DevOpsDays Beijing

DevOpsDays

2017 DevOpsDays Beijing

时间：2017.03.18 地点：北京朝阳区万达索菲特大酒店

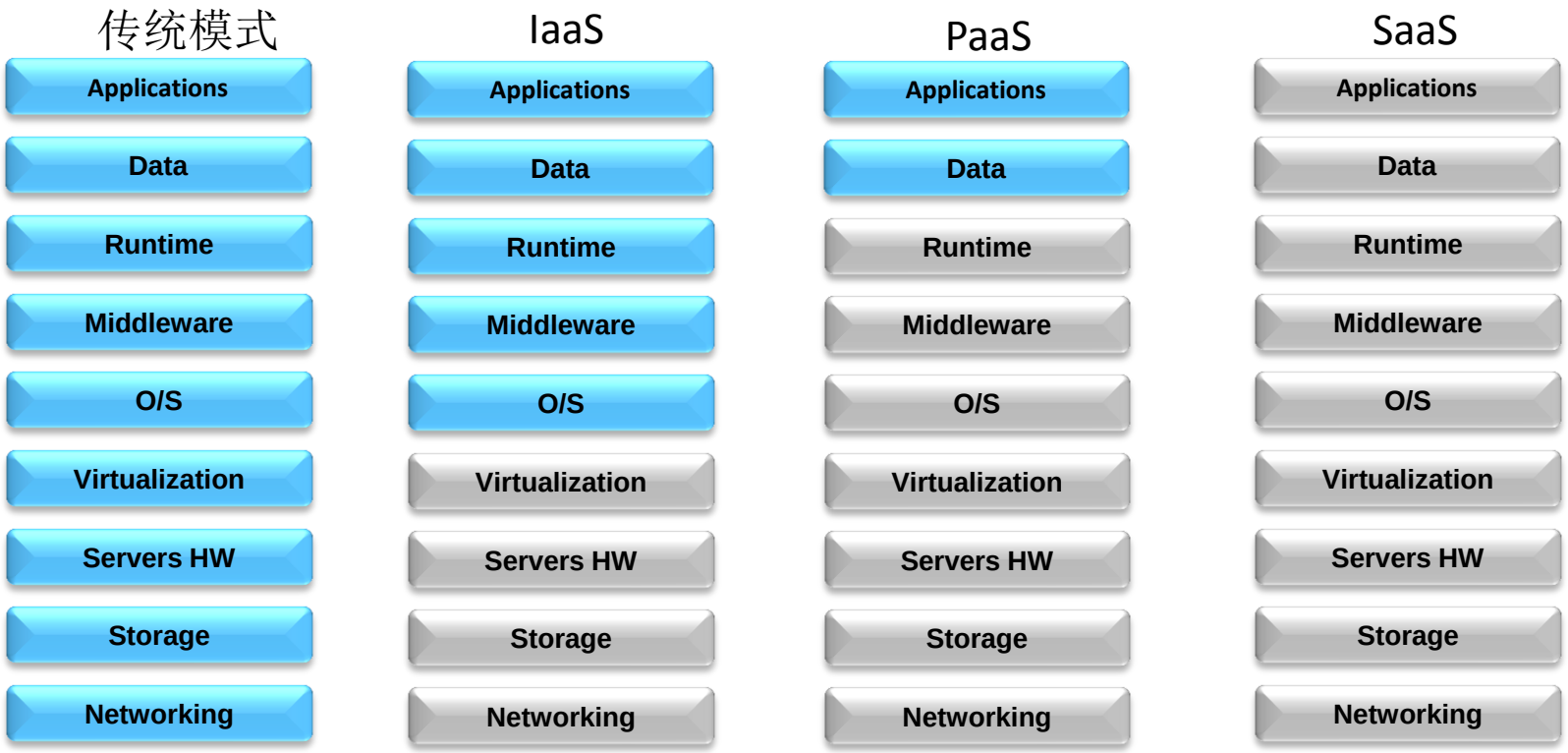
主办单位：高效运维社区 GreatOPS Community　Best Practice 最佳实践

