DevData Talks | 思码逸

利用MVP方法

构建最小可行的效能度量实例

直播嘉宾:关钦杰

在 DevData Talks
开放务实地聊聊研发效能

分享合辑

研效交流群

每月直播 干货满满

行业 KOL + 企业研发效能负责人 + 资深效能顾问
从效能建设路径到不同场景应用，分享前沿观点与先进实践

开放探讨 共同成长

直播 Q&A + 交流社群
直播嘉宾面对面答疑解惑，交流群随时随地探讨效能话题

关于思码逸

作为DevData Talks 的发起方，思码逸专注为企业研发团队提供效能数据分析，呈现效率、质量、人才发展等多视角数据洞察，辅助团队决策与工程改进。

思码逸已服务 腾讯、美团、滴滴出行、中国平安、泰康保险、戴尔EMC等众多行业标杆客户。

讲师介绍

关钦杰（Amy）

北京思码逸科技有限公司　咨询总监

原中兴努比亚研发提效内部顾问、敏捷实践教练、质量责人。曾为比亚迪、泰康人寿、中电万维、戴尔EMC等多家大中型企负业提供研发提效咨询及培训服务。10余年研发效能领域实践及咨询经验，曾带领团队基于过程性能分析，使产品性能及质量得到有效提升。在软件可视化度量及分析应用领域有着丰富的经验和深厚积累。

