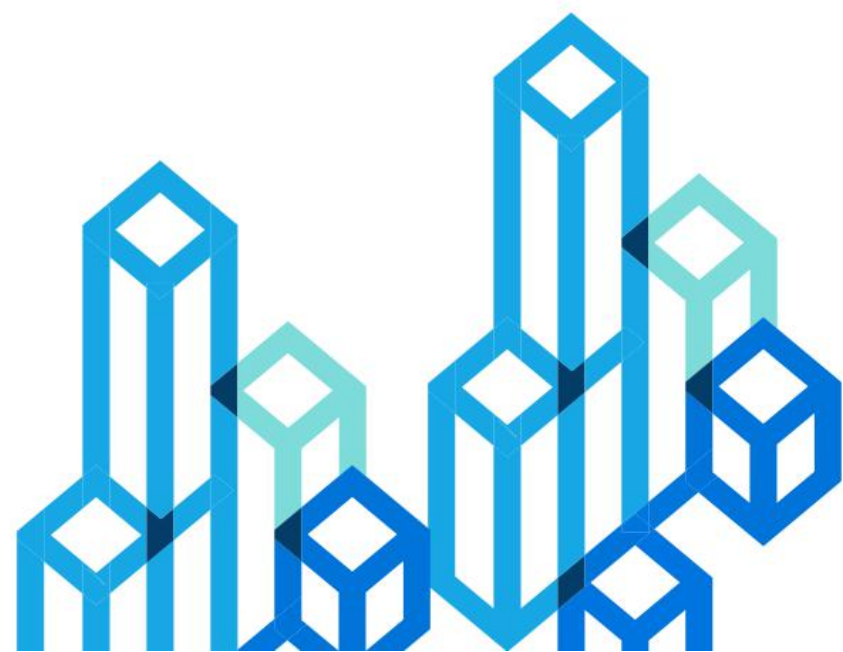
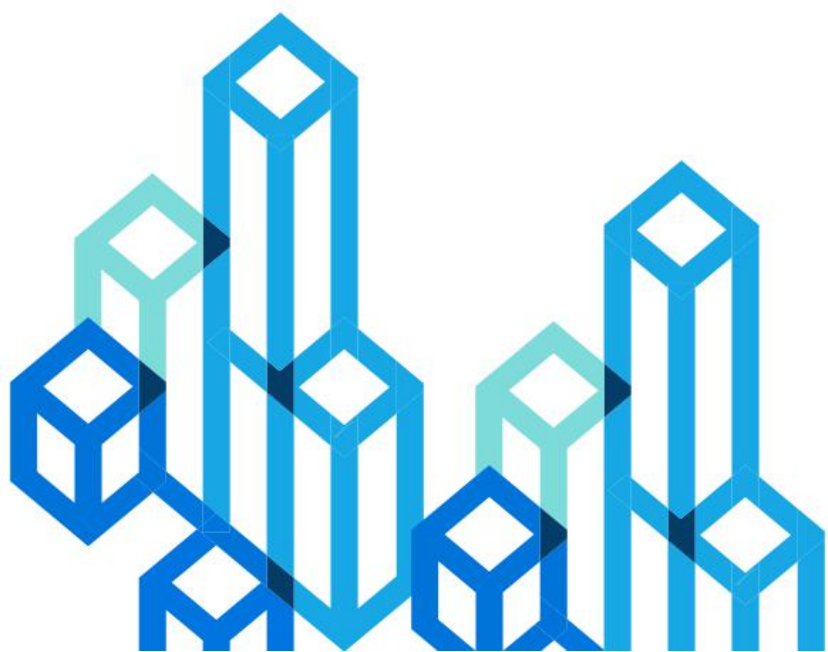


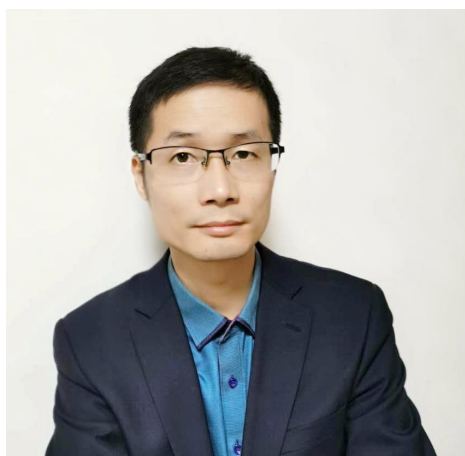
# 云原生时代的研发效能

黄国峰

前腾讯 工程效能专家



# 个人简介



## 黄国峰

前腾讯 工程效能专家

10 多年的软件和互联网从业经验；现任腾讯工程效能部，负责持续集成、研发流程和构建系统等平台；曾任职唯品会高级经理，负责架构团队。在云原生平台下的研发效能方向有丰富的理论知识和实践经验。

