

# Такая разная разработка

Александр Нозик для RSHB Backend Dev Meetup

# Обо мне



- Директор Центра Научного Программирования.
- К. ф.–м. н. по физике частиц.
- Преподаватель МФТИ.
- (Со-)руководитель московского KUG.
  
- <https://sciprogramming.center/people/Nozik>
- <https://twitter.com/noraltavir>
- <https://t.me/noraltavir>

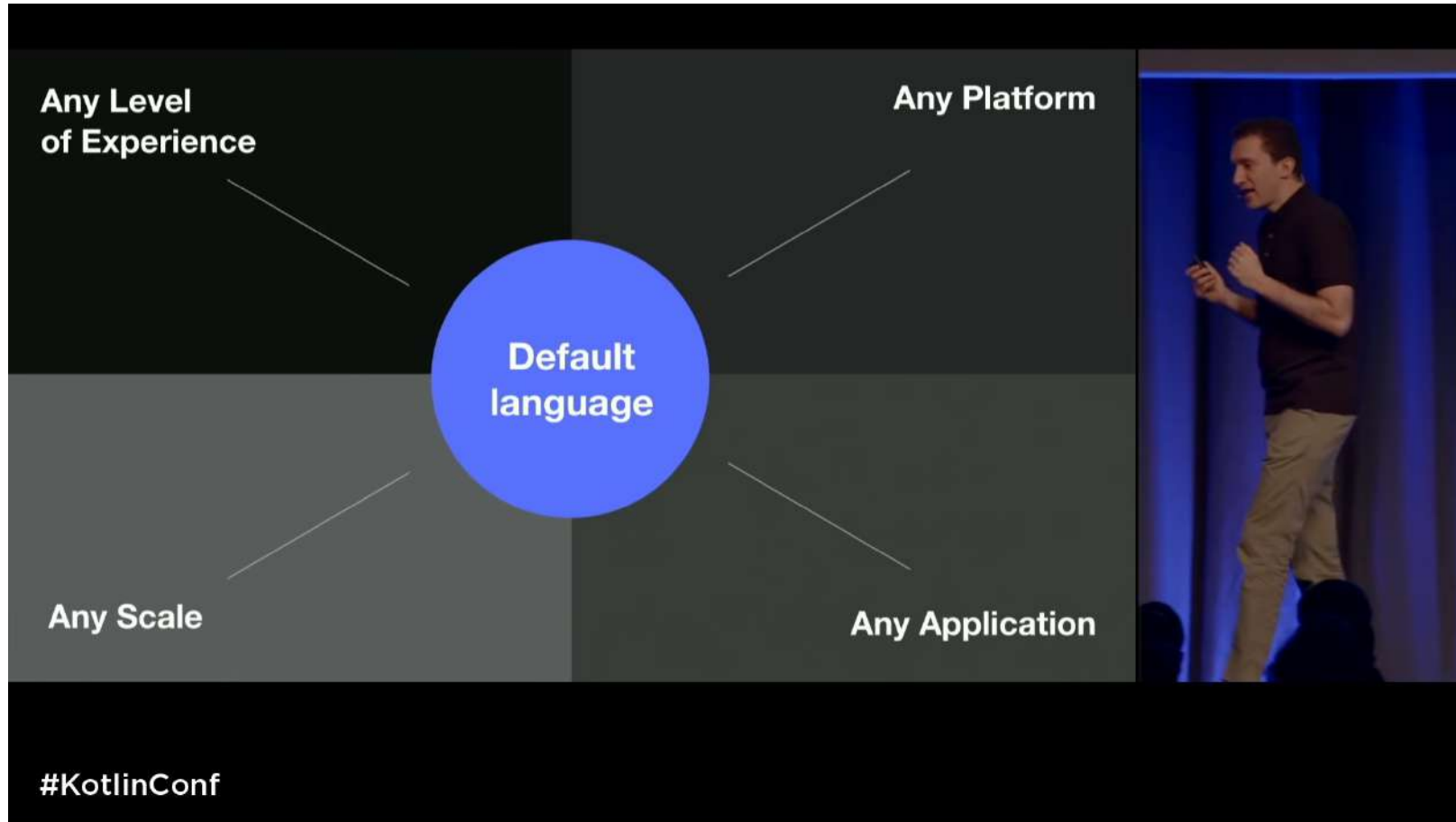


# Все говорят, разработка...

- Какой язык выбрать?
- Какой фреймворк выбрать?
- Какую команду выбрать?
- Какой формат работы с командой выбрать?

