



# DevOpsDays Beijing

2018 中国·北京



# 运营商研发效能提升践行之道

中国移动杭州研发中心 质量测试部

李伟光



# 目录

- 1 新时代：电信运营商面临的新挑战
- 2 聚焦痛点：探索DevOps提效之路
- 3 从0到1的蜕变：遇见更美的DevOps
- 4 思考：快与慢

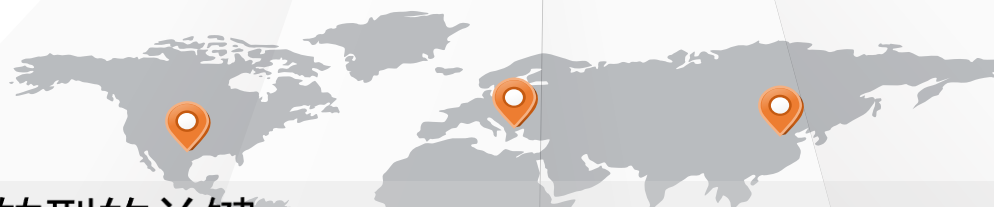


# 目录

- 1 新时代：电信运营商面临的新挑战
- 2 聚焦痛点：探索DevOps提效之路
- 3 从0到1的蜕变：遇见更美的DevOps
- 4 思考：快与慢



# 如何“拥抱”数字化转型



数字化转型的关键：

不仅仅是业务重塑、技术革新，  
更是**研发效率和研发能力的提升。**

## 我们亟待解决的两大挑战



1个待改善  
1个待提升



直接创造价值  
间接创造价值  
从不创造价值



$$\text{过程的有效性} = \frac{\text{创造价值活动的时间}}{\text{整个过程所花费的时间}}$$



测试人员40%时间  
都在等待环境数据



过程标准不规范  
流转沟通成本高



每次上线发布前  
要提交各种文档



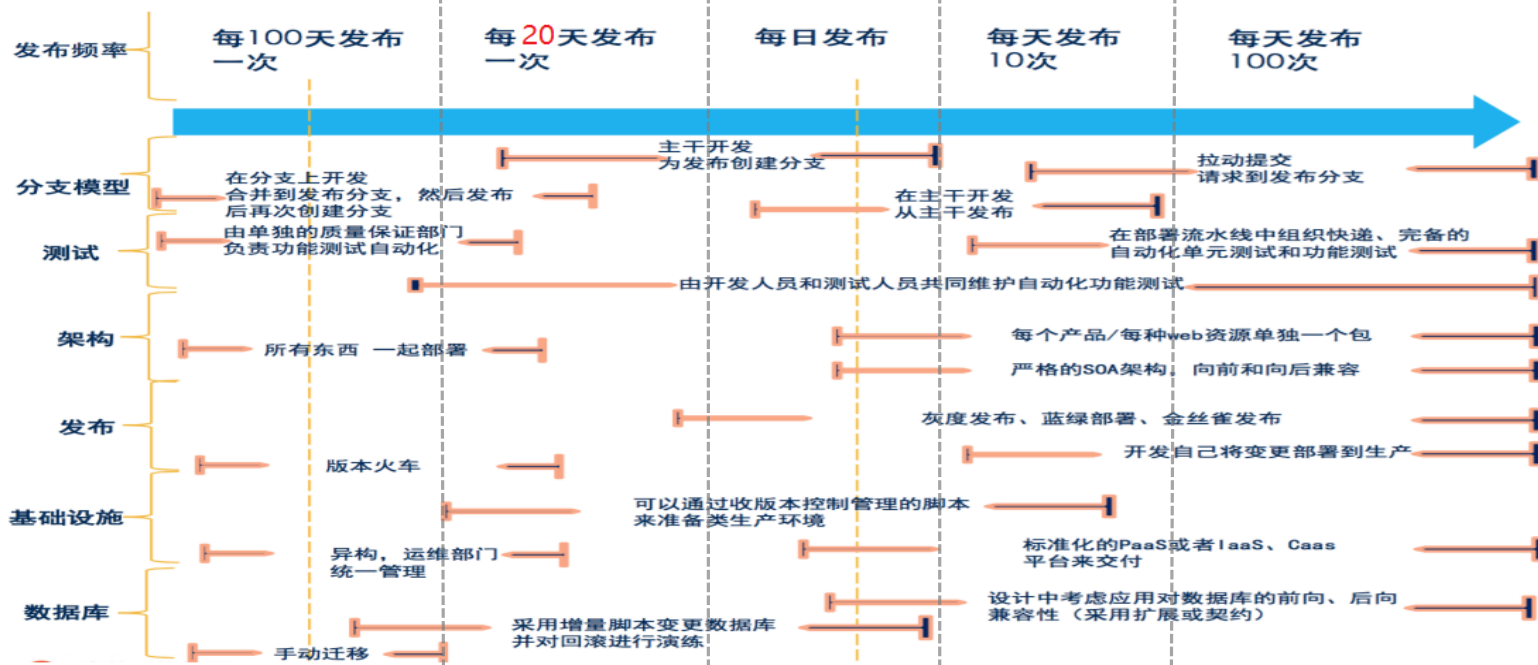
# 组织级效能待提升

电信行业  
软件类公司  
(月)

中移杭研  
(周)

国内  
互联网巨头  
(小时)

国际  
互联网巨头  
(分钟)