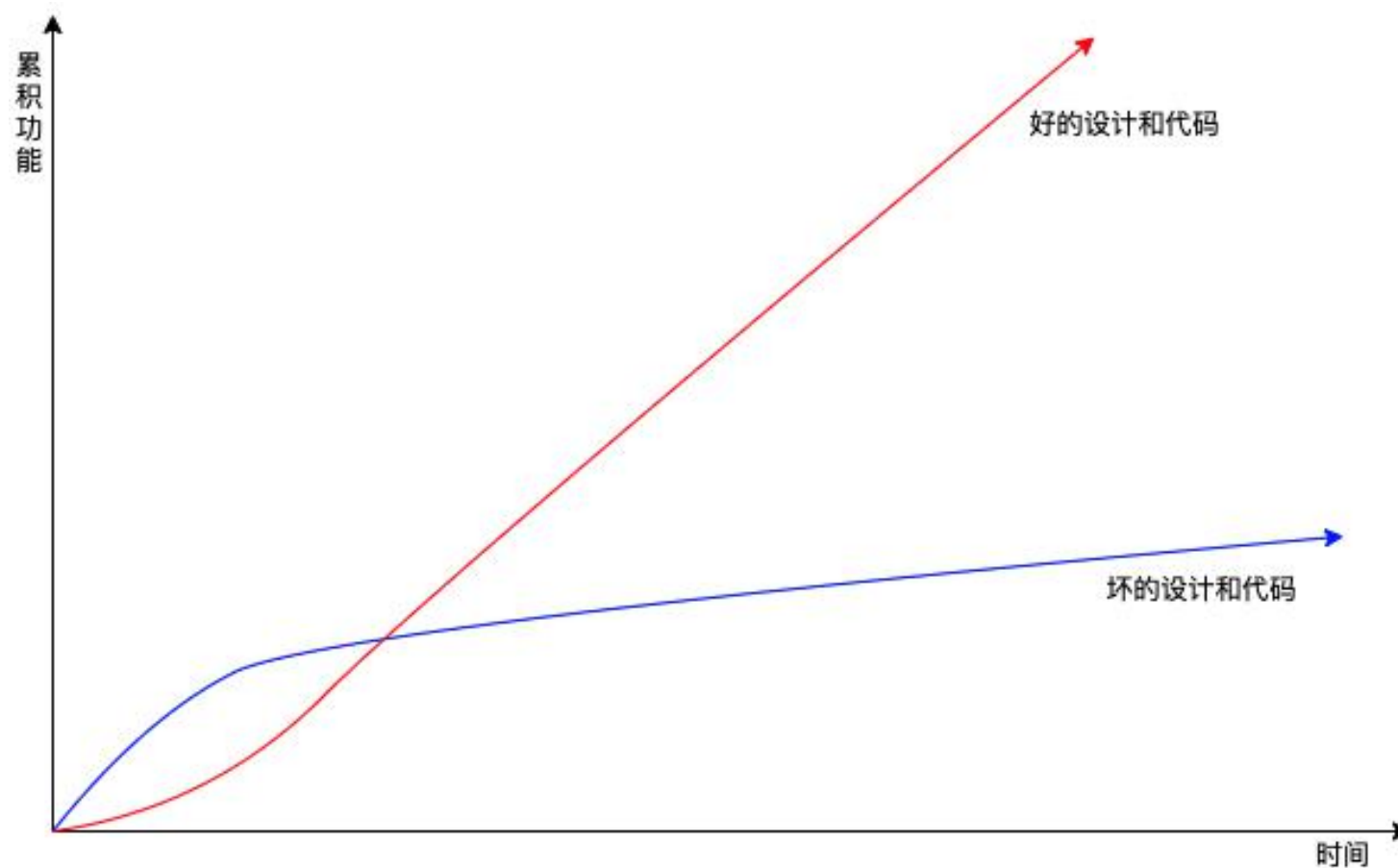
# 目录

1. 研发效能的定义
2. 研发效能提升之路
3. 我们的实践
4. 挑战与解决
5. 研发效能的未来方向

# 研发效能 — 定义与要素

