火山引擎 DataFun.

# 2023 DataFunCon

## 智能化、自动化，揭秘字节跳动数据质量前沿探索

演讲人：周方圆 火山引擎DataLeap团队

# 智能数据质量

## What & Why & How

# 行业动向

国内的互联网行业更彻底的进入了大数据的应用时代：

- 云上大数据体系：数据收集、存储、处理、应用的技术栈
- 加易用、低门槛，同时有更成熟的数据应用工具，如可视化、自动化决策、机器学习

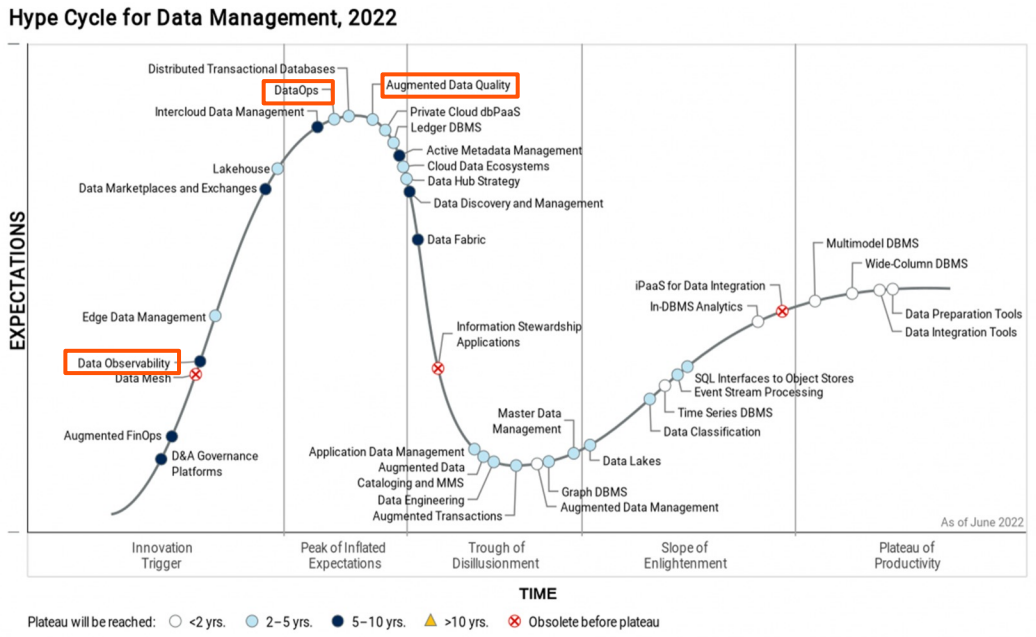
数据“用起来”的问题基本的得到了解决。随之而来的是数据治理的问题。

以火山引擎的大数据产品为例：

# 行业动向

数据质量：为什么要重视数据质量？

*规模小的时候，速度就是效率，规模大到一定程度时，质量就是效率*



# 数据质量的基础概念

**经典的数据质量保障方法：配置质量检查规则 (Assertions)**

Freshness：数据新鲜度 Data Delay Alarm

Volumn：数据量 Missing or Too much data tests