Disclaimer:  
Не обязательно Kotlin

# Kotlin – default language



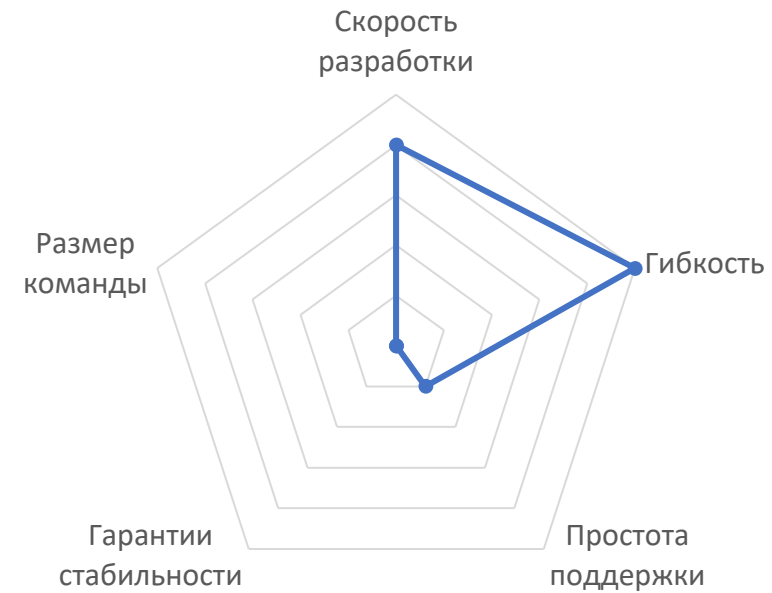
<https://youtu.be/0xKTM0A8gdl>

# Пет-проект

- Делаем то, что нравится на том, что нравится.

НО

- Пет проект может превратится в что-то другое...



