

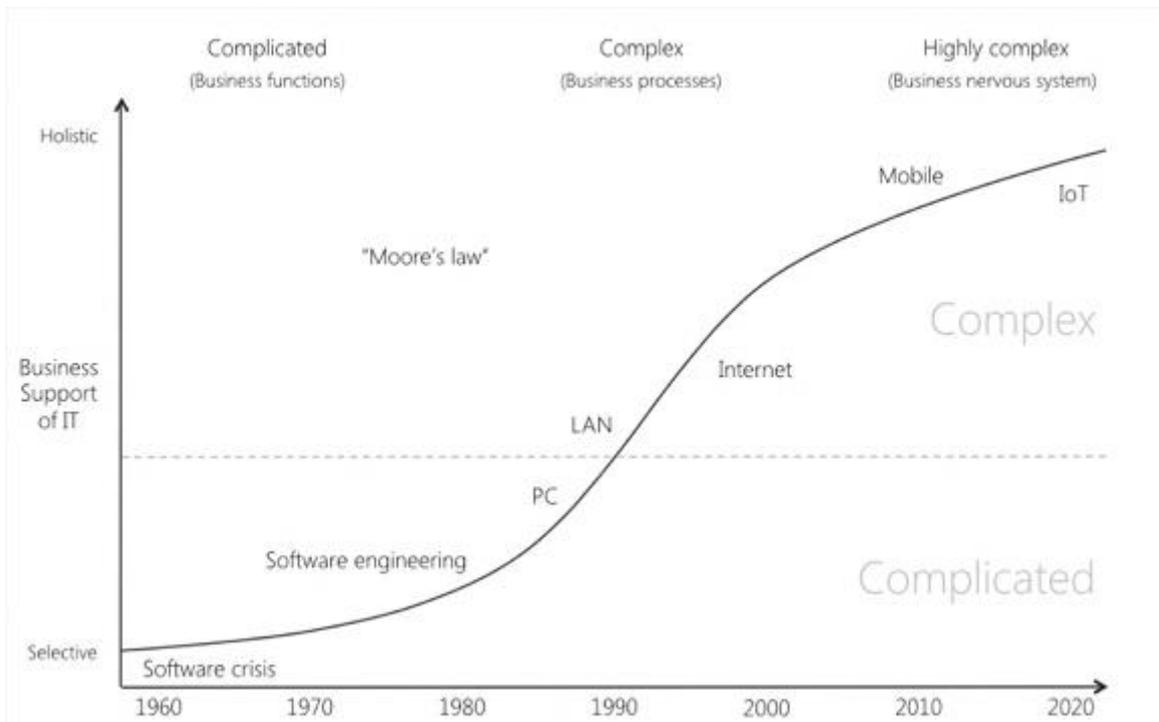


大型互联网项目的持续交付模式

技术创新，变革未来

重新思考现代企业中IT的价值

IT已经从企业内部的一种运营工具，逐步演进为一种竞争优势

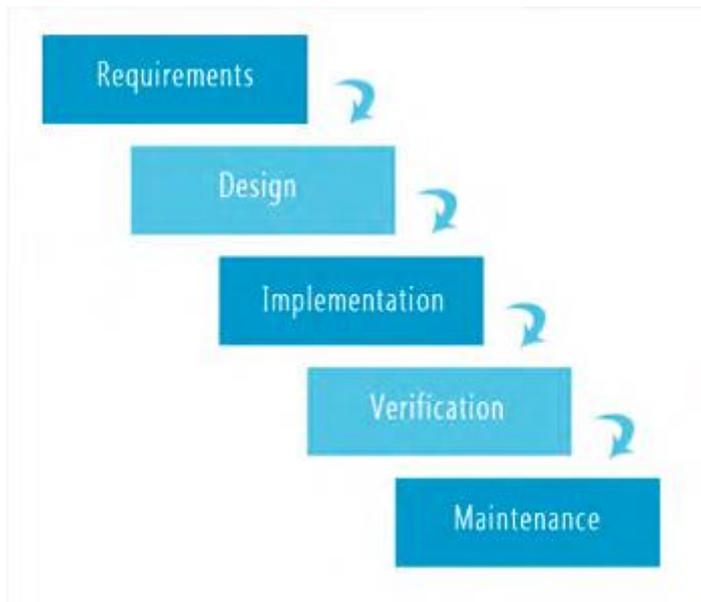


Evolution of IT

- 传统IT项目管理模式有其时代背景，越来越难适应快速创新周期
- 现代企业IT需要达到以下目标：
 - 快速，降低交付周期
 - 灵活响应变化
 - 高可用和适应性
 - 聚焦价值
 - 持续改进

