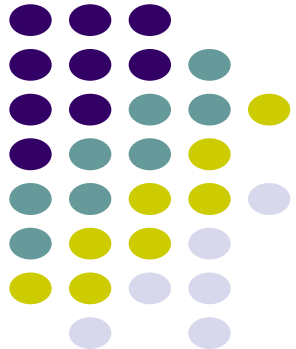
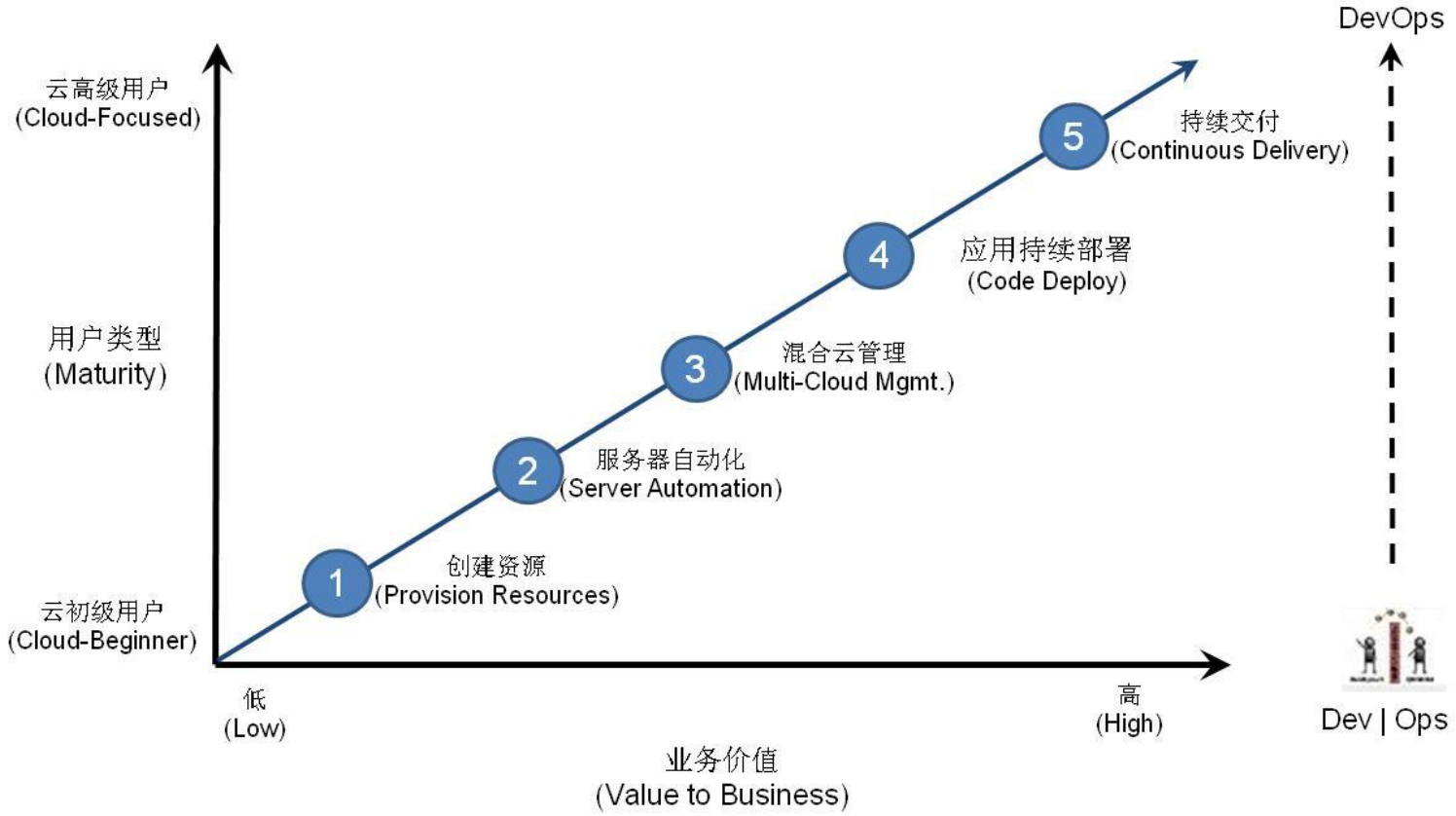
DevOps解决方案



