

# Монорепозиторий

Денис Александров

Менеджер разработки, X5 Tech

# Обо мне

✦ 18 лет в ИТ

---

✦ Менеджер разработки

---

✦ Пишу много кода (ну как я))



tg: @guitariz

# О чём поговорим

- 01 Что такое  
монорепозиторий
- 02 Основные плюсы и минусы
- 03 Почему он нужен  
вам и сейчас
- 04 Как его реализовать



01

# Что такое монорепозиторий



# Определение

## Монорепозиторий –

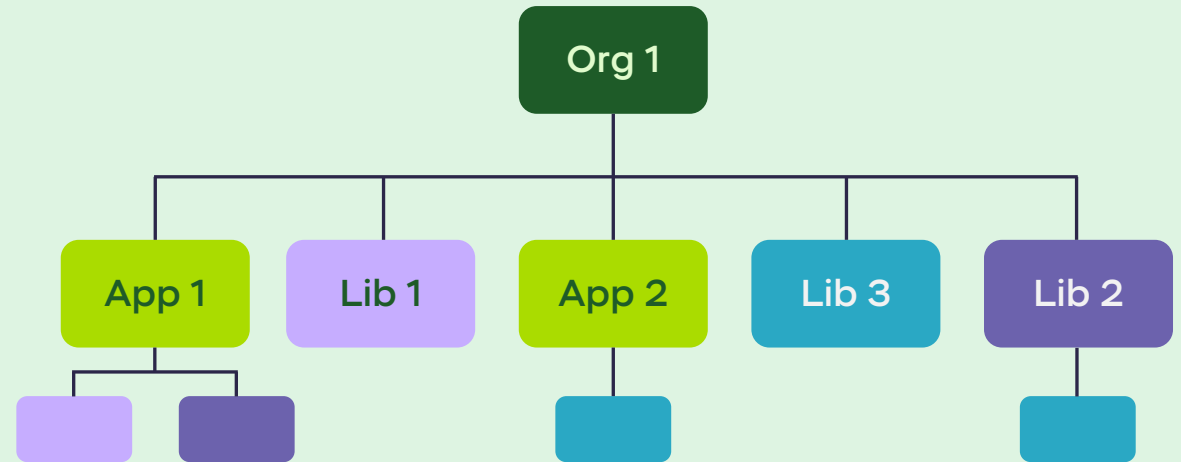
практика централизованного хранения всего кода проекта, появилась для повышения эффективности разработки

## Монопроект –

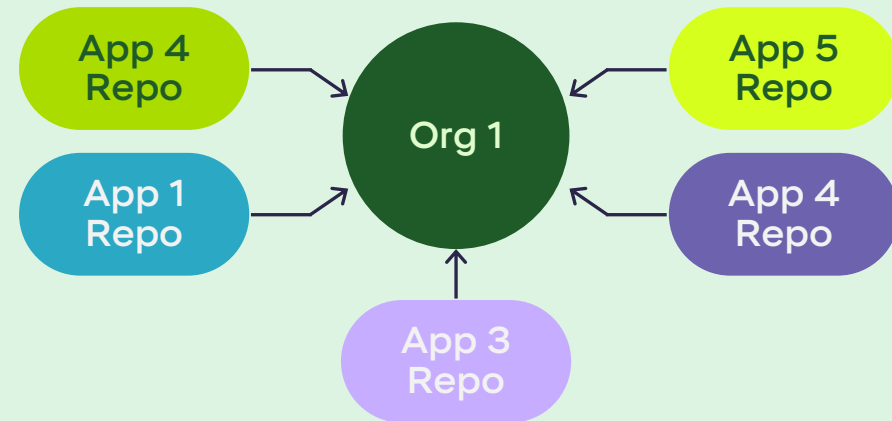
способ подключения кода как единого проекта в IDE (например, несколько мобильных приложений, бэкендов)

Монорепозиторий и монопроект – это **ключевые элементы** современного технологического Бигтеха

### Monorepo



### Polyrepo



# Так делают:

Google

Meta\*

Microsoft

X5 Tech

Ozon

VK

Яндекс

\*запрещённая на территории РФ организация

# TLDR – Монорепозиторий + Монопроект у нас



Все мобильные приложения  
Витрин в одном git

Релизы по торговым сетям  
раз в неделю

100+ активных  
разработчиков

Принцип «один код –  
все витрины»

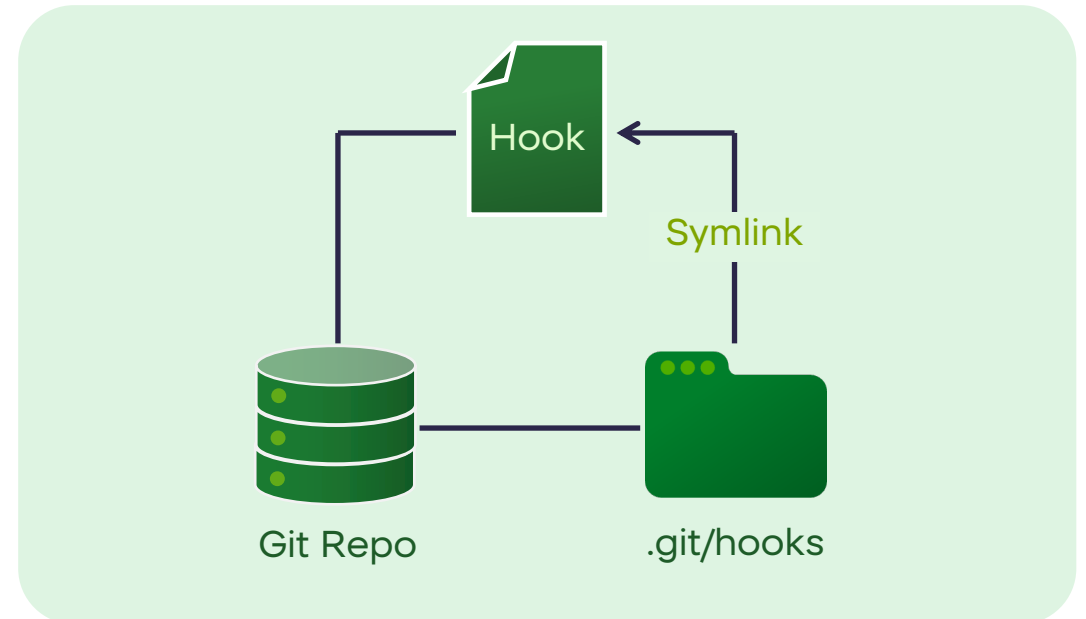
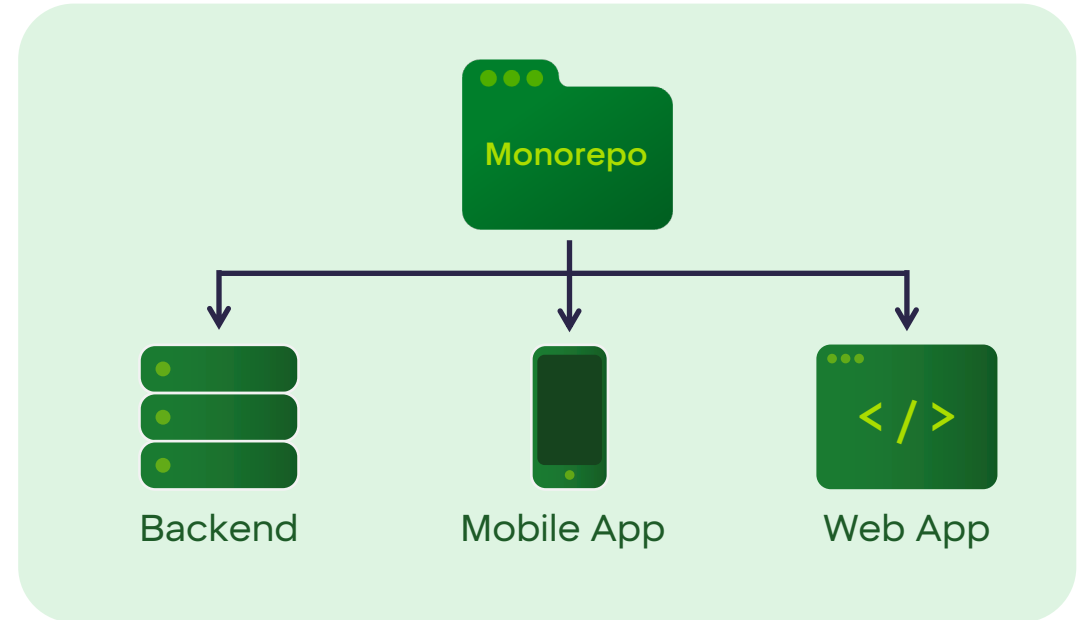
3000 Мр-ов(!!!) в месяц

