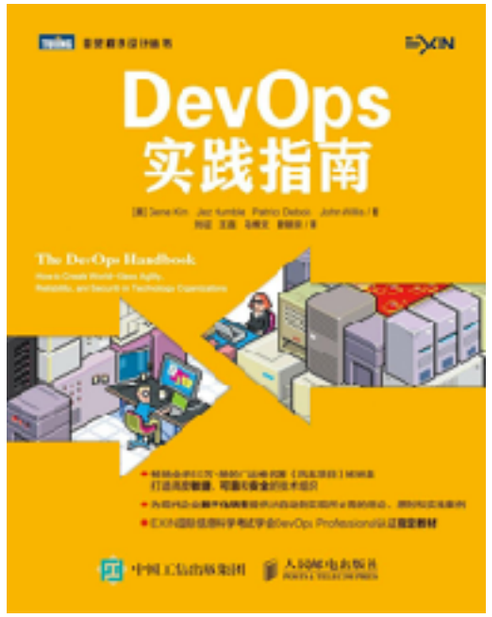
微服务架构与实践指南

关于我

华为2012技术专家

ThoughtWorks首席咨询师

Sybase Software Engineer



- 微服务、DevOps、持续交付有丰富的实践经验
- 《微服务架构与实践》作者
- 《DevOps实践指南》译者
- 中国首批EXIN DevOps Master教练

