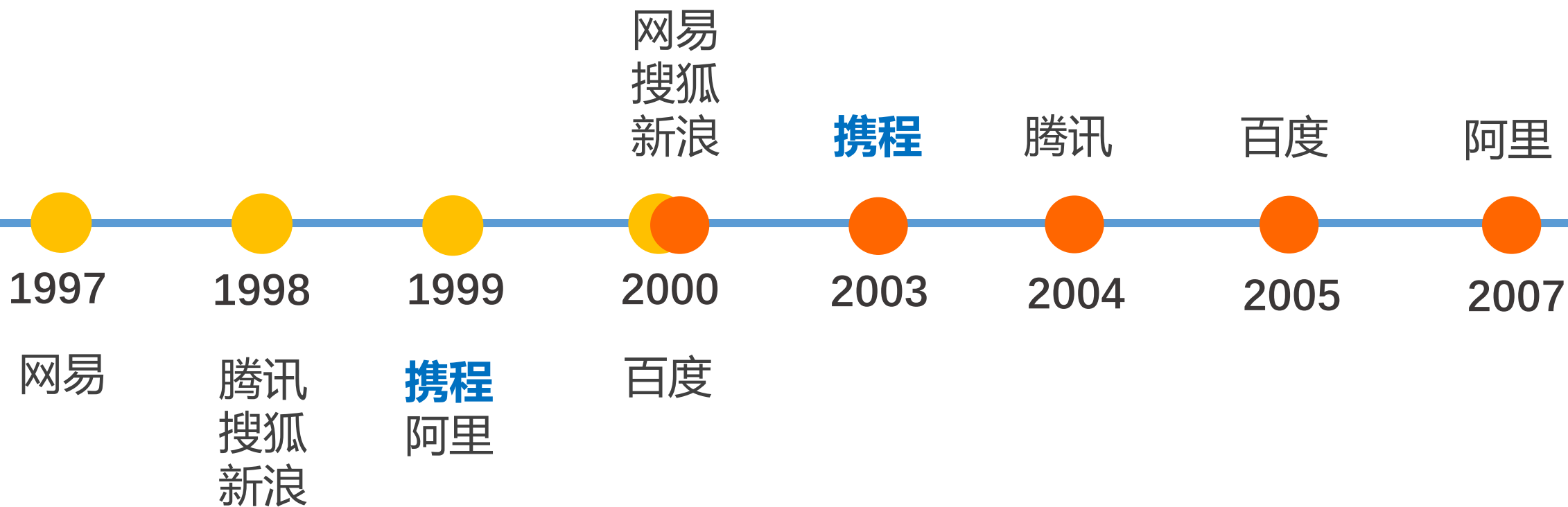


# 卓越运维之路



# 回顾历史



# Sarbanes-Oxley法案(SOX)

2000年初在美国“安然”“世通”等一系列经济丑闻爆发，彻底打击了投资者对美国资本市场的信心，为改变这一局面，美国国会和政府2002年7月通过了SOX法案，其主旨在于“**遵守证券法律以提高公司披露的准确性和可靠性，从而保护投资者及其他目的。**”

法案中最重要的404条款，要求在美上市公司在公司年报中包含一份管理层对内部控制体系有效性的评估报告，报告其内部控制的效力，其中包括利用公司数据通过计算机系统生成报告的可靠性，如果没有有效设计IT系统的控制措施，数据的质量和可靠性就会受到影响。

SOX法案明确要求公司管理层对公司财务信息披露和内部控制效力负有直接责任，**对违反法案的公司主管（CEO、CFO）处罚款高达500万美元，并可判处高达20年的监禁；**