DevOps，驱动应用从运维走向管理

王津银　优维科技CEO

目录

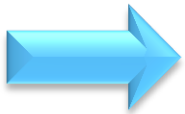
从云的层次看应用管理的所处位置



- 应用管理的边界在哪儿？

从应用的生命周期看应用管理

运维的视角从基础设施到应用



应用管理存在的困难

01

应用的关联要素多种多样，数据、资源、变更，如何规范化管理等等

02

应用的类型多且复杂，如OA、ERP、2C应用、APP等等

03

应用的交付频繁，如何安全、高效、可持续进行

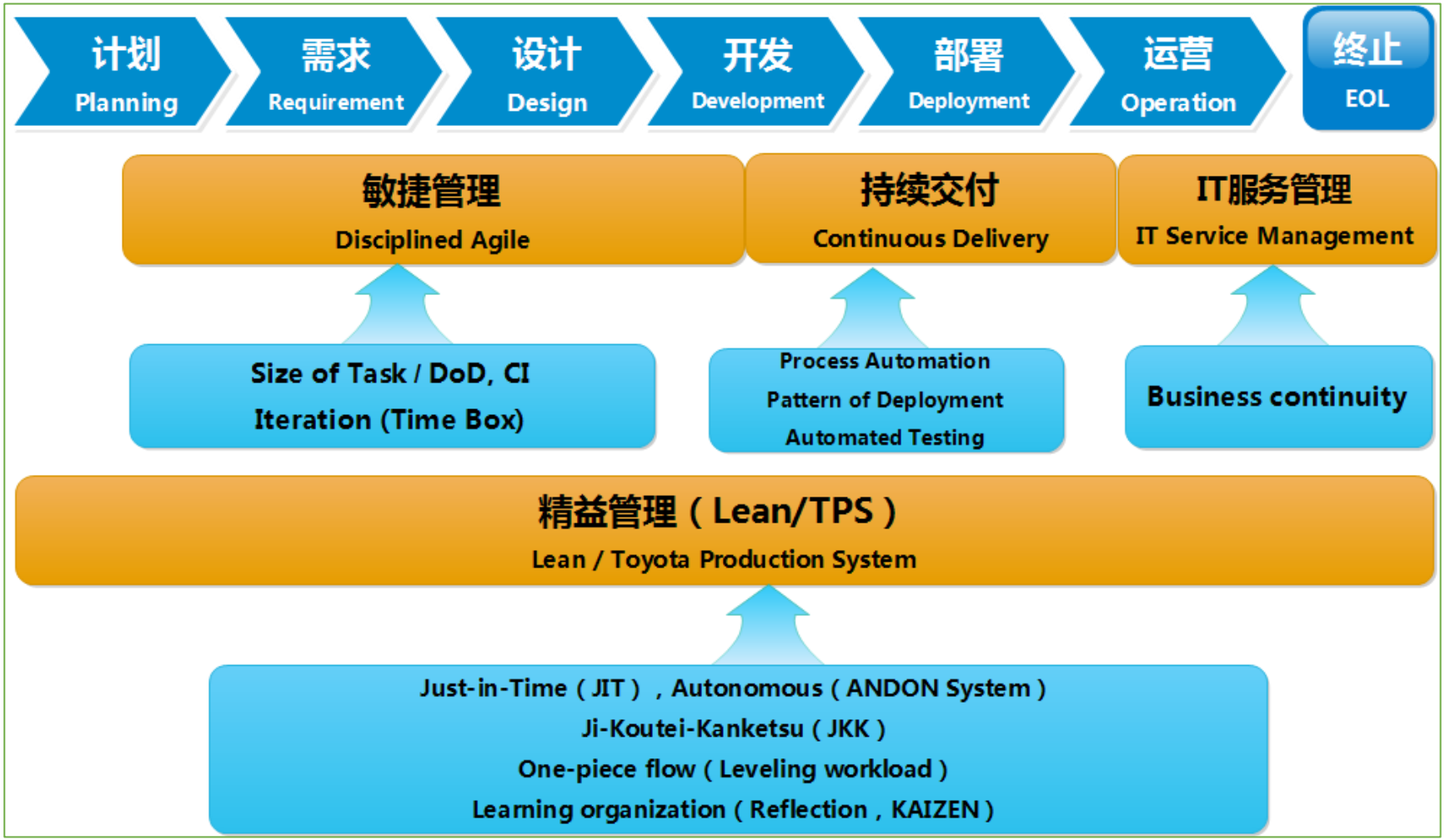
04

应用的开发语言很多，管理是否可以做到对语言抽象？

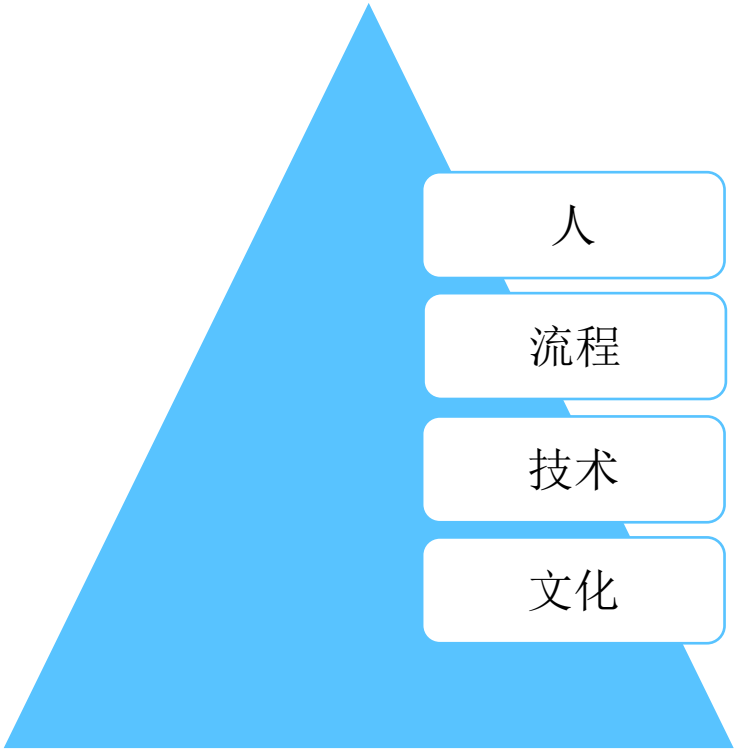
05

业务—应用—资源，应用是核心衔接点，它的作用到底是什么？

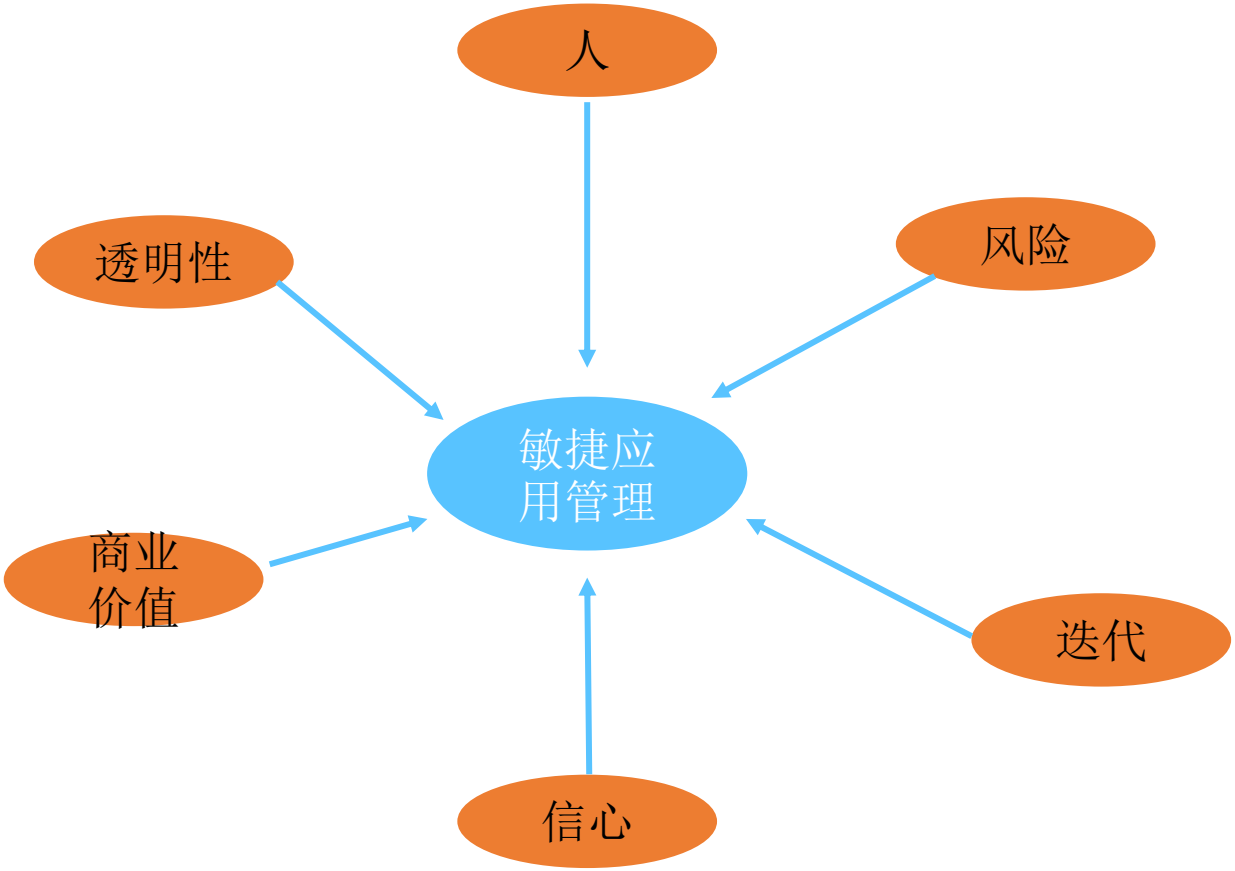
DevOps驱动应用全生命周期管理



从传统IT视角看应用管理要素



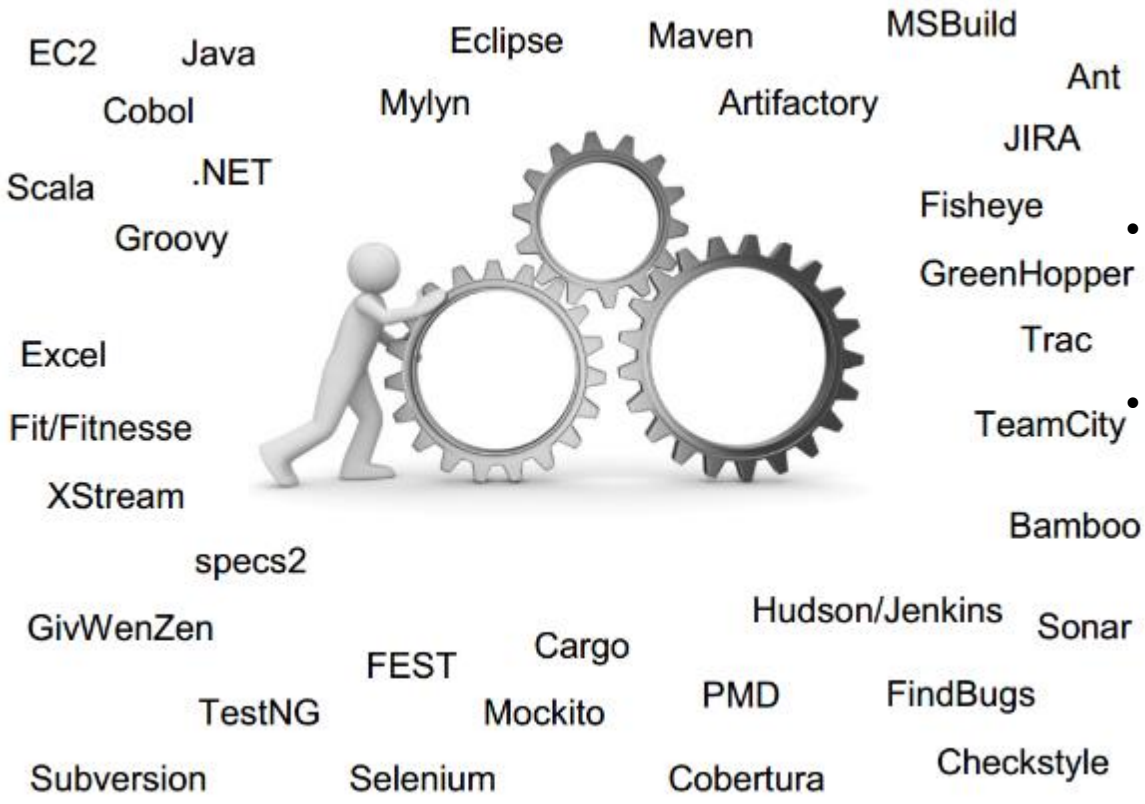
从敏捷的角度看应用管理要素



基于敏捷下的应用管理框架

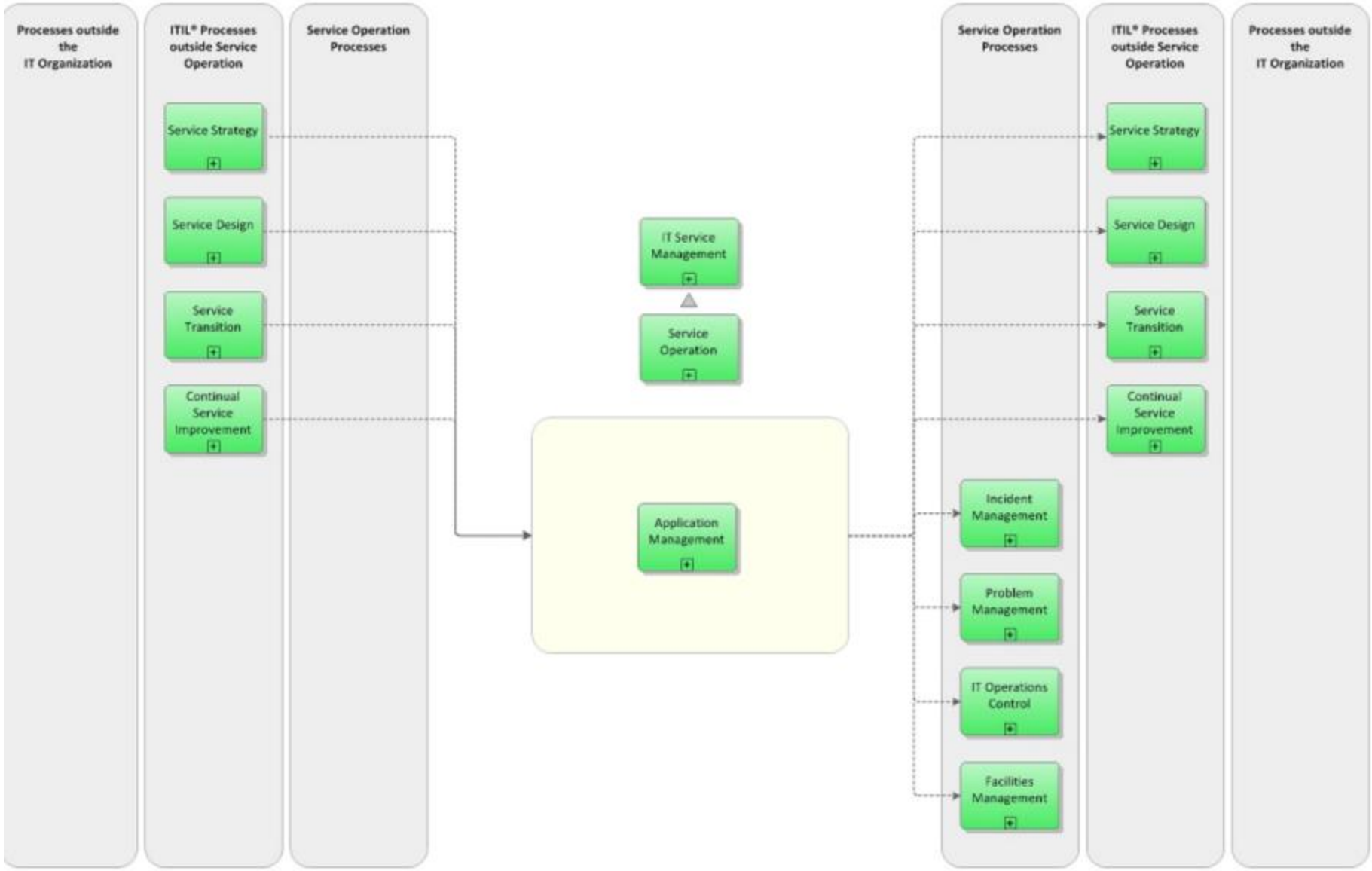
- 责任文化
- 部门之间的分离而非融合
- 流程机制给效率带来阻碍
- 远离业务，而非亲近业务

从持续交付角度看应用管理要素

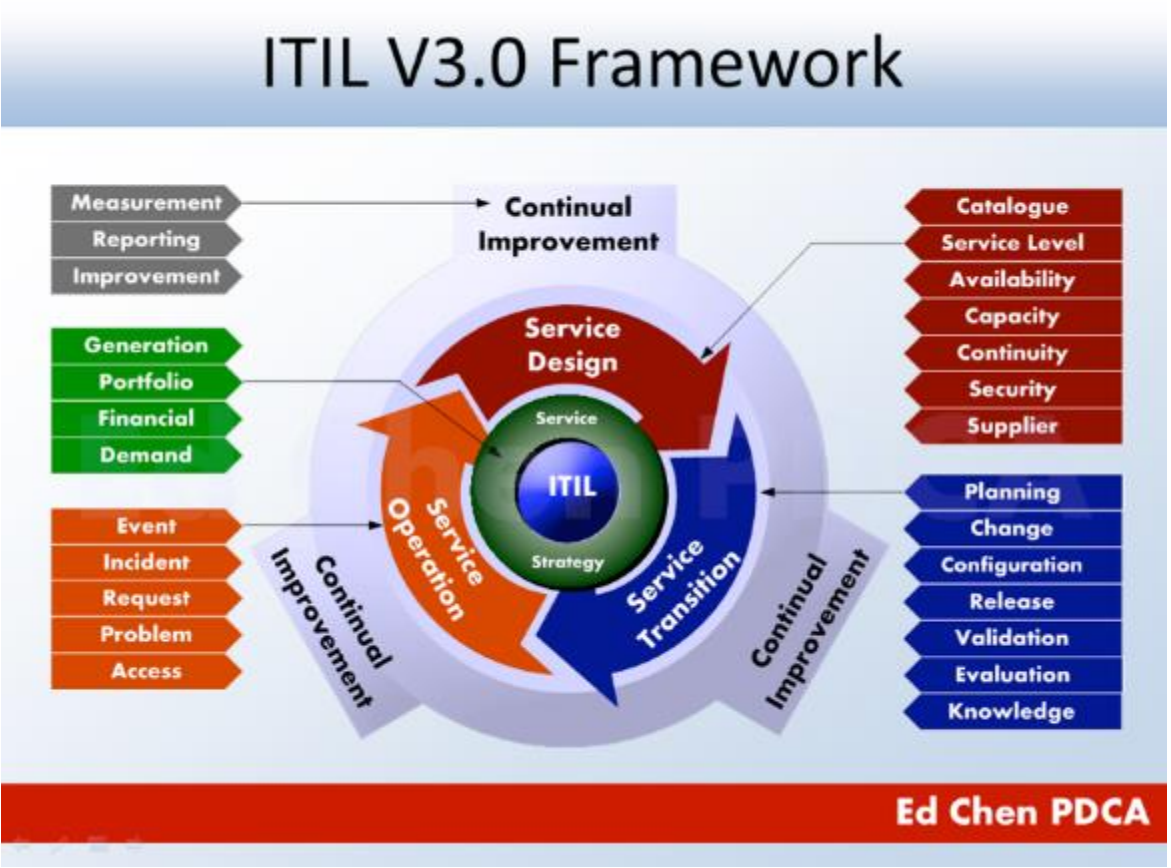


- 基于交付链打造深度的平台、工具能力集成
- 由人推动

基于ITIL的 应用管理流程框架

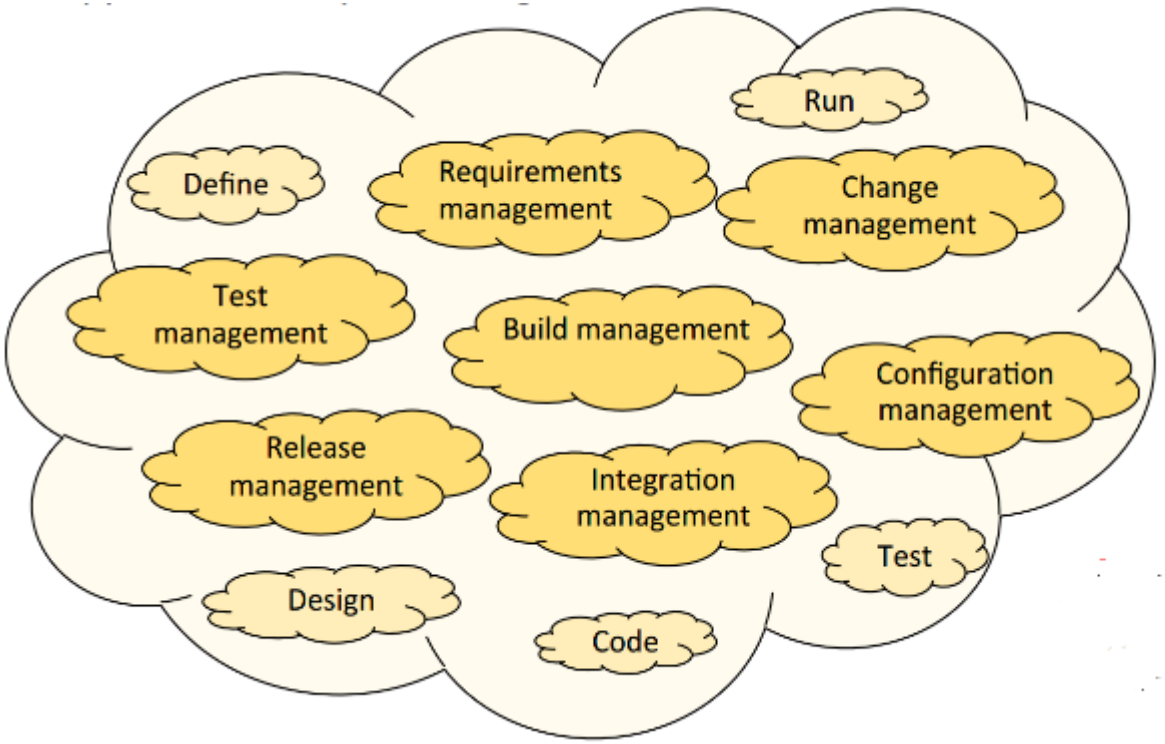


ITIL遗留的IT管理问题



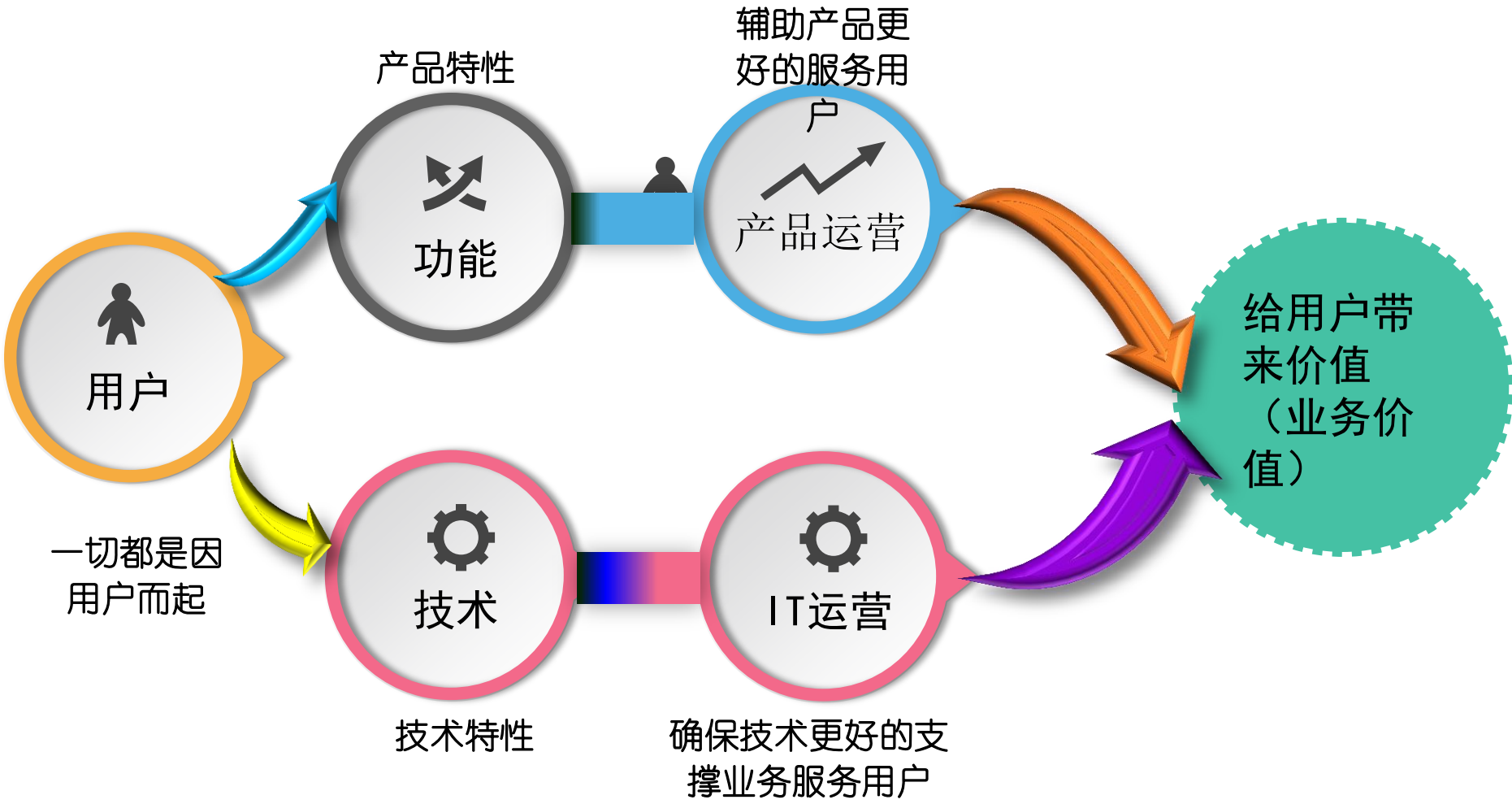
- 责任文化
- 部门之间的分离而非融合
- 流程机制给效率带来阻碍
- 远离业务，而非亲近业务

