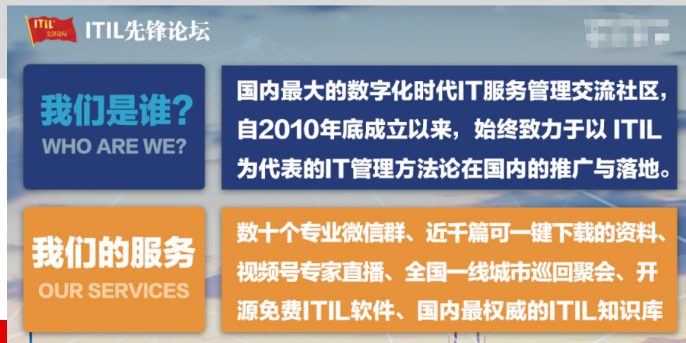


企业数字化转型中devops对于运维的改变与实践

解决方案架构师



ITIL 先锋论坛

我们是谁? WHO ARE WE?	国内最大的数字化时代IT服务管理交流社区，自2010年底成立以来，始终致力于以 ITIL 为代表的IT管理方法论在国内的推广与落地。
我们的服务 OUR SERVICES	数十个专业微信群、近千篇可一键下载的资料、视频号专家直播、全国一线城市巡回聚会、开源免费ITIL软件、国内最权威的ITIL知识库