- 微服务架构的核心
- 微服务架构生态系统
- 微服务参考模型与实践

容错性

快速上线

复杂度增加

高可用性

需求快速响应

流量不确定

可管理性

独立发布

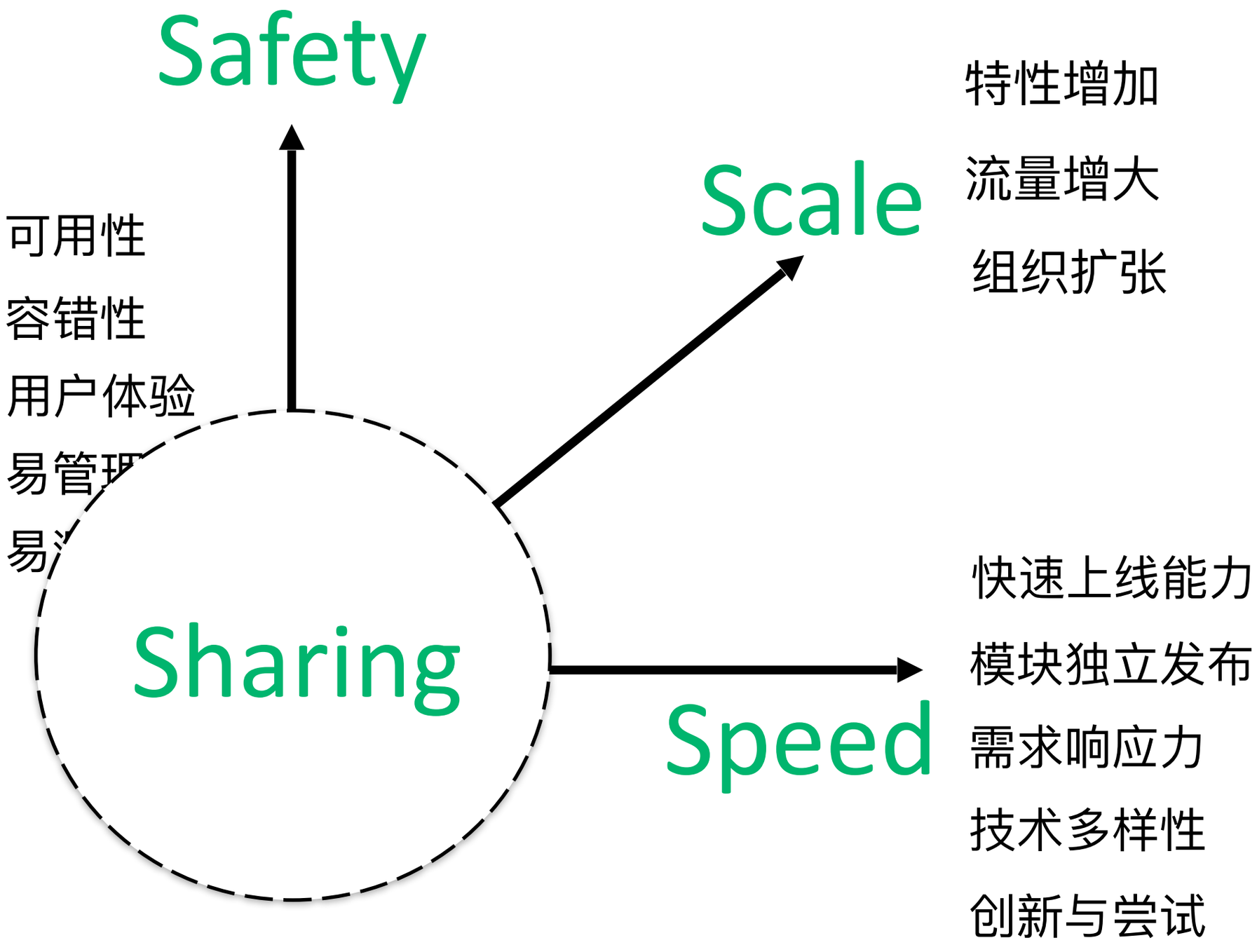
创新与尝试

灰度发布

可测试性

技术多样性

组织扩张


Safety
可用性
容错性
用户体验
易管理
易运
Scale
特性增加
流量增大
组织扩张
Sharing
Speed
快速上线能力
模块独立发布
需求响应力
技术多样性
创新与尝试

什么是微服务架构

微服务架构

Microservices - the new architectural style

Martin Fowler, Mar 2014

微服务架构是一种架构模式，它提倡将单一应用程序划分成**一组小的服务**，服务之间互相协调、互相配合，为用户提供最终价值。

每个服务运行在其**独立的进程中**，服务与服务间采用**轻量级的通信机制**互相协作（通常是基于HTTP协议的RESTful API）。

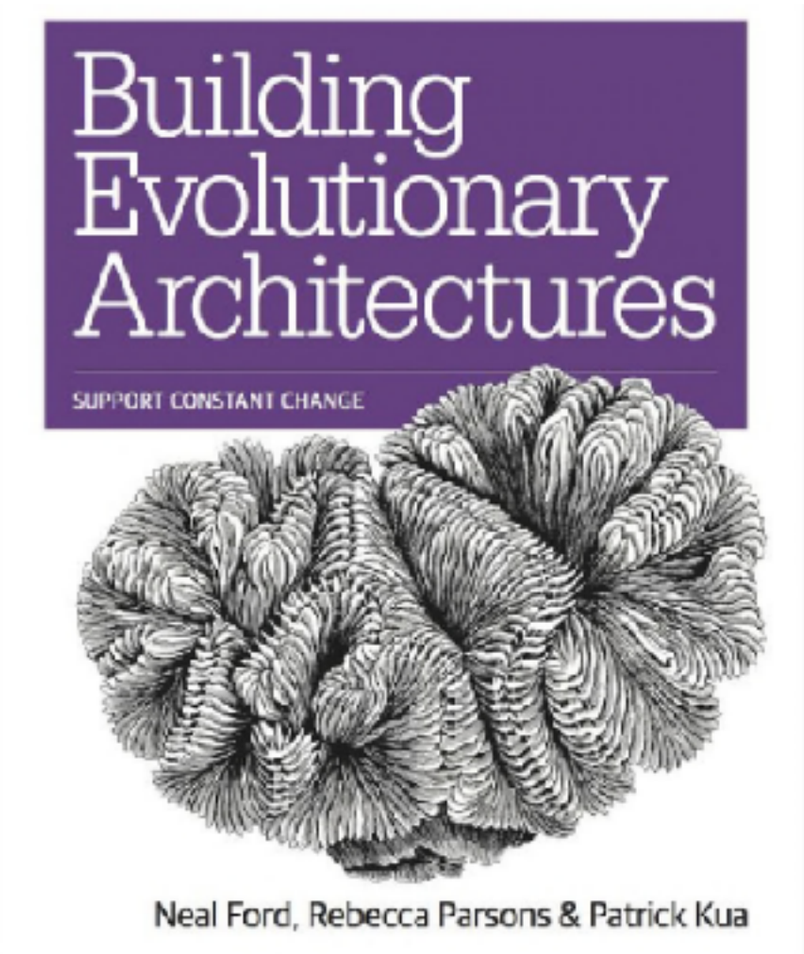
每个服务都围绕着具体业务进行构建，并且能够**被独立的部署**到生产环境、类生产环境等。