应用生命周期管理的能力详细



- 覆盖代码、设计、测试、运行等多个阶段的管理能力
- 从CI、测试、到部署变更、到运营监控管理端到端覆盖

IT运营是IT能力的产品化体现

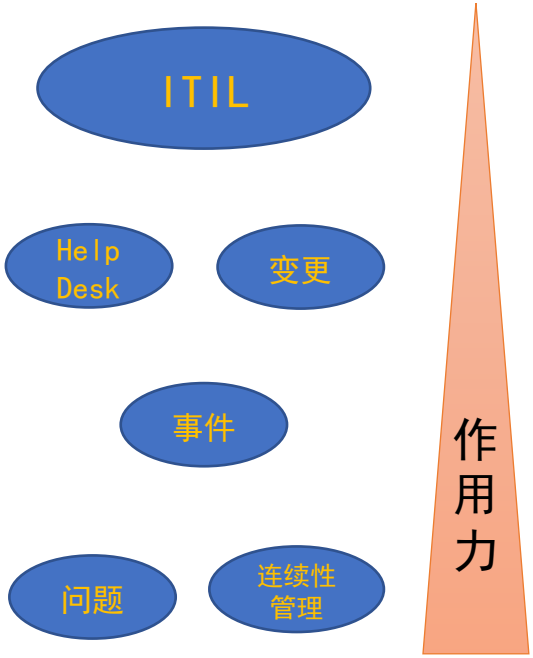


ITIL与DevOps在应用维度上的融合

数据化信息需求、自动化任务需求

面向管理过程（ITIL流程）

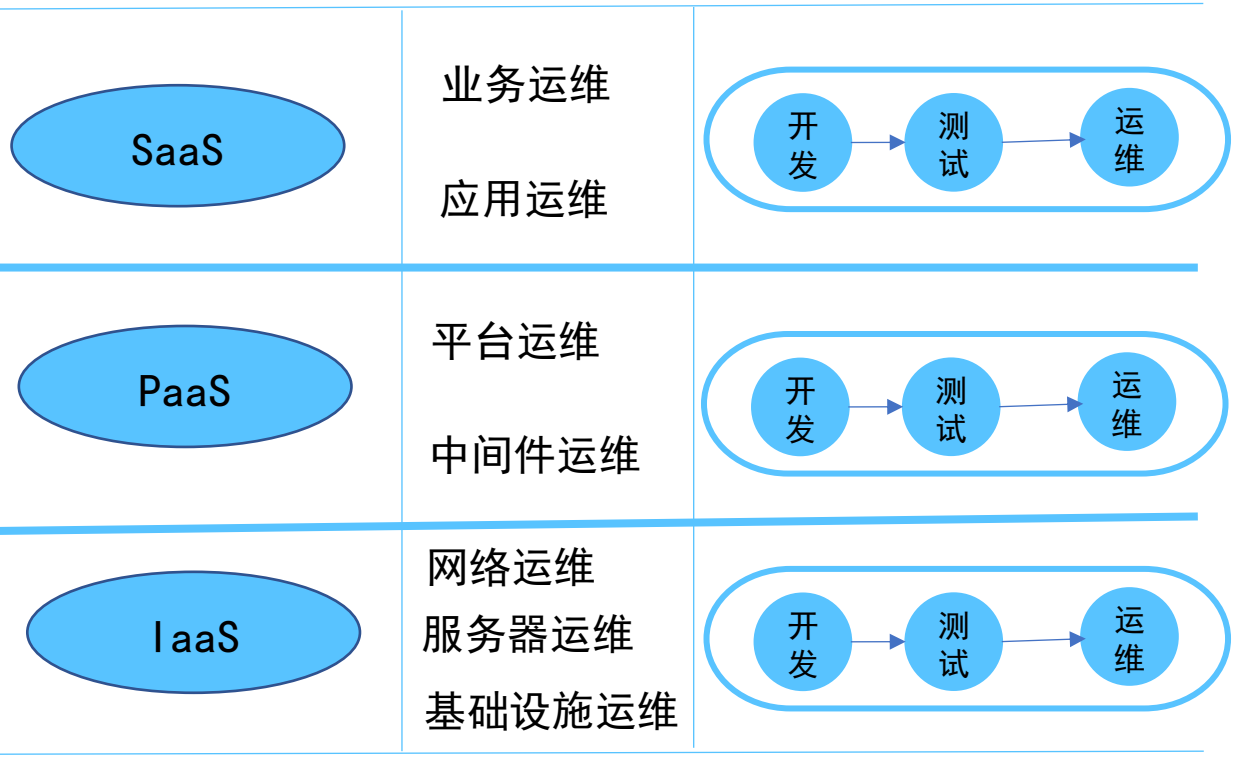
以“离线任务”管理为主要特征



规范优先、效率低、成本高

面向IT运营过程（执行）--DevOps

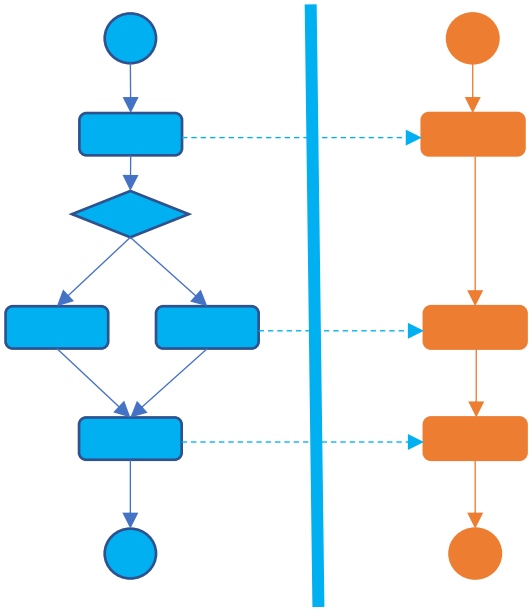
以“在线服务”管理为主要特征



质量、效率、成本、规范之间平衡

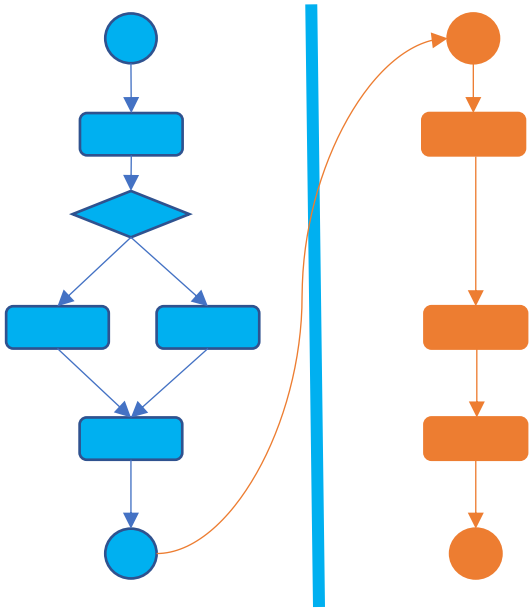
ITIL与DevOps的逻辑关系

模式1：



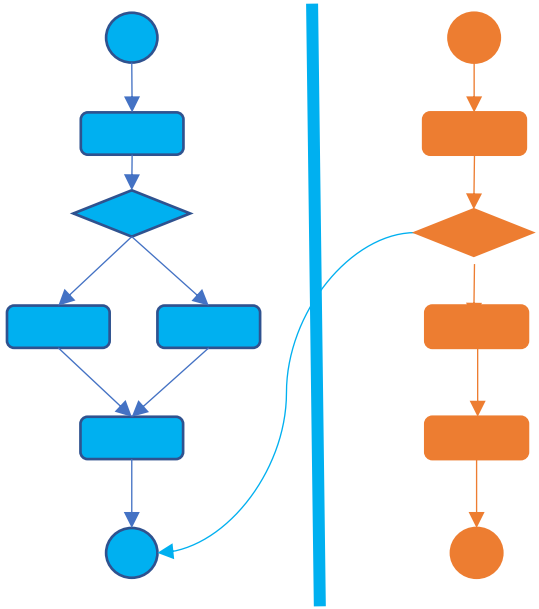
- 在线服务开通流程
- 资源管理流程

模式2：



- 重大变更流程
- 高稳定性服务保障流程、变更流程

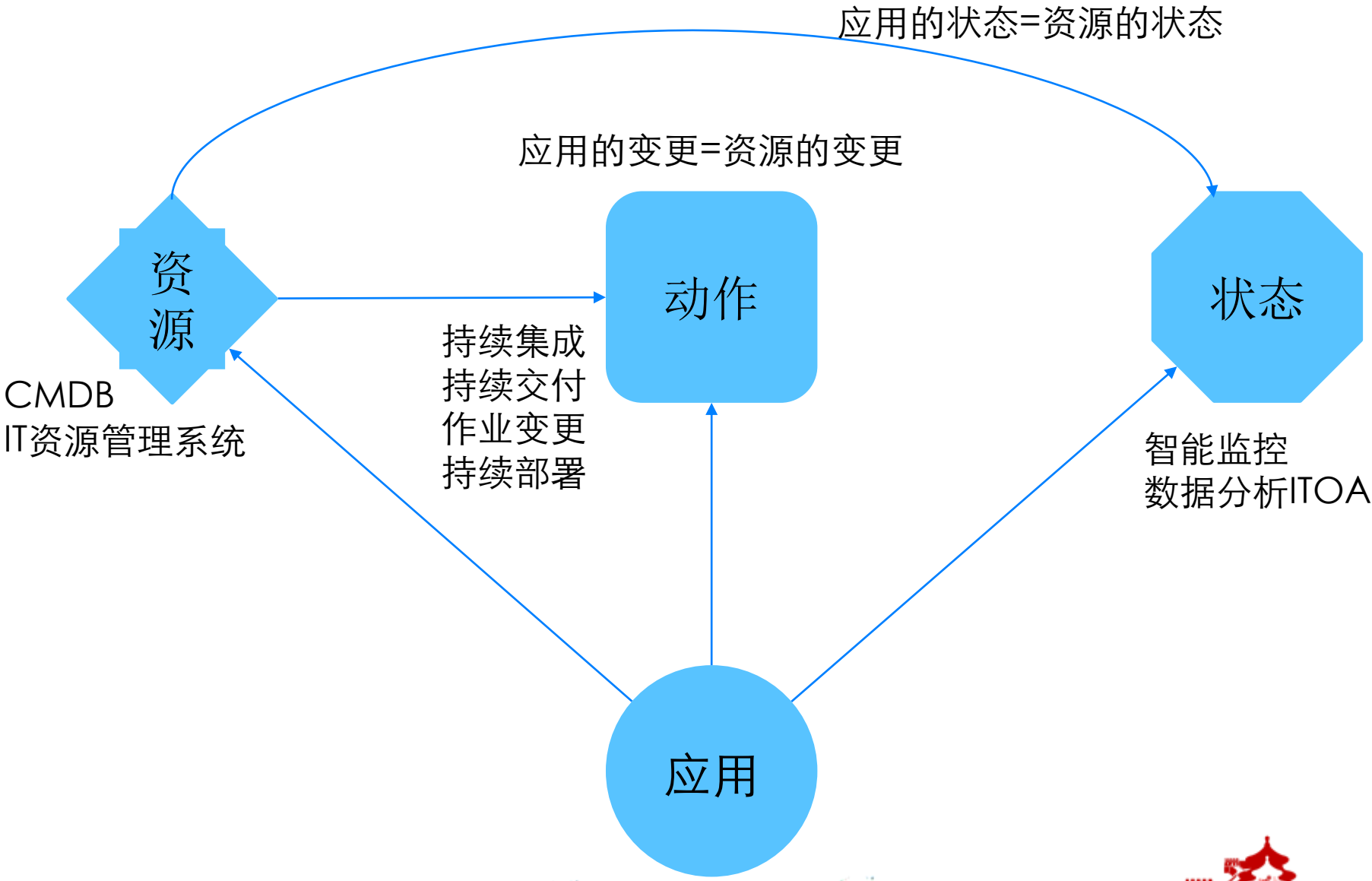
模式3：



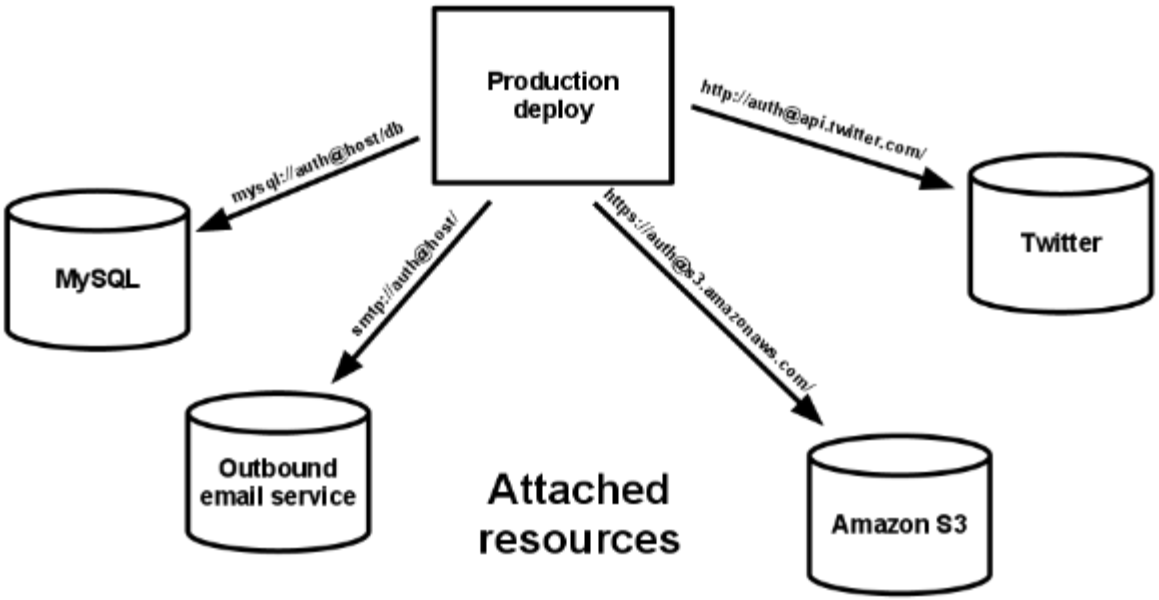
- 敏捷发布流程
- DevOps变更流程

目录

面向应用的能力管理框架抽象

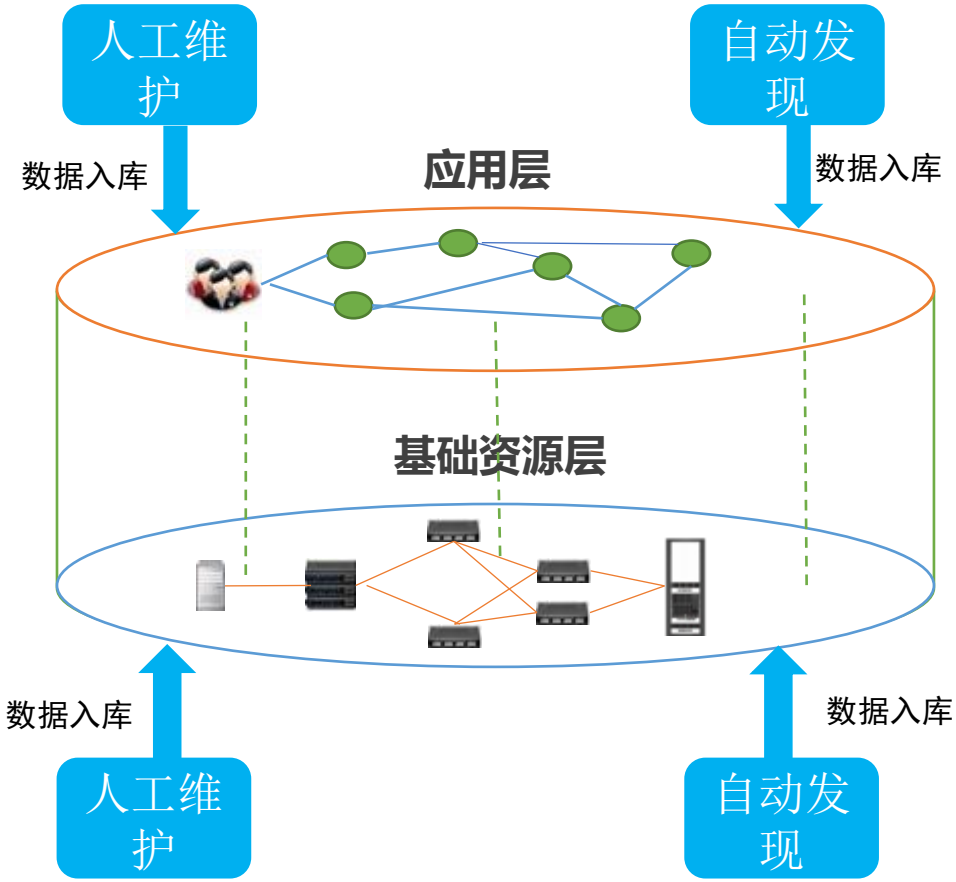