Recommend
火龙果软件

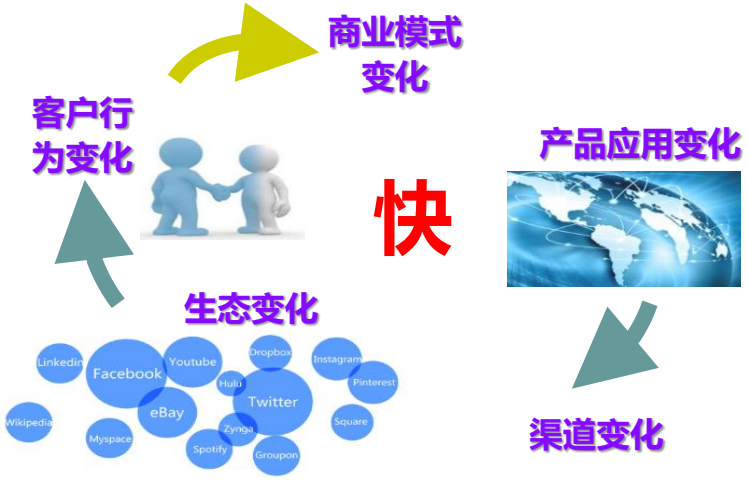
Devops是云计算发展的高级阶段



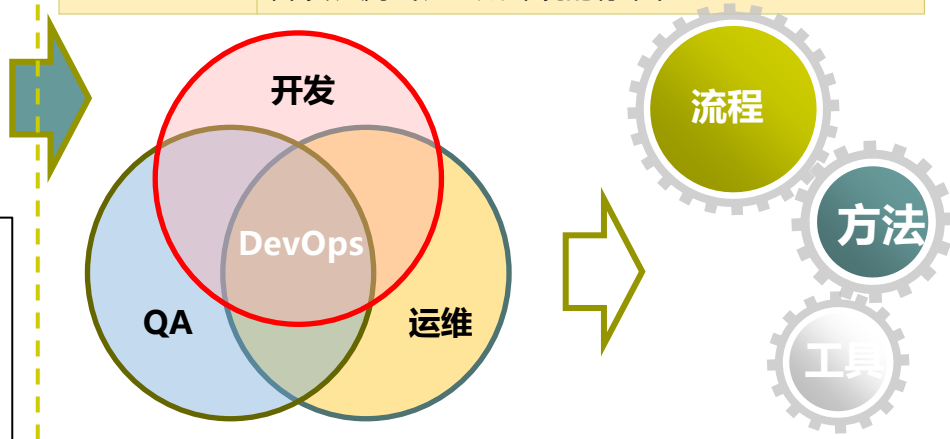
Recommend
火龙果软件

移动互联网时代更需要DevOps

移动互联网时代的特征就是快，产品的决策快、推出快、迭代快、变革快，快能抓住机遇、掌握主动。



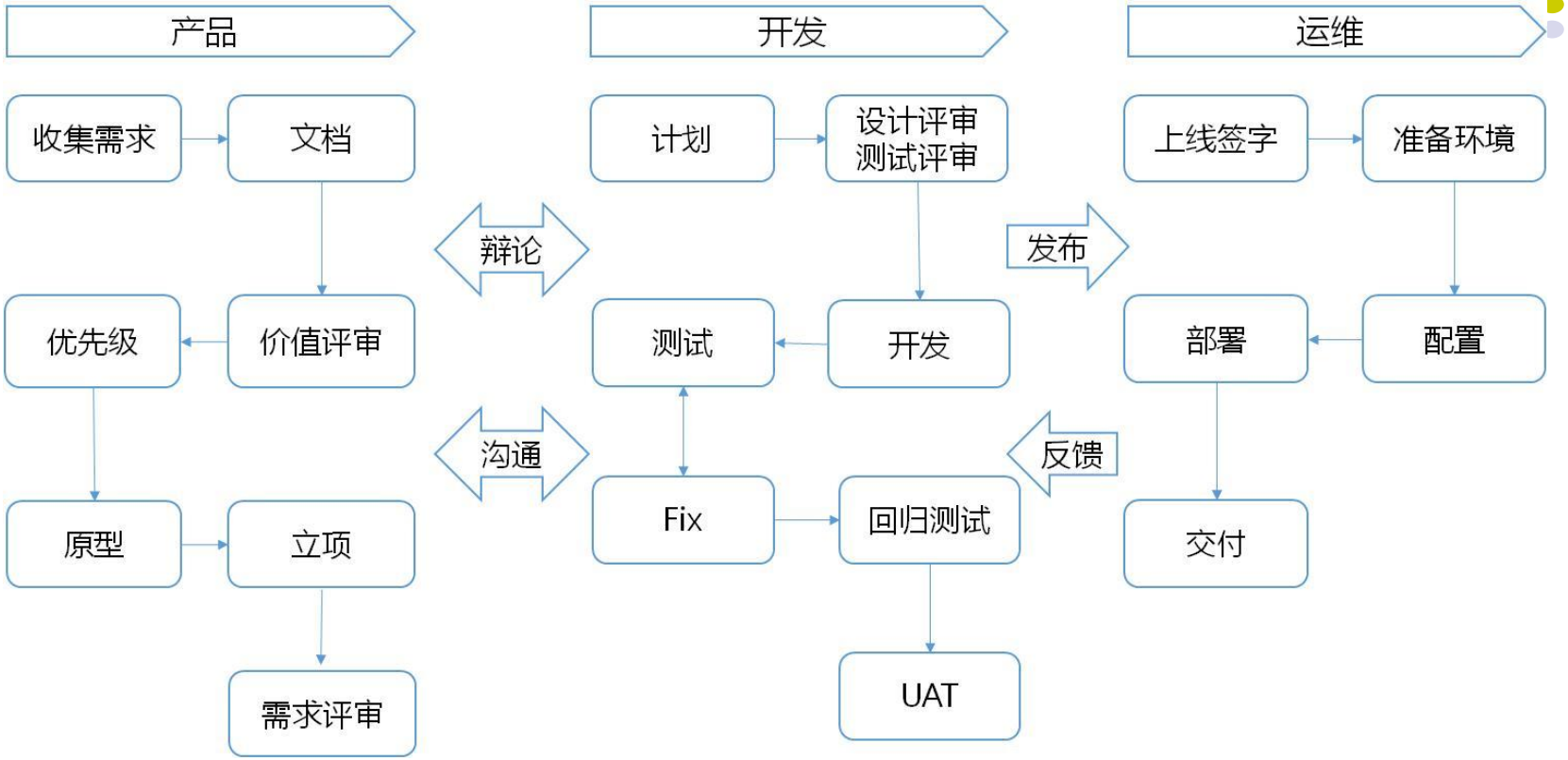
阶段	要求
开发	业务架构：多中心、能力开放
	技术架构：SOA、微服务、技术组件标准化/服务化
QA	持续集成
	自动化测试
运维	快速发布、滚动升级、灰度发布、弹性伸缩
	开发、测试、生成环境的标准化



案例分析：支撑系统有2000多万行代码，代码构建一次需要40多分钟；由于各个应用之间错综复杂，在集中提交代码模式下构建发布失败率超过20%；一次产品发布需要2周以上的时间

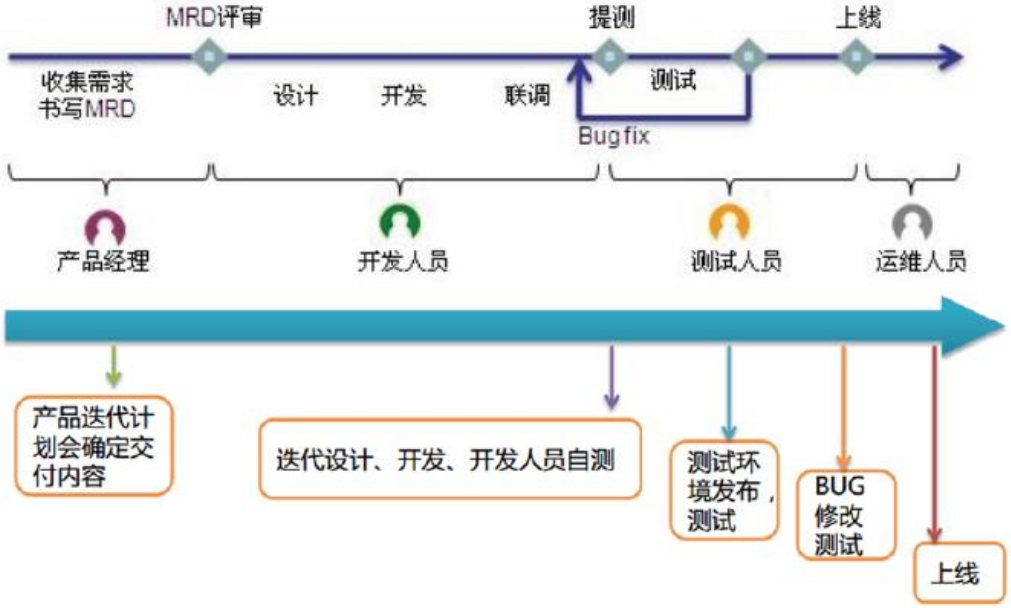
Recommend
火龙果软件

软件开发与交付相关过程



Recommend
火龙果软件

传统软件开发模式



- 上哪个版本，版本没错吧？
- XX请假了，怎么上线？手抖了····
- 怕出线上问题啊···
- 哪些库文件要改···

形成部门壁垒，降低工作效率

Recommend
火龙果软件

传统开发模式面临的问题

测试环境挑战

测试资源缺乏

- 业务增长快，项目多，测试需求多，现有服务器不能满足需求
- 资源“烟囱式”，由各部门负责分配，有的资源占用却没有充分利用，有的却没有资源分配
- 开发测试资源混用，资源变换导致部署工作量大
- 开发环境与测试环境不统一，部署负责

机房成瓶颈

- 大量新购设备造成机房空间紧张，能耗成本大量增加；

手工部署工作量大 部署周期长

- 项目差异大，无标准化，按照项目要求配置环境，部署差异大
- 多项目共用物理主机，造成应用版本冲突，灵活性不足
- 管理员少，生产、测试环境都需负责，加班也难以应对
- 虽有传统流程平台，但申请、审批、部署、使用、回收无法关联

无生命周期管理 资源循环慢

- 项目人员在需要使用资源时着急，但是在使用后，不会主动释放资源，资源浪费大
- 管理员忙于部署新建，对资源使用疏于管理，无回收机制
- 开发部门与测试以及管理部门之间协作浪费大量时间