以缩短缩短交付周期为核心
基于DevOps
的演进式架构

A thousand Hamlets in a thousand people’s eyes.

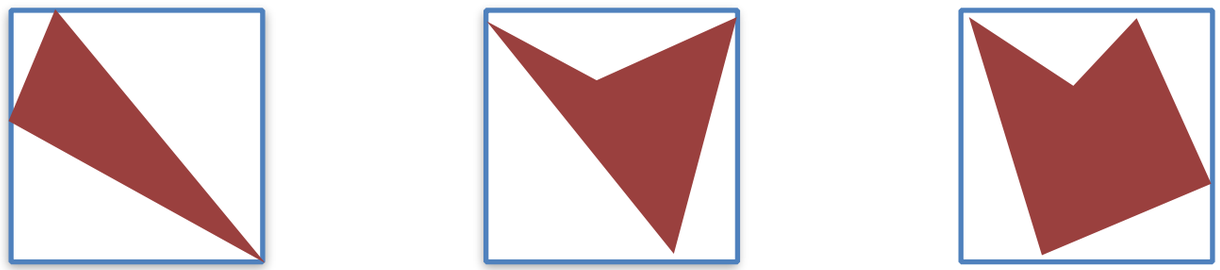
Schakespear

什么是演进式架构?



《演进式架构》
O’Reilly 2017 11

- 架构一旦确定，很难改变



- 软件的交付周期需求
- 瀑布流程的根深蒂固
- 部署环境的变更成本

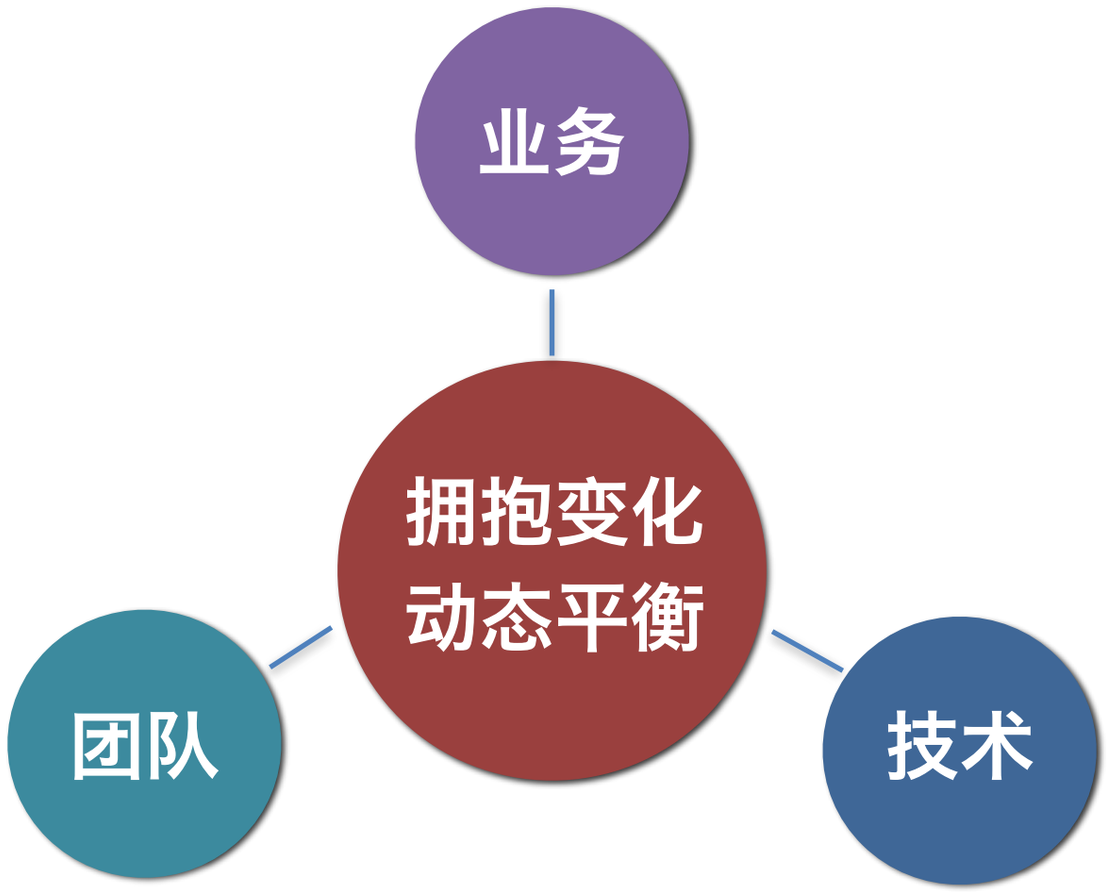
什么是演进式架构?

