GOPS 2021
Shenzhen

GOPS

GOPS 全球运维大会

2021
-XOPS 风向标

深圳站

中国·深圳

指导单位：

主办单位：

时间：2021年5月21日-22日

GOPS 2021
shenzhen

外包模式下度量驱动研发质量效能提升

温粉莲 效能运营分析主管

温粉莲

效能运营分析主管

广东移动网络管理中心质量效能运营分析主管，多年电信设备研发运维、信息系统集成工作经验。目前负责广东移动DevOps平台建设及系统质量效能运营分析。

GOPS 2021
shenzhen

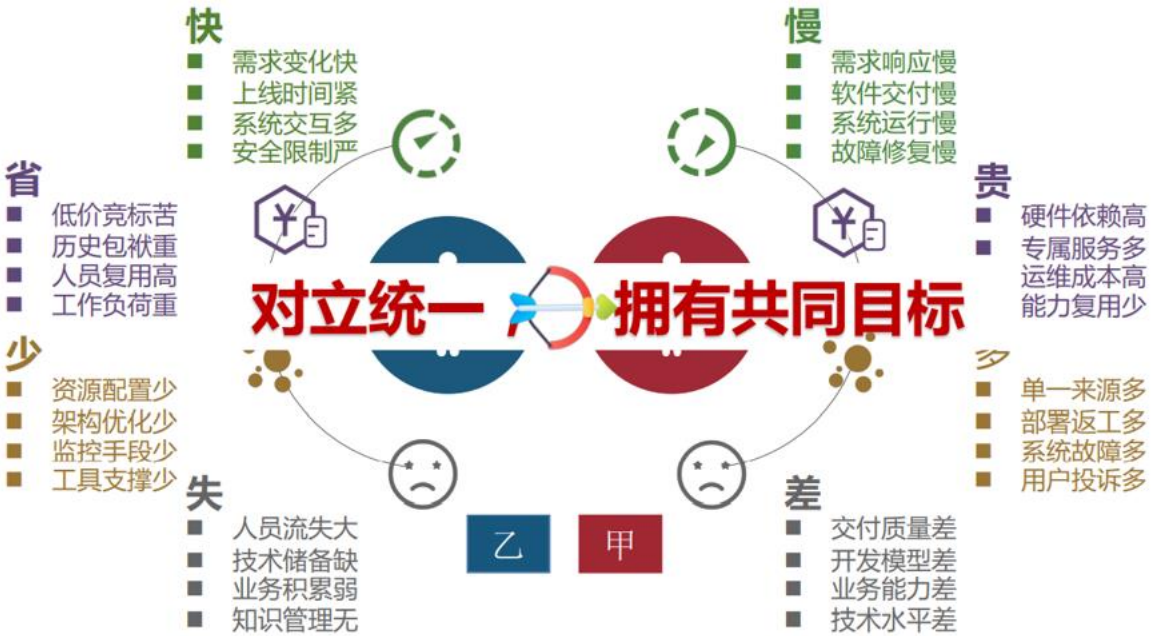
目录
CONTENTS

GOPS 2021
shenzhen

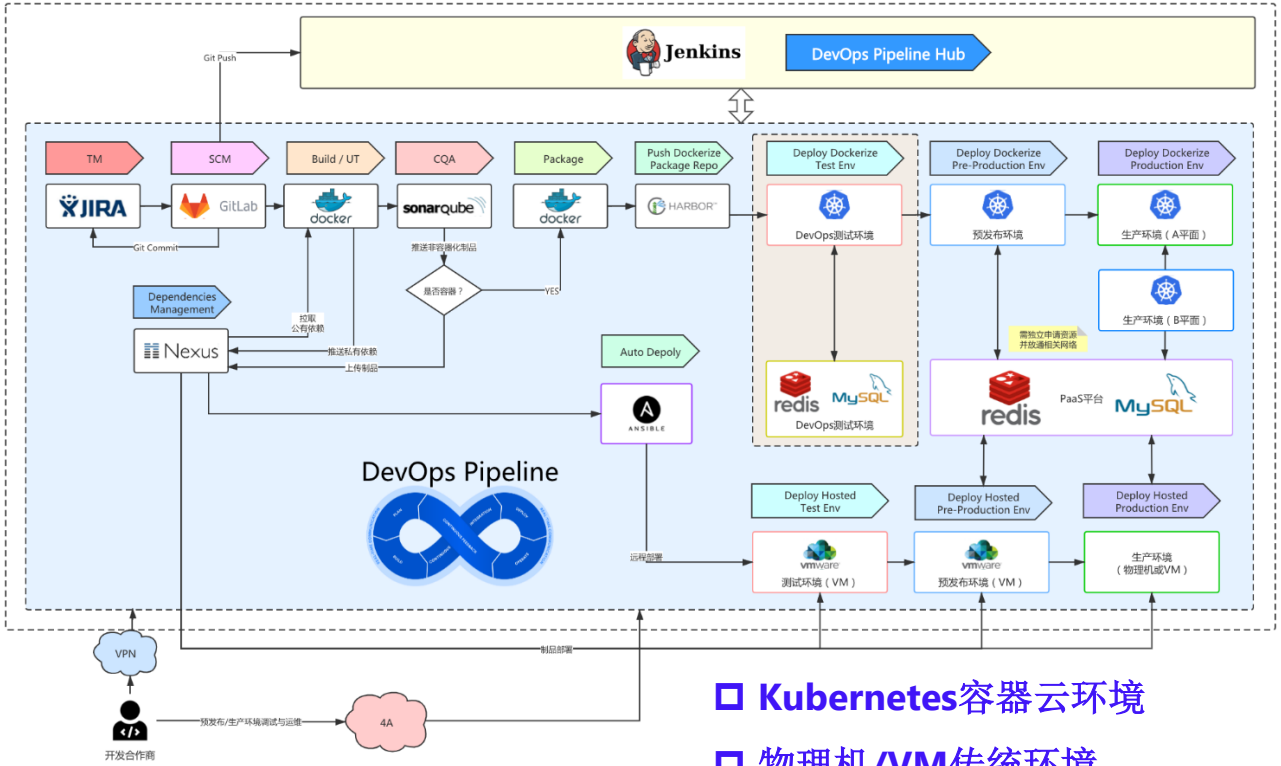
为什么要在DevOps中引入度量

01

IT外包模式下面临的挑战



自主搭建DevOps平台



