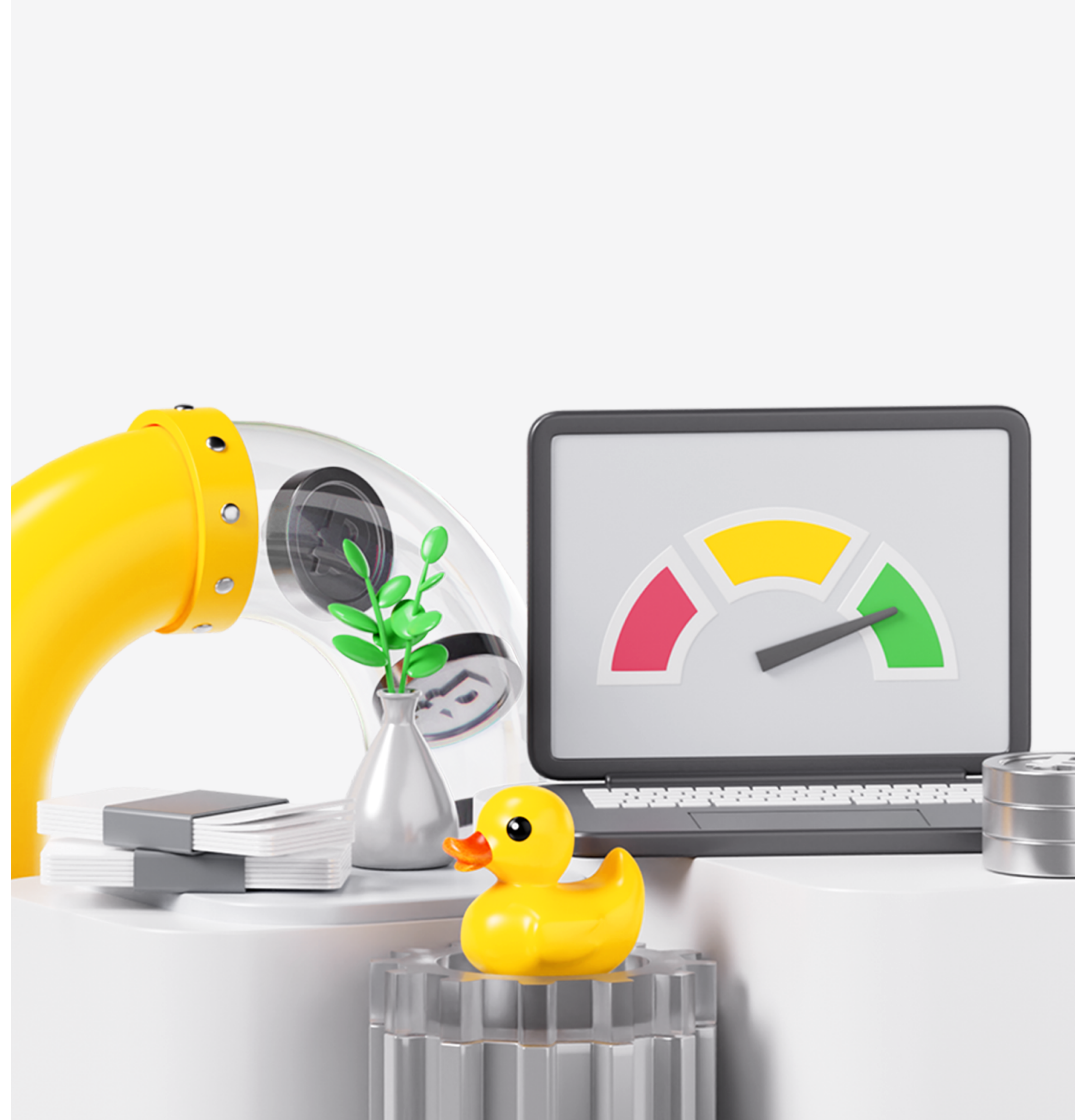




# Инструменты Data Quality: как, зачем, почему. Опыт Т-Банка

Руднев Дмитрий



# Руднев Дмитрий



@dorudnev

- 10 лет опыта разработки инструментов для работы с данными: Data Visualization (Dashboard), Data Preparation & ETL, Data Governance (Data Catalogs & DQ)
- Team Lead and Product Owner
- Руководжу направлением разработки платформы управления данными в Т-Банк

**О чем  
поговорим?!**

# О чем поговорим?!



Поговорим зачем нужны DQ  
инструменты

# О чем поговорим?!



Поговорим зачем нужны DQ  
инструменты



Пройдем путь по выбору подхода  
к внедрению DQ инструментов

# О чем поговорим?!



Поговорим зачем нужны DQ  
инструменты



Пройдем путь по выбору подхода  
к внедрению DQ инструментов



Оценим плюсы и минусы каждого  
из подходов

# О чем поговорим?!



Поговорим зачем нужны DQ инструменты



Пройдем путь по выбору подхода к внедрению DQ инструментов



Оценим плюсы и минусы каждого из подходов



Расскажу опыт разработки и внедрения DQ инструментов в Т-Банке

**Этап 1.**  
**Осознание**



**Качество данных —  
важно**



# Качество данных



6f3643aa	ИВ@НОВ	15	+79991112233
3039c97e	Петров	45	89991112233
3039c97e		-1	+7999111223
0d42f28d	Смирнов	21	+79991112233

# Качество данных

- Субъективно и ситуативно



6f3643aa	Ив@нов	15	+79991112233
3039c97e	Петров	45	89991112233
3039c97e		-1	+7999111223
0d42f28d	Смирнов	21	+79991112233

# Качество данных

- Субъективно и ситуативно
- Делай сейчас —  
результат потом



6f3643aa	Ив@нов	15	+79991112233
3039c97e	Петров	45	89991112233
3039c97e		-1	+7999111223
0d42f28d	Смирнов	21	+79991112233

# Качество данных

- Субъективно и ситуативно
- Делай сейчас — результат потом
- Сложно измерить эффект



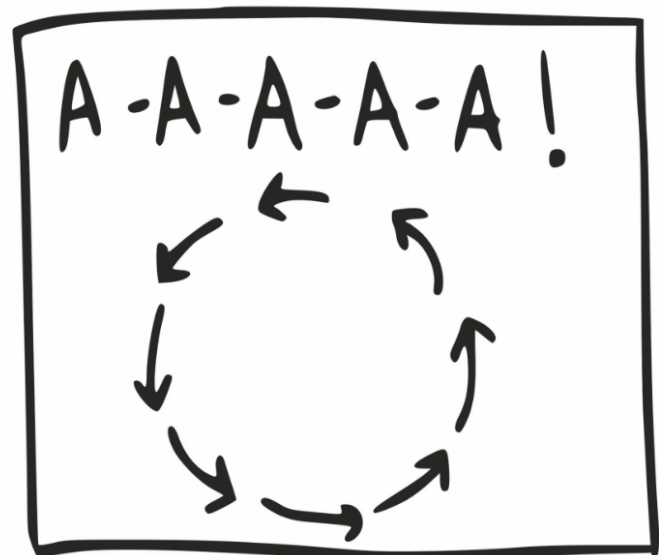
6f3643aa	Ив@нов	15	+79991112233
3039c97e	Петров	45	89991112233
3039c97e		-1	+7999111223
0d42f28d	Смирнов	21	+79991112233



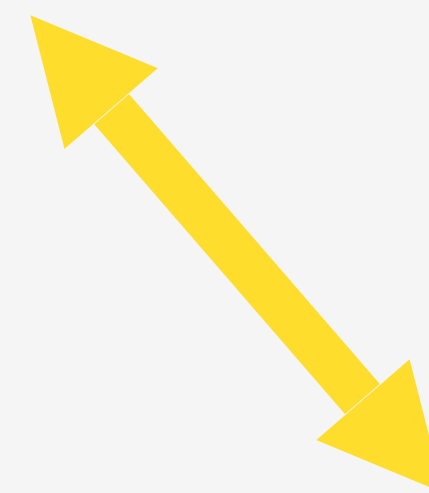
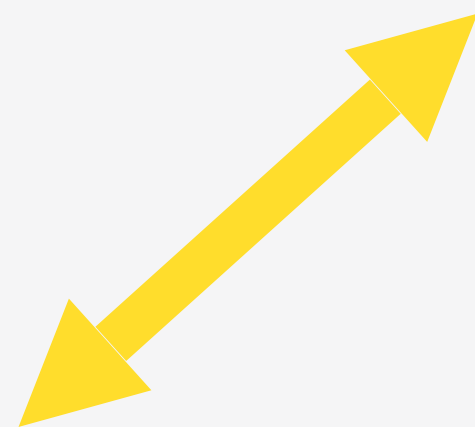




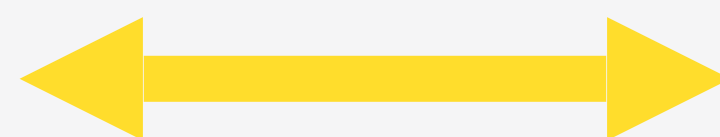
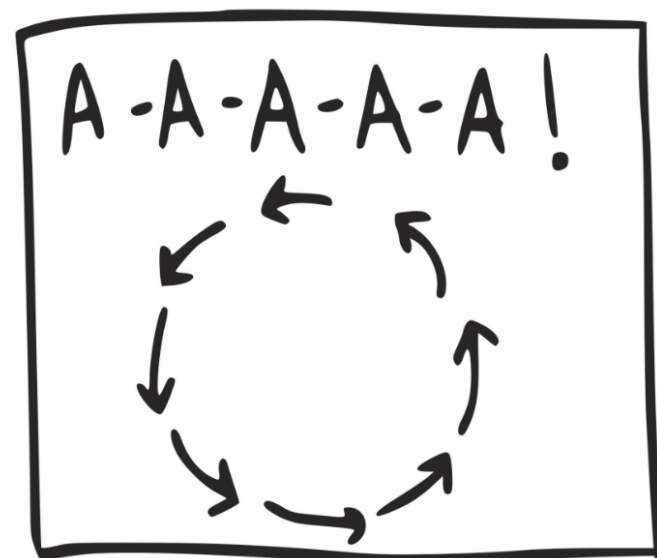
ПЛАН ЭВАКУАЦИИ

