

研发数字化的四个层次 和效能度量的五项精进

张乐
2022/01



个人简介

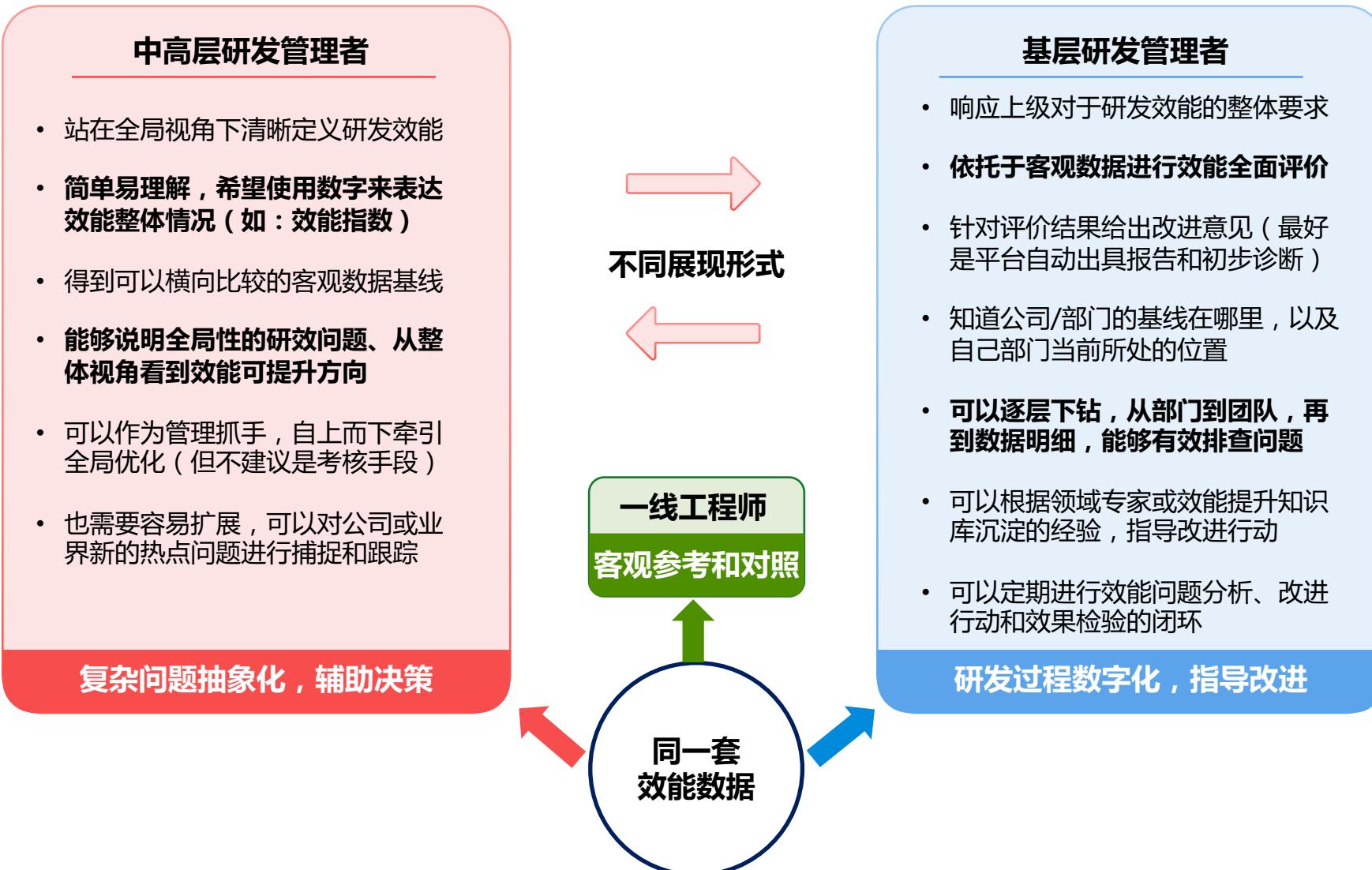


张乐

- 腾讯 技术工程事业群 DevOps与研发效能资深技术专家
- 多年一线互联网大厂工程效率领域工作经历（百度、京东等）
- 历任埃森哲、惠普等全球五百强企业资深咨询顾问、技术专家
- 长期工作在一线，专注于研发效能提升、敏捷与DevOps实践落地、万人研发规模 DevOps 平台设计、研发效能度量体系建设等方向
- DevOps 起源国际组织 DevOpsDays 中国区核心组织者
- 多个国内技术大会联席主席、DevOps/ 研发效能专题出品人
- EXIN DevOps 全系列国际认证授权讲师、凤凰项目沙盘授权教练
- 译著《独角兽项目：数字化时代的开发传奇》，《DevOps最佳实践》
- 即将出版：《软件研发效能提升实践》、《软件研发效能权威指南》

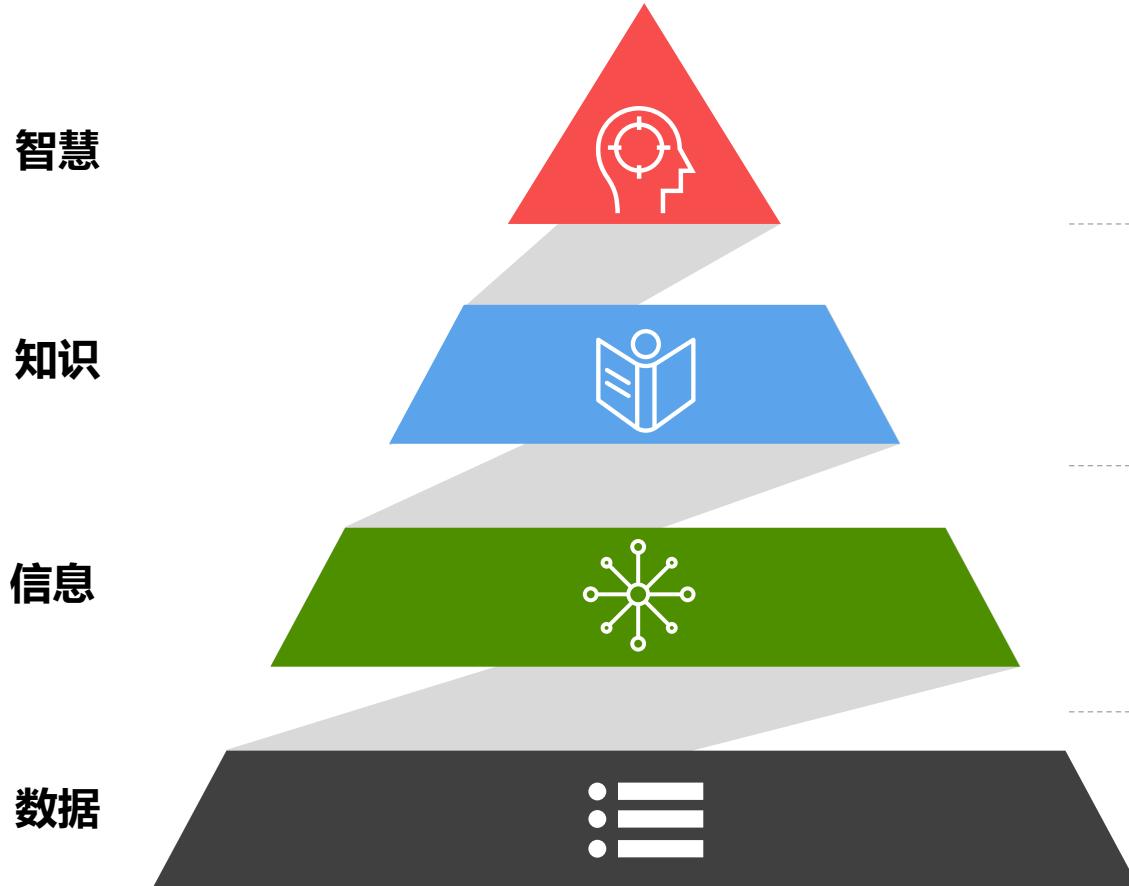


研发效能度量的不同关注点



研发数字化的四个层次

数字化就是从物理世界中，开采出数据，粗炼出信息，精炼出知识，聚合出智慧，最终提高生产率。



把知识聚合成智慧

- 聚合更多知识
- 推荐解决方案

研发效能优化方案

- 工程效率优化方案...
- 敏捷协作优化方案...

把信息精炼出知识

- 整合多种信息
- 进行分析和判断

研发效能分析

- 工程质量分析...
- 团队交付流分析...

把数据粗练出信息

- 去掉无用信息
- 识别出有价值的信息

研发度量指标

- 代码圈复杂度...
- 需求交付周期...

从物理世界开采出数据

- 文字、图片
- 记录、日志

研发过程原始数据

- 代码提交日志...
- 需求流转记录...

研发效能度量的五项精进

05 数据驱动，实验思维

- **目标：让研发效能可量化、可分析、可提升**
- 数据不是武器（古德哈特定律），而是学习和持续改进的工具；度量的是工作而不是工作者
- 用真实故事核实结论，结合现场观察(Gemba)

03 度量分析模型

- **分析原则：趋势>绝对值，分析相关性和因果**
- 模型建立：组织效能、团队效能、工程师效能
- 分析方法：趋势分析、相关性分析、诊断分析
- 分析场景：问题驱动模式、体检模式...