CONTENTS 目录

- ① 行业现状
- ② 建设方案
- ③ 案例分享

01

行业现状

DevOps典型需求

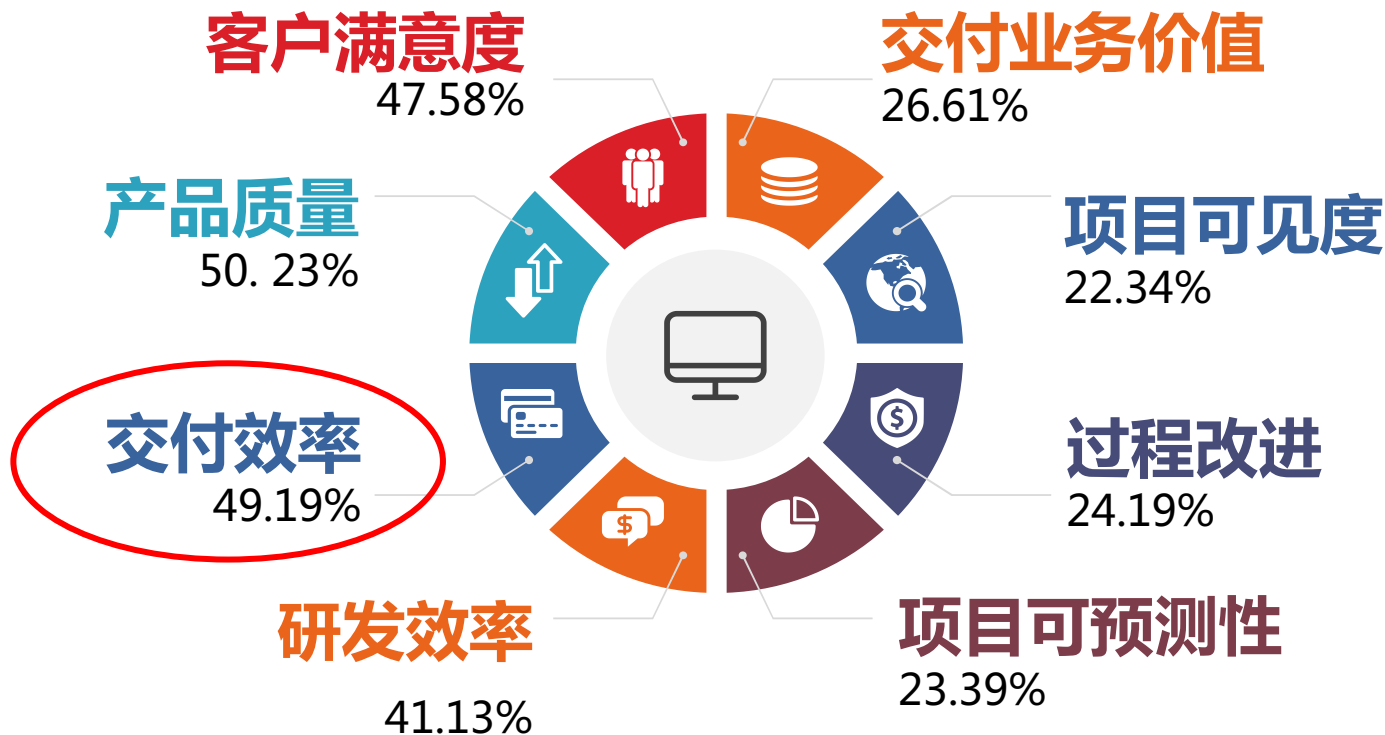
通过敏捷快速迭代，实现：快速交付成果，及时验证价值和响应市场变化

通过端到端的高度自动化，减少非必要支撑工作，提升研发和交付效率，降低成本

通过一系列标准和规范的推行和平台化落地，提升开发质量和效率

通过度量分析实现研发进度、质量、风险、投入的实时可见，可及时调整优化

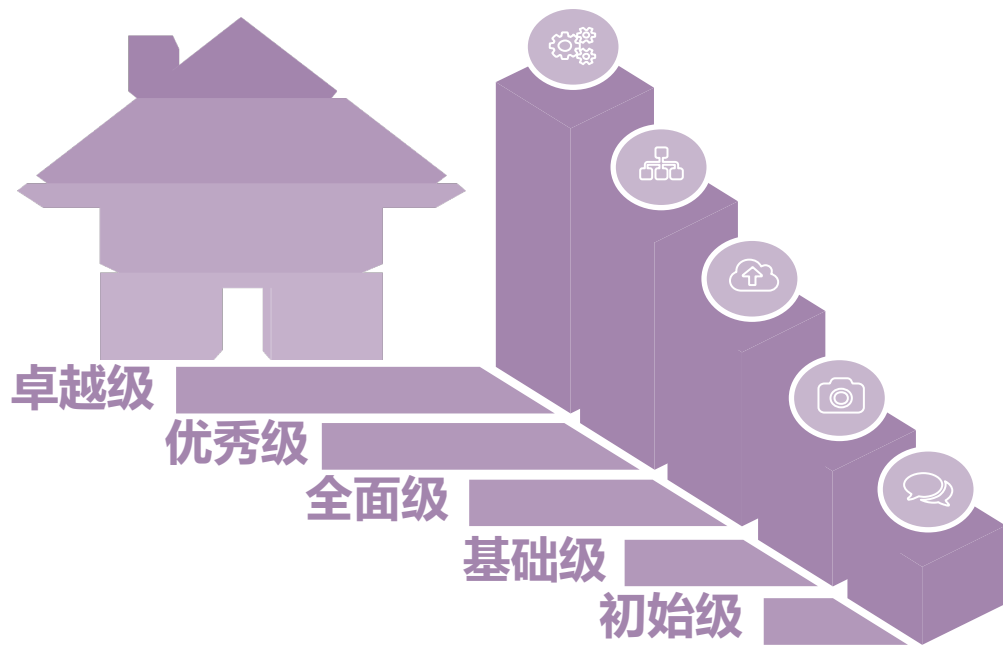
DevOps期望中对于Ops的关注



源引：《中国 DevOps 现状调查报告（2019 年）》
——云计算开源产业联盟

建设全面卓越级的DevOps体系

- ✓ **卓越级**
在组织内全面形成持续改进的文化并不断驱动 DevOps 在更大范围内取得成功。
- ✓ **优秀级**
在组织内全面落地 DevOps 并可按需交付用户价值达到整体效率最优化。
- ✓ **全面级**
在组织内全面推行 DevOps 实践并贯穿软件全生命周期获得整体效率提升。
- ✓ **基础级**
在组织较大范围内推行 DevOps 实践并获得局部效率提升。
- ✓ **初始级**
在组织局部范围内开始尝试 DevOps 活动并获得初期效果。