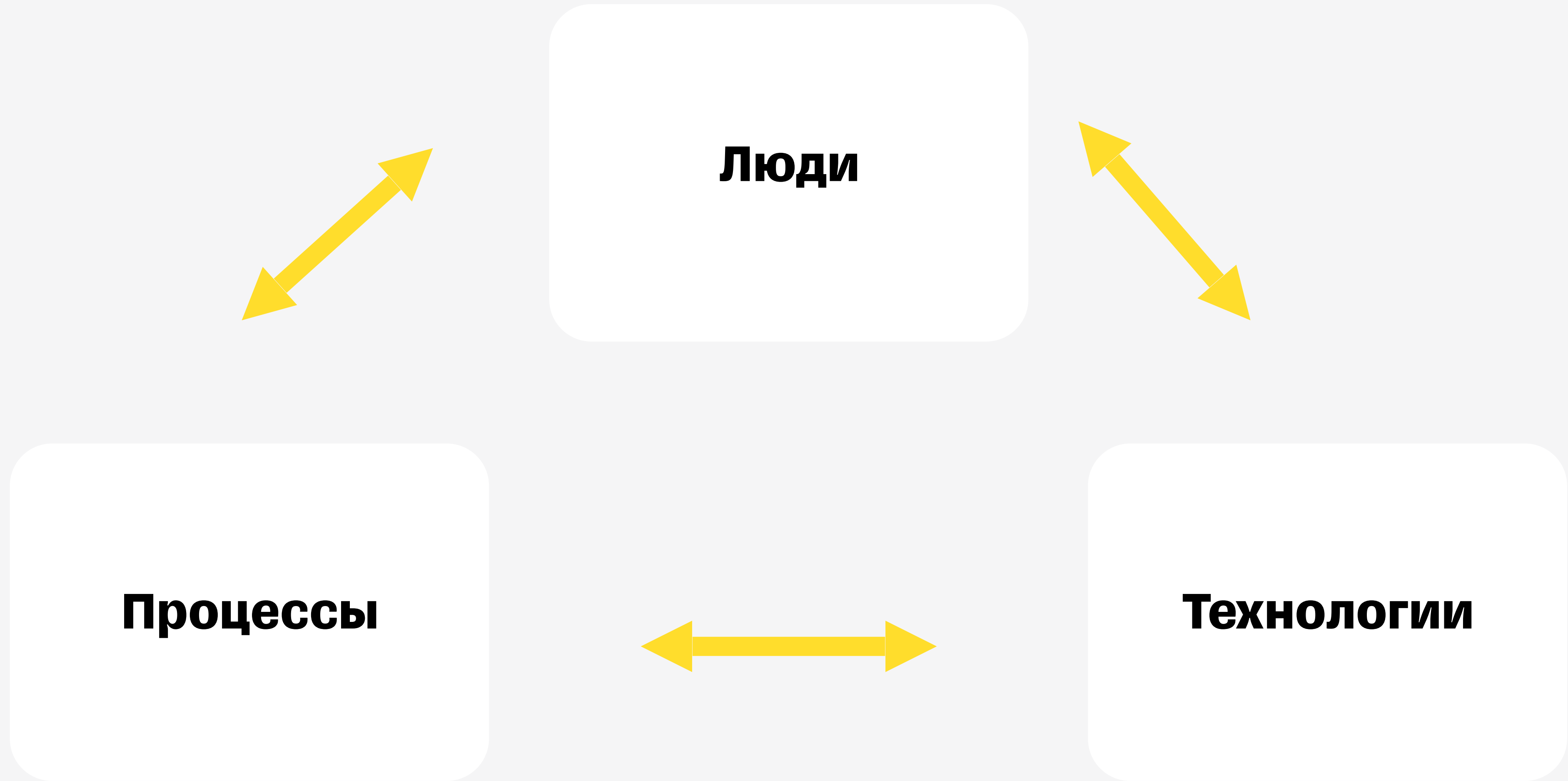






ПЛАН ЭВАКУАЦИИ



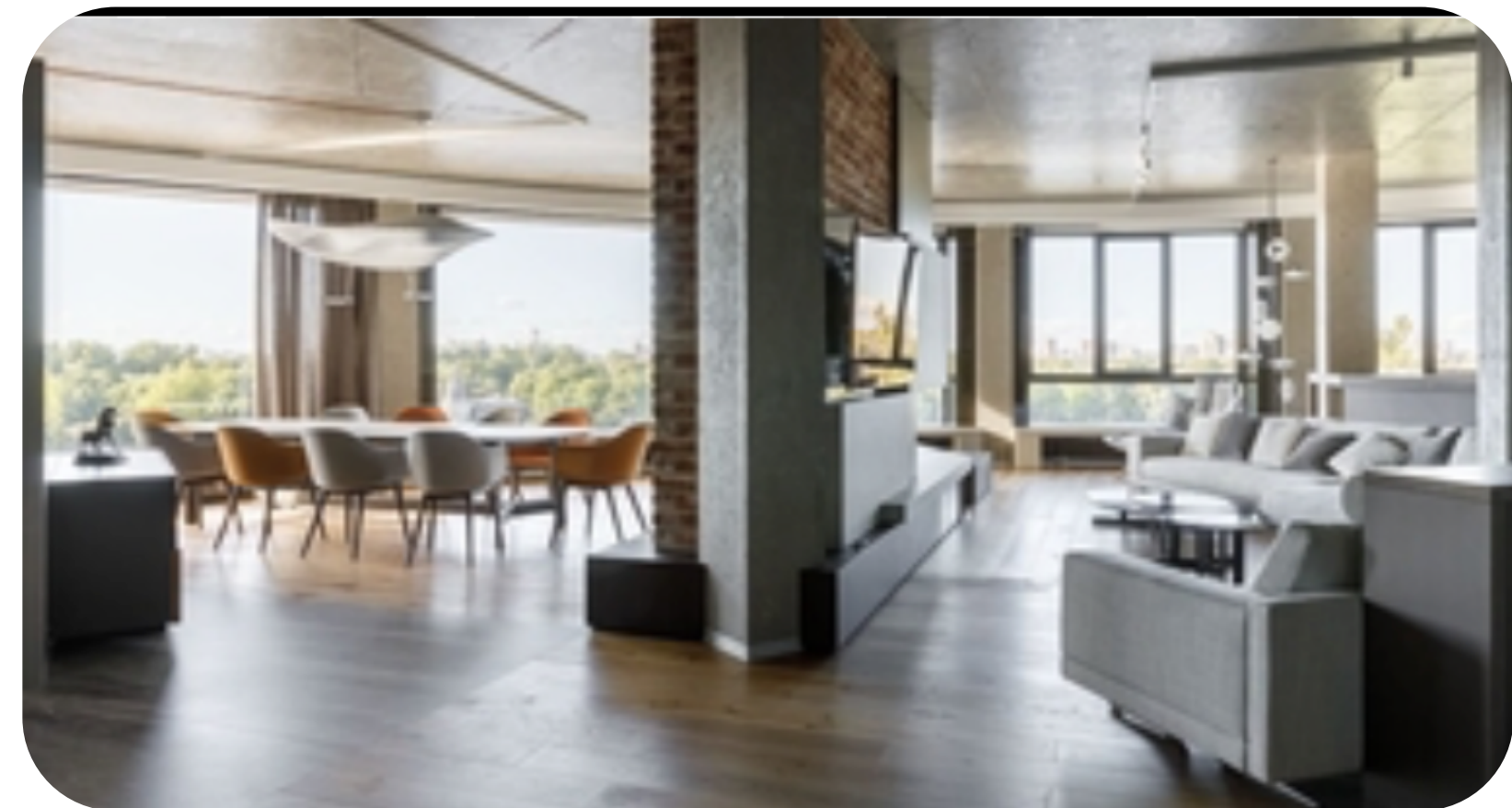


People, Process, Technology Framework

# Осознание особенностей КОМПАНИИ



**VS**



# Осознание особенностей компании

# Осознание особенностей компании

- Размер и структура компании

# Осознание особенностей компании

- Размер и структура компании
- Уровень зрелости процессов

# Осознание особенностей компании

- Размер и структура компании
- Уровень зрелости процессов
- Уровень компетенции людей

# Осознание особенностей компании

- Размер и структура компании
- Уровень зрелости процессов
- Уровень компетенции людей
- Уровень критичности процессов на данных



# Осознание особенностей компании

- Размер и структура компании
- Уровень зрелости процессов
- Уровень компетенции людей
- Уровень критичности процессов на данных
- IT ландшафт

# Особенности Data Platform Т-Банк

# Особенности Data Platform Т-Банк

## Единая Data Platform

- MAU 16k+, 170k+ запросов к GP ежедневно, 15+ кластеров GP

# Особенности Data Platform Т-Банк

## Единая Data Platform

- MAU 16k+, 170k+ запросов к GP ежедневно, 15+ кластеров GP

## Много бизнес доменов

- 27 крупных групп пользователей
- Разный уровень зрелости, разные потребности в качестве данных

# Особенности Data Platform Т-Банк

## Единая Data Platform

- MAU 16k+, 170k+ запросов к GP ежедневно, 15+ кластеров GP

## Много бизнес доменов

- 27 крупных групп пользователей
- Разный уровень зрелости, разные потребности в качестве данных

## “Динамичный” технологический ландшафт

# Выводы первого этапа

# Выводы первого этапа

- Качество данных – субъективно.

# Выводы первого этапа

- Качество данных – субъективно.
- Помимо инструментов, нужны люди и процессы.



# Выводы первого этапа

- Качество данных – субъективно.
- Помимо инструментов, нужны люди и процессы.
- Нужно понимать особенности компании.