# 目录



1 新时代：电信运营商面临的新挑战

2 聚焦痛点：探索DevOps提效之路

3 从0到1的蜕变：遇见更美的DevOps

4 思考：快与慢





# 普遍存在的4个痛点问题

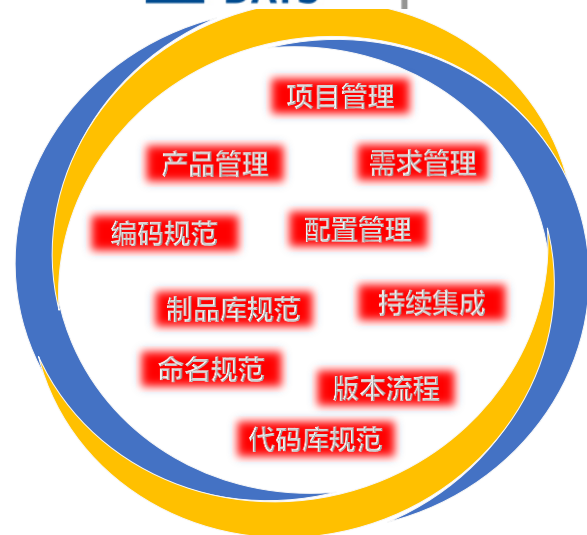


精益七大浪费  
全部命中

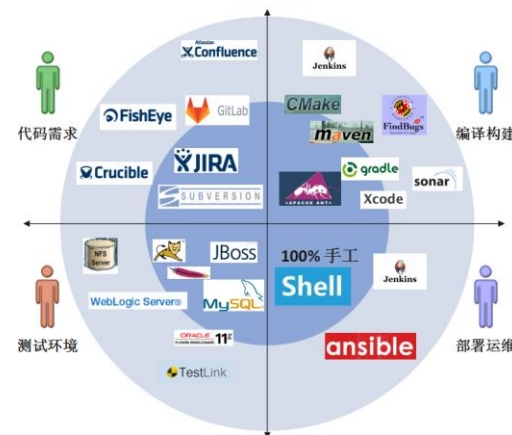
流程标准缺失  
沟通靠人

环境部署  
自动化程度低

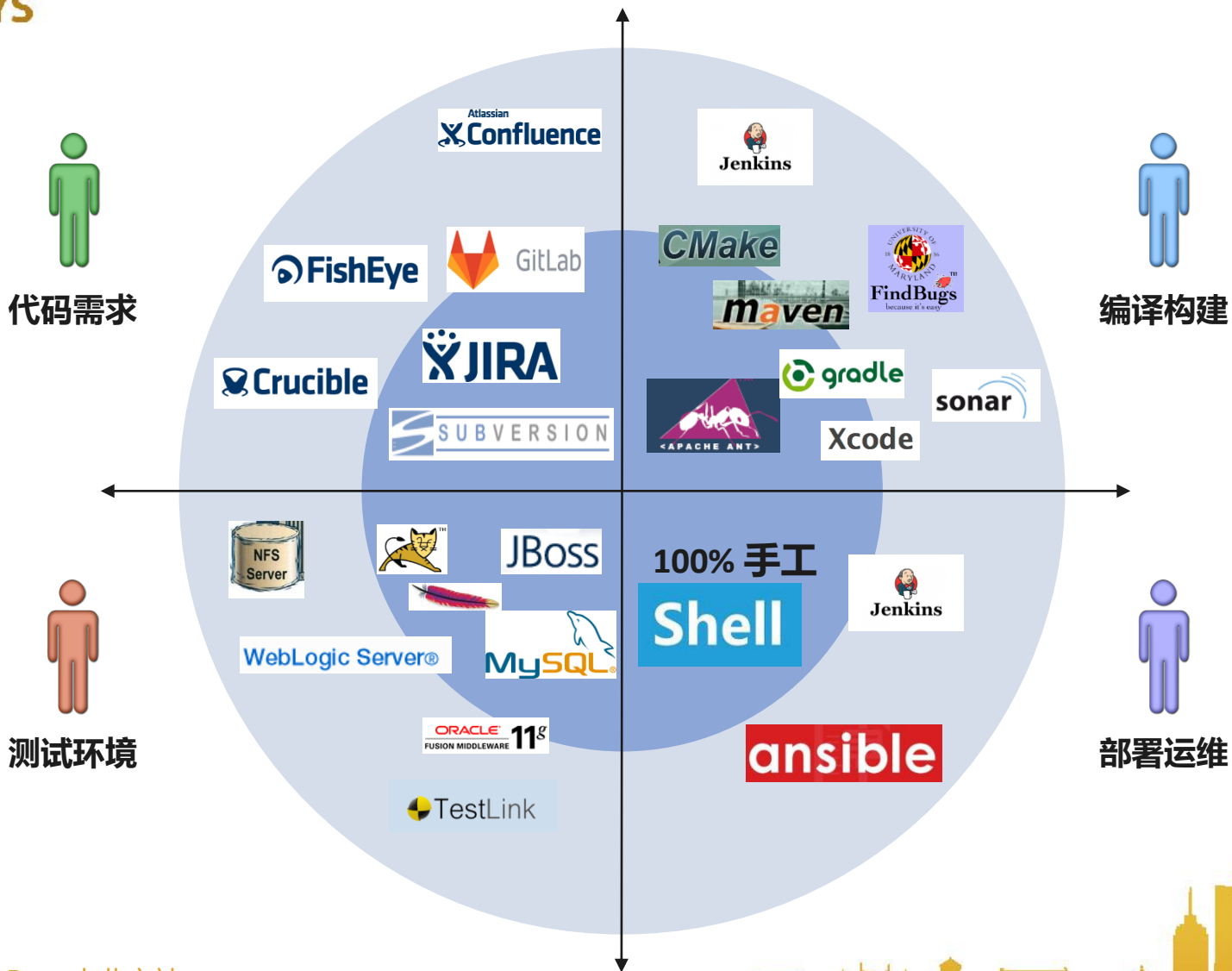
质效难控  
研发工具松散



<p><b>开发测试环境多</b></p> <p>开发环境、联调环境、功能测试环境、验收测试环境、性能测试环境、stage环境等等。</p>	<p><b>重新部署效率低</b></p> <p>当出现问题需要重新部署时，要重复一次完整的流程。</p>
<p><b>多人的依赖较大</b></p> <p>每套环境的每次部署都要人工部署。比较耗时，而且复制不易，资源紧张，利用率低下。</p>	<p><b>回滚历史版本难</b></p> <p>因为没有对历史交付件进行管理，因此想回退到某个历史版本十分困难。</p>
<p><b>不同服务器配置差异</b></p> <p>因为人的因素，不同服务器的环境配置难免存在差异，未开发测试、排查问题增加了难度。</p>	<p><b>影响开发测试效率</b></p> <p>以上问题，造成开发、测试人员的大部分精力都在环境上，极大地影响了开发测试的效率。</p>