面向目标，如何转型？

启动和实施敏捷转型：研发模型



严格阶段划分，每个阶段控制准入和准出

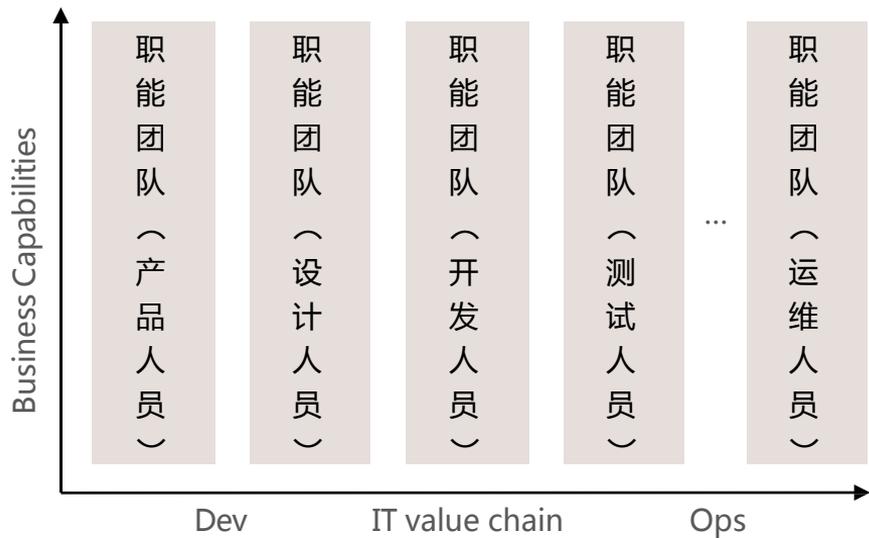


迭代开发，拥抱变化，快速反馈，聚焦价值



面向目标，如何转型？

启动和实施敏捷转型：组织结构



分工细化，专业化运作，各领域深耕

缺少整体目标感，工作交接和确认，出问题必须界定清楚



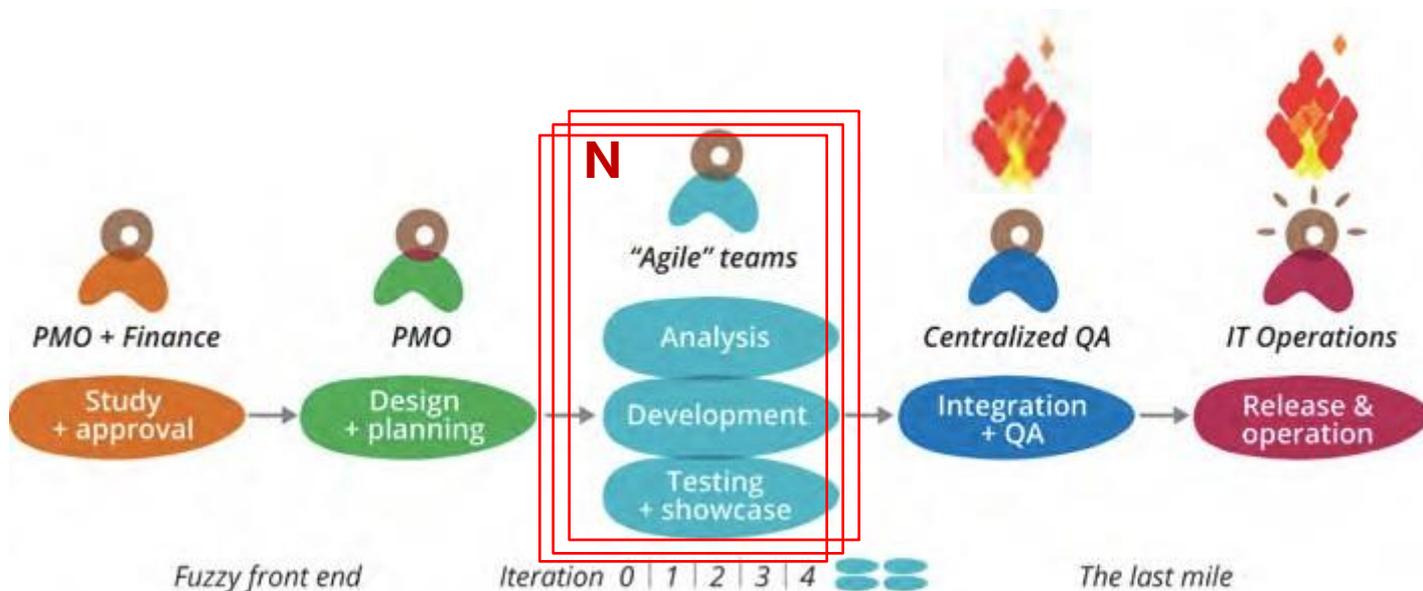
通过目标驱动，不同角色共享责任，团队自组织

端到端负责，减少交接损耗，降低问题解决成本

实际的转型效果怎样？

在很多具备一定复杂度的企业IT环境中，局部优化很难产生显著成效

- 很多敏捷转型的改进主要局限在开发测试，但未端到端的打通
- 受架构限制，缺失并行交付能力，集成点很可能是全局阻塞点