# Пет проект на Kotlin

Ktor

Легковесный веб-фреймворк

Kotlinx-coroutines

Лучший асинхронный фреймворк

Kotlinx-serialization

compile-time фреймворк для сериализации

Kotlin/JS

Пишем фронтенд, не отходя от кассы

Compose-multiplatform

«Декларативный» фреймворк на все платформы

Kotlin/wasm

Модная и молодежная платформа



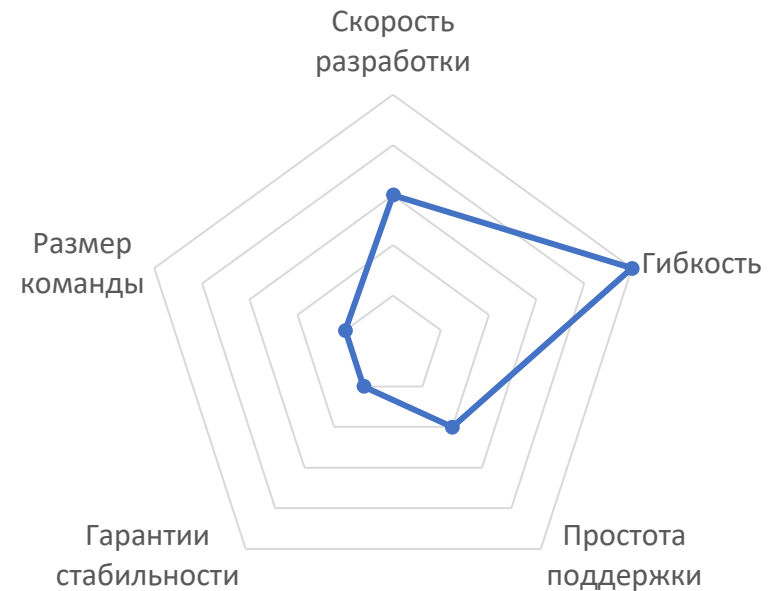
# Команда пет проекта

Created with Bing

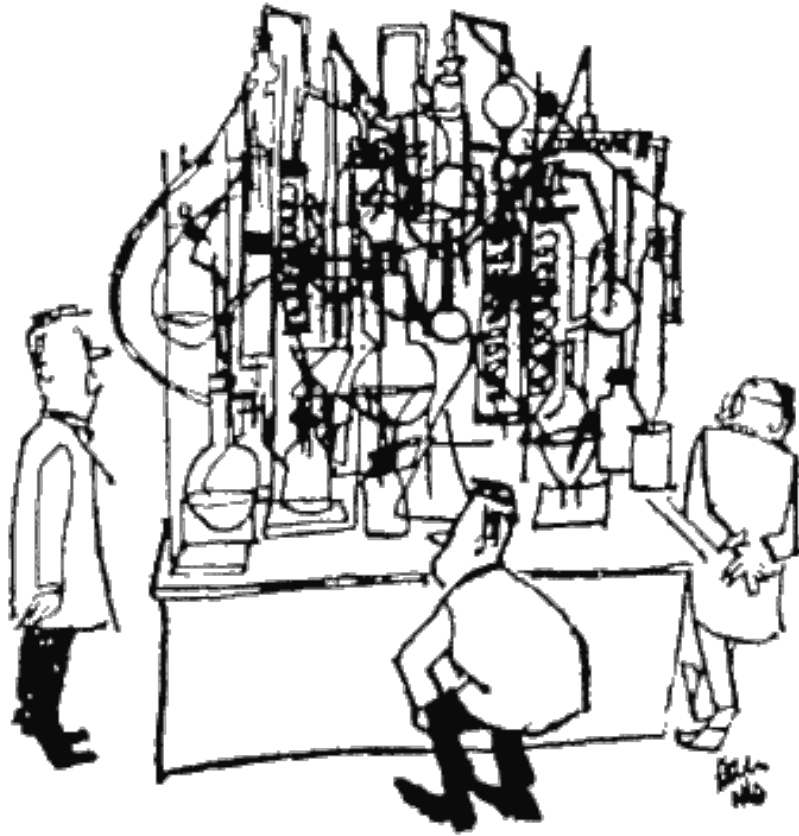


# Исследования

- Фаза исследований предшествует любому крупному проекту.
- Задача исследования – определить выполнимость проекта, а также наилучший набор инструментов для него







## Команда исследований

- Scrum не работает.
- ТЗ пишется по ходу работы.
- Нужно тщательно следить за состоянием базы знаний (там главный результат проекта).

# Исследования на Kotlin

Ktor

Compose-multiplatform

Kotlinx-coroutines

Kotlin/wasm

Kotlinx-serialization

Kotlin/JS

Xodus

Встраиваемая ключ-значение  
база данных (YouTrack)