Время пайплайна перед  
влитием – менее 60 минут

# Важно: монорепозиторий – не всегда монопроект

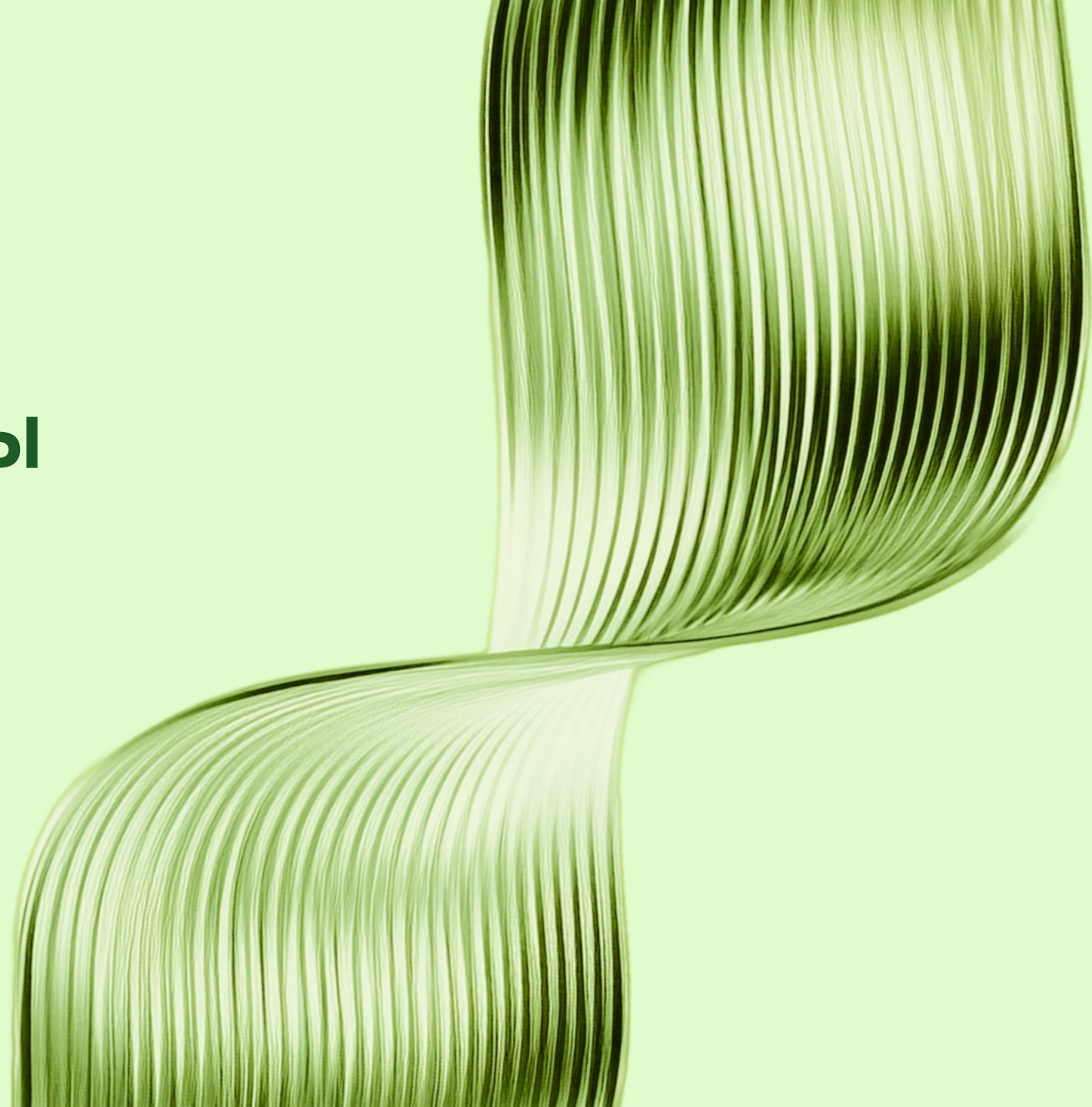
✦ Монорепозиторий может состоять **из нескольких проектов** (при этом они все лежат в одном гите)

✦ Монопроект может быть **из нескольких репозитиев** (через симлинки)



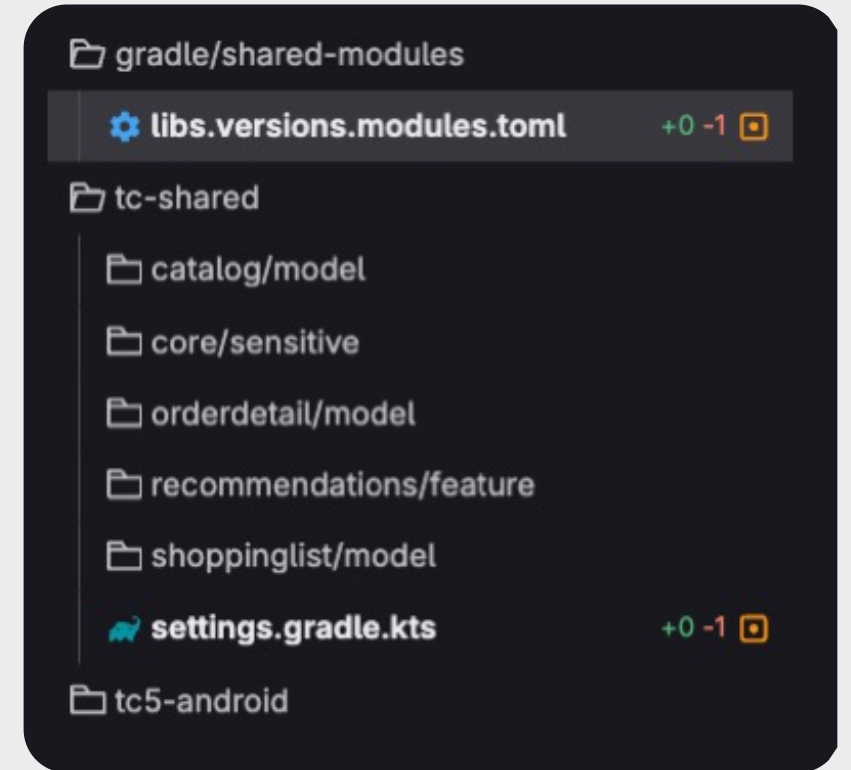
02

# Основные плюсы и минусы



# Преимущества (для всех)

1. Максимально быстрое переиспользование и правка кода



## MOBTECH-567 remove-sensitive

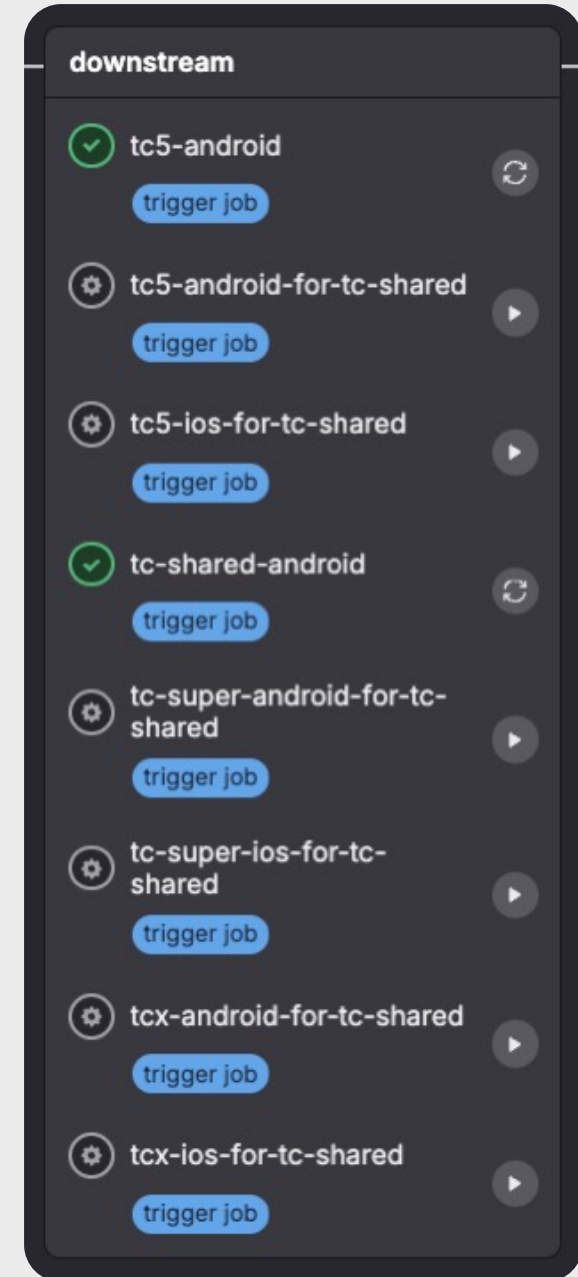
Merged Aleksandrov, Denis requested to merge feature/MOBTECH-567-remove...