02 度量指标体系



业务结果

设计原则：结果指标>过程指标；全局指标>局部指标

- 指标用途：结果指标用于评估，过程指标指导分析改进
- 通过先导性指标事前干预，通过滞后性指标事后复盘

组织效能模型

研发进展
研发效能

资源分配
收益/满意度

价值流交付模型

交付周期
吞吐量
流分布

质量/安全
流效率
流负载

工程师效能模型

编程习惯
代码贡献

代码评审
代码质量

04 度量产品建设

- **将数据转化为信息，然后将信息转化为知识**
- 数据是透明和开放的，可以被整个组织访问
- 用户可以自助消费数据，主动进行分析和洞察

效能大盘

默认仪表盘
多维度聚合

自定义仪表盘
多条件筛选

报表订阅
多层次下钻

效能诊断

趋势分析
异常识别

对比分析
风险预测

相关性分析
分析结论

诊断订阅
改进建议

01 自动采集效能数据

- 系统分层：数据源、接入层、存储计算层、分析层
- 多种取数方式：API、MQ、JDBC、大数据平台
- 多种计算方式：实时/批量，流批一体

基于价值流的度量

需求

开发

测试

发布

运营

业务结果



+\$

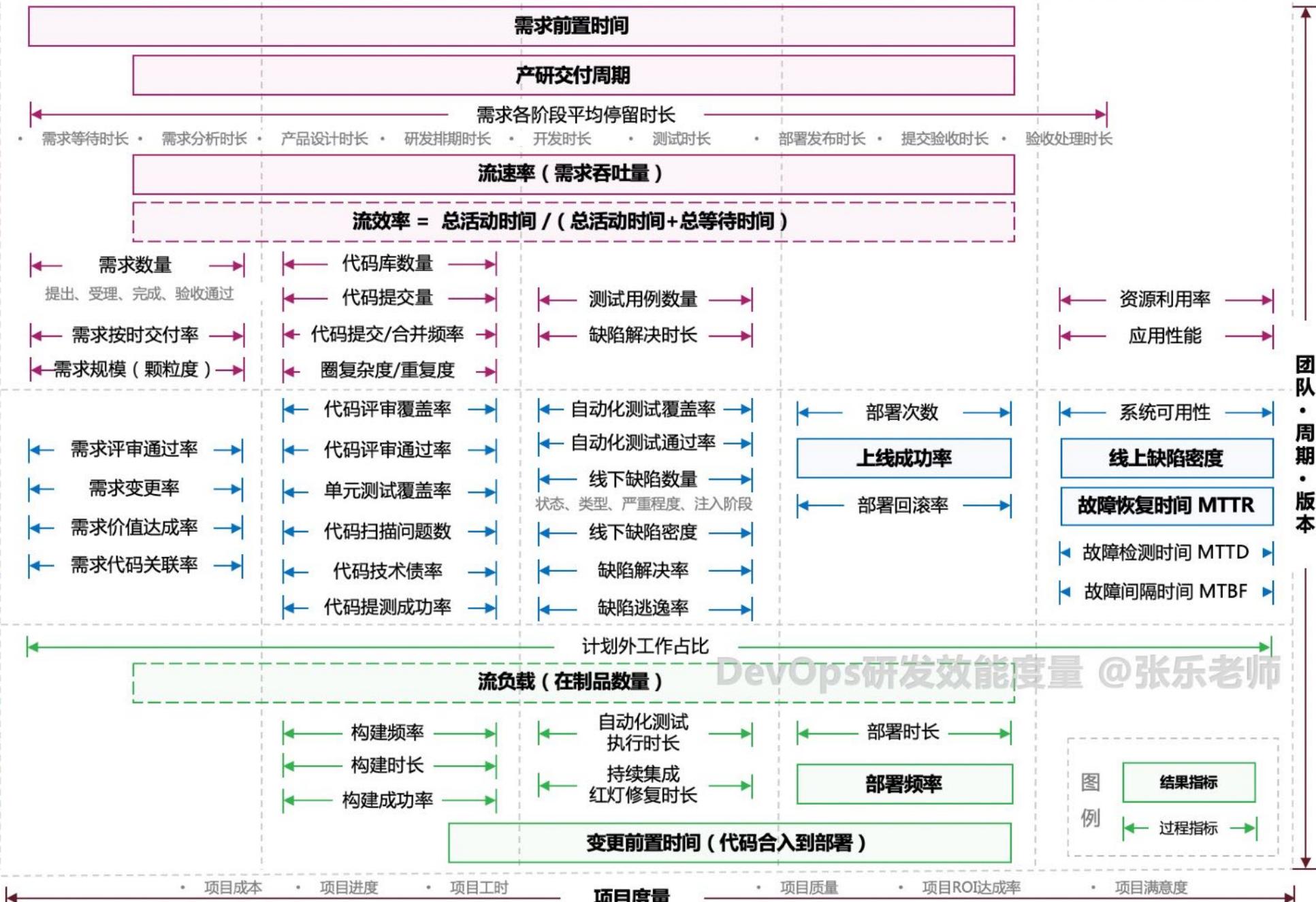
业务价值

-\$

交付成本

◎

客户满意度



度量指标举例：需求前置时间

定义 从需求提出，到完成开发、测试、上线的时间周期

价值 反映了业务和产研团队对客户问题或业务机会的交付(响应)速度，依赖整个组织各职能部门的协调一致和紧密协作

公式 需求前置时间=上线完成时间-需求提出时间

展现形式



健康度度量

> 需求前置时间小于0.1(天)的占比

如：某部门4月需求前置时间小于0.1天占比为8.6% (290/3374)，此类需求通常为补录，并非体现真正的交付周期，需要通过规范需求提交流程，降低占比

> 开发开始时间早于需求提出时间的占比

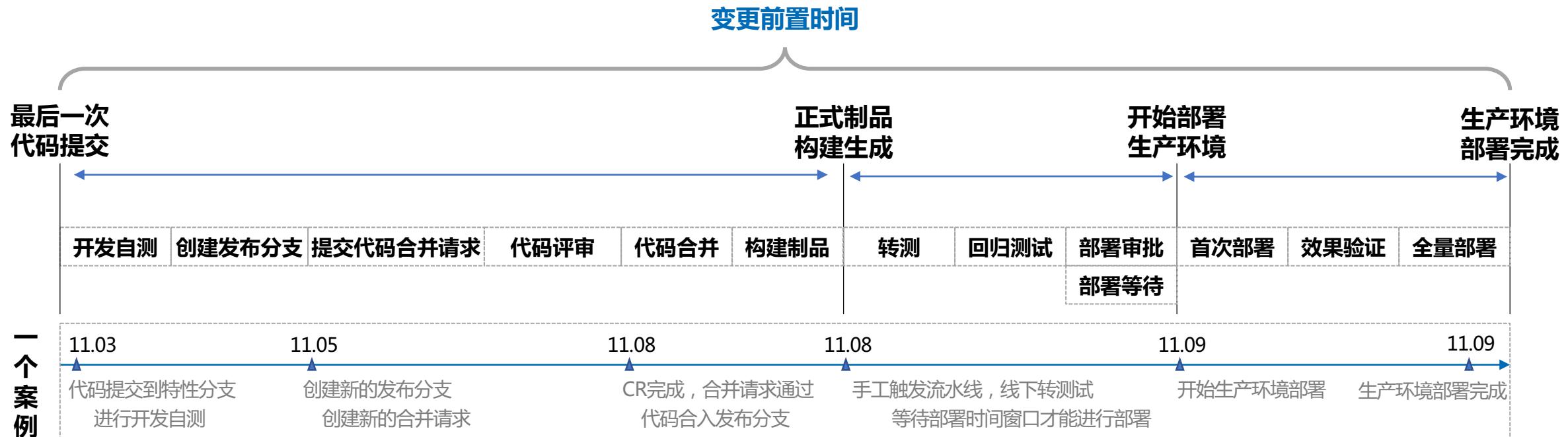
如：某部门4月开发开始时间早于需求提出时间占比为22.5% (751/3374)，此类需求通常为补录，并非体现真正的交付周期，需要通过规范需求提交流程，降低占比

度量指标举例：变更前置时间

定义 从代码提交到部署到生产环境的时间周期

价值 研发过程整体上可以分为不确定的创造性活动（如设计、写代码等），以及确定性、重复性的活动（如编译构建、代码扫描、自动化测试执行、环境部署等），后者特点是高频、重复、可自动化，是研发效能平台需要重点优化的部分。从代码写好，到投产使用之间的时间越短约好，越短则代表工程能力越强。