# 合规框架

**COBIT**

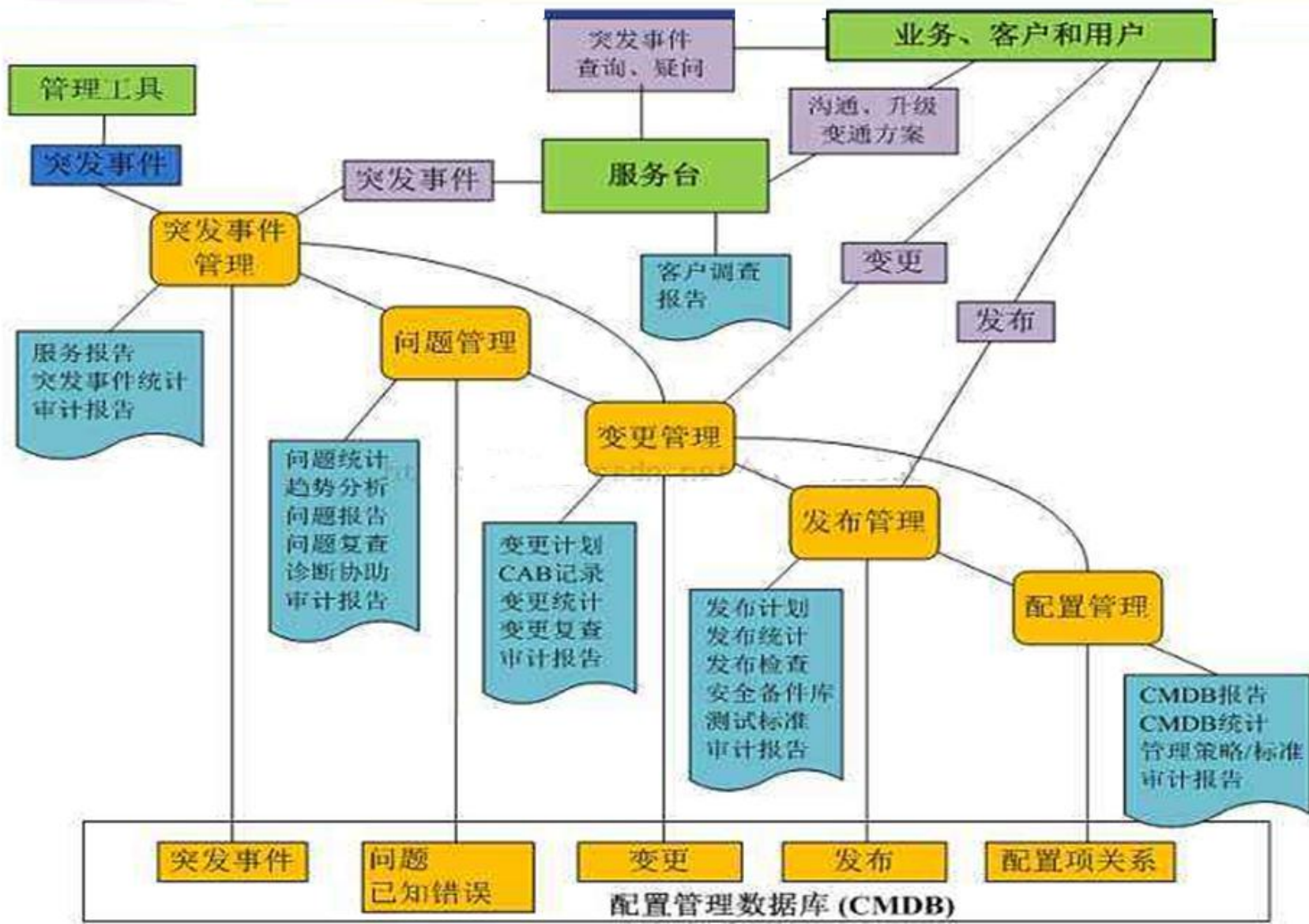
**ISO17799**

**ITIL**

**COSO**

**BS7799**

# ITIL在Ctrip落地



# 基于ITIL的运维-四有

## 有流程

事件管理、配置管理、变更管理、问题管理，采购管理，数据中心管理，服务器生命周期管理等等；

## 有规范

配置规范，巡检规范，机房管理规范，OS配置规范，容器配置规范，安全加固规范；

## 有文档

一个新系统上线，必须交付四篇文档，架构，配置，维护，排错

## 有记录

建立了CMDB，所有所有生产环境的变化都有记录、方便跟踪回溯；

# 基于ITIL的运维-主要问题

## 流程过重

流程引入了多个的角色，申请、实施人、审批人、复核人等，节点并不增值，只是走过场；

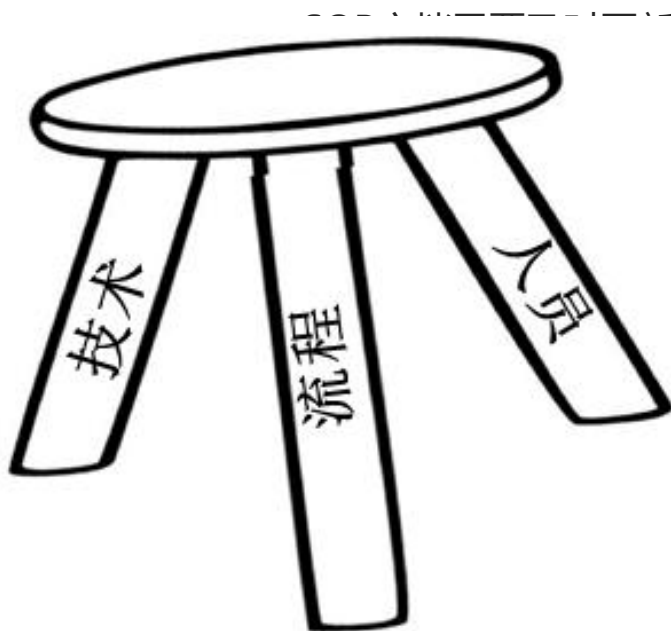
## 流程驱动运维

没有流程定义的日常工作的执行，及绩效标准，没有流程指导工作，工程师会出现无所适从