# Выводы первого этапа

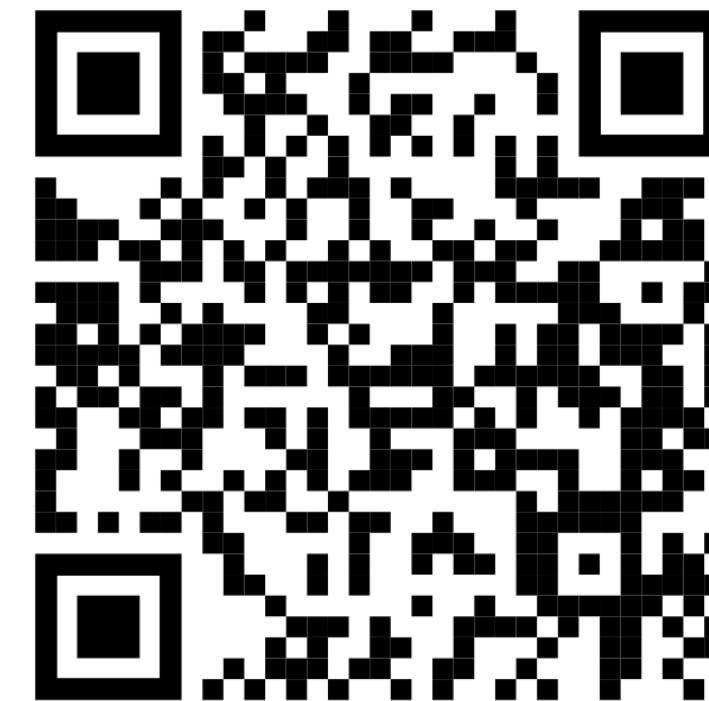
- Качество данных – субъективно.
- Помимо инструментов, нужны люди и процессы.
- Нужно понимать особенности компании.
- Нет универсального решения.

**Этап 2.**

**Выбор типа инструмента**

# Функциональные возможности DQ инструментов

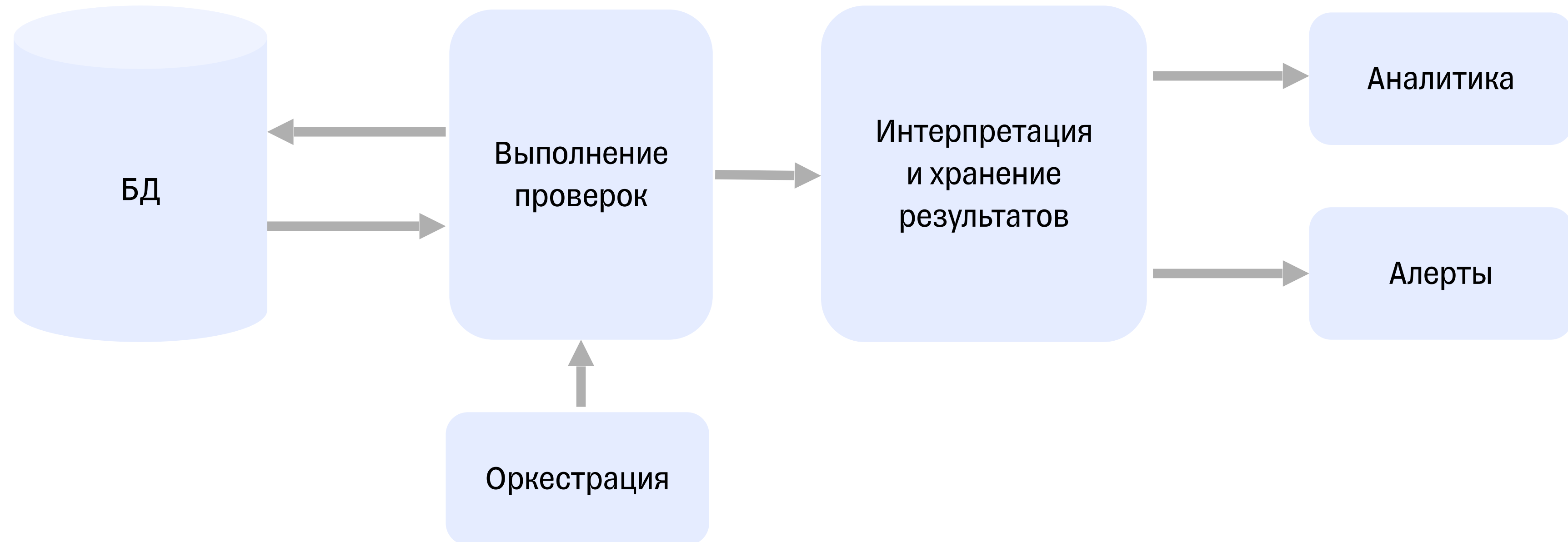
- Мониторинг (Observability)
- Профилирование данных (Profiling)
- Подготовка данных (Parsing, standardizing and cleansing)
- Инструменты фикса данных (Data Fixing)
- Поддержка DataOps (CI/CD)
- ... еще 12 возможностей по версии Gartner



[Magic Quadrant for Data Quality Solutions 2022.](#)

# Инструменты мониторинга данных

Узнавать о проблемах с данными раньше чем потребители



**Этап 3.**  
**Выбор архитектуры**

# Структура DQ инструмента

# Структура DQ инструмента

**Выполнение  
проверок**

- оркестрация
- коннект к данным
- среда выполнения запросов



# Структура DQ инструмента

## Выполнение проверок

- оркестрация
- коннект к данным
- среда выполнения запросов

## Логика проверок

- технические проверки
- бизнесовые проверки

# Структура DQ инструмента

## Выполнение проверок

- оркестрация
- коннект к данным
- среда выполнения запросов

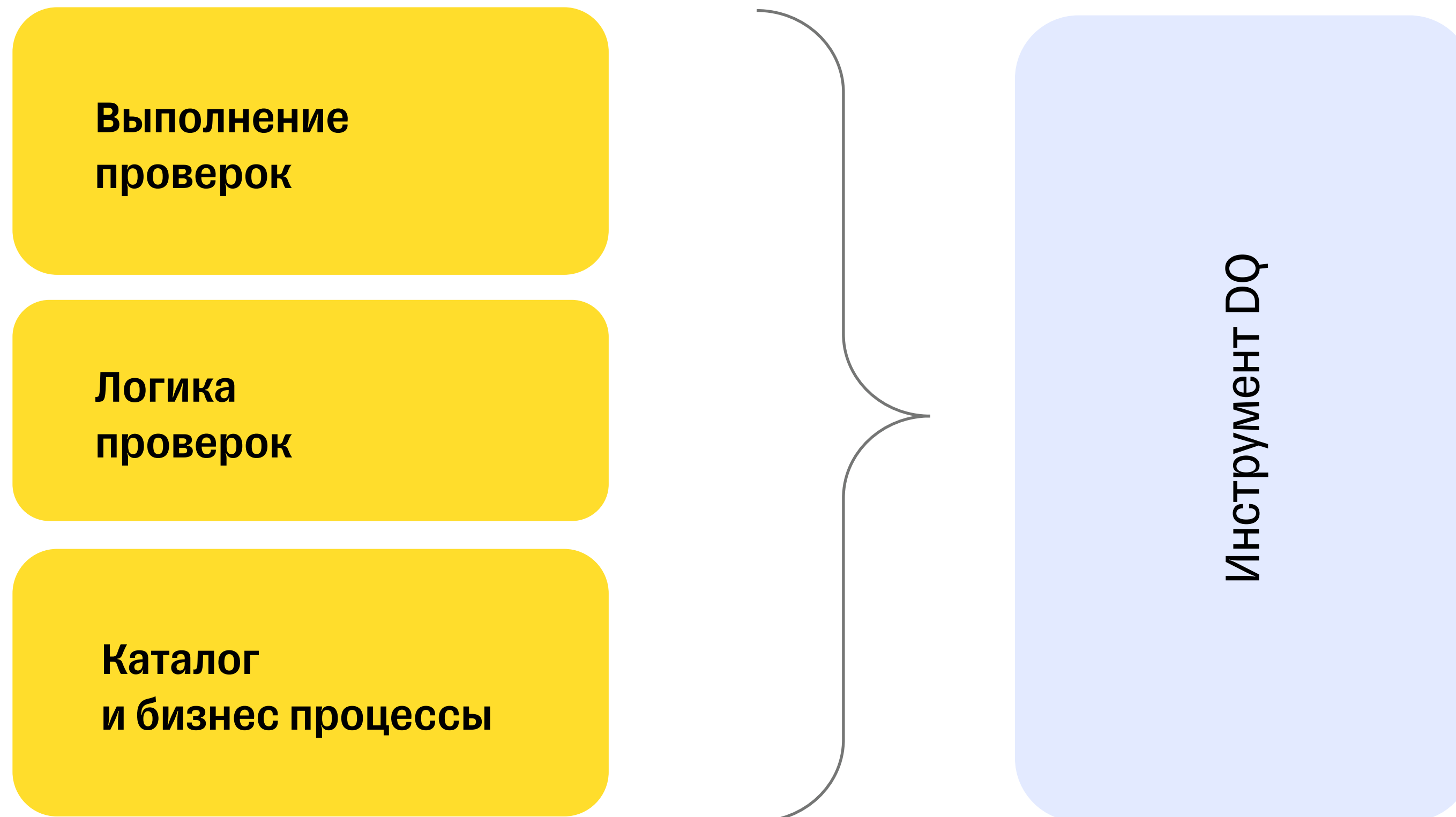
## Логика проверок

- технические проверки
- бизнесовые проверки

## Каталог и бизнес процессы

- хранение результатов
- аналитика
- нотификации, владение

# Архитектура “все-в-одном”



# Архитектура «все-в-одном»

- Informatica Data Quality (IDQ)
- SAP Data Services
- Ataccama ONE
- Oracle Enterprise Data Quality
- Microsoft Data Quality Services
- SAS Data Quality
- DQLabs Platform
- ... И Т.Д.