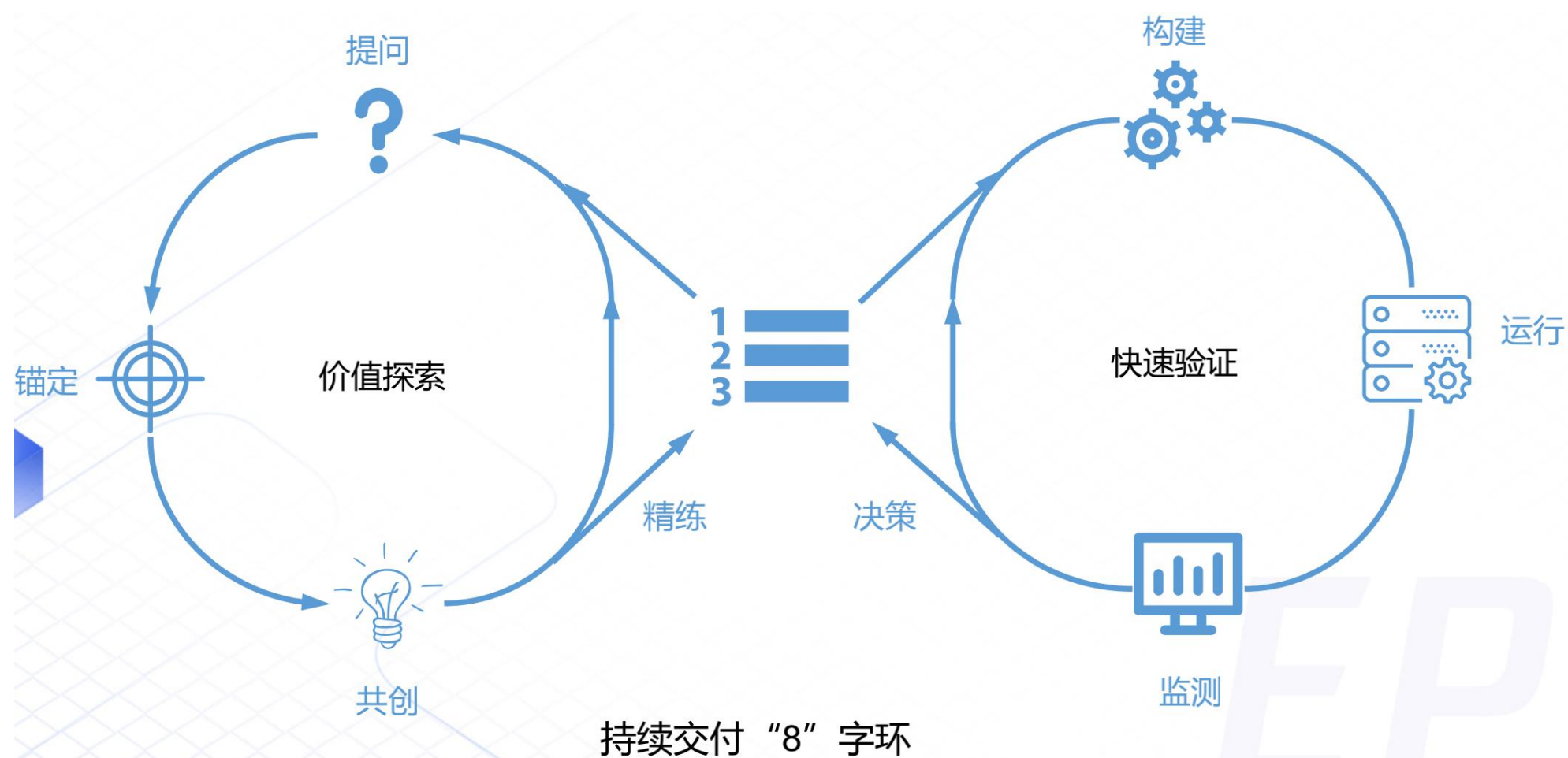
1. 研发效能的定义  
—— 能够持续地为用户产生有价值的效率
2. 研发效能的要素
  - 有效性 (Effectiveness)
  - 效率 (Efficiency)
  - 可持续性 (Sustainability)