01 研发效能度量有哪些坑点？

02 如何构建效能度量的MVP？

03 MVP方法应用实例及原则

理想 vs 现实

– 你希望的度量 –

灵活、可靠、端到端、高大上

– 现实中的度量 –

受限、低效、噪音大、体验差

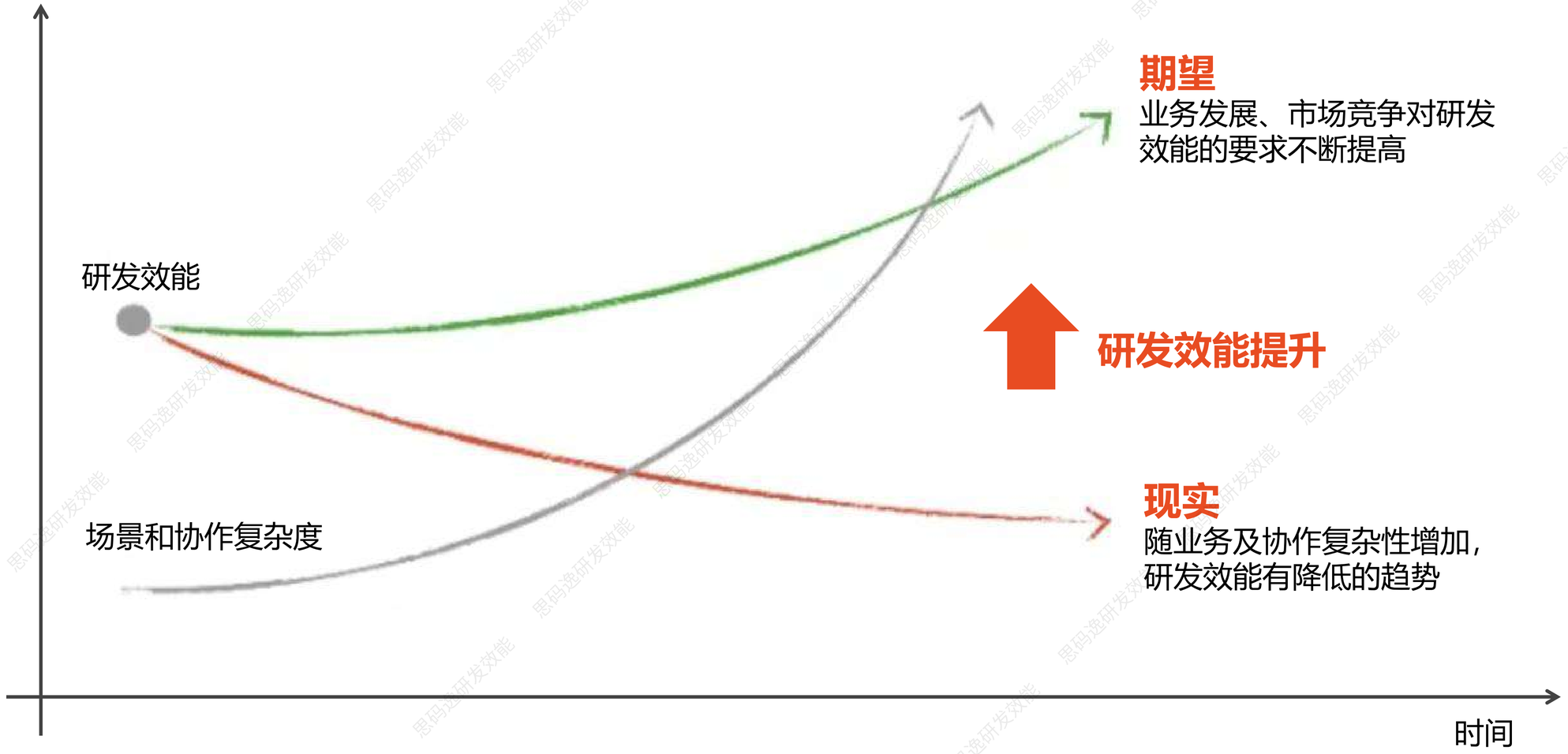
度量失败的例子

古德哈特定律（Goodhart's law）

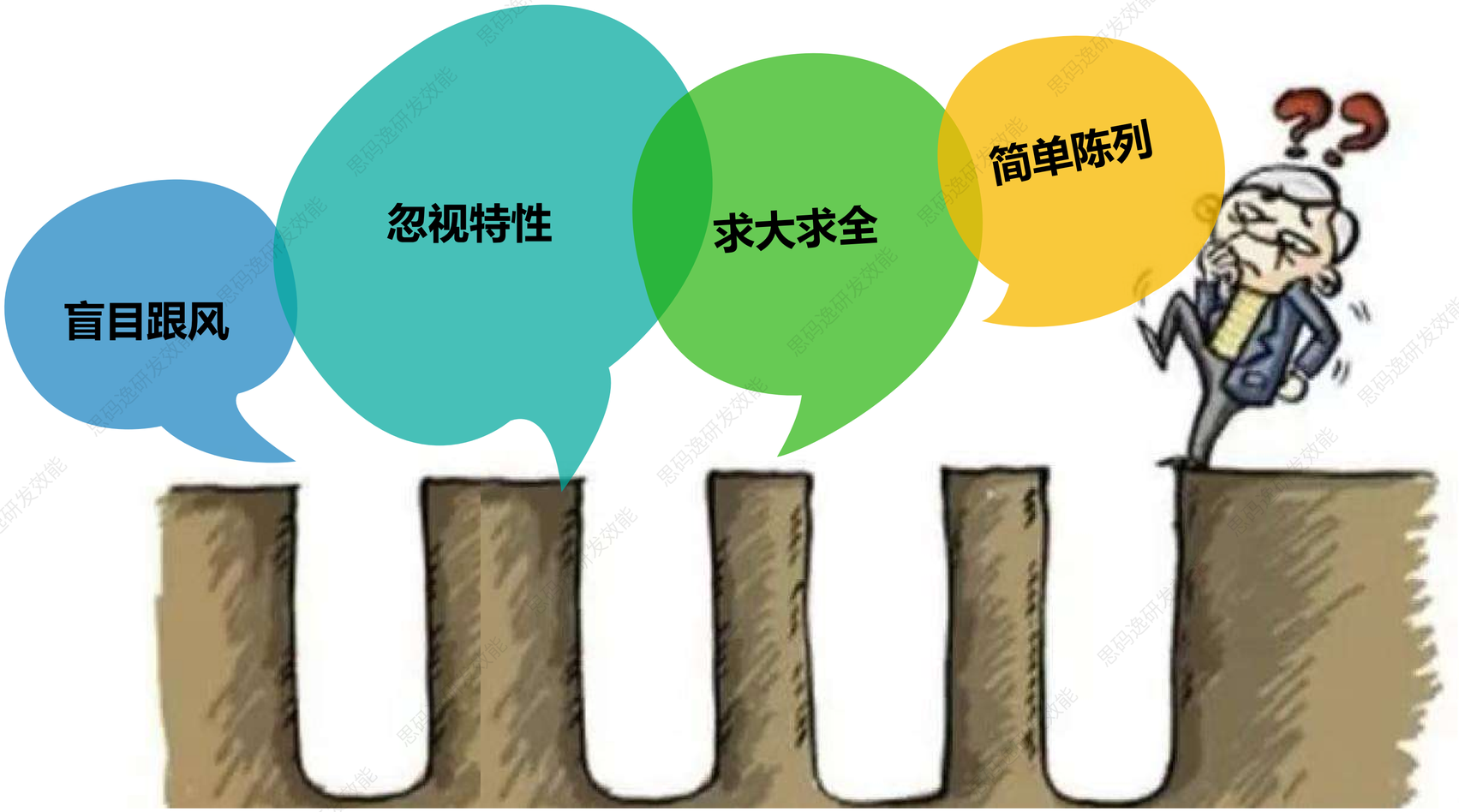
眼镜蛇事件

修了拆、拆了修的城市道路

研发效能的度量更具挑战性



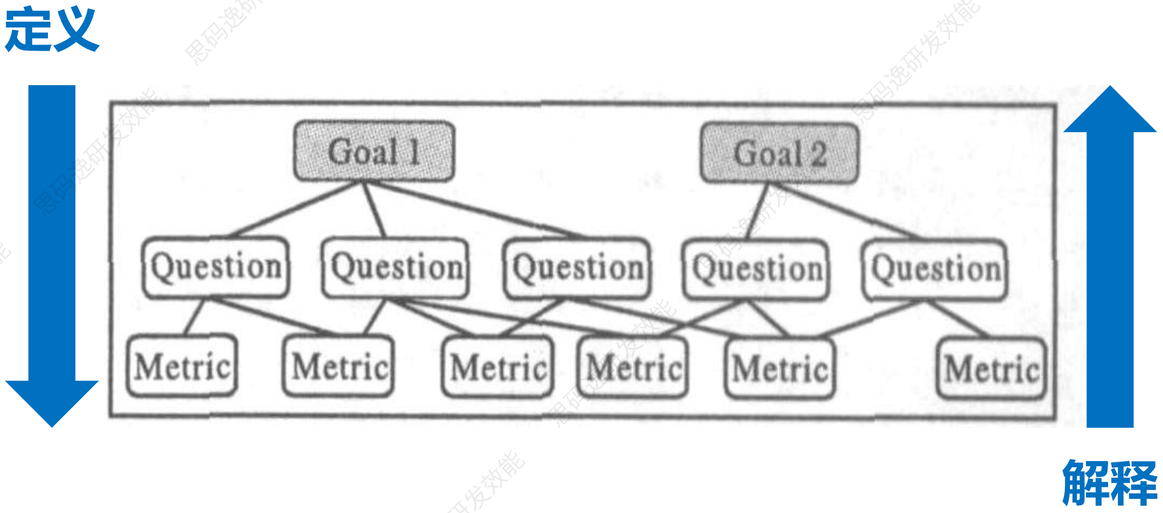
度量设计的四大坑点



避坑指南一：

要目标驱动，不要盲目跟风

GQM三层分析法



第一层：确定目标（Goal）

合理的目标通常是一项活动能否成功的关键。

第二层：提出问题（Question）。

针对目标，提出一系列问题，这些问题描绘出达到目标所需要关注的各个方面。

第三层：定义度量指标（Metric）。

根据问题，选择相应的度量指标。

避坑指南二：

要灵活取舍，不要忽视特性

☑

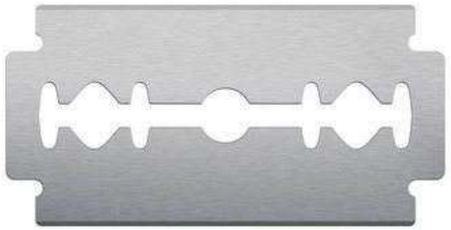
☒

避坑指南三：

DevData Talks × 思码逸

要少而精，不要大而全

如无必要，勿增实体。——奥卡姆剃刀原理



- 每一个**活动**是否都有价值?
- 每一份**文档**是否都有价值?
- 每一个**指标**是否都有价值?
- 每一个指标是否都能回答一个**关键问题**?
- 获取指标的**成本** vs 度量**收益** 哪个更高?