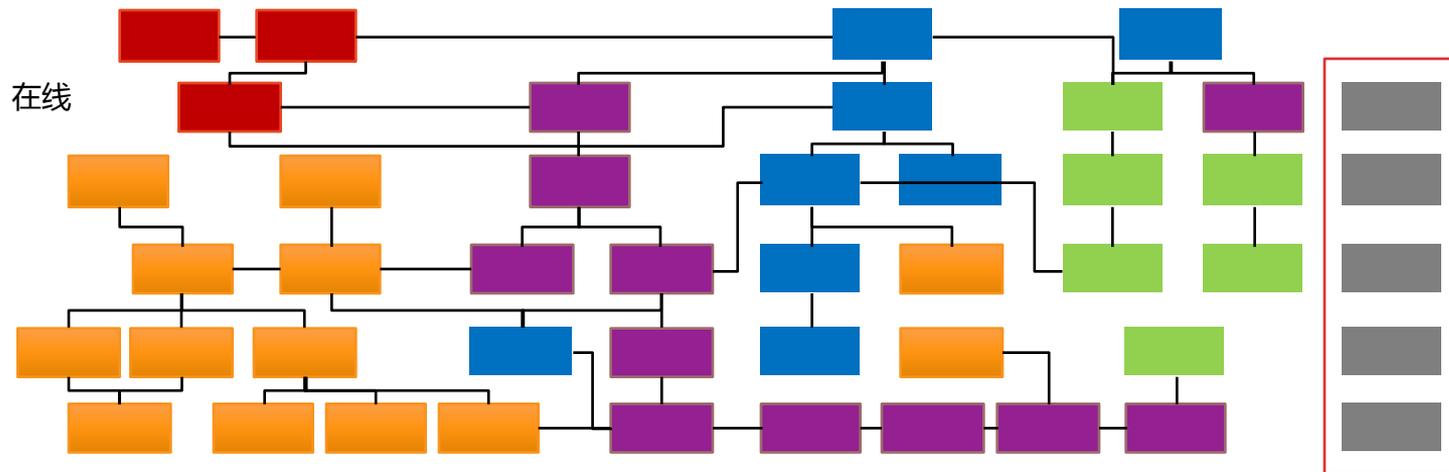


分析一个实际的案例

某互联网+业态下的典型产品

- 业务链条较长，交易涉及多系统间协作，整体复杂度较高
- 已经启动了敏捷转型，按迭代方式运作，但改进效果有限

涉及团队：6+ 人员规模：数百 相关子系统：50+



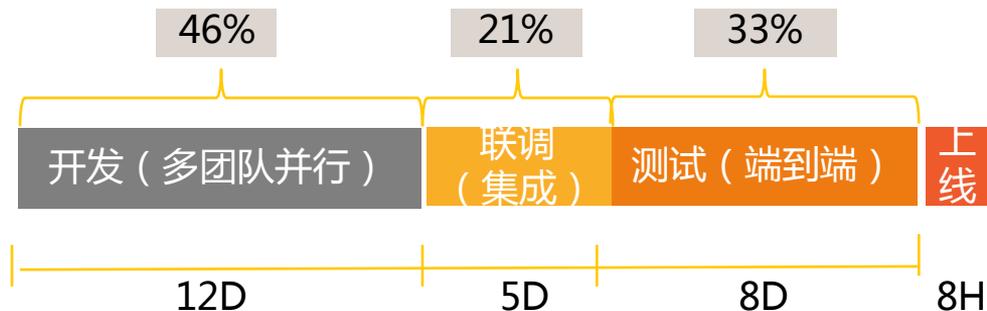
分析一个实际的案例

问题现象：

- 整个版本周期较长，有时一个月才能发布大版本
- 线上常有质量问题，需消耗较多精力排查和修复

问题分析：

- 各阶段周期时间分解
- 关注等待/阻塞/浪费



问题定位：

系统紧耦合，相互影响和阻塞

- 多个子系统无法做到并行交付
- 故障传播不可控，相互影响

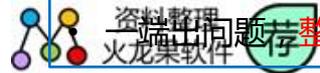
资料整理