管理不直观

- 管理员无法监控资源使用率和空闲设备
- 领导无法了解项目投入和使用以及回报的关联
- 项目人员无法了解实际使用，只能通过扩大配置需求应对业务高峰

开发	• 沟通问题 • 环境一致 • 系统发布
测试	• 沟通问题 • 环境一致 • 系统发布
运维	• 沟通问题 • 环境一致 • 系统发布

Recommend
火龙果软件

开发与运维分开面临的问题

Recommend
火龙果软件

DevOps希望达成的目标

- 开发和IT运维之间的高度协同
- 高频部署的同时，提供生产环境的可靠性、稳定性、弹性和安全性
- 价值流
  - 业务 (需求定义)
  - 客户 (价值交付)
- 起源于2009年前后
  - 一天10次部署
  - 基础设施即代码
  - 敏捷基础设施
  - 敏捷系统管理
  - 持续集成和发布
  - 平台即服务

Recommend
火龙果软件

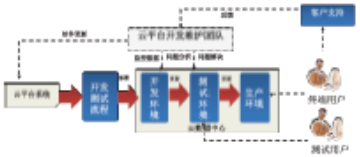
天云DevOps解决方案总体架构

项目组1 | 项目组2 | 项目组3 | 项目组4

SkyForm云管理平台

开发测试云

行业应用PaaS



云运维管理

企业云服务（IAAS）

vmware | openstack CLOUD SOFTWARE | PowerVM | 裸机

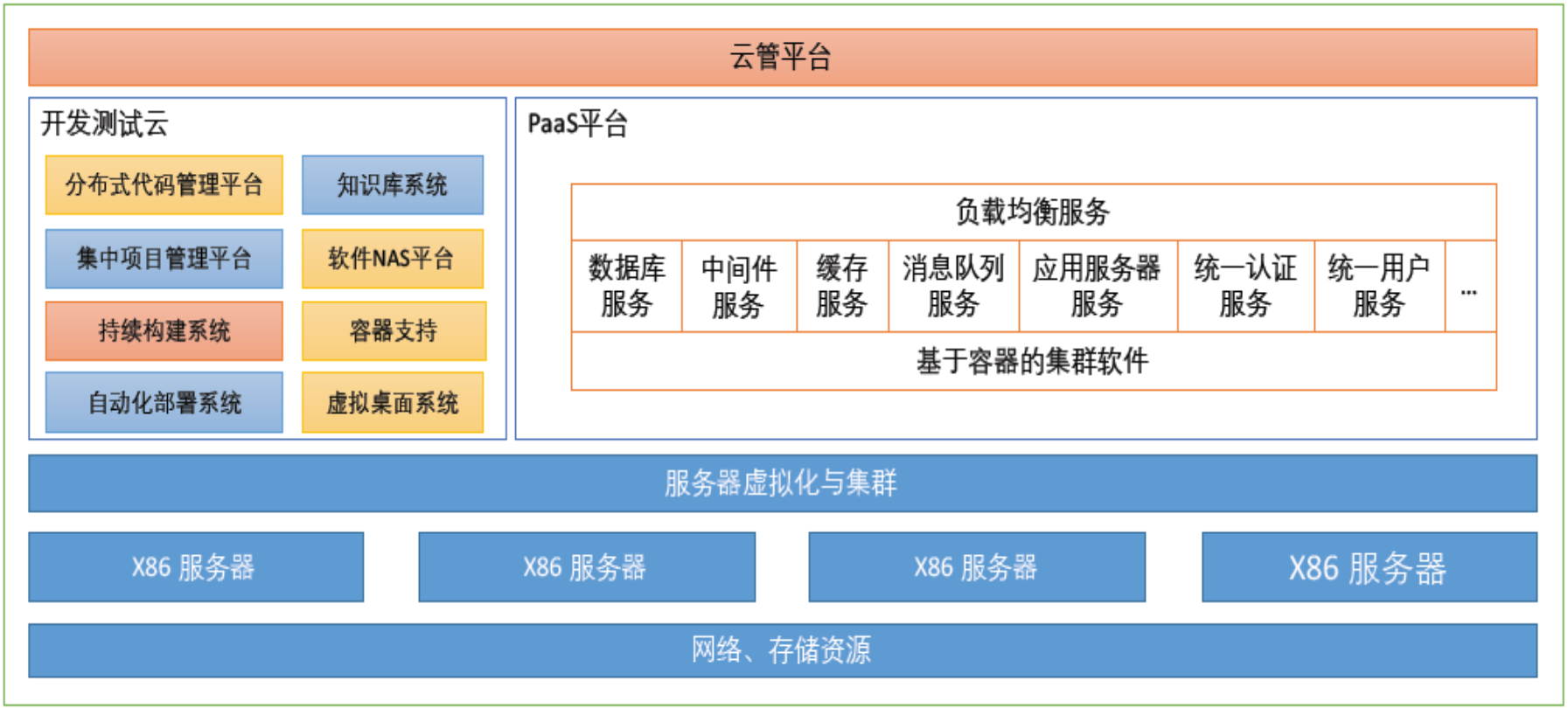
- 多租户模式
- 按项目进行管理
- 统一云管门户系统
- 自服务模式
- 在线资源申请

- 基于云端的开发测试环境
- 分布式开发与代码管理
- 集中构建与版本管理
- 自动部署到PaaS系统
- 分阶段的持续集成
- PaaS自动伸缩
- 可视化运维