Compiler plugins

Compile-time  
модификация кода

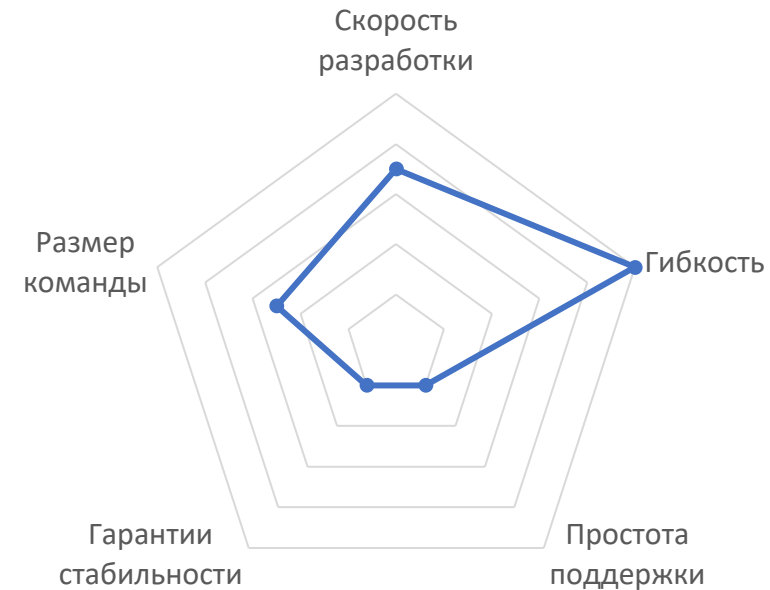
Exposed

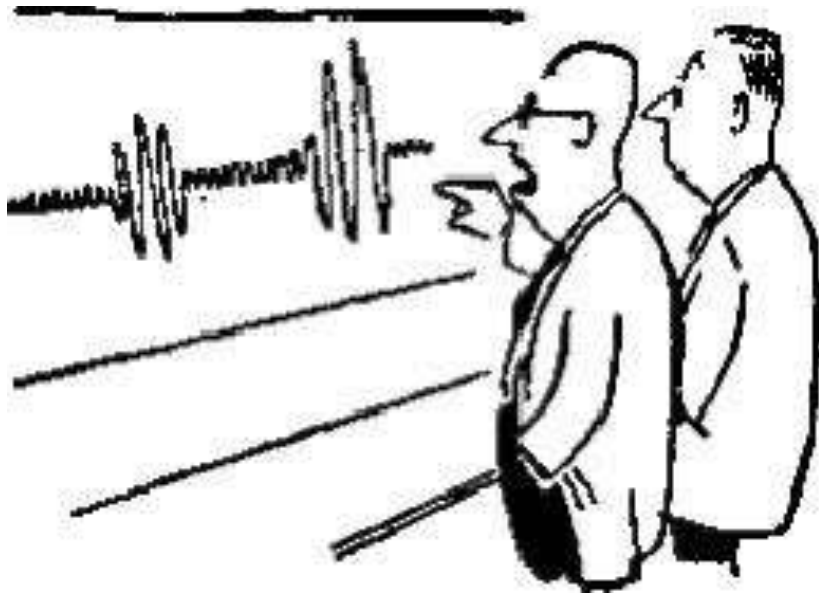
ORM + SQL builder



# Аналитика и DS

- Очень широкая область:
  - Data engineering;
  - Аналитика данных (рынка);
  - Разработка методик
  - Автоматизированные системы анализа данных;
- Стык разных профессий и разных процессов





## Аналитик в команде

- Аналитик – часто не программист. Надо объяснить программисту, что хотят аналитики, а аналитикам, что хотят программисты.
- У аналитиков часто Python в анамнезе.
- Большая нагрузка на DevOps из-за интеграции разных решений на разных языках.
- Аналитические задачи часто плохо дробятся на подзадачи.

# Аналитика на Kotlin

## Kotlin notebooks

Ноутбуки  
на Kotlin  
для Kotlin

<https://youtu.be/4Sg2Qju67kE>

Lets-plot  
(как GGPlot)  
Plotly-kt  
(как Plotly)

Kotlin-spark  
Крутая библиотека  
для работы со Spark

## KotlinDL

как Keras

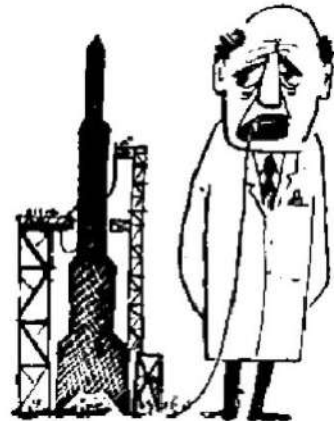
<https://kotlinconf.com/talks/385181/>

## Kotlin-dataframe

Как Pandas

# Стартап

- Побыстрее в прод, чтобы дали денег.
- Уникальные специалисты. Но все наперечет.
- Надо думать о росте (а вдруг выстрелит?).



# Команда стартапа

- Мало людей, которые выполняют много ролей.
- Легко попасть в ловушку скорости первичной разработки в ущерб расширяемости.
- Надо думать о расширении команды.





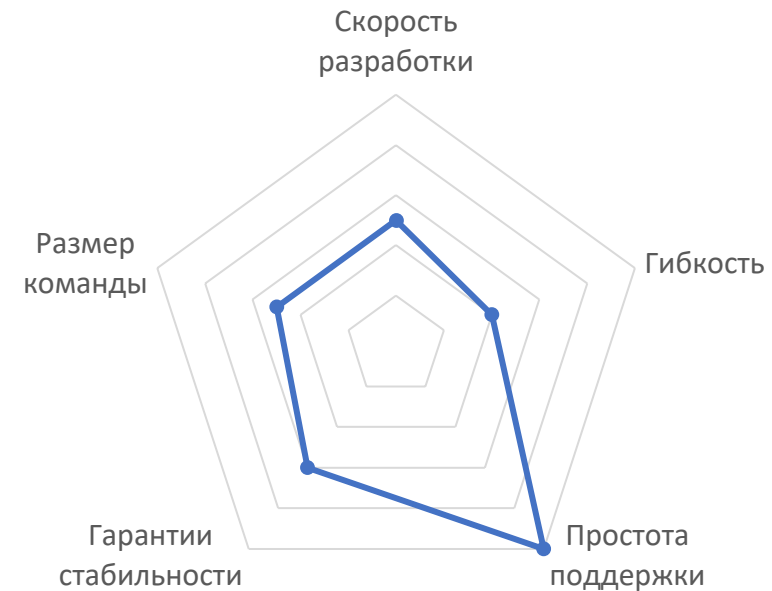
# Kotlin для стартапа

- Full-stack JS + JVM
- Kotlin Multiplatform Mobile
- Compose-multiplatform
- Kotlinx-coroutines
- Kotlinx-serialization
- Exposed



