры и таймеры

**Техническое обслуживание:**

Плита готова к работе до открытия

**Результат:**

Плита та же, но скорость выдачи блюд  
выросла в 5 раз

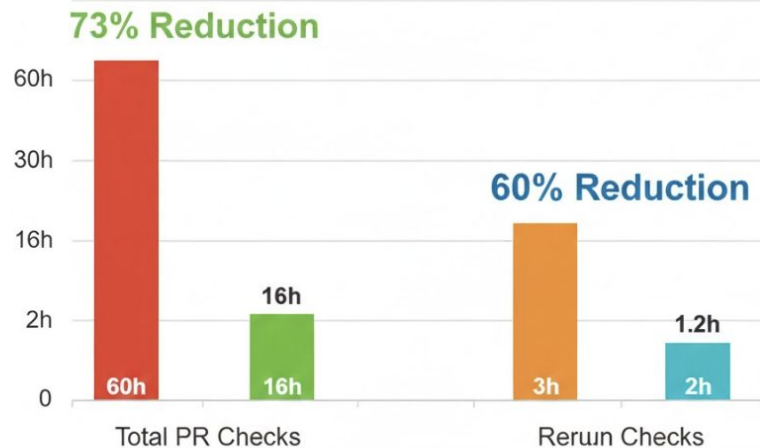
# Результаты и профит

1 Линтеры и гитхуки как фильтр Dev-ошибок

2 Стабилизация Core-слоя

3 Сокращение цикла поставки (Time-to-Market)

## Efficiency Boost: CI/CD Optimization



# Результаты и профит

**SLA конвейера повысили с 72% до 96%**

**Скорость Delivery time выросла на 30%**

**Нагрузка на девопс сократилась на 60%**



# Основные инсайты

1. Пробуйте, экспериментируйте, не ждите
2. Берите то, что уже есть под рукой, и прототипируйте
3. Даже самая простая визуализация помогает в стратегировании
4. SLA конвейера — это зрелость вашего продукта

# tutu

ВАДИМ НЕКИПЕЛОВ

nekipelov@tutu.tech

tg: @nek\_vi

# Спасибо