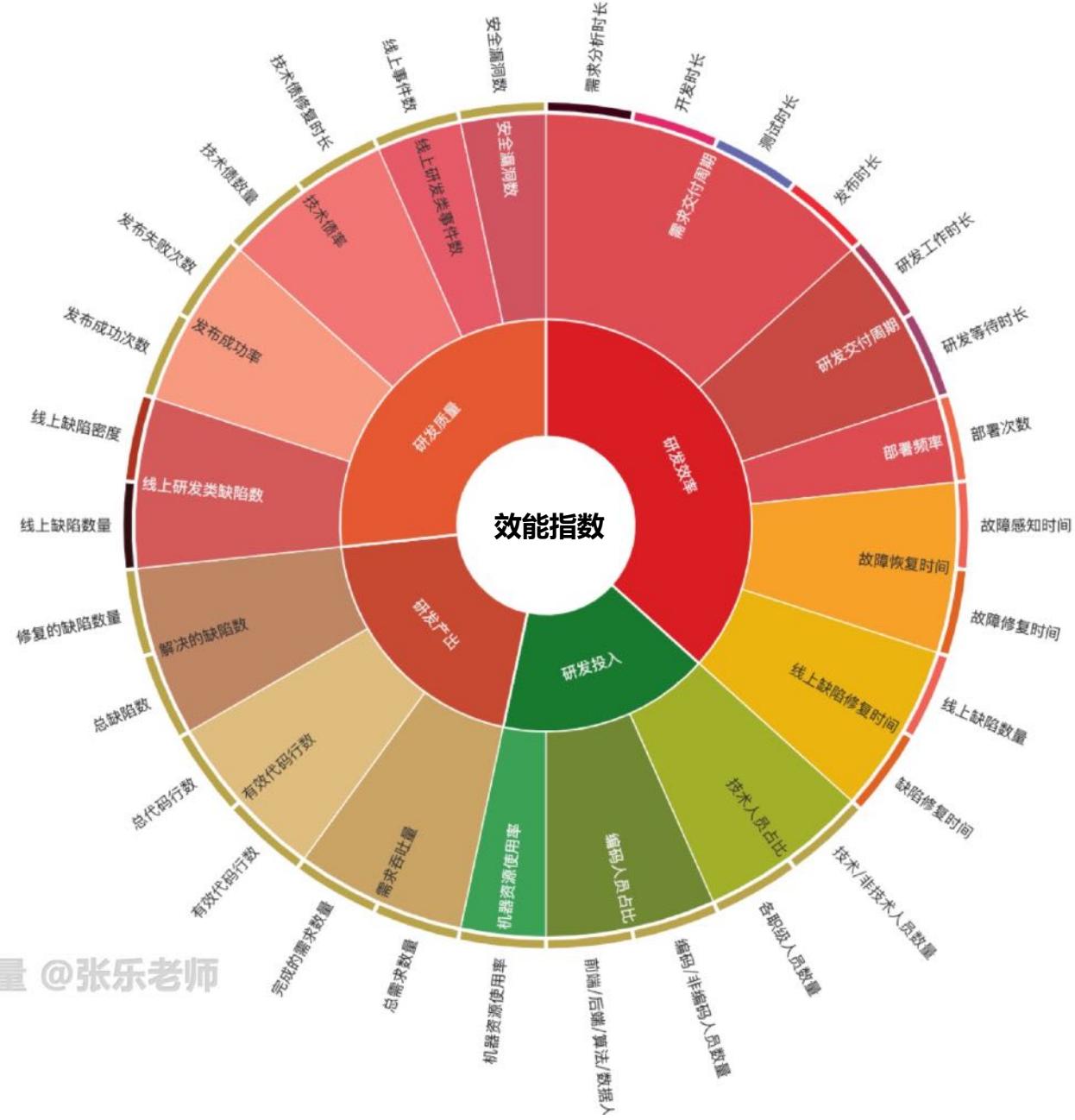
公式 从代码（最后一次）提交，到对应制品成功部署到生产环境的时间



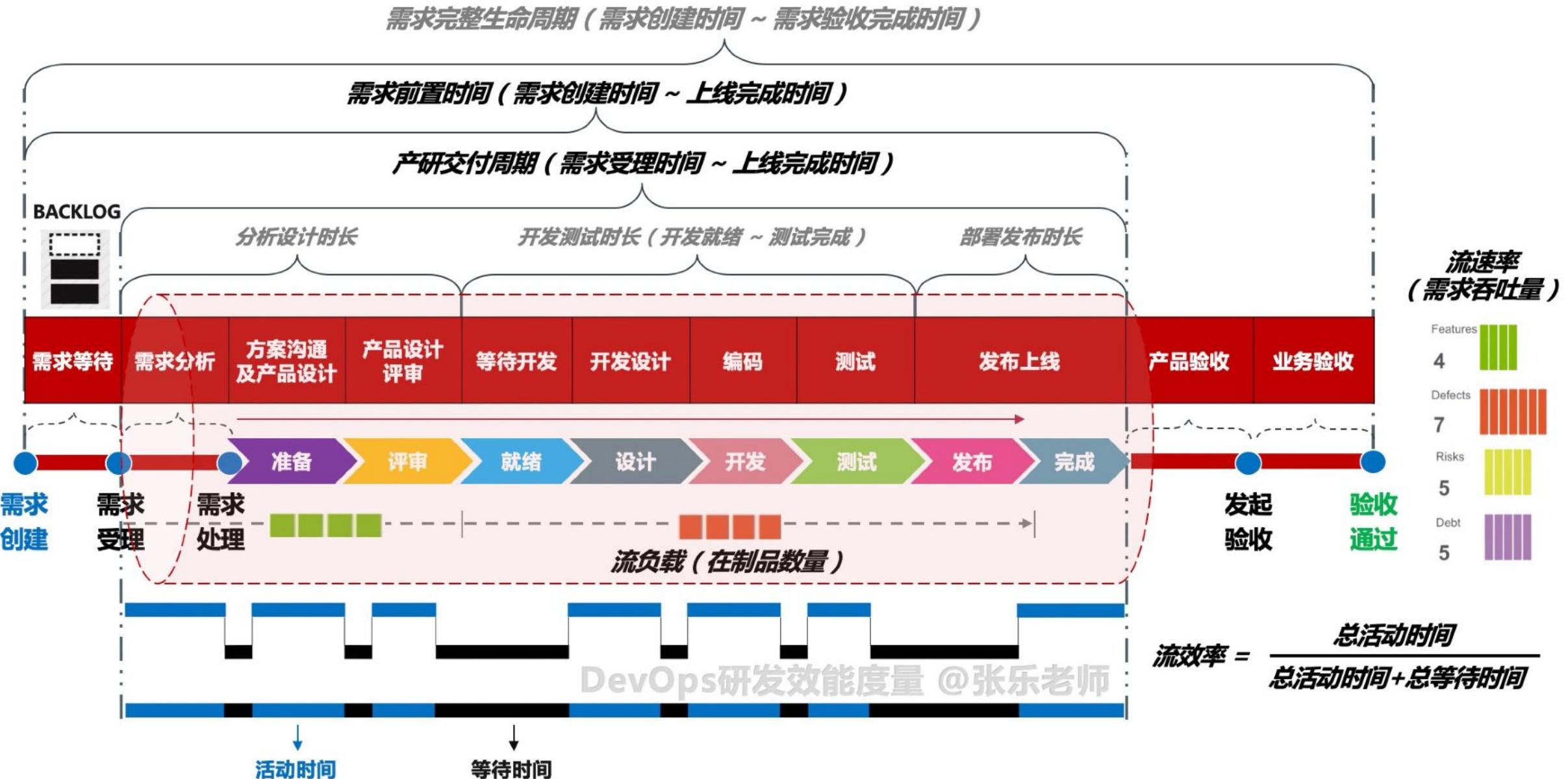
问题点分析

- 协同人数少的模块，分支模式可以简化
- 手工测试为主，缺少自动化验证
- CR请求没有被及时处理
- 流水线由手工触发
- 转测试过程线下，手工执行
- 等待发布窗口进行部署，产生等待浪费
- 生产部署过程复杂，需要人工编排实现

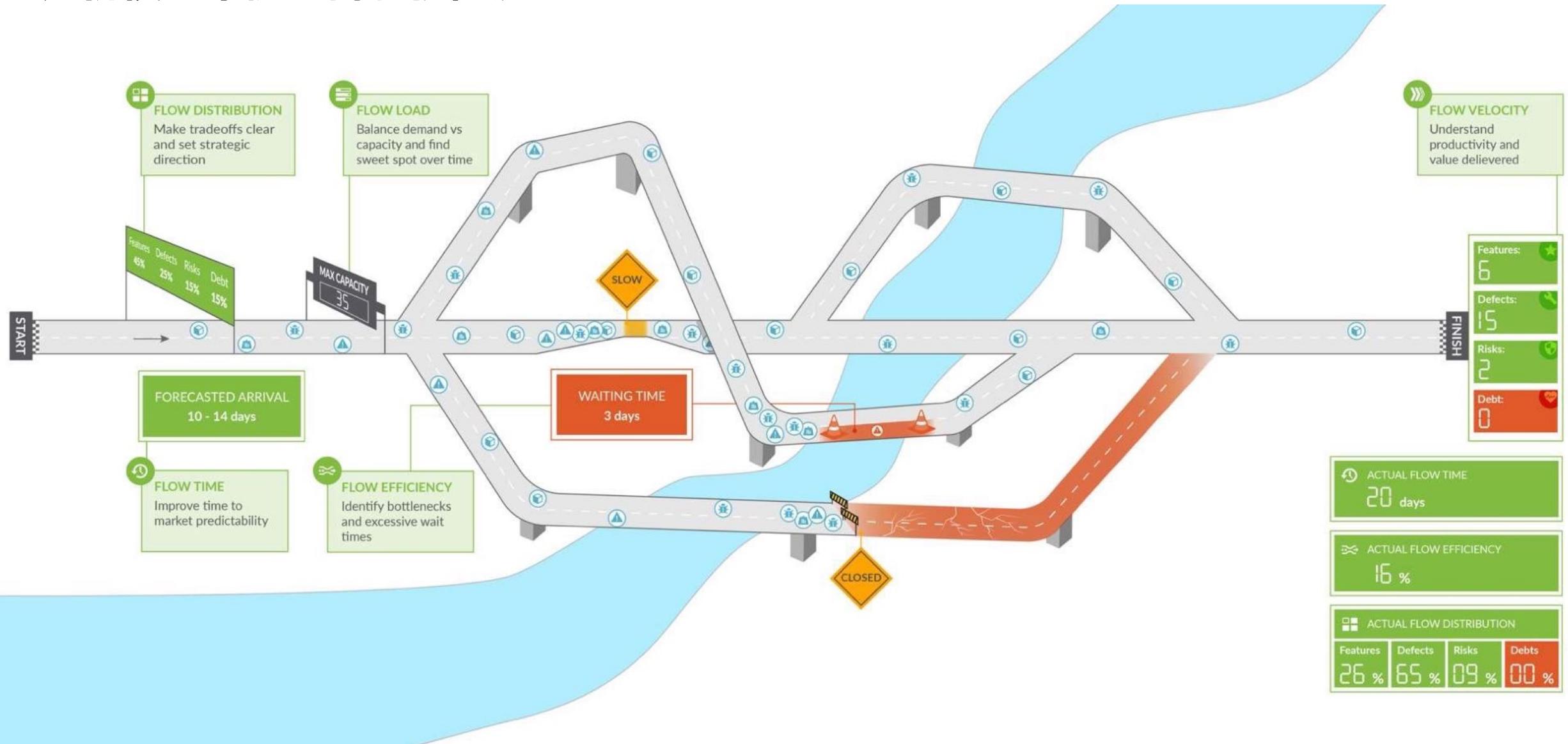
研发效能度量的分析模型



分析模型举例：需求价值流



分析模型举例：需求价值流



指标项: 产研实施周期 数据源: 需求及卡片完成 按需求完成时间

时间范围:

统计维度(X轴): 月趋势 需求来源: 全部 过滤团队空间

查询

返回上级部门

导出

卡片维度导出

研发部门视角

指标说明:

产研实施周期=卡片完成时间-需求受理时间, 需求受理时间及卡片完成时间取最晚的时间。单位(天)

受理部门视角

代码库单点分析

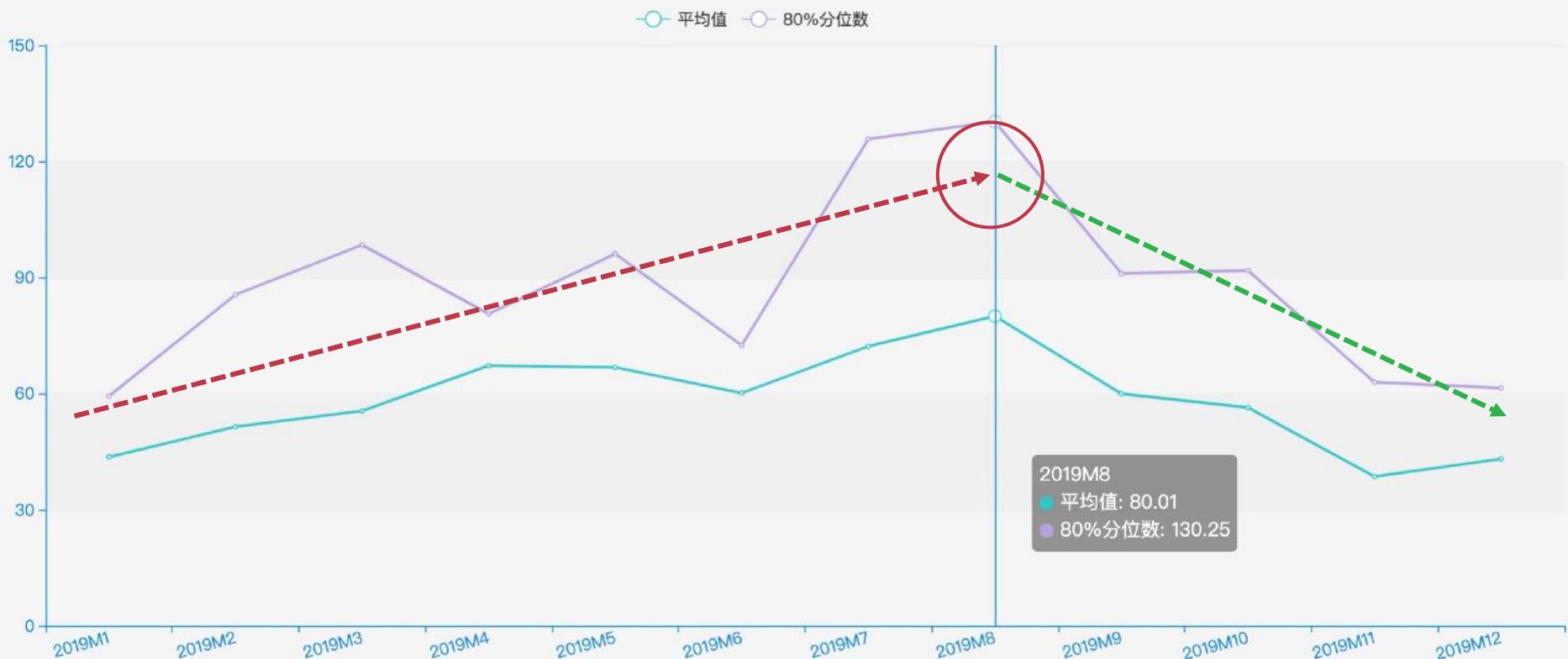
关联分析

全链路关联分析

问题诊断分析

设置

监控预警



首页概览



研发效能度量平台

DevOps产品研发部



全链路仪表盘

自定义仪表盘

单点分析



指标项: 实际开发周期
数据源: 卡片完成

按卡片完成时间

时间范围:



需求单点分析



统计维度(X轴): 阶段停留时长
需求来源: 全部
 过滤团队空间

查询

返回上级部门

导出

选择导出维度

研发部门视角

受理部门视角

代码库单点分析

关联分析



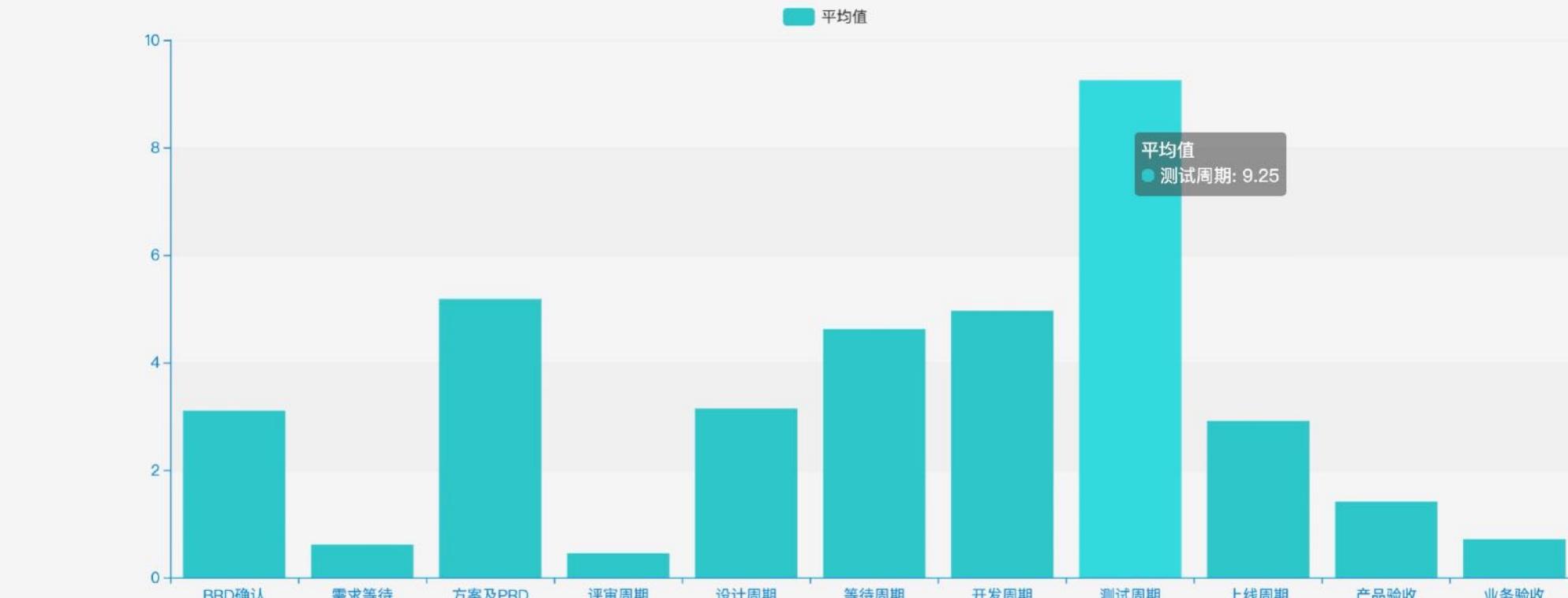
全链路关联分析

问题诊断分析

设置



监控预警





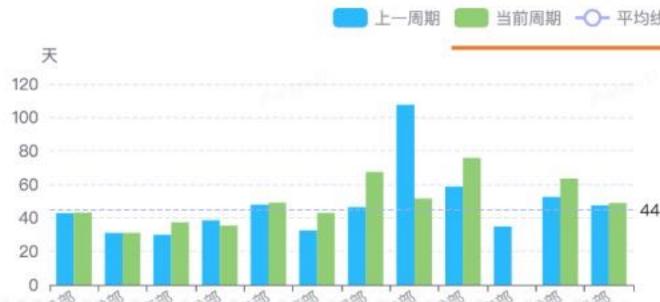
研发效能

研发进展

资源投入

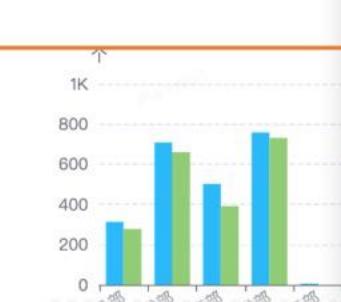
需求全周期

指标解读

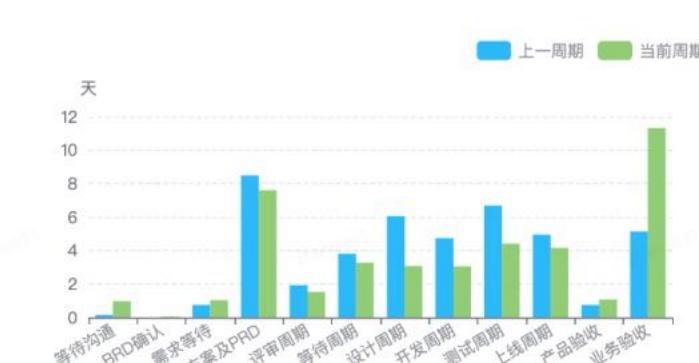


需求吞吐量

指标解读

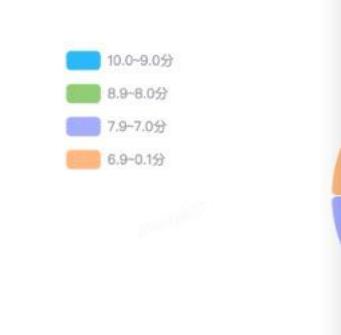


各阶段停留时长



项目满意度评分占比-执行方

指标解读



指标解读

同级部门对比

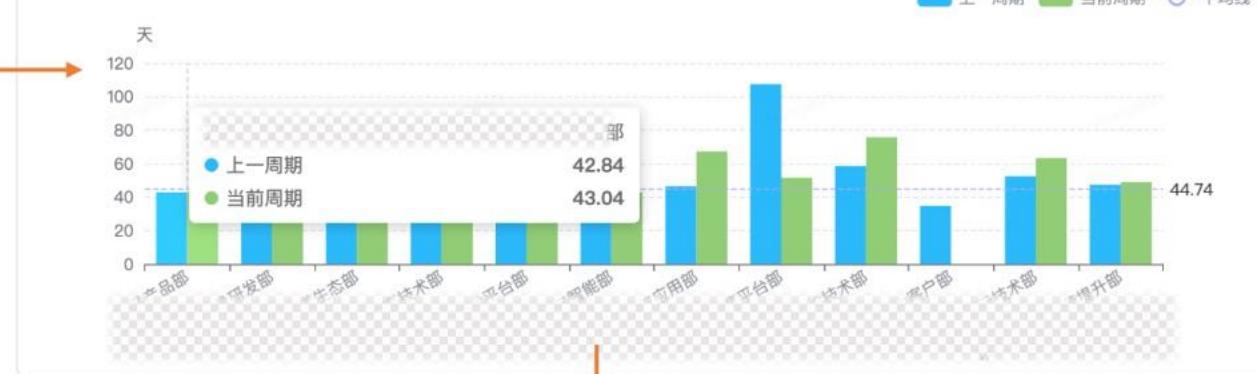
本部门趋势

指标下钻

需求全周期

?