- 异构化虚拟平台支持
- 资源集中监控
- 资源集中管理
- 可视化资源使用
- 自动资源回收
- 裸机支持

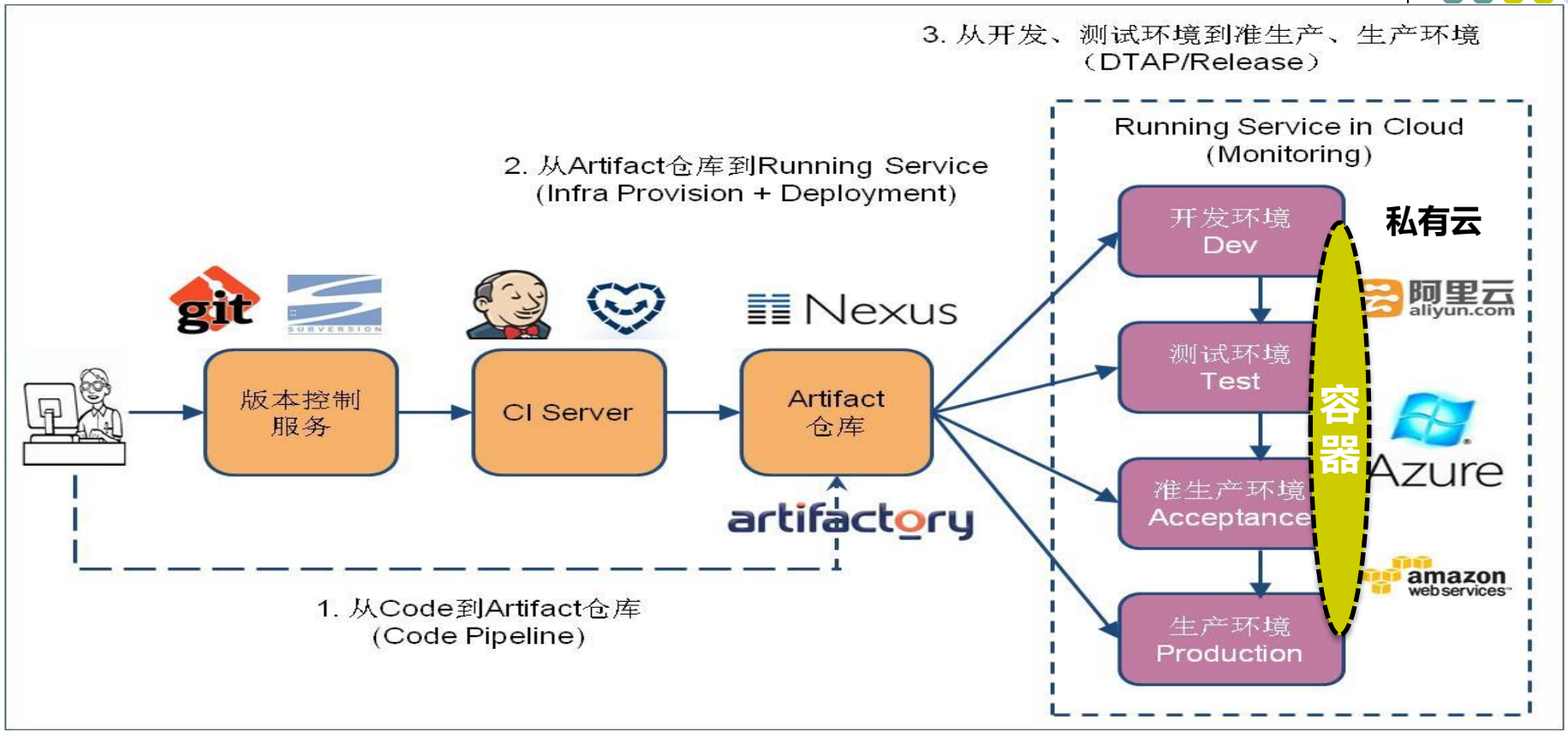
Recommend
火龙果软件

天云DevOps解决方案物理架构



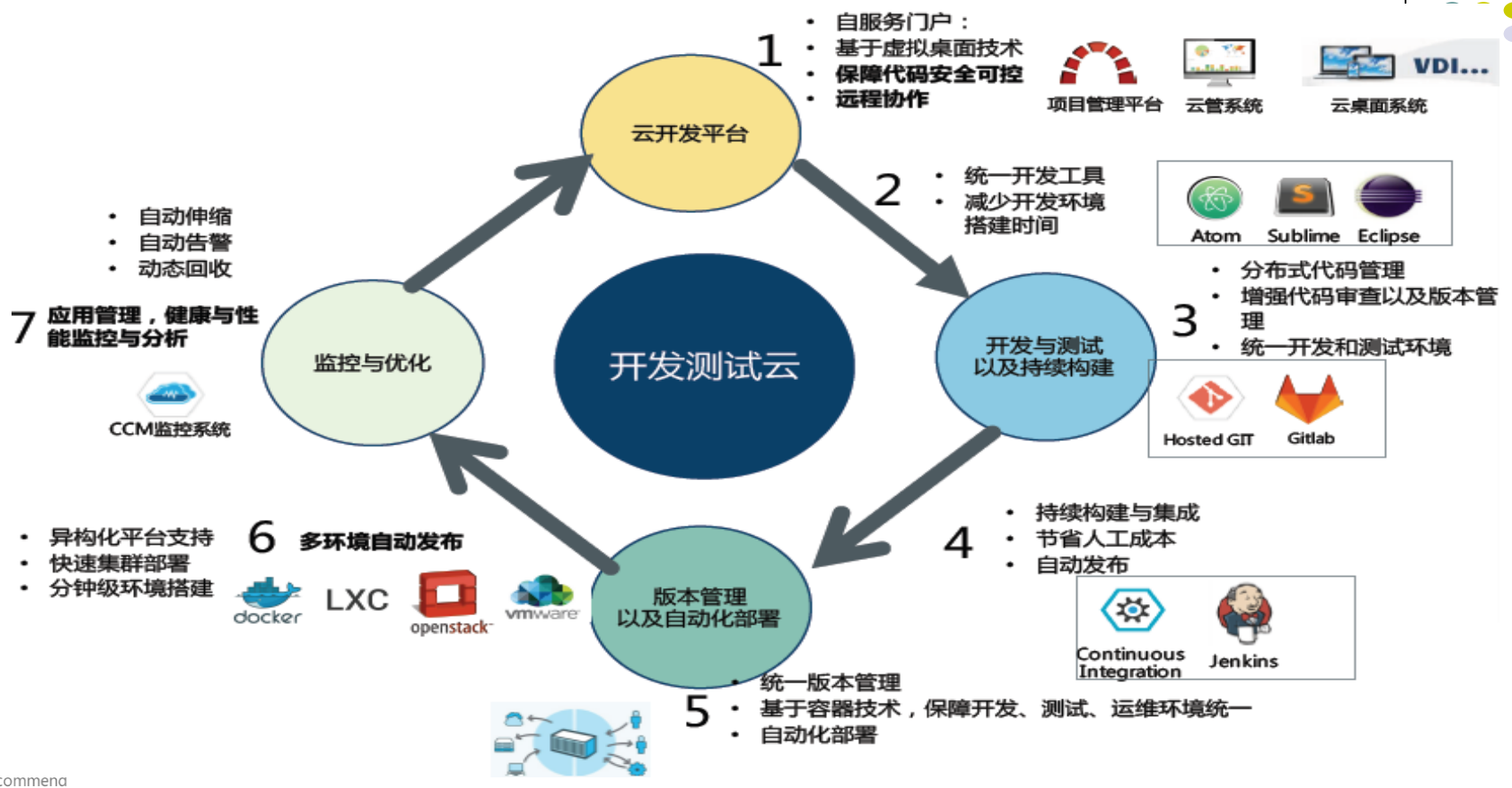
Recommend
火龙果软件

DevOps与云平台的关系



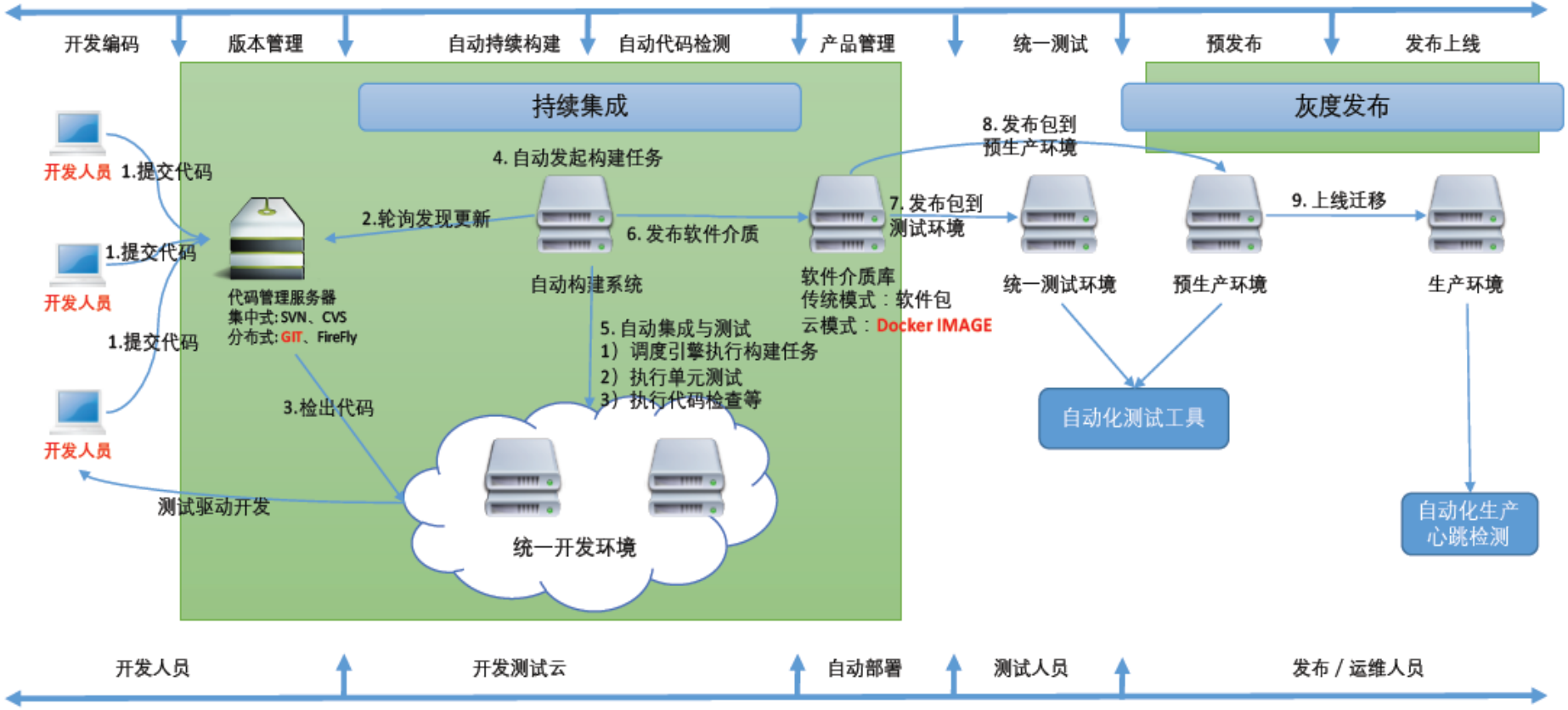
Recommend
火龙果软件

天云DevOps解决方案全业务流程



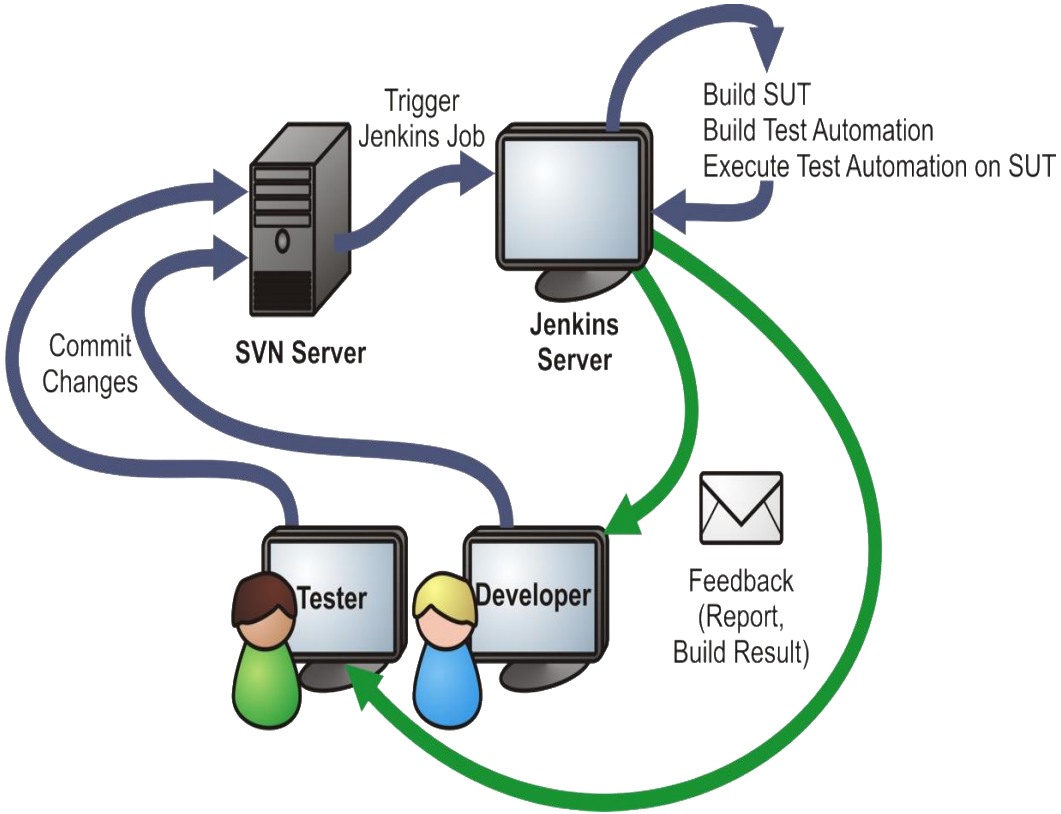
Recommena
火龙果软件

CI/CD全过程管理



Recommend
火龙果软件

为什么需要持续集成

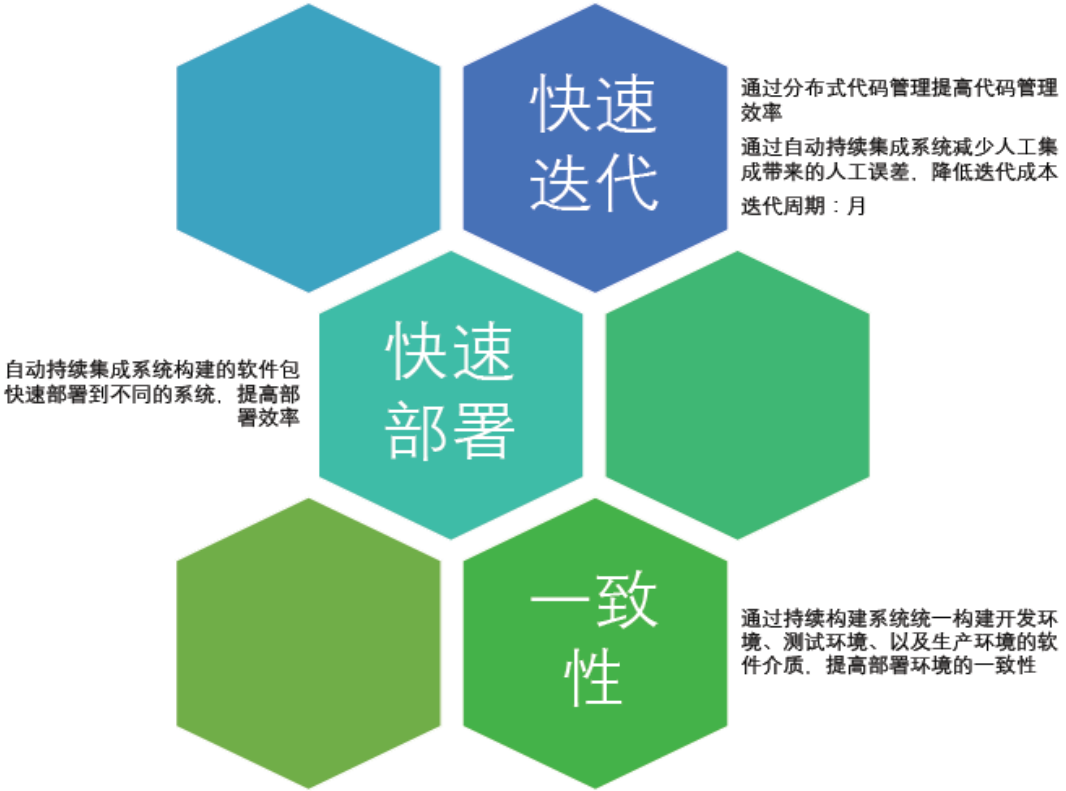