Overview 1 Commits 7 Pipelines 6 Changes 23

# Преимущества (для всех)

1. Максимально быстрое переиспользование и правка кода

2. Единый пайплайн  
(пишем один CI/CD, одни триггеры, одну инфраструктуру)



# Преимущества (для всех)

1. Максимально быстрое переиспользование и правка кода

2. Единый пайплайн (пишем один CI/CD, одни триггеры, одну инфраструктуру)

3. Согласованность библиотек и продуктов

```
gradle/libs/libs.versions.toml
@@ -7,7 +7,7 @@ targetSdk = "36"
7 7   superapp-version-name = "0.0.1"
8 8
9 9   # Kotlin
10 10  - kotlin = "2.2.21"
11 11  + kotlin = "2.3.10"
12 12  # KotlinX
13 13  #https://github.com/Kotlin/kotlinx-datetime/releases
14 14  kotlin-datetime = "0.7.1"

tc-shared/core/camera/src/androidMain/kotlin/com/xfivetech/camera/domain/CameraController.android.kt
@@ -4,7 +4,6 @@ import android.content.Context
4 4   import android.graphics.Bitmap
5 5   import android.graphics.BitmapFactory
6 6   import android.graphics.Matrix
7 7   - import android.media.ExifInterface
8 8   import androidx.annotation.OptIn
9 9   import androidx.camera.core.Camera
10 10  import androidx.camera.core.CameraSelector
11 11  @@ -19,6 +18,7 @@ import androidx.camera.core.resolutionselector.ResolutionStrategy
19 18  import androidx.camera.lifecycle.ProcessCameraProvider
20 19  import androidx.camera.view.PreviewView
21 20  import androidx.core.content.ContextCompat
22 21  + import androidx.exifinterface.media.ExifInterface
23 22  import androidx.lifecycle.LifecycleOwner
24 23  import com.xfivetech.camera.domain.enums.FlashMode
25 24  import com.xfivetech.camera.domain.enums.QualityPrioritization
26 25  @@ -28,6 +28,7 @@ import kotlinx.coroutines.delay
28 28  import kotlinx.coroutines.flow.StateFlow
29 29  import kotlinx.coroutines.suspendCancellableCoroutine
30 30  import java.io.ByteArrayOutputStream
31 31  + import java.io.File
32 32  import java.util.concurrent.Executors
33 33  import kotlin.coroutines.resume
```

# Преимущества (для всех)

1. Максимально быстрое переиспользование и правка кода
2. Единый пайплайн  
(пишем один CI/CD, одни триггеры, одну инфраструктуру)
3. Согласованность библиотек и продуктов
4. Масштабируемость  
(+1 модуль сильно легче, чем +1 внешний проект – у нас уже ~350 модулей)



# А в нашем монорепе...

| Проект          | Строк кода        |
|-----------------|-------------------|
| tc-shared (KMP) | 181 632           |
| tc5-android     | 276 080           |
| tc5-ios         | 577 757           |
| tcx-android     | 236 931           |
| tcx-ios         | 244 001           |
| tc4-rn          | 305               |
| tc4-ios         | 11 448            |
| tc4-android     | 109 933           |
| tc4-shared      | 559               |
| legacy          | 33 995            |
| <b>Всего</b>    | <b>~1 672 641</b> |

| Проект       | Gradle-модулей |
|--------------|----------------|
| tc-shared    | 307            |
| tc5-android  | 27             |
| tcx-android  | 4              |
| tc4-rn       | 3              |
| tc4-ios      | 5              |
| tc4-android  | 4              |
| tc4-shared   | 22             |
| <b>Всего</b> | <b>372</b>     |

# Мифы про монорепозиторий

Миф



Как будто  
без разницы,  
за что деньги  
получать  
(например,  
ты в аутстаффе)

Не правда, потому что:

✘ Компания-получатель  
эффективнее работает

---

✘ Разработчик растёт  
как специалист

Миф



Нам платят  
за часы работы,  
а не за  
эффективность  
(для аутсорса)

Не правда, потому что:

✘ Эффективность аутсорса →  
конкурентное преимущество →  
больше денег

---

✘ Разработчик более востребован  
как специалист

Миф



Нужно  
много железа,  
компьютеры,  
подписки —  
дорого  
(геймдев, эдтех)

Не правда, потому что:

✘ Это всё не дорого в сравнении с зарплатным фондом

✘ Это повышает эффективность лучше, чем найм кожаных компютеров

✘ Хороший ноутбук/Mac mini стоит +100к к обычному железу

Миф



Придётся  
нанимать  
умных ребят

Не правда, потому что:

✘ Можно растить ребят внутри

---

✘ Это всё равно экономически оправдано

# 03

**Почему  
монорепозиторий  
нужен вам здесь  
и сейчас**



# Первый важный плюс - Single instance of truth



1. Единый источник правды  
действительно единый



# Первый важный плюс — Single instance of truth

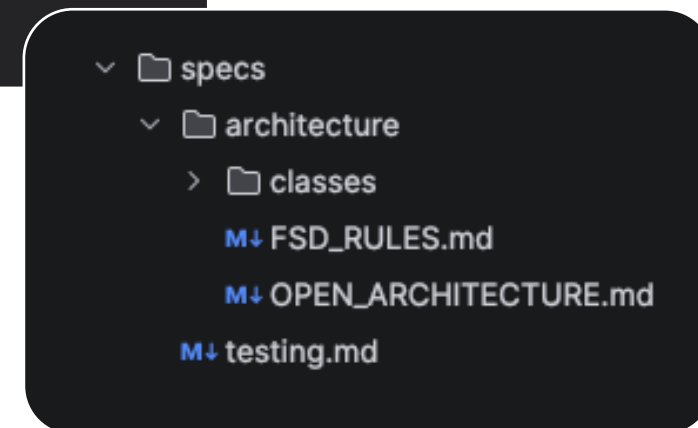
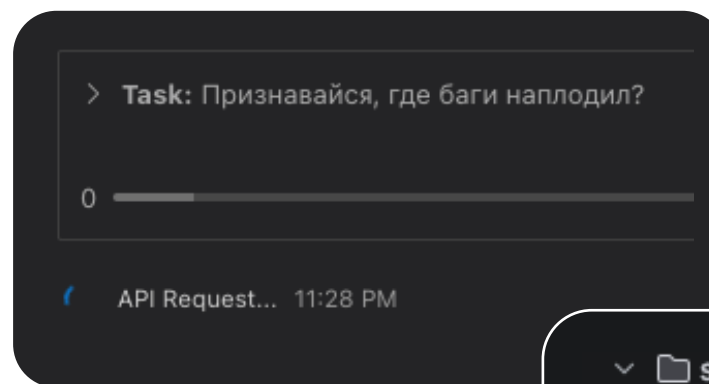


# Самый важный и жирный плюс – удобство для автоматизации и ИИ



1. Единый источник правды  
действительно единый

2. Удобный контекст и навигация  
для кожаных и ИИ



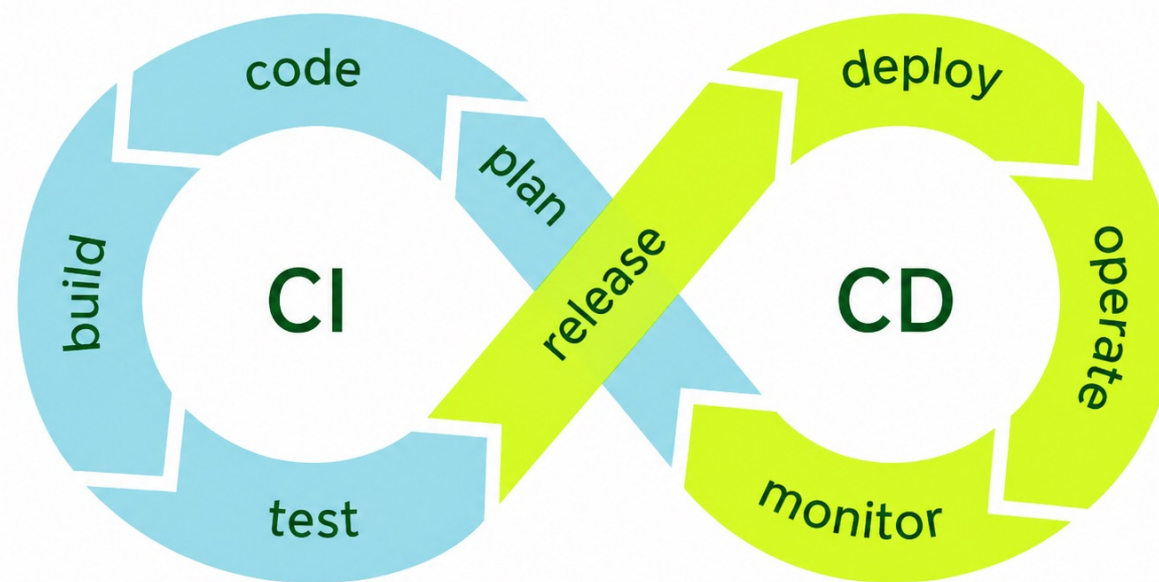
# Третий плюс – замкнутая автоматизация CI-CD