指标解读



返回上一级

指标明细 (共332条)

需求编号	需求名称	需求全周期(天)	卡任务处理人	需求完成时间
R2021072112663	【日常需...	34.13	gao...	2021-08-24 20:25:44
R2021072114779	【VTP】...	15.87	xia...	2021-08-06 19:06:14
R2021072112994	变更【京...	11.67	sun...	2021-08-02 10:06:54
R2021072112905	日常需求...	19.01	liuz...	2021-08-09 18:08:04
R2021072536388	【VTP】...	12.31	xia...	2021-08-06 19:06:23
R2021072536390	【VTP】...	12.31	xia...	2021-08-06 19:06:21
R2021072536371	【VTP】...	12.32	xia...	2021-08-06 19:06:08

流效率分析

按阶段下钻分析计算团队流效率，寻找低效瓶颈点

数据洞察

- 通过5个团队空间卡片状态的统一配置，明确各状态的准入条件，并规范团队操作。从而能够准确度量团队流效率。
- 等待时长主要集中在需求就绪和待测试阶段，为低效瓶颈点。

部门结果指标

团队结果指标

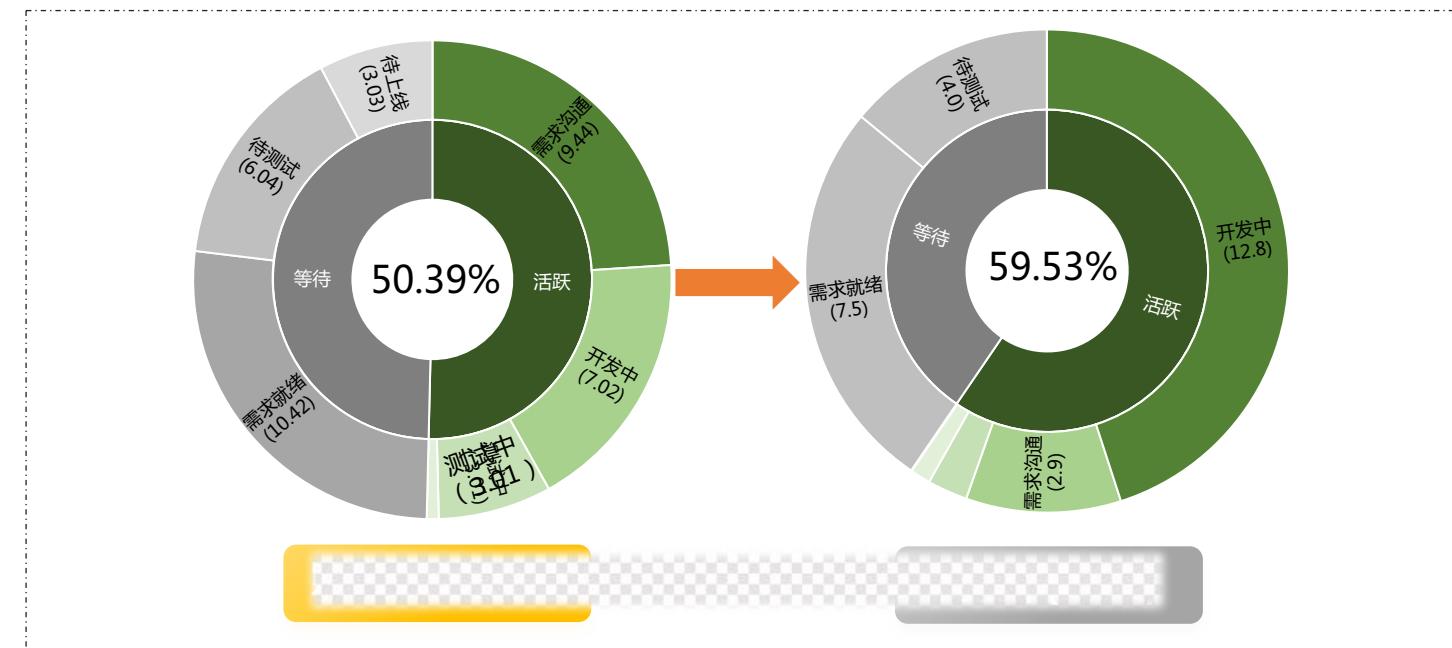
团队过程指标

团队最佳实践

团队空间统一配置														
阶段	准备	准备	评审	评审	就绪	就绪	设计	开发	测试	测试	发布	发布	完成	挂起
状态	需求沟通	待预审	需求待评审	配置类需求排期专用	需求就绪	需求已排期	技术方案	开发中	待测试	测试中	待上线	发布中	完成	挂起
属性	活跃	等待	等待	等待	等待	等待	活跃	活跃	等待	活跃	等待	活跃	等待	等待

下一步举措

- 通过限制在制品数量和小批量交付的最佳实践，缩短“就绪时间”的等待时长，进而提升团队效率。
- 通过优化测试排期或提升自动化测试比例，缩短待测试时长。



迭代	需求沟通	技术方案	开发中	测试中	发布中	待预审	需求待评审	配置需求排期专用	需求就绪	需求已排期	待测试	待上线	平均活跃	平均等待
V4	9.44	0.00	7.02	3.01	0.32	0.00	0.00	0.00	10.42	0.00	6.04	3.03	19.8	19.5
V6	2.95	0.76	12.78	0.38	0.02	0.00	0.00	0.00	7.52	0.00	3.96	0.00	16.9	11.5
对比	-68.79%		82.04%	-87.35%	-93.42%				-27.80%		-34.38%	-100.00%	-14.69%	-41.07%

需求交付周期(天)

33.72

环比 ↑11.99%

产研交付周期(天)

18.07

环比 ↑19.83%

需求完成(个)

292

环比 ↓5.19%

产研团队流负载(个)

159

环比 ↑43.24%

研发侧平均规模(小时)

26.75

环比 ↓7.96%

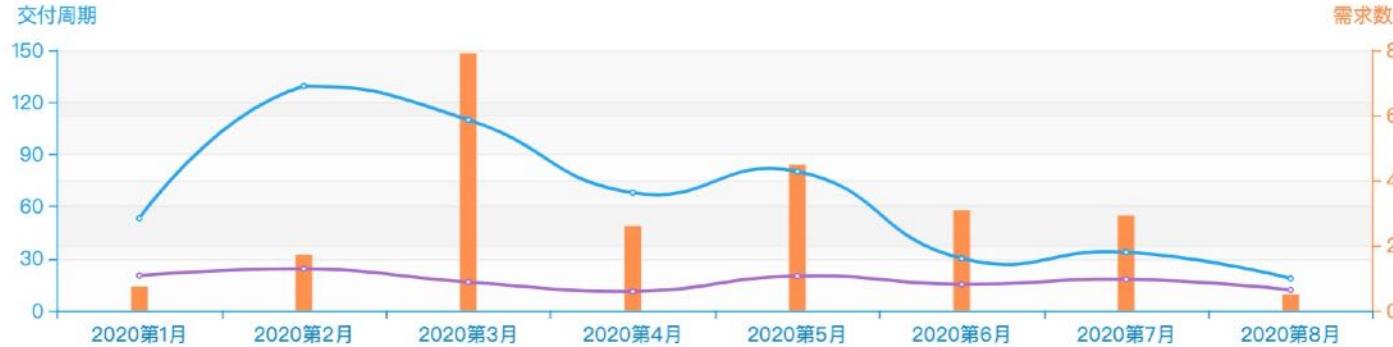
需求新增(个)