- 快速反馈
- 减少项目风险
- 每个人都是项目的Owner
- 持续开发
- 将一些重复的事情交给机器去做

- 单一代码仓库
- 经常提交 (Commit Often)
- 让你的Build可以自动化测试
- 自动构建
- 快速构建

Recommend
火龙果软件

持续集成（CI/CD）带来的优势

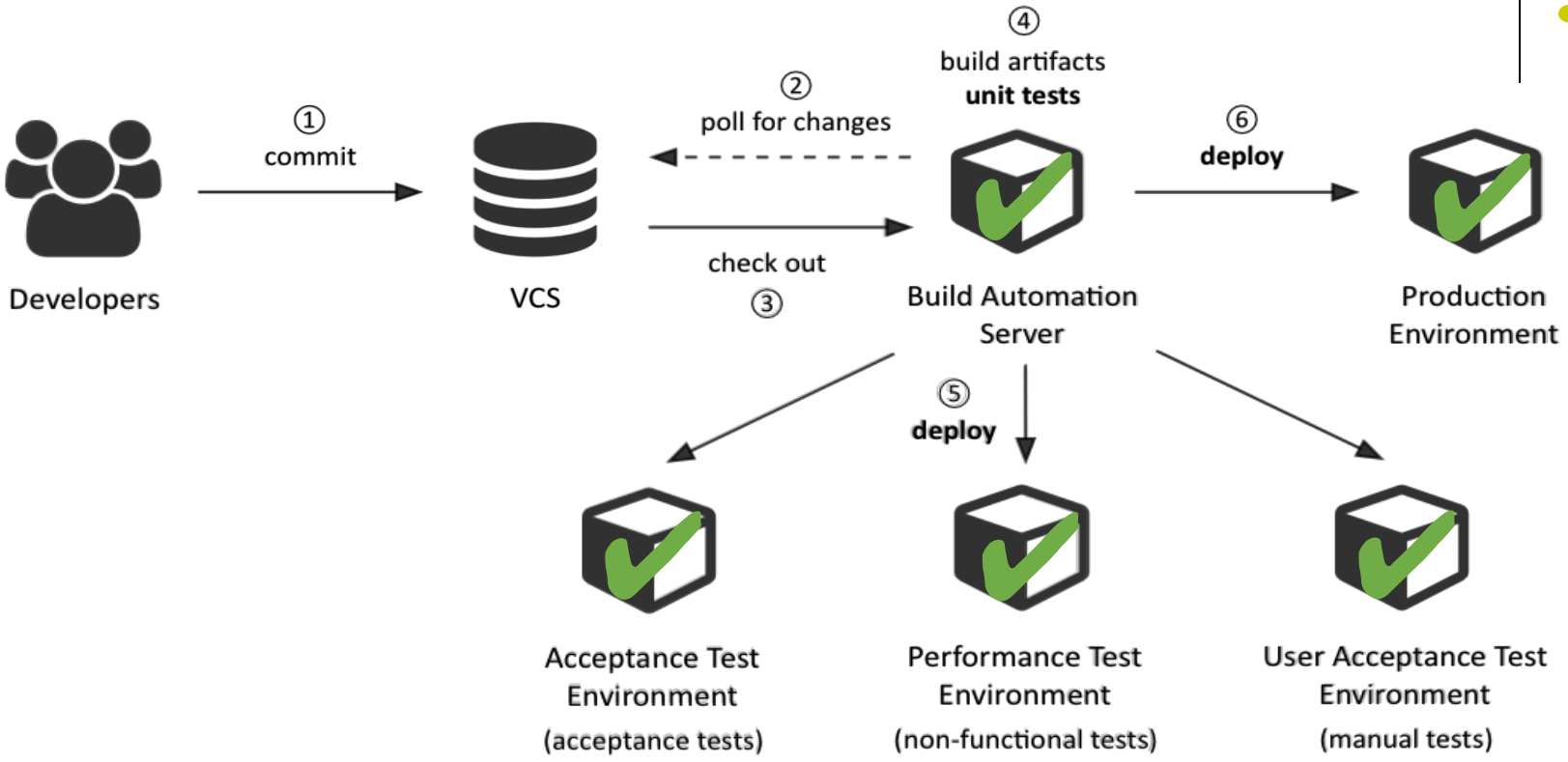


典型适应场景：

1. 企业OA、ERP以及CRM等系统
2. 传统业务场景

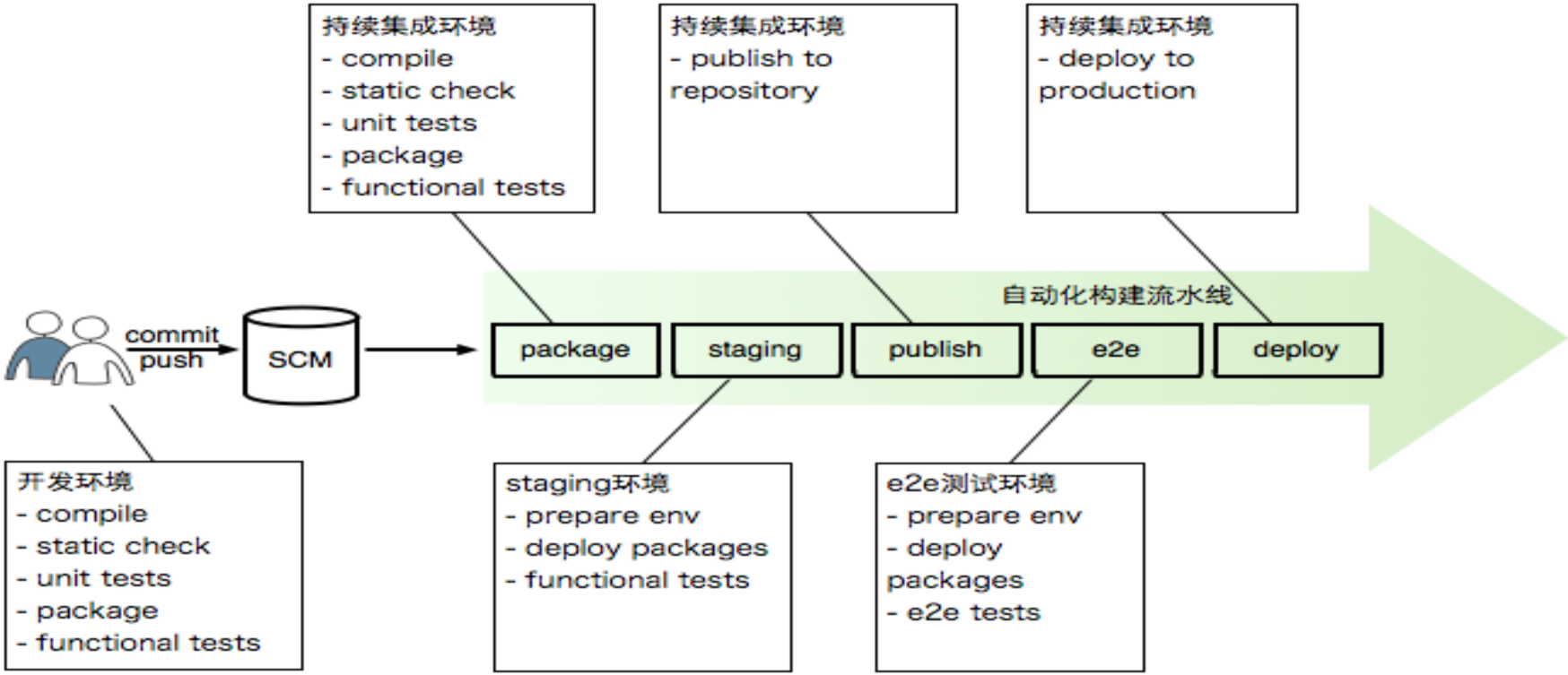
Recommend
火龙果软件

持续部署



Recommend
火龙果软件

构建流水线



Recommend
火龙果软件

开发运行环境定义

运维人员

开发人员

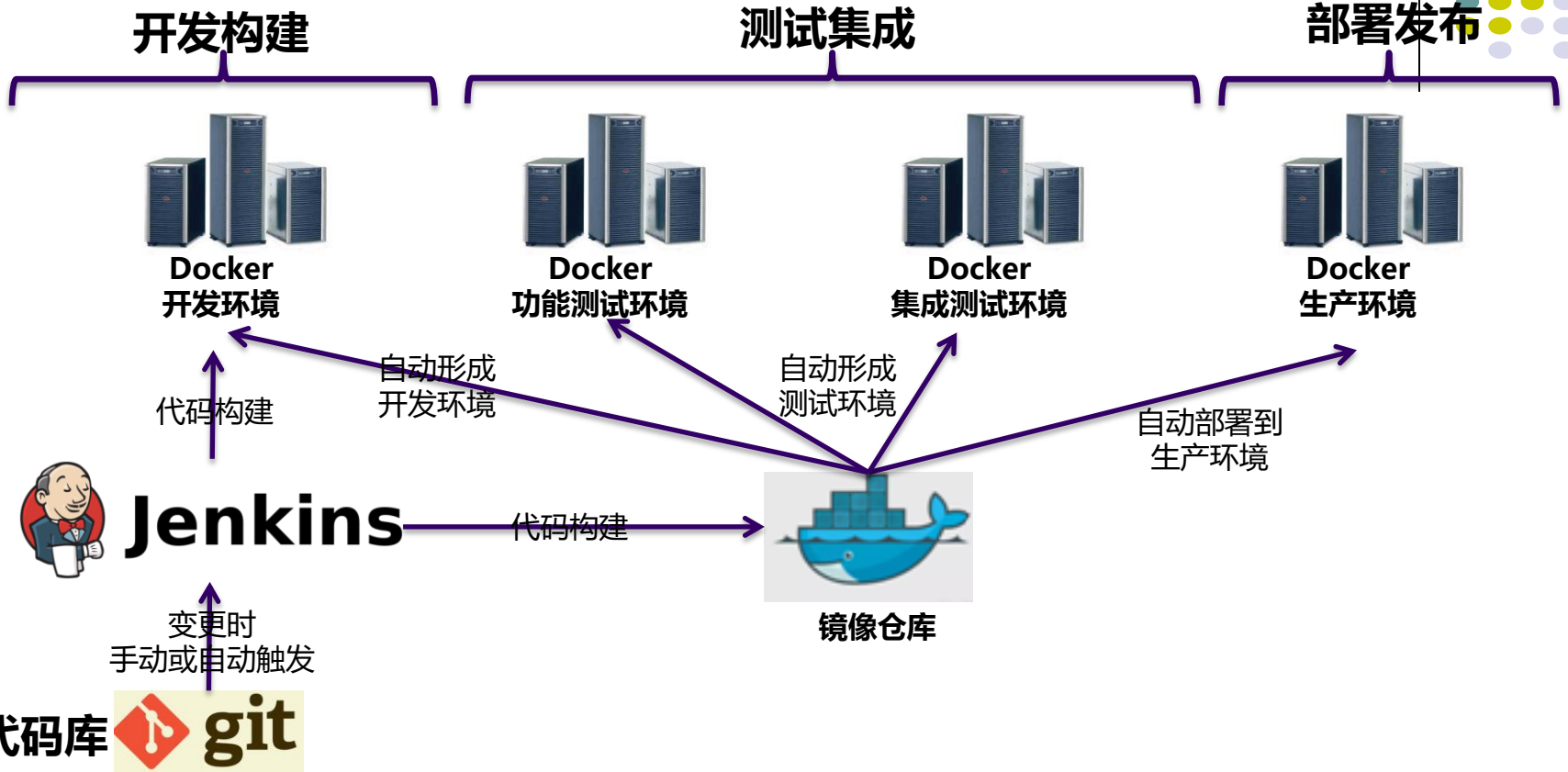