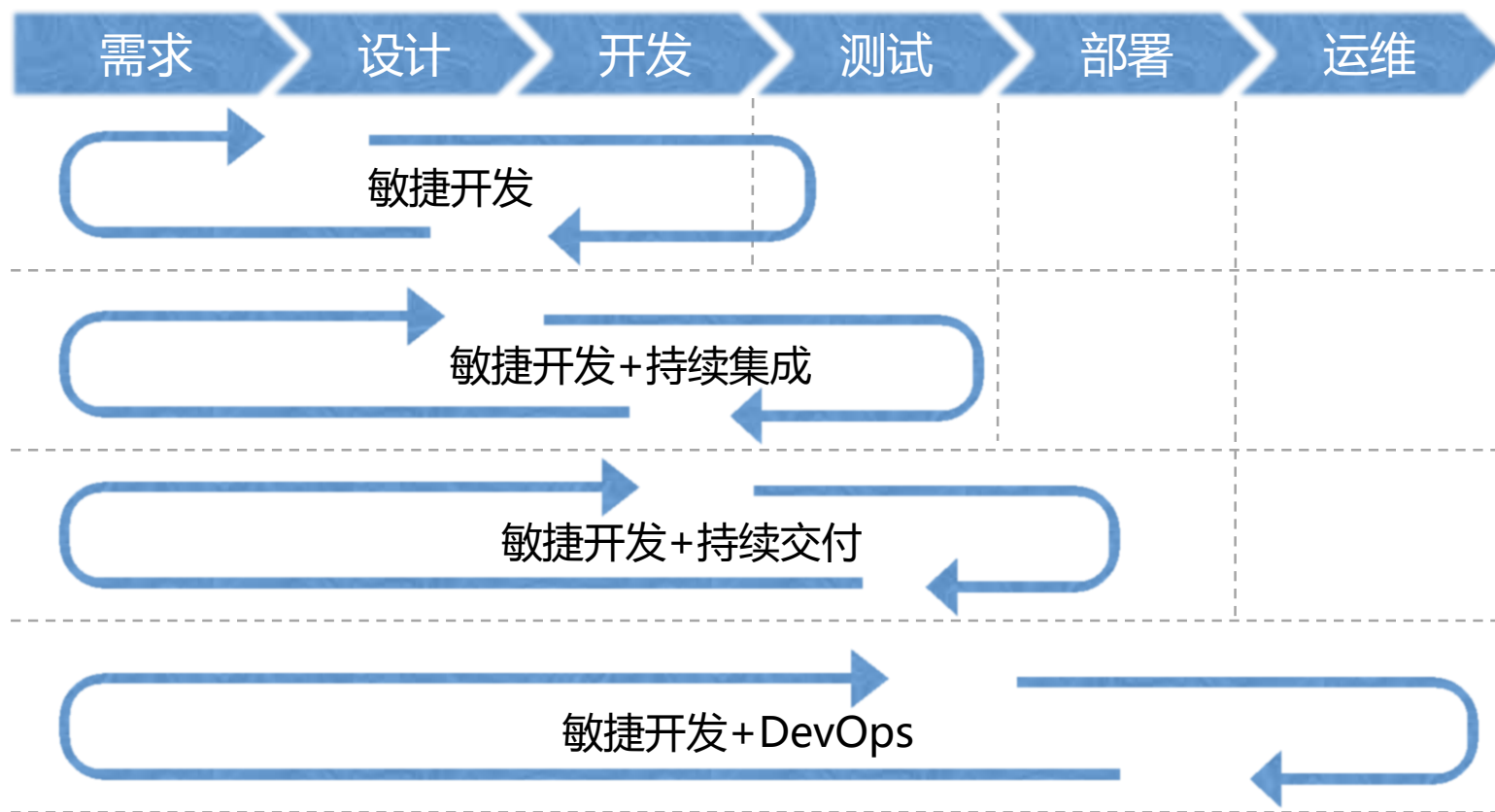


# 松散的工具 ≠ 效能的提升

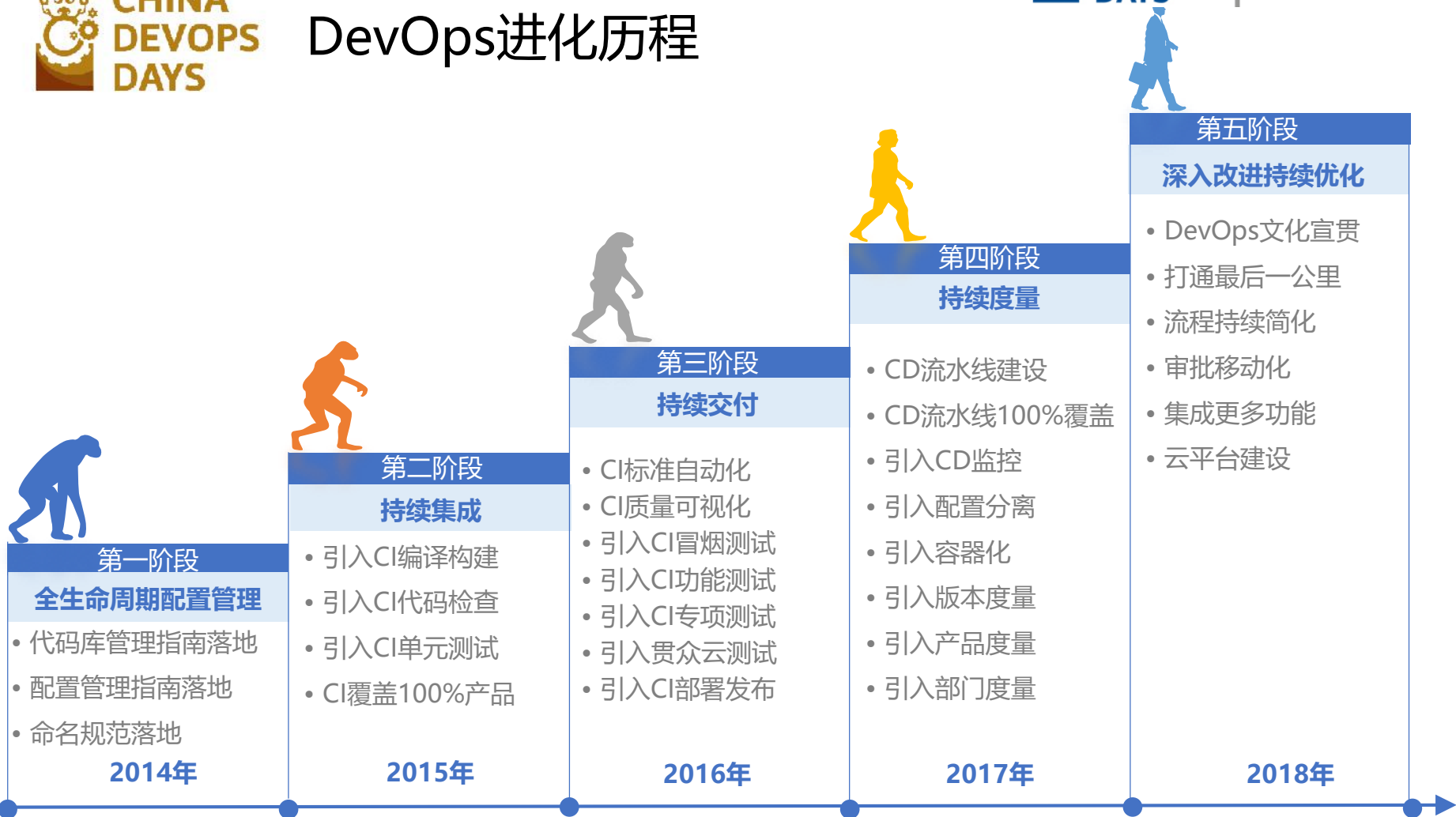




# 为什么是DevOps?



# DevOps进化历程



# 目录



1 新时代：电信运营商面临的新挑战

2 聚焦痛点：探索DevOps提效之路

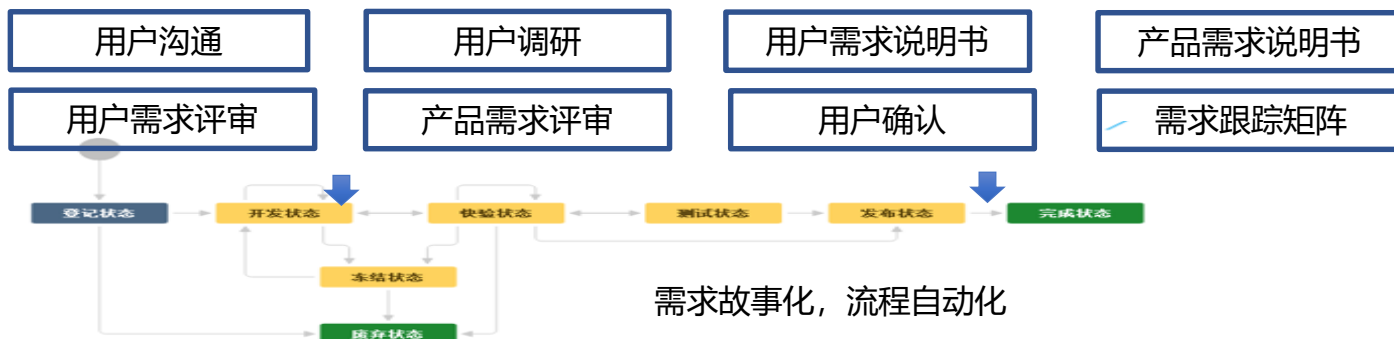
3 从0到1的蜕变：遇见更美的DevOps

4 思考：快与慢

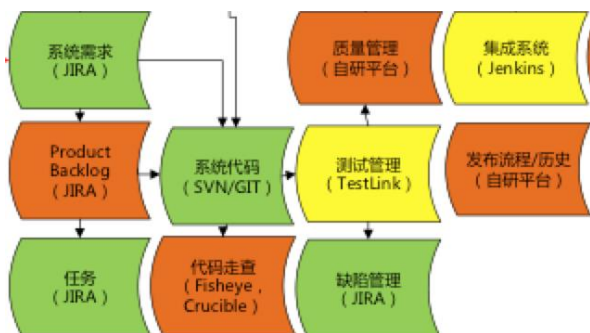


# 需求管理：抓住要素，化繁为简

流程改造



关联和追溯



以jira 需求key为唯一识别号，建立全过程关联和追溯：

- 需求和代码关联
- 代码和代码评审关联
- 代码和代码检查关联
- 需求和测试用例关联
- 需求和测试缺陷关联
- 需求和提测、发布审批关联

全景视图

功能模块数 38 | 需求总数 256 | 进行中的需求数 93

准备状态 开发状态 快验状态 测试状态 发布状态 完成状态 废弃状态

全景视图展示产品全貌和价值流动

准备状态	开发状态	快验状态	测试状态	发布状态	完成状态	废弃状态
<b>• PCC运营流程</b> 1188: 二级POP-D平台策略流程 about 1日 22h 1189: 二级POP-D平台策略流程 about 1日 12d 1190: 二级POP-R/POP-D之间的流转流程 about 1日 1d	1188: 二级POP-D平台策略流程 about 2日 1189: 二级POP-D平台策略流程 about 2日 1190: 二级POP-R/POP-D之间的流转流程 about 1日	1188: 二级POP-D平台策略流程 about 2日 1189: 二级POP-D平台策略流程 about 2日 1190: 二级POP-R/POP-D之间的流转流程 about 5日	1188: 二级POP-D平台策略流程 about 2日 1189: 二级POP-D平台策略流程 about 2日 1190: 二级POP-R/POP-D之间的流转流程 about 5日	1188: 二级PCC策略运营平台_V2.4.3 优化二级POP-D平台策略流程 about 2日 28d 1189: 二级PCC策略运营平台_V2.4.2 新增二级POP-R/POP-D之间的流转流程 about 5日 17d	722: 二级POP-R/D优化对于全网网络... 