- Kubernetes容器云环境
- 物理机/VM传统环境

- 测试环境
- 预发布环境
- 生产环境

接入DevOps之后...

老板

接入的系统使用情况？还有哪些没接入？

产生什么样的效果？

业务部门

为什么我的需求交付还是这么慢？

HOW?

运维

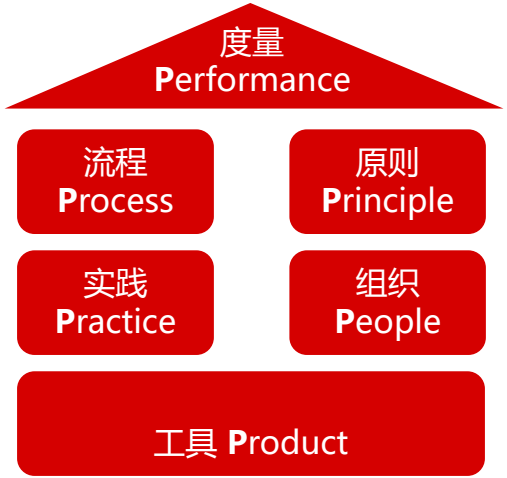
为什么上线还是有故障？

项目经理

故障恢复能更快吗？

自动化只是先决条件

流水线不是为了更快的交付低质量的软件



整 体 优 化

GOPS 2021
shenzhen

如何设计有效的度量体系

02

度量的目的

现　状

目　标

精　准　改　进

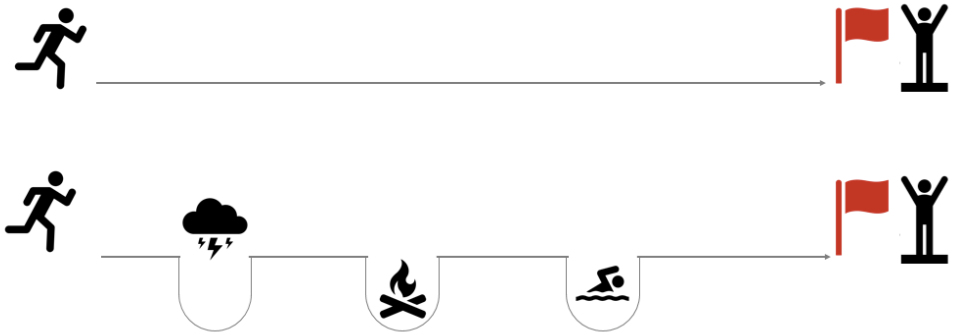
If you can’t measure it, you can’t change it.　-- *Peter Ducker*