Accuracy：数据的正确性 Numeric distribution tests, String pattern tests

Completeness：数据完整性 NULL values tests

Uniqueness：数据唯一性 Unique key tests

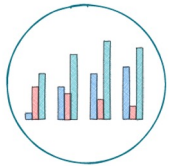
Integrity：数据的主外键正确性 Referential tests

# 数据质量的基础概念

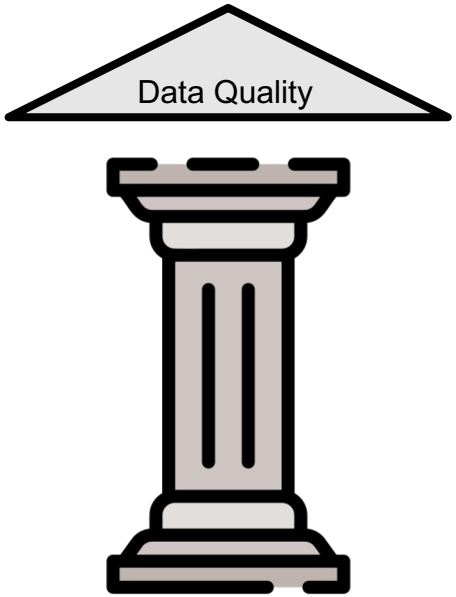
质量检查(Assertions)是数据开发的单元测试 + 持续监控

数据探查
Profiling

设置规则
Apply Rule



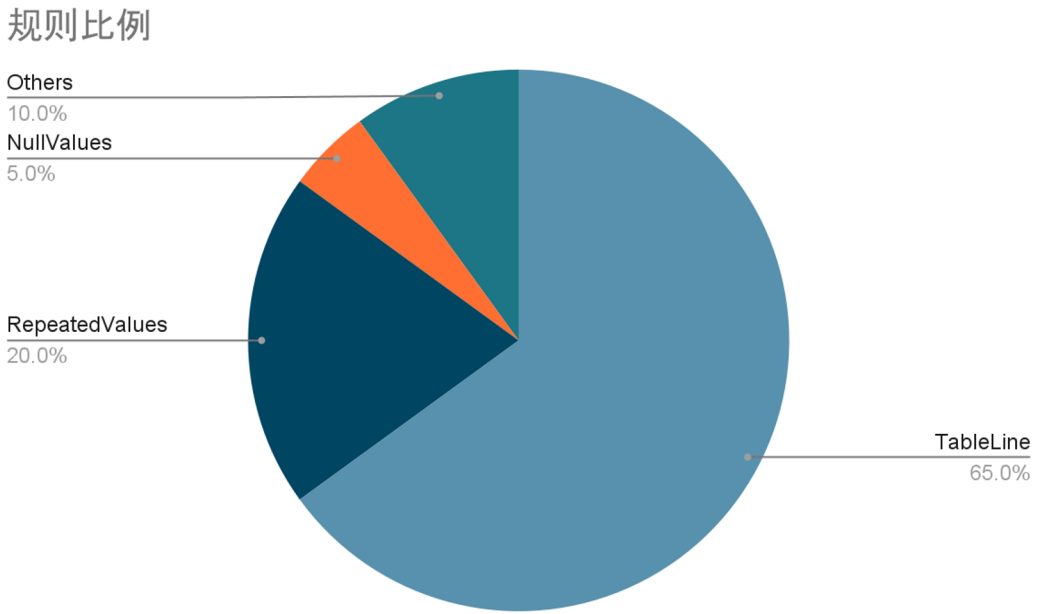
例行监控
Monitor




Assertions

# 质量检查方法的问题

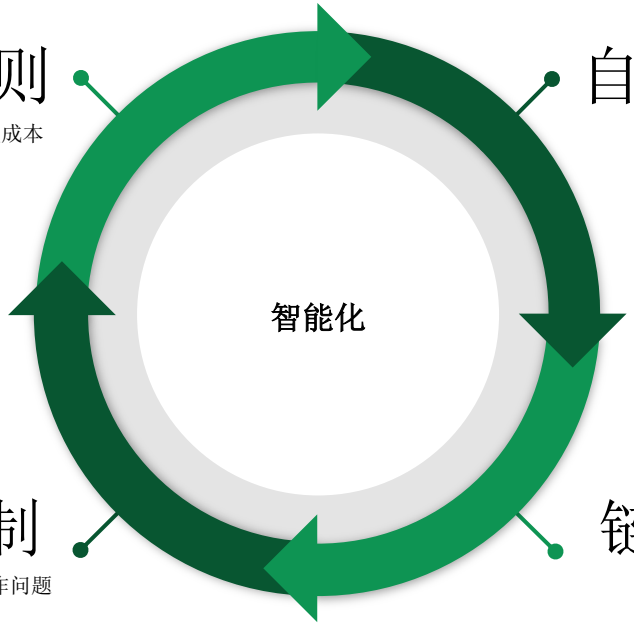