1. Единый источник правды действительно единый

2. Удобный контекст и навигация для кожаных и ИИ

3. Закрывает автоматический цикл разработки и деплоя



04

# Как реализовать монорепозиторий





1

В этом деле  
спешка  
не поможет

2

Есть безопасные  
маленькие шаги

3

Начать получать  
выгоду можно уже  
сегодня (завтра)

# Как сделать монорепу у себя и зажить хорошо (но потом)

0 Проверьте условия (найти релевантную гит документацию)

- LFS
- Squash commits
- Clear history

## Improving monorepo performance

A monorepo is a repository that contains sub-projects. A single application often contains interdependent projects. For example, a backend, a web frontend, an iOS application, and an Android application. Monorepos are common, but they can present performance risks. Some common problems:

- Large binary files.
- Many files with long histories.
- Many simultaneous clones and pushes.
- Vertical scaling limits.
- Network bandwidth limits.
- Disk bandwidth limits.



<https://docs.gitlab.com/user/project/repository/monorepos/>

# Как сделать монорепу у себя и зажить хорошо (но потом)

0 Проверьте условия (найди релевантную гит документацию)

1 Монорепозиторий

- Вы просто лежите в одной папке
- Можно постепенно выравниваться
- Но надо **rebase** заменить на **merge**

## Ключевые отличия rebase от merge

| Критерий                        | rebase  | merge  |
|---------------------------------|---|--|
| История коммитов                | Линейная, чистая. Нет «шума» от коммитов слияния  | Сохраняет полную историю, включая коммиты слияния ( Merge branch 'feature' into main ) |
| Коммиты                         | Создаёт <i>новые</i> коммиты (даже если код не изменился)   | Сохраняет оригинальные коммиты без изменений   |
| Сложность разрешения конфликтов | Конфликты разрешаются <i>по одному</i> , в порядке следования коммитов                                  | Все конфликты разрешаются <i>за один раз</i> при создании коммита слияния              |
| Безопасность                    | Опасен для <i>опубликованных</i> веток: переписывание истории может сломать работу других разработчиков | Безопасен для опубликованных веток: не изменяет существующие коммиты                   |
| Визуализация                    | Простая, прямая линия развития  | Граф с «ответвлениями» и точками слияния   |

# Как сделать монорепу у себя и зажить хорошо (но потом)

0 Проверьте условия (найди релевантную гит документацию)

1 Монорепозиторий

- Вы просто лежите в одной папке
- Можно постепенно выравниваться
- Но надо **rebase** заменить на **merge**
- **Squash commits**

## Ключевые отличия rebase от merge

| Критерий                        | rebase  | merge  |
|---------------------------------|---|--|
| История коммитов                | Линейная, чистая. Нет «шума» от коммитов слияния  | Сохраняет полную историю, включая коммиты слияния ( Merge branch 'feature' into main ) |
| Коммиты                         | Создаёт <i>новые</i> коммиты (даже если код не изменился)   | Сохраняет оригинальные коммиты без изменений   |
| Сложность разрешения конфликтов | Конфликты разрешаются <i>по одному</i> , в порядке следования коммитов                                  | Все конфликты разрешаются <i>за один раз</i> при создании коммита слияния              |
| Безопасность                    | Опасен для <i>опубликованных</i> веток: переписывание истории может сломать работу других разработчиков | Безопасен для опубликованных веток: не изменяет существующие коммиты                   |
| Визуализация                    | Простая, прямая линия развития  | Граф с «ответвлениями» и точками слияния   |

Да, вот так  
просто

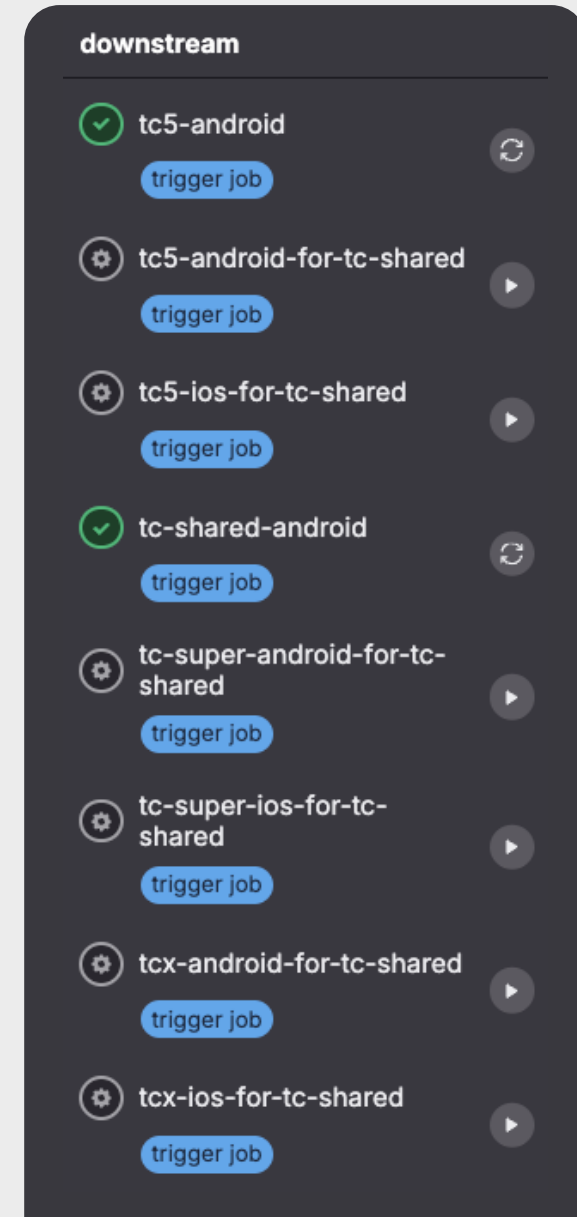


## Структура проекта

```
tc-mono/
├── build-scripts/           # Общие скрипты сборки
├── code-quality/           # Конфигурация качества кода
├── config/                 # Общие конфигурации
├── gradle/                 # Конфигурация Gradle
│   └── build-system/       # Кастомные плагины сборки
├── plugins/               # Плагины
├── legacy/                # Устаревшие модули
├── tc-shared/             # Общие модули Kotlin Multiplatform
├── tc5-android/           # Android приложение Пятёрочки
├── tc5-ios/               # iOS приложение Пятёрочки
├── tcx-android/           # Android приложение Перекрёстка
├── tcx-ios/               # iOS приложение Перекрёстка
├── build.gradle.kts        # Корневой build.gradle
├── settings.gradle.kts     # Настройки проекта
└── gradle.properties      # Свойства Gradle
```

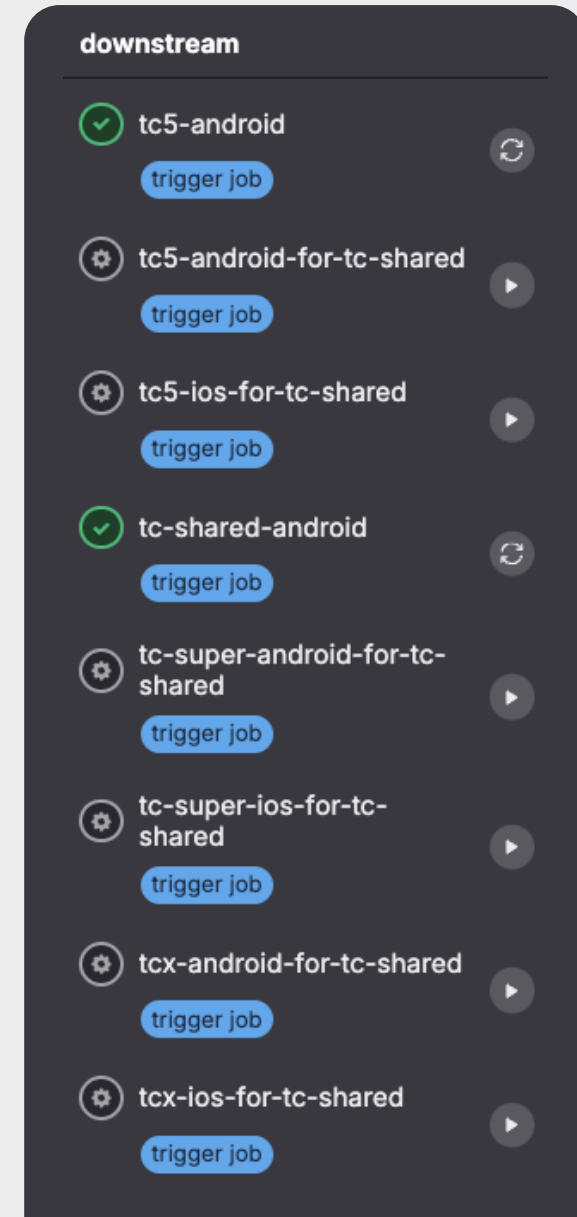
# Как сделать монорепу у себя и зажить хорошо (но потом)