## 缺乏工具支持

各环节大量人工操作，缺乏有效工具支持，流程流转效率低，各环节衔接耗时长；

## 技术人员非技术负担重



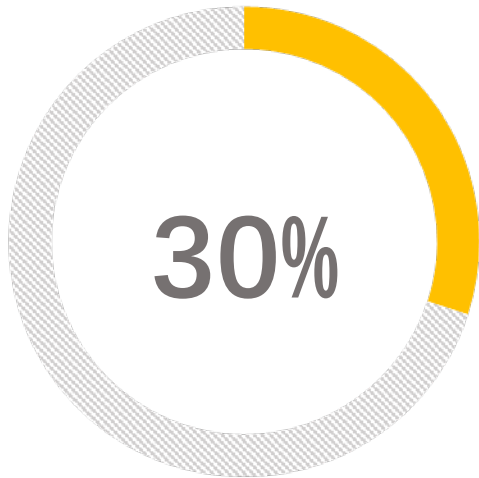
反映实际情况，但往往难以做到，

维护，数据的录入，检查，校

费大量时间，包括，影响范围、

，侧重 流程执行情况，OLA  
质量，标准化变更率，项目交  
对晋升加薪有加分；

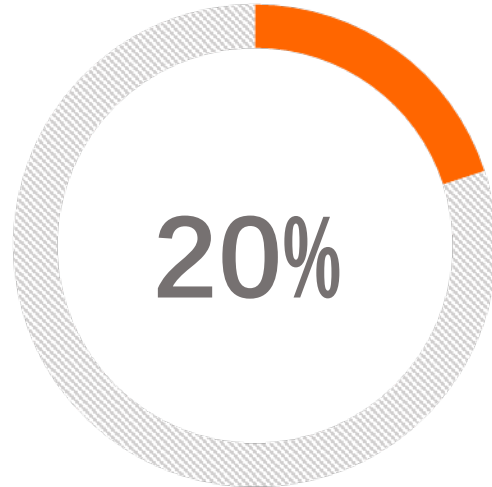
# 2012年前-基于ITIL的运维



## 鼠标+水泥

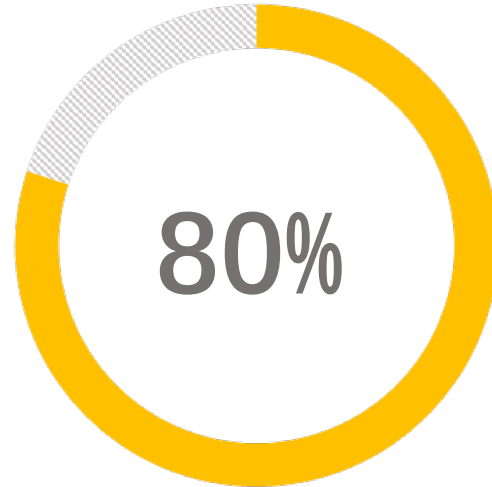


30%预订 PC  
70%预订 电话



## Linux占比

Windows + .Net技术  
栈  
核心DB IOE架构



## 商业/自研

IT管理方案80%为商业  
产品



## 以流程为中心

流程为中心,以控制为  