实际配置情况：表行数，主键重复 > 80%



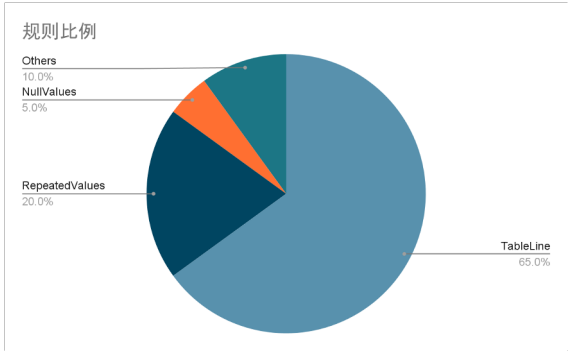

规则配置渗透率不及预期的原因

1. 配置繁琐
2. 依赖经验
3. 往往是事后补充

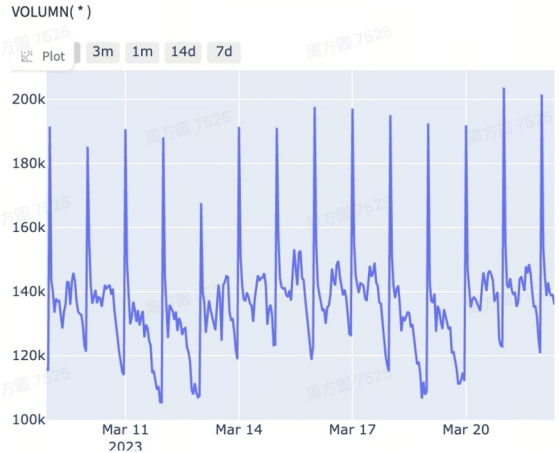
# 智能数据质量

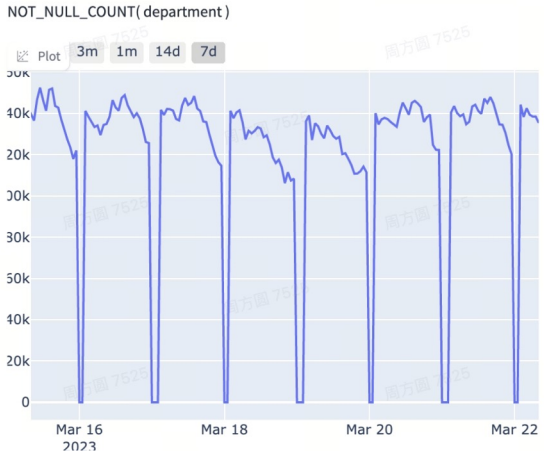


# 自动检测




无规则：基于自动异常检测算法发现异常



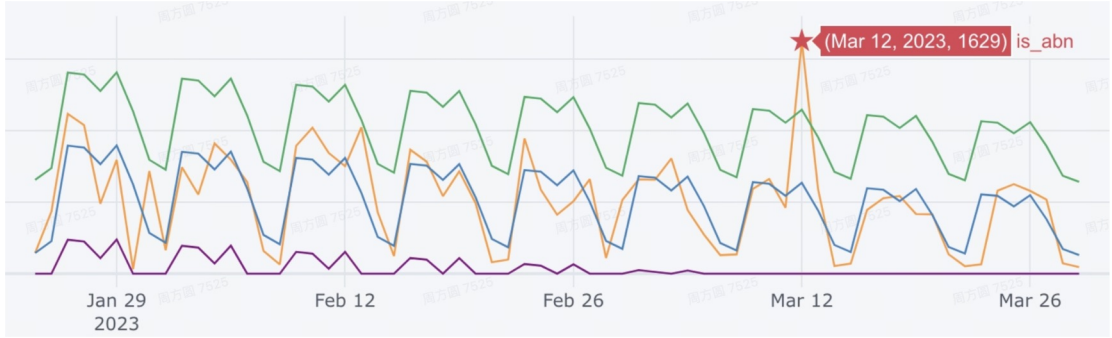




无规则的缺点：复杂指标收集的成本较高

1. Cardinality 数据维度

2. Regex 字符串模式匹配

3. Percentile 数据分布



# 规则推荐

减少指标收集成本：
场景推荐规则

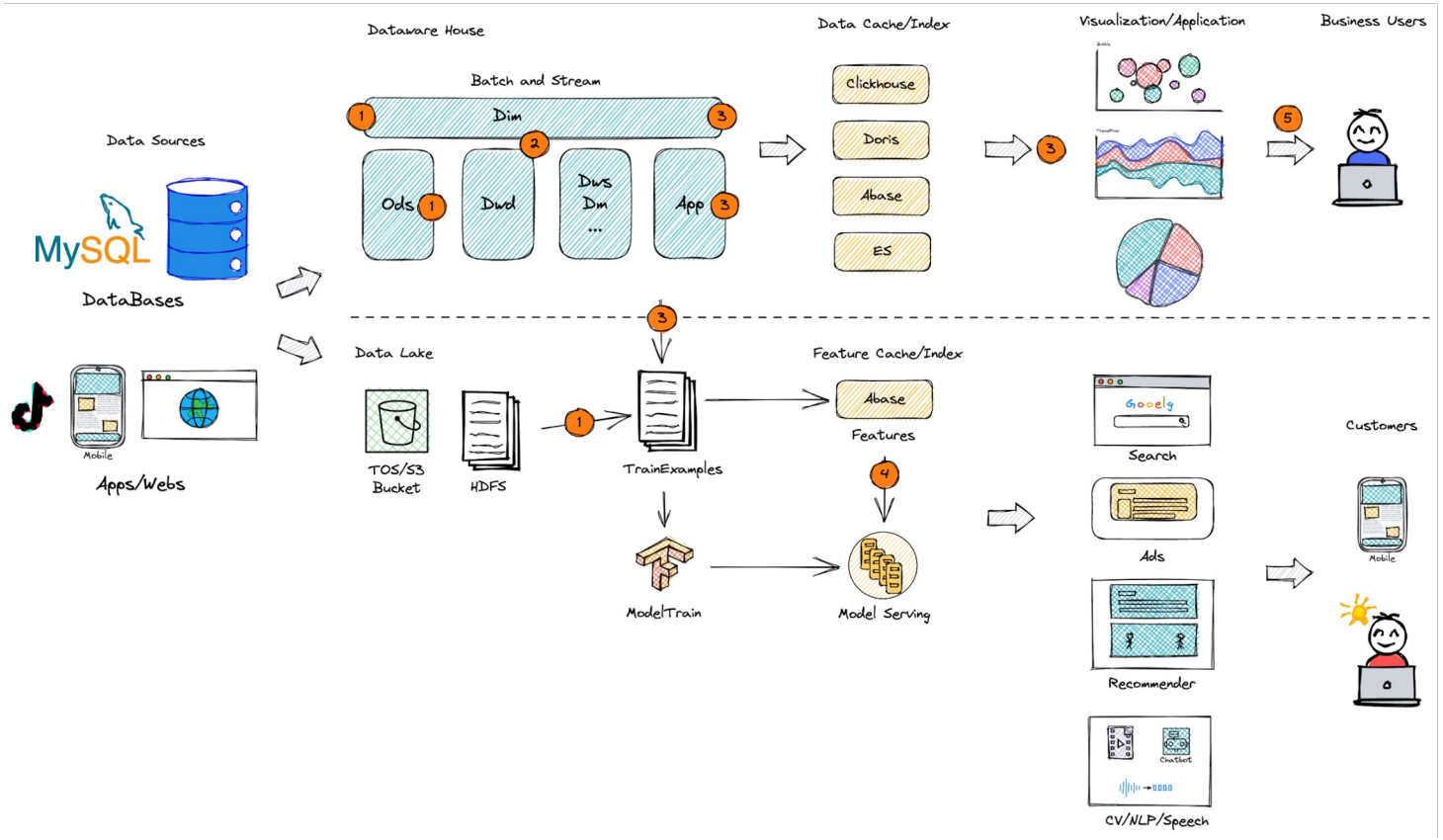
①外部数据入口

②数据链路开发

③数据应用

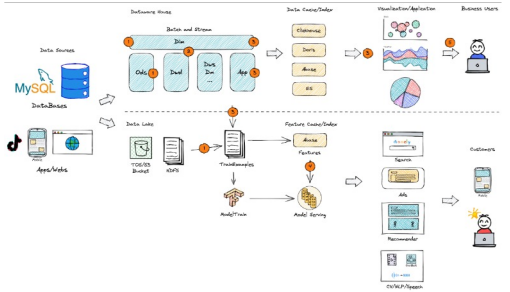