CI服务器

版本库

本地或公用Maven库

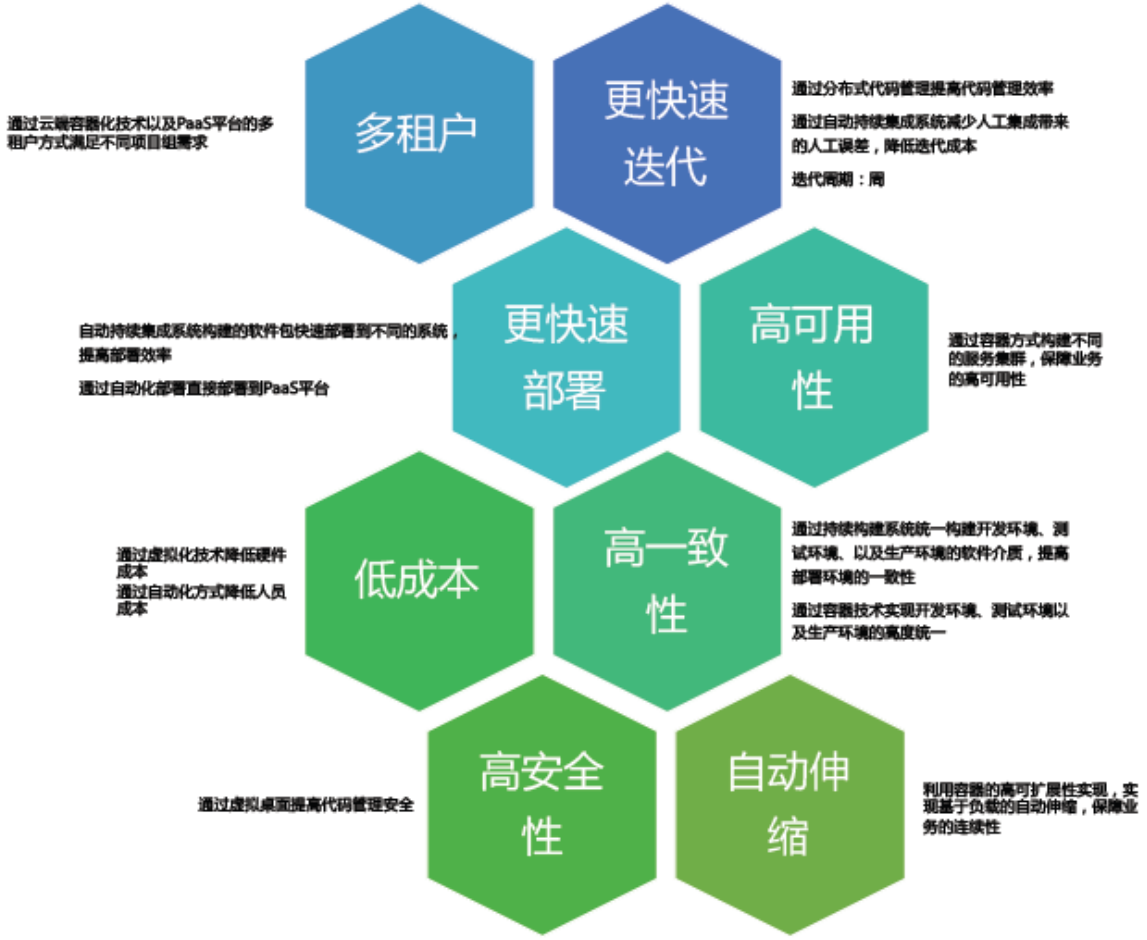
Recommend
火龙果软件

基于容器构筑具备完全一致性的开发测试环境



Recommend
火龙果软件

容器化带来的优势

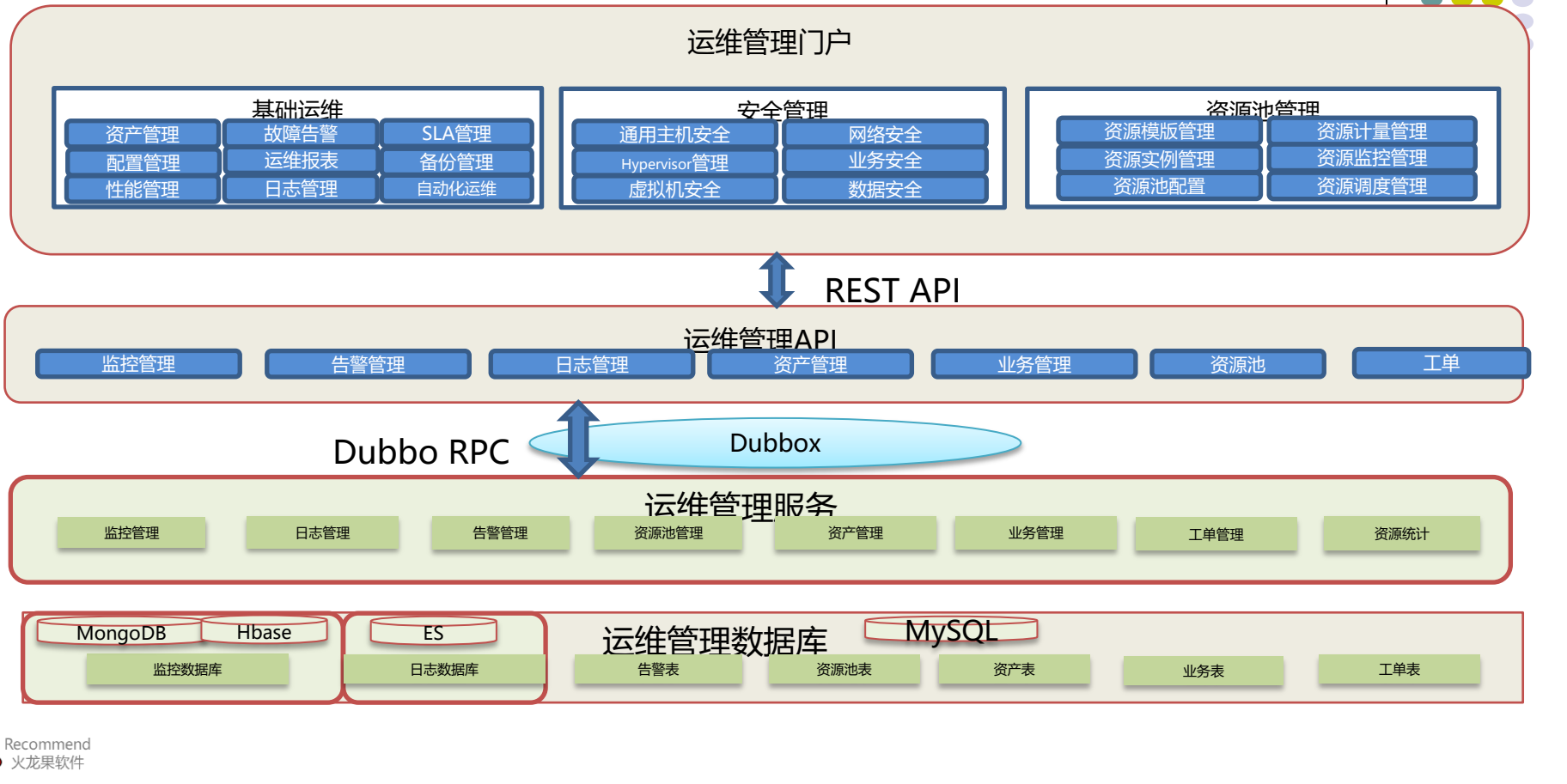


典型适应场景：

1. 电商
2. 移动客户端
3. 其他类互联网应用

Recommen
火龙果软件

自动化运维框架



Recommend
火龙果软件

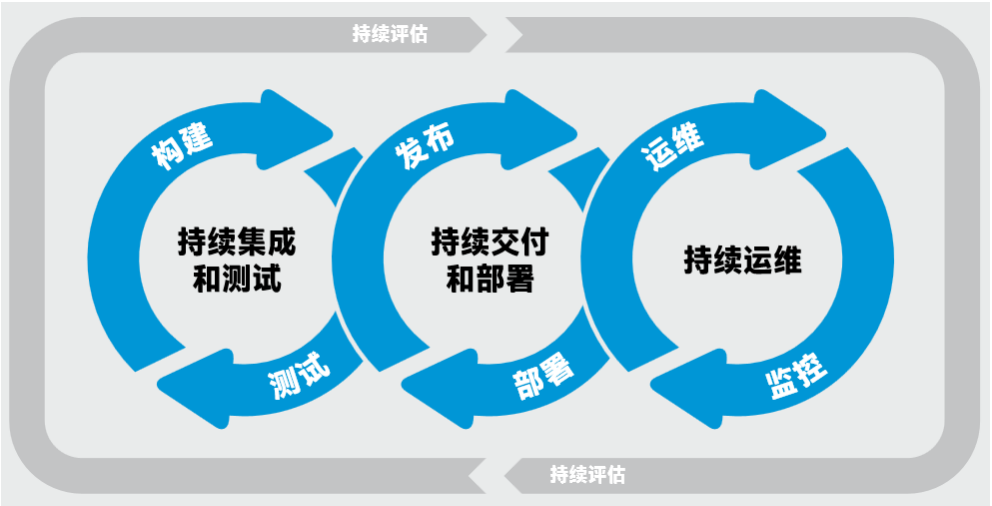
DevOps成熟度模型

级别	环境和部署	流程管理	配置管理	构建和持续集成	开发测试质量	运维监控和度量
原始阶段	纯手工过程部署软件系统、配置环境，不能频繁部署、过程不可靠	流程流转靠人的经验管理来完成	部分文件使用版本控制，签入代码不频繁	纯手工方式构建，缺少管理和报告	手工测试	人工监控为主，自动监控为辅