何为应用的资源



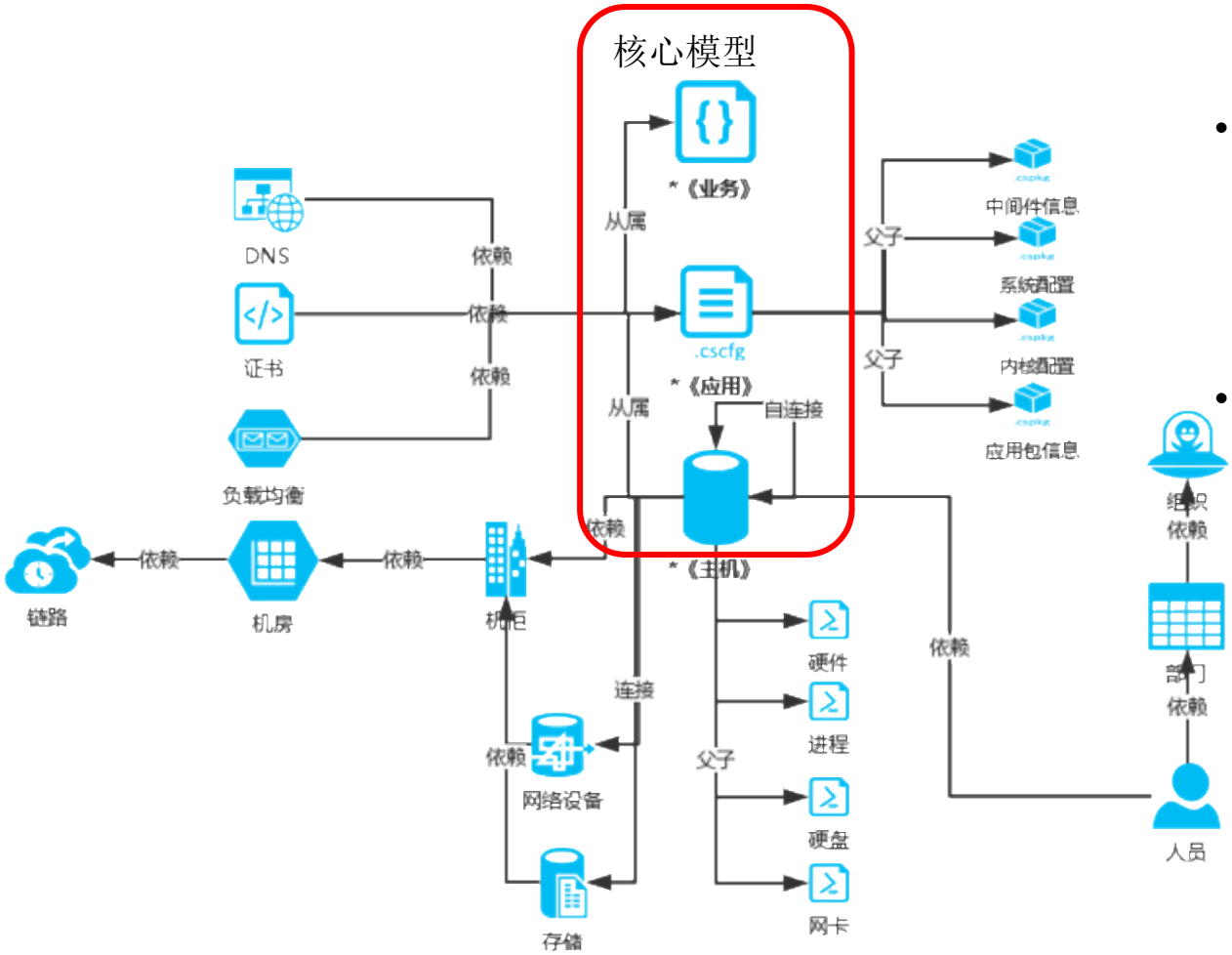
- 应用的资源包括本地的资源和第三方资源。
- 本地资源是应用Host所占用的资源，如主机、端口、程序包等等
- 远端资源是关联的第三方服务，比如说cache、分布式rds等等
- 每个不同的后端服务都是一个资源

应用资源从基础设施分离



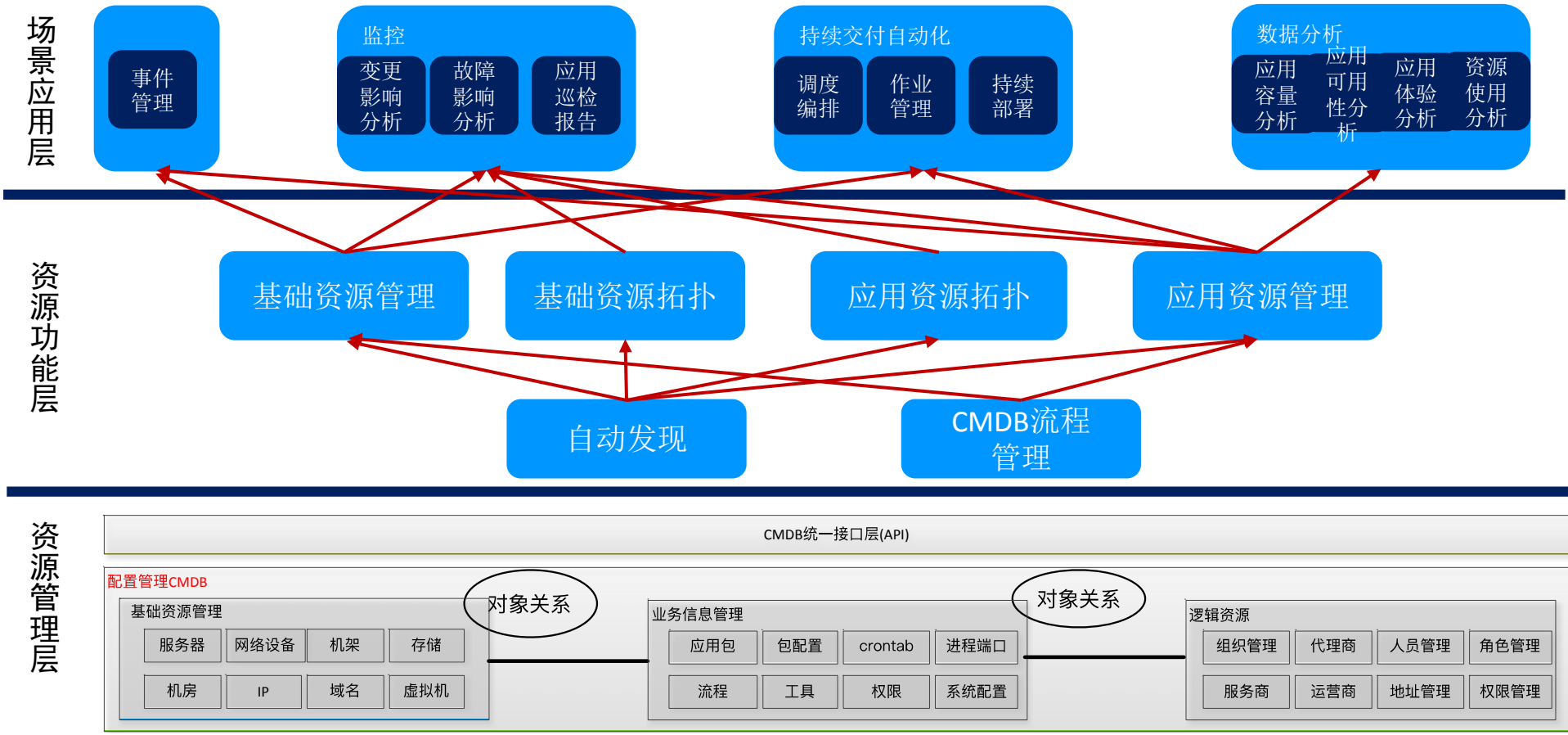
- CMDB架构分基础资源层架构和应用资源层架构
- 应用层资源架构把相关的资源以应用为中心实现资源整合。
- 资源及其资源的关系称之为拓扑（应用拓扑、物理拓扑）
- 资源管理方式有人工维护和自动发现两种方式。流程是人工维护的一种复杂场景和手段。

应用的资源管理模型



- CMDB分核心模型和扩展模型。核心模型是业务、应用、主机和程序包；扩展模型是基于这个实例的关联对象。
- 建立以应用为中心的资源管理模型

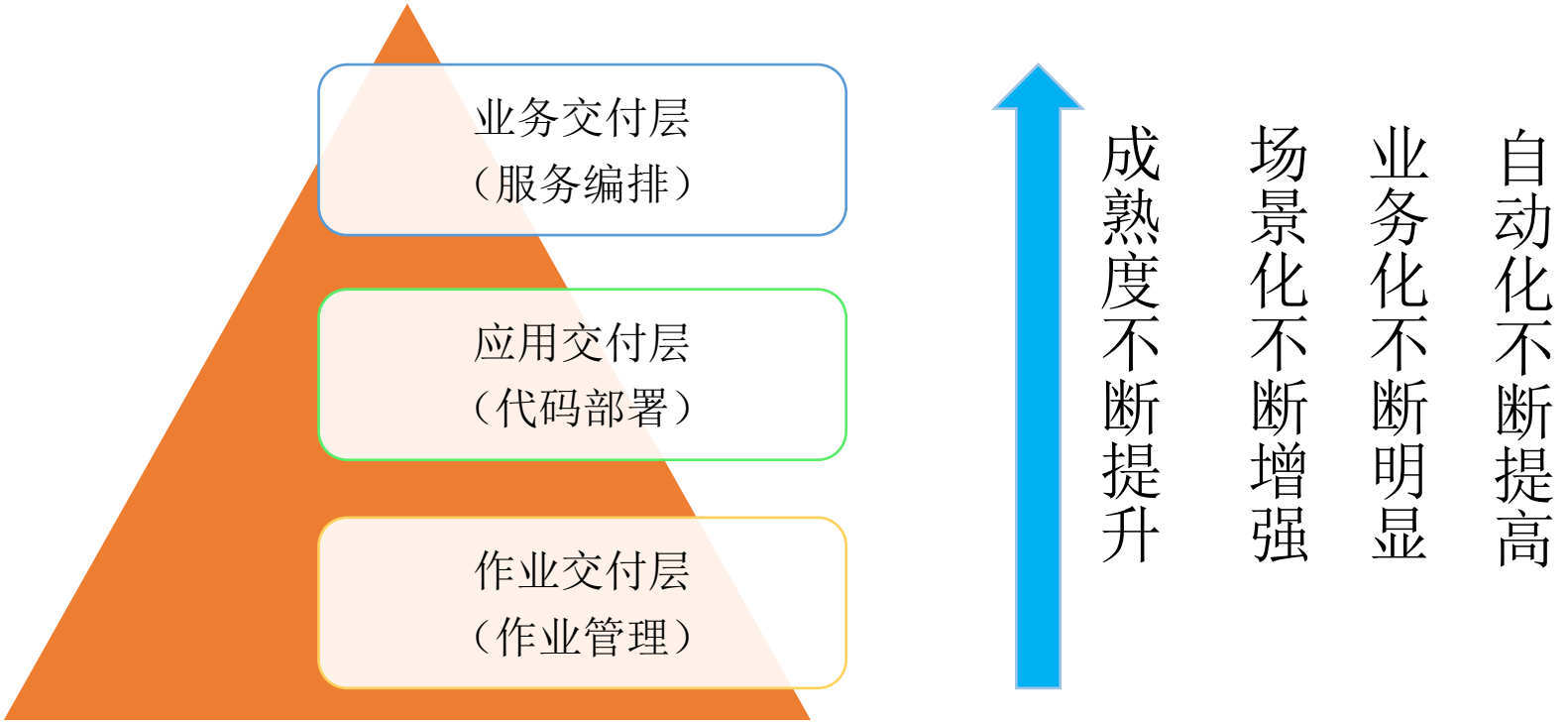
面向应用的资源管理框架



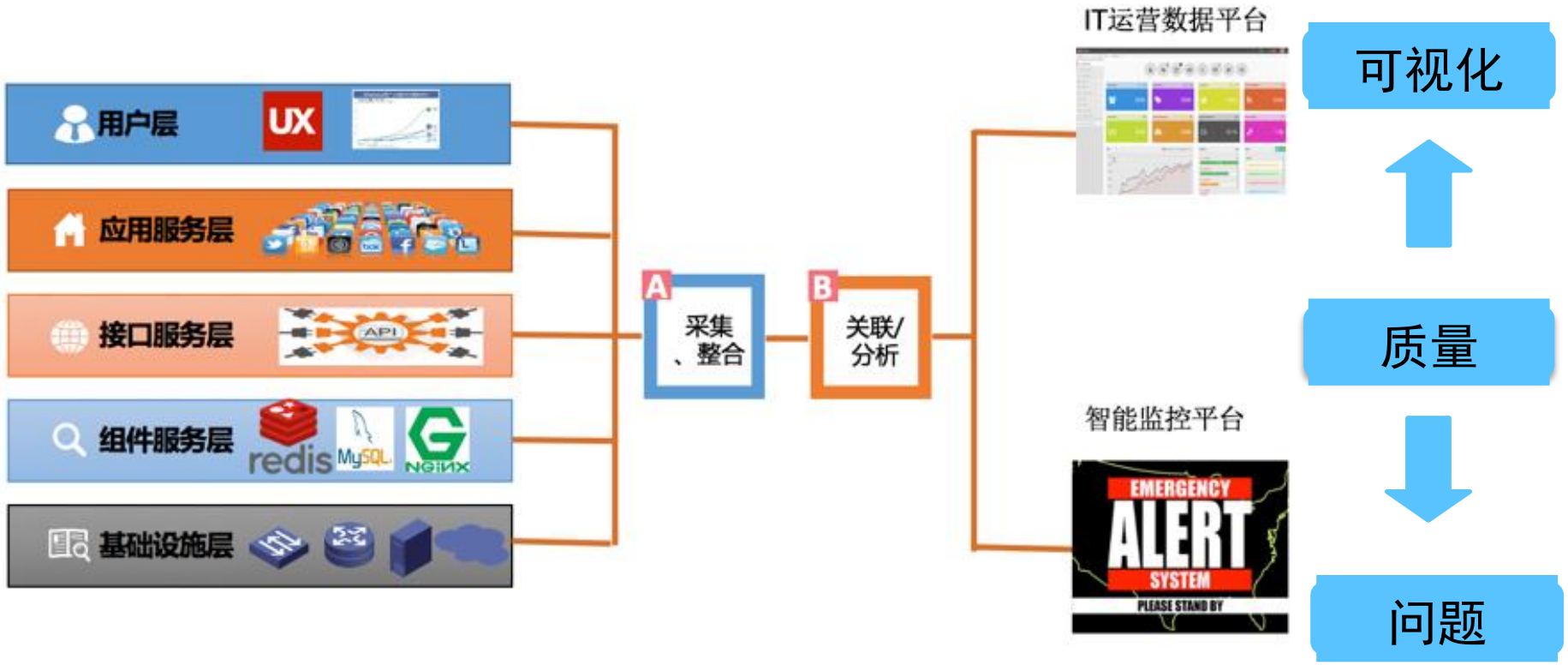
面向应用的动作管理框架

- 作业平台
  - 基本变更单元，控制输入输出。类似配置管理
- 调度平台
  - 支持复杂的编排调度
  - 支持外围能力的插件化集成
- 场景化IT
  - Pipeline［结合应用交付平台，打通持续集成、持续部署、持续反馈］
  - 主机巡检
  - 虚拟化自动变更
  - 故障自愈、故障现场信息采集
  - ……