- 0 Проверьте условия (найти релевантную гит документацию)
- 1 Монорепозиторий
- 2 CI/CD



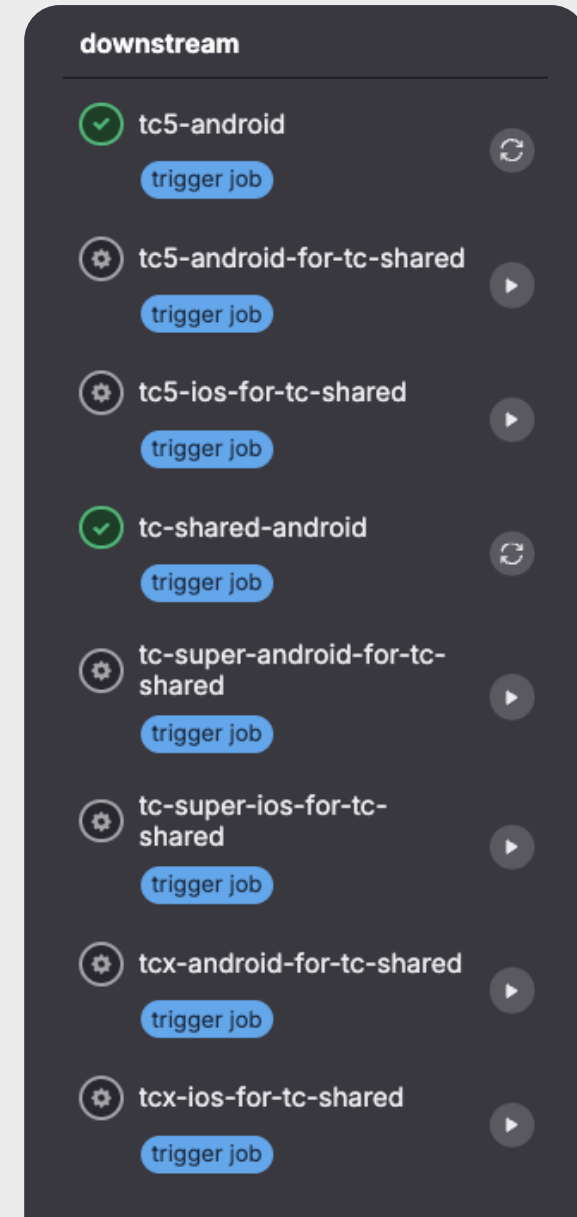
# Как сделать монорепу у себя и зажить хорошо (но потом)

- 0 Проверьте условия (найти релевантную гит документацию)
- 1 Монорепозиторий
- 2 CI/CD
  - Добавляем билды проектов



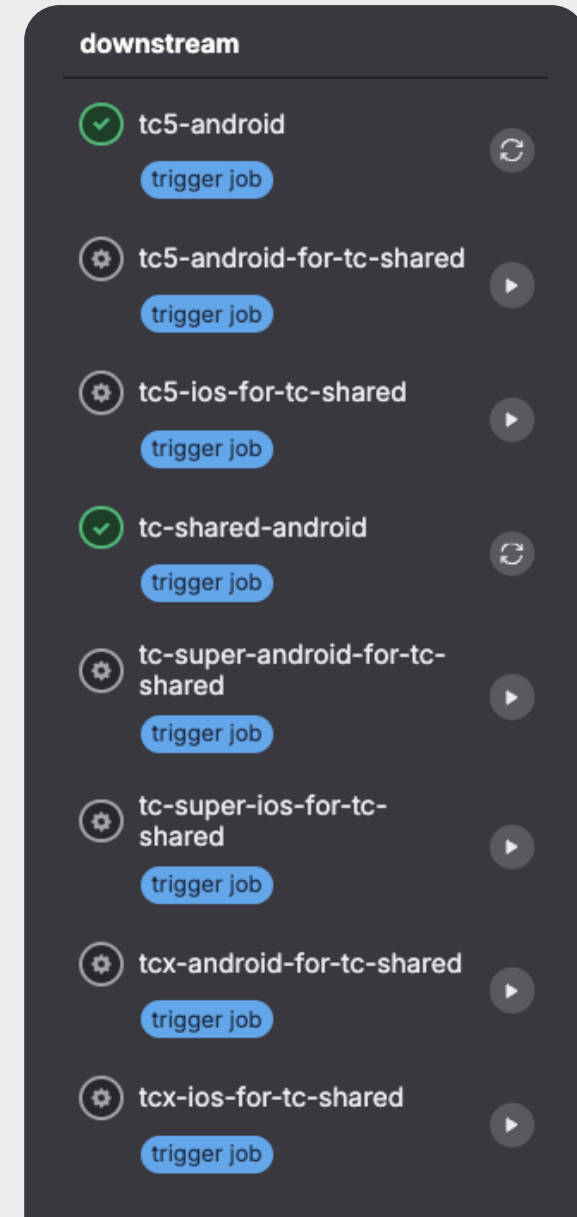
# Как сделать монорепу у себя и зажить хорошо (но потом)

- 0 Проверьте условия (найти релевантную гит документацию)
- 1 Монорепозиторий
- 2 CI/CD
  - Добавляем билды проектов
  - Умная автоматика изменений (проверяем то, что изменилось)



# Как сделать монорепу у себя и зажить хорошо (но потом)

- 0 Проверьте условия (найти релевантную гит документацию)
- 1 Монорепозиторий
- 2 CI/CD
  - Добавляем билды проектов
  - Умная автоматика изменений (проверяем то, что изменилось)
  - Добавляем Unit- / e2e-тесты



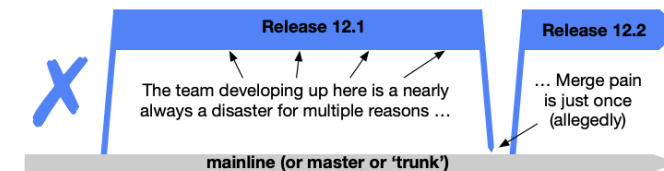
# Как сделать монорепу у себя и зажить хорошо (но потом)

- 0 Проверьте условия (найти релевантную гит документацию)
- 1 Монорепозиторий
- 2 CI/CD
- 3 TBD, или trunk based development

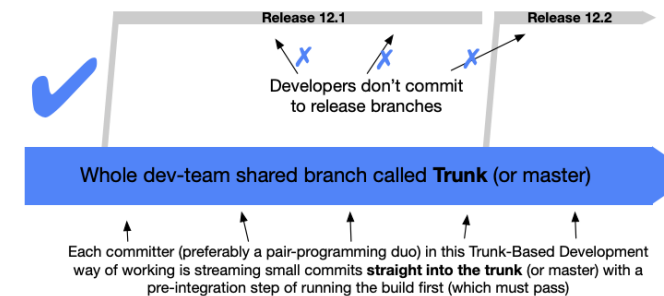
<https://trunkbaseddevelopment.ru/>



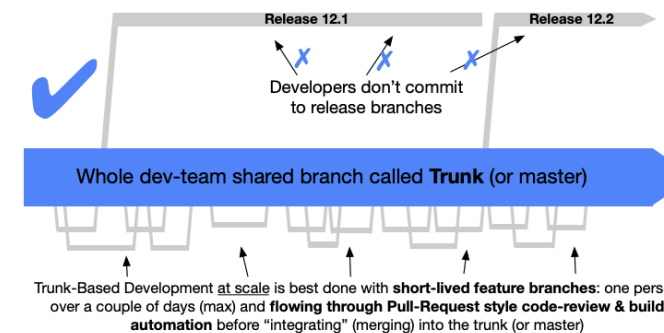
Общие ветки вне mainline/master/trunk это плохо при любом релизном цикле:



Trunk-Based Development для небольших команд:

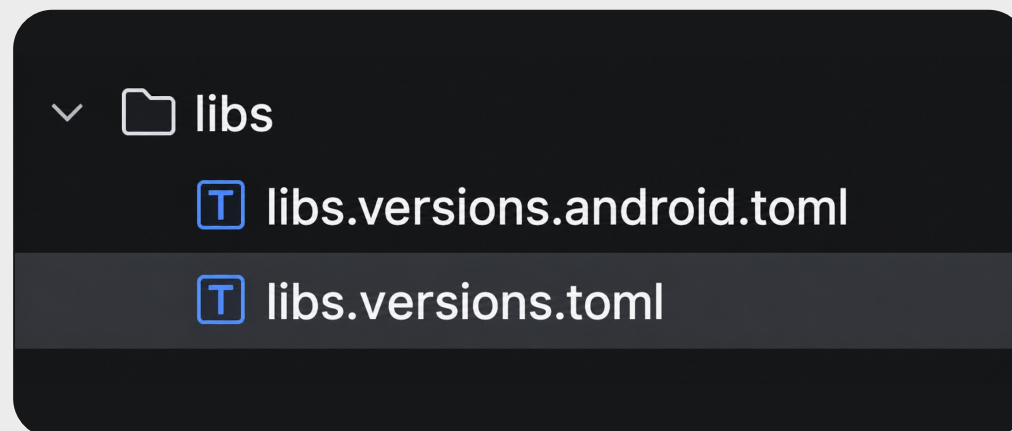


Масштабированный Trunk-Based Development:



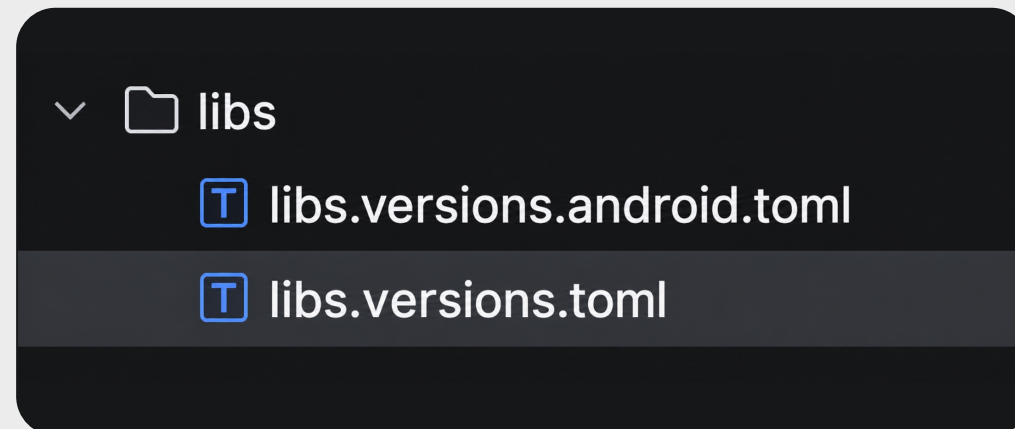
# Как сделать монорепу у себя и зажить хорошо (но потом)

- 0 Проверьте условия (найди релевантную гит документацию)
- 1 Монорепозиторий
- 2 CI/CD
- 3 TBD, или trunk based development
- 4 Выверните версии проектов
  - Заведите единые томлы



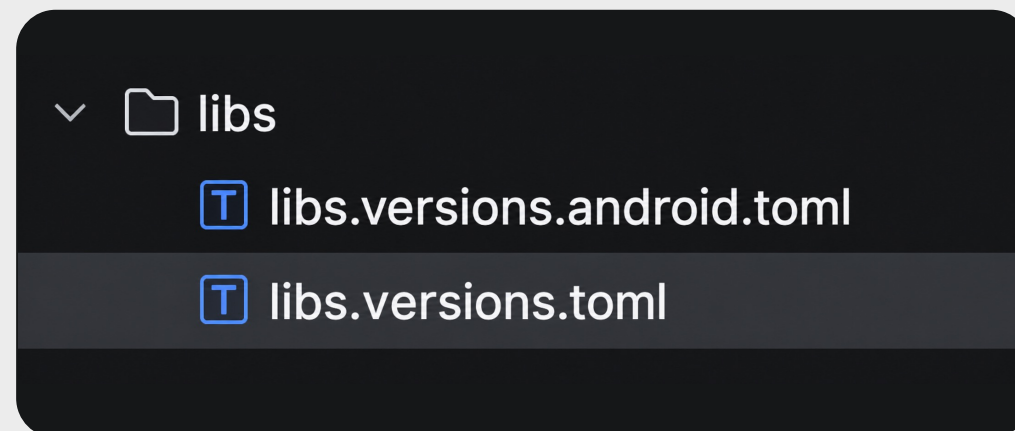
# Как сделать монорепу у себя и зажить хорошо (но потом)

- 0 Проверьте условия (найти релевантную гит документацию)
- 1 Монорепозиторий
- 2 CI/CD
- 3 TBD, или trunk based development
- 4 **Выровняйте версии проектов**
  - Заведите единые томлы
  - Томлов может быть несколько



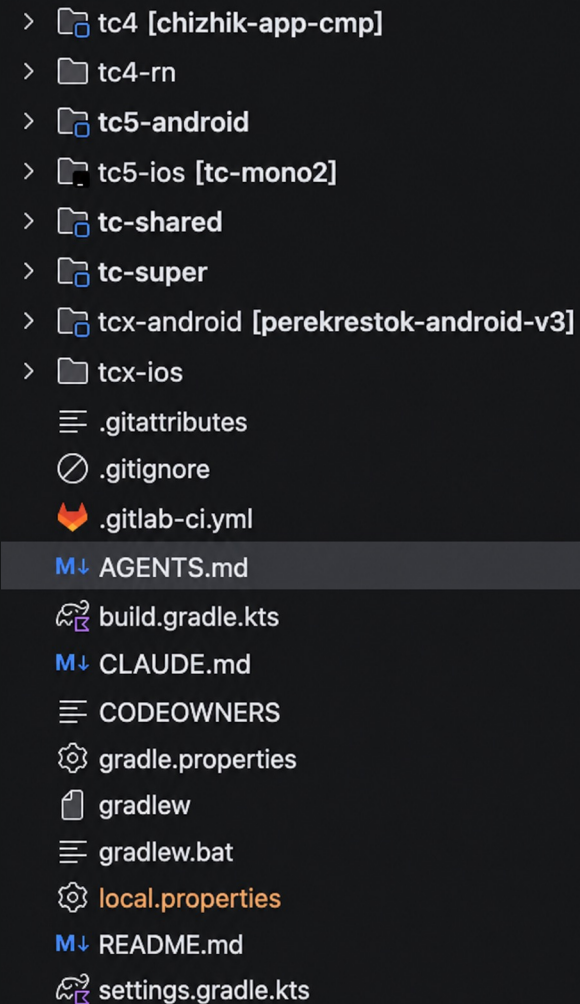
# Как сделать монорепу у себя и зажить хорошо (но потом)

- 0 Проверьте условия (найти релевантную гит документацию)
- 1 Монорепозиторий
- 2 CI/CD
- 3 TBD, или trunk based development
- 4 **Выровняйте версии проектов**
  - Заведите единые томлы
  - Томлов может быть несколько
  - Удаляйте мусорные зависимости