Gartner Data Quality Solutions Reviews

# **Архитектура «все-в-одном» — наш опыт**

# Архитектура «все-в-одном» — наш опыт

- Web приложение на Scala

# Архитектура «все-в-одном» — наш опыт

- Web приложение на Scala
- Настройка проверок, запуска и нотификации через UI

# Архитектура «все-в-одном» — наш опыт

- Web приложение на Scala
- Настройка проверок, запуска и нотификации через UI
- Собственный синтаксис написания проверок КД



# Архитектура «все-в-одном» — наш опыт

- Web приложение на Scala
- Настройка проверок, запуска и нотификации через UI
- Собственный синтаксис написания проверок КД
- Собственный коннект к данным

# Архитектура «все-в-одном» — наш опыт

- Web приложение на Scala
- Настройка проверок, запуска и нотификации через UI
- Собственный синтаксис написания проверок КД
- Собственный коннект к данным
- Оркестрация запусков внутри приложения

# Архитектура «все-в-одном» — наш опыт

- Web приложение на Scala
- Настройка проверок, запуска и нотификации через UI
- Собственный синтаксис написания проверок КД
- Собственный коннект к данным
- Оркестрация запусков внутри приложения
- Спустя время: “внешние проверки”

# Архитектура «все-в-одном» — наш опыт

- Web приложение на Scala
- Настройка проверок, запуска и нотификации через UI
- Собственный синтаксис написания проверок КД
- Собственный коннект к данным
- Оркестрация запусков внутри приложения
- Спустя время: “внешние проверки”

# **Архитектура «все-в-одном» — наш опыт**

# Архитектура «все-в-одном» — наш опыт

- Противоречивые требования от пользователей

# Архитектура «все-в-одном» — наш опыт

- Противоречивые требования от пользователей
- Дорого “догонять” динамичную Data Platform

# Архитектура «все-в-одном» — наш опыт

- Противоречивые требования от пользователей
- Дорого “догнать” динамичную Data Platform
- Ограниченный ресурс разработки



# Архитектура «все-в-одном» — выводы

# Архитектура «все-в-одном» — ВЫВОДЫ



Удобна для не технического пользователя

# Архитектура «все-в-одном» — выводы



Удобна для не технического пользователя



Хороша, когда ты ее покупаешь как коробку вместе с другими инструментами

# Архитектура «все-в-одном» — выводы



Удобна для не технического пользователя



Хороша, когда ты ее покупаешь как коробку вместе с другими инструментами

Дорого и долго разрабатывать самим

# **Этап 3. Выбор архитектуры (заход второй)**

# Структура DQ инструмента

## Выполнение проверок

- оркестрация
- коннект к данным
- среда выполнения запросов

## Логика проверок

- технические проверки
- бизнесовые проверки

## Каталог и бизнес процессы

- хранение результатов
- аналитика
- нотификации, владение

**Выполнение  
проверок**

**Логика  
проверок**

**Каталог  
и бизнес  
процессы**

**Выполнение  
проверок**



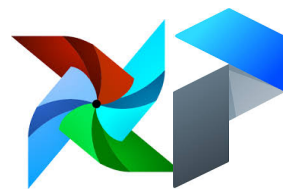
**Standalone  
orchestration**

**Логика  
проверок**

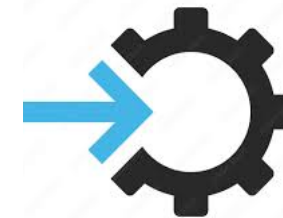
**Каталог  
и бизнес  
процессы**



**Выполнение  
проверок**



**Standalone  
orchestration**

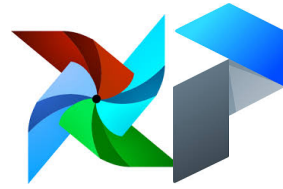


**Integrate in  
ETL tool**

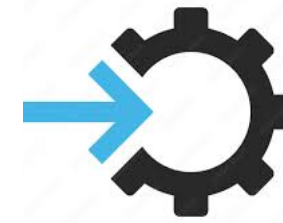
**Логика  
проверок**

**Каталог  
и бизнес  
процессы**

**Выполнение  
проверок**



Standalone  
orchestration



Integrate in  
ETL tool

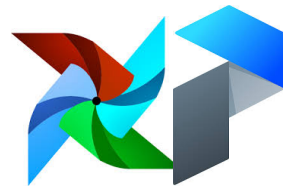
**Логика  
проверок**



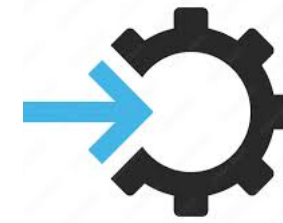
raw/functions

**Каталог  
и бизнес  
процессы**

**Выполнение проверок**



Standalone  
orchestration



Integrate in  
ETL tool

**Логика проверок**



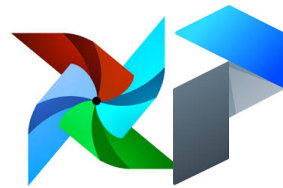
raw/functions



raw/in-house library

**Каталог  
и бизнес  
процессы**

**Выполнение проверок**



Standalone orchestration



Integrate in ETL tool

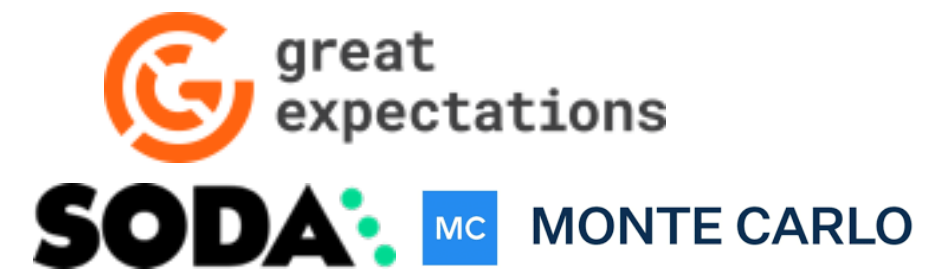
**Логика проверок**



raw/functions



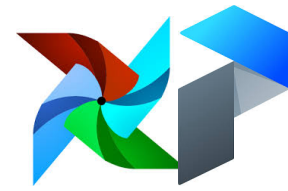
raw/in-house library



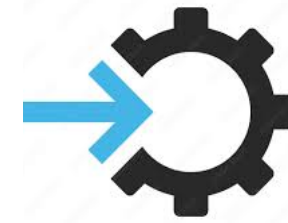
open source library

**Каталог и бизнес процессы**

**Выполнение проверок**



Standalone orchestration



Integrate in ETL tool

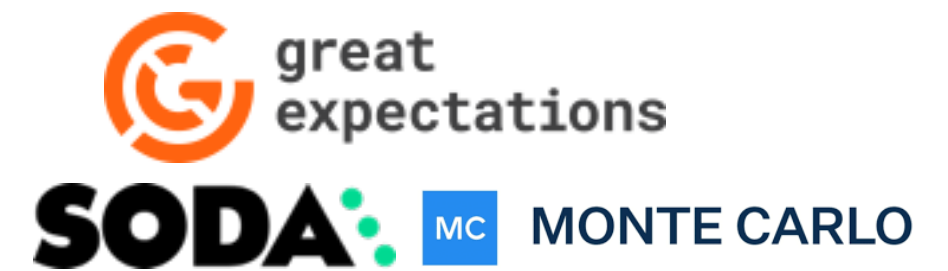
**Логика проверок**



raw/functions



raw/in-house library



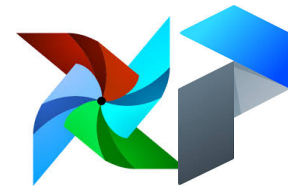
open source library

**Каталог и бизнес процессы**

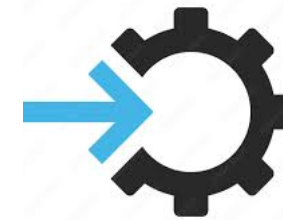


DB table

**Выполнение проверок**



Standalone orchestration

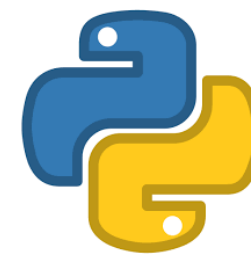


Integrate in ETL tool

**Логика проверок**



raw/functions



raw/in-house library



open source library

**Каталог и бизнес процессы**

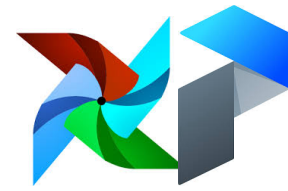


DB table

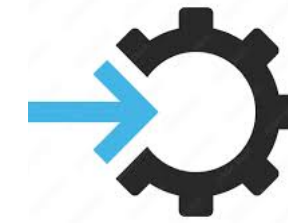


observability tools

**Выполнение проверок**



Standalone orchestration



Integrate in ETL tool

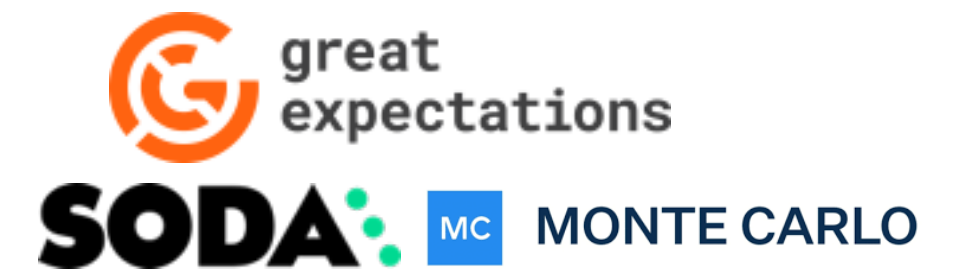
**Логика проверок**



raw/functions



raw/in-house library



open source library

**Каталог и бизнес процессы**



DB table

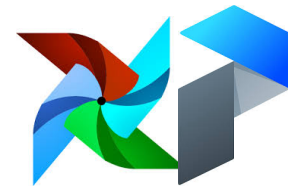


observability tools

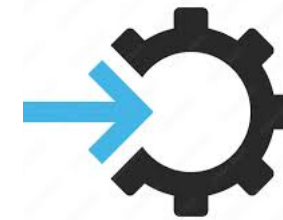


data catalogs

**Выполнение проверок**



Standalone orchestration

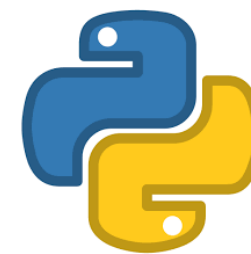


Integrate in ETL tool

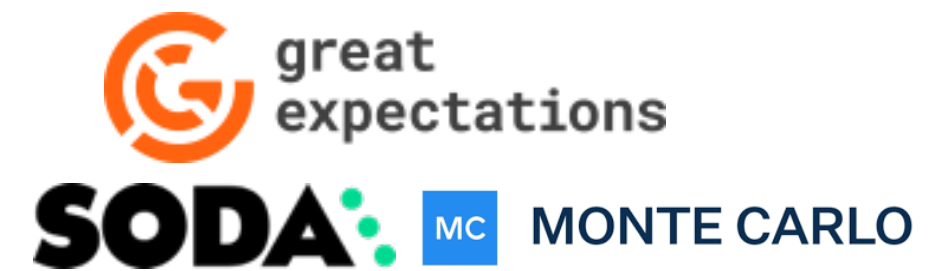
**Логика проверок**



raw/functions



raw/in-house library



open source library

**Каталог и бизнес процессы**



DB table



observability tools



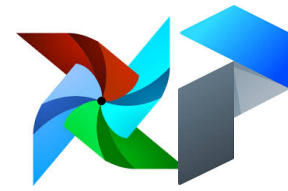
data catalogs



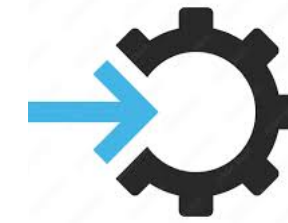
cloud



**Выполнение проверок**



Standalone orchestration



Integrate in ETL tool

**Логика проверок**



raw/functions



raw/in-house library



open source library

**Каталог и бизнес процессы**



DB table



observability tools



data catalogs



cloud



in-house

**Выполнение  
проверок**



Helicopter  
(Notebooks)