面向应用的动作管理框架



面向应用的数据管理视图



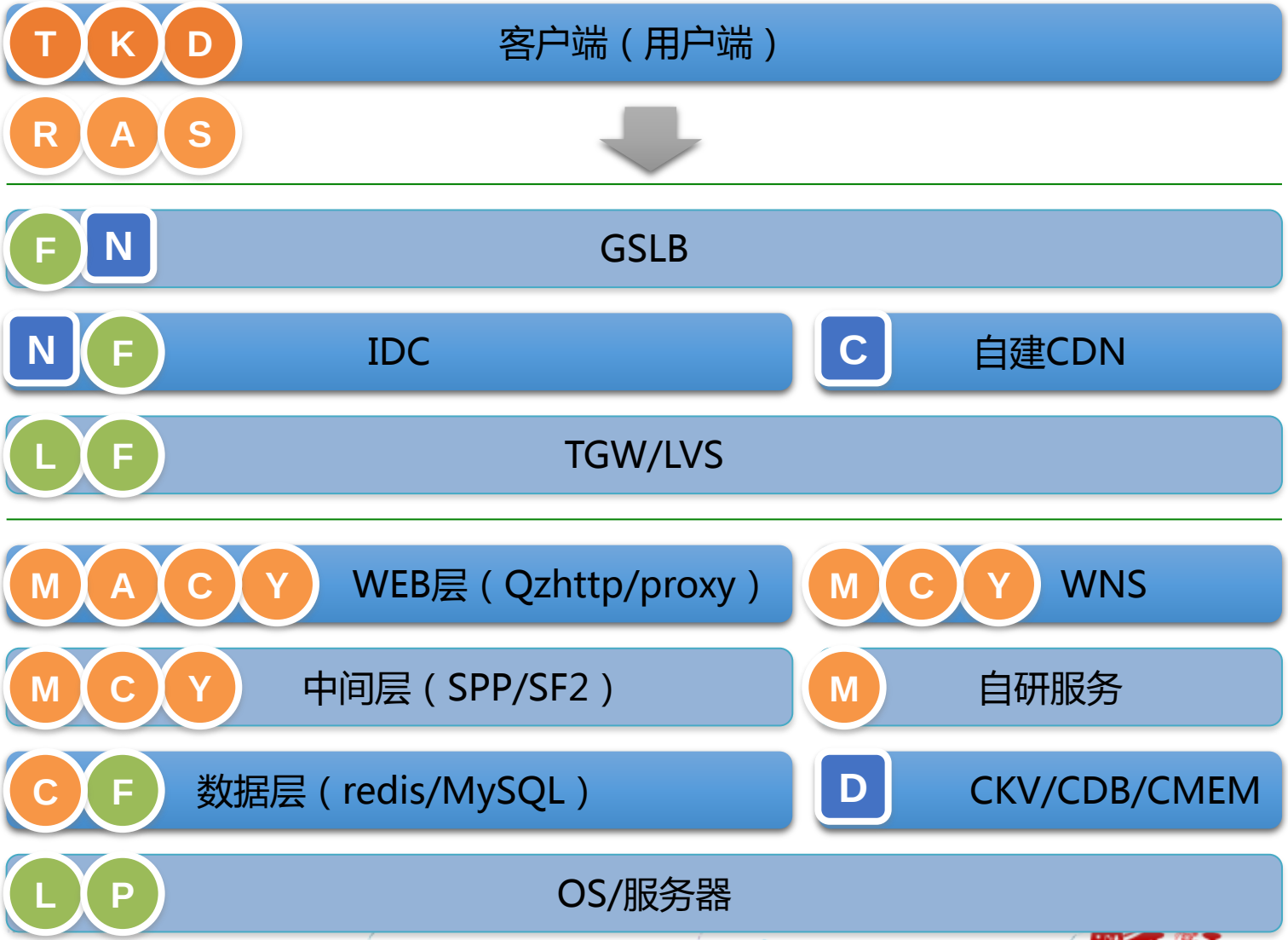
面向应用的立体化监控栈（案例）

TEG服务监控：
N: 网络质量监控
C: CDN监控
D: 数据层监控

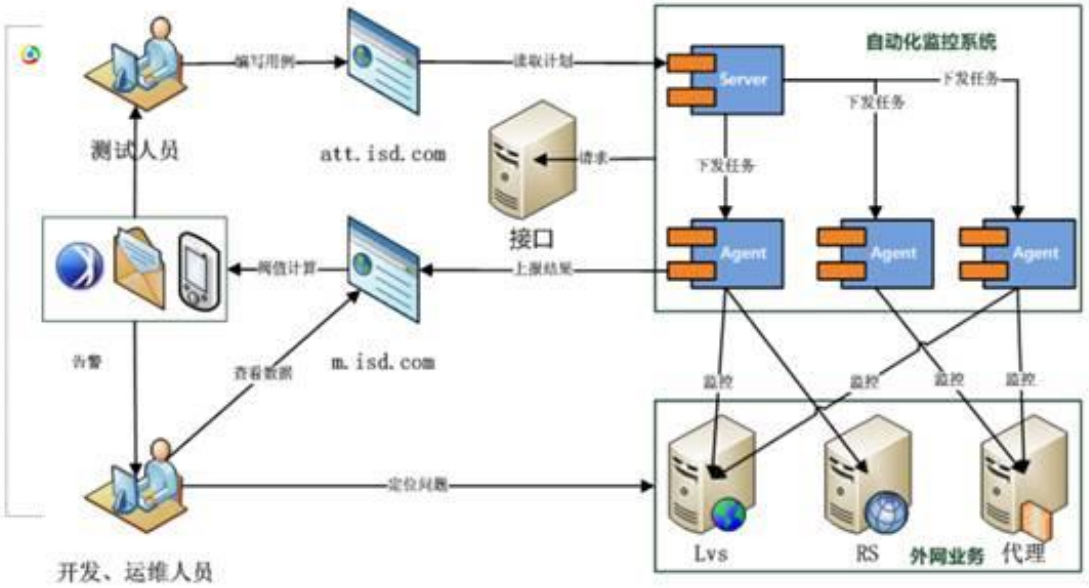
SNG服务监控：
Y: 业务染色监控
R: 返回码监控
S: 测速系统
A: 自动化测试
M: 模块间调用
C: 组件监控

基础监控：
L: 容量管理
P: 进程监控
F: 特性监控

移动端监控：
T: 舆情监控
K: 卡慢监控
D: 多维监控

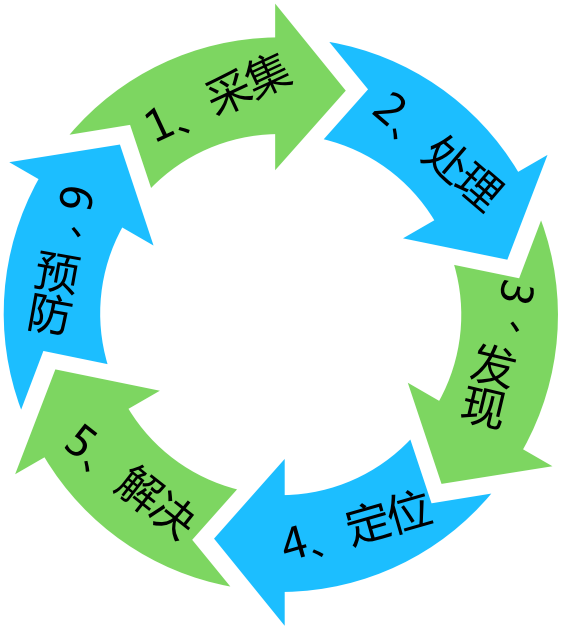


自动化测试



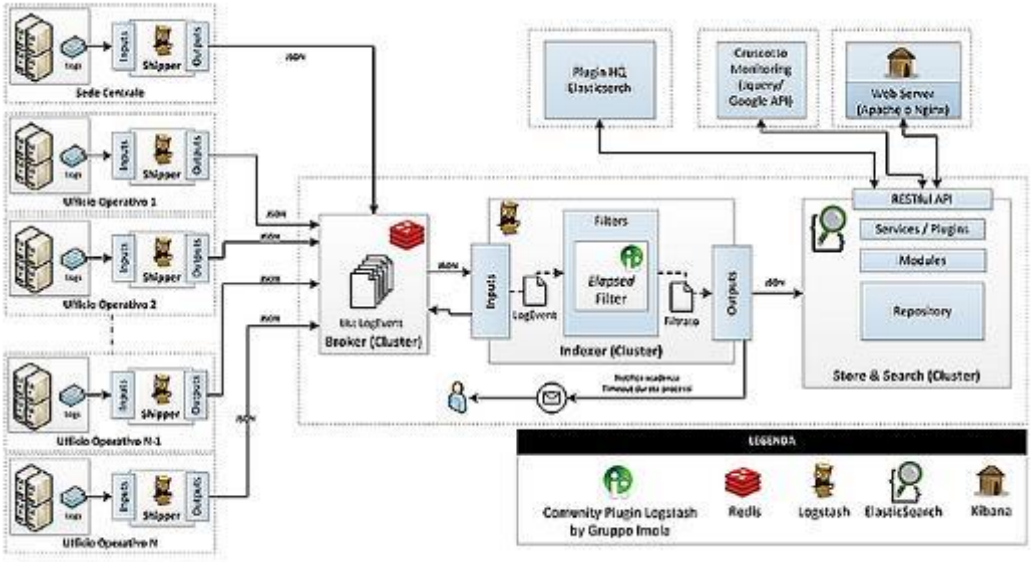
- 模拟用户的行为，做到快速、高效发现故障
- 自动化测试是测试能力到生产环境的延伸，进一步发挥测试价值
- 自动化测试对生产环境的测试需要考虑业务自身的数据隔离性和安全性要求

面向应用的监控管理过程



- 监控是一条能力链，告警能力只是其中的一环。
- 监控的能力是以数据为基础的，应该以标准化的数据处理思路来提升监控能力。
- 监控能力需要闭环

面向应用的数据处理技术栈

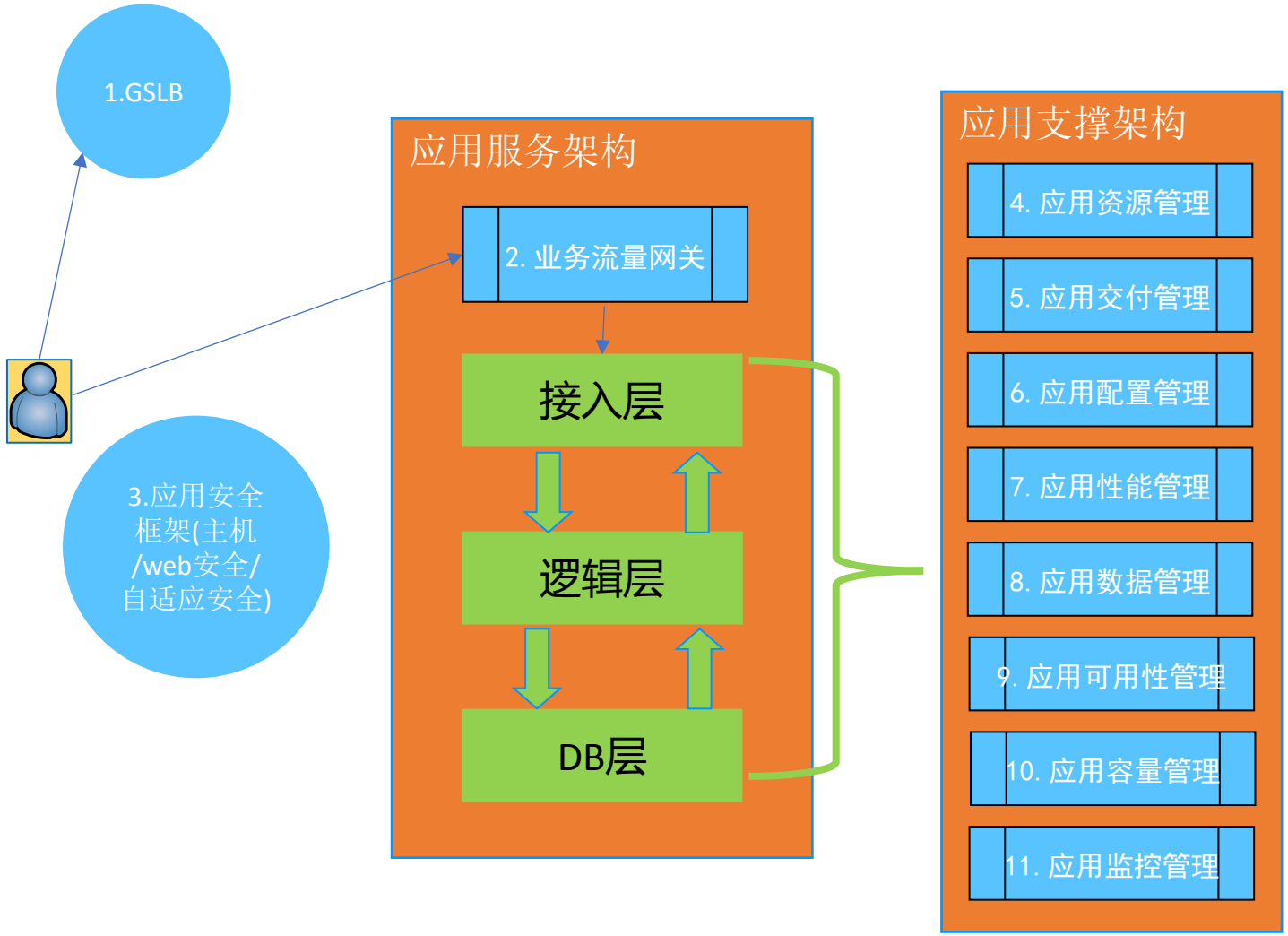


- 应用的日志是一种典型的数据，非结构化
- 应用的日志应该当成流式数据来进行处理
- 应用的日志需要结合实时、多维处理分析，展现数据价值

面向应用的数据看板—DashBoard

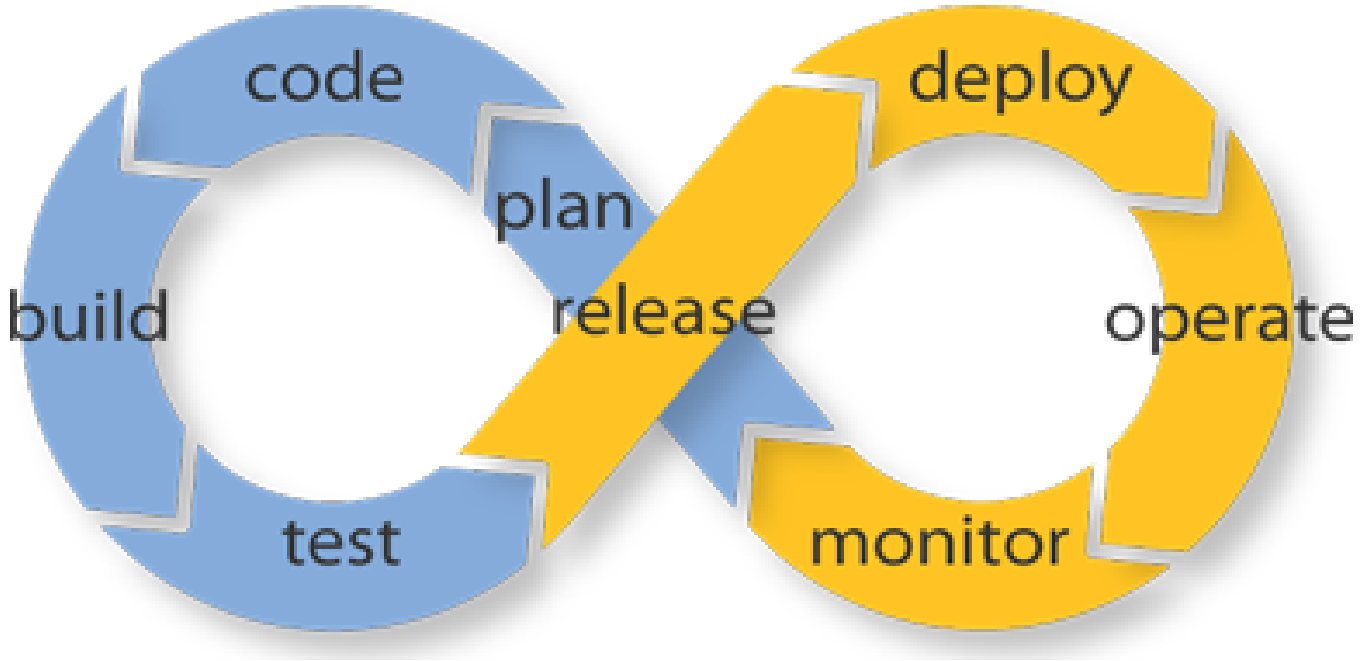
- 数据看板分业务价值看板BVD和IT服务能力看板
- 业务价值看板关注的是业务指标的可视化
- IT服务能力看板关注的是各类技术指标的可视化
- 数据看板是面向不同的角色的数据需求的