713: 二级POP-D新增省级网络策略... 666: 对于全网策略、省级策略的生命... 665: 二级POP-R/D全网策略运营流程... 664: 二级POP-R/D省级运营流程优化	719: 二级POP-D增加二级POP-R传递... 714: 二级POP-D新增省级策略指示...
<b>• POP-D FOA测试</b>						

## 1. 将版本控制链接到整个周期，实现可追溯可度量



## 2. 配置管理基础工作以自服务形式开放

资源申请  
权限助手  
安全审计  
账号运维  
证书运维  
资源下载  
统计报表  
客服帮助

开发工程统计

部门	项目名称	姓名	IP地址	国籍	所属部门/地区	所属部门/地区	工作地点	职位名称	职位类别	提交代码行数	投入工时	每月代码量
智慧医疗产品部								web前端开发工程师	软件开发	1843	14	131.91
传统应用产品部							iOS开发工程师	软件开发	软件开发	1403	15	93.53
传统应用产品部							Android开发工程师	软件开发	软件开发	1476	11	134.11
智慧医疗产品部							web前端开发工程师	软件开发	软件开发	27257	15	1,817.1
中央产品部							web前端开发工程师	软件开发	软件开发	50896	14	3,615.4
智慧医疗产品部							web前端开发工程师	软件开发	软件开发	2864	10.5	272.71
传统应用产品部							java开发工程师	软件开发	软件开发	533	17	31.35
智慧医疗产品部							web前端开发工程师	软件开发	软件开发	10547	13	842.01



# 持续集成：频繁集成、快速迭代反馈优化

- 任务配置：将任务模板化配置化，通过统一入口进行创建维护。
- 任务监控：识别低效任务，推送责任人优化，主动驱动构建加速。
- 资源调度：节点资源实现动态调度，解决等待，提高资源利用率。
- 质量报告：从组件、产品、部门3个维度，展现实时代码质量情况。

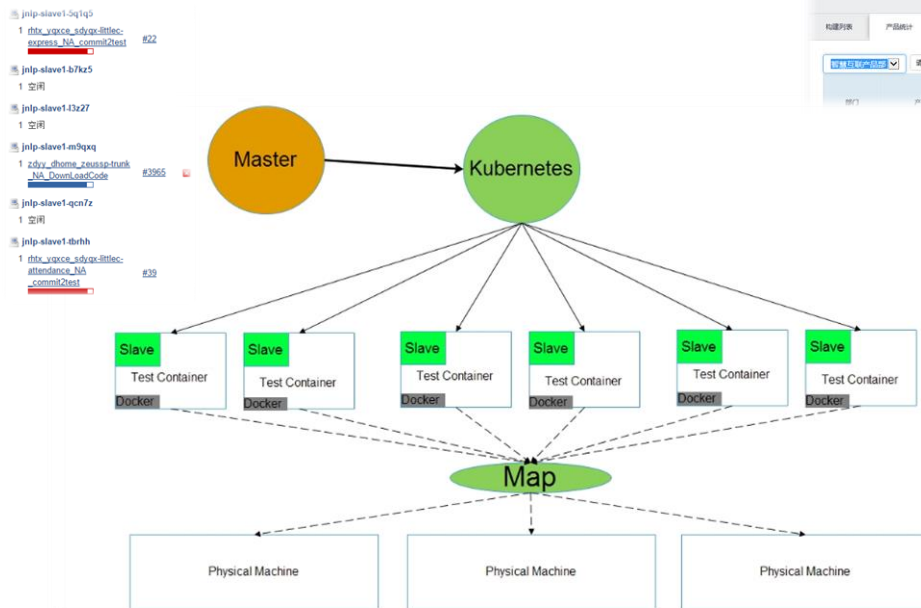


PROJECT 项目管理 产品管理 测试管理 报告

当前位置：报告 > 持续集成 > 每日构建

选择日期范围:  请输入产品名称:

部门	产品	代码量				单元测试 行覆盖率	静态代码扫描缺陷数						
		总量	有效	核心	占比		JAVA-FB-H	JAVA-FB-N	IOS-OC	C/C++-Error	空行	JAVA-H	JAVA-N
		126113	104548	51434	49.29%	80.21%	0	0	0	NA	0	0	0
		97787	57812	41248	71.35%	86.76%	0	0	0	NA	0	0	0
		24892	24892	8612	34.60%	84.00%	0	0	NA	NA	0	0	0
中组		39095	39095	18168	46.47%	80.00%	0	0	NA	NA	0	0	0
		105629	75254	15403	20.47%	88.80%	0	0	0	NA	0	0	0
中组		134392	61295	33146	34.08%	83.24%	0	0	NA	NA	0	0	0
		150938	288773	122747	42.51%	81.87%	0	0	NA	NA	0	0	0



模块名称: marketing 版本初始日期: 2017-07-03

### 提交阶段

	编译	静态检查	单元测试
构建次数	11	11	11
成功次数	11	11	11
成功率	100%	100%	100%
平均耗时	112s	189s	71s
FB HN		0	
CS HN		0	
有效代码行			17264
核心代码行			9285
行覆盖率			84%

### 冒烟测试阶段

	打包/部署	冒烟测试
构建次数	2	1
成功次数	1	1
成功率	50%	100%
平均耗时	378s	591s
第一次转耗时	2560s	187s

### 功能测试阶段

	功能测试
构建次数	1
成功次数	1
成功率	100%
平均耗时	0.03h
发现缺陷数	0
总耗时	0.03h
用例总数	9
通过用例总数	9

### 构建详情

#84 qinqinyan 5fd6f9/5fd6f9 2017-07-28 00:22:38 http://git.komect.net/opp/marketing.git

请输入流水线号或revision



# CI/CD流水线：聚焦价值交付



以持续、快速交付有质量的产品为目标，版本控制、自动化、内建质量和持续改进为核心，提炼产品交付价值流，建立质量门，打通并推广流水线，形成产品交付自动化、版本质量和效率显性化能力

流水线名称	对应组件	创建者	流程	最近执行时间
marketing_gate	marketing	haozhenyu-ry	编译 静态检查 单元测试 分支 打包/部署 冒烟测试 功能测试 	2017-08-01 15:56:50 <a href="#">查看详情</a>
freeflow_gate	freeflow	haozhenyu-ry	编译 静态检查 单元测试 分支 打包/部署 冒烟测试 功能测试 安全测试 	2017-11-16 09:01:00 <a href="#">查看详情</a>
freeflow-server_gate	freeflow-server	haozhenyu-ry	编译 静态检查 分支 打包/部署 冒烟测试 功能测试 	2017-10-26 09:01:00 <a href="#">查看详情</a>
freeflow-web_gate	freeflow-web	haozhenyu-ry	编译 静态检查 单元测试 分支 打包/部署 冒烟测试 功能测试 兼容性测试 	2017-11-17 10:38:00 <a href="#">查看详情</a>



持续交付流水线在各产品部全面落地，已持续应用于1580多个产品版本，总构建次数百万级别

实现代码提交到测试完成端到端触发和融合

只有提交阶段检查通过的代码才能被正式提交和测试，避免技术债的累积

通过对测试自动化程度、测试轮次、发现缺陷数等执行监控，促进快速反馈环建立和自动化程度提升

测试根据需要任意触发进入到测试候选列表版本，实现版本选择的自服务

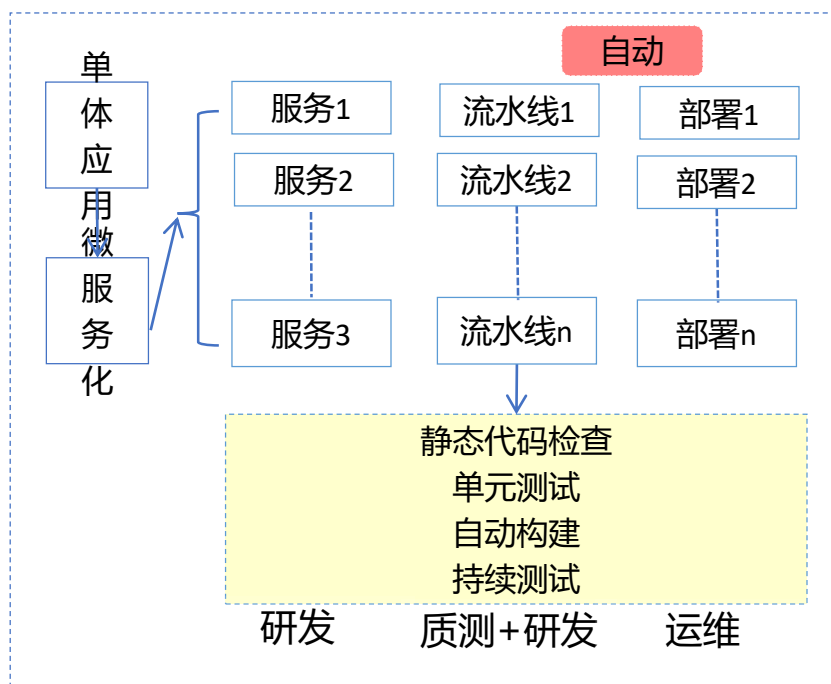
微服务架构引导

全过程监控和研发效率显性化



# 环境管理：从测试环境开始尝试容器化

微服务化导致环境部署更具挑战



## 更大的挑战

- 微服务化改造使得需要部署的模块更多
- 部署环境时配置、依赖将变得更加复杂
- 环境多：Dev、QA、Stage、Production
- 持续审查、持续构建，持续测试的压力更大