**Логика  
проверок**



raw/functions



raw/in-house library

**Каталог  
и бизнес  
процессы**



DB table

**Выполнение проверок**



Helicopter  
(Notebooks)

**Логика проверок**



raw/functions



raw/in-house library

**Каталог и бизнес процессы**



DB table



Каталог проверок

# Каталог проверок

# Каталог проверок

- Быстрый поиск информации о качестве данных

# Каталог проверок

- Быстрый поиск информации о качестве данных
- Возможность переиспользовать созданные проверки

# Каталог проверок

- Быстрый поиск информации о качестве данных
- Возможность переиспользовать созданные проверки
- Быстрое создание проверок качества данных

# Каталог проверок

- Быстрый поиск информации о качестве данных
- Возможность переиспользовать созданные проверки
- Быстрое создание проверок качества данных
- Нотификации “из коробки”



# Каталог проверок

- Быстрый поиск информации о качестве данных
- Возможность переиспользовать созданные проверки
- Быстрое создание проверок качества данных
- Нотификации “из коробки”
- Аналитика по качеству данных

# Каталог проверок

- Быстрый поиск информации о качестве данных
- Возможность переиспользовать созданные проверки
- Быстрое создание проверок качества данных
- Нотификации “из коробки”
- Аналитика по качеству данных
- Построение бизнес процессов по Data Quality и Data Governance

# Каталог проверок – пишем свое

# Каталог проверок – пишем свое

Облачные решения – недоступны

# Каталог проверок – пишем свое

Облачные решения – недоступны

Observability инструменты – не хватило специальных фичей

# Каталог проверок – пишем свое

Облачные решения – недоступны

Observability инструменты – не хватило специальных фичей


Open source – только в составе дата каталогов

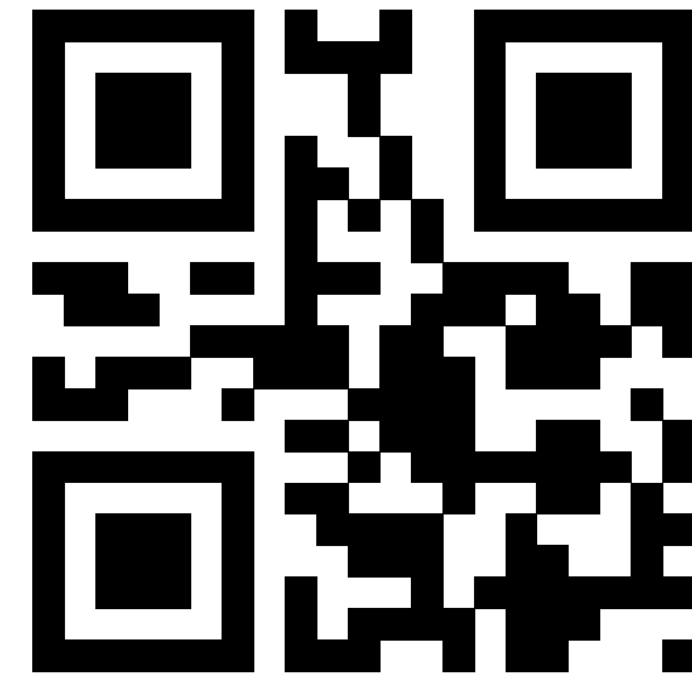
# Каталог проверок – пишем свое

Облачные решения – недоступны

Observability инструменты – не хватило специальных фичей

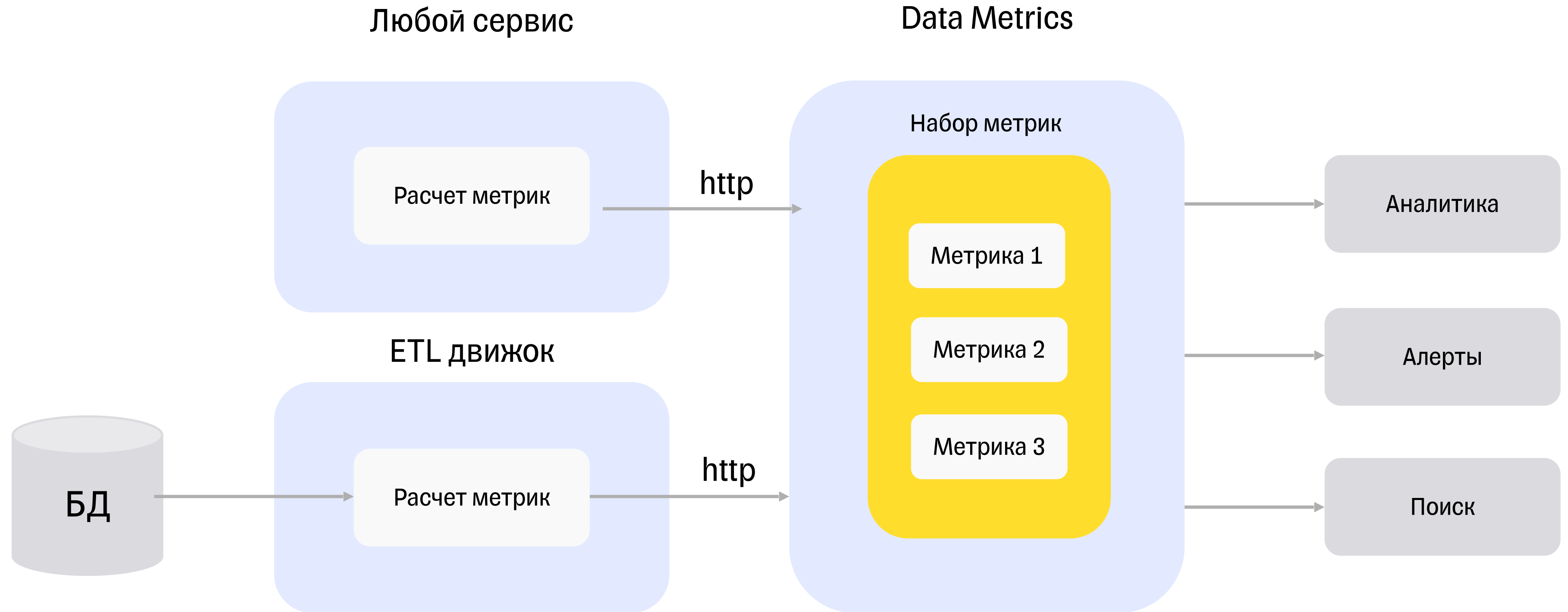
Open source – только в составе дата каталогов

 DQ Ops – open source каталог (релиз v1 – январь 2024)



DQOps

# Data Metrics – каталог проверок





# Data Metrics

## Хранилище метрик качества данных

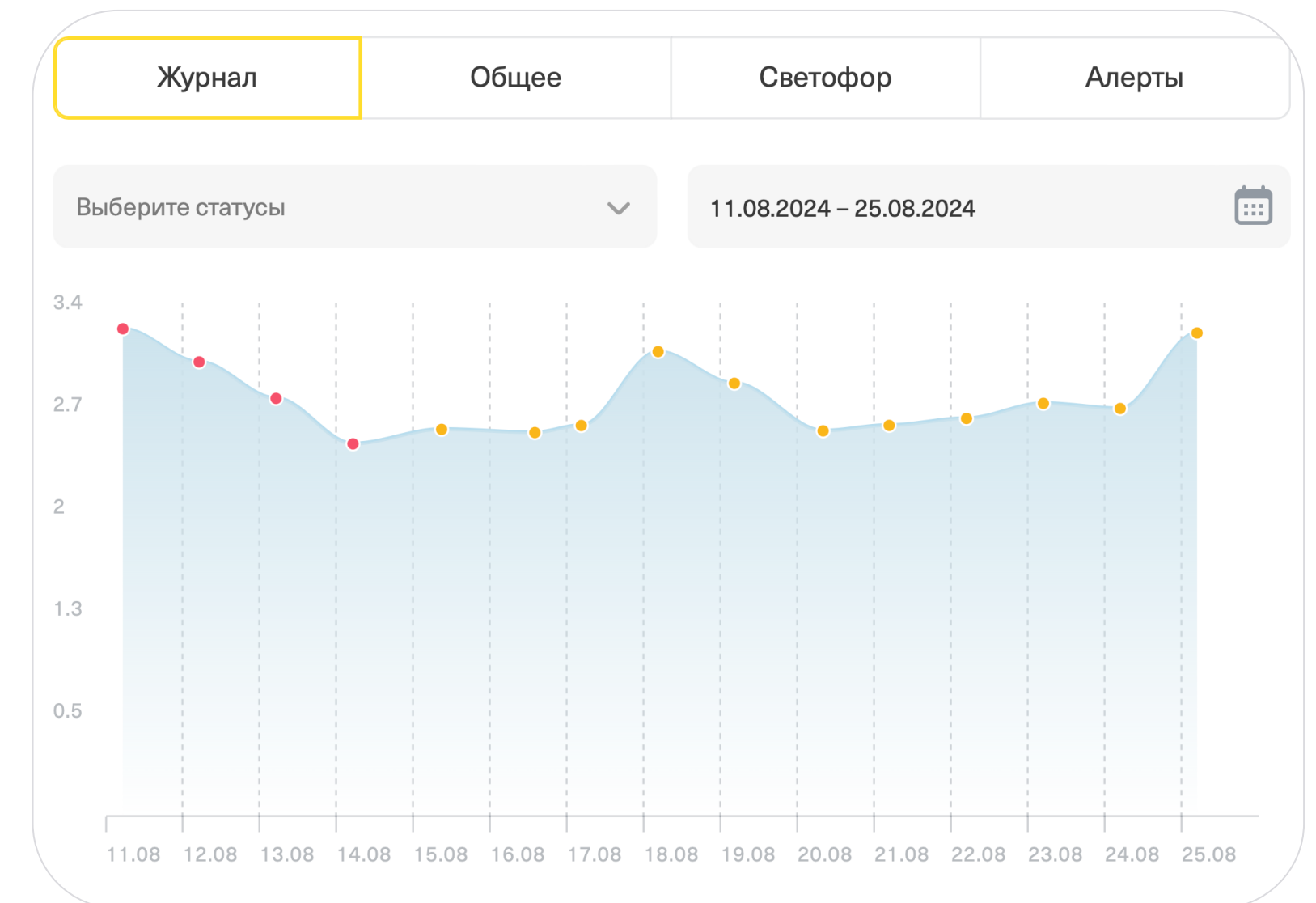
DataChecker. Быстрый старт ⚠ Неудовлетворительное