面向应用的整体管理框架



目录

面向应用的DevOps持续管理框架

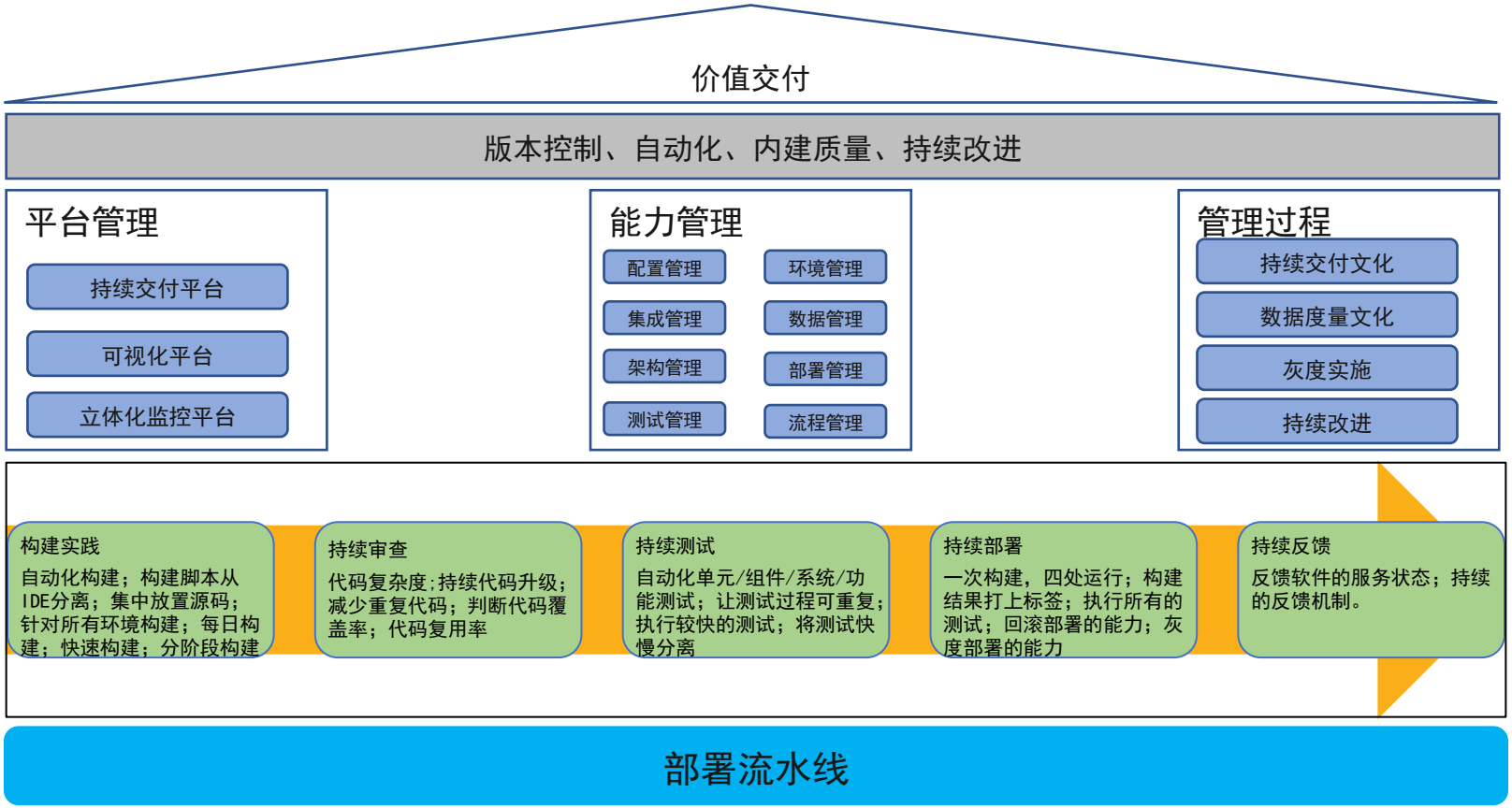


计划->编码->构建->测试

↑ ↓

监控<-运营<-部署<-发布

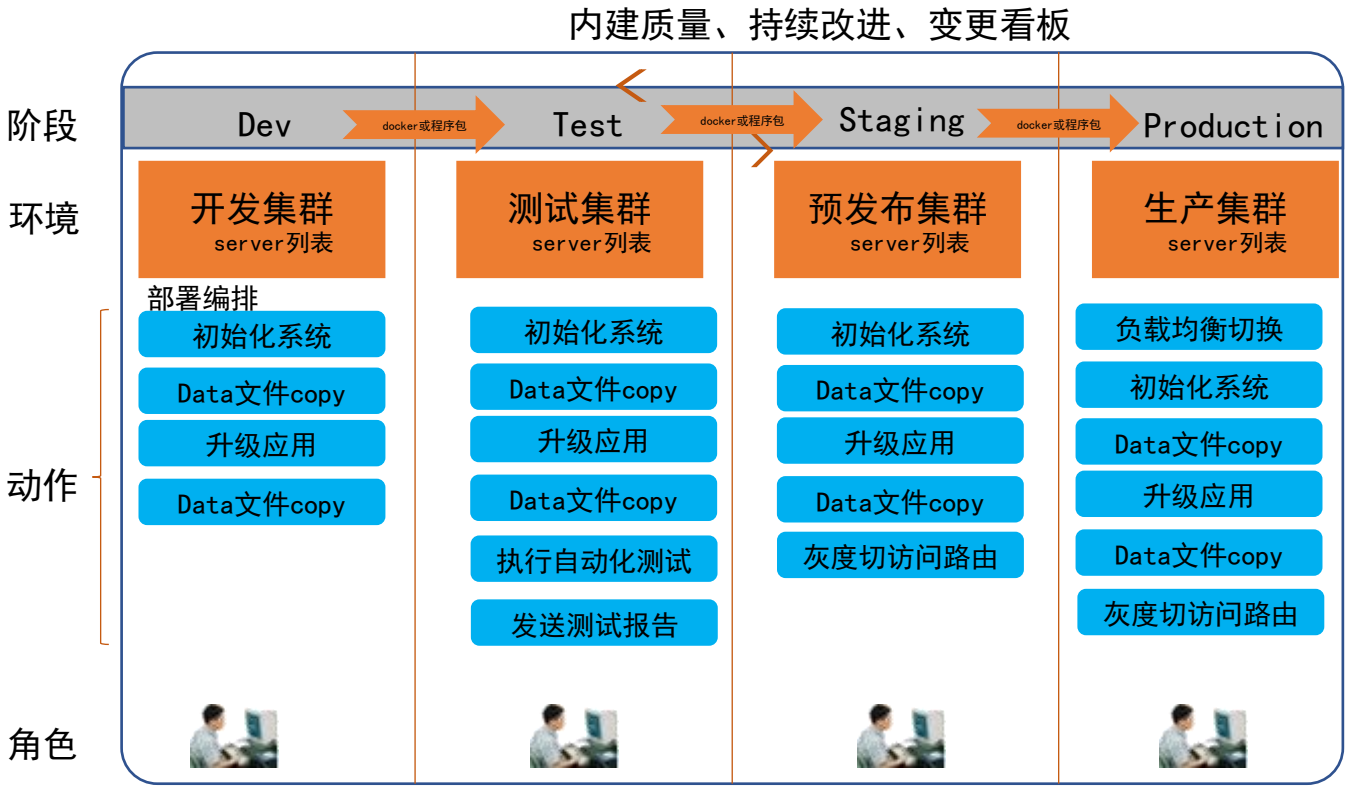
DevOps之最佳实践—持续交付



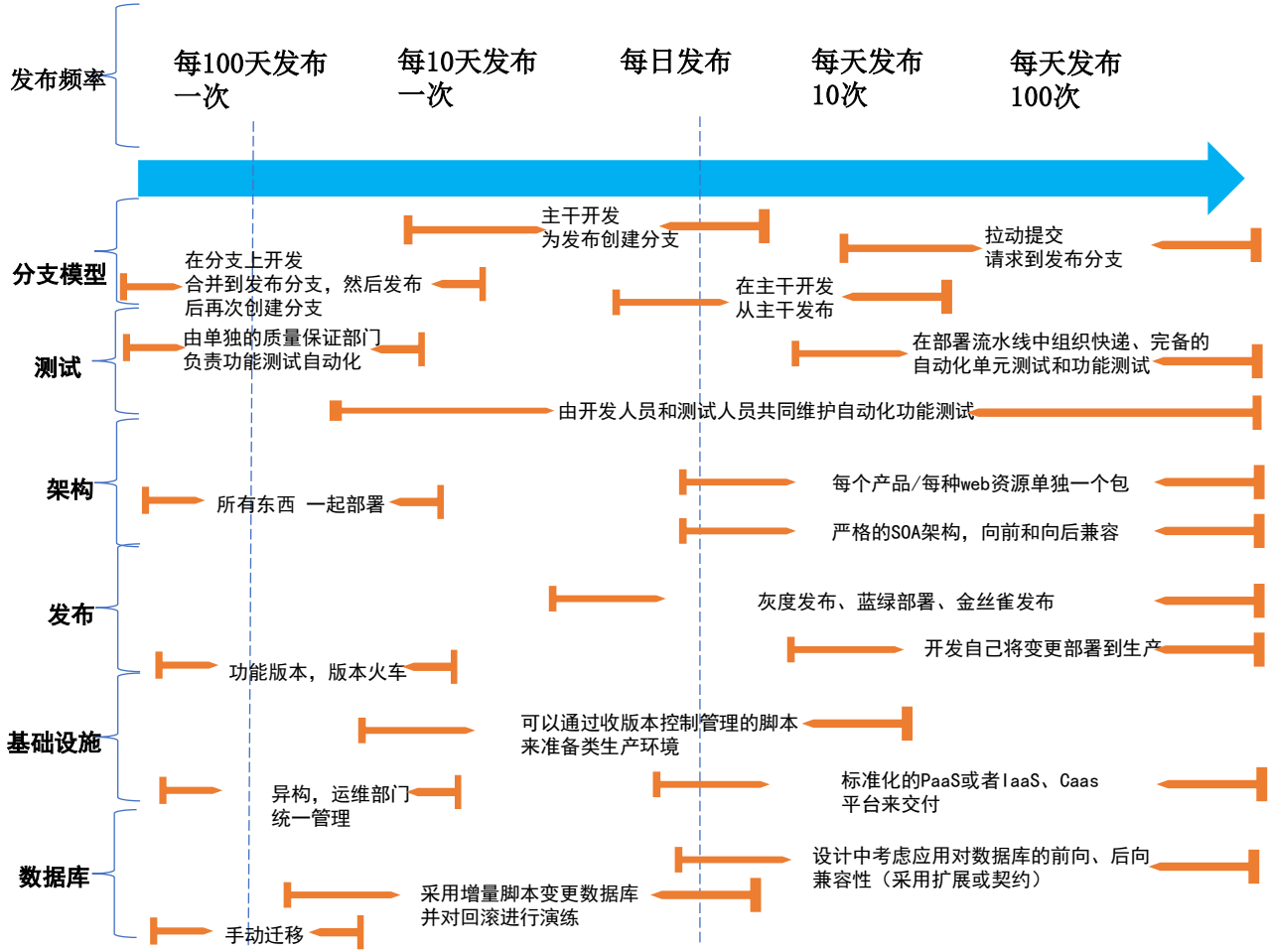
持续交付的原则

- 为软件的发布创建一个可重复且可靠的过程
- 将几乎所有事情自动化
- 把所有的东西都纳入版本控制
- 提前并频繁地做让你感到痛苦的事情
- 内建质量
- “DONE”意味着“已发布”
- 交付过程是每个成员的责任
- 持续改进

持续交付流水线功能架构



持续交付频率与能力关系表

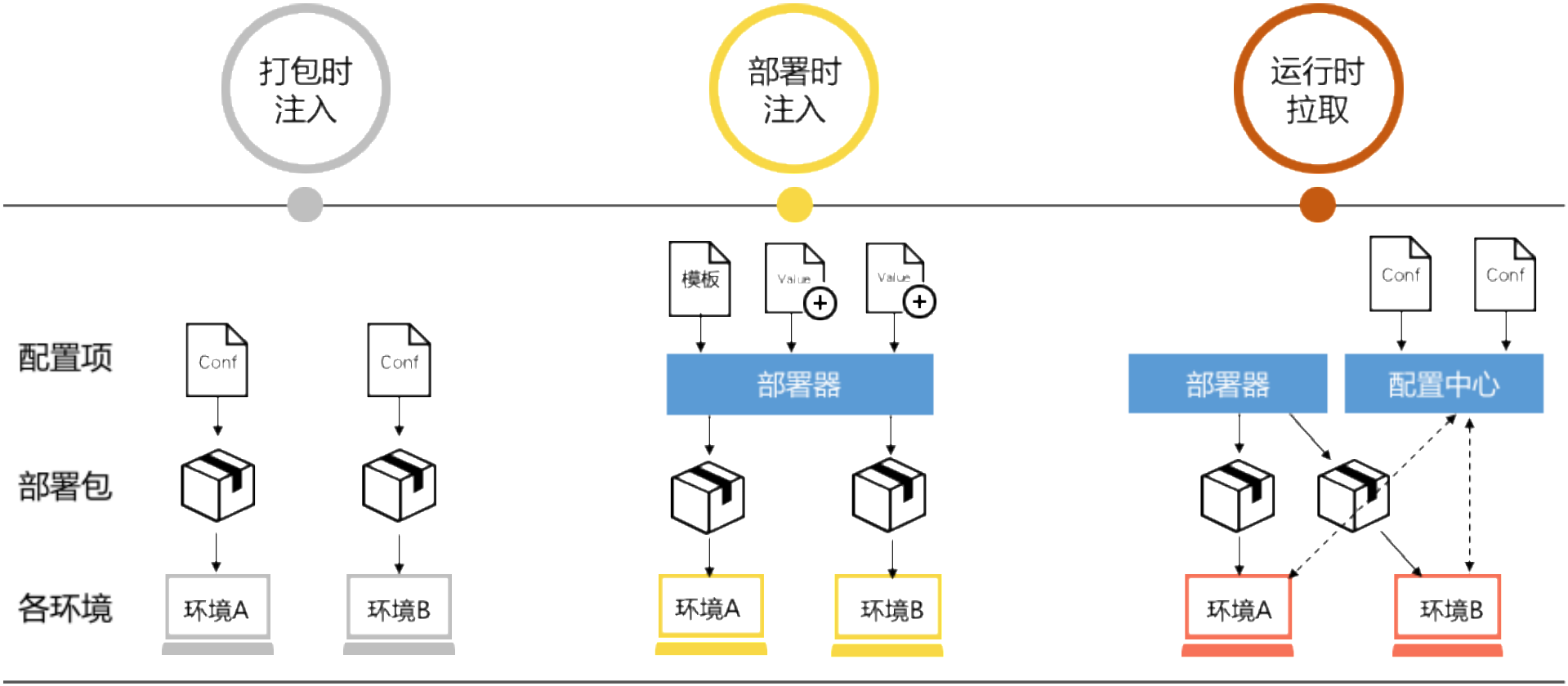


应用的集成管理

构建实践	数据库构建	持续测试	持续审查	持续部署	持续反馈
• 自动化构建 • 构建脚本从IDE分离 • 集中放置源代码 • 标准的目录结构 • 针对所有环境构建 • 使用统一集成构建服务器 • 执行快速构建 • 分阶段构建	• 自动化数据库集成 • 本地数据库沙盒 • DBA成为开发团队的一员	• 自动化单元测试 • 自动化组件测试 • 自动化系统测试 • 自动化功能测试 • 让组件测试可重复 • 先执行较快的测试 • 讲测试进行快慢分组	• 降低代码复杂性 • 持续进行设计审查 • 减少重复的代码 • 判断代码覆盖率 • 持续评估代码复用率	• 随时随地发布可工作软件 • 为库中资产打上标签 • 得到干净的环境 • 执行所有的测试 • 创建构建反馈报告 • 回滚构建的过程能力 • 灰度部署的能力	• 反馈软件的服务状态 • 使用持续的反馈机制