390

环比 ↑54.76%

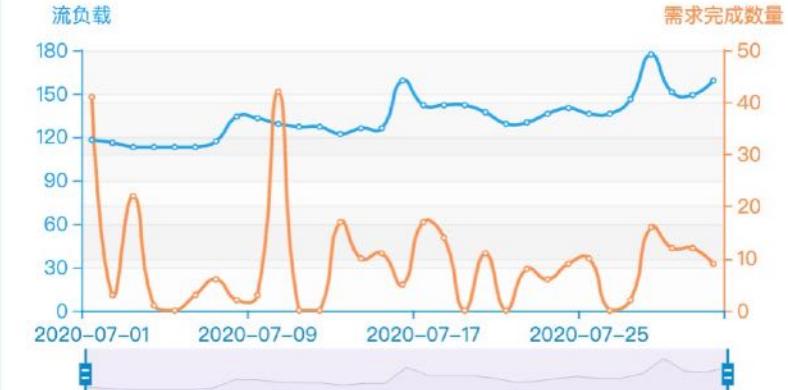
交付周期

需求交付周期 产研交付周期 需求数量



交付数量趋势

流负载 需求完成数量



各阶段停留时长

阶段停留时长



卡片状态累积流

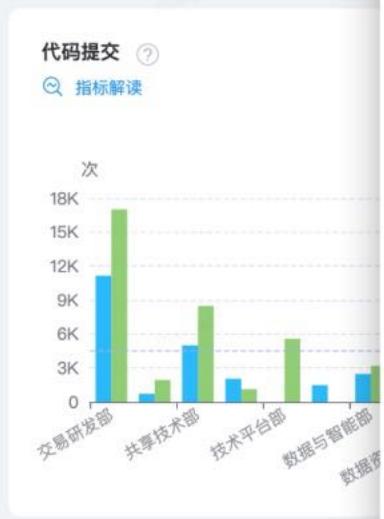
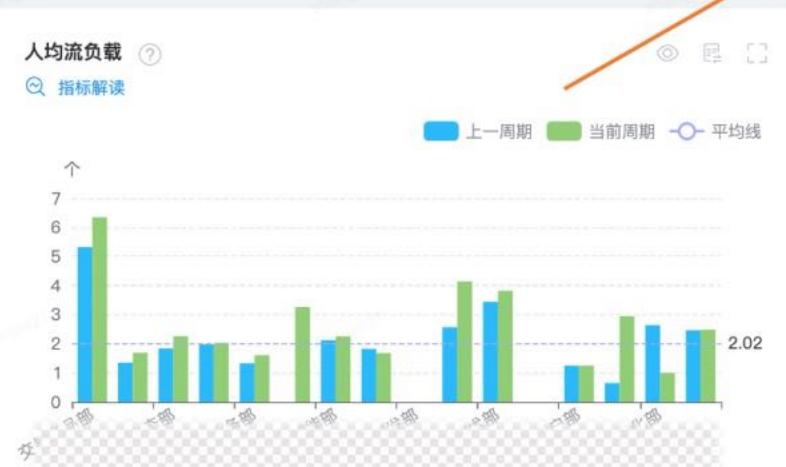
完成 发布 测试 开发 设计 就绪 评审 准备



需求交付周期分布图

需求数量 80%分位数



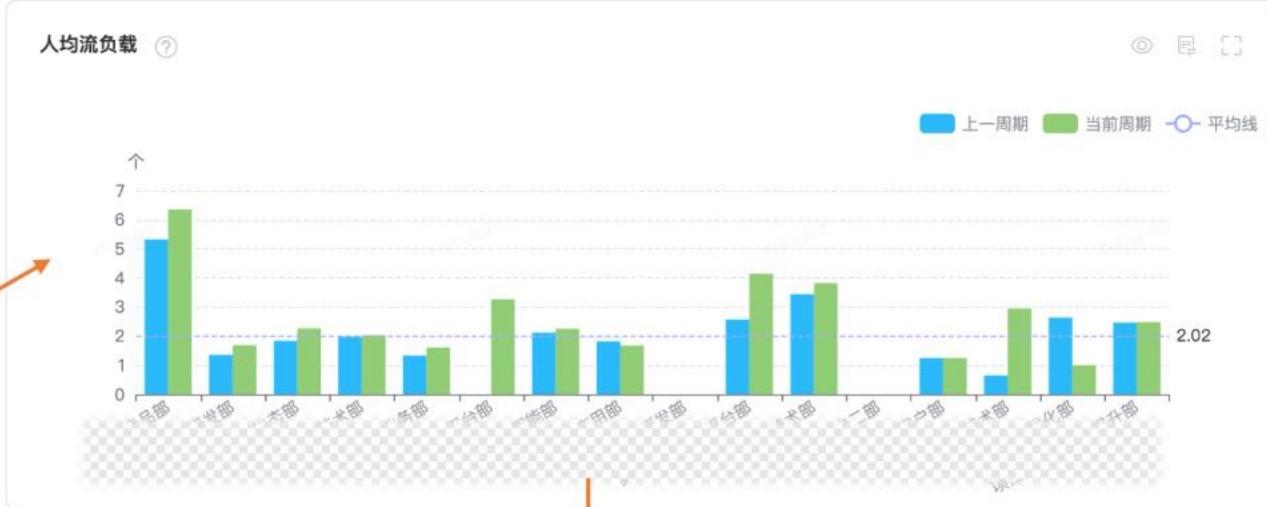


指标解读

同级部门对比

本部门趋势

指标下钻



指标明细 (共484条)

兩處

需求编号	需求名称	需求状态	需求阶段	阶段停留时长(天)	卡任务处理人	需求受理人
R2021030345204	BC...	处理中	准备	163.61	...	niu...
R2021020530123	【集...	处理中	准备	123.34	...	gu...
R2021062420512	京东...	处理中	开发	29.44	...	che...
R2021070615205	印尼...	处理中	开发	22.48	...	liuj...
R2021070717733	【P...	处理中	准备	52.54	...	zh...
R2021072110753	京东...	处理中	发布	22.47	...	sar...
R2021072000944	数据...	处理中	测试	16.37	...	lix...

业务
结果

+\$

业务
价值

-\$

交付
成本

◎

客户
满意度



交付效率

端到端快速及早交付

需求前置时间

产研交付周期

需求吞吐量

需求细分阶段耗时

流负载 (在制品数)

流效率 (等待浪费)

需求规模 (颗粒度)

需求变更率 (返工)

需求评审通过率

需求平均代码量

代码技术债率

代码复杂度/重复度

代码提交/合并频率

代码评审覆盖率

单元测试覆盖率

自动化测试覆盖率

构建执行时长

构建频率/成功率

线下缺陷密度

缺陷解决时长

缺陷逃逸率

流水线红灯修复时长



交付质量

端到端高质量交付

上线成功率

线上缺陷密度

故障恢复时间



交付能力

卓越工程能力 持续交付

部署频率

变更前置时间

DevOps 团队

目标对齐

自组织

跨职能

学习型

敏捷和精益产品开发实践

价值流管理

可视化管理

小批量交付

限制在制品

需求拆分和实例化

敏捷迭代规划

轻量级变更流程

持续实验和反馈

度量驱动持续改进

持续交付工程技术实践

分支策略

版本控制

自动化测试

测试数据管理

自动化部署

测试环境管理

安全左移

数据库变更管理

架构解耦

仿真模拟器

持续集成

低风险发布技术

代码可维护性

监控/可观测性

DevOps 研发效能度量 @张乐老师

云原生 IT 基础设施

服务和流量管理

生命周期管理

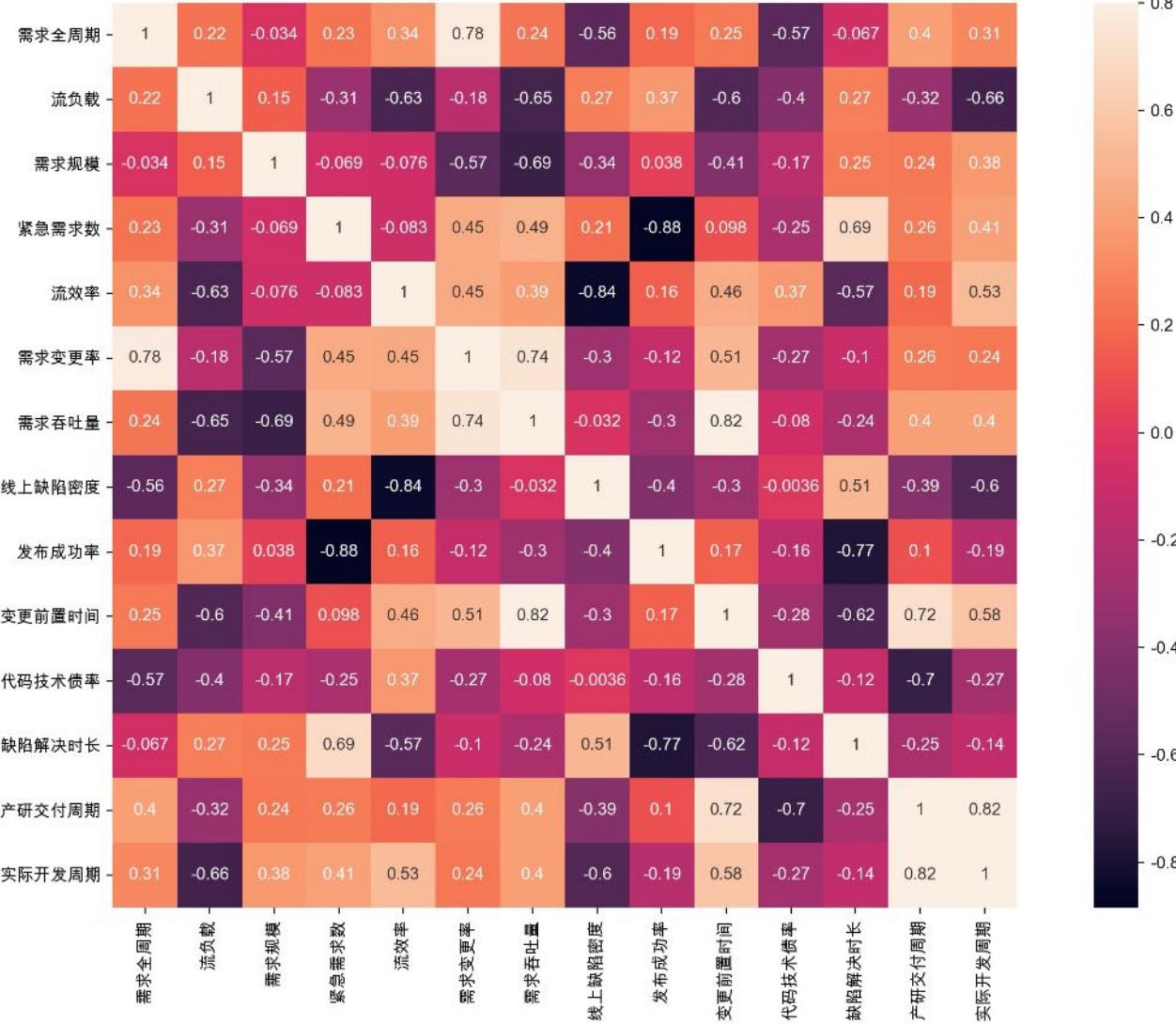
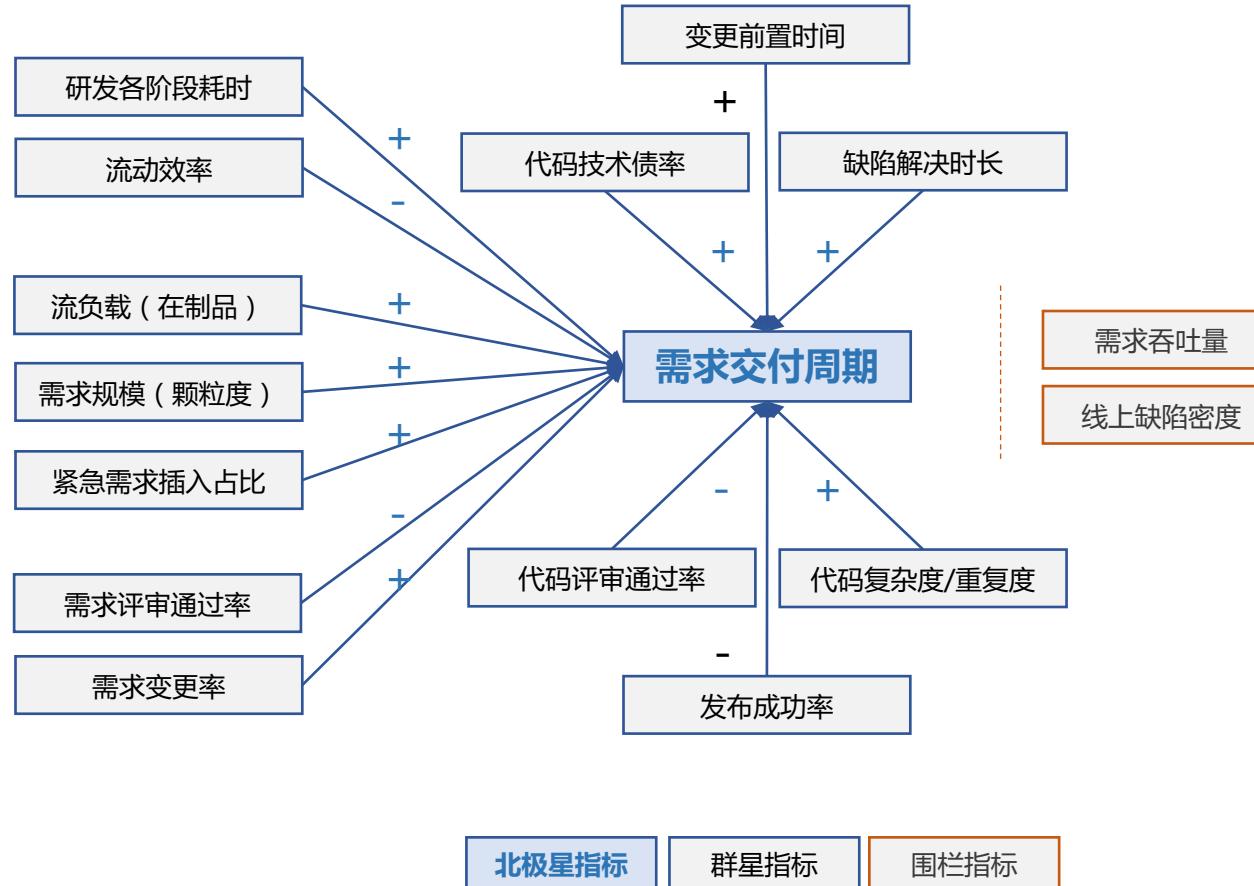
云服务依赖管理