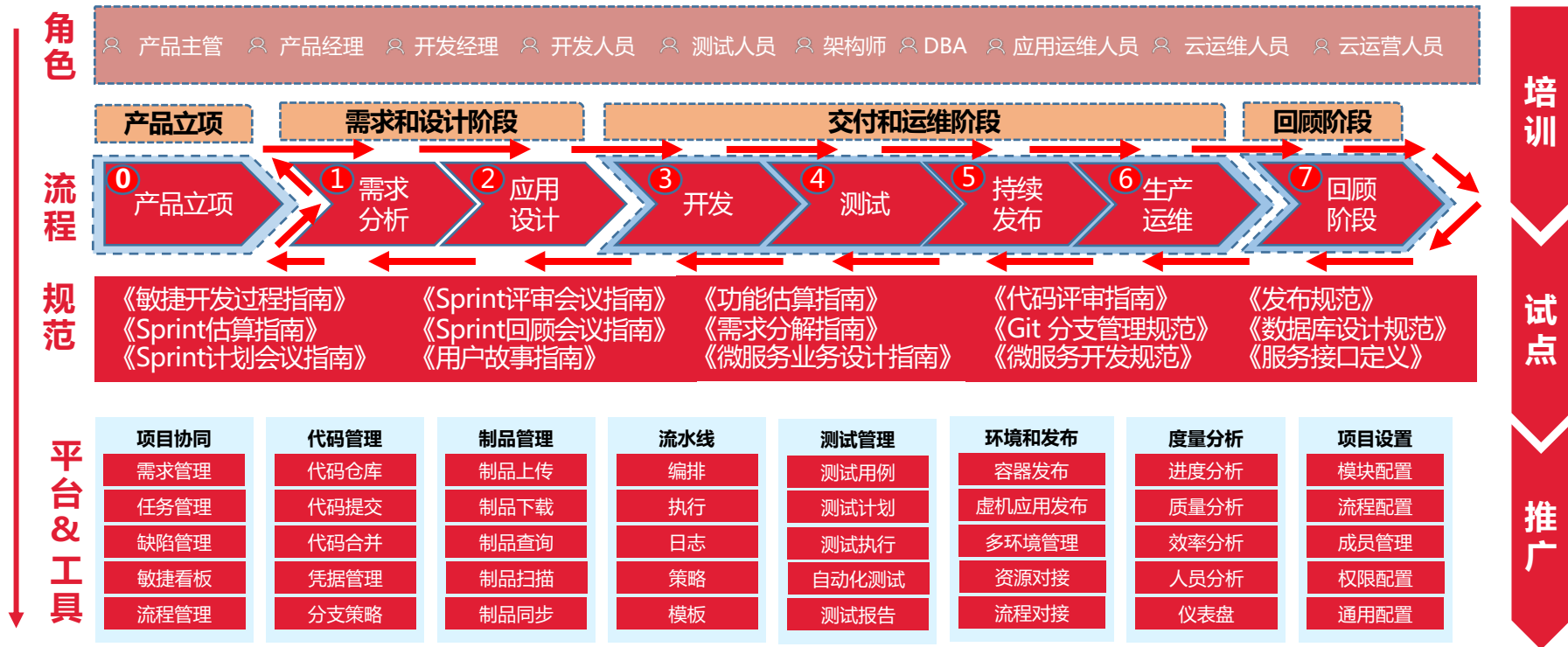


数据来源：中国信息通信研究院

02

建设方案

DevOps整体建设思路：咨询+产品+实施



能力类	一、研发运营一体化（DevOps）过程											
能力域	1.敏捷开发管理			1. 自动化流水线			持续交付					
能力子域	敏捷	敏捷	敏捷	持续集成	测试管理	部署与发布管理	环境管理	数据管理	1.度量...	监控管理	事件管理	
能力项	需求工作	价值流	敏捷角色	版本控制	构建实践	测试 分层策略	部署与 发布模式	环境管理	测试 数据管理	度量指标	指标采集	1.容器
	需求活动	仪式活动	团队结构	变更管理	持续集成	代码 质量管理	持续部署 流水线		数据 变更管理	度量 驱动改进	监控 数据处理	
						自动化测试					异常识别	

- ① 团队技术能力强、配合程度高
- ② 系统的业务连续性要求相对较低
- ③ 希望快速发布

参考实施路径3：博云



OPS有哪些痛点

- 各个系统都各自手动维护自己的系统/应用元数据(比如资源、流水线、制品、监控等), 元数据不统一/不准确, 无法进行更精细化的运营管控, 难以有效降低成本。~如应用与资源的关系, 应用与数据库/中间件的关系等
- 针对不同的框架结构建设了多个应用管理系统, 各系统和服务相互独立, 缺少统一管理模型和全局视角。降低了运维效率, 增加了大量的额外成本。
- 应用打包、部署、监控、故障处理、资源管理、流程审批各自独立, 难以标准化, 导致操作分散, 带来了大量的协调成本
- 部分应用部署、变更自动化程度低, 大量手动命令行操作, 带来了稳定性隐患
- 可观测性目前散落在各个系统, 如何将监控、事件、trace结合, 提供应用/可执行模块的可观测性, 提升业务稳定性

影响与解决方案

影响

- 低可维护性：部署发布上线效率低，易出错；监控不完整，未联动
- 稳定性风险：部署变更操作不标准，增加了稳定性风险（50%以上的事故来自变更）
- 额外的成本：应用视角的资源管理混乱，带来额外的资源成本



解决方案

建设“统一应用管理平台”

以标准化/自动化为基础，从应用视角，整合各类应用管理能力。实现减少管理成本、提升稳定性、降低资源成本的目标

IT运维转型的趋势：从工具化/流程化 到 服务化/集成化



统一应用管理平台的定位与落地措施

统一应用管理平台

目标

统一系统管理入口

整合各类管理能力，减少管理成本，提升效率

统一系统包和配置规范、高度自动化

实现部署发布上线的高度自动化，提升稳定性

运营降成本

基于准确数据，精细化运营，降成本

用户群体与收益

应用运维人员

- ① 高度自动化的发布部署上线
- ② 分级发布，稳定性提升
- ③ 一站式应用全生命周期管理，可用性提升
- ④ 架构可视化，监控变更联动可视，降低变更风险，支撑快速问题定位