- 支持增量式变更作为第一原则

- 持续的动态演进
- 运维意识是关键
- 痛苦的事情提前做

演进式架构的核心

- **动态演进**

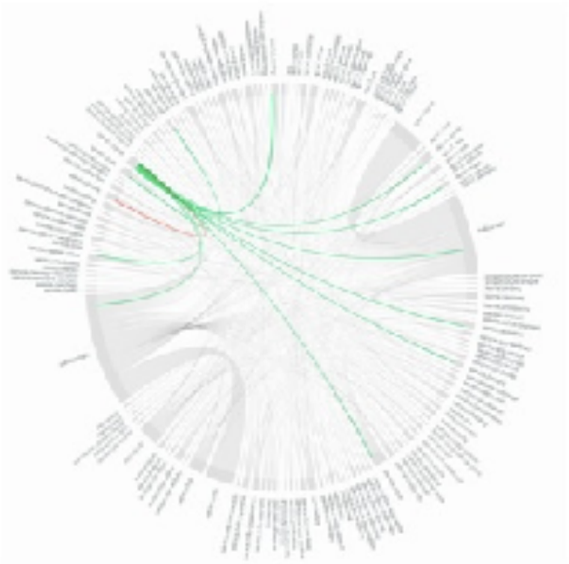


基于业务、技术和团队的**动态平衡与演进**

演进式架构的核心

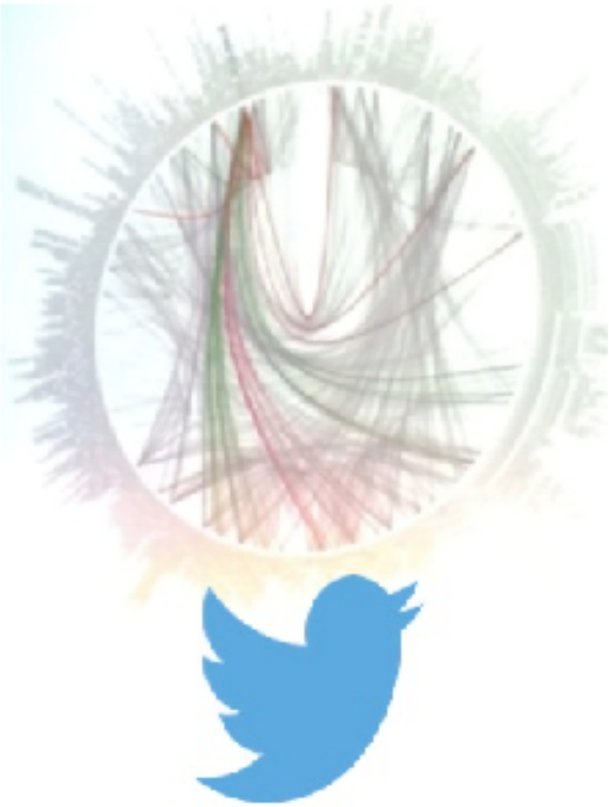
- **架构师的运维意识**
  - 架构只是抽象，直到**真正上线**并投入运维**产生价值**
  - 软件世界不断的变化，而架构只是**演进过程的快照**