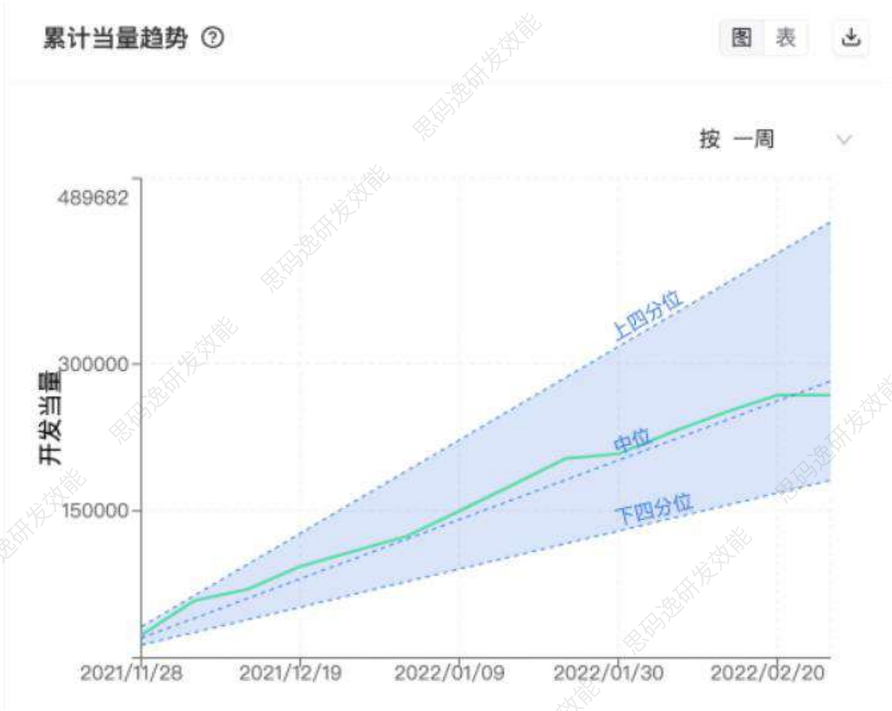
化繁为简，把握关键。抓住重点，找到要害。

merico

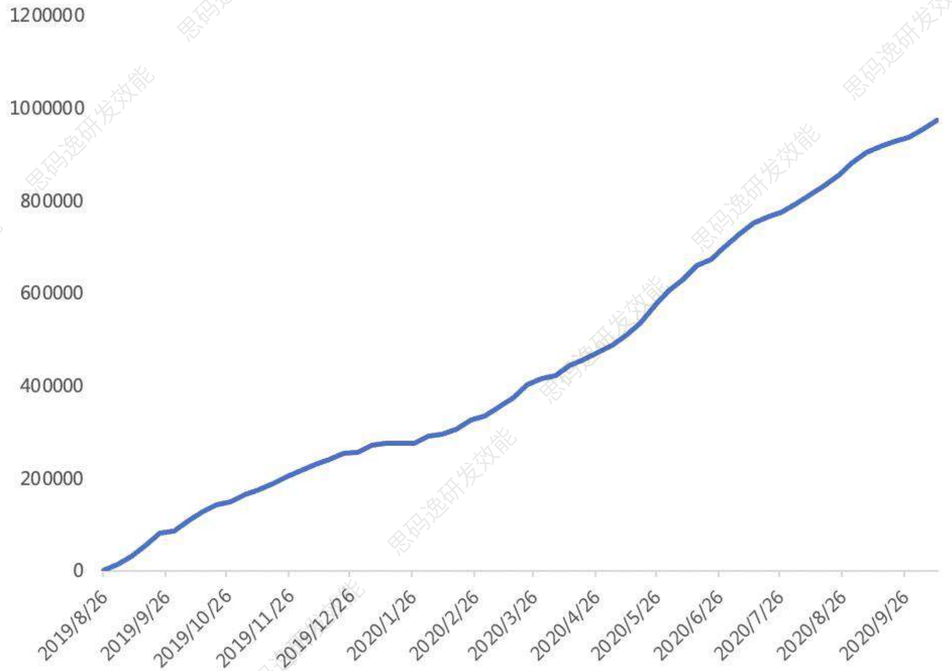
避坑指南四：

要建立标尺，提示异常，不要简单陈列

代码增速合理，处于行业中等水平



团队累计代码贡献（当量）走势



图形来源：思码逸深度代码分析平台

01 研发效能度量有哪些坑点？

02 如何构建效能度量的MVP？

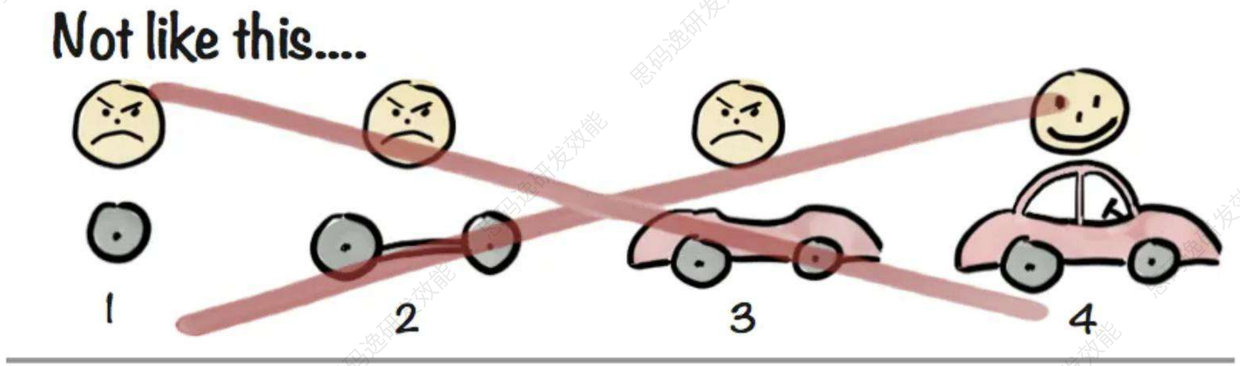
03 MVP方法应用实例及原则

面对度量设计的挑战，有什么切实的可行的方法？

☒ 不要堆砌

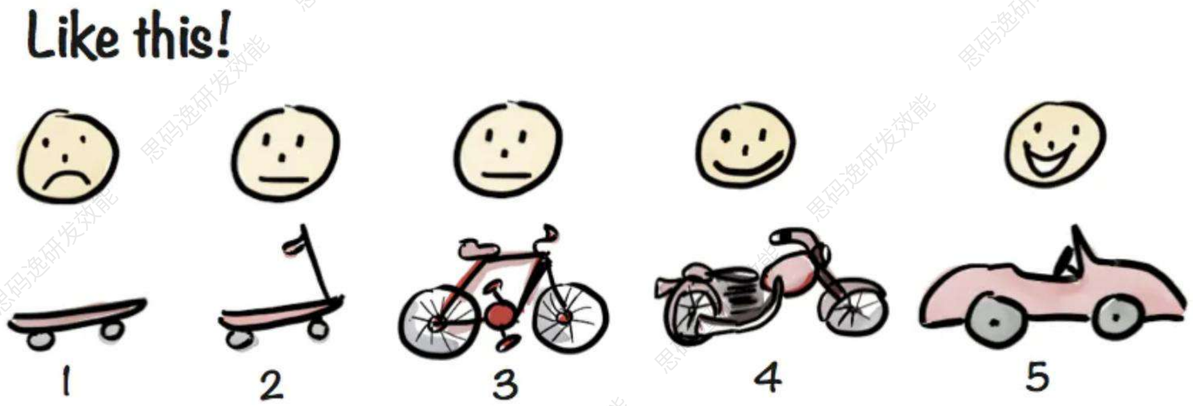
不要重复

……



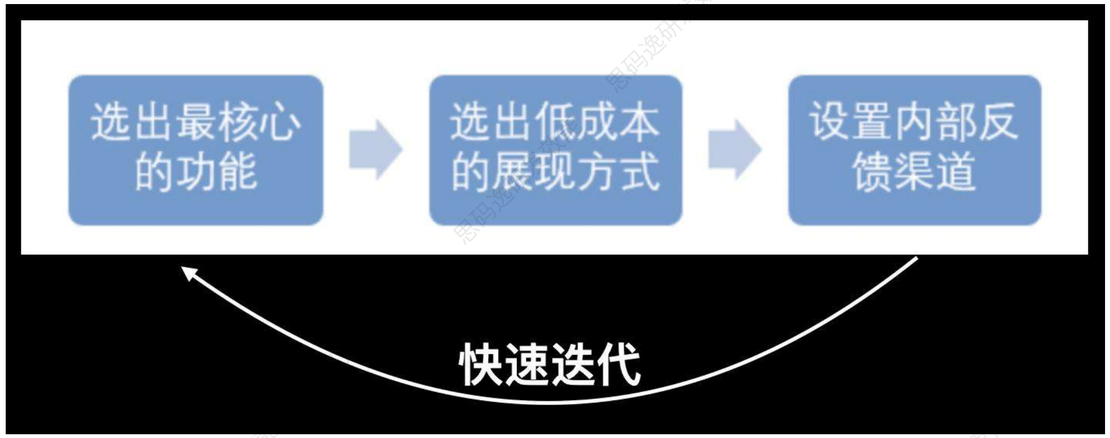
☑ 可视化

端到端



度量体系是产品而不是项目

产品的MVP → **效能度量的MVP**



核心原则：从最终用户可见的**价值内核**开始、进行**持续迭代**

效能度量MVP版本的构建框架

目标驱动，构建原型、获取反馈，迭代演进。

方案构建

选出最核心的指标

目标驱动，聚焦场景及指标的最小集

+

低成本的展现方式

构建原型，梳理思路、建立认知

持续沟通

方案实施

获取反馈：短周期反馈

迭代演进，扩展，调整

01 研发效能度量有哪些坑点？

02 如何构建效能度量的MVP？

03 **MVP方法应用实例及原则**

实例背景

组织特征

- 业务部门：17个
- 业务差异：大（产品线、运维、资源...etc.）
- 代码仓库：9000+
- 研发工具：数据分布于不通工具、平台
- 项目模式：复杂（有敏捷、有瀑布）

主要诉求

- 借助**代码当量**，打开**开发黑盒**，建立满足**效能管理**需求的度量体系
- 为**不同角色**提供数据抓手，判断现状，识别异常
- 部门生产压力大，**聚焦做真正有价值**的事情

- **为效能管理提供数据抓手**
- **建立各部门北极星指标**
- **快速落地、闭环验证**

MVP度量版本实现的5步法

MVP初版

挖需求

Top 3的需求是什么?

需求背后的动机是什么?

做减法

哪些需求具有导向性?

哪些场景是信息及管理需求的最小集?

建原型

价值和结果是否可视化了?

是否给出了问题识别、预判的规则?

听反馈

业务方能得出想要的结论吗?

需要更多解释来理解指标和指示器吗?