开发测试人员

- ① 一键获取开发测试环境
- ② 流程联动，快速多环境流转

管理者

- ① 实现CMDB元数据准确性，更好的支撑CMDB周边系统建设
- ② 通过基于准确数据的运营分析，降低系统成本

落地保障

组织

从上到下推广支持，守住入口卡点

机制

标准化、规范化、流程化

工具

平台化，统一操作入口

统一应用管理平台功能架构

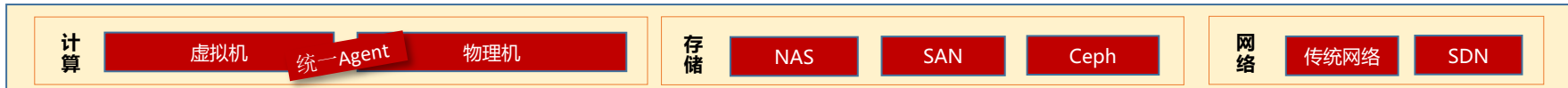
统一
应用
管理
平台



周边
平台

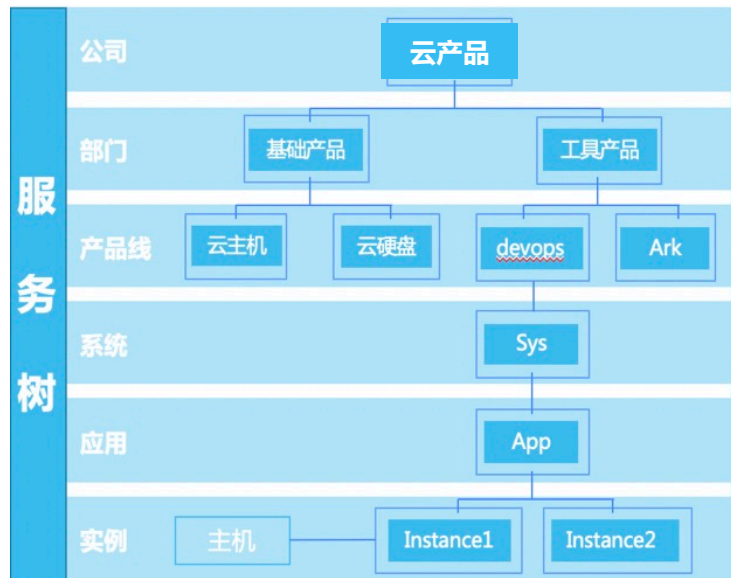
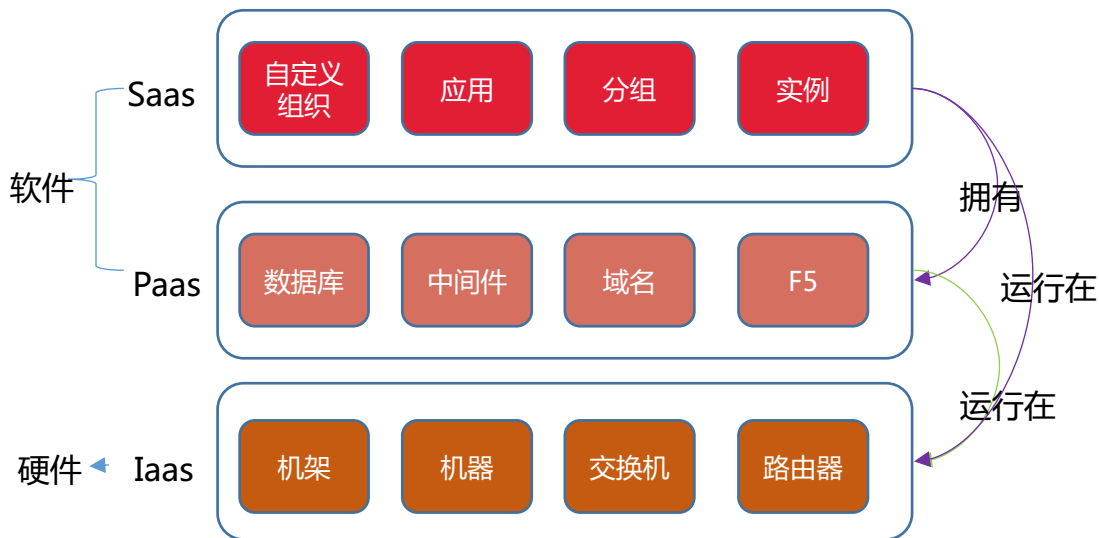


基础
设施

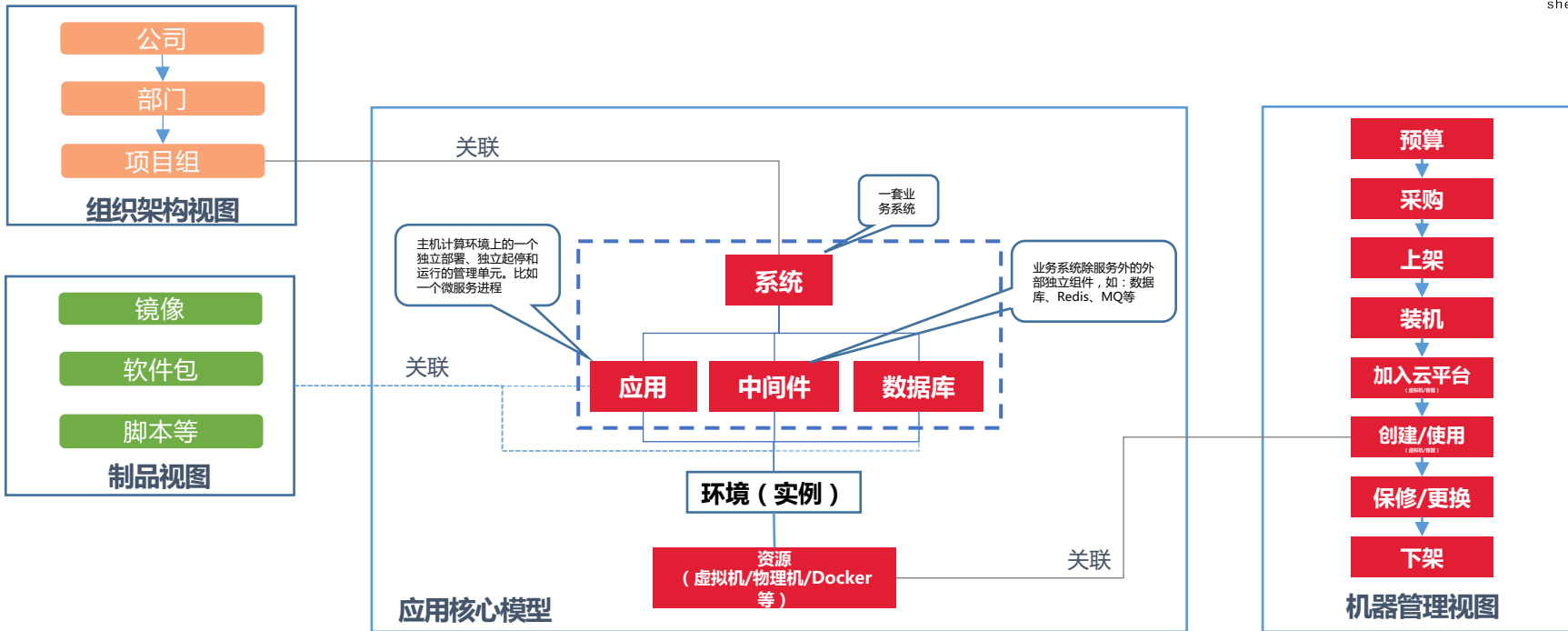


服务树- 书同文 车同轨

- 以应用为视角，推进IT资源的元信息建设，Saas->Paas->Iaas
- 硬件资源--Iaas资源的元数据（机器、容器、机架、路由器等硬件资源）
- 软件资源--Paas资源的元数据（依赖的中间件、域名、路由设备上面的ACL规则等）
- 软件资源--Saas(应用)的元数据（用户自身的程序配置，比如服务名称、架构描述、部署路径端口等）