④模型特征

⑤业务应用



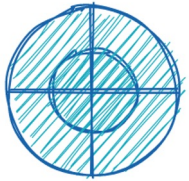
# 规则推荐

| 场景 | 核心问题 | 常见推荐规则 |
| --- | --- | --- |
| 外部数据入口 | 稳定性、规范性 | 新鲜度、数据量、数值范围、字符串模式... |
| 数据链路开发 | 数据模型符合预期 | 重复数据 |
| 数据应用 | 语义级数据质量 | 真实类型判断、数值范围、字符串模式、时序范围预估、完整性检查... |
| 模型特征 | 数据分布漂移 | 数据缺失、数据分布距离、OOV值... |
| 业务应用 | 指标监控 | 波动率阈值、异常检测 |




| object | instance | rule |
| --- | --- | --- |
| table | dataleap_meta_dw.dwd | • 行数 >0 |
| field | task_id | • 空值占比 <0.0%<br>• 重复值占比 <3.46% |
| field | project_id | • 空值占比 <0.0% |
| field | database_name | • 空值占比 <0.0% |
| field | table_name | • 空值占比 <0.0% |
| field | owner_user_name | • 空值占比 <0.0% |
| field | project_name | • 空值占比 <0.0% |
| field | engine_id | • 空值占比 <0.0%<br>• 枚举值 [0] |
| field | storage_engine | • 空值占比 <0.0%<br>• 枚举值 [1, 5, 2, 9, 3, 4, 6, 0] |
| field | path | • 空值占比 <16.08% |
| field | code | |
| field | task_type | • 空值占比 <0.0% |
| field | conf | • 空值占比 <0.0% |
| field | status | • 空值占比 <0.0%<br>• 枚举值 [2.0, 4.0, 3.0, 1.0] |