450 microservices



500+ microservices

500+ microservices

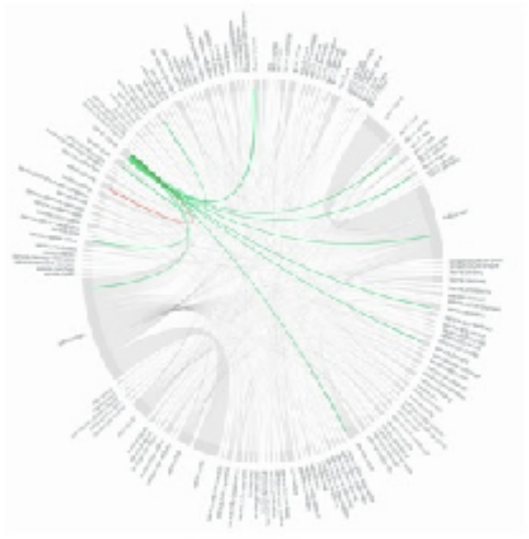


演进式架构的核心

- **痛苦的事提前做**
  - **不断**识别问题并用**自动化**的手段消除痛苦
  - 持续集成、持续部署、基础设施即代码

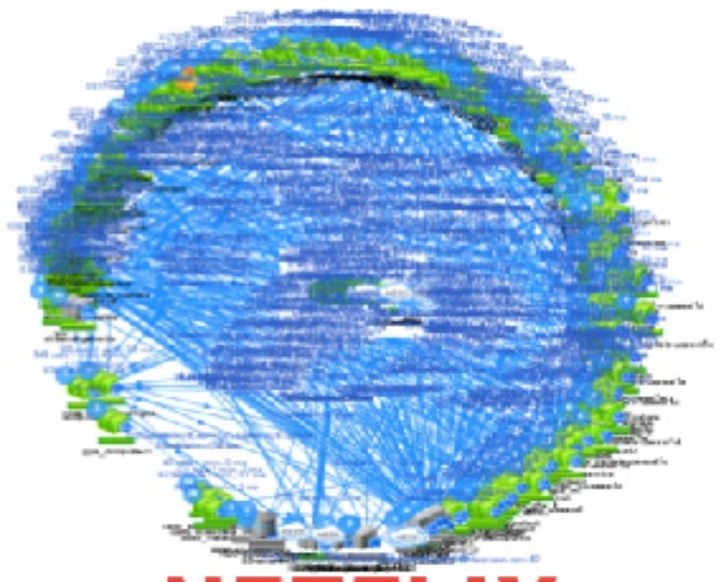
微服务架构是一种演进式架构

450 microservices



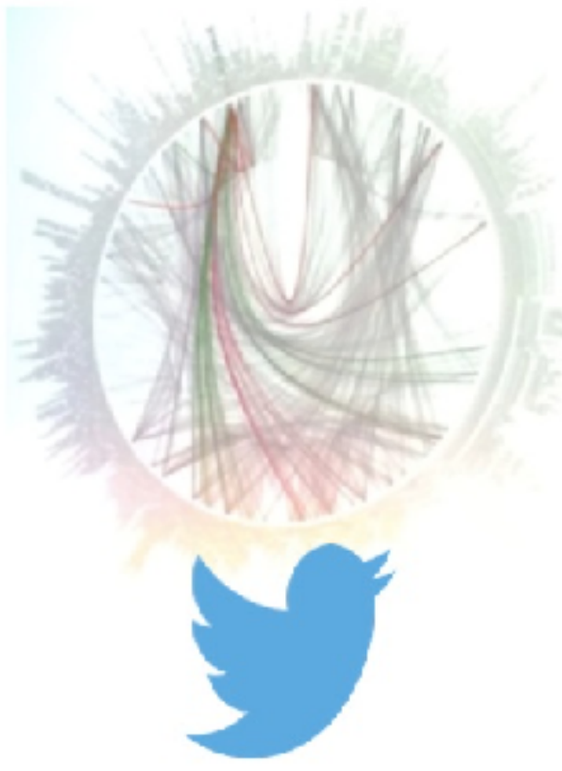
HAIL O

500+ microservices



NETFLIX

500+ microservices



Source:
Netflix: http://www.slideshare.net/BruceWong3/the-case-for-chaos
Twitter: https://twitter.com/adrianco/status/441883572618948608
Hail-o: https://sudo.hailoapp.com/services/2015/03/09/journey-into-a-microservice-world-part-3/

以缩短**交付周期**为核心

基于**DevOps**

构建的**演进式架构**

I am not perfect in my walk but I want to do the right thing.