火龙果软件
一端出问题，整个版本阻塞

各端未控制质量，集成时问题爆发

- 开发自测不足，缺陷蔓延到集成阶段
- 自动化能力缺失，大量依赖人工处理
- 团队间存在资源争抢，相互冲突严重

环境交付未归一化，发布效率低

- 环境多依赖复杂，线下线上不一致
- 环境准备时间长，维护成本高
- 串行发布上线，需要长时间停服



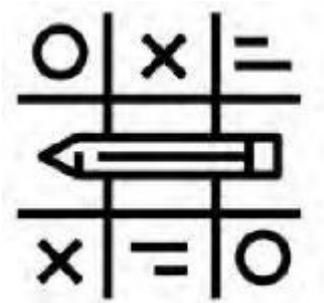
DevOps的解决思路



全局化视角



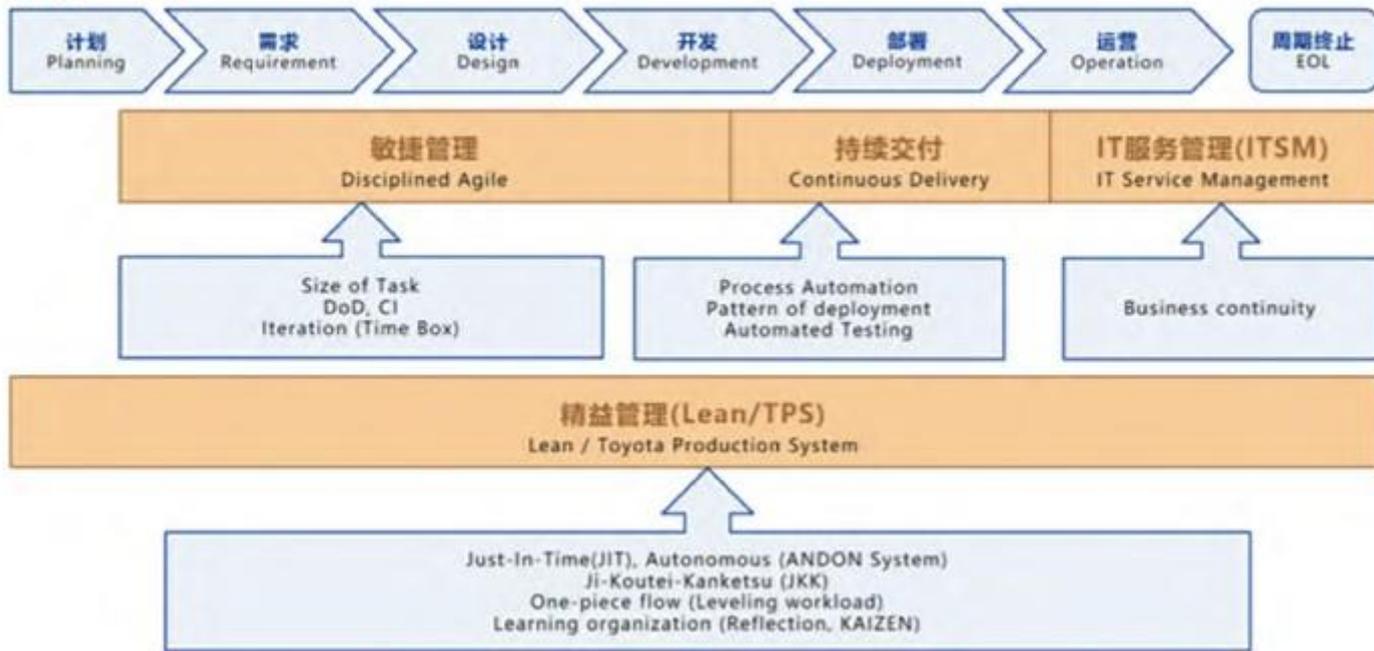
系统化思考



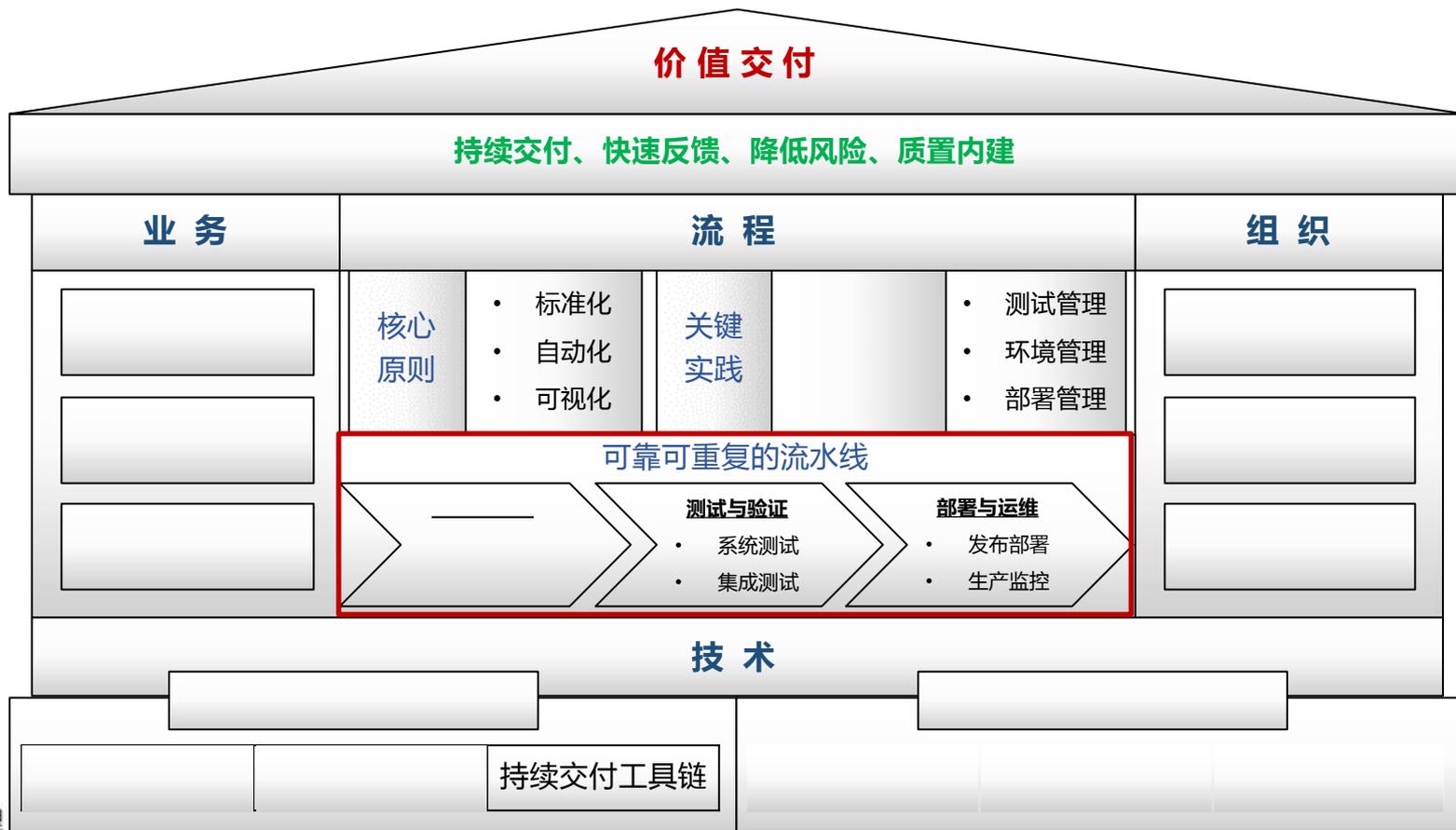
最佳实践导入

全局化视角

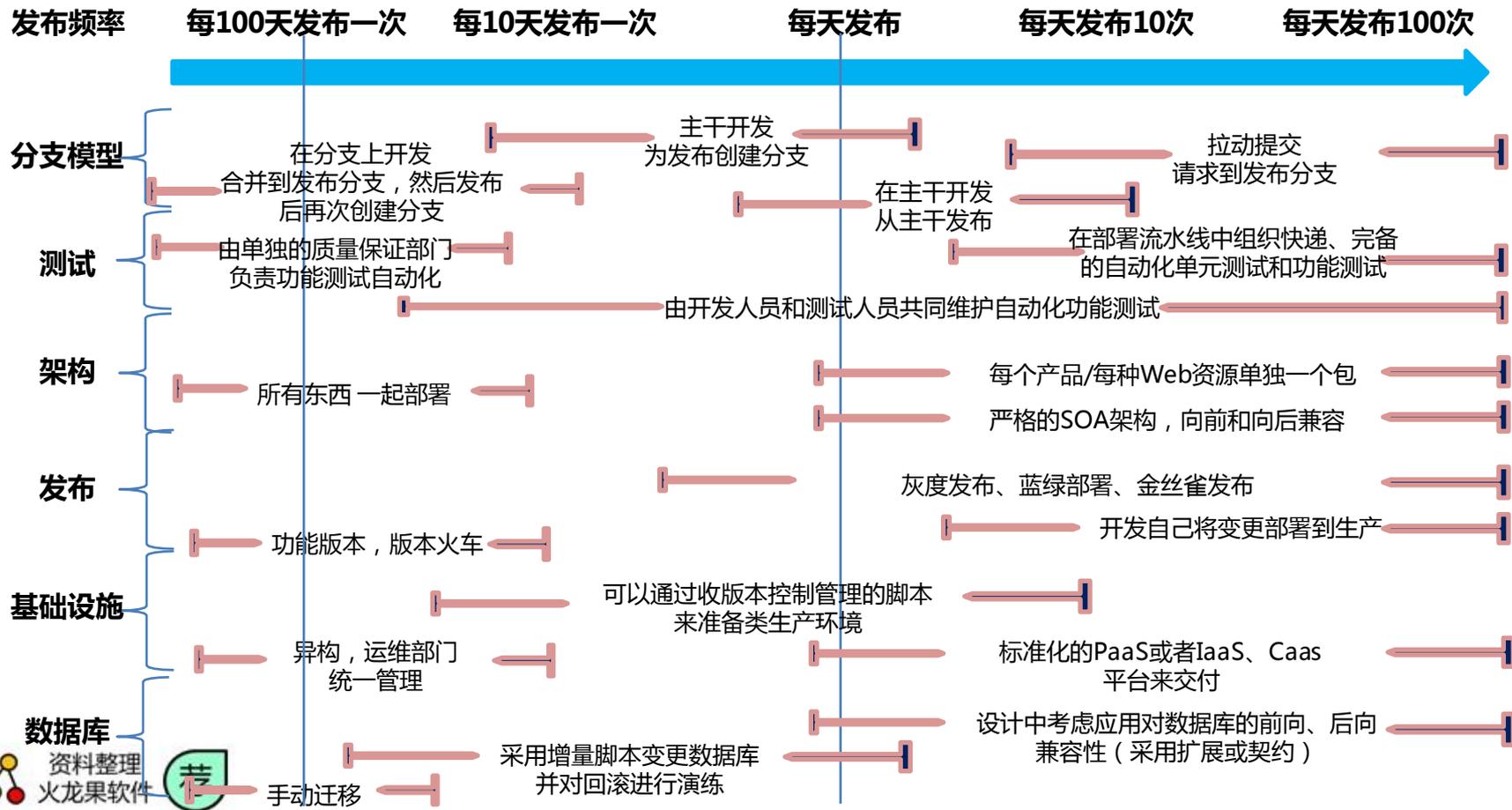
流程 (Process)