# 规则推荐

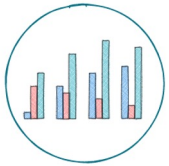


场景感知
Scenery
Strategy

适应性数据探查
Adaptive Profiling

自动检测
Auto-Detect

规则推荐-设置
Rec Rule & Apply

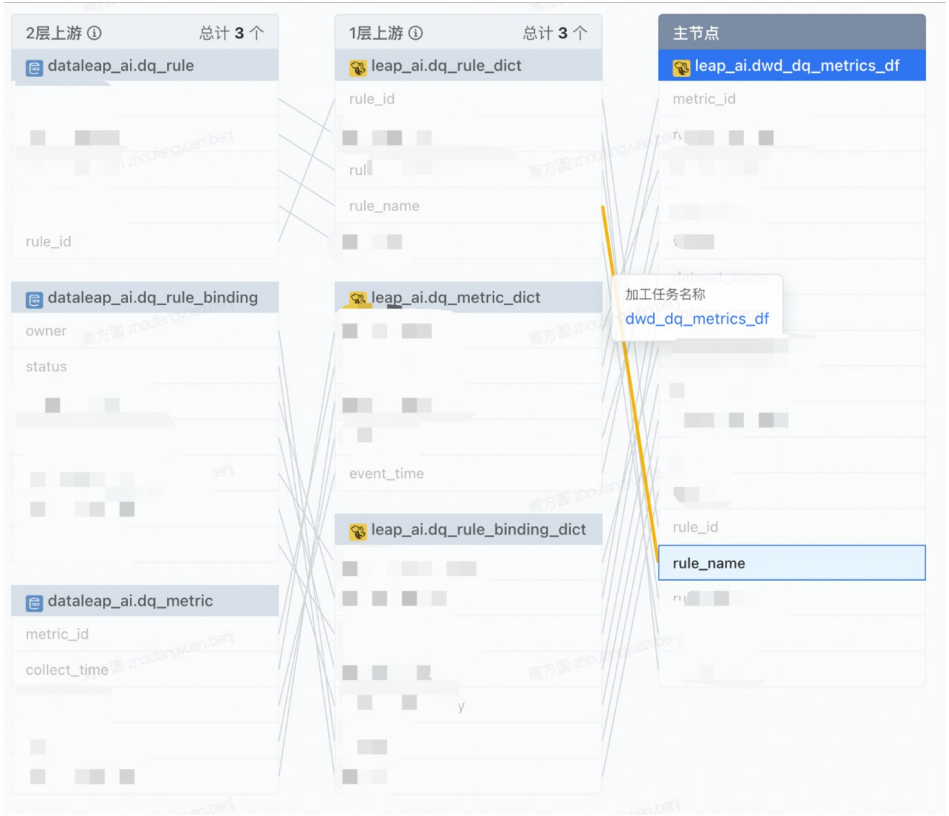


例行监控
Monitor

# 链路根因诊断

数据链路是一个整体，应用层质量问题需要在上游表中追查问题

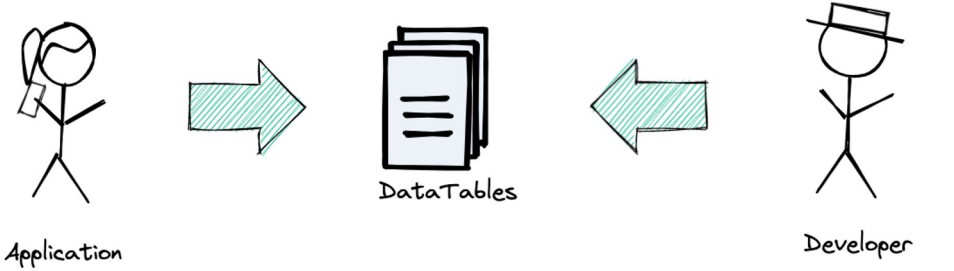
借助DataLeap的字段级全链路血缘功能，配合链路指标收集实现自动根因诊断



# 协作：数据质量协议

开发者和应用者的质量预期鸿沟：

*金额为什么是-1？*
*枚举值为什么增加了？*
*有些视频点击比曝光还多？*
*部分邮件/电话格式无效？*
*XX字段缺失率升高？*
*…*




*业务建模特性*
*枚举值增加业务变更不受数仓控制*
*业务处理特性导致*
*数据检验不严格导致（但也不能丢）*
*业务变更/节假日/事件导致*
*…*

数仓需要保证数据质量！

基础质量有保证！这些不是质量问题！

你要保障什么规则，提需求过来，给你配置好。

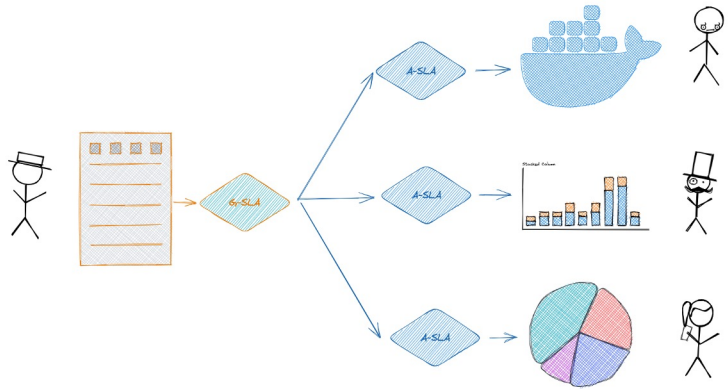
最终结果往往是往往可能是发生故障后，“运动式”配置一批监控，保障最重要的数据

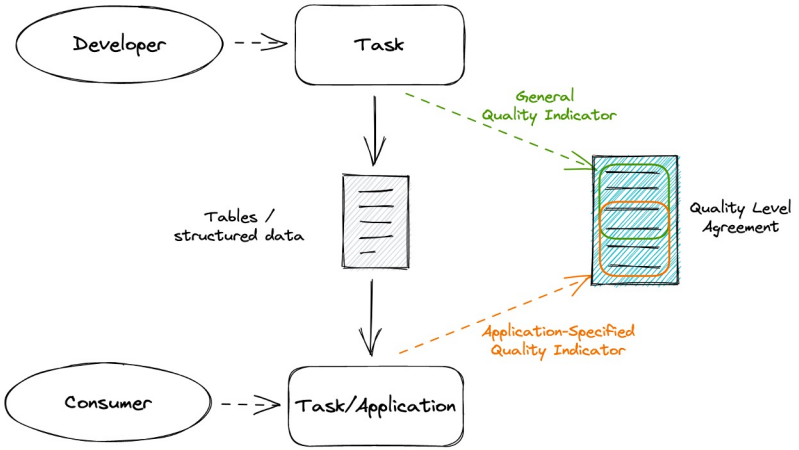
# 协作：数据质量协议

分级质量协议