# 研发效能 — 为什么需要？



软件累积功能与时间的比例

# 研发效能提升之路 —— 持续交付



# 我们的实践

## 1. 统一团队认知

- 技术基础：人力情况和技术能力
- 时间投入：业务松紧、迭代周期
- 轻重缓急：业务技术难点、优先保障核心模块健康
- 架构选型：基建模块及依赖服务

## 2. 多管齐下

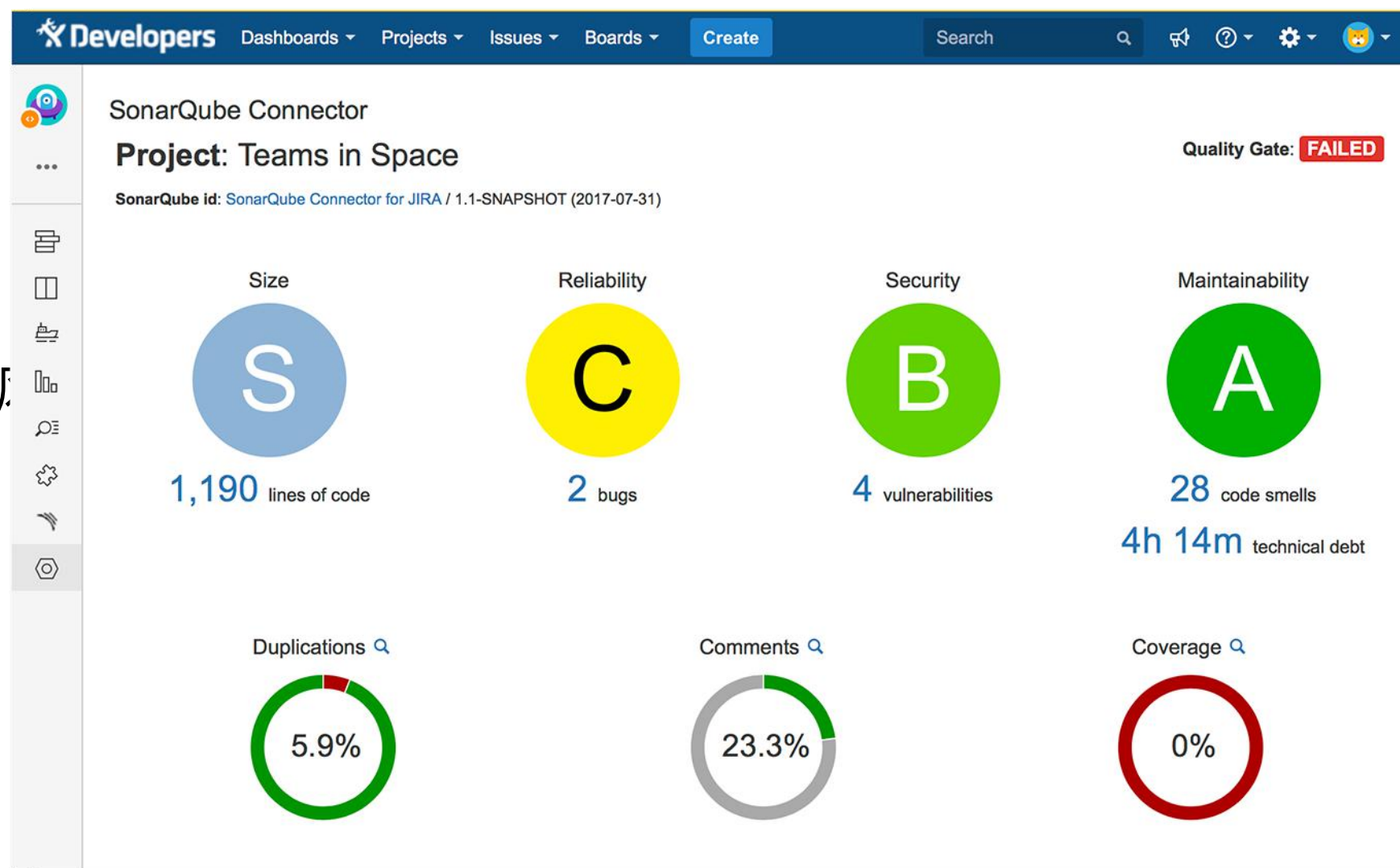
- 代码质量
- 分支模式
- CI/CD（持续集成/持续部署）
- 自动化测试
- 单体代码仓库