服务树-联动以应用为中心的CMDB管理核心模型



- ◆ 应用CMDB建设的核心场景是支撑应用从制品、测试、发布、运行管理、升级/回滚的全生命周期管理
- ◆ 应用CMDB要有统一的基础资源CMDB作为基础，但是应用管理和资源管理两个视图应该是分开的

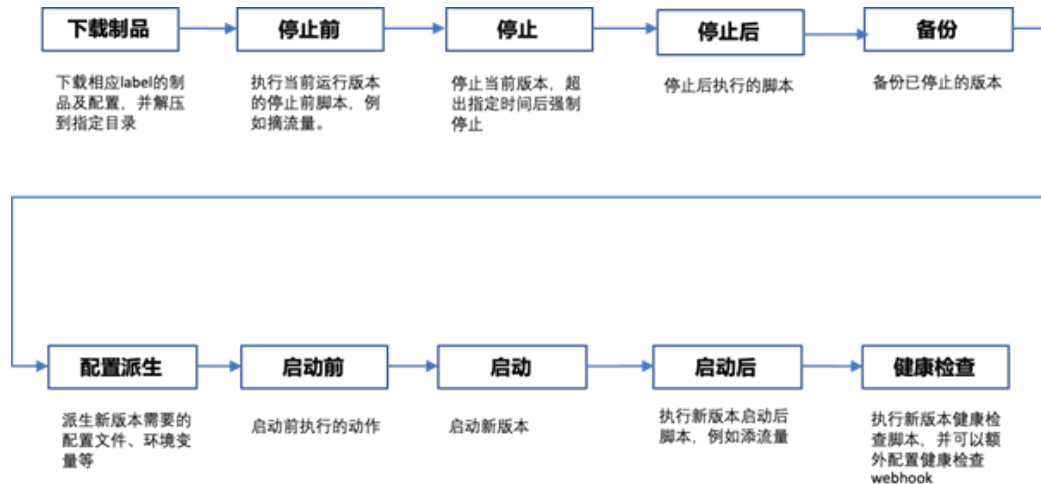
规范化自动发布

◆ 统一制品、打包、发布规范