Kirk Cameron

- 微服务架构的核心
- 微服务架构生态系统
- 微服务参考模型与实践

为什么需要生态系统?

系统化的工程

- 分布式系统
- 服务的治理与维护
- 测试策略与自动化
- 持续交付流水线

框架层出不穷

- API网关
- 服务开发框架
- 测试验证框架
- 部署运维工具

多维度互相依赖

- 基础设施(私有云/公有云)
- 持续集成/持续部署流水线
- 团队的敏捷/工程化实践
- 端到端的工具链

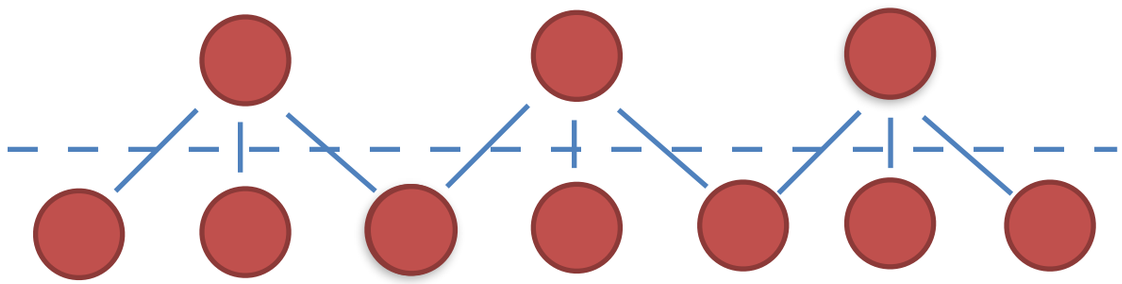
微服务生态系统

接入层

API网关/Edge Service

业务层

- 聚合服务
- 基础服务



支撑层

注册发现 | 集中配置 | 容错 | 调用链 | 授权认证 | 路由 | 日志聚合 | 监控

基础设施

CaaS/PaaS

IaaS

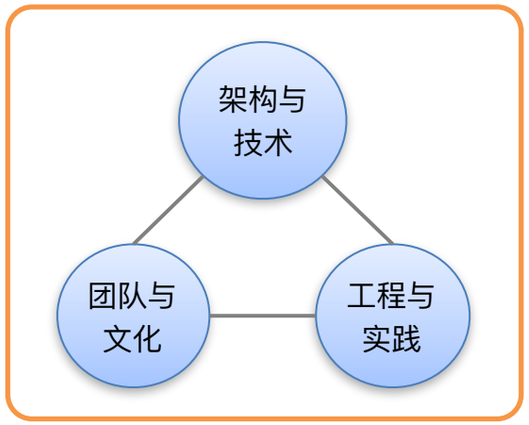
交付流水线与工程实践

- 服务开发框架
- 持续交付流水线
- 端到端的工具链
- 工程实践与规范

- 微服务架构的核心
- 微服务架构生态系统
- 微服务参考模型与实践

微服务实施参考模型

从3个方向，5个等级，8个维度，116+工程实践，26+项度量指标演进微服务



- **架构与技术**
  - 服务定义与实现
  - 服务支撑组件
  - 运维管理
- **流程与工具**
  - 持续集成
  - 部署管理
  - 测试管理
- **团队与文化**
  - 全功能团队
  - 敏捷实践