## 3. 规范执行的3个关键

- 流程卡点
- 依赖工具，而不是人
- 定期 Review 规范、工具、执行和数据

# 我们的实践 - 代码规范

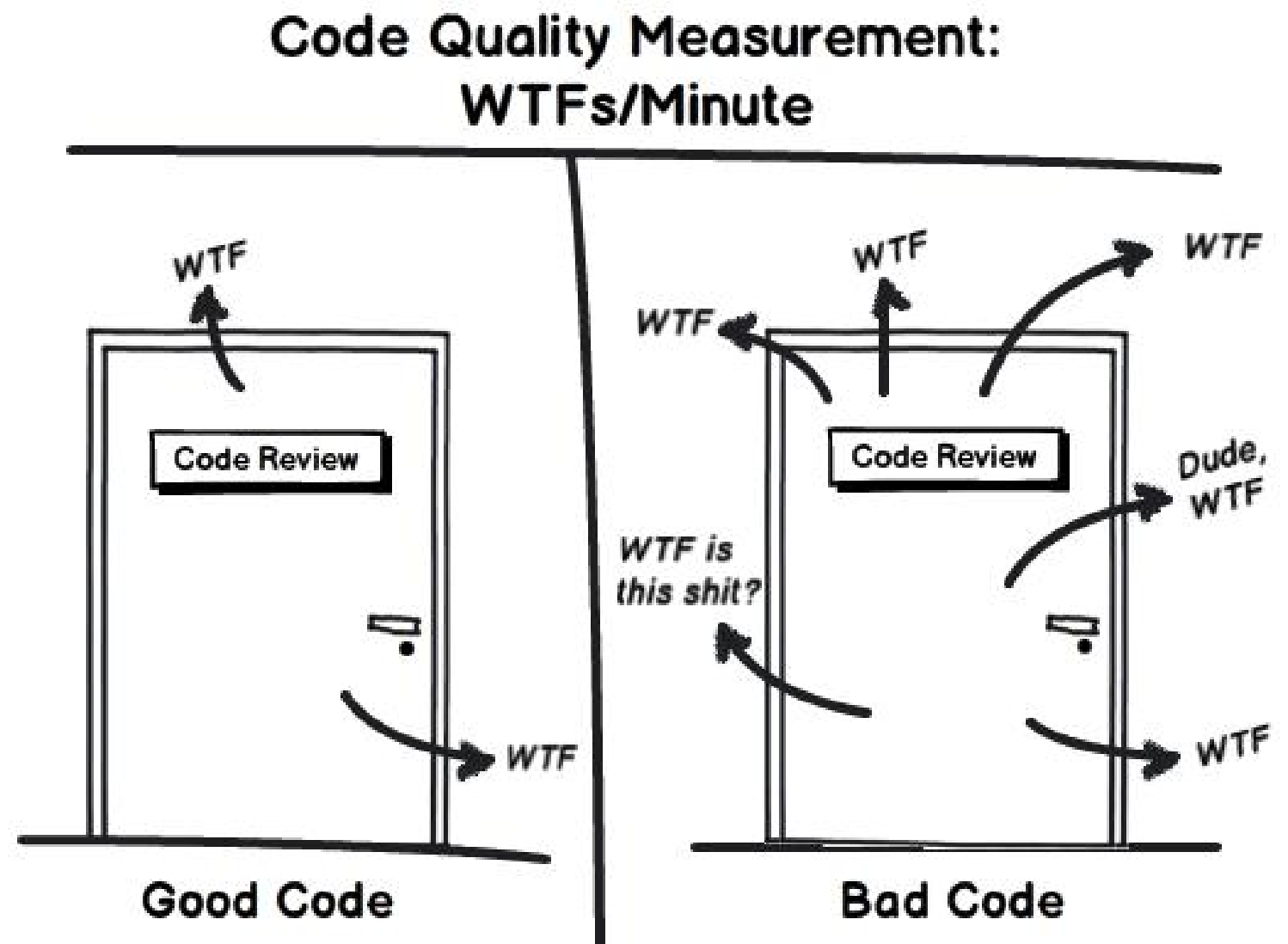
1. 代码缺陷
2. 安全漏洞