通过数据度量，发现瓶颈，持续改进

度量设计应遵循的原则

度量是一个系统工程，需要不断演进

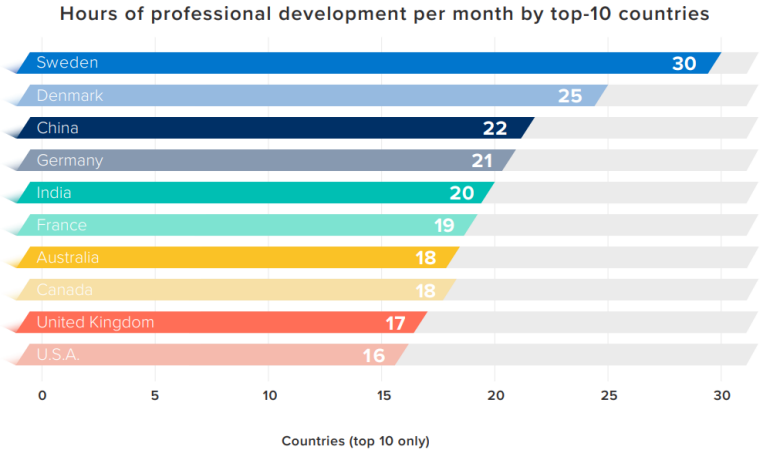
- *度量不是免费的*，需要成本，需要考虑有效性，需要考虑可靠性
- *度量是阶段性的*，选择合适当前阶段的指标，重点优化这些指标
- *度量是动态变化的*，需要不断的迭代，增加有用的，删除无效的