等级	定义	描述
Level 0	初始级（Initial）	服务交付的过程执行多借助手动，流程通常是混乱和无序的，缺乏稳定的环境，产品的交付经常超出成本或赶不上进度
Level 1	已管理级（Managed）	服务实施过程得到管理，服务交付的过程已经被计划、执行、跟踪和管控，部分过程自动化，服务状态可视化
Level 2	已定义级（Defined）	定义和建立了团队级的标准过程，用标准、规范、工具和方法等描述了服务的交付过程，并在整个组织范围内得到认可，在应用生命周期上实现了自动化过程
Level 3	量化管理级（Quantitatively Managed）	定义了过程度量指标，能够评价服务交付过程执行的性能，而且交付过程是可以被测量、控制和预测的，构建了可视化的度量收集方法并持续跟踪
Level 4	持续优化级（Optimizing）	形成了成熟的自治团队，能够自行分析和解决过程执行中的相关问题，可以有效识别和控制交付过程中的相关风险，建立了适合自身的持续优化和创新过程，并能够在组织内有效复制

XXX背景介绍

✓ **客户：** 一线 MKT/行销人员

✓ **定位：** “以营促销”转型，支撑精准“看病” 快速“开方”

✓ **现状**： 200W+代码，团队50+ 发布周期2~3个月

修改需要重新构建整个应用

测试时间长，部署时间上，代码耦合度高

新员工上手时间长，维护成本高

期望交付速度2~3周，支撑项目创新

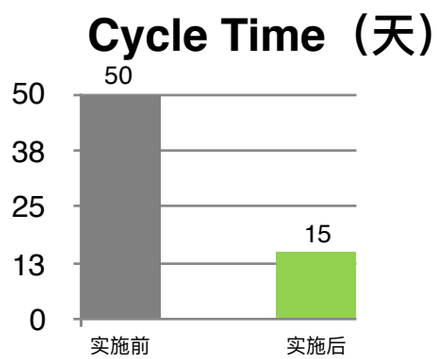
-

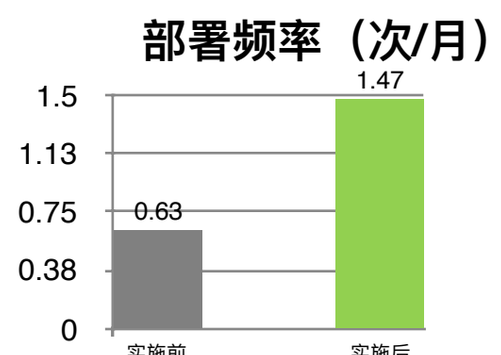
应用一系列实践

3个月后.....