◆ 统一流程模板和目录规范

◆ 统一容器、VM的部署体验

流程模板示例



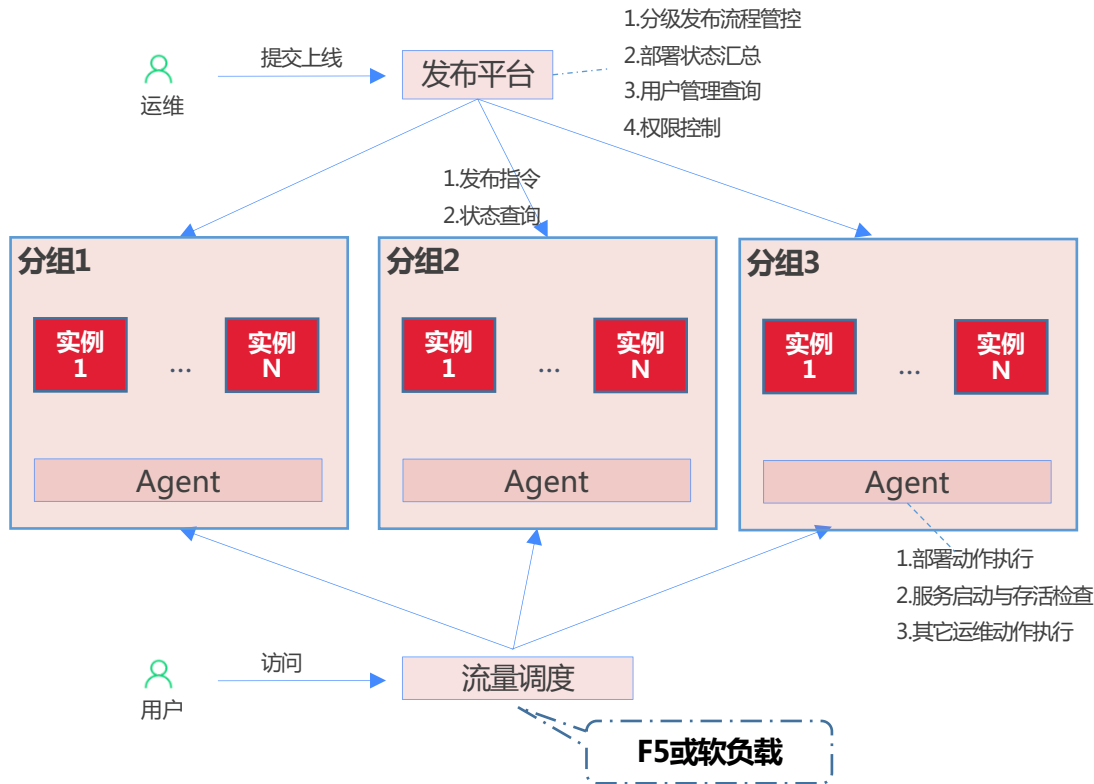
自动发布-分级发布策略

部署发布

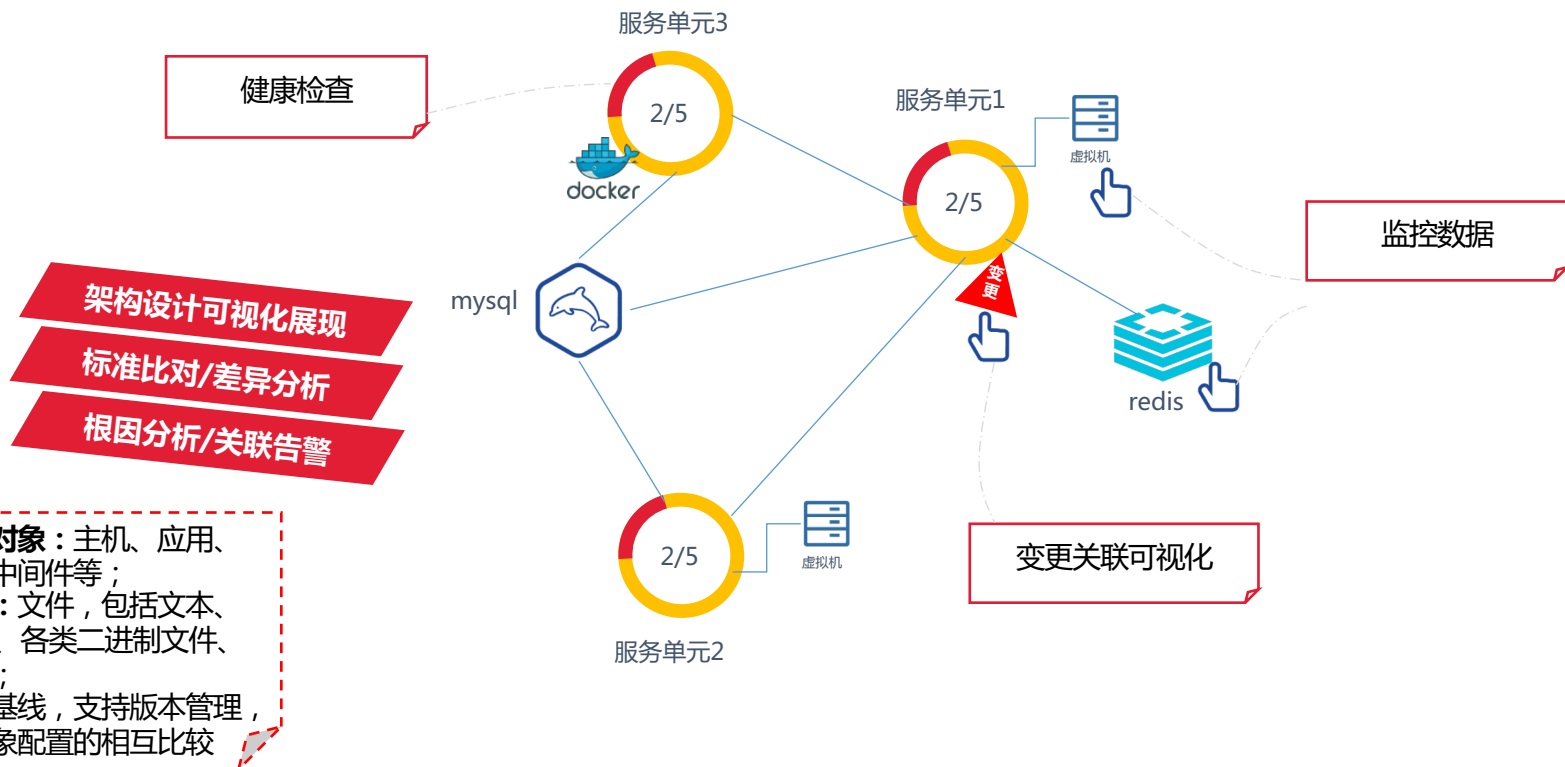
- 支持服务按分组进行蓝绿发布
- 支持小流量灰度发布
- 支持预发与线上环境分离
- 支持串行、并行等多种上线策略