1. General Service Level Agreement: 开发者提供的关键质量承诺
2. Application Service Level Expectation: 使用者预期的质量指标

应用者主动参与到质量工作中，多方应用者与开发者形成互动



# 总结：数据质量的四大支柱


| Assertions | Metrics | Lineage | Collaboration |
| --- | --- | --- | --- |
| 基础手段 | 自动检测 | 系统整合 | 开放协作 |

# 探索：ChatGPT与数据质量

减少规则设计的门槛：自然语言 -> 质量规则

**VIDEO**

# 火山引擎DataLeap：端到端的数据质量保障

## 自主探查

- 事前-数据探查

上线前的数据进行测试，查看内容的分布和数据特征，保证数据符合业务预期，避免下游用户因为数据错误导致决策失误

## 强规则熔断

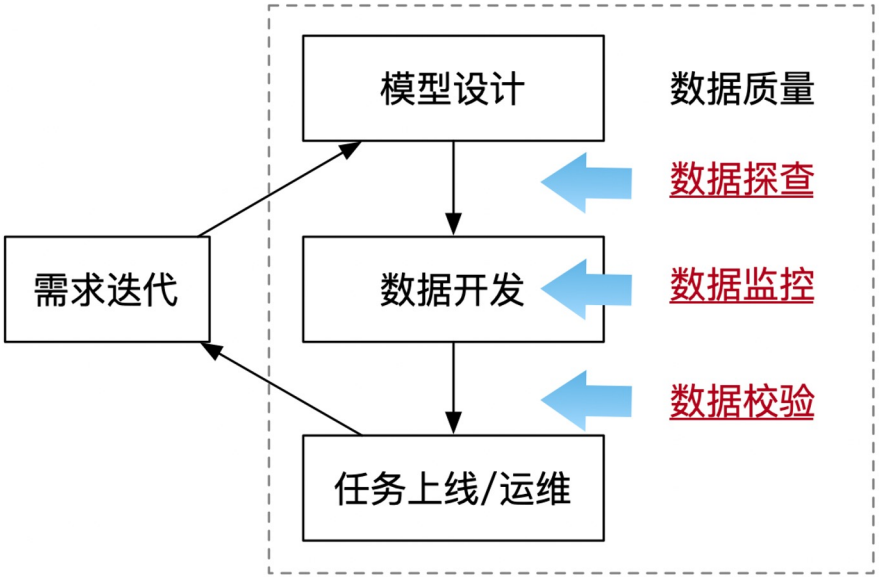
- 事中-数据监控

六要素监控模板，支持多表关联、多行多列等复杂质量规则配置，满足复杂的业务场景需求，强弱规则机制，阻断下游持续污染

## 数据校验

- 事后-数据对比

丰富的自定义对比能力，验证开发代码逻辑的准确性与数据结果，保障数据按预期产出



# 欢迎联系我们

# Q & A

“字节跳动数据平台” 微信公众号

回复“招聘”，查看DataLeap最新岗位

扫码添加小助手

进入字节跳动数据平台官方交流群

# 2023 DataFunCon

— THANKS —

# 感谢您的观看