K8S 容器编排

相关性分析

如果一种度量真的很重要，那是因为它必须对决策和行为产生一些可以想象的影响。

--Douglas W. Hubbard



避免踩坑：效能度量的常见误区

- 使用简单的、易于获取的指标
- 过度关注资源效率类指标
- 使用成熟度评级等基于活动的度量
- 把度量指标设置为KPI进行绩效考核
- 片面地使用局部过程性指标
- 手工采集，人为加工和粉饰指标数据
- 不顾成本，堆砌大量非关键指标
- 货物崇拜，照搬业界对标的指标
- 舍本逐末，为了度量而度量
- 仅从管理角度出发，忽略了为工程师服务



比尔·盖茨：“用代码行数来衡量软件的生产力，就像用飞机的重量来衡量飞机的生产进度一样。”



狭隘的、以活动为导向的度量往往效果不好。**研发效能首先应该度量结果而不仅是过程**，端到端价值流的局部优化对结果的改进效果也许很小，因为可能根本就没有解决效能瓶颈。



把度量与绩效挂钩就一定会产生“造数据”的数字游戏。应该**把度量作为一种目标管理方法、效能提升的参考工具**。促进团队明确效能目标、分析效能问题，指导团队针对性优化



可以使用**GSM（目标/信号/指标）框架**来指导目标导向型指标的创建，度量一定从目标出发。如果你不能对结果做任何事情，那么它很可能不值得度量。



做研发效能提升，本质上还是要多关注工程师的感受，他们对工作环境、工作模式、工作负载、研发基础设施、项目协作、团队发展、个人提升**是否感到满意**，是否有阻碍工程师发挥更大创造性和产生更大生产力的因素存在。



People who can't understand numbers are useless.
The Gemba where numbers are not visible is also
bad. However, people who only look at the numbers
are the worst of all.

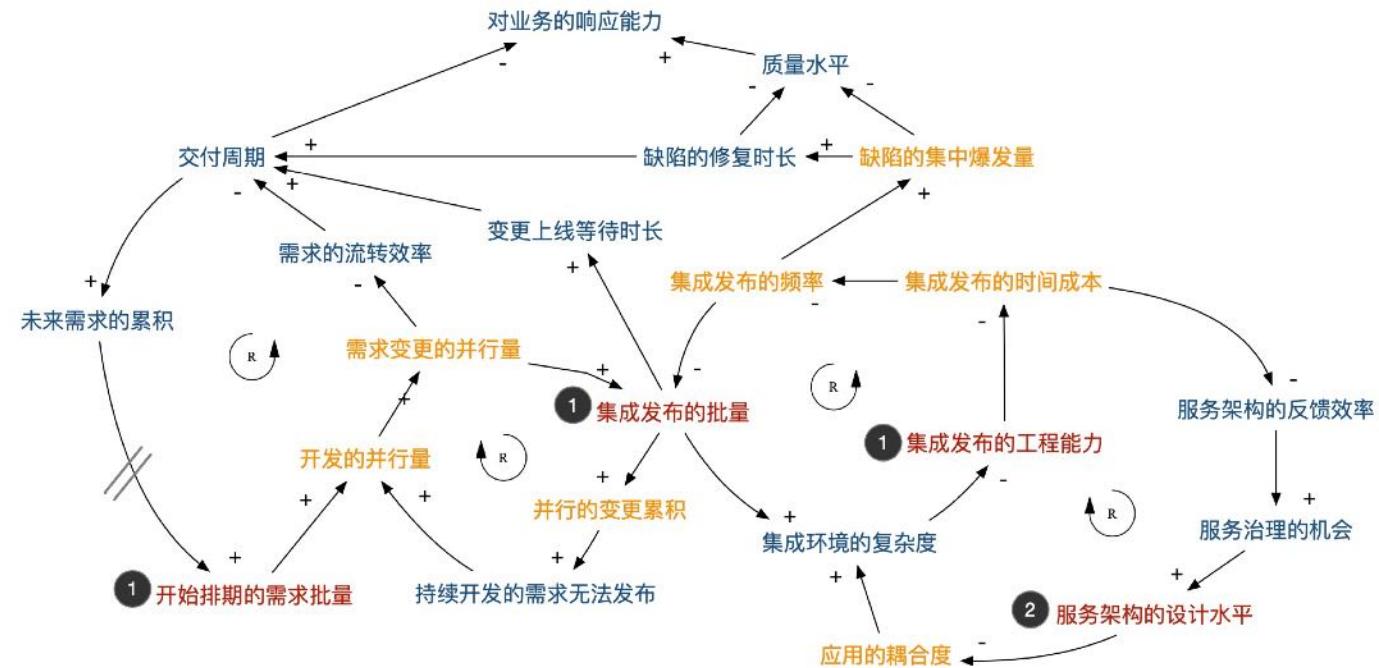
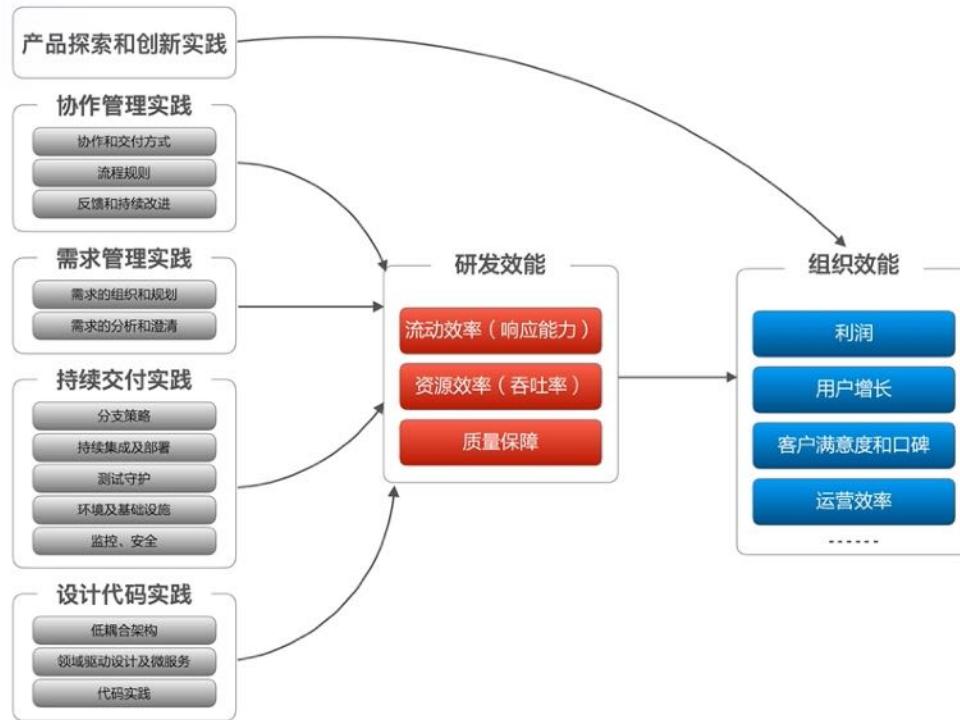
— *Taiichi Ohno* —