流量接入

- 调度灵活，支持多种负载均衡策略
- 流量秒级自动切换
- 支持单独管理实例流量
- 扩容自动接入流量，支持服务自动扩展



架构守护



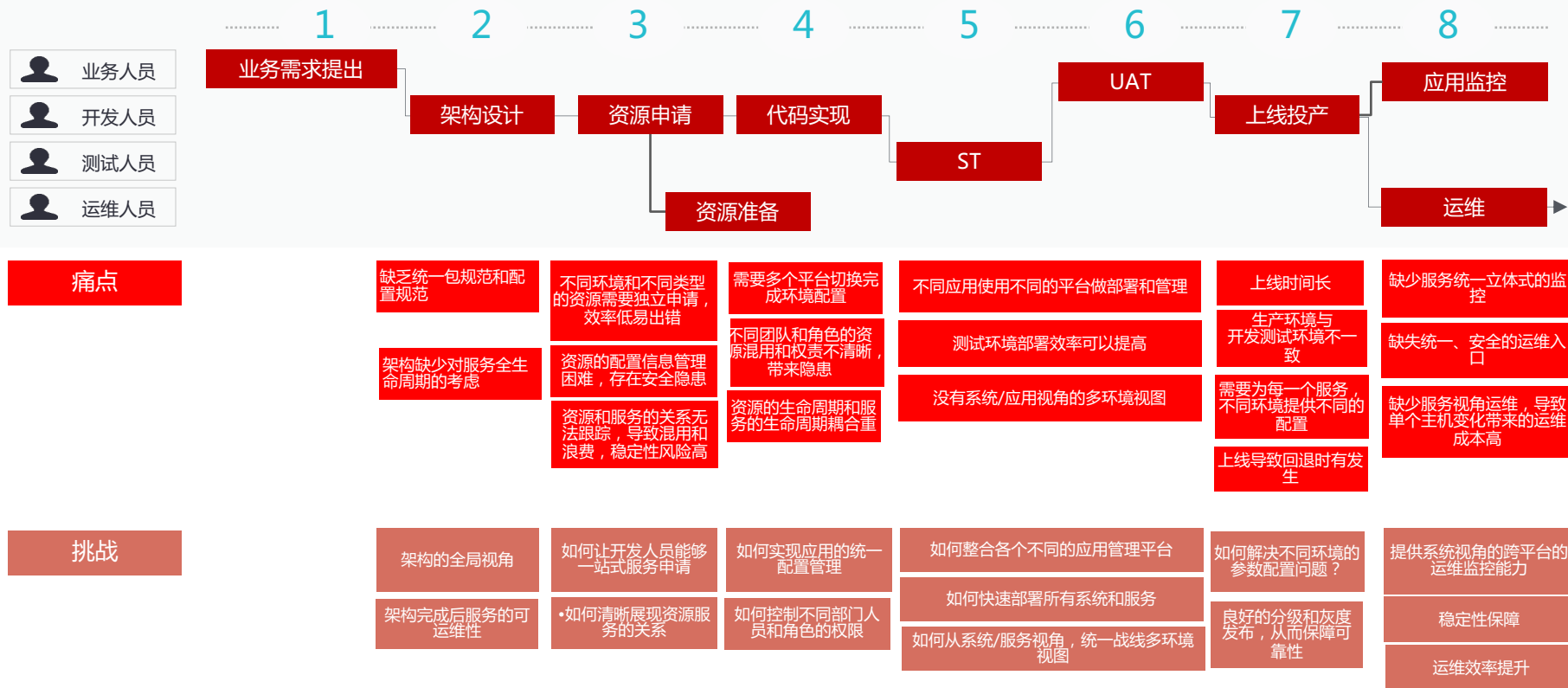
平台建设的价值



03

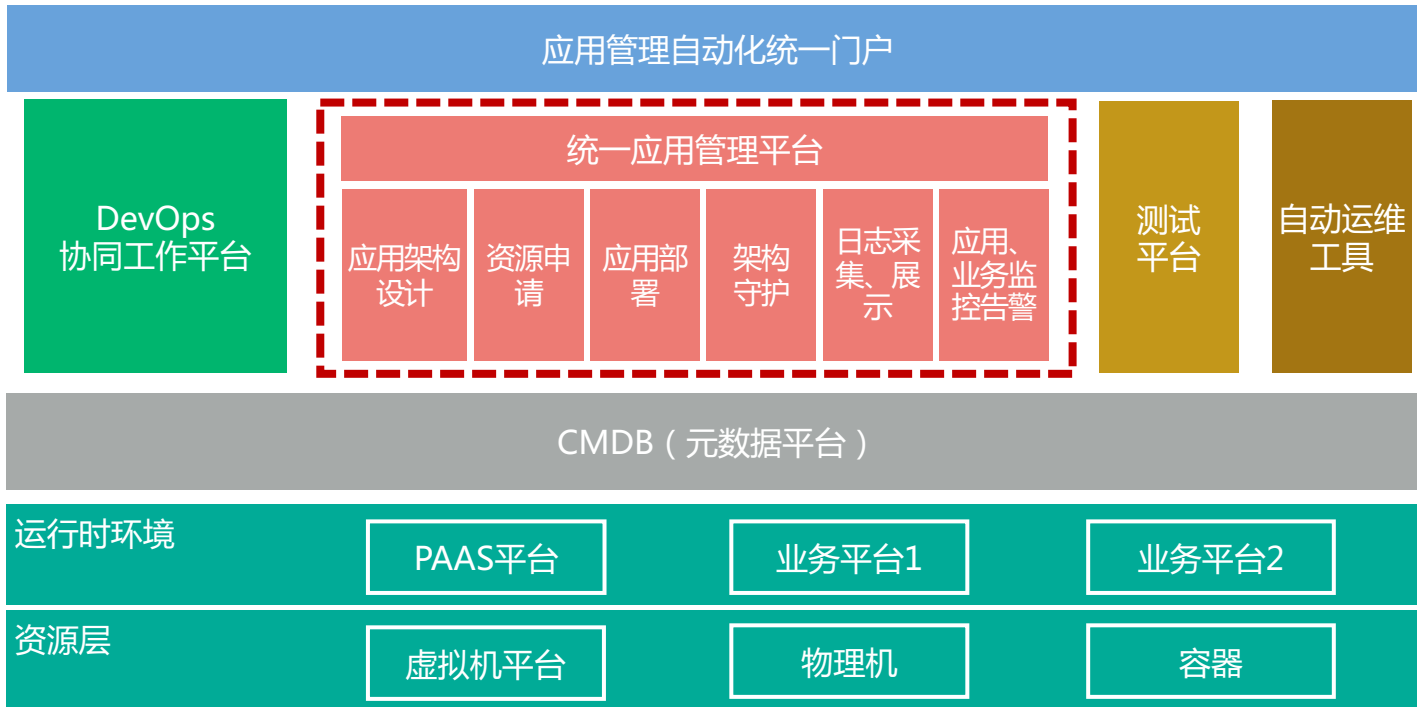
案例分享

某大型股份制银行应用管理现状



平台主要建设内容与定位

平台定位于应用全局和全生命周期管理平台



收益1：环境一致性管理

1、硬件环境，包括服务器环境、与网络环境

- 虚拟机配置
- 操作系统配置
- 网络配置
- 负载配置

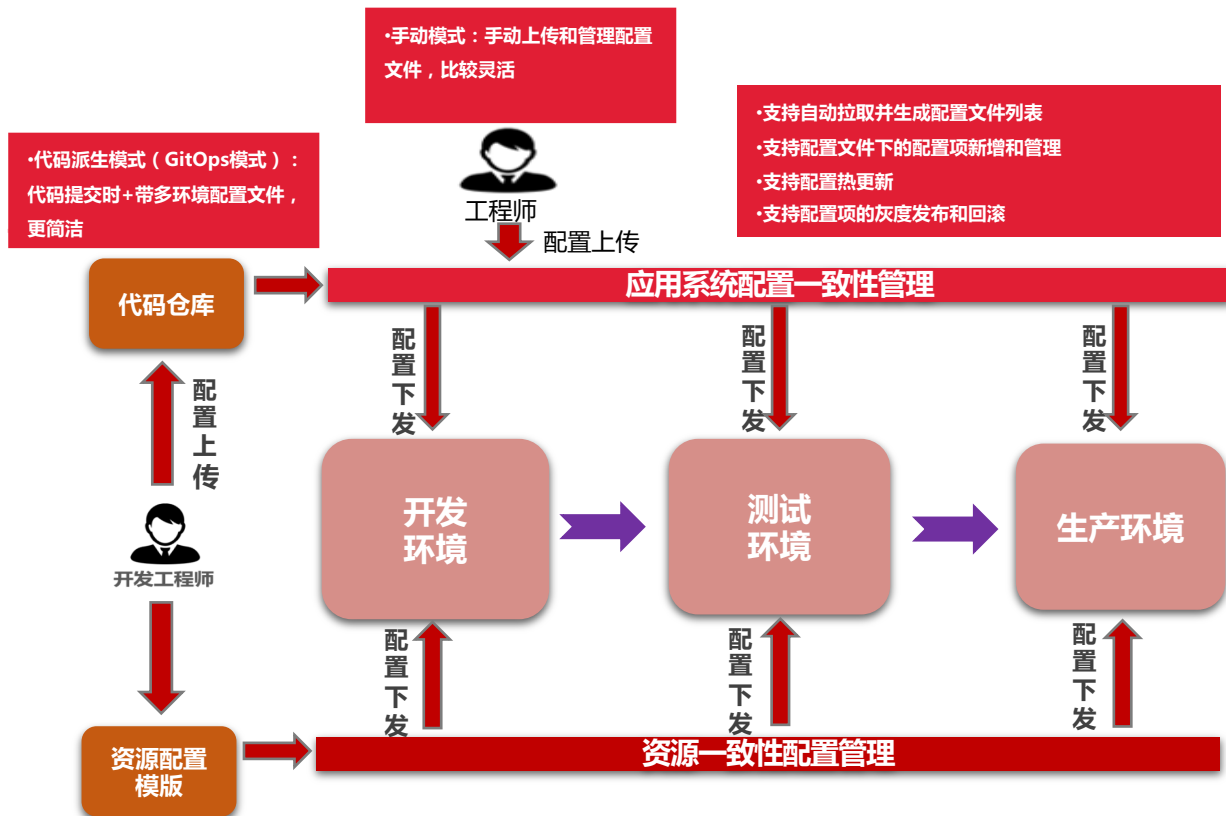
2、软件环境

- 版本一致性

包括包括操作系统、数据库、中间件的版本，被测系统的版本。

- 配置一致性

系统（操作系统/数据库/中间件/被测试系统）参数的配置一致，



收益2：非容器和容器统一应用管理模型

基于BUILD/SHIP/RUN标准操作，统一容器和非容器管理模型

Build :