# Пользовательское приложение

- Короткий (сравнительно) жизненный цикл
- Много изменений
- Разные команды могут работать над разными кусками



# Kotlin для приложений



Dall-E так видит покраску кнопок

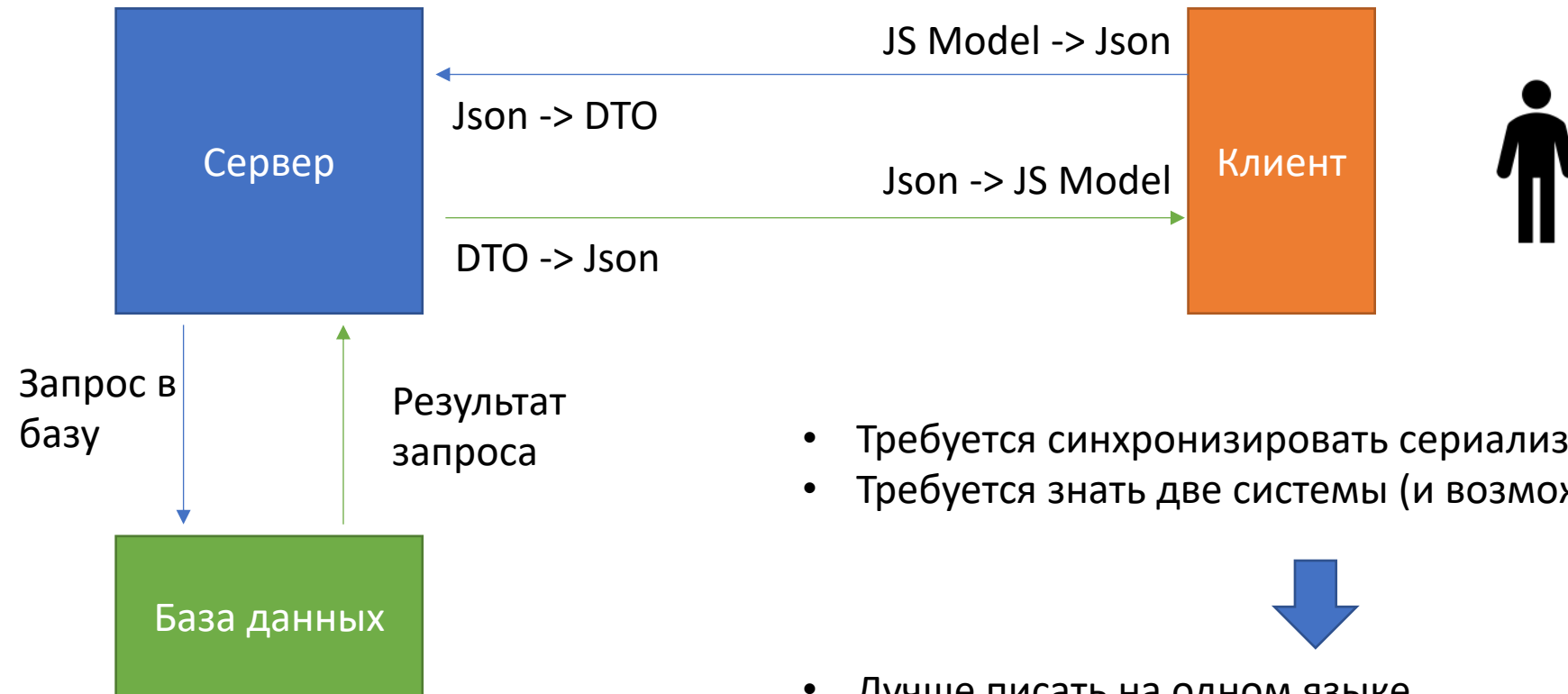
- Full-stack JS + JVM
- Kotlin Multiplatform Mobile
- Compose-multiplatform
- Kotlinx-coroutines
- Kotlinx-serialization
- Exposed

# Не все приложения одинаковы

- Приложение часто имеет две компоненты front-end и back-end.
- Front-end и back-end команды работают по-разному и в общем случае на разном стеке. Нужны дополнительные инструменты и люди для того, чтобы синхронизировать изменения с двух сторон.
- Все это можно упростить, если писать и front и back на одном языке. Хотя, разумеется, это не бесплатно.
- Основная проблема – верстка графических интерфейсов.