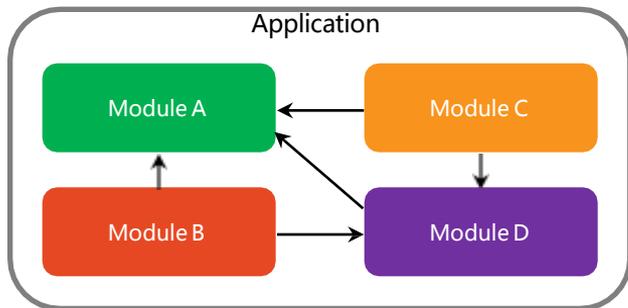
系统化思考



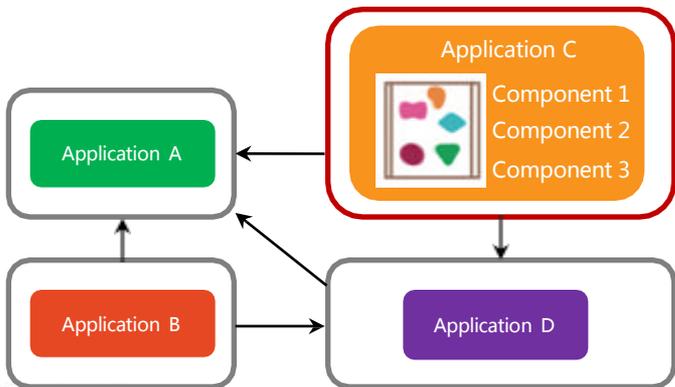
最佳实践导入



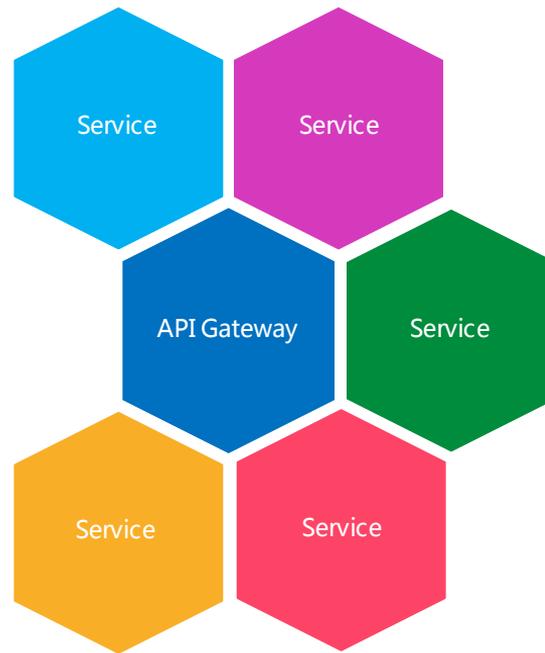
1. 架构的演进



整体式服务架构



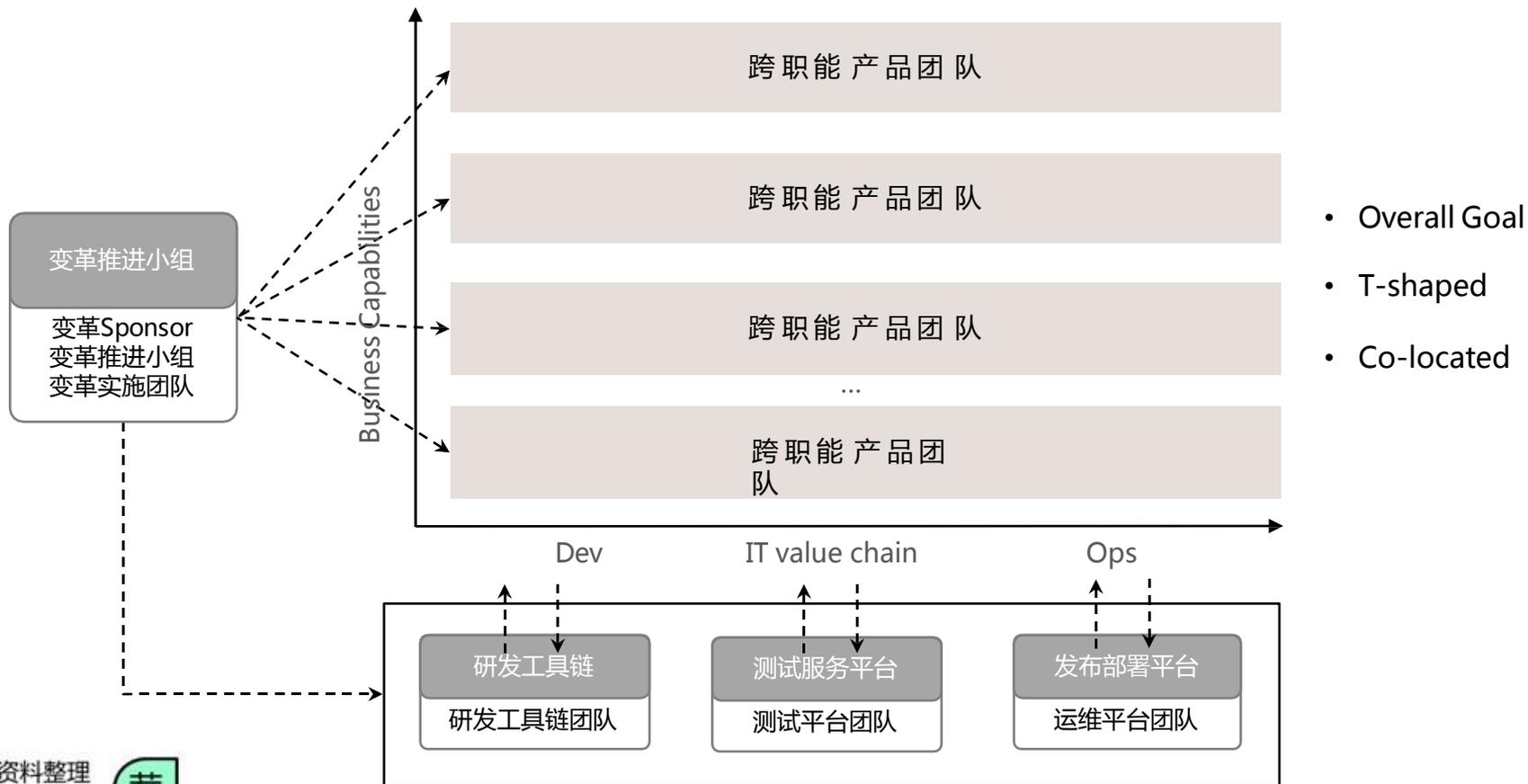
单体应用式服务架构



微服务架构

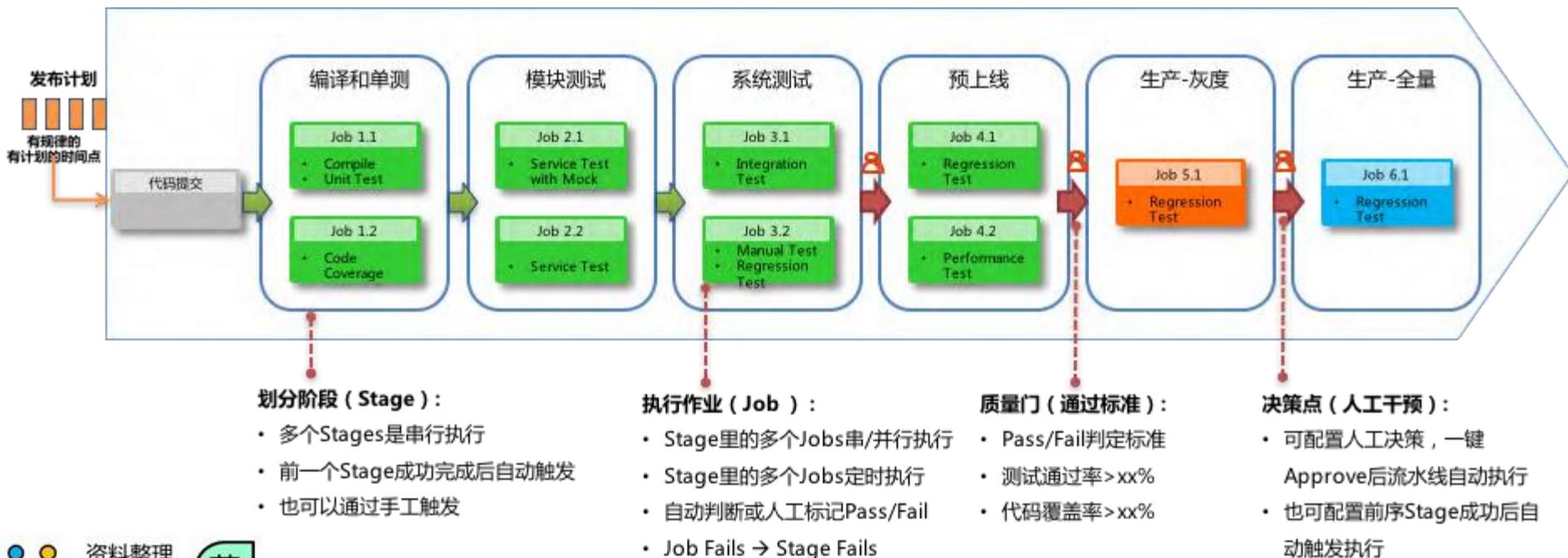


2. 组织的演进

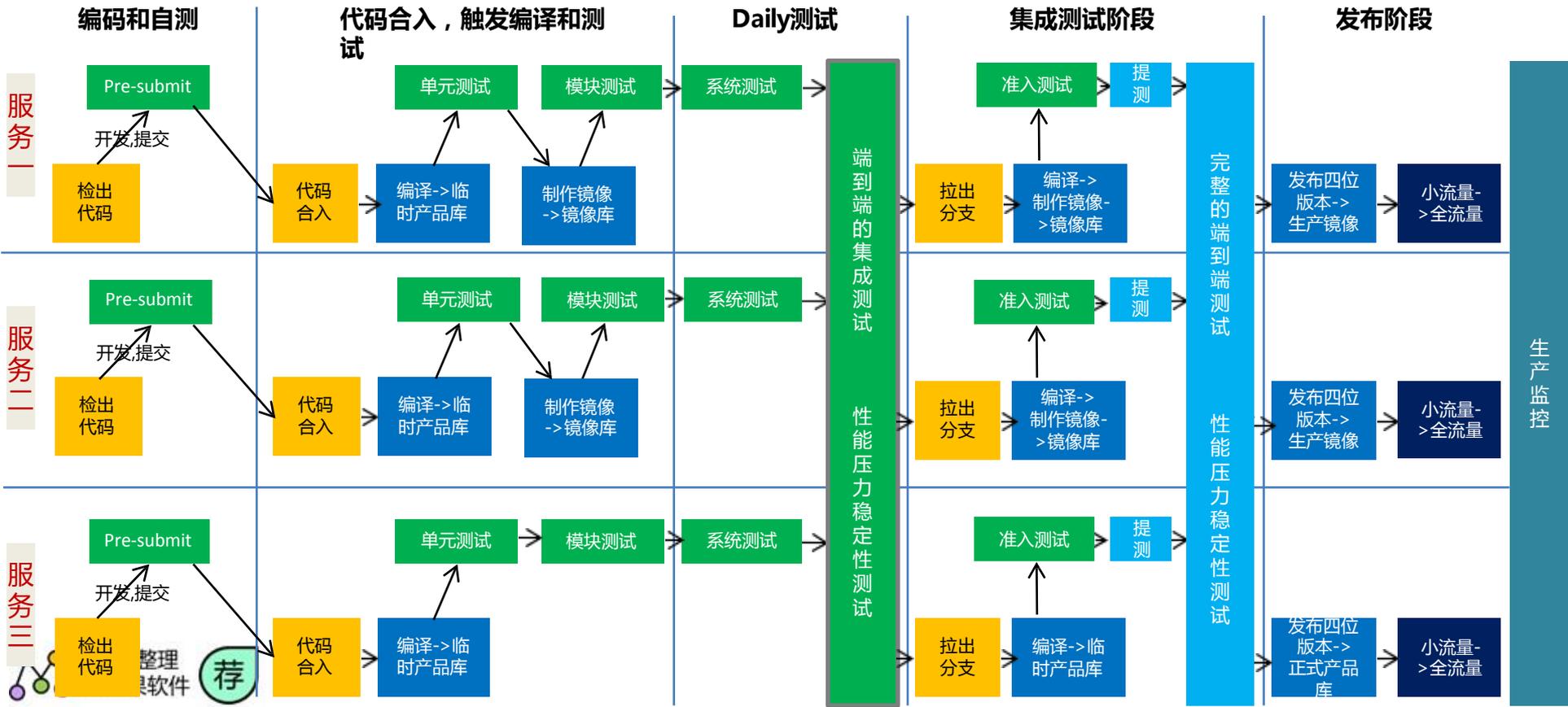


3. 建设可靠可重复的交付流水线

通过交付流水线，将全局过程标准化、自动化、可视化
关键流程和节点管控，并行开发过程中的协同和管理



3. 建设可靠可重复的交付流水线- 多服务聚合



3. 建设可靠可重复的交付流水线- 工具落地



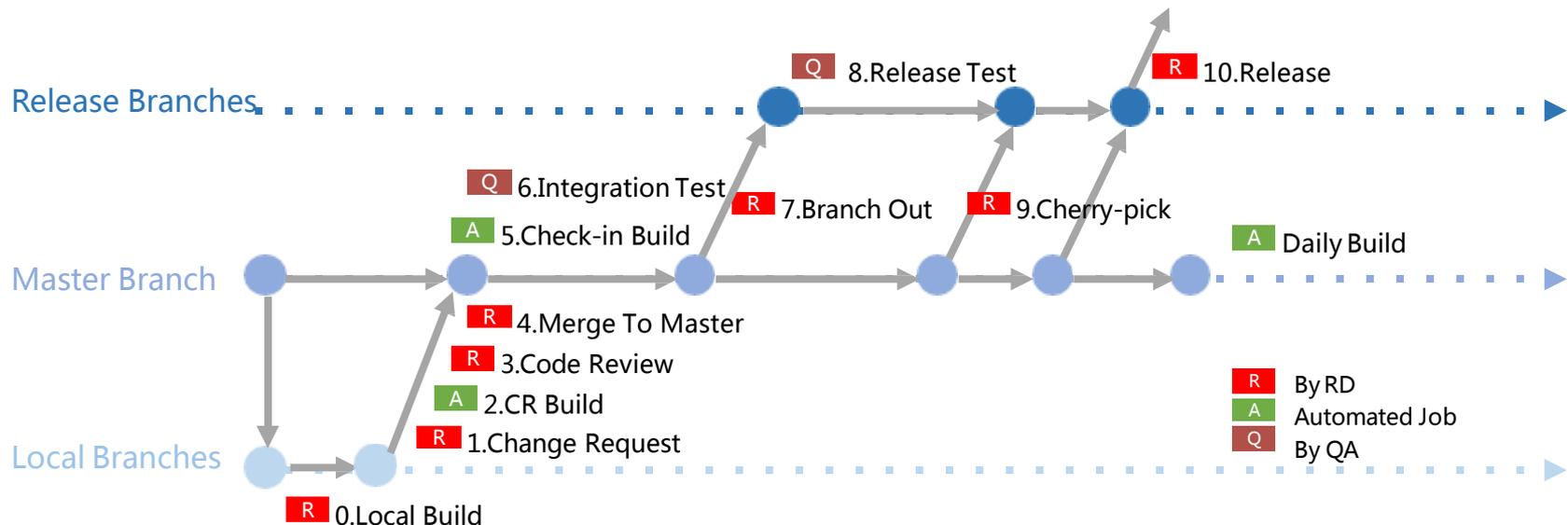
多服务聚合交付流水线



4. 配置管理及代码协作 workflow

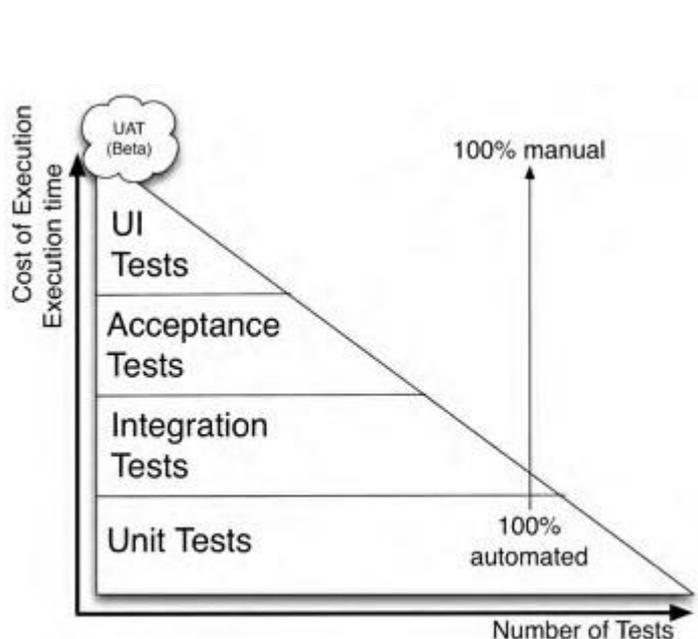
代码库从SVN迁移到Git，提升本地操作和分支管理效率

应用Git代码托管平台，简化操作并集成代码协作 workflow



5. 测试管理 - 分级质量保障

建立分级测试体系，从多个层次和多个验证角度实现质量防护网



© Allan Kelly



6. 测试管理 – 测试服务平台建设

场景数据构造