目标



# 业务战略变化-Mobile First



**大幅增加技术研发投入**  
**调整技术组织架构人员**  
**研发人员数量快速增长**  
**产品迭代周期大幅缩短**  
**用户访问量快速增加**

# 主要矛盾-资源交付效率低



环境创建，物理机、VMs、网络、存储、负载均衡，DNS，基础软件安装，监控添加等均需手工完成，2周的SLA承诺，无法满足业务快速上线的需求；

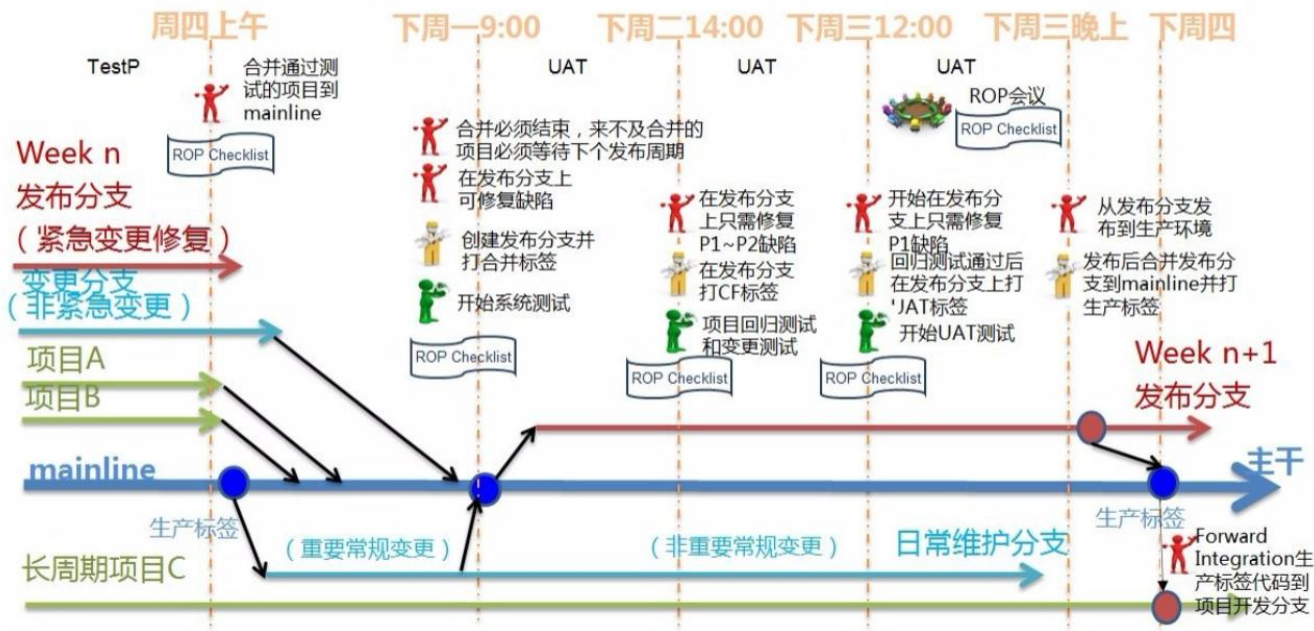
## 工具化、自动化

- 人员结构调整，技能培养
- 工程师文化宣传，绩效考评引导
- 定义全局开发规范，众人拾柴火焰高

## 资源池化

- 到货库存改到货上架
- VM虚拟化推广

# 主要矛盾-应用部署时间长



Train发布方式，常规发布每周1次，18：00进行，但紧急发布频发；  
测试环境与生产环境一致性差，常常导致集成测试耗时长；  
单机多应用混合部署，存在站点冲突，多应用部署需要排队，发布耗时长，小时级；