基础阶段	自动化部署到某些环境，创建一套新环境比较简单快捷，过程相对可靠	需求、开发、运维之间交换有规范、不存在信息遗漏	所有构建软件系统所需的东西都纳入版本控制，包括源代码、配置、构建脚本、部署构建、数据脚本等	定时或人工触发自动化构建	部分自动化测试	监控作为常态化运维保障工作，故障发生时通过邮件、短信等方式及时通知
可持续阶段	全自动化、提供自助服务式的部署方式，每个环境的部署都共用同一个过程	各个环境流程管理是闭环，并在平台中管理，环节之间流转具备自动化能力	库和依赖也被管理起来	每次变更提交都会触发自动化构建和测试	自动化单元、接口、UAT测试	有统一的监控平台，并能对监控数据进行分析和异常定位
成熟阶段	完全自动化，实现端到端的交付	从平台上可以看到整个过程的信息	频繁签入代码，开发人员每天至少签入一次代码到主分支	收集构建度量及测试结果数据，并进行可视化分析和展示	质量度量和趋势跟踪，自动化测试可以覆盖的都采用自动化测试	所有系统都纳入统一监控平台，开发运维可通过监控平台提升系统可靠性管理
优化阶段	所有资源和环境都会高效的管理起来，自动化地配置管理，提供统一的PaaS服务平台提供虚拟化、容器化的资源	团队从需求、开发、测试、部署、运维，所有工具和管理都在一个平台，可管理、可流转、可追溯	支持高效的协作、快速开发、可审计的变更管理流程	持续改进流程，优化反馈机制，提高透明度	几乎没有生产回退、缺陷能及时发现和修复	系统监控和业务反馈及时，快速进行优化提升

Recommend
火龙果软件

分阶段实施DevOps

- 第一阶段：核心实践和试点
- 第二阶段：持续集成和测试
- 第三阶段：持续部署和交付
- 第四阶段：持续运维
- 第五阶段：持续评估改进



Recommend
火龙果软件

知识回顾
Knowledge Review

祝您成功！