好的度量

$$\text{人均代码产出量} = \frac{\text{总代码变更行数}}{\text{开发人员数量}}$$

软件开发 ➡ 写代码？



容易理解，可比较

好的度量

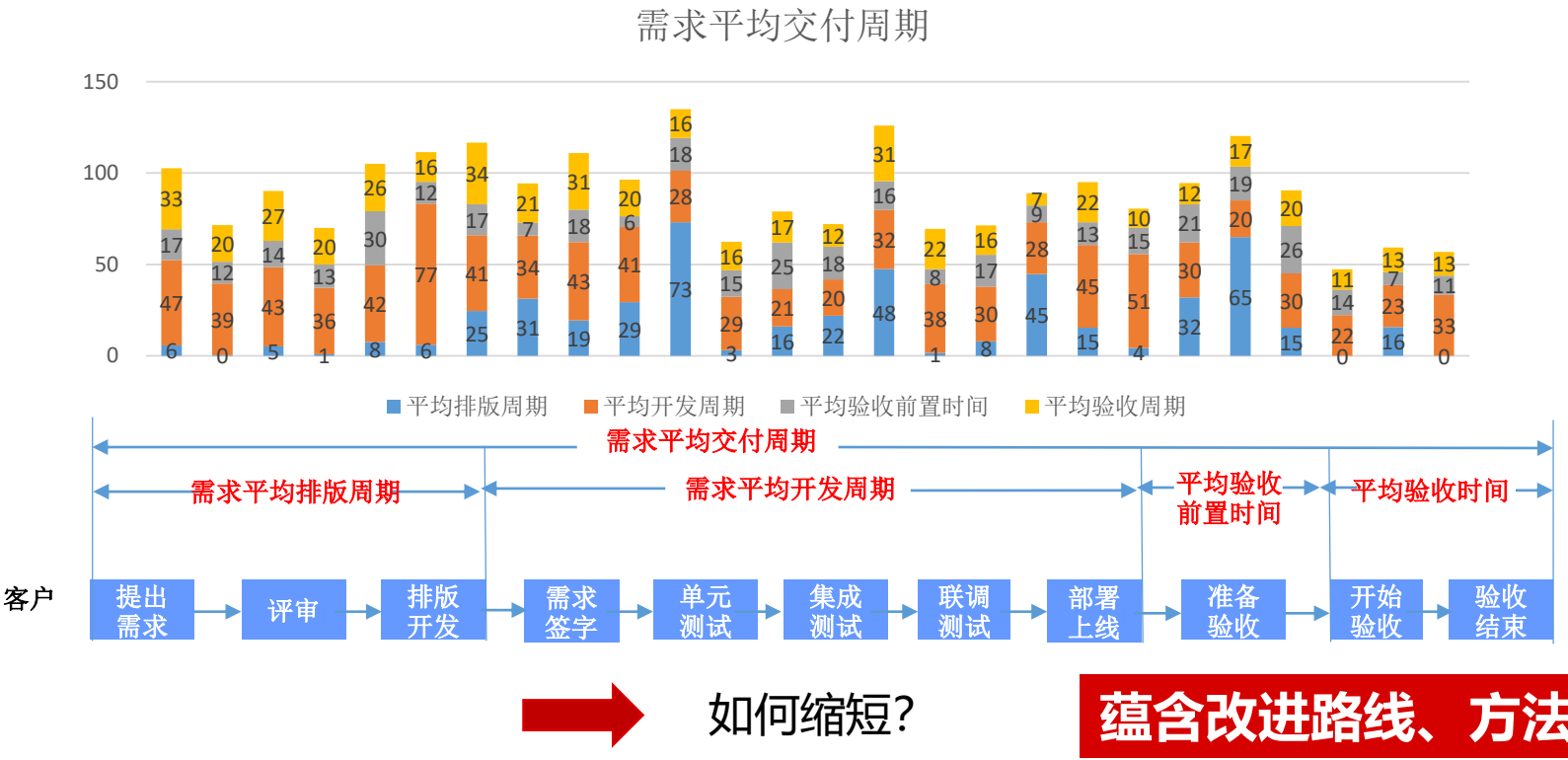
$$代码提交频率 = \frac{总代码提交次数}{开发人员数量}$$

代码提交频率提升 → 更高的集成频率？

OKR ? KPI

结果导向

好的度量



如何缩短?

蕴含改进路线、方法

好的度量

千行bug率是好指标吗?

单测覆盖率是好指标吗?

指标矩阵

- 多维度
- 相互制约
- 解读正确

小结

1. 可比较、容易理解
2. 结果导向
3. 蕴含改进路径
4. 多维度，避免片面

好的度量具备的特点

1. 评估效果
2. 发现问题
3. 体现价值

达到的效果

GOPS 2021
shenzhen

实施路径

DevOps运营分析实践

03

建立运营分析体系


分析方法、维度
定量分析、环比、排序、趋势分析多种方法综合使用
运营分析对象
识别运营分析的不同对象，调整运营分析频次、策略
01
02
03
04
效能指标
从两种基础指标、六大类效能指标出发，确定指标的定义、可能原因、改进举措
闭环机制
制定基于指标分析问题，确定改进和闭环机制

确立效能基础指标

GOPS 2021 shenzhen

研发运营一体化能力成熟度模型 第1部分：总体架构

GOPS2017 Shanghai

一、研发运营一体化（DevOps）过程															
敏捷开发管理			持续交付								技术运营				
需求管理	计划管理	过程管理	配置管理	构建与持续集成	测试管理	部署与发布管理	环境管理	数据管理	度量与反馈	安全管理与服务	监控服务	数据服务	容量服务	连续性服务	运营反馈
需求收集	需求澄清和拆解	迭代管理	版本控制	构建实践	测试分级策略	部署与发布模式	环境供给方式	测试数据管理	度量指标	安全开发	应用监控	数据收集能力	容量规划能力	高可用规划	业务知识管理
需求分析	故事与任务排期	迭代活动	版本可追踪性	持续集成	代码质量管理	持续部署流水线	环境一致性	数据变更管理	度量驱动改进	运营安全	质量体系管理	数据处理能力	容量平台服务	质量体系管理	项目管理
需求与用例	计划变更	过程可视化及流动			测试自动化					数据安全	事件响应及处置	数据告警能力	运营成本管理		业务连续性管理
需求验收		度量分析								风险与威胁模型	监控平台				运营服务管理
二、研发运营一体化（DevOps）应用架构															
三、研发运营一体化（DevOps）组织结构															

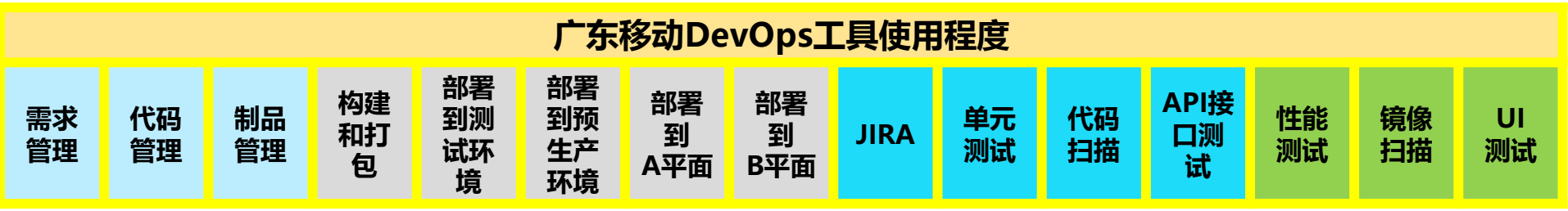
级别	英文	中文
1级	Regressive	阻碍的
2级	Repeatable	可重复的
3级	Consistent	一致的
4级	Quantitative	量化的
5级	Optimizing	优化的