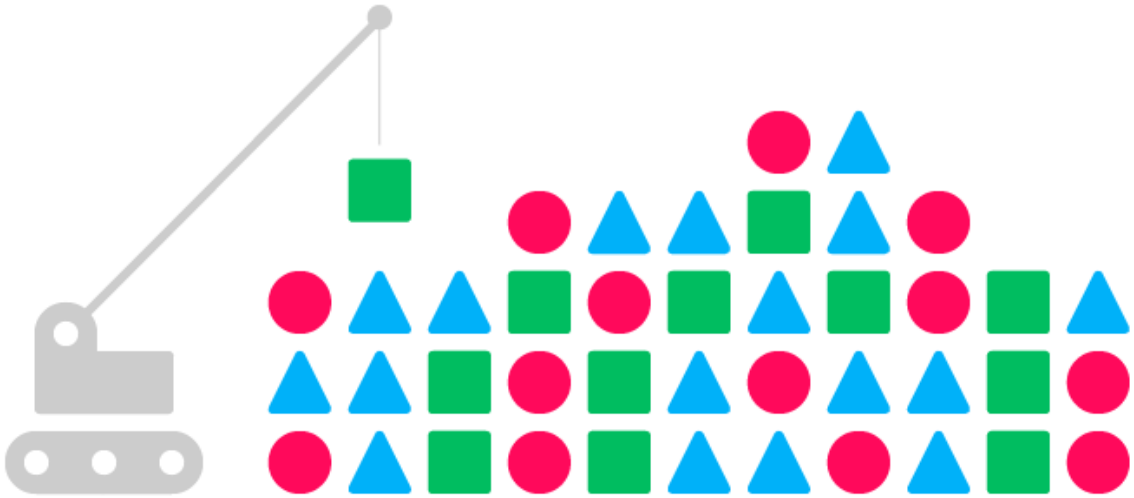
持续交付工具平台（作业自动化）

应用的配置管理



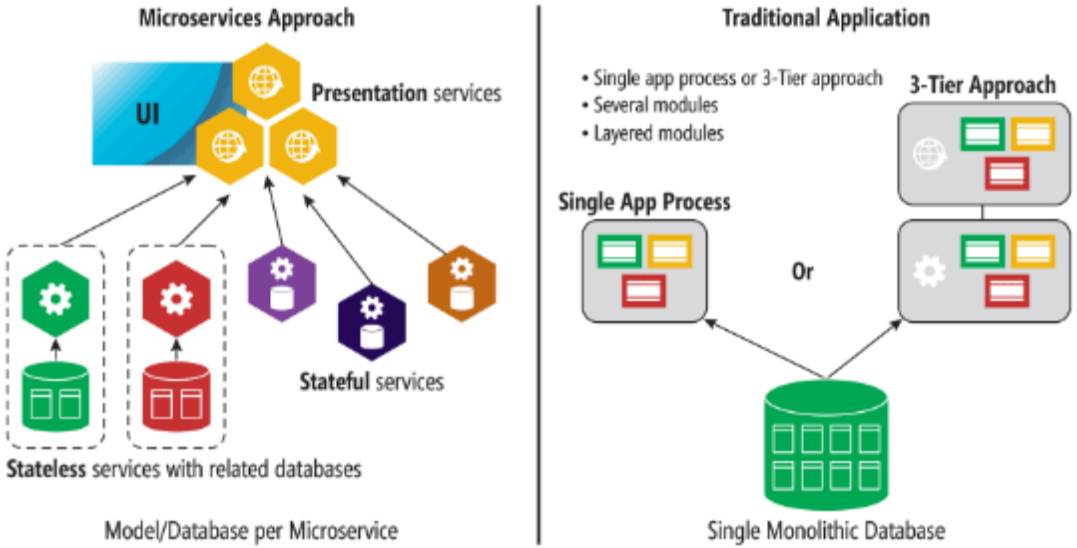
- 原始文件管理（File模式）/配置项管理（KV模式）/分布式配置中心管理

应用的环境管理



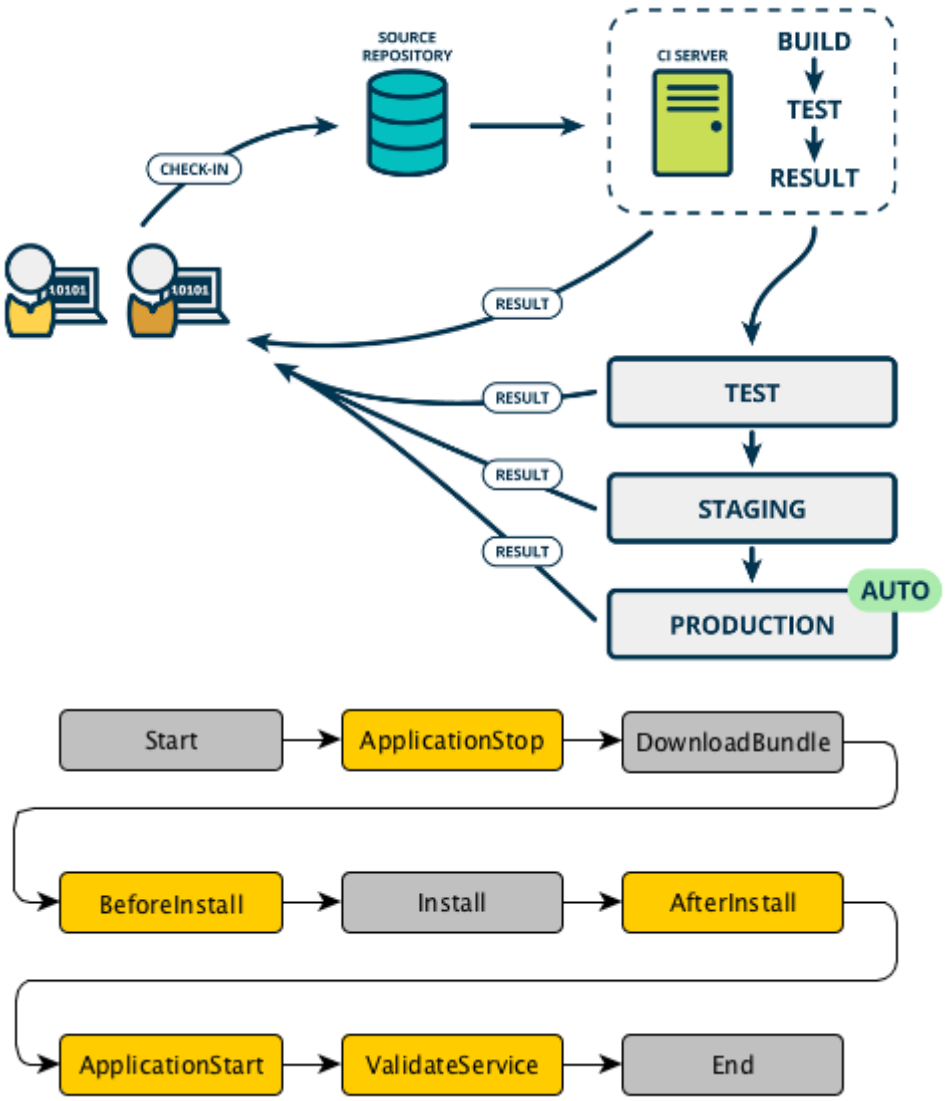
- 应用的环境是资源的一种，如开发、测试、生产环境等等
- 应用的环境管理包括环境的基本管理、环境的参数管理、环境的状态管理。
- 应用的环境管理一致性是核心
- 环境是有生命周期状态的
- 不可变基础设施是环境管理一致性终极手段

应用的架构管理



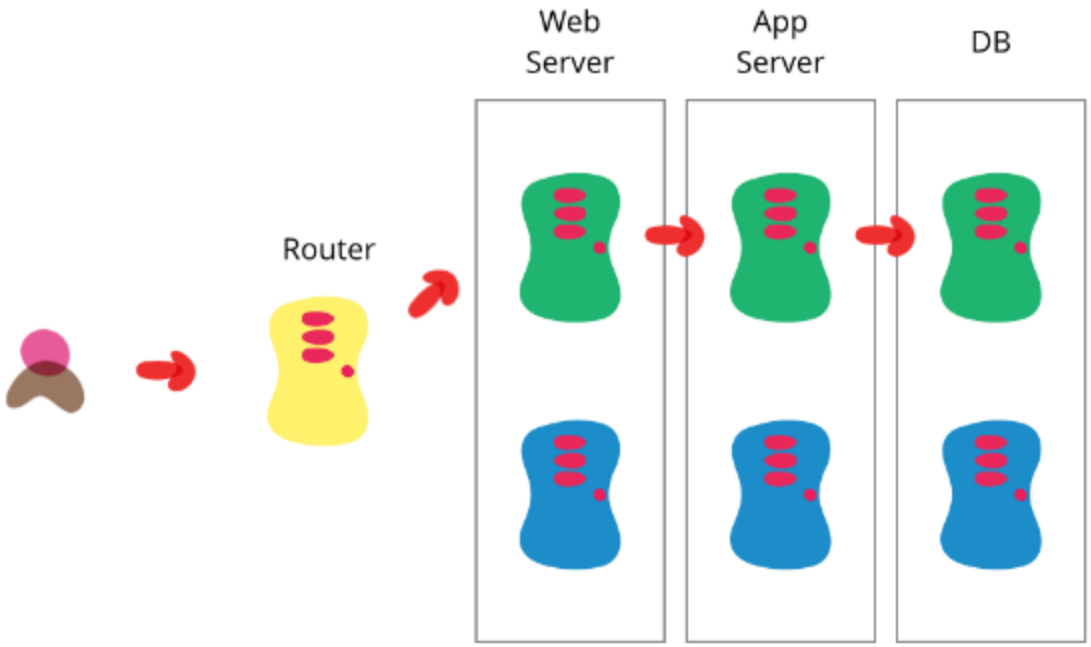
- 微服务化架构的背后是组织形式的折射。
- 微服务的力度是随着应用要支撑的能力变化而变化的
- 微服务是SOA的一种分布式、去中心化、低耦合的表现形式
- 微服务确保了应用自治

应用的部署管理



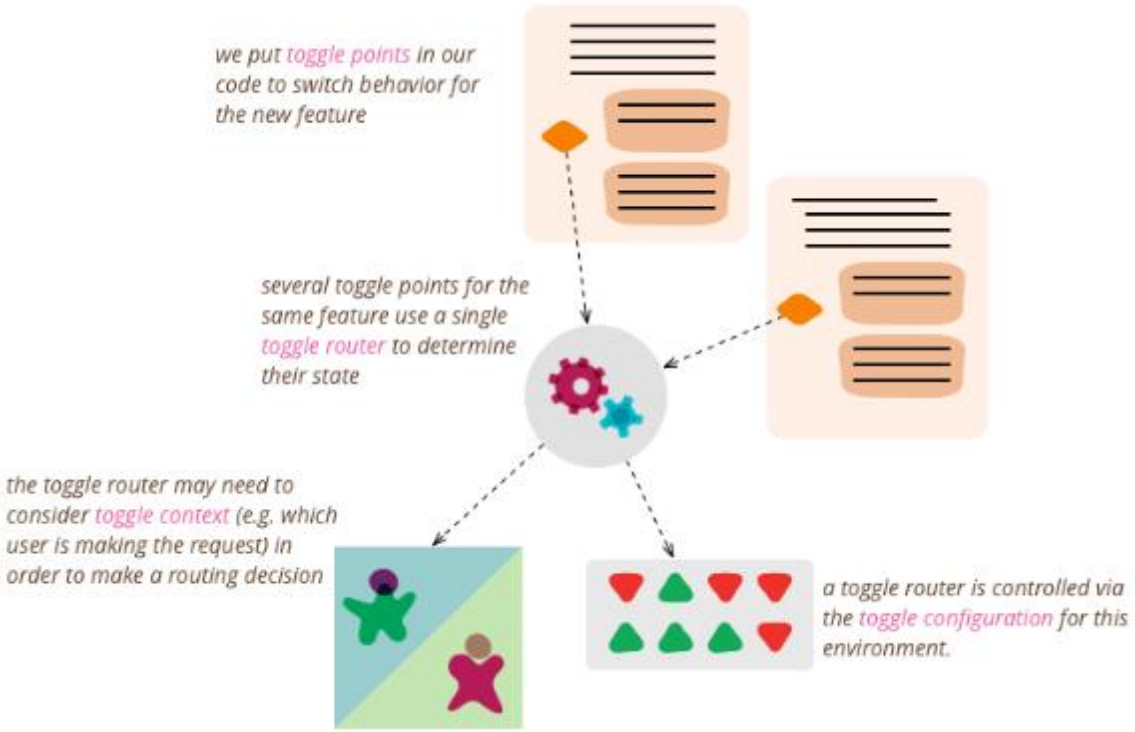
- 发布是针对业务的，部署是针对环境的
- 部署可以频繁、安全、可持续的进行
- 部署的版本包全部来自于人工构建库
- 使用相同的脚本、相同的部署方式对所有环境进行部署，确保一致性
- 为了确保安全性，部署可以采用蓝绿部署、灰度部署等能力。

应用的部署管理之蓝绿部署



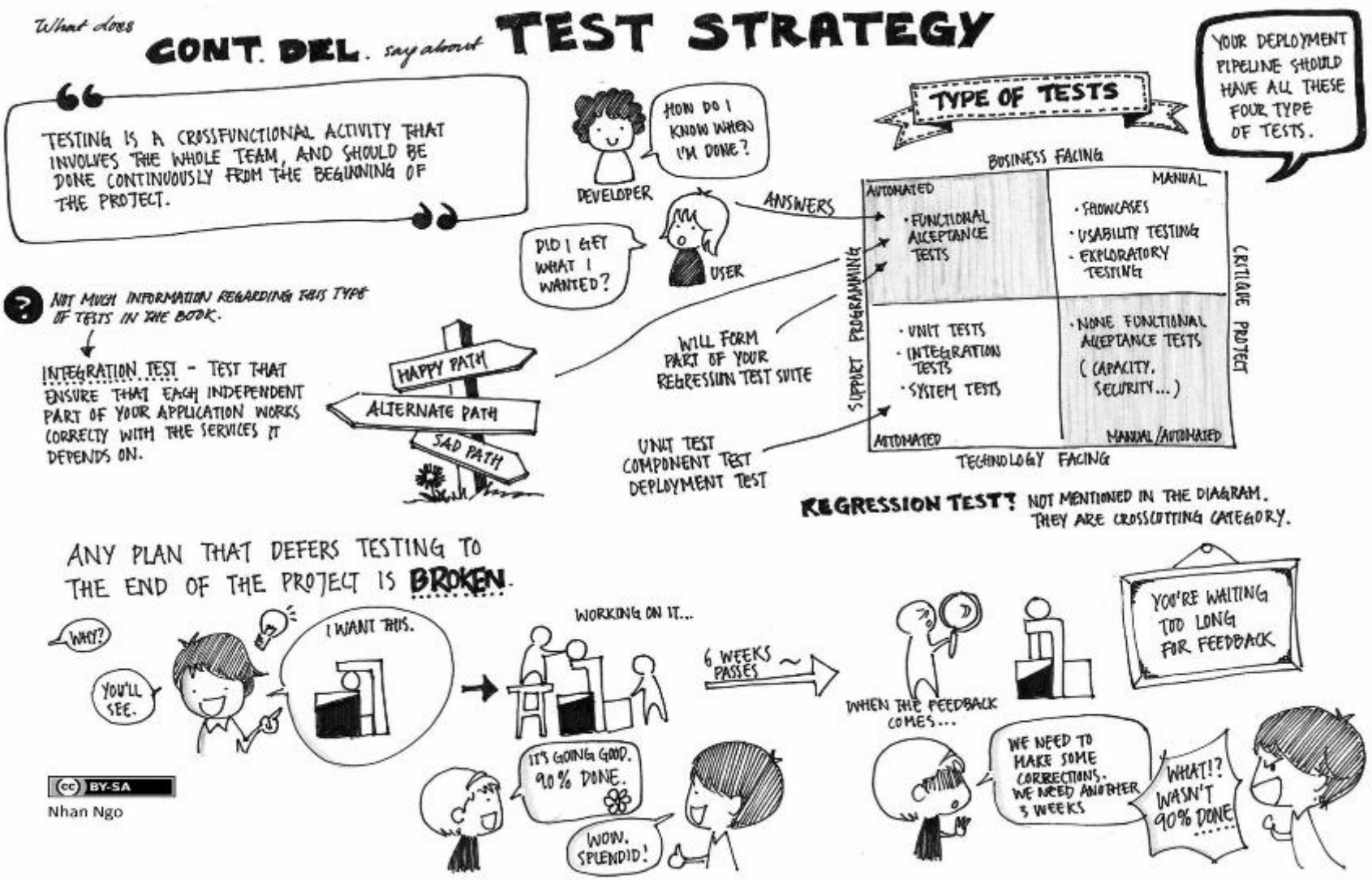
- 蓝绿部署是一种安全的部署行为
- 蓝绿环境是完全隔离的，无论是前端还是后端数据库
- 蓝绿部署的代价是资源代价、环境一致性管理代价、数据管理代价高