3. 技术债务
4. 帮助团队养成良好代码风格





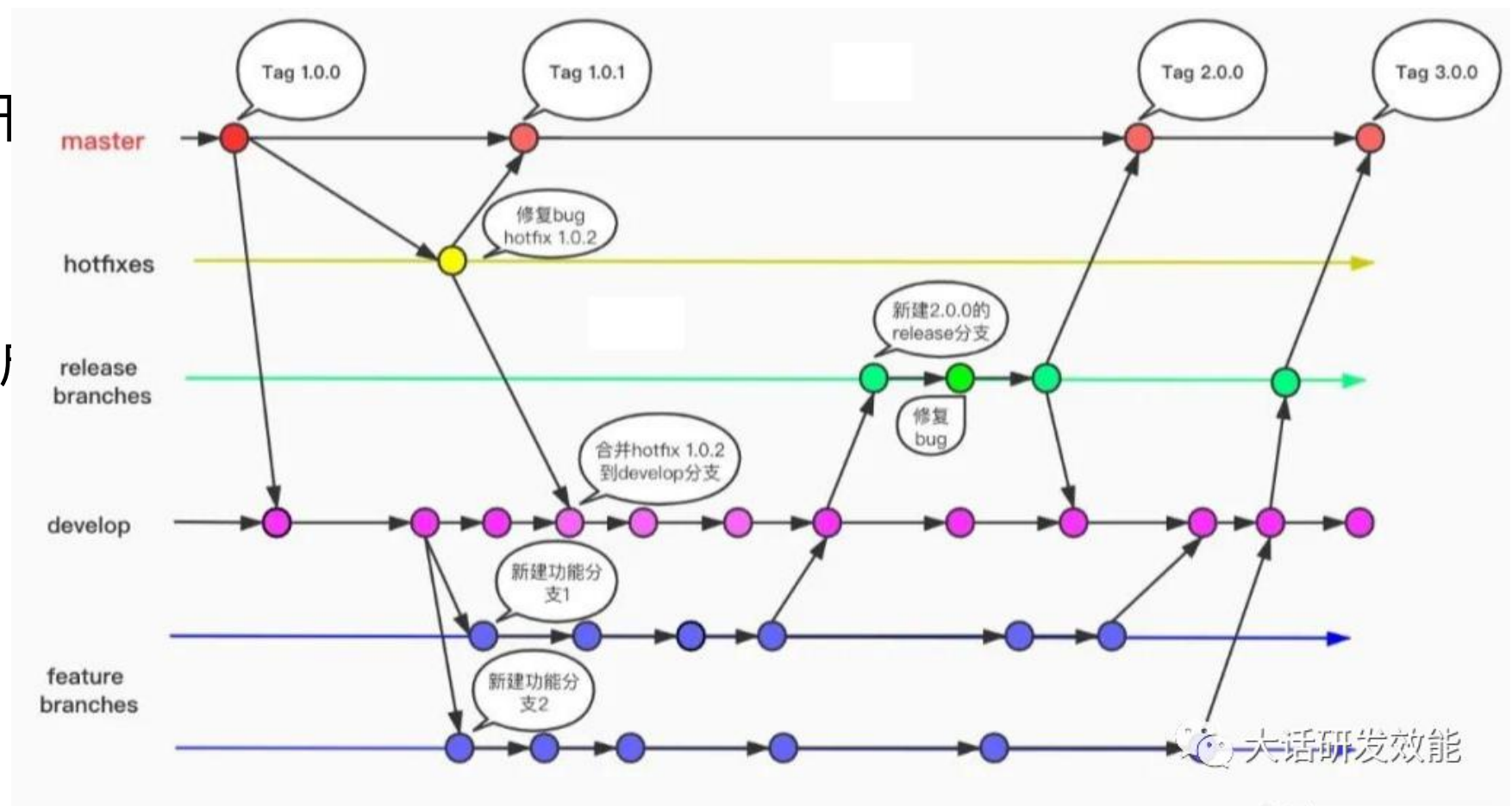
# 我们的实践 - 代码评审

- 提升代码质量 vs. 互助传承?
- 由谁来做?
- 怎么做?
- 好的代码评审是怎样的?