$$\text{接入广度} = \frac{\text{接入的子系统/模块个数}}{\text{在用子系统/模块个数}}$$

DevOps工具链使用成熟度

数据分析的基本特征：可采集，可分析，可推论

基于工具的成熟度定义



L1：自动化配置管理能力

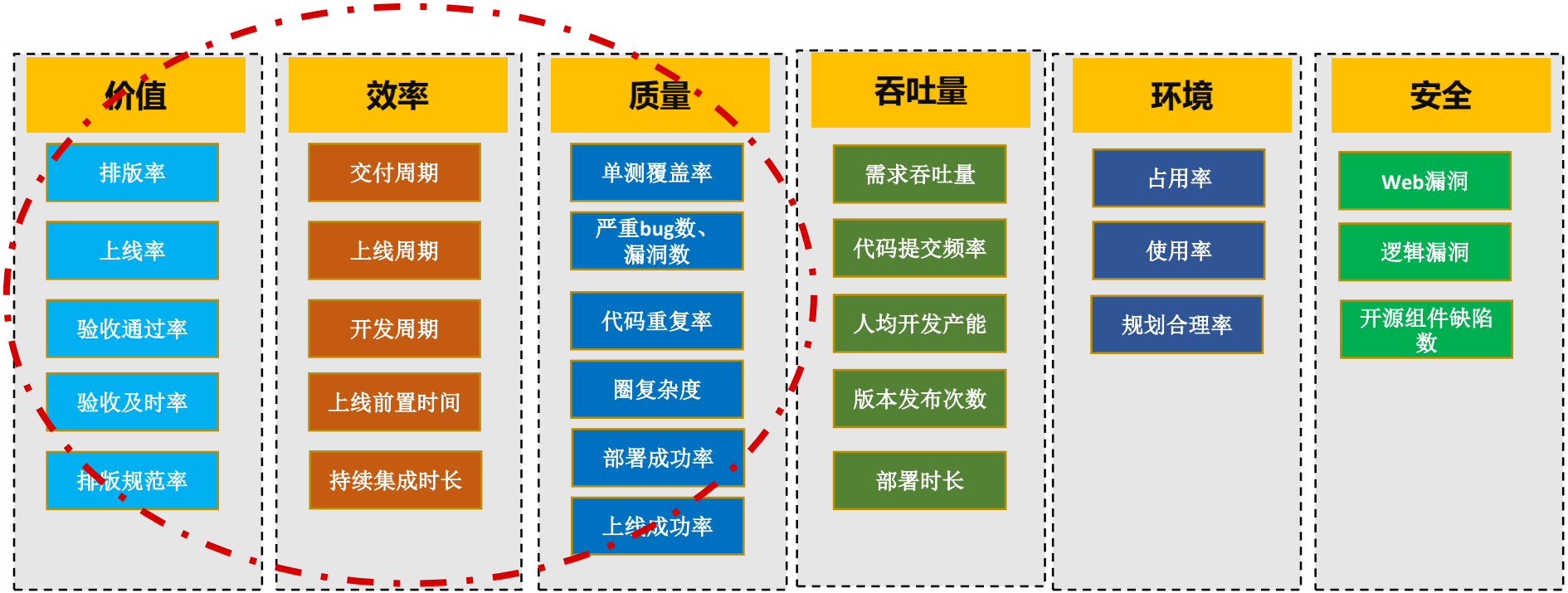
L2：持续集成/持续部署能力

L3：持续质量改进能力

L4：持续高质量和安全改进能力

自动化工具的使用程度，反映系统开发过程持续改进的能力

确立研发质量效能指标


价值
排版率
上线率
验收通过率
验收及时率
排版规范率
效率
交付周期
上线周期
开发周期
上线前置时间
持续集成时长
质量
单测覆盖率
严重bug数、漏洞数
代码重复率
圈复杂度
部署成功率
上线成功率
吞吐量
需求吞吐量
代码提交频率
人均开发产能
版本发布次数
部署时长
环境
占用率
使用率
规划合理率
安全
Web漏洞
逻辑漏洞
开源组件缺陷数