做优化

聚焦目标

化繁为简

Step 1 Step 2 Step 3 Step 4 Step 5

持续沟通

Step1：挖需求

通过深度访谈，挖掘管理痛点，聚焦核心需求。

关键思考：

- **Top 3**的需求是什么？
- 需求背后的**动机**是什么？

访谈对象

- **不同角色视角**
  部门管理者、团队管理者
- **不同特性团队**
  产研团队、实施交付、资源中心

访谈问题

- **开放式问题**
- **封闭式问题**

深度访谈 → **挖掘痛点**

识别痛点

- **总结、反思访谈内容**
- **结构化管理痛点与诉求**

聚焦重点

- **圈定数据用户**
- **确定核心诉求**

Step2：做减法

DevData Talks × 思码逸

深入访谈、**挖掘需求**

	部门负责人	团队负责人	技术经理	项目经理
管理范畴	• 一般部门下面3个团队，1个团队下3个小组。 • 区分资源能力中心、行业部门、实施交付 1. **行业部门**：面向市场，向终端客户交付产品。负责软件生面周期的交付管理，包括售前支撑、产品规划、后期项目运维。 2. **实施交付**：全公司华中片区的交付实施，成熟产品的实施、推进项目验收。	• 协同多个业务完成项目目标，团队各项目间资源调配、问题处理。 • **团队的项目类型**： a. **产品研发**：通用项比较高的产品；政策导向指向的产品研发。 b. **交付项目**：对客户签订合同的项目 c. **运维实施**：负责成熟产品的实施、推进项目验收；代码产出主要为实施类脚本、实施工具的使用等代码。功能性代码由产品负责迭代升级。	• **项目的SM**。负责项目研发管理。 1. 梳理需求、分配团队成员、搭建架构、评审代码； 2. 组织团队成员拆解任务、估算需求； 3. 协助项目经理协调资源。 • **管理人数5~8人** • **开发工具**：idea，eclipse	• **项目的PO**。负责已签订合同的交付项目。对项目的需求负责，对交付负责。 • **项目类型**：纯软、集成、硬件、实施运维项目（以纯软项目为基础）。 • **项目周期**：项目周期为几月~年。周期内均有新功能交付，以迭代形式进行，迭代周期为2月/2周。 • **管理工具**：Jira
管理/决策场景	首要的是工作可视化、资源调配。（外部资源调配、内部资源利用、找到真正很忙的人避免人力依赖）	• 大项目间的资源协同（紧急的事务需要团队内部或者外部其他团队的协助） • 大客户推进中的问题解决	——	——
待提升/关注点/痛点	• **团队、小组整体贡献、项目产出。** 1. 团队产出在平均线以上，平稳向上，团队工作饱满 2. 生产率、需求的实际交付效率、需求交付周期 3. 代码内建质量达标，打标签的通用组件是被调用的 4. 计划与实际偏差、发布的及时性 5. 面向客户持续交付、代码及时提交发布	• **工作划分均衡性不足**、有些成员负责工作多、新进入人员工作量较轻。--实施 • **团队精细化管理有待提升**，不能了解到每一个人的需求及问题。--研发 • **关注客户满意度**： a. 初验、终验、上线节点按时完成。 b. 交付项目的质量容错率、性能指标 c. 缺陷占比、返工率 线上问题会分析：前端问题、数据问题、功能问题，问题分类分析。	• 单元测试覆盖率(60%函数覆盖) • 安全漏洞 • 缺陷密度（内部+线上）	• 项目、人员工作进度；
希望思码逸提供的量化信息	**--行业部门--** 1. **资源的调配优化**：各团队间负荷是否均衡，有数据的持续更新和趋势变化。 2. **团队&成员贡献**：团队、项目生产率的提升，代码注释、单元测试是否到位。 3. **个人技能瓶颈优化**：个人维度技能提升建议，能看到哪些做的好。 4. **价值产出**：希望看到工程师产出的价值--需求维度、代码质效&影响力。更希望将代码和需求关联，看需求的价值。 **--研发&产品--** 5. **代码复杂度的评估**：如增删改和逻辑复杂的代码编写，区分哪些同事是负责核心业务的。 6. **代码反复修改的频率或记录**：看代码的返工率或业务变化的情况作为管理参考。	• 指标的设定要考虑计划、周期，指标适用全部开发模型，保证灌布、敏捷项目都能获得帮助。 **--实施--** 1. **周报信息**：反馈交付进度、问题修复进度，可反馈给客户。 2. 纯实施层面的指标，阶段性上线的系统数量。 **--研发--** 3. 成员工作效能对比、成员的技能水平 4. 代码封装复用率高低排名 5. **正确衡量成员代码量、工作量**：单纯算代码提交量不能衡量一个人的工作量，因为有些项目有外部协作情况。 **--实施+交付--二次开发** 开发人员的每日负荷、近期负载变化情况。	• **人才培养**：个人成长、特长、人员除了提交代码量，更擅长哪些方面，可以针对性培养。 • **人员贡献度**：非单纯代码产出，看代码产出的效率、影响等。 • **工具直接展现具体人员的工作进展。**	**问题预警**：进度、质量。因为是交付型项目，项目经理和技术更关注质量

聚焦核心、少而精

角色维度 - 承上启下的**部门管理者**

场景维度 – **工作饱和度、效率、质量**

关键思考：

- 哪些需求具有导向性?
- 哪些场景是信息及管理需求的最小集?