# Как сделать монорепу у себя и зажить хорошо (но потом)

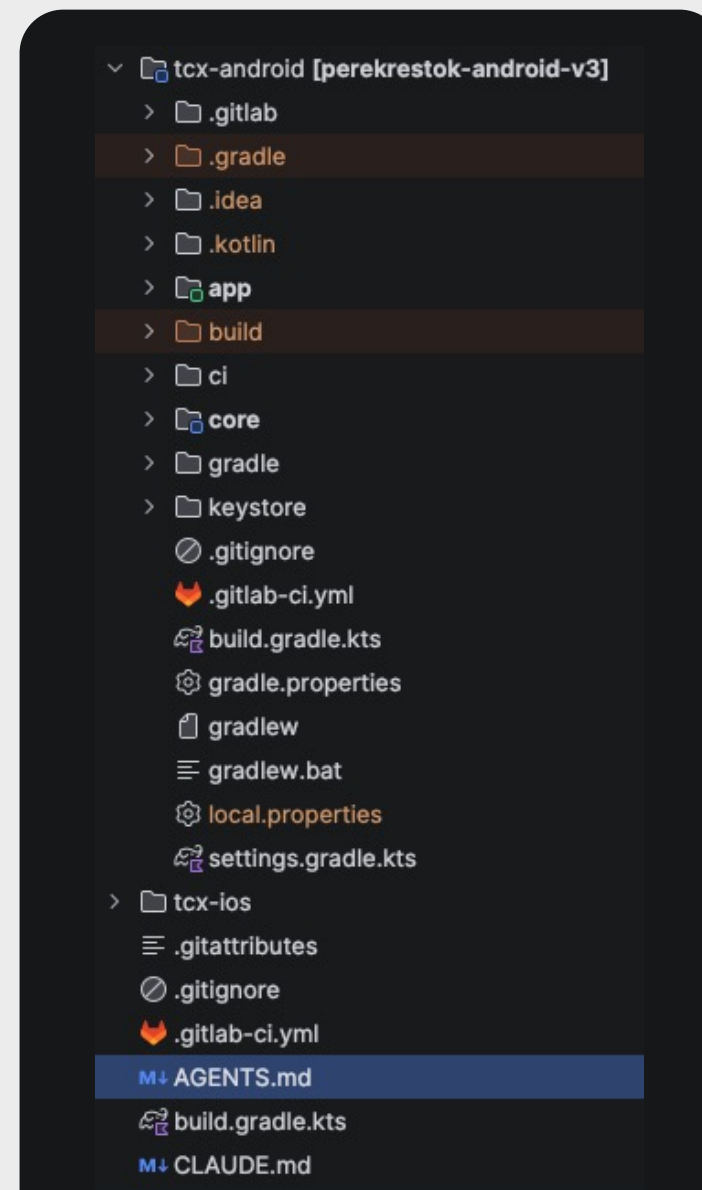
- 0 Проверьте условия (найти релевантную гит документацию)
- 1 Монорепозиторий
- 2 CI/CD
- 3 TBD, или trunk based development
- 4 Выровняйте версии проектов
- 5 Монопроект
  - Общий билд конфиг
  - Единый gradle для объединения подпроектов



```
> tc4 [chizhik-app-cmp]
> tc4-rn
> tc5-android
> tc5-ios [tc-mono2]
> tc-shared
> tc-super
> tcx-android [perekrestok-android-v3]
> tcx-ios
  .gitattributes
  .gitignore
  .gitlab-ci.yml
M↓ AGENTS.md
  build.gradle.kts
M↓ CLAUDE.md
  CODEOWNERS
  gradle.properties
  gradlew
  gradlew.bat
  local.properties
M↓ README.md
  settings.gradle.kts
```

# Как сделать монорепу у себя и зажить хорошо (но потом)

- 0 Проверьте условия (найти релевантную гит документацию)
- 1 Монорепозиторий
- 2 CI/CD
- 3 TBD, или trunk based development
- 4 Выровняйте версии проектов
- 5 **Монопроект**
  - **Общий билд конфиг**
  - **Единый gradle для объединения подпроектов**
  - **Отдельные конфиги для подпроектов (можно открывать для ускорения работы)**



# Как сделать монорепу у себя и зажить хорошо (но потом)

- 0 Проверьте условия (найти релевантную гит документацию)
- 1 Монорепозиторий
- 2 CI/CD
- 3 TBD, или trunk based development
- 4 Выровняйте версии проектов
- 5 Монопроект
- 6 Тесты, тесты, тесты



# Как сделать монорепу у себя и зажить хорошо (но потом)

- 0 Проверьте условия (найти релевантную гит документацию)
- 1 Монорепозиторий
- 2 CI/CD
- 3 TBD, или trunk based development
- 4 Выровняйте версии проектов
- 5 Монопроект
- 6 Тесты, тесты, тесты
- 7 Detekt / kover / ai checks



# Как сделать монорепу у себя и зажить хорошо (но потом)

- 0 Проверьте условия (найти релевантную гит документацию)
- 1 Монорепозиторий
- 2 CI/CD
- 3 TBD, или trunk based development
- 4 Выровняйте версии проектов
- 5 Монопроект
- 6 Тесты, тесты, тесты
- 7 Detekt / kover / ai checks
- 8 Развивайте инфру
  - Очереди задач (интеграция)



# Как сделать монорепу у себя и зажить хорошо (но потом)

- 0 Проверьте условия (найти релевантную гит документацию)
- 1 Монорепозиторий
- 2 CI/CD
- 3 TBD, или trunk based development
- 4 Выровняйте версии проектов
- 5 Монопроект
- 6 Тесты, тесты, тесты
- 7 Detekt / kover / ai checks
- 8 Развивайте инфру
  - Очереди задач (интеграция)
  - Перекрестные проверки (меняете одно – проверяете другое)



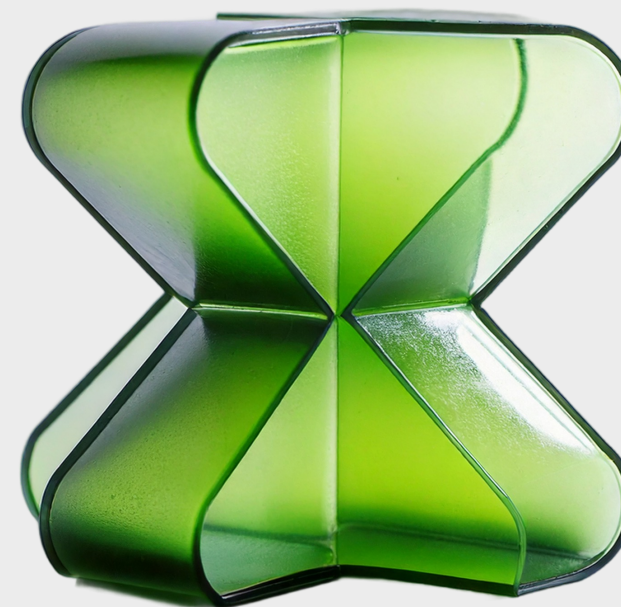
# Как сделать монорепу у себя и зажить хорошо (но потом)

- 0 Проверьте условия (найти релевантную гит документацию)
- 1 Монорепозиторий
- 2 CI/CD
- 3 TBD, или trunk based development
- 4 Выровняйте версии проектов
- 5 Монопроект
- 6 Тесты, тесты, тесты
- 7 Detekt / kover / ai checks
- 8 Развивайте инфру
  - Очереди задач (интеграция)
  - Перекрестные проверки (меняете одно – проверяете другое)
  - Отчёты и автоматика