Журнал Основное Метрики Источники Алерты Светофор

Поиск по названию набора и ID Выберите статусы Период

Результатов: 19

Метрика	Название метрики	Значение метрики	Значение светофора
5 апреля 2024 15:20:51 ⚠			
<a href="#">usr_healer_financial_account_account_rk_duplicate_count</a>	democases.usr_healer_financial_account_ac...	2	⚠ Неудовлетворительное
<a href="#">usr_healer_financial_account_account_rk_missing_count</a>	democases.usr_healer_financial_account_ac...	0	✅ Отличное
<a href="#">usr_healer_financial_account_appl_rej_reason_real_cd_missing_count</a>	democases.usr_healer_financial_account_ap...	4	⚠ Неудовлетворительное
<a href="#">usr_healer_financial_account_pd_range_invalid_count_10bef5f2</a>	democases.usr_healer_financial_account_pd...	3	✅ Отличное
5 апреля 2024 15:13:08 ❌			
<a href="#">usr_healer_financial_account_account_rk_duplicate_count</a>	democases.usr_healer_financial_account_ac...	2	⚠ Неудовлетворительное
<a href="#">usr_healer_financial_account_account_rk_missing_count</a>	democases.usr_healer_financial_account_ac...	2	❌ Плохое
<a href="#">usr_healer_financial_account_appl_rej_reason_real_cd_missing_count</a>	democases.usr_healer_financial_account_ap...	2	⚠ Неудовлетворительное
<a href="#">usr_healer_financial_account_pd_range_invalid_count_10bef5f2</a>	democases.usr_healer_financial_account_pd...	2	✅ Отличное



# Data Metrics

## Группировка проверок по пространствам и наборам метрик Поиск проверок

**Пространства**

Поиск по названию и ID пространства Выберите теги

Только мои Результатов: 119

<b>Chatbot</b> chatbot Наборы метрик: 1	<b>coin</b> coin Наборы метрик: 0	<b>Compliance Core</b> compliance_core Наборы метрик: 1
<b>CreditRegistry</b> CRE Наборы метрик: 1	<b>CROSSDATA</b> crossdata Наборы метрик: 39	<b>CROSSDATA Recruitment</b> CROSSDATA_Recruitment Наборы метрик: 1
<b>Daniil_test</b> test_metrics_daniil Наборы метрик: 1	<b>Data Mart</b> DTMRT Наборы метрик: 1	<b>Data Quality Checks</b> Data_Quality_Checks Наборы метрик: 7

**Демонстрационные кейсы**

Наборы Общее

Поиск по названию набора и ID Значение светофора Период

Только мои Результатов: 8

Заголовок набора метрик	Название	Значение светофора	Последний запуск	Теги
<a href="#">Democase4. Качество данных бизнес ...</a> 3	democases.my_business_process	⚠ Неудовлетворительное	2024.04.05 18:18:18+03:00	
<a href="#">Democase2. Качество данных по объе...</a> 4	democases.product_scoring_common	✅ Отличное	2023.05.10 21:33:39+03:00	
<a href="#">Democase1. Качество данных по объе...</a> 2	democases.financial_account_common	⚠ Неудовлетворительное	2024.04.05 18:18:18+03:00	democase
<a href="#">Democase3. Качество данных по объе...</a> 4	democases.united_call_common	⚠ Неудовлетворительное	2023.05.10 21:40:48+03:00	
<a href="#">Democase5. Datachecker числовые м...</a> 0	democases.datachecker_examples_numeric_0	⊖ Нет данных	-	
<a href="#">DataChecker. Быстрый старт</a> 4	democases.datachecker_getting_started	⚠ Неудовлетворительное	2024.04.05 18:20:51+03:00	
<a href="#">Data Checker - Failed Row Samples</a> 1	democases.failedrowsamples	❌ Плохое	2023.09.29 12:32:18+03:00	
<a href="#">Democases. Confidence</a> 3	democases.confidence	✅ Отличное	2023.12.22 17:13:33+03:00	

# Data Metrics

## Метаданные о проверках КД

Democase1. Качество данных по объекту financial\_account ⚠ Неудовлетворительное

Журнал Основное Метрики Источники Алерты Светофор

Заголовок набора метрик

Democase1. Качество данных по объекту financial\_account

Название\*

democases.financial\_account\_common

Владелец

d.o.rudnev

[Назначить меня](#)

Бизнес-процесс

DM

Теги

democase X

Описание набора

Введите описание набора

 usr\_healer\_financial\_account\_account\_rk\_duplicate\_count

Создайте метрику, задайте уровень качества значений и настройте нотификации

Название метрики\*

usr\_healer\_financial\_account\_account\_rk\_duplicate\_count

Тип значений метрики\*

Числовой

Кластер\*

gp

Схема и таблица\*

usr\_healer.financial\_account

Атрибуты проверки

Введите атрибуты

Проверяемые характеристики

Точность и достоверность X

Теги

|

# Data Metrics

## Интерпретация значений метрик (светофоры)

### Helicopter

Untitled 1

Source: gp

```
1 select
2     -- Неотрицательность полей
3     count(*) filter(where end_dttm <= start_dttm) as end_dttm_corrent,
4     -- null-ы начало\окончание звонка
5     count(*) filter(where united_call_end_dttm is null) as call_end_null_count
6 from usr_healer.united_call;
```

Text Table-1

#	end_dttm_corrent	call_end_null_count
1	2	0

### Data Metrics

usr\_healer\_financial\_account\_pd\_range\_invalid\_count\_10bef5f2

Создайте метрику, задайте уровень качества значений и настройте нотификации

Журнал    Общее    Светофор    Алерты

Использовать светофор

Способ вычисления

Конструктор        Формула        Инпут   

Числовой тип значения метрики    Значение светофора

≤	5	+ -	✔ Отличное
≤	10	+ -	⚠ Неудовлетворительное
>	10	+ -	✖ Плохое

# Data Metrics

## Оповещения

[← Назад](#)

**ABM Metrics** ✖ Плохое

Журнал   Основное   Метрики   Источники   **Алерты**   Светофор

Здесь собраны оповещения на заданные значения светофора набора

Отличное

Заголовок алерта\*

Проверка корректности текущих связей распределения в модели

Получатели\*

@v.sim ✕ ?

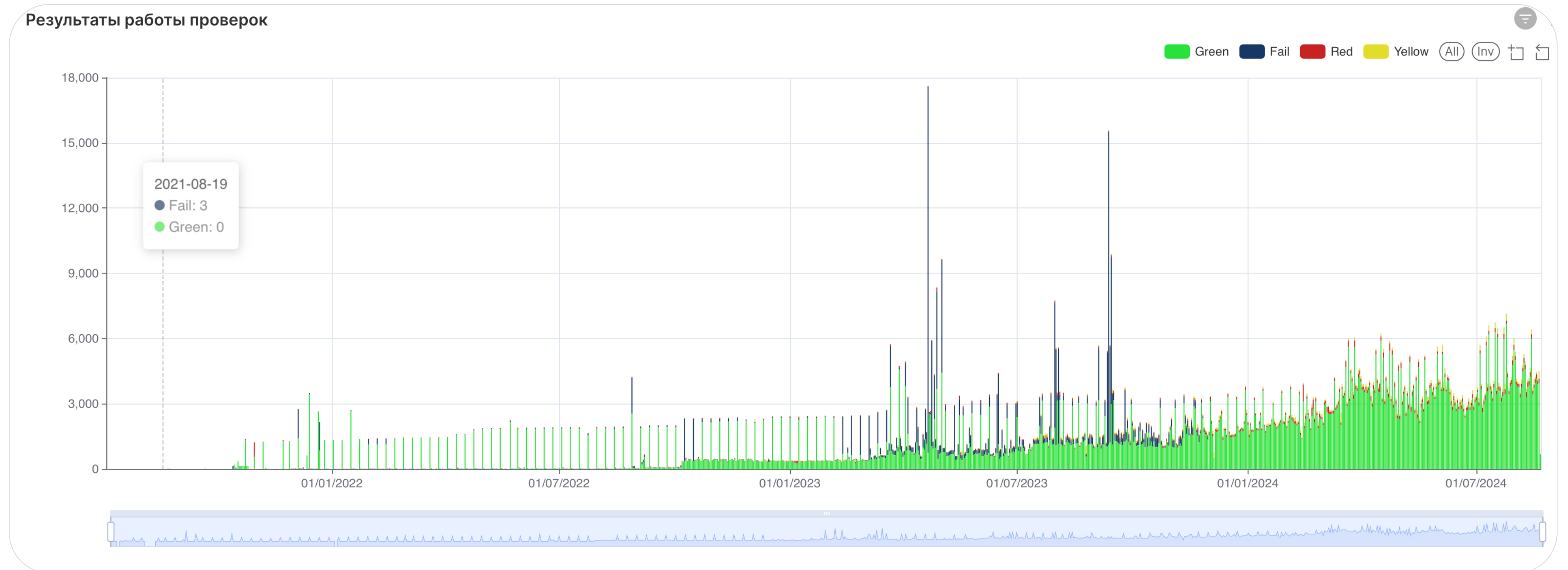
Подписать меня на алерт

Текст алерта\*

Проверка не выявила проблемных точек!