merico

样例：定义指标

关键思考：

- 是否可以保证数据的**准确性**？
- 是否能保证指标**健康度**？
- 指标的获取**成本**是否大于**收益**？

样例：GQM指标设计

有价值的、健康的

希望得到的答案

最小的、必要的

典型性

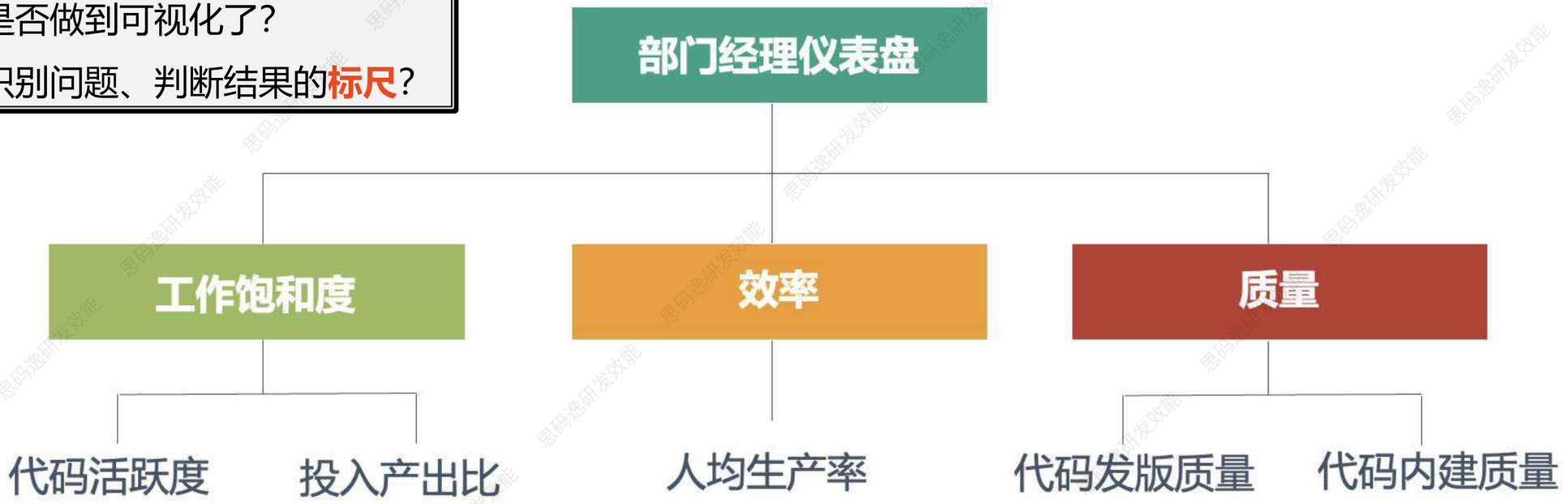
差异化

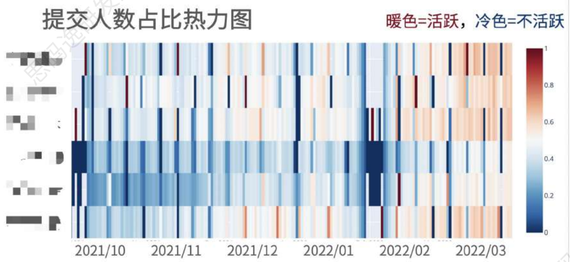
业务特性	目标（Goal）	回答问题（Question）	指标（Metric）	分析说明
资源部门	衡量团队成员的工作饱和度，以合理调配人力。	团队工作饱和度如何?	代码活跃度	定义： 空置率：了解开发过程中空闲等待的时间。 溢出率：了解加班频率及工作量
		人力的投入与产出是否匹配?	投入产出比	定义： • 投入产出比=新增当量/人力 % 标尺： • 历史基线：历史投入产出比的变化和增长幅度。
产研部门	了解团队的开发效率，以合理预估项目人力及交期。	团队开发效率如何?	人均代码当量 同比/环比 行业中位数	标尺： • 与行业中位数对标，了解对标水平。 • 同比/环比，了解变化水平。
实施部门	了解代码发版质量，以判断可部署的稳定版本。	代码的发版质量如何?	千当量缺陷率	标尺： • 缺陷收敛周期 • 缺陷密度
		代码的内建质量如何?	五项质量指标：	指标维度： 代码模块性、代码复用度、重点问题积压密度、单元测试覆盖率、注释覆盖率

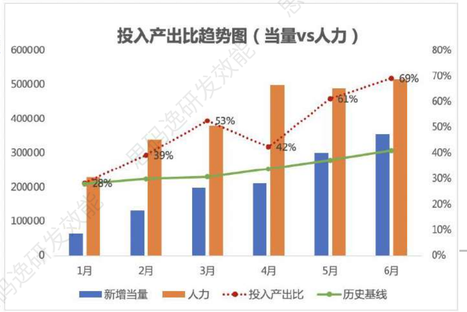
Step3：建原型

关键思考：

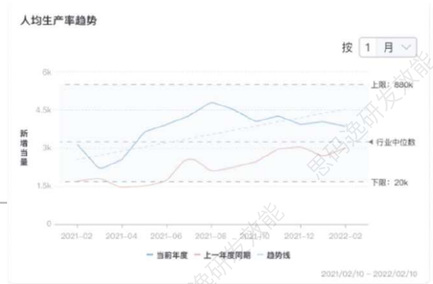
- **价值**和**结果**是否做到可视化了？
- 是否提供了识别问题、判断结果的**标尺**？