## 测试环境梳理

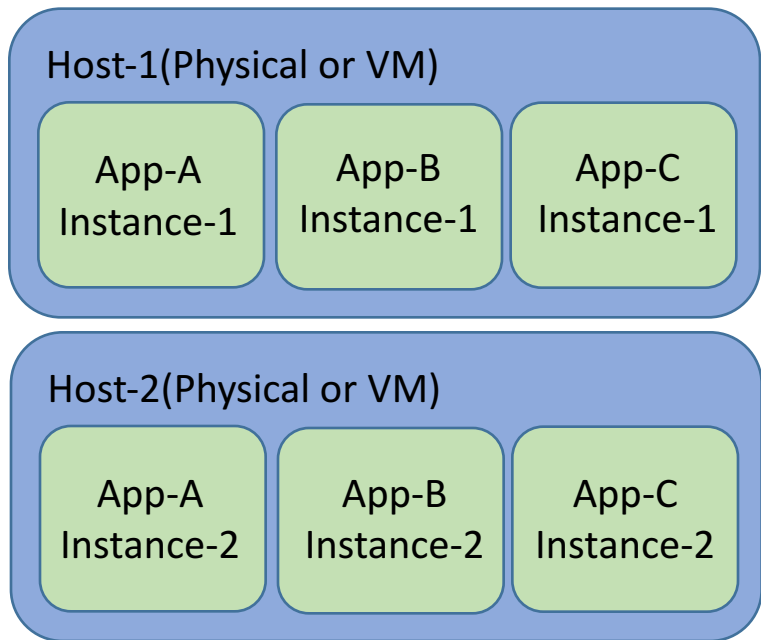
- 测试环境仿真（配置，数据）
- 测试环境可用性提升

## 调整应用部署模式

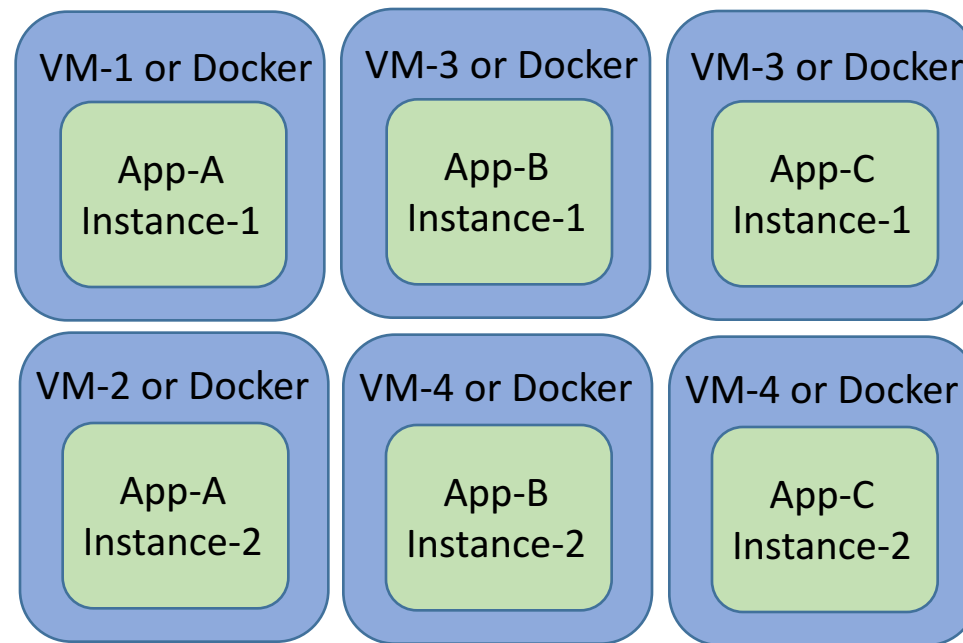
- 以应用为中心的管理模式
- 单机多应用部署专项单机单应用

## 发布流程、工具优化

# 应用部署模式

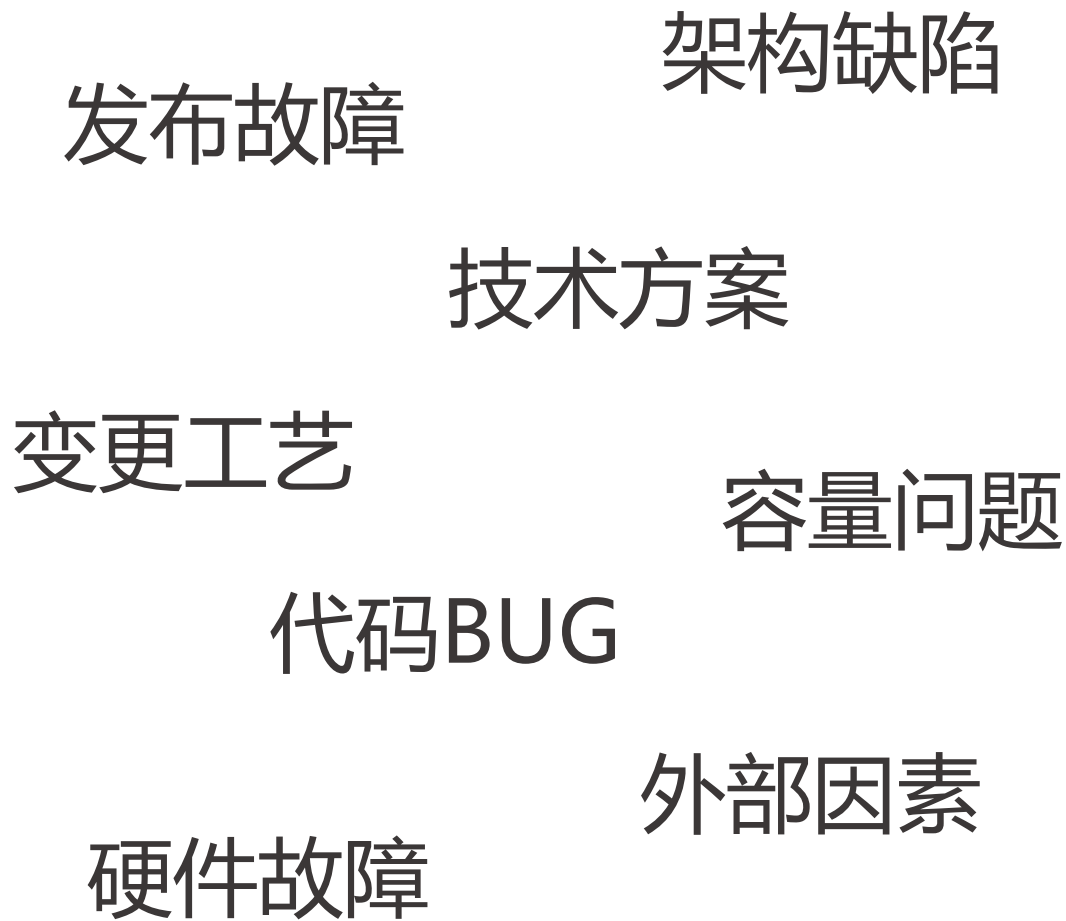


单机多应用混合部署，并共享相同服务端口



单机(容器)单应用部署

# 主要矛盾-可用性



## 故障

- 成立NOC中心，完善应急响应机制
- 优化监控，提升完整性、及时性
- 提升技术人员问题定位能力

## 架构

- 消除架构SPOF
- 建立应用架构评审制度
- 推进架构解耦，消除技术债
- 服务降级能力，熔断限流

## 工具

- 发布刹车防呆