应用的部署管理之灰度发布

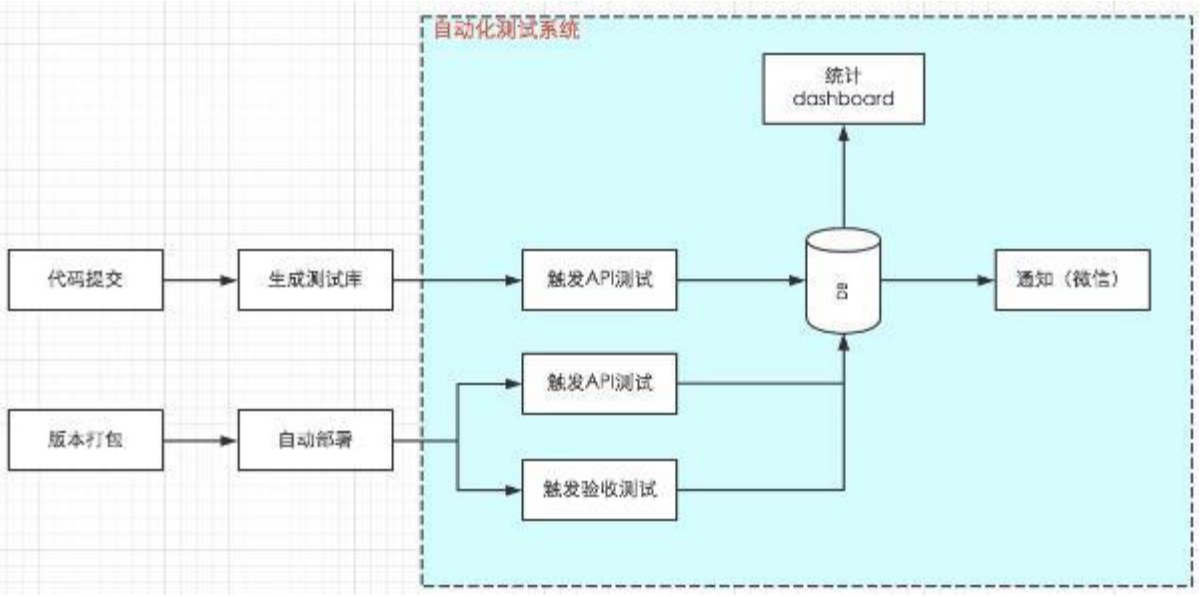


- 灰度发布有多种：金丝雀发布、A/B测试等等
- 灰度发布分成特性灰度、机器灰度、服务灰度等等
- 灰度发布是控制用户来源请求和目标服务之间的匹配关系

应用的测试管理

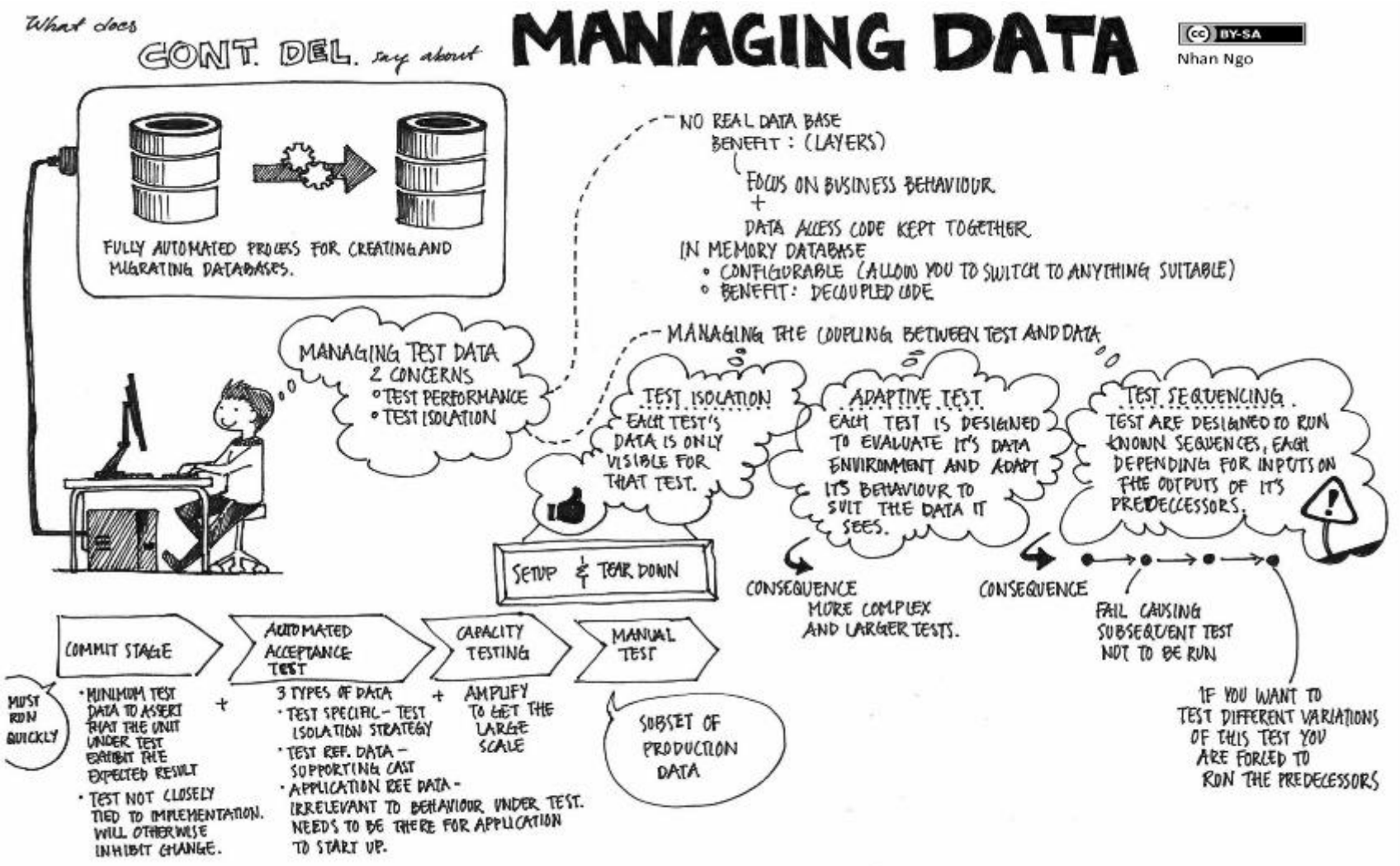

What does CONT. DEL. say about TEST STRATEGY
TESTING IS A CROSSFUNCTIONAL ACTIVITY THAT INVOLVES THE WHOLE TEAM, AND SHOULD BE DONE CONTINUOUSLY FROM THE BEGINNING OF THE PROJECT.
DEVELOPER
HOW DO I KNOW WHEN I'M DONE?
ANSWERS
USER
DID I GET WHAT I WANTED?
TYPE OF TESTS
YOUR DEPLOYMENT PIPELINE SHOULD HAVE ALL THESE FOUR TYPE OF TESTS.
BUSINESS FACING
AUTOMATED
MANUAL
· FUNCTIONAL ACCEPTANCE TESTS
· SHOWCASES
· USABILITY TESTING
· EXPLORATORY TESTING
· UNIT TESTS
· INTEGRATION TESTS
· SYSTEM TESTS
· NONE FUNCTIONAL ACCEPTANCE TESTS (CAPACITY, SECURITY...)
AUTOMATED
MANUAL/AUTOMATED
SUPPORT PROGRAMMING
CRITIQUE PRODUCT
TECHNOLOGY FACING
NOT MUCH INFORMATION REGARDING THIS TYPE OF TESTS IN THE BOOK.
INTEGRATION TEST - TEST THAT ENSURE THAT EACH INDEPENDENT PART OF YOUR APPLICATION WORKS CORRECTLY WITH THE SERVICES IT DEPENDS ON.
HAPPY PATH
ALTERNATE PATH
SAD PATH
WILL FORM PART OF YOUR REGRESSION TEST SUITE
UNIT TEST
COMPONENT TEST
DEPLOYMENT TEST
REGRESSION TEST? NOT MENTIONED IN THE DIAGRAM. THEY ARE CROSSCUTTING CATEGORY.
ANY PLAN THAT DEFERS TESTING TO THE END OF THE PROJECT IS BROKEN.
WHY?
YOU'LL SEE.
I WANT THIS.
WORKING ON IT...
6 WEEKS PASSES
WHEN THE FEEDBACK COMES...
YOU'RE WAITING TOO LONG FOR FEEDBACK
IT'S GOING GOOD. 90% DONE.
WOW. SPLENDID!
WE NEED TO MAKE SOME CORRECTIONS. WE NEED ANOTHER 3 WEEKS
WHAT!? WASN'T 90% DONE
BY-SA
Nhan Ngo

平台的自动化测试（案例）

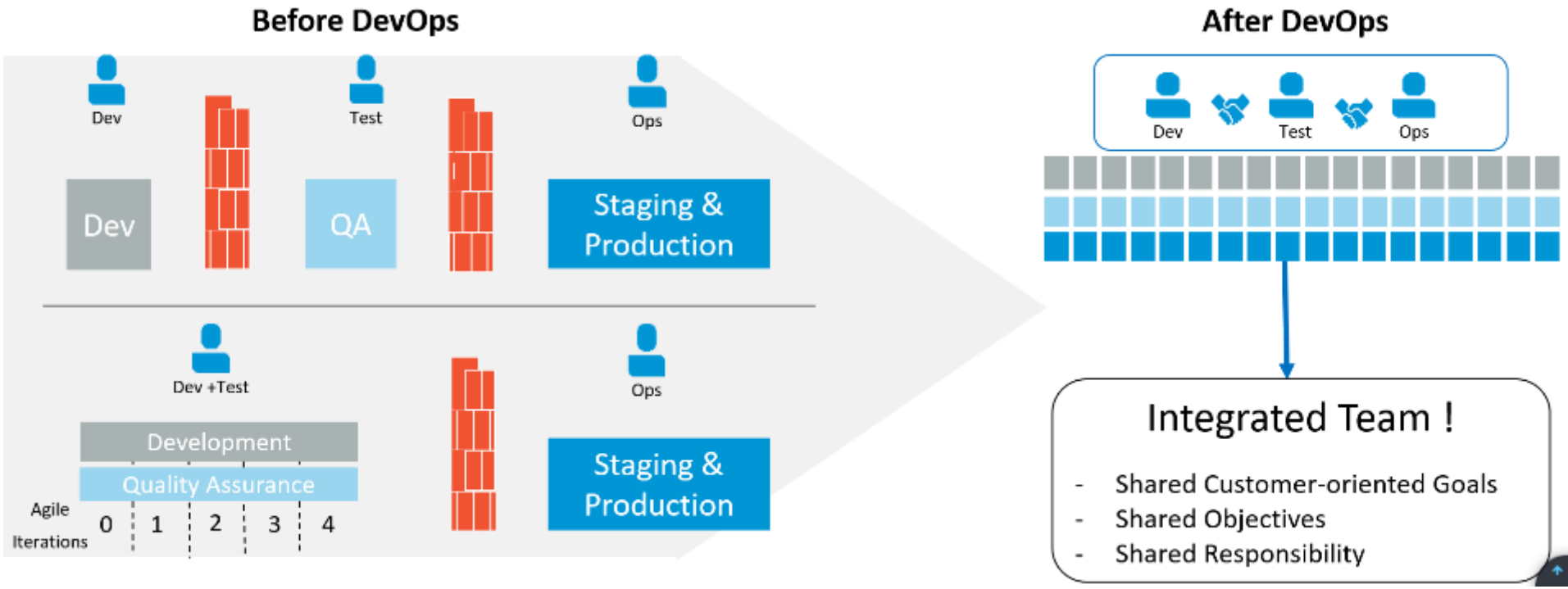


- 自动化测试是CI过程的关键实践
- 基于验收标准自动化生成测试库（通过代码注释完成）
- 验收测试是针对Happy Path的全流程测试；
- API测试接口级别的功能测试

应用的数据管理


What does CONT. DEL. say about
MANAGING DATA
BY-SA
Nhan Ngo
FULLY AUTOMATED PROCESS FOR CREATING AND MIGRATING DATABASES.
NO REAL DATA BASE
BENEFIT : (LAYERS)
FOCUS ON BUSINESS BEHAVIOUR
+
DATA ACCESS CODE KEPT TOGETHER
IN MEMORY DATABASE
• CONFIGURABLE (ALLOW YOU TO SWITCH TO ANYTHING SUITABLE)
• BENEFIT: DECOUPLED CODE
MANAGING TEST DATA
2 CONCERNS
• TEST PERFORMANCE
• TEST ISOLATION
MANAGING THE COUPLING BETWEEN TEST AND DATA
TEST ISOLATION
EACH TEST'S DATA IS ONLY VISIBLE FOR THAT TEST.
ADAPTIVE TEST
EACH TEST IS DESIGNED TO EVALUATE IT'S DATA ENVIRONMENT AND ADAPT IT'S BEHAVIOUR TO SUIT THE DATA IT SEES.
TEST SEQUENCING
TEST ARE DESIGNED TO RUN KNOWN SEQUENCES, EACH DEPENDING FOR INPUTS ON THE OUTPUTS OF IT'S PREDECESSORS.
SETUP & TEAR DOWN
CONSEQUENCE
MORE COMPLEX AND LARGER TESTS.
CONSEQUENCE
FAIL CAUSING SUBSEQUENT TEST NOT TO BE RUN
IF YOU WANT TO TEST DIFFERENT VARIATIONS OF THIS TEST YOU ARE FORCED TO RUN THE PREDECESSORS
COMMIT STAGE
MUST RUN QUICKLY
• MINIMUM TEST DATA TO ASSERT THAT THE UNIT UNDER TEST EXHIBIT THE EXPECTED RESULT
• TEST NOT CLOSELY TIED TO IMPLEMENTATION. WILL OTHERWISE INHIBIT CHANGE.
+
AUTOMATED ACCEPTANCE TEST
3 TYPES OF DATA
• TEST SPECIFIC – TEST ISOLATION STRATEGY
• TEST REF. DATA – SUPPORTING CAST
• APPLICATION REF DATA – IRRELEVANT TO BEHAVIOUR UNDER TEST. NEEDS TO BE THERE FOR APPLICATION TO START UP.
+
CAPACITY TESTING
AMPLIFY TO GET THE LARGE SCALE
MANUAL TEST
SUBSET OF PRODUCTION DATA

持续交付的应用全局视角



- 从瀑布到敏捷到DevOps，是IT工程模式的变化
- DevOps基于持续交付、文化、工具、组织等多个角度，实现了面向应用的全局端到端能力构建

目录

DevOps关注业务目标

想法

快速的交付想法

尝试与反馈

让客户尽快尝试，从而获得反馈

改进

快速响应客户反馈

DevOps更快交付业务价值

- 在服务交付价值链打破孤岛
- 整合开发和运维的能力为一个协作的团队
- 端到端服务交付流程的变革
- 对新的应用和服务，加快实现价值的时间
- 不影响安全性，兼容性，性能

DevOps关注IT价值

多快好省

DevOps驱动IT性能提升

01 lead time for changes
变更时长

02 release frequency
发布频率

03 time to restore service
服务恢复时长

04 change fail rate
变更失败率

- （局部/辅助）自动化测试覆盖率
- （局部/辅助）代码库特征
- （局部/辅助）缺陷数量
- （局部/辅助）交付速度
- （局部/辅助）构建情况：成功、失败、时间
- （局部/辅助）提交次数

高效运维社区

GreatOPS Community

GOPS 4月深圳 / 7月北京 / 11月上海

EXIN DevOps Master 认证研修

DevOpsDays 3月北京 / 8月上海

DevOps 企业内训 / 咨询服务

DevOps China 全国巡回技术沙龙

其他量身定制服务项目

商务经理：刘静女士
电话 / 微信：13021082989
邮箱：liujing@greatops.com

Thanks

荣誉出品

高效运维社区

国际最佳实践管理联盟