各阶段的数据采集

- 需求排版率
- 需求上线率
- 需求验收率
- 规范率

- 代码提交频率

- 构建频率
- 构建时长
- 构建成功率

- 单侧覆盖率
- 技术债务
- 代码重复率
- 圈复杂度
- 自动化测试数量
- 自动化测试成功率
- 手动测试数量

- web漏洞数
- 逻辑漏洞数
- 第三方漏洞数

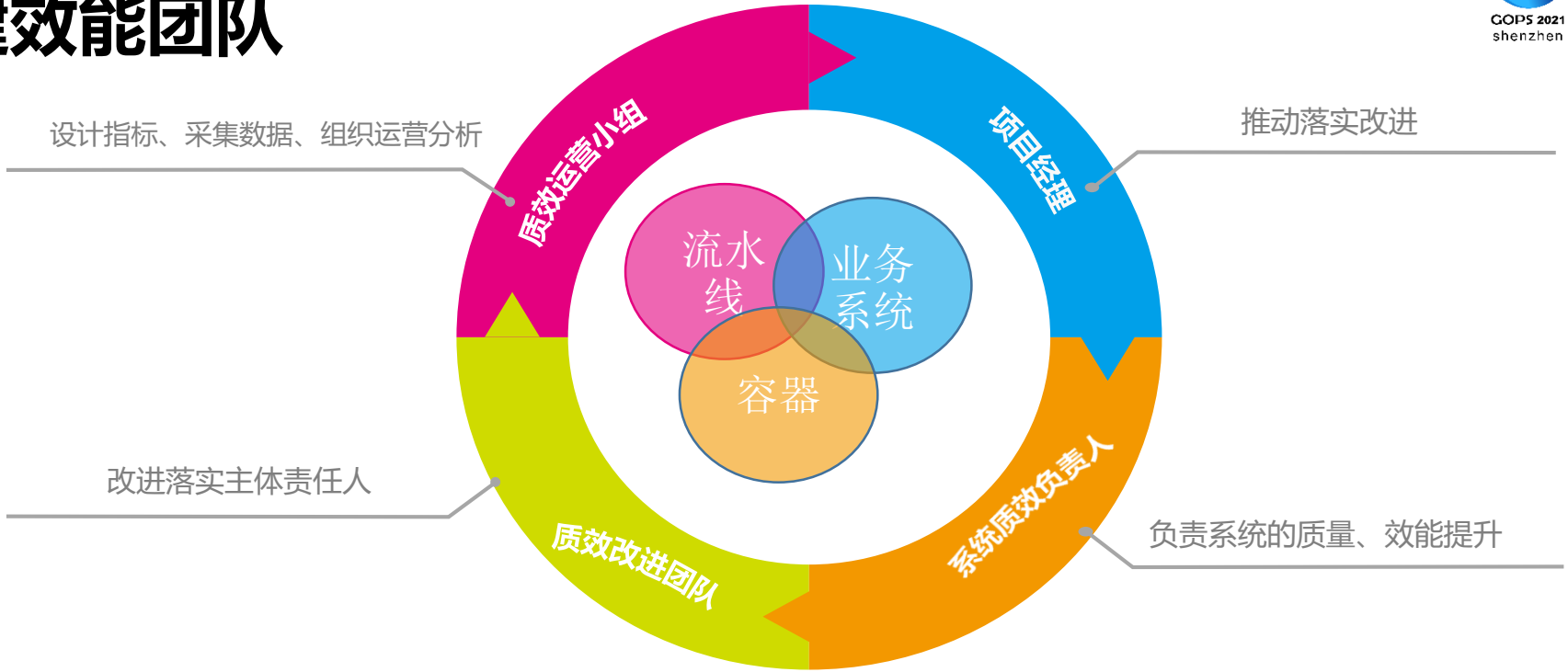
- 资源使用统计
- 持续集成时长

- 部署时长
- 部署成功率

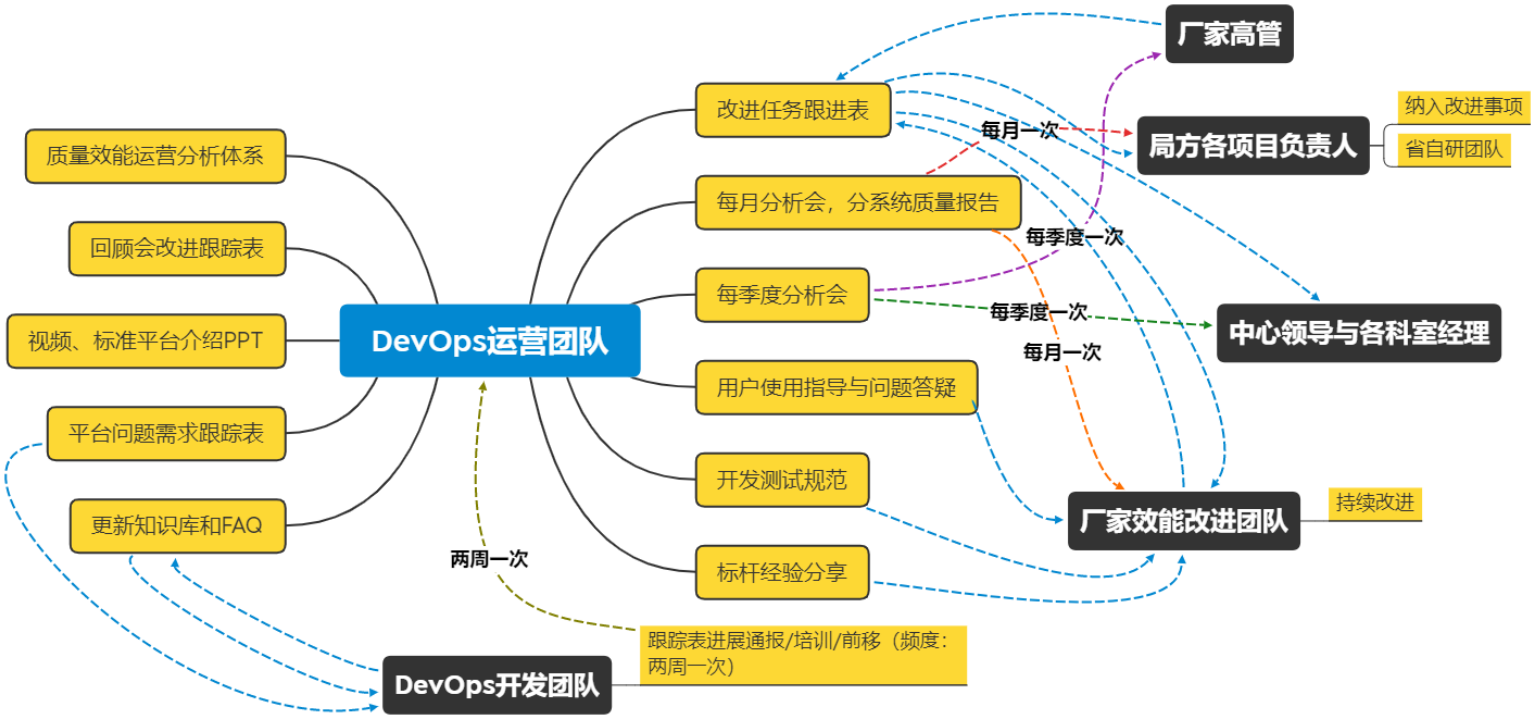
- MTTR

Prometheus + 脚本

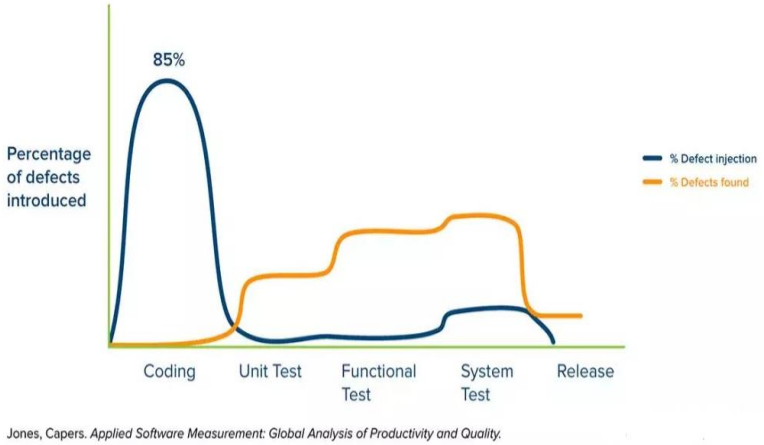
组建效能团队