通过接口自动构造中间态测试数据，减少测试依赖，提升稳定性和效率

The interface is divided into two main sections: '基本信息' (Basic Information) and '测试账户' (Test Account). Under '基本信息', there are fields for '类别' (Category), '测试环境' (Test Environment) with options ENV1, ENV2, LT, and SANDBOX, '数据切换' (Data Switch) with options REAL and MOCK, and '操作序列' (Operation Sequence) with checkboxes for '保存四项' (Save four items), '预信' (Pre-credit), '补录' (Supplement), and '基础信息' (Basic information). Under '测试账户', there are input fields for '登录名' (Login Name), '密码' (Password) with the example '123qwe', '手机号' (Mobile Number), and '银行卡号' (Bank Card Number). A note below the mobile number field says '注意：系统会真实发送短信' (Note: The system will actually send SMS).

Mock平台

模拟同步、异步、多协议接口返回，减少测试依赖，提升独立性和效率

The interface shows configuration options for a mock service. Fields include: '接口名称' (Interface Name) set to 'Test', '接口类型' (Interface Type) set to '同步返回' (Synchronous Return), '监听地址' (Listening Address) set to 'tcp://localhost:18111', '回调地址' (Callback Address), '业务名称' (Business Name), '字符集' (Character Set) set to 'GBK', '通信协议' (Communication Protocol) set to 'TCP', 'HTTP请求方法' (HTTP Request Method) set to 'POST', '延时回送/发送读取时间' (Delay Return/Send Read Time) set to '0', '响应后是否自动关闭(仅TCP)' (Whether to automatically close after response (TCP only)) set to '是' (Yes), '解析语言类型' (Parsing Language Type) set to '定长报文' (Fixed Length Message), and '协议码解析规则(仅同步返回)' (Protocol Code Parsing Rule (TCP only)) set to '{0,6}'.