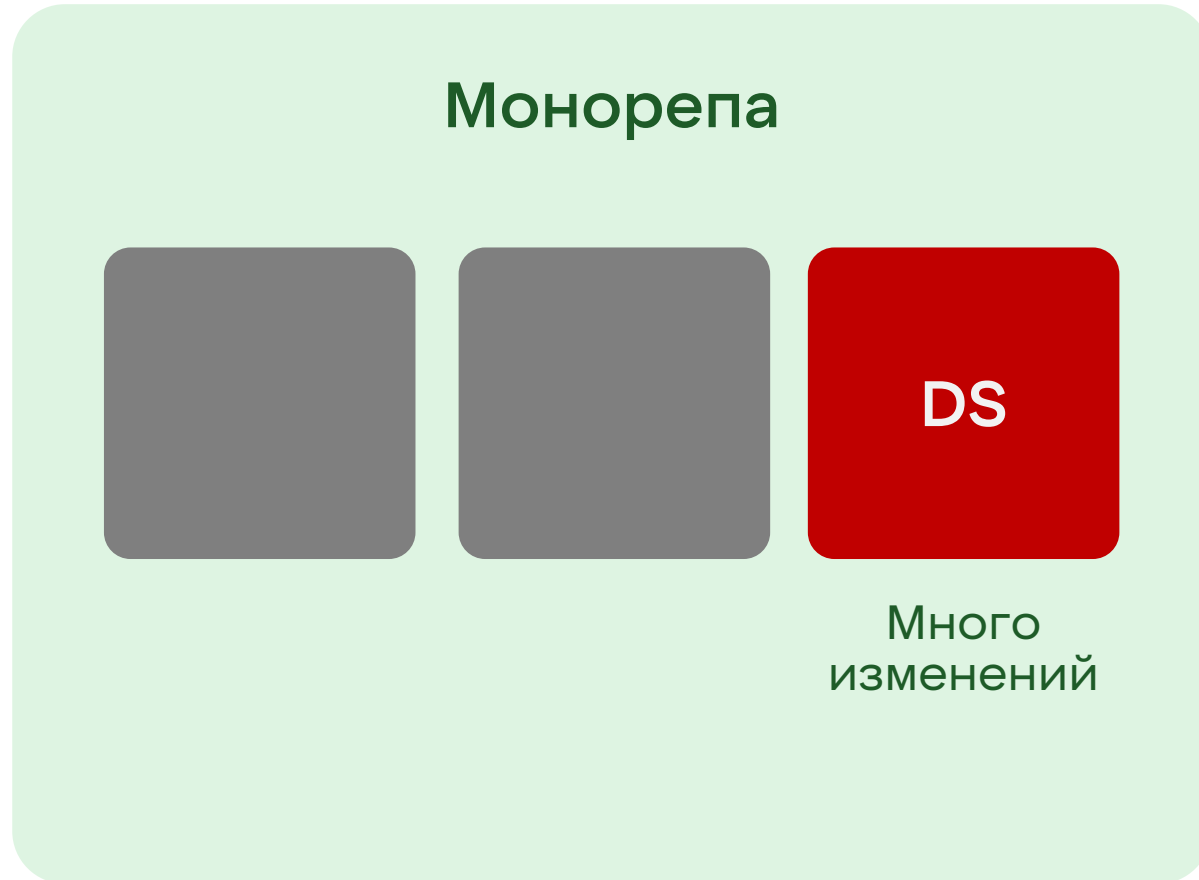


# Как сделать монорепу у себя и зажить хорошо (но потом)

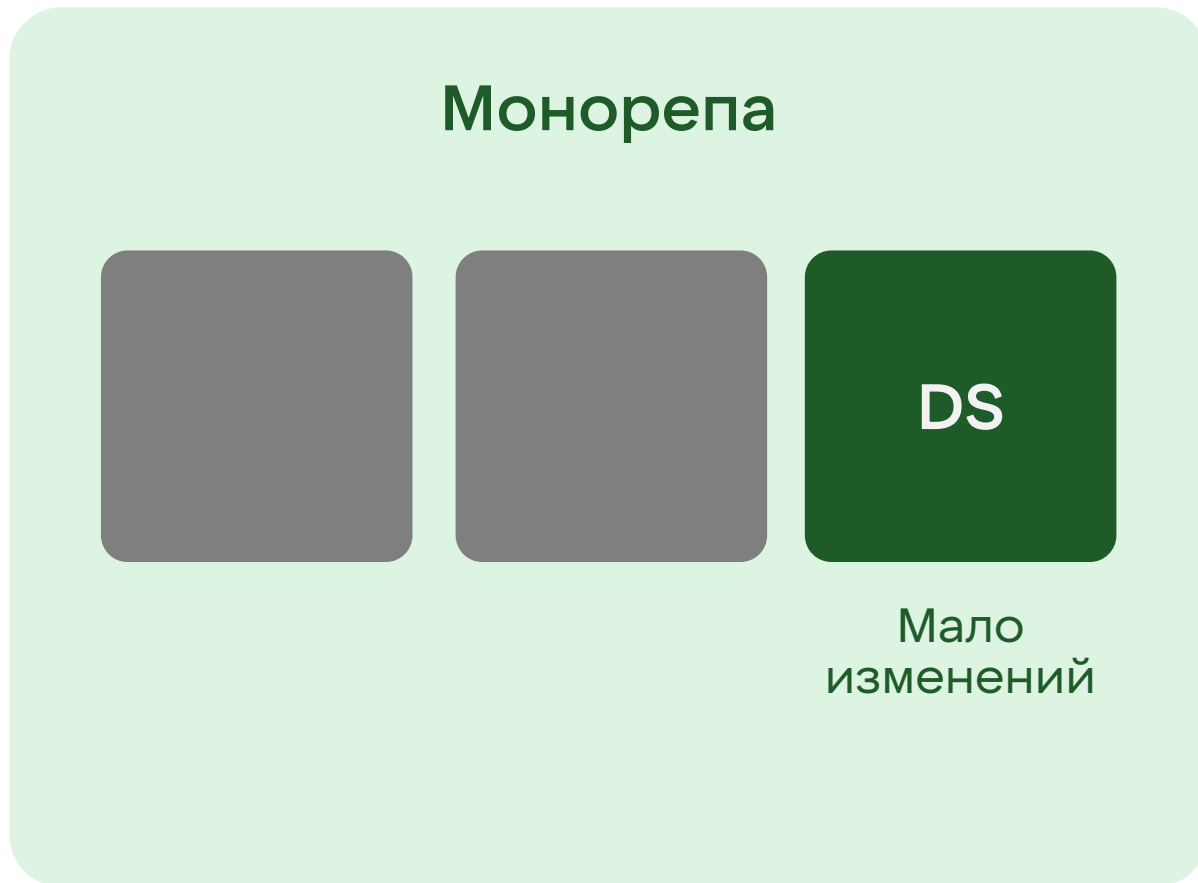
- 0 Проверьте условия (найти релевантную гит документацию)
- 1 Монорепозиторий
- 2 CI/CD
- 3 TBD, или trunk based development
- 4 Выровняйте версии проектов
- 5 Монопроект
- 6 Тесты, тесты, тесты
- 7 Detekt / kover / ai checks
- 8 Развивайте инфру
- 9 ...Можно разъезжаться в другие репозитории



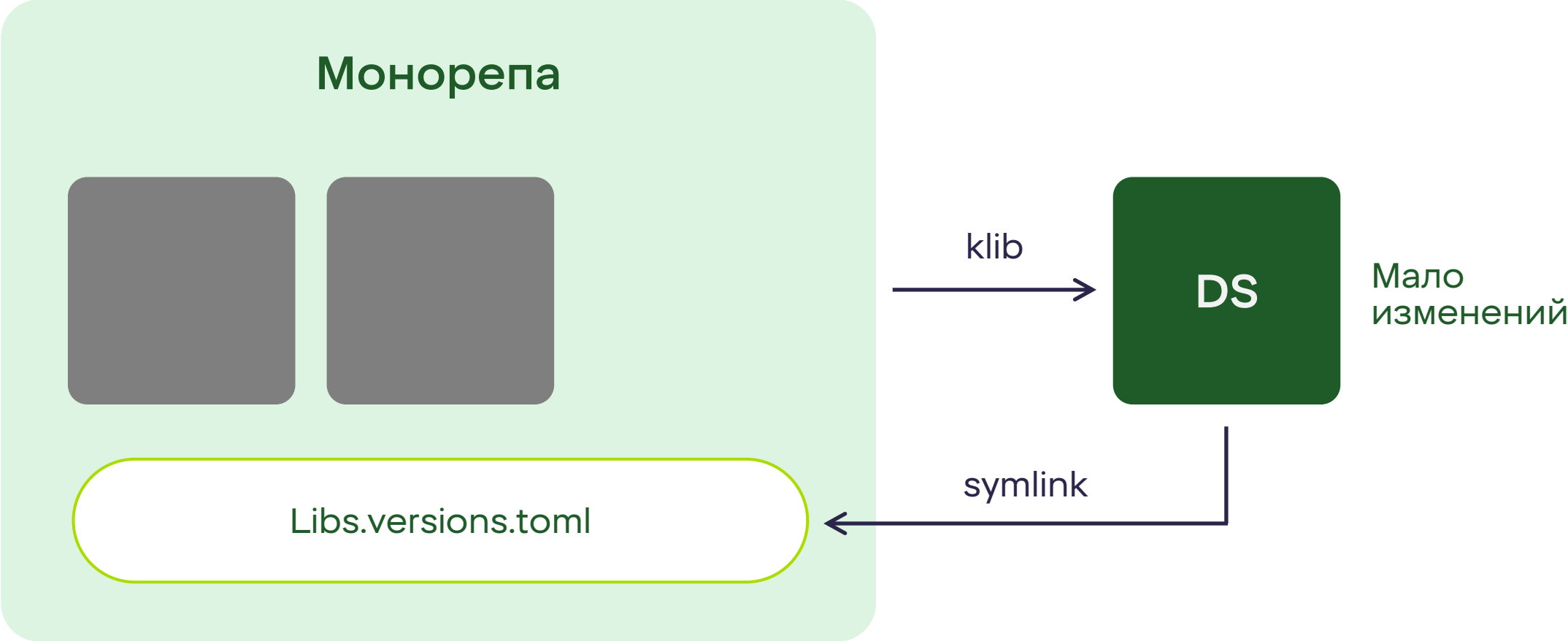
# Когда пора съезжать



# Когда пора съезжать



# Когда пора съезжать



# Итого: ПЛЮСЫ

- ✓ Можно работать с отдельными проектами, как будто ничего не произошло
- ✓ Открывать общий проект
- ✓ Все плюсы монорепозитория
  - Максимально быстрое переиспользование и правка кода
  - Единый пайплайн (пишем один CI/CD, одни триггеры, одну инфраструктуру)
  - Согласованность библиотек и продуктов
  - Масштабируемость (+1 модуль сильно легче, чем +1 внешний проект – у нас уже ~250 модулей)
  - Работа с ИИ



# Когда не надо

- ✗ Разные работы
- ✗ Код с минимальными изменениями за последний промежуток времени (делайте либы)
- ✗ Версии библиотек тоже можно вынести и повесить на simlink, чтобы синхронизироваться с библиотеками



**Сомневаетесь?  
Приходите к нам  
на стенд!**

**Поштормим ваши  
возможности монорепы 🤗**

# Спасибо! Вопросы?

Денис Александров  
Менеджер разработки,  
X5 Tech



tg: @guitariz



vk: d\_aleksandrov