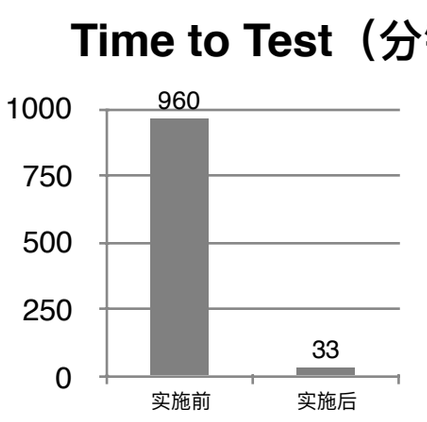
应用实践效果

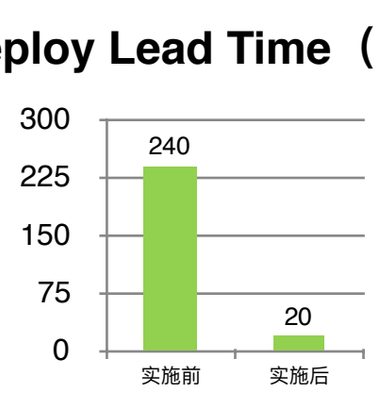
一、结果类度量是核心指标，反映了服务实施过程中端到端的交付效率

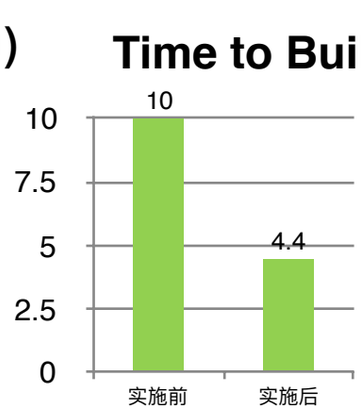


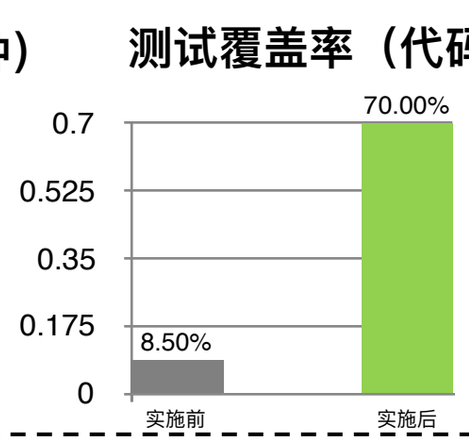


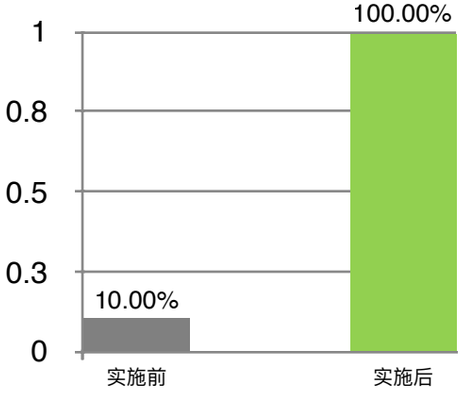
二、过程类度量指标是可选项，反映了服务实施过程中局部的优化效率

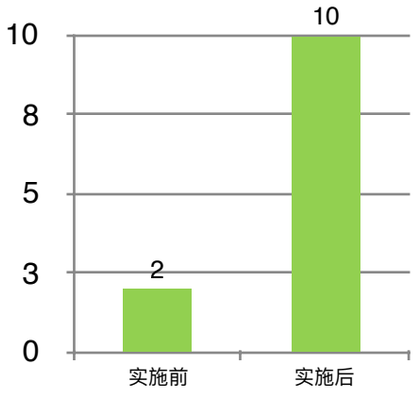


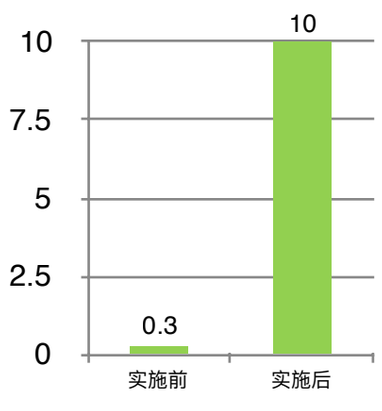


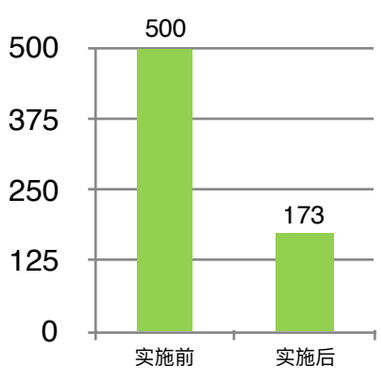












落地20+项实践，试点服务的交付周期从50天降低到14天。

谢谢