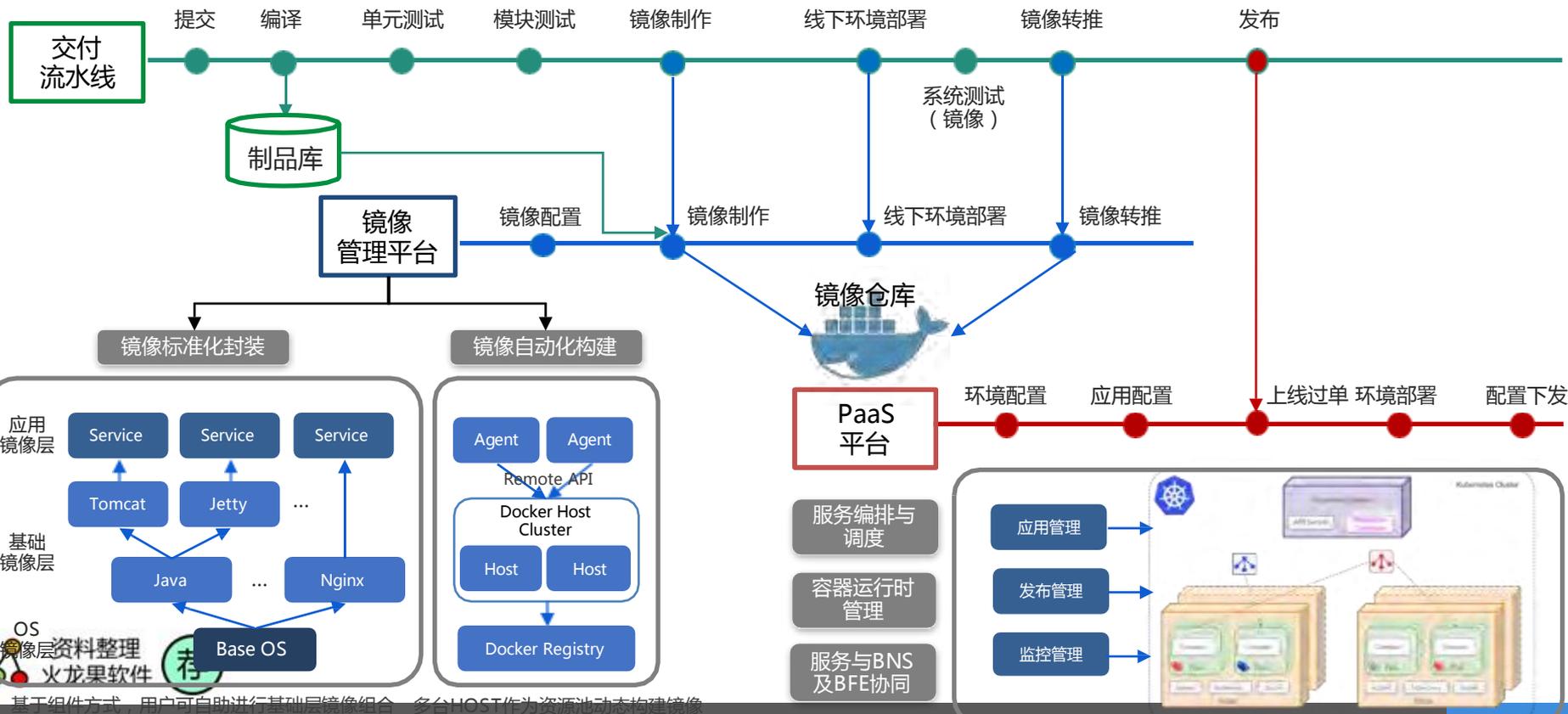
问题定位平台

根据交易ID及各类信息，自助跨服务追踪和定位问题，降低被动配合成本

The interface features a search section with '环境' (Environment) set to '线下联调环境' (Offline Joint Debug Environment), '查询ID' (Query ID) set to '2500278538', and '日志级别' (Log Level) with radio buttons for 'ALL', 'NOTICE', 'INFO', and 'ERROR'. A '查询' (Query) button is present. Below are three log viewing sections: '入口MTP日志列表' (Entrance MTP Log List), '详细日志列表' (Detailed Log List), and 'X日志列表' (X Log List). Each section has a table with columns: 'logid', 'system', 'module', 'loglevel', 'logdate', 'action', and 'mes'.

7. 应用交付归一化及基础设施建设

应用交付方式和过程归一化，并通过PaaS平台实现自助化和自动化



9. 习惯培养及度量数据驱动改进

- 重点是对原则的坚持
 - 频繁集成
 - 红灯修复
- 建立度量指标模型
 - 结果指标
 - 过程指标
- 数据驱动持续改进

1 实施情况总览、分团队总览



3 核心数据汇总，环比变化趋势

指标名称	当前值	环比变化	环比变化率	环比变化趋势	环比变化趋势
构建成功率	100%	+100%	100%	↑	↑
构建效率	100%	+100%	100%	↑	↑
构建稳定性	100%	+100%	100%	↑	↑
构建成本	100%	+100%	100%	↑	↑

2 数据筛选和下钻，各团队数据

序号	团队	构建次数	构建成功率	构建效率
1	团队A	2	100%	100%
2	团队B	3	100%	100%

4 自动分析和异常报表推送邮件



时间范围: 2016-10-31 至 2016-11-07

查询

对比 清除对比内容

产品线(权重)	总分	CI次数	团队习惯(15)			编译能力(5)			测试完备性(45)				构建稳定率(20)			构建效率(15)			操作		
			主干开发	CI频率	红灯修复时间	总分	编译时间	总分	local	trunk	覆盖率	daily	pre-online	总分	trunk成功率	异常构建率	daily成功率	总分		trunk耗时	总分
13%[-]	32.16	204	18.63%	1.16	159.00	8.15	1.06	5.00	0%	15.07%	0.29%	15.34%	0%	2.92	59.33%	40.67%	0%	2.72	25.45	13.37	发起评级
transactionsyste	45.53	45	24.44%	0.82	18.24	10.61	1.18	5.00	0%	12.5%	0%	0%	0%	1.00	94.87%	5.13%	0%	13.92	4.08	15.00	发起评级
backend-credity	35.50	65	21.54%	1.44	35.96	10.34	0.91	5.00	0%	12.5%	0.29%	0%	0%	1.00	66.67%	33.33%	0%	4.16	2.32	15.00	发起评级



改进效果

	改进之前	改进之后
发布方式	整体串行	独立并行
打包格式	程序包	镜像
迭代周期	月	周
发布频率	周	按需（可每天多次）
前置周期	数小时	数分钟

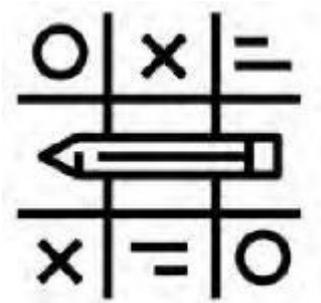
变革回顾和经验总结



全局化视角



系统化思考



最佳实践导入

痛点驱动，渐进式变革

数据驱动，持续化改进

One more thing...

基于开源工具的交付流水线实现，从0到1快速建设DevOps实施能力