## Kubernetes + Docker

- 提供诸多能力，如高可用性、自我修复、弹性伸缩、服务发现、负载均衡、配置管理等

对于新接触的技术，有疑虑！从测试环境开始！质量测试部从2016年Q4开始  
Kubernetes+Docker的实践

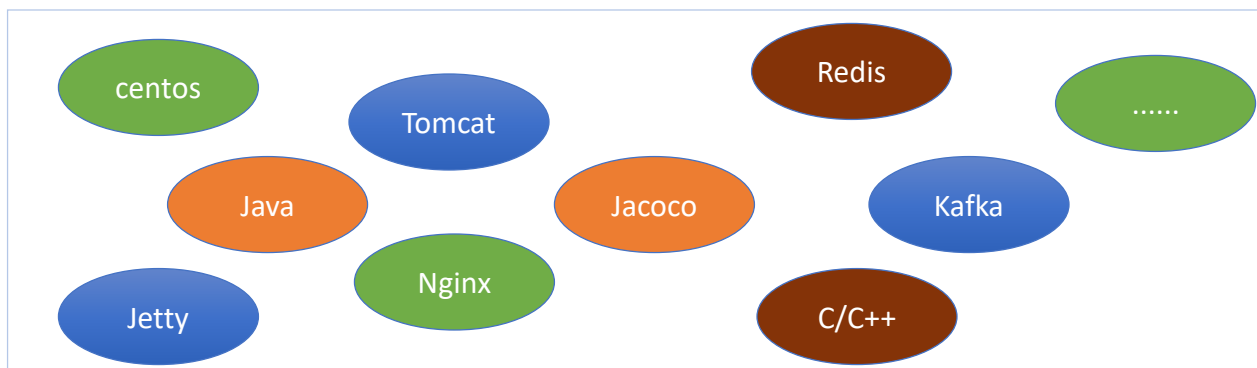


# 环境管理：从测试环境开始尝试容器化

## 历程



## 基础镜像



## 应用创新

- 一键创建压测集群环境
- 一键创建中间件集群环境

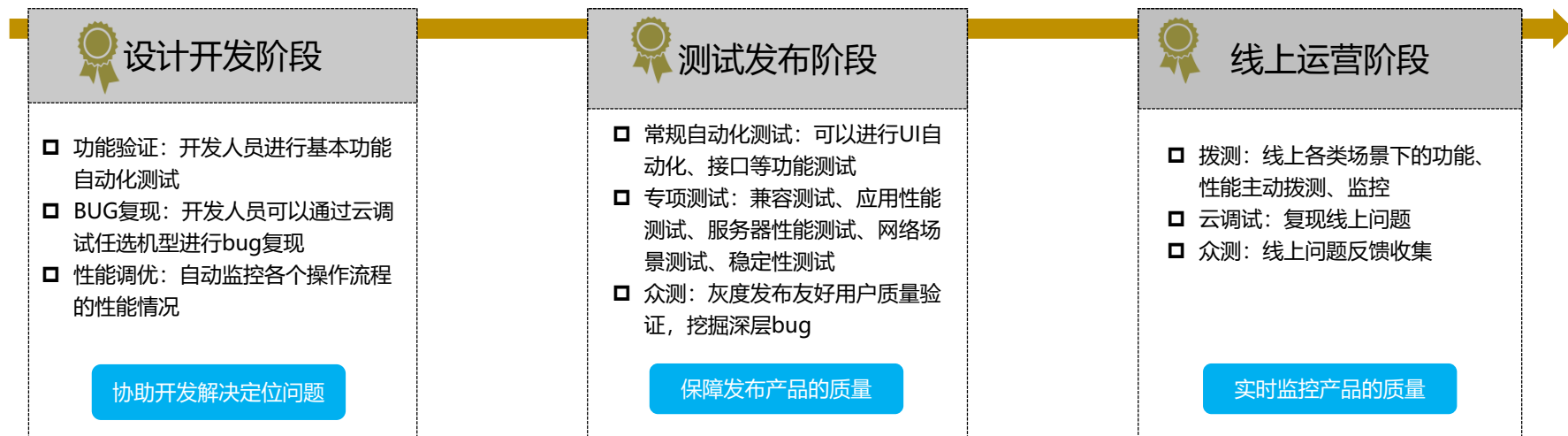
## 接着做

- 研发生产环境容器化
- 容器组件标准化
- 一次创建到处运行

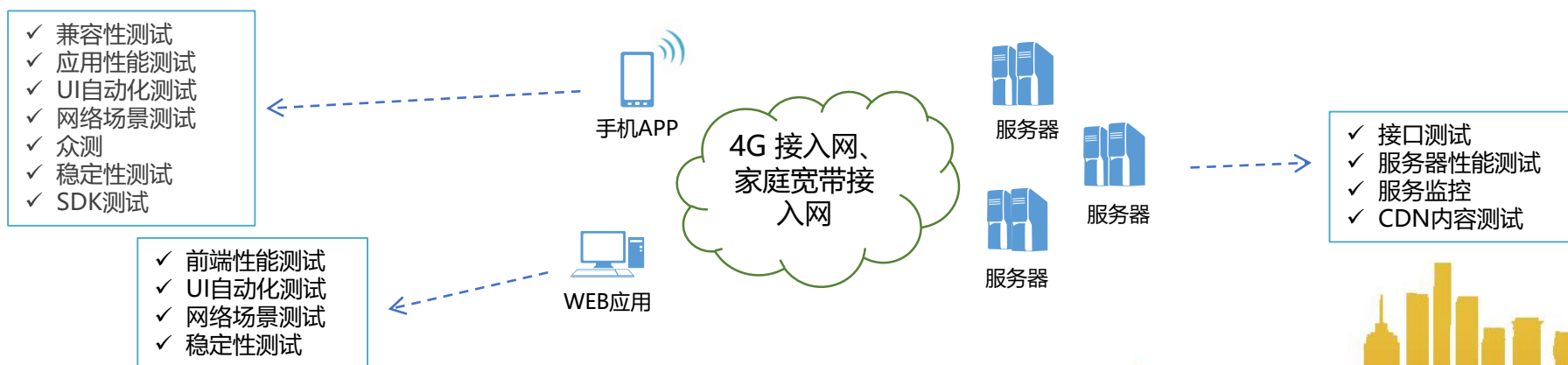


# 测试管理：贯众云测覆盖研发各阶段

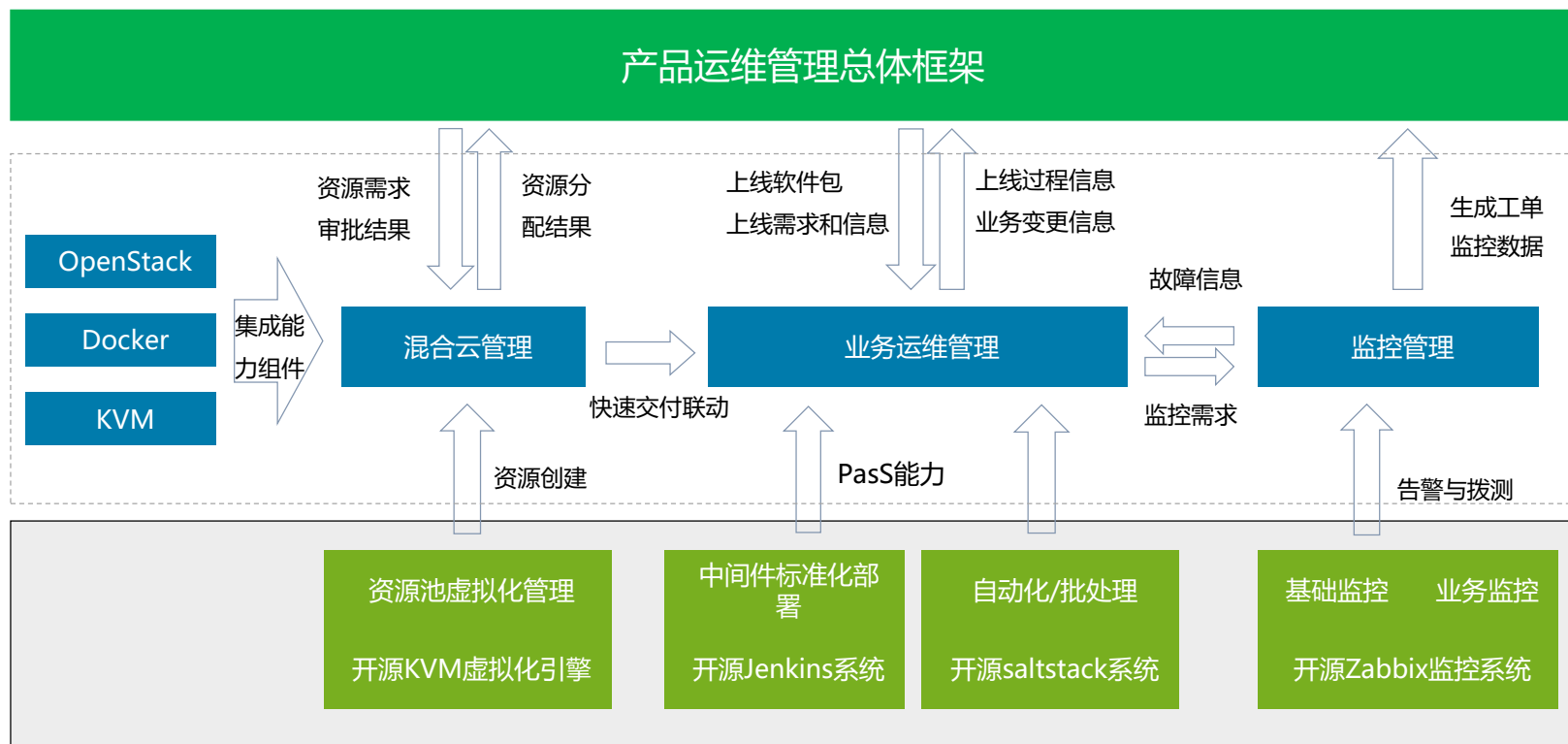
- 中国移动自研的贯众云测试自动化测试平台目前已经全面覆盖了开发、测试、运营研发流程各阶段，全面保障产品的质量