# Data Metrics

## Аналитика по проверкам качества данных



**Выполнение  
проверок**



Helicopter  
(Notebooks)

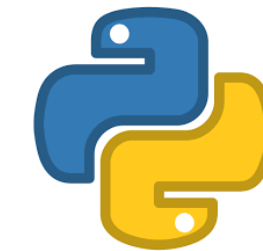


Tedi  
(ETL)

**Логика  
проверок**



raw/functions



raw/in-house library

**Каталог  
и бизнес  
процессы**



(Data Metrics)  
Каталог проверок

**Выполнение проверок**



Helicopter  
(Notebooks)



Tedi  
(ETL)

**Логика проверок**



raw/functions



raw/in-house library



Библиотека DQ  
проверок

**Каталог  
и бизнес  
процессы**



(Data Metrics)  
Каталог проверок



# Библиотека DQ проверок

# Библиотека DQ проверок

- Снижение порога входа в DQ

# Библиотека DQ проверок

- Снижение порога входа в DQ
- Уменьшение времени на разработку проверок

# Библиотека DQ проверок

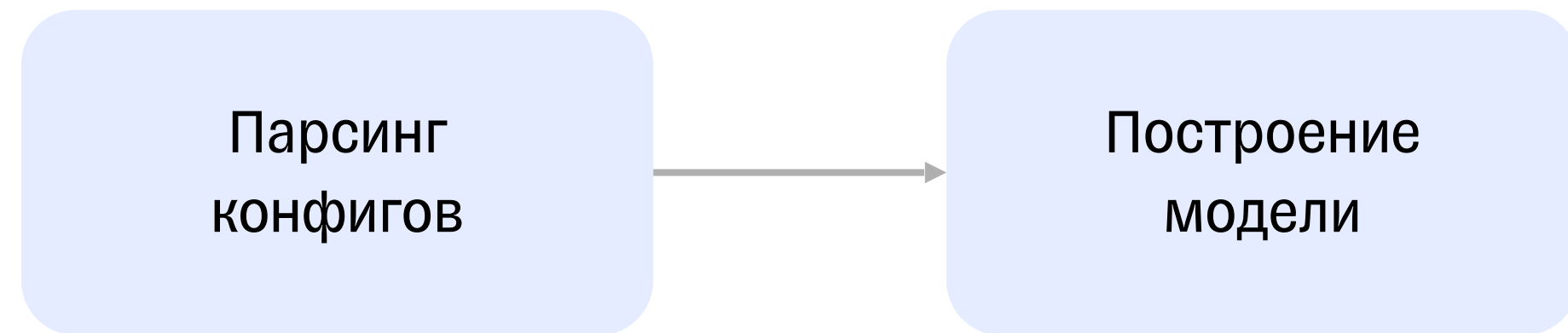
- Снижение порога входа в DQ
- Уменьшение времени на разработку проверок
- Оптимальные запросы

# Библиотека DQ проверок

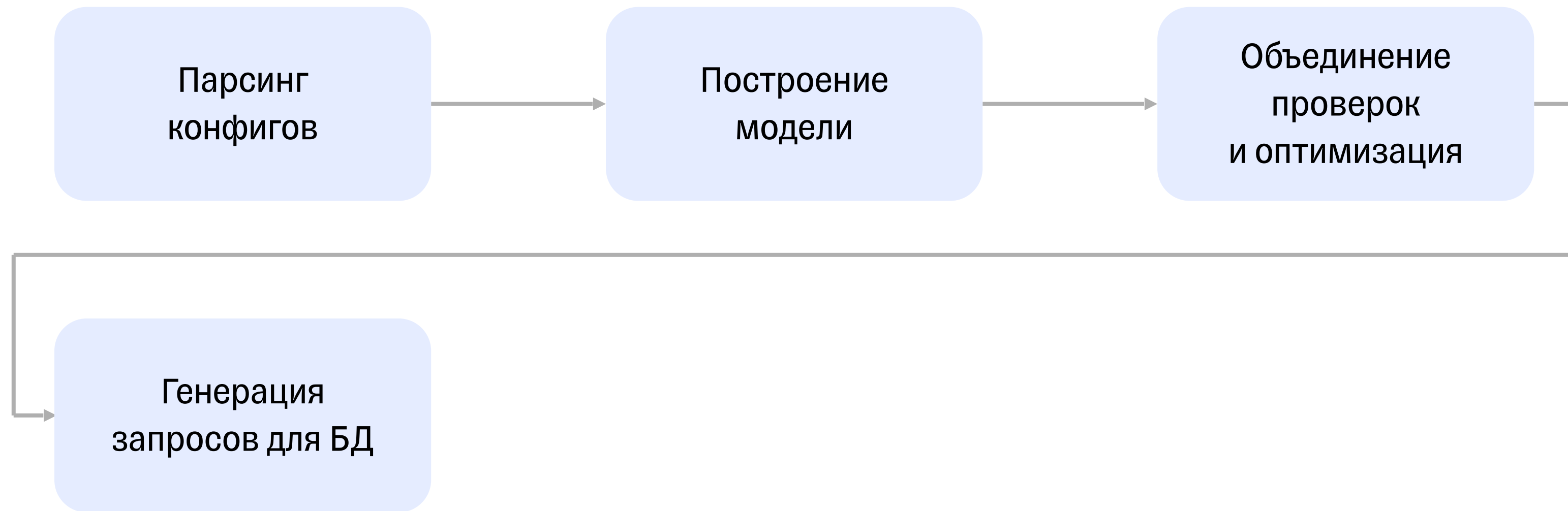
- Снижение порога входа в DQ
- Уменьшение времени на разработку проверок
- Оптимальные запросы
- Единый синтаксис под разные БД

# Архитектура библиотека DQ проверок

# Архитектура библиотека DQ проверок

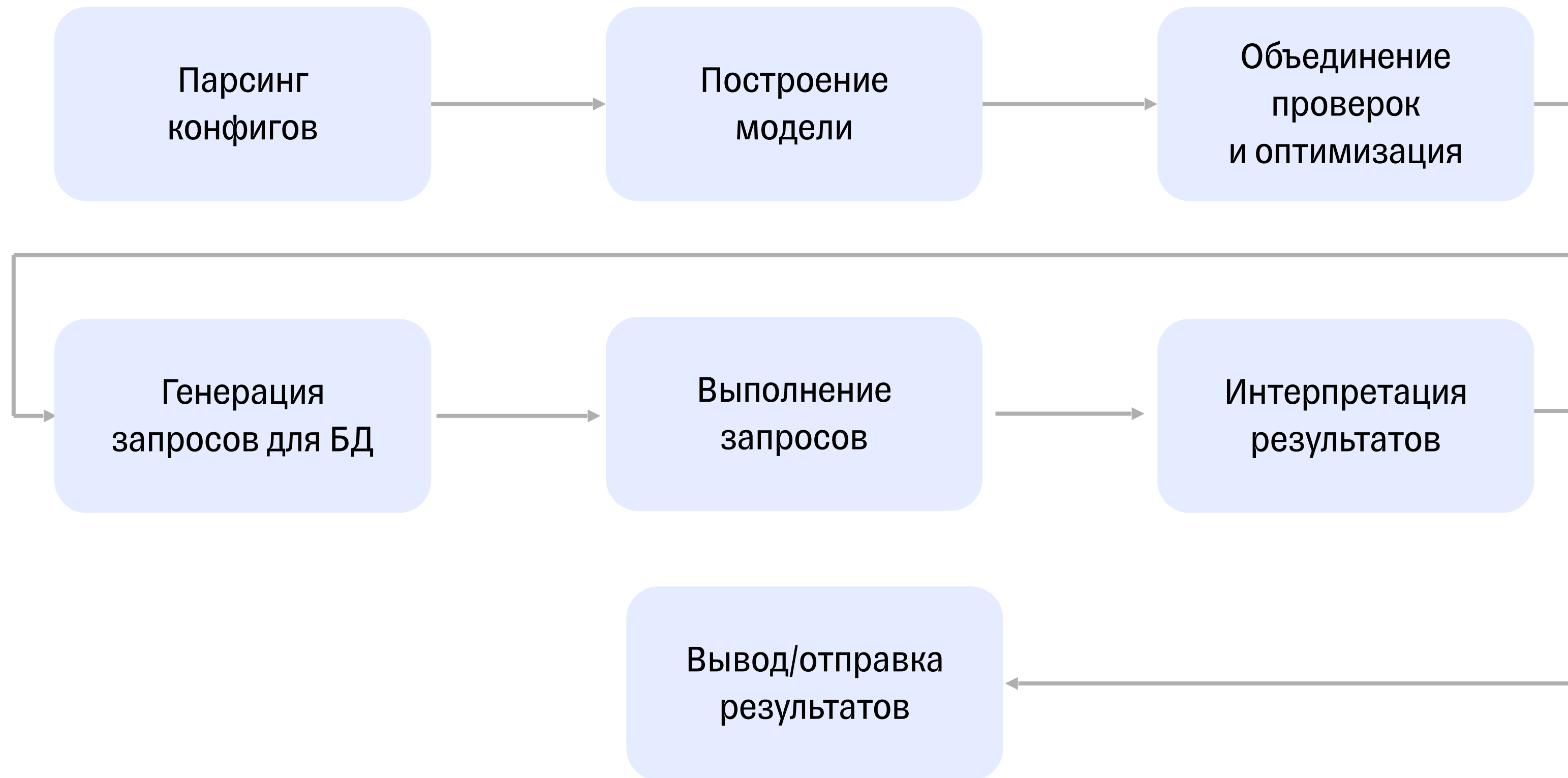


# Архитектура библиотека DQ проверок





# Архитектура библиотека DQ проверок



**SODA** 



 **great  
expectations**

# Soda Core **SODA**

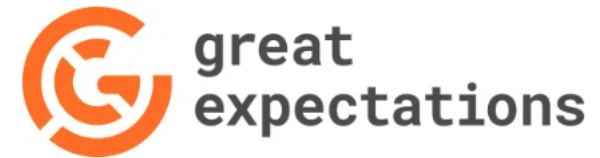
- Github Start: 1846
- API: yaml-based DSL
- Генерация SQL: строки
- Мало зависимостей
- Низкая модульность

```
# Checks for basic validations
checks for dim_customer:
  - row_count between 10 and 1000
  - missing_count(birth_date) = 0
  - invalid_percent(phone) < 1 %:
    valid format: phone number
  - invalid_count(number_cars_owned) = 0:
    valid min: 1
    valid max: 6
  - duplicate_count(phone) = 0
```



soda.io

# Great Expectation



- Github Start: 9700
- API: python function или JSON
- Генерация SQL: SQLAlchemy
- Много зависимостей
- Высокая модульность