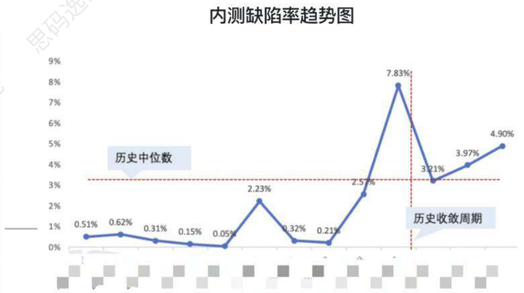


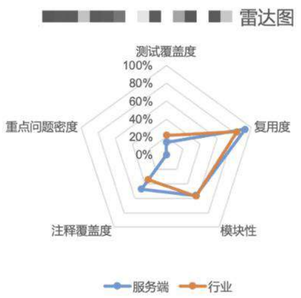




人均生产率较去年同比提升37%，环比提升12%







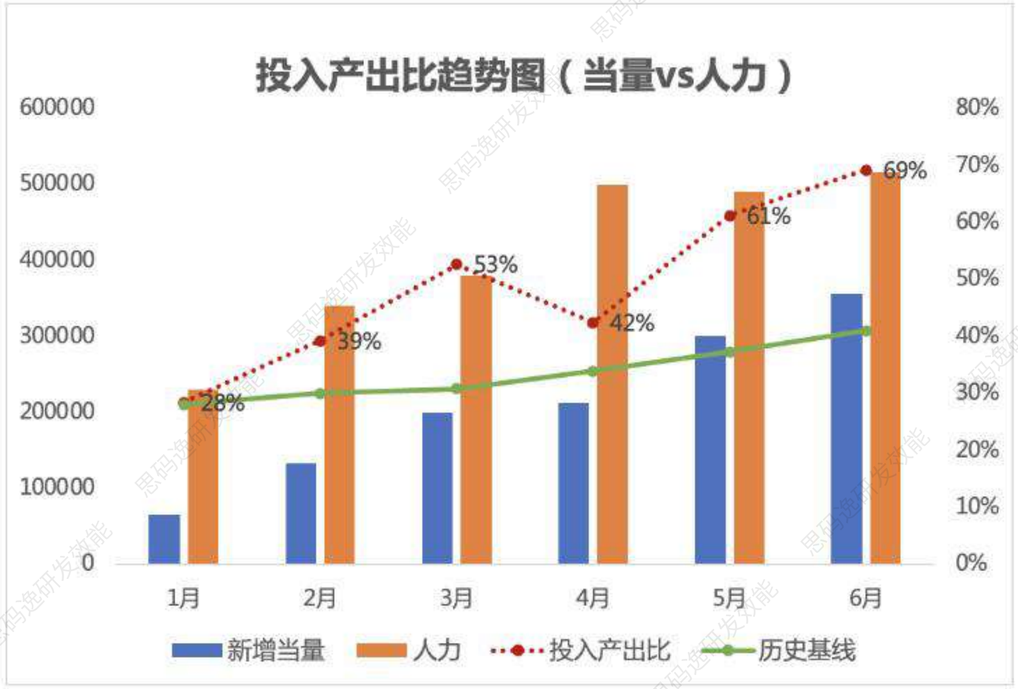
原型样例：工作饱和度

关键思考：

- 是否结合趋势、下钻、关联分析识别了**效能瓶颈**？

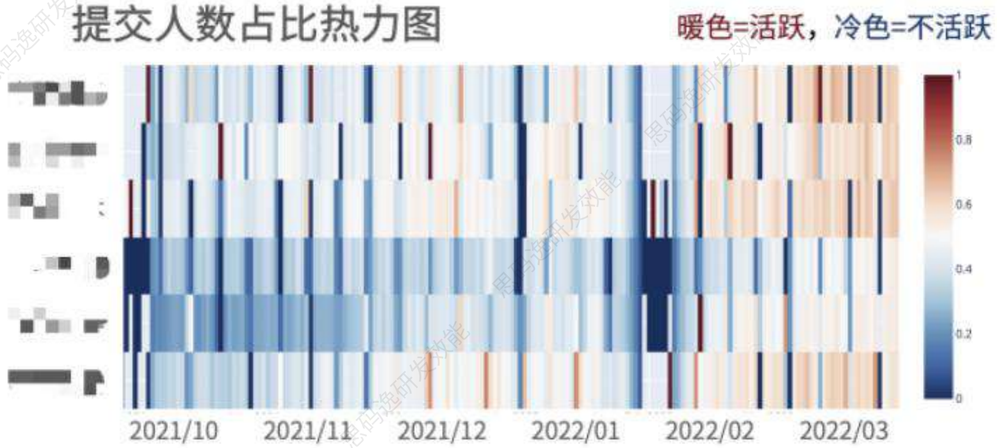
应用场景：衡量团队工作饱和度，为评估开发负载，及调配人力提供参考。

4月投入产出比显著下降



下钻分析

团队D、E工作饱和度偏低，空置率占60%



空置率	0.55	0.553	0.547	0.541	0.697	0.655
溢出率	0.015	0.015	0.021	0.018	0.016	0.015

图形来源：思码逸深度代码分析平台

merico

原型样例：效率

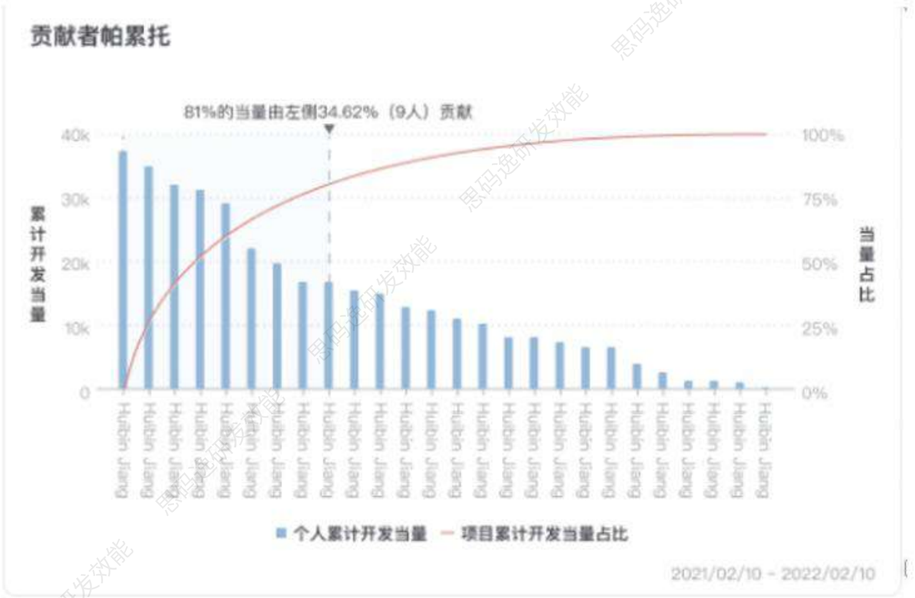
应用场景：了解团队的开发效率，为评估项目人力及交期提供参考。

人均生产率较去年同比提升37%，环比提升12%

关联分析

代码贡献依赖少数成员

80%的贡献来自20%的开发者



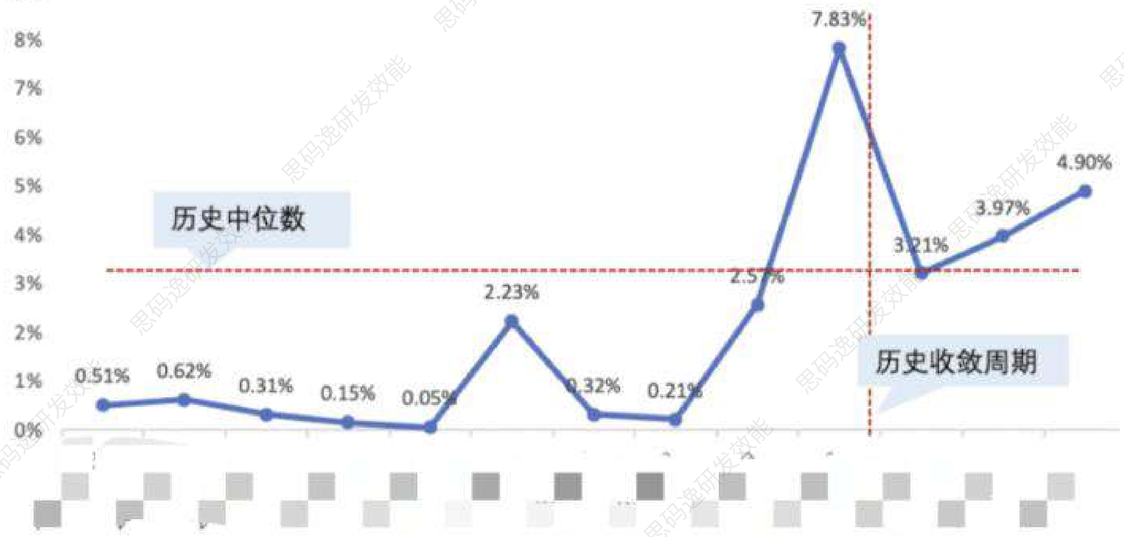
图形来源：思码逸深度代码分析平台

原型样例：质量

应用场景：了解代码发版质量，为评估版本的稳定性提供参考。

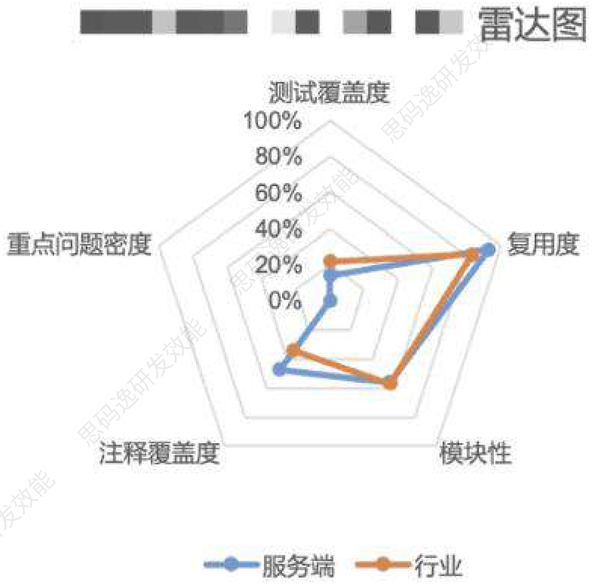
内测/线上 发版质量不稳定，千当量缺陷率异常偏高

内测缺陷率趋势图