- 贯众云测试平台覆盖了APP应用、WEB应用和服务端



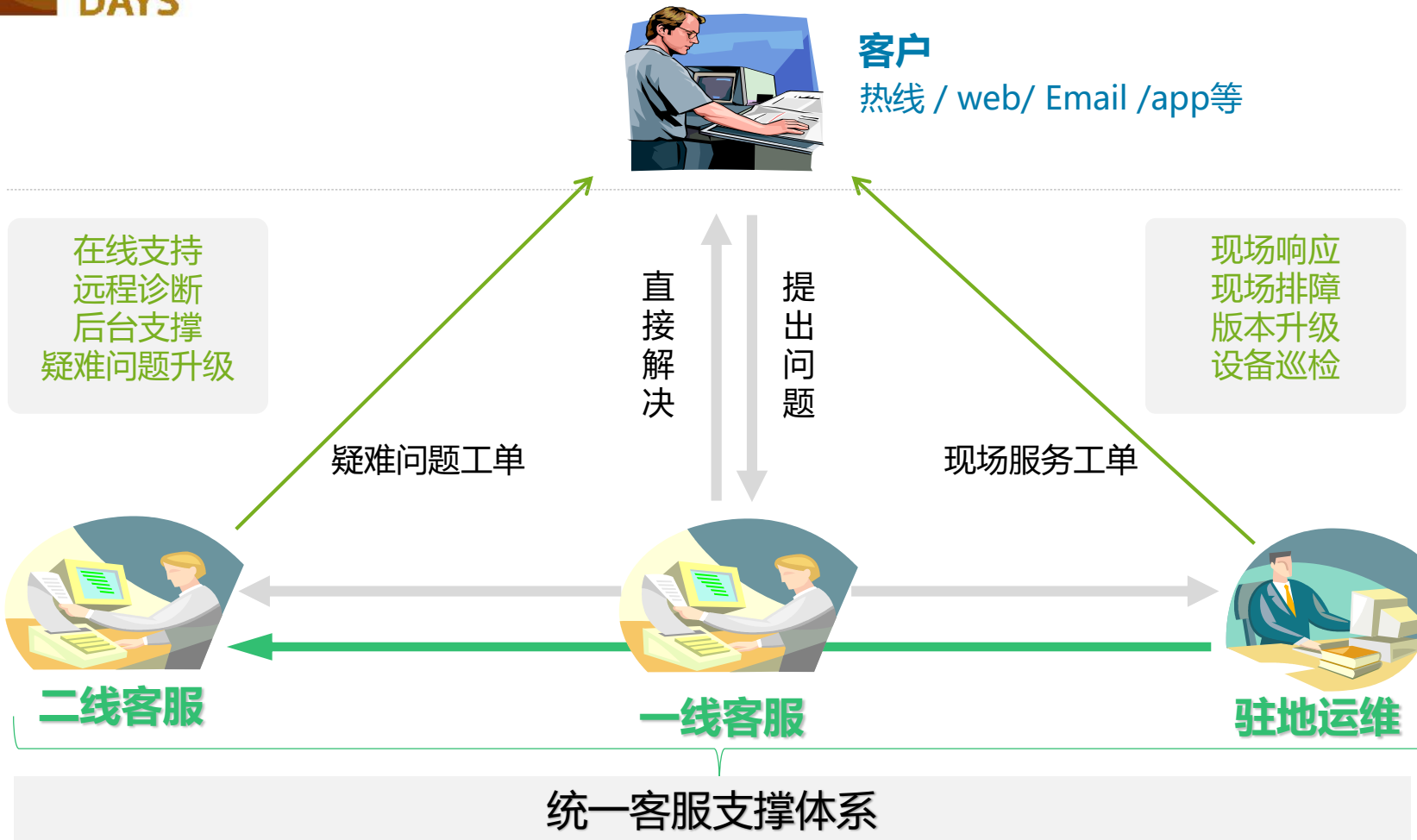
基于混合云管理子、业务运维管理、监控管理的子框架，完善“运维过程一体化”。



- 资源一体化管理，缩短交付周期。
- 各类开源运维工具的平台化，故障处理的集中化。
- 故障信息在线反馈，完善监控覆盖。



# 三级客服支撑体系





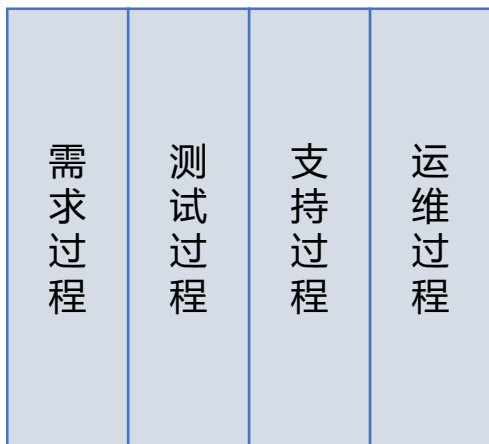


# 度量反馈：从朴素度量到精益度量

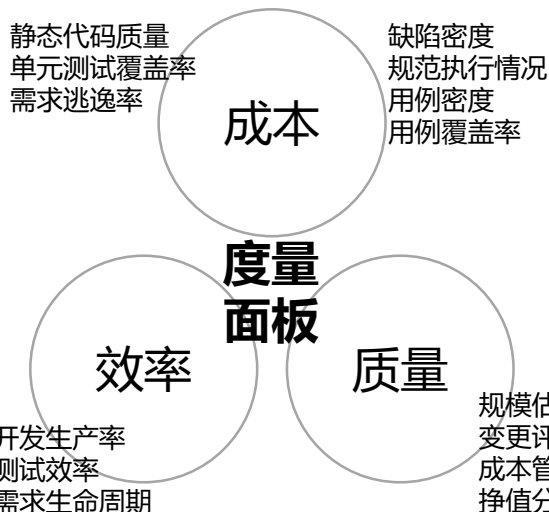
## 2016年 基于过程的度量

## 2017年 基于目标的度量

## 2018年 基于价值的度量



通过分散的各类工具，收集过程数据，包括需求、测试、支持、运维几个环节



通过集中化的度量面板，显性体现组织性能，包括成本、效率、质量几个方面



以价值交付为导向定制精益看板，层层拆分过程中各个环节，找到浪费并消除，实现价值最大化，加速流动。

结合持续交付和精益思想的理念，通过度量面板，驱动改进，消除浪费增加价值，提高研发过程的效率和质量。

# 4级度量、反馈与改进

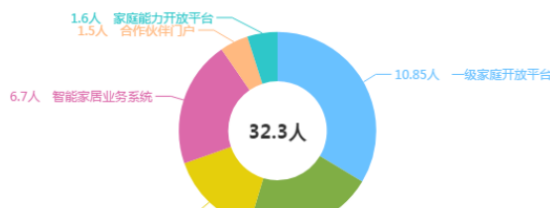
- 1.版本级、产品级反馈即时推送。
- 2.部门级反馈每月推送。
- 3.组织级反馈每月形成质量公文。

## 2018年3月产品质量报告

为推进产品质量和组织效能的提升，质量测试部通过收集研发过程数据、线上质量数据等，对产品研发各过程域进行了质量分析，并给出了相关的改进建议。同时，基于业界 DevOps 最佳实践，对杭研的现状也进行了对标，帮助我们更好的审视当前研发能力的成熟度阶段。重点产品方面，部门在2018年将重点开展对标测试的工作，从竞品角度发现产品的待改进点，不断推动产品的持续改进。

现发布质量情况数据3月刊，供各部门查阅与参考。

### 一、公司整体情况



内部运维热线支撑：666119

Apps问题反馈群支撑：小移人家



DevOps专题系列之  
“DevOps的初识” **第一期**

各位杭研小伙伴，DevOps专题推广系列开始啦！  