设计指标、采集数据、组织运营分析
质效运营小组
项目经理
推动落实改进
流水线
业务系统
容器
改进落实主体责任人
质效改进团队
系统质效负责人
负责系统的质量、效能提升

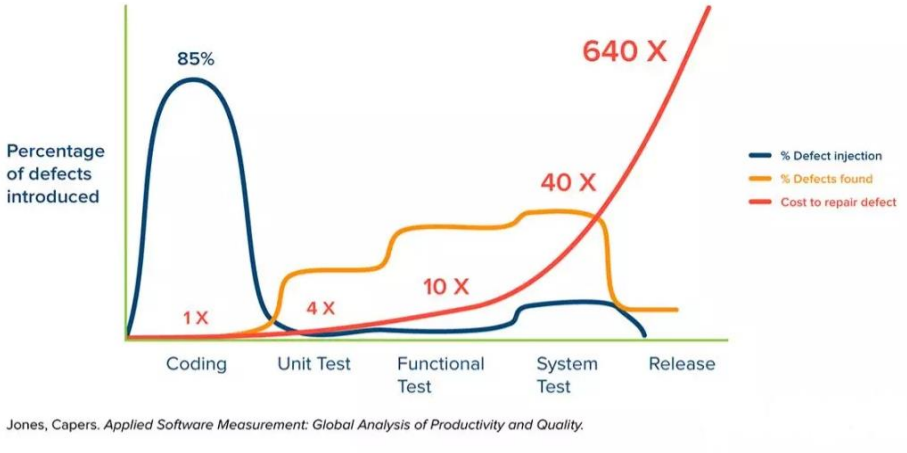
推广运营


厂家高管
改进任务跟进表
每月一次
局方各项目负责人
纳入改进事项
省自研团队
质量效能运营分析体系
每月分析会，分系统质量报告
每季度一次
回顾会改进跟踪表
每季度分析会
每季度一次
中心领导与各科室经理
视频、标准平台介绍PPT
DevOps运营团队
每月一次
用户使用指导与问题答疑
平台问题需求跟踪表
开发测试规范
厂家效能改进团队
持续改进
更新知识库和FAQ
两周一次
标杆经验分享
跟踪表进展通报/培训/前移（频度：两周一次）
DevOps开发团队