- 自动化频繁、重复的日常操作

## 管理

- RCA复盘机制
- 容量管理
- 发布流程工艺优化
- 代码质量

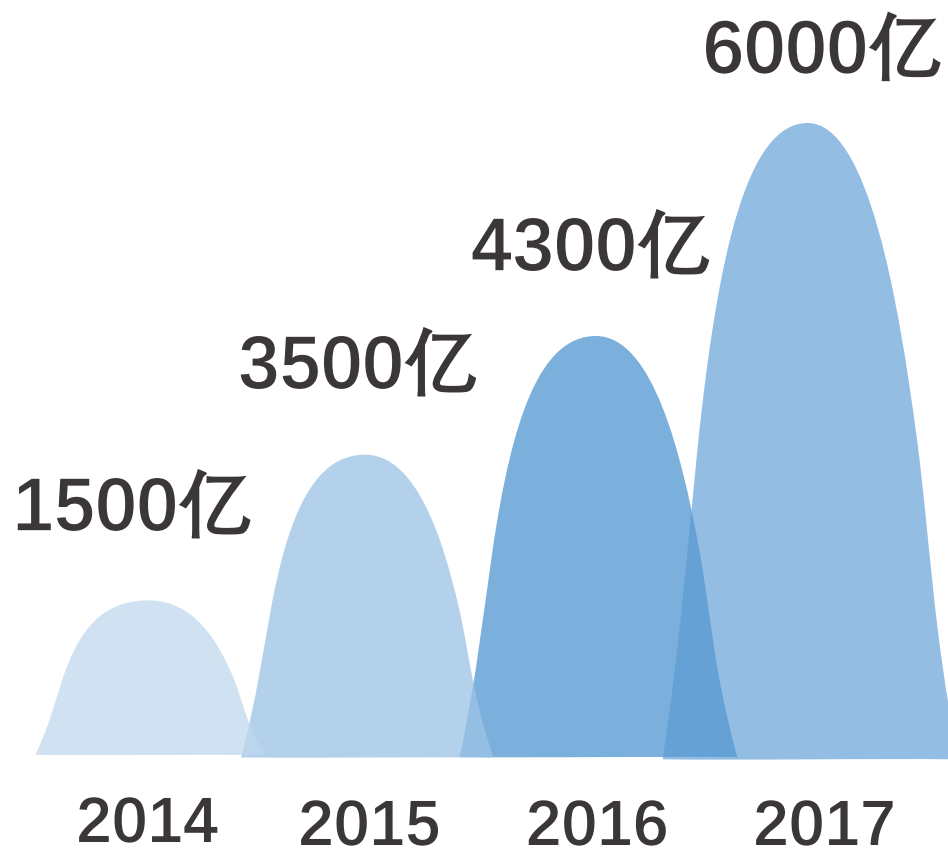
## 文化

- ThinkTwice

# 可用性对Ctrip的重要性

Level of Availability	Percent of Uptime	Downtime per Year	Downtime per Day
1 Nine	90%	36.5 days	2.4 hrs.
2 Nines	99%	3.65 days	14 min.
3 Nines	99.9%	8.76 hrs.	86 sec.
4 Nines	99.99%	52.6 min.	8.6 sec.
5 Nines	99.999%	5.25 min.	.86 sec.
6 Nines	99.9999%	31.5 sec.	8.6 msec