# 我们的实践 - 分支模式

- 特性分支开发和主干开
- 主干开发有什么好处？
- 主干开发有什么缺点/



# 我们的实践 — CI/CD

- 持续集成是什么？
- 持续部署是什么？
- CI/CD 是连接研发效能中众多实践的链条！

## 持续集成

- ☒ 代码库管理
- ☒ 代码编译
- ☒ 质量验证
- ☒ 自动化测试
- ☒ Code Review
- ☐ 漏洞扫描
- ☐ 代码混淆
- ☐ 代码加固

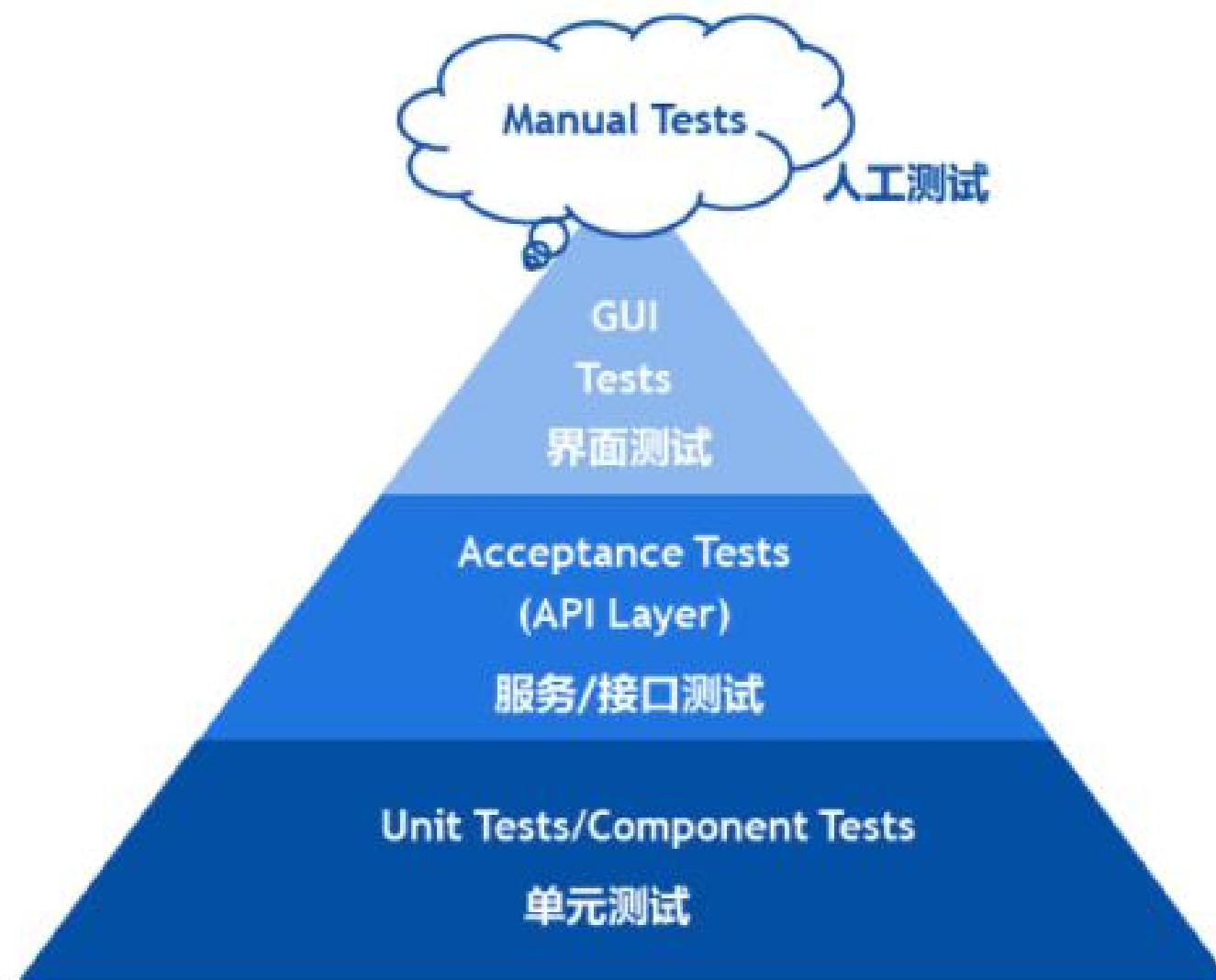
开发人员

## 持续部署

- ☒ 部署生产环境
- ☒ 用户验收测试
- ☒ 配置管理
- ☒ 软件包版本管理
- ☒ 容器管理
- ☒ 数据版本管理
- ☐ 包依赖管理
- ☐ 特性开关
- ☐ 灰度发布
- ☐ 金丝雀发布