那些不懂数字的人是糟糕的，
而最最糟糕的是那些只盯着数字看的人

AZ QUOTES

研发效能度量行业发展方向

1. 逐步认知和尝试摆脱反模式
2. 从度量指标到效能诊断分析
3. 从堆叠指标到效能提升闭环



内容参考：张燎原，研发效能度量背后的系统洞察

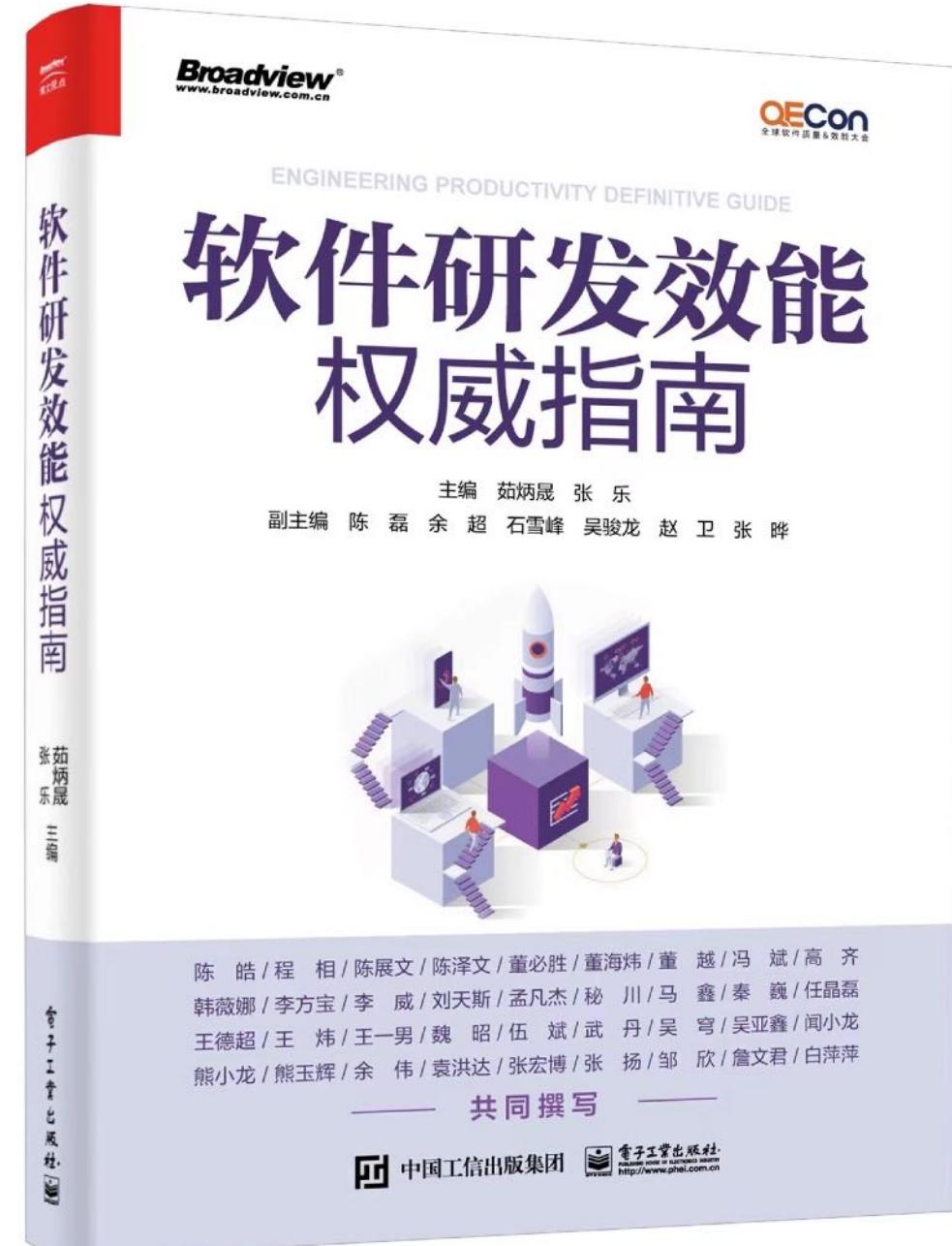
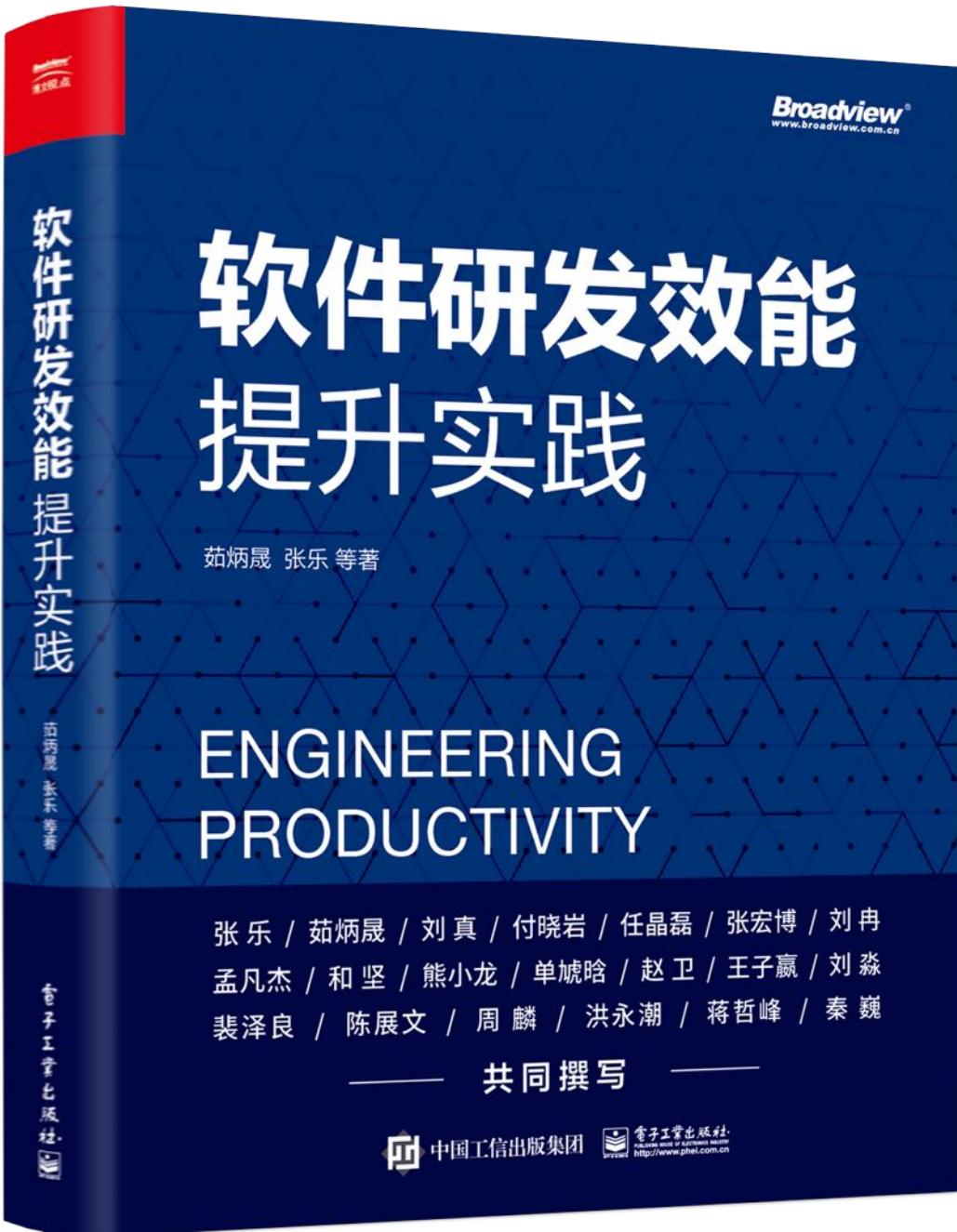
TAKEAWAYS

- 研发数字化的四个层次：开采出数据，粗炼出信息，精炼出知识，聚合出智慧
- 研发效能度量的五项精进：自动数据采集、度量指标体系、度量分析模型、度量产品设计、数据运营体系
- 深刻理解每一个指标背后的逻辑，以终为始，从目标和动机出发，通过指标牵引出期望的行为改变
- 研发效能度量分析模型是多层级的，从细分领域模型入手，先分析和解决特定场景的问题
- 重视数据驱动和实验思维，最重要就是让度量始终保持在正确的方向上
- 行业发展趋势：逐步认知和尝试摆脱反模式，从度量指标到效能诊断分析，从堆叠指标到效能提升闭环



效能度量系列文章 - 研发效能度量核心方法与实践：

- (一) 难点和反模式 <https://www.infoq.cn/article/vTucU06GZrOFHSmveuCW>
- (二) 行业案例和关键原则 <https://www.infoq.cn/article/kXCDJ0gMEE4cSv05XVNq>
- (三) 实践框架和指标体系设计 <https://www.infoq.cn/article/kINcaQBBC0uyvFmVlvzx>
- (四) 常用分析方法 <https://www.infoq.cn/article/b7G05ot0VJkFbVWQaLlx>
- (五) 落地实施建议 <https://www.infoq.cn/article/jk4hqaprvKcTttYhIIBP>





Thanks.

不躺平、不内卷，行稳致远