2万/秒



# Ctrip ATP模型

ATP = Availability To Promise



# 遭遇528滑铁卢

如何避免再次发生？

大面积故障如何快速恢复？

能在多长时间内重建整个站点？

全员教育，对生产环境敬畏之心  
梳理所有自动化工具，增加防呆  
机制

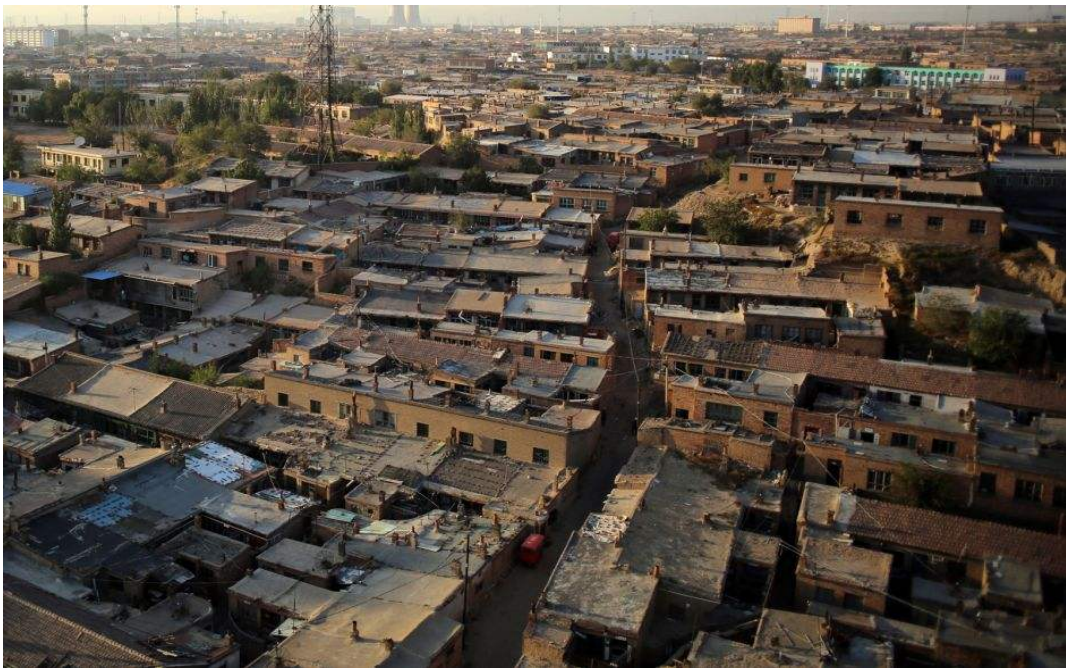
成立DR Program，建设多IDC  
DR Site

建立CD Program，推动CD流  
程、工具重构





# 去技术债



Windows .NET转Linux JAVA

SLB 转SOA服务直连

单机多应用 转 单机单应用



DB去SP

SQL Server转MySql

消除架构SPOF

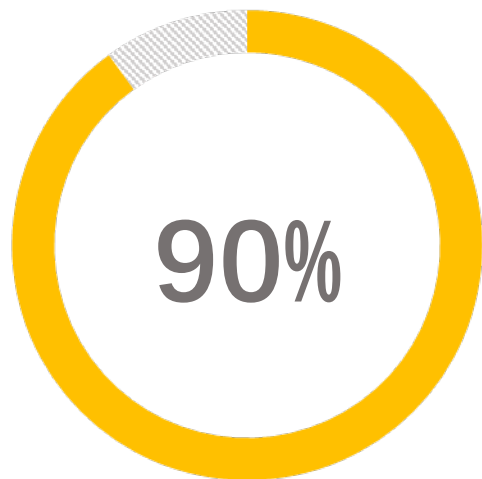
.....