关联分析

单元测试覆盖率低于行业指标



图形来源：思码逸深度代码分析平台

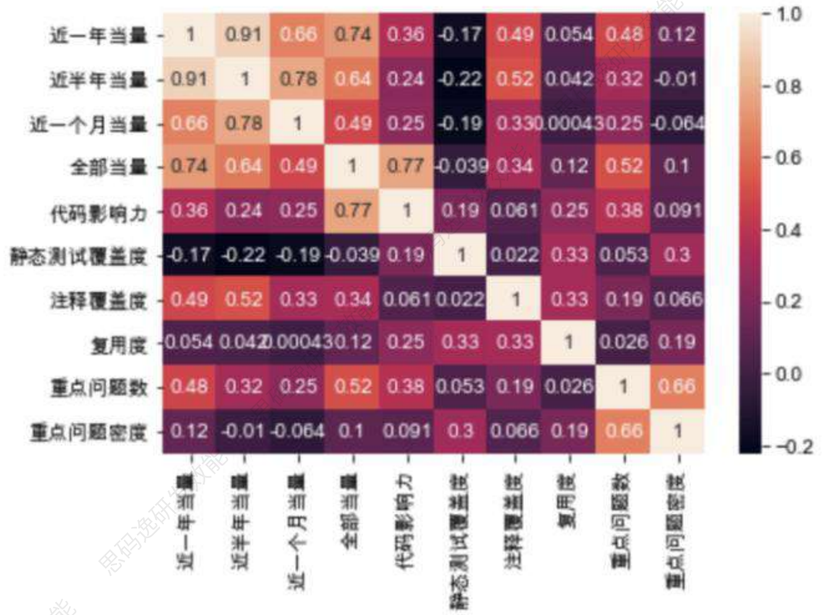
原型样例：质量

应用场景：通过分析指标间的关联关系，找到过程改进的切入点。

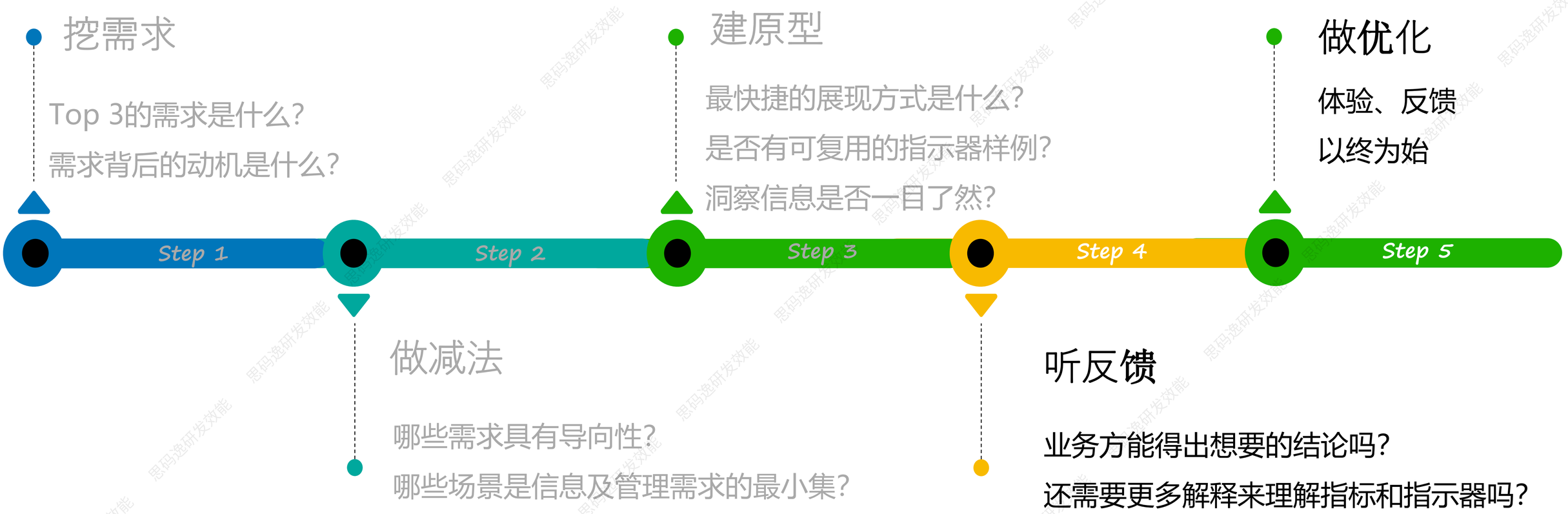
内测/线上 发版质量不稳定，千当量缺陷率异常偏高

关联分析

单元测试与产能相关性高



MVP构建度量的5步法



持续沟通

01 研发效能度量有哪些坑点？

02 如何构建效能度量的MVP？

03 MVP方法应用实例及原则

MVP方法使用中的注意事项

- **目标驱动：**

要倾听需求，但不要完全照着做（了解背后的动机是什么？）

- **注重特性：**

以终为始，关注价值，而非提需求的人。

- **少而精：**

不要以偏概全，那些不懂数据的人很糟糕，而最最糟糕的人是那些只看数字的人。

- **建立标尺：**

度量数据本身不会骗人，但数据的呈现和解读却有很大的空间可以利用。

在 DevData Talks
开放务实地聊聊研发效能

分享合辑

研效交流群

每月直播 干货满满

行业 KOL + 企业研发效能负责人 + 资深效能顾问
从效能建设路径到不同场景应用，分享前沿观点与先进实践

开放探讨 共同成长

直播 Q&A + 交流社群
直播嘉宾面对面答疑解惑，交流群随时随地探讨效能话题

关于思码逸

作为DevData Talks 的发起方，思码逸专注为企业研发团队提供效能数据分析，呈现效率、质量、人才发展等多视角数据洞察，辅助团队决策与工程改进。

思码逸已服务 腾讯、美团、滴滴出行、中国平安、泰康保险、戴尔EMC等众多行业标杆客户。