质量内建,引导观念转变



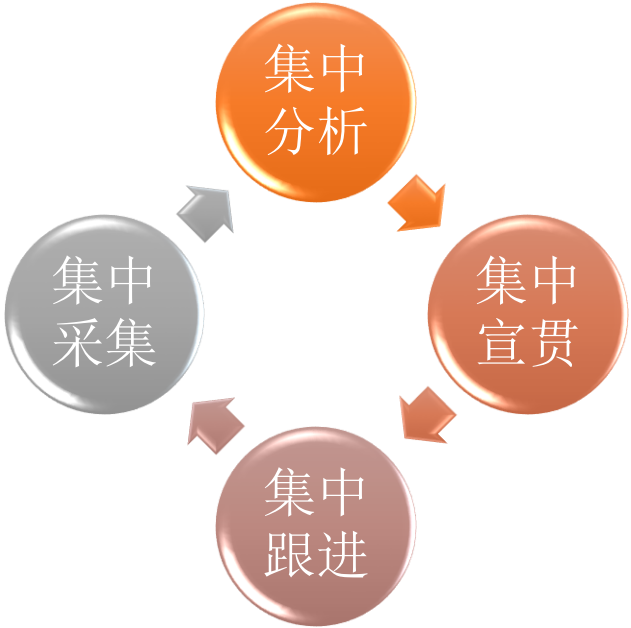
Jones, Capers. *Applied Software Measurement: Global Analysis of Productivity and Quality.*



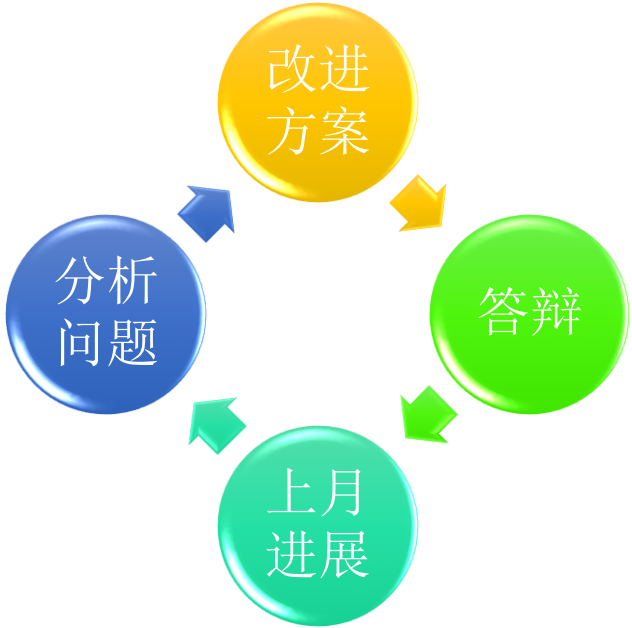
Jones, Capers. *Applied Software Measurement: Global Analysis of Productivity and Quality.*

测试左移

调动积极性，主动改进



保 姆 式 运 营



主 动 式 运 营

聚焦问题，形成综合打分机制

需求管理混乱：	需求排版规范率
客户满意度低：	一次验收通过率
需求延期 ：	上线及时率
DevOps组件使用少：	DevOps成熟度
集成测试通过率低：	单侧覆盖率
部署时间长：	部署成功率
代码质量差：	千行bug率、重复率、漏洞数
上线故障多：	一次性上线成功率

结果导向：

每系统、每月得分作为《网管支撑定制软件研发项目考核办法》的打分依据

得分结果评估团队改进成效

经验总结、分享

优秀项目分享自动化测试经验

质量差项目的失败原因分析

比学赶帮超

优秀项目分享项目开发管理经验

项目获得DevOps转型成功奖

高效的DevOps组织

总结

度量的必要性 → 现状更清晰 / 目标更明确 / 改进更精准

好的度量 → 可比较 / 结果导向 / 蕴含改进路径

持续运营、不断改进 → 建立体系 / 推广运营 / 调整优化

Thanks

高效运维社区

开放运维联盟

荣誉出品