# Ctrip 运维技术体系概览



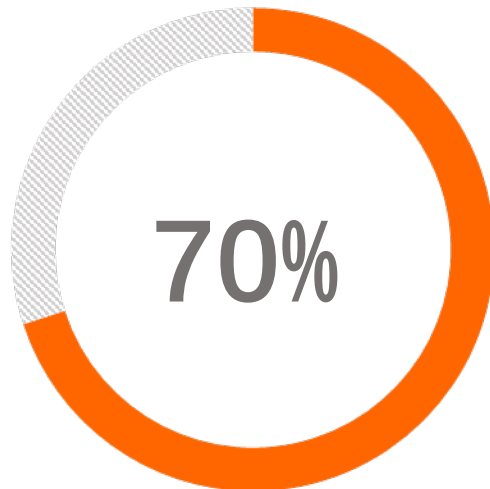
# 现阶段-DevOps



## 拇指+水泥

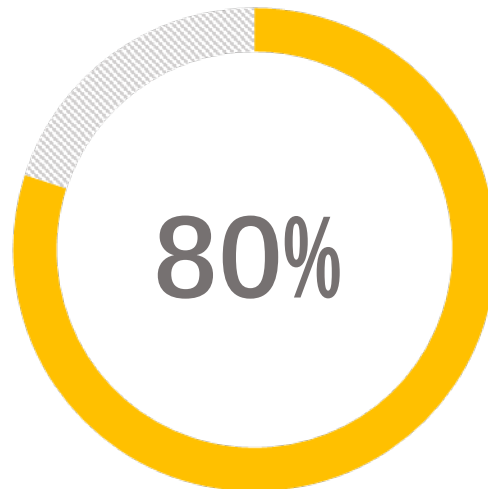


90%预订 手机  
10%预订 PC+电话



## Linux占比

Linux + Java + PHP  
+ Go + Node js /  
Windows + .Net  
MySQL > 60%



## 自研/商业

IT管理方案80%为自研



## 以技术为核心

以质量、成本、效率、  
安全为手段，以促进业  
务发展为目标

# 部分运维数据参考

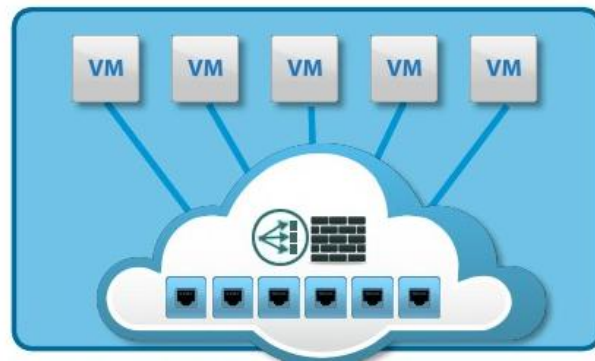
300+	周告警数	11000+
150+	周变更数	6000+
200+	周变发布数	5000+
N/A	网站可用性	>99.97%
N/A	成本 ( core )	1/2 公有云
1200+	服务器实例数	60000+
2200+	应用数量	8000+
服务器 1~2周	交付时长	应用端到端交付 VM : 分钟级 Docker : 秒级
~ 2012		Now

# 部分新进展



## AIOPS

- 异常检测
- 智能告警
- 智能诊断
- 故障关联
- ChatOps
- 容量管理



## Docker/SDN

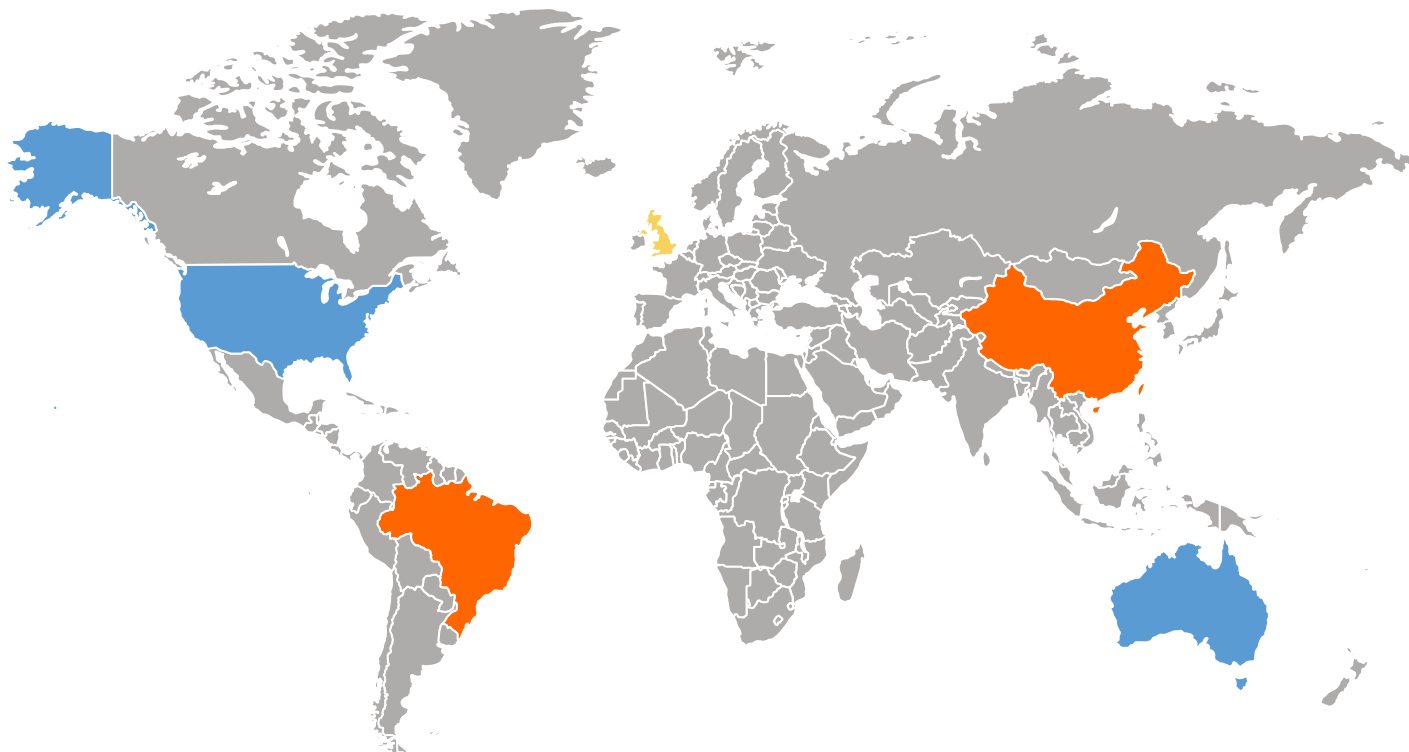
- Redis/DB容器化
- 自研SDN POC



## COST

- 应用画像标签
- Online/Offline混部
- 自动扩容

# 新挑战-Ctrip国际化战略



海外基础设施支持

海外用户体验优化

海外站点建设

海外CallCenter建设

海外法规遵从

海外技术支持

.....

# THANKS !

卓越的本质就是不断改善，持续优化  
我们始终追求卓越！