DevOps小君将持续推出DevOps相关专题，帮助你掌握DevOps的主要知识和实践。



DevOps专题系列之  
“DevOps的初识” **第二期**

各位杭研小伙伴，DevOps专题推广系列开始啦！  
DevOps小君将持续推出DevOps相关专题，帮助你掌握DevOps的主要知识和实践。



DevOps专题系列之  
“DevOps的初识” **第三期**

各位杭研小伙伴，DevOps专题推广系列开始啦！  
DevOps小君将持续推出DevOps相关专题，帮助你掌握DevOps的主要知识和实践。

我们今天重点聊三个话题：

我们正在经历的时代

DevOps是什么？如何理解

DevOps

今天我们要聊的是：

敏捷需求管理

今天我们要聊的是：

持续集成



# 目录

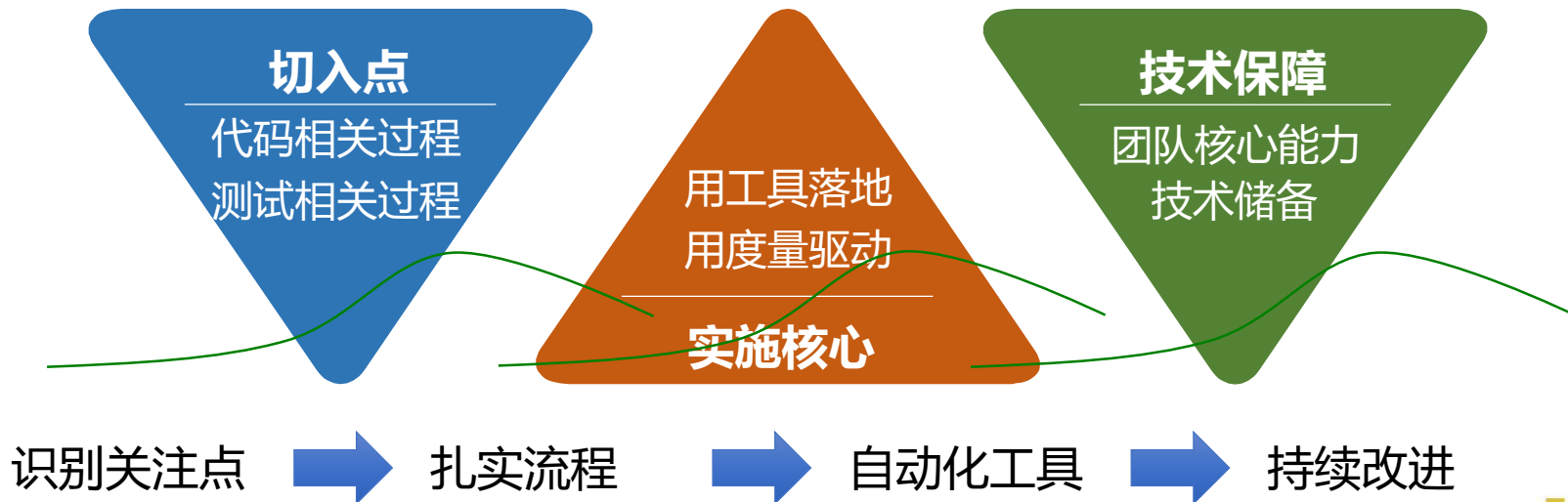
- 1 新时代：电信运营商面临的新挑战
- 2 聚焦痛点：探索DevOps提效之路
- 3 从0到1的蜕变：遇见更美的DevOps
- 4 思考：快与慢



## 经验小结



以快速实践反馈环的问题驱动，快速学习实践落地，形成持续学习改进闭环。



# 未来：全面的DevOps深度实践

## 小目标

实现有质量的按需部署（小时级部署前置时间）

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	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	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	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					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								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 ) 安全管理														
四、研发运营一体化 ( DevOps ) 组织结构														

图 1 研发运营一体化 (DevOps) 标准总体架构





# THANKS