- ◆ 对容器：通过Dockerfile生成Image；
- ◆ 对软件包：生成统一软件包格式
***软件包+脚本=Image**

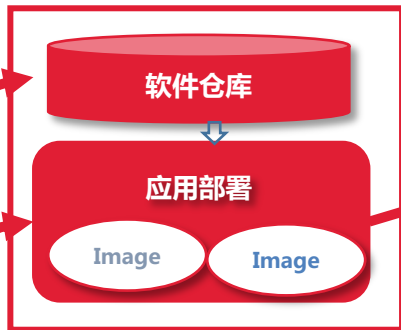
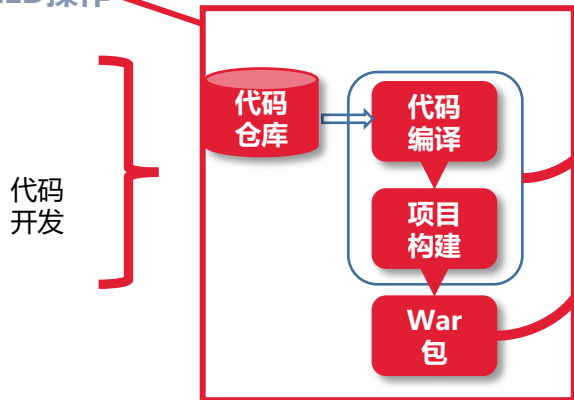
Ship :

- ◆ 对容器：Image在仓库和服务器的传递
- ◆ 对软件包：软件包在仓库和服务器的传递

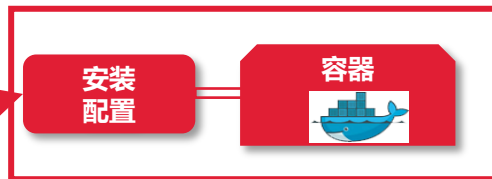
Run :

- ◆ 对容器：应用部署、启动和停止，基于Docker命令实现
- ◆ 对软件包：应用部署、启动和停止由自动脚本控制

BUILD操作



SHIP操作



RUN操作

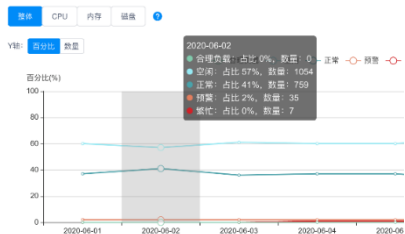
收益3：降成本、提升效率和稳定性

通过应用发布入口的统一，实现准确的资源使用统计，从而识别闲置资源，降低资源成本



降成本

通过精细化资源运营，对资源使用情况进行度量，为资源增加和回收提供决策依据，降低资源成本



提效率