Python

```
validator.expect_column_values_to_not_be_null("pickup_datetime")
validator.expect_column_values_to_be_between(
    "passenger_count", min_value=1, max_value=6
)
validator.save_expectation_suite(discard_failed_expectations=False)
```

# Soda Core vs Great Expectation

## Great Expectation

Type: python ▾

```
1 batch.add_source("select * from prod_advtmart.social_stats where business_dt > cur_date")
2 batch.expect_column_values_to_not_be_null("advt_cabinet_nm")
3 batch.expect_compound_columns_to_be_unique(["business_dt", "banner_rk", "source_code"])
4 batch.expect_column_values_to_not_be_null(column="advt_group_nm", row_condition='col("source_code")== "mytarget"')
```

## Soda Core

Type: python ▾

```
1 '''
2 filter prod_advtmart.social_stats [daily]:
3 |   where: business_dt > current_date'
4
5 checks for prod_advtmart.social_stats [daily]:
6 |   - missing_count(banner_nm)
7 |   - duplicate_count(business_dt, banner_rk, source_code)
8 |   - missing_count(advt_group_nm):
9 |     filter: sales_territory_key = 11
10 '''
```

# Soda Core vs Great Expectation

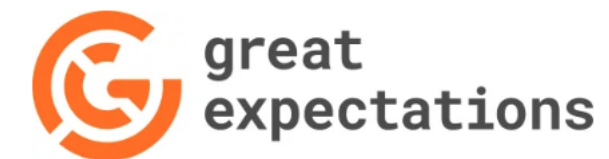
По функционалу  
проверок — паритет

Тип метрики	Подтип	GreatExpectation	Soda Core
По колонке	Числовые агрегаты	+	+
	Строковые агрегаты	+	+
	Дубликаты/уникальность	+	+
	Null	+	+
	countif, количество или процент по условию	+	+
	Непрерывность	+	-
По паре колонок	Сравнение значения	+	+
	Консистенстность/ссылочная целостность	+	+
На основе агрегатов		+	-
Статистика	PSI и тд	+	+
ML	Аномалии	-	+
	Прогнозирование	-	+
Проверка на схему		+	+

# Soda Core vs Great Expectation

The logo for Soda Core, featuring the word "SODA" in a bold, black, sans-serif font, followed by three small green dots of varying sizes.

- yaml-based DSL
- Низкая модульность
- Мало зависимостей
- Все проверки транслируются в SQL

The logo for Great Expectations, featuring an orange circular icon with a white stylized 'G' inside, followed by the text "great expectations" in a lowercase, sans-serif font.

- Python или JSON
- Высокая модульность
- Много зависимостей
- Часть проверок реализованы только в pandas

# **Data Checker – DQ библиотека**



# Data Checker – DQ библиотека

- Python библиотека основанная на Soda Core

# Data Checker – DQ библиотека

- Python библиотека основанная на Soda Core
- Декларативное описание проверок с помощью yaml

# Data Checker – DQ библиотека

- Python библиотека основанная на Soda Core
- Декларативное описание проверок с помощью yaml
- Большой набор встроенных функций проверки данных из коробки

# Data Checker – DQ библиотека

- Python библиотека основанная на Soda Core
- Декларативное описание проверок с помощью yaml
- Большой набор встроенных функций проверки данных из коробки
- Расчет примеров ошибок

# **Data Checker – доработки Soda Core**

# Data Checker – доработки Soda Core

- Интеграция с Helicopter (Notebooks)

# Data Checker – доработки Soda Core

- Интеграция с Helicopter (Notebooks)
- Интеграция с Data Metrics (Каталог проверок)

# Data Checker – доработки Soda Core

- Интеграция с Helicopter (Notebooks)
- Интеграция с Data Metrics (Каталог проверок)
- Переделали сборку пакетов



# Data Checker – доработки Soda Core

- Интеграция с Helicopter (Notebooks)
- Интеграция с Data Metrics (Каталог проверок)
- Переделали сборку пакетов
- Добавлены проверки на аномалии на ETNA (T-bank time series library)

**Выполнение проверок**



Helicopter  
(Notebooks)

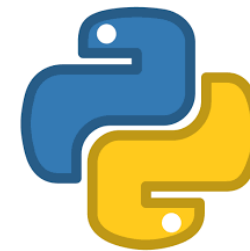


Tedi  
(ETL)

**Логика проверок**



raw/functions



raw/in-house library



Библиотека DQ  
проверок

**Каталог  
и бизнес  
процессы**



(Data Metrics)  
Каталог проверок

# Модульная архитектура – выводы

# Модульная архитектура – ВЫВОДЫ



Дешевле за счет переиспользования ГОТОВЫХ КОМПОНЕНТОВ

# Модульная архитектура – ВЫВОДЫ



Дешевле за счет переиспользования ГОТОВЫХ КОМПОНЕНТОВ



Легче встраивать в другие инструменты

# Модульная архитектура – ВЫВОДЫ



Дешевле за счет переиспользования ГОТОВЫХ КОМПОНЕНТОВ



Легче встраивать в другие инструменты



Легче подстраивать под пользовательские процессы

# Модульная архитектура – ВЫВОДЫ



Дешевле за счет переиспользования ГОТОВЫХ КОМПОНЕНТОВ



Легче встраивать в другие инструменты



Легче подстраивать под пользовательские процессы

Дорого и долго разрабатывать самим

# Что не хватает





# Что не хватает



«Простого» заведения проверок  
в простых кейсах



# Что не хватает



«Простого» заведения проверок  
в простых кейсах



Фрагментированный  
пользовательский опыт — работа  
в нескольких инструментах

**Этап 4.**  
**Внедрение**

# Кейс 1. Генерация кода для «массовых проверок»

Type: python ▾

```
1 tables = get_all_tables_form_schema("prod_v_schema")
2 check_str = ""
3
4 for (table, keys) in tables:
5     check_str += f'''
6         checks for {table}:
7         - missing_count({keys}):
8           name: {table_name}_missing_count
9         - duplicate_count({keys}):
10          name: {table_name}_nonunique
11         - confidence level for rowcount with daily granularity of dttm > 0.95:
12          name: {table_name}_rowcount_anomaly
13         - row_count:
14          name: {table_name}_count
15     '''
16
17 dc.run(check_str)
```

Заголовок набора метрик

prod\_ods\_siebel 408

prod\_ods\_tcrm\_proc\_development 33

prod\_ods\_credb 261

prod\_ods\_fwb 350

prod\_ods\_xxi 691

prod\_ods\_tcrmp\_backend 96

prod\_ods\_tcrm\_tasks 440

prod\_ods\_acq 534

# Кейс 2. Шаблоны

<a href="#">Проверки DDS.CHAT_SEGMENT за все время</a> 15	dwh_dko.DDS_CHAT_SEGMENT	⚠ Неудовлетворительное	2024.08.25 03:37:29+03:00
<a href="#">Проверки DDS.CHAT_SEGMENT за вчера</a> 16	dwh_dko.DDS_CHAT_SEGMENT_yesterday	✅ Отличное	2024.08.25 03:33:24+03:00
<a href="#">Проверки DDS.CHAT_SEGMENT за месяц</a> 15	dwh_dko.DDS_CHAT_SEGMENT_month	❌ Плохое	2024.08.25 03:33:00+03:00

Тип: python ▾

```
1 conditions = {
2     "yesterday": "processed_dttm >= CURRENT_DATE - INTERVAL '1 day' AND processed_dttm < CURRENT_DATE",
3     "this_month": "'processed_dttm >= DATE_TRUNC('month', CURRENT_DATE) AND processed_dttm < DATE_TRUNC('month', CURRENT_DATE) + INTERVAL '1 month'"
4 }
5
6 check_str = f'''
7     filter prod_v_dds.crm_user [filtered]:
8         where: {conditions[yesterday]}
9     checks for prod_v_dds.crm_user [filtered]:
10        - duplicate_percent(crm_user_rk, valid_from_dttm):
11            name: DDS_CRM_USER_crm_user_rk_dupl
12        - missing_percent(crm_user_rk):
13            name: DDS_CRM_USER_crm_user_rk_notnull
14        - missing_percent(crm_user_id):
15            name: DDS_CRM_USER_crm_user_id_notnull
16        '''
```

# Кейс 3. Notebook based UI

Динамические формы

Data Metrics Parameters (required)

Metrics Space	\$space	Metrics Set (required)	\$metrics_set	Token (required)	\$token
dwh_fs	X	dwh_feature_store	X	9ede333a-f5b1-47c1-bddf-6a7c83c	X

---

Data Checker Configuration wiki container (required)

Wiki space Id	\$wiki_space_id	Business process page name	\$business_process_name	Config block title	\$block_title
DW	X	DQ DWH_Feature_Store	X	Описание метрик для Data C	X

---

Metrics Calculation Parameters

<input type="checkbox"/> Sync metrics	\$sync_metrics	<input type="checkbox"/> Update alerts	\$update_alerts	Data Metrics Set	\$server	GreenPlum Service	\$gp_service
				prod	X	gp	X

---

Helicopter note URL

\$paragraph_url
DQ DWH Feature Store ci

---

Alerts Configuration Parameters

Helicopter note URL	\$paragraph_url	Responsible	\$responsible	Recipients	\$recipients
https://helicopter.tcsbank.ru/notes/	X	@m.m.nikiforov	X	~dwh-dq-risk-alerts	X

# **Заключение**

# Заключение



# Заключение



Повышение качества  
данных – ежедневная  
рутина

# Заключение



Повышение качества  
данных – ежедневная  
рутина



Никакие инструменты  
не решат ваших проблем,  
но упростят путь  
к качественным данным

# Заключение



Повышение качества данных – ежедневная рутина



Никакие инструменты не решат ваших проблем, но упростят путь к качественным данным



DQ инструменты могут адаптировать под запросы и возможности, начните с простого



**Спасибо!**