# Full-stack. Что это?

Невыносимая легкость  
визуализации данных в  
Kotlin full stack, Joker 2022  
<https://youtu.be/uT5j-xOXC3E>



- Требуется синхронизировать сериализатор DTO и JS.
- Требуется знать две системы (и возможно два языка)



- Лучше писать на одном языке

Обычно это JS

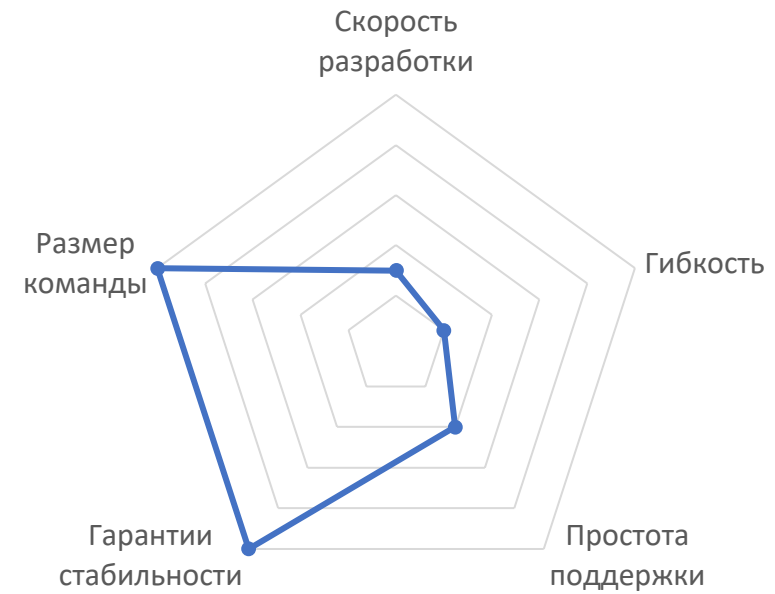
# Kotlin/Wasm – будущее рядом?

- Чего нам не хватает для полного счастья с Kotlin на Front-end?
- Компактные бандлы.
- Отсутствие JS!
  
- Kotlin 1.8.20 включает экспериментальную поддержку Wasm со сборкой мусора.



# Энтерпрайз

- Главное – доступность кадров.



# Kotlin в энтерпрайзе

- **Проблема: Kotlin сложнее Java.**
- Kotlin можно использовать в одном приложении с Java, но не идиоматичный Kotlin.
- Идиоматичный Kotlin это как минимум extension-oriented design  
(<https://elizarov.medium.com/extension-oriented-design-13f4f27deae6>)