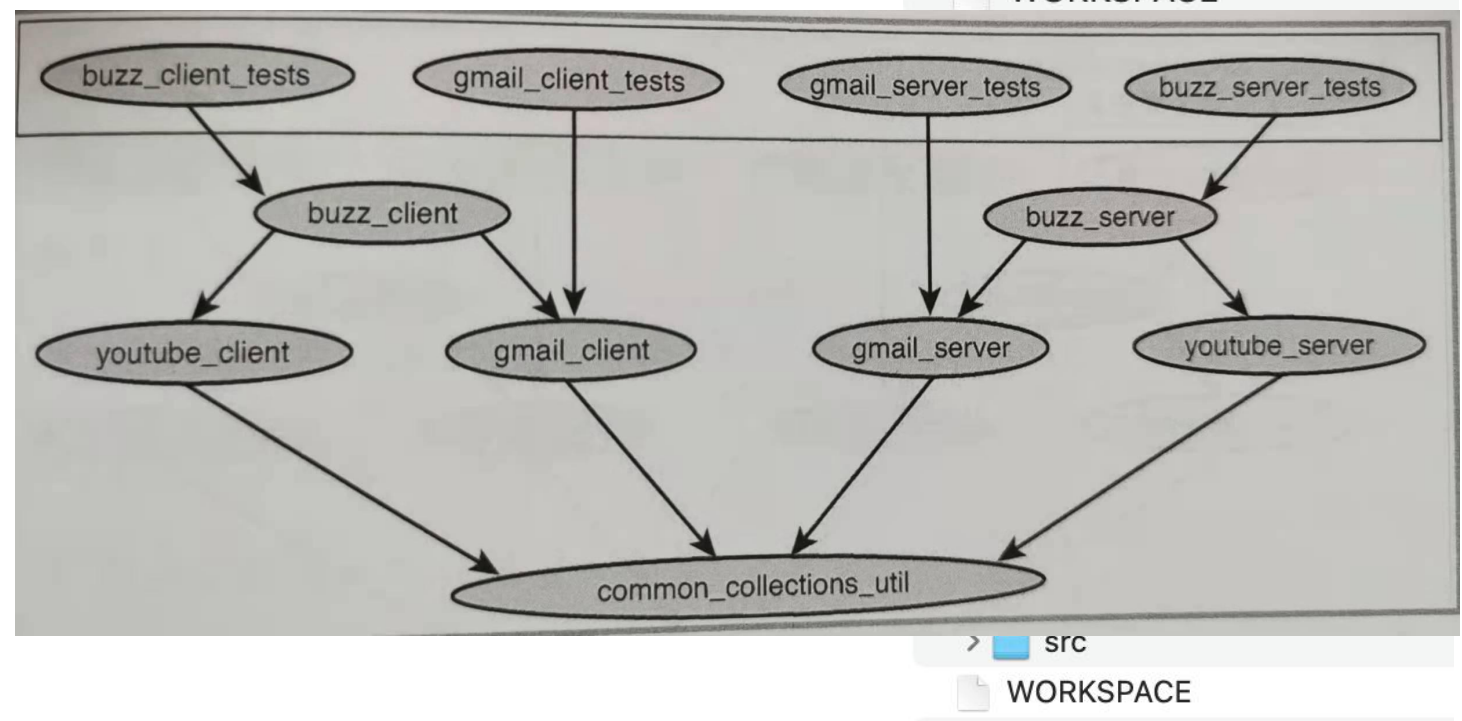
# 我们的实践 - 自动化测试

- 单元测试（小型测试）
- 服务/接口测试（中型测试）
- 自动化界面测试（大型测试）
- 人工测试



# 我们的实践 - 单体代码仓库

- 什么叫单体代码仓库 (Monolithic Repository) ?
- 有什么好处或优势?
- 有什么缺点或成本?



# 挑战与解决

1. 如何权衡业务诉求和技术基建?
  - 定目标
  - 圈资源
  - 数据驱动
2. 如何调动团队参与，提高效率?
  - 自动化
  - 给甜头
  - 树标杆
3. 如何解决历史遗留问题?
  - 质量红线
  - SSH (Shit Stop Here) —— 增量指标

# 研发效能的未来方向

- 快速迭代
- 质量左移
- 质量右移

Question?





麦思博(msup)有限公司是一家面向技术型企业的培训咨询机构，携手2000余位中外客座导师，服务于技术团队的能力提升、软件工程效能和产品创新迭代，超过3000余家企业续约学习，是科技领域占有率第1的客座导师品牌，msup以整合全球领先经验实践为己任，为中国产业快速发展提供智库。



高可用架构主要关注互联网架构及高可用、可扩展及高性能领域的知识传播。订阅用户覆盖主流互联网及软件领域系统架构技术从业人员。高可用架构系列社群是一个社区组织，其精神是“分享+交流”，提倡社区的人人参与，同时从社区获得高质量的内容。