<https://kotlinlang.org/lp/server-side/>

Vert.x

Quarkus

Spring/MVC

GRPC

Spring/WebFlux

RSocket

Kotlinx-html

HTML как код

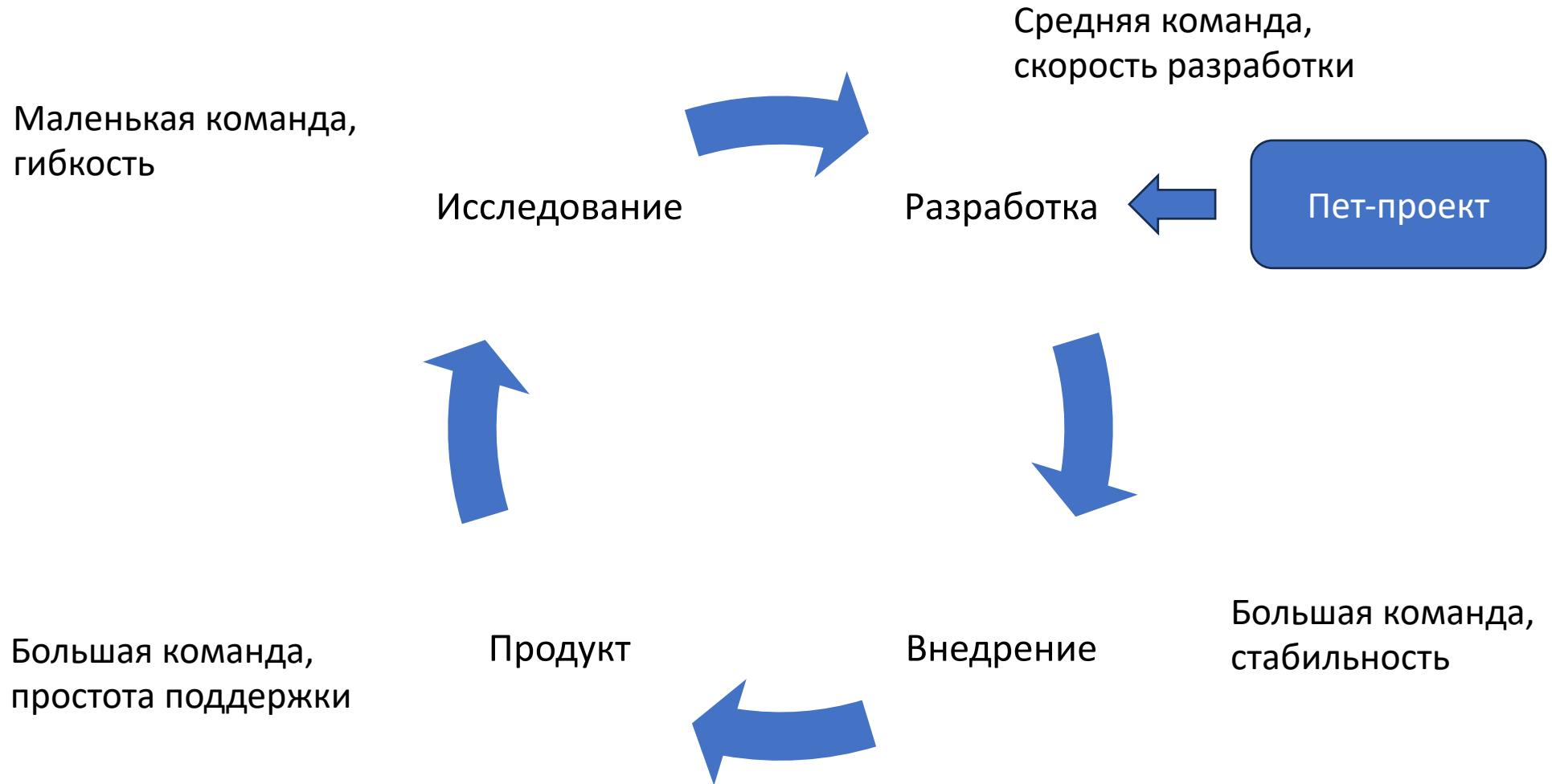
Kotlin-graphql

Kafka

# Круговорот кода в природе

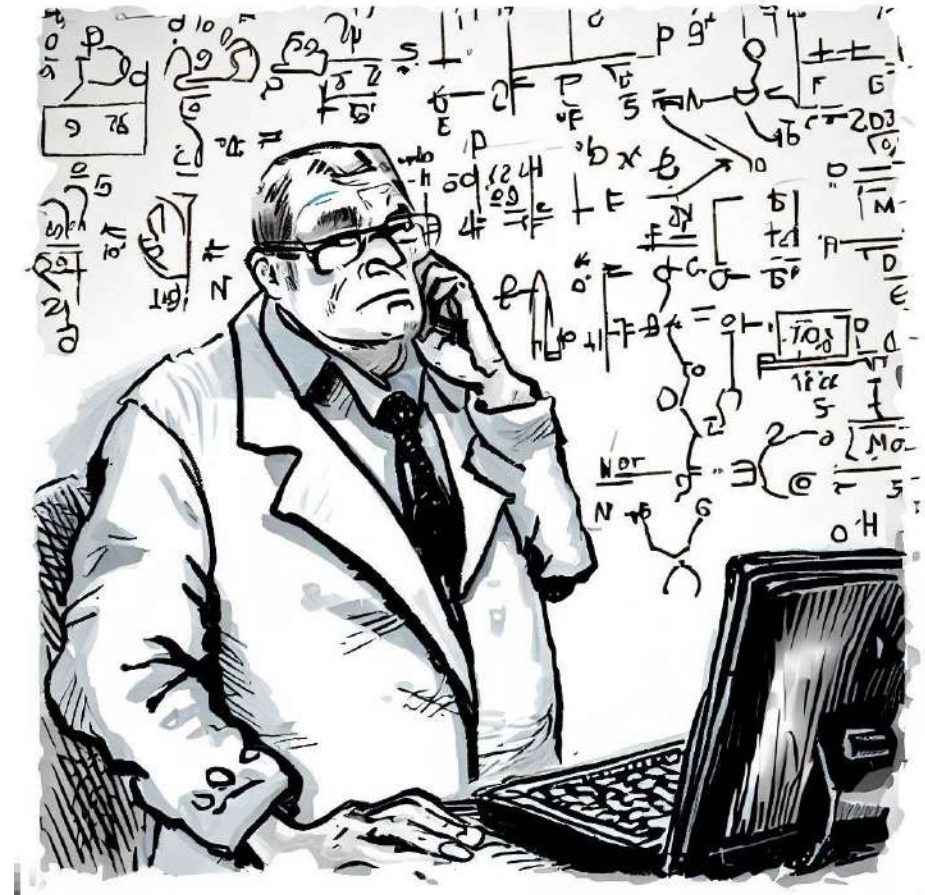


# Жизненный цикл продукта



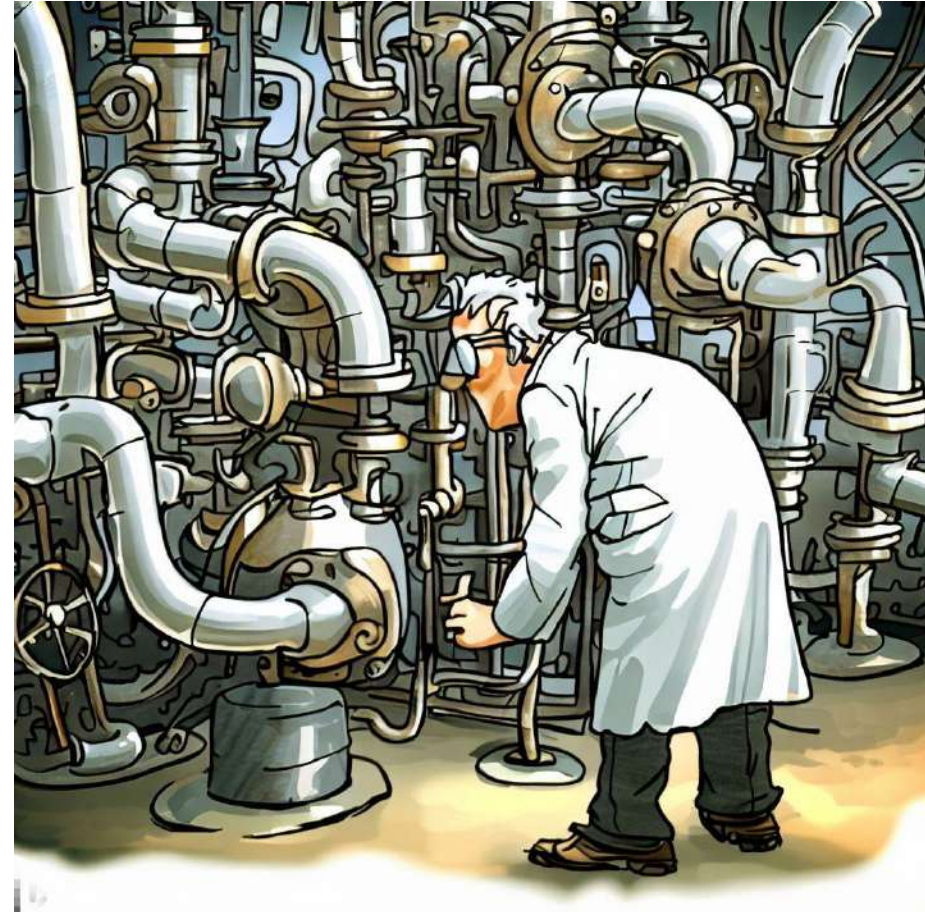
# От исследования к разработке

- Исследователи пишут слишком сложный и слишком нестабильный код.
- Их не волнует, кто будет этот код поддерживать и где взять специалистов нужного профиля (могут писать на Haskell).
- Хорошие исследования обычно документированы.
- Исследования реально увеличивают ценность продукта.
- Требуется специальный человек, который будет следить за совместимостью команд.



# От разработки к исследованиям

- Монолит.
- Плохо документирован.
- Нельзя трогать, а то рассыпется.
- Исследователи могут только работать над отдельным модулем или разрабатывать все с нуля.
- Но без исследований не получится развития.
- Следует закладываться на исследовательский цикл при планировании проекта.



# Выводы

- Разработка не однородна. На разных этапах используются разные инструменты, разные методы и разные типы команд.
- Можно упростить себе жизнь, используя на разных этапах одни и те же инструменты (например из Kotlin стека).
- Но люди все равно разные. Поэтому планировать цикл разработки надо заранее.
- В идеале все фазы цикла должны работать одновременно, но это требует наличия специальных людей, которые организуют взаимодействие.