CHINA DEVOPS DAYS

CHINA DEVOPS DAYS | IT大咖说 知识共享平台

基于 DevOps 「三位一体」的交付实践


Exp:
10 Years
ToB
Clients:
Dongfeng Moto
Air China
Danske Bank
Tokyo Gas
Daimler
Certificate:
RedHat
Oracle
VMware
IBM
EXIN
张 扬
Career:
Ops Engineer
Solution Designer
Sales Engineer
DevOps Consultant
Company:
Dawnpro
IBM
DaoCloud
ThoughtWorks

CHINA DEVOPS DAYS

目录

引 子

2004年的夏天

洛杉矶湖人

洛杉矶 斯台普斯球馆
菲尔·杰克逊（11次总冠军）
沙奎尔·奥尼尔（鲨鱼）
卡尔·马龙（邮差）
里克·福克斯
科比·布莱恩特（黑曼巴）
加里·佩顿（手套）

底特律活塞

底特律 奥本山宫殿球场
拉里·布朗（1次总冠军）
本·华莱士
拉希德·华莱士
泰肖恩·普林斯
理查德·汉密尔顿
昌西·比卢普斯

CHINA DEVOPS DAYS

三位一体

- 「组织」

目标和团队

- 「方法」

框架和流程

DevOps 是一个能够让团队产生化学反应的「体系」

- 「技术」

架构和工具

组 织

目标 和 团队

CHINA DEVOPS DAYS

背景

痛 点

- 过多的信息系统
- 低效的系统性能
- 共享账号安全隐患
- 高额的成本投入
- 重复的数据录入
- 受限的流程支持
- 数据价值低下
- 业务运营不透明

移动新零售

愿 景

- 为经销商提供便利
- 移动优先解决方案
- 基于敏捷方法打造
- 销售售后端到端支持
- 最好的用户体验

目标

- 产品目标

  聚焦业务价值

- 迭代目标

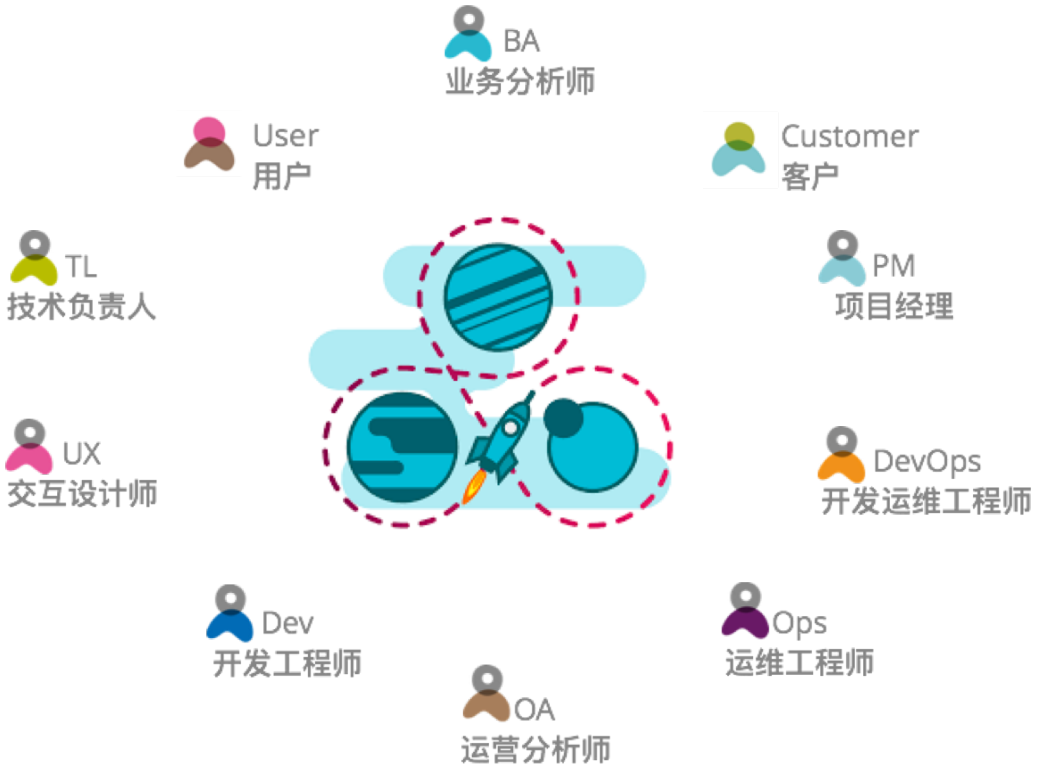
  聚焦需求实现

- 改进目标

  聚焦反馈改善

价值流闭环

CHINA DEVOPS DAYS 全功能团队



- **角色边界**

  最小职责和最大职责

- **扁平化**

  领导者来取代管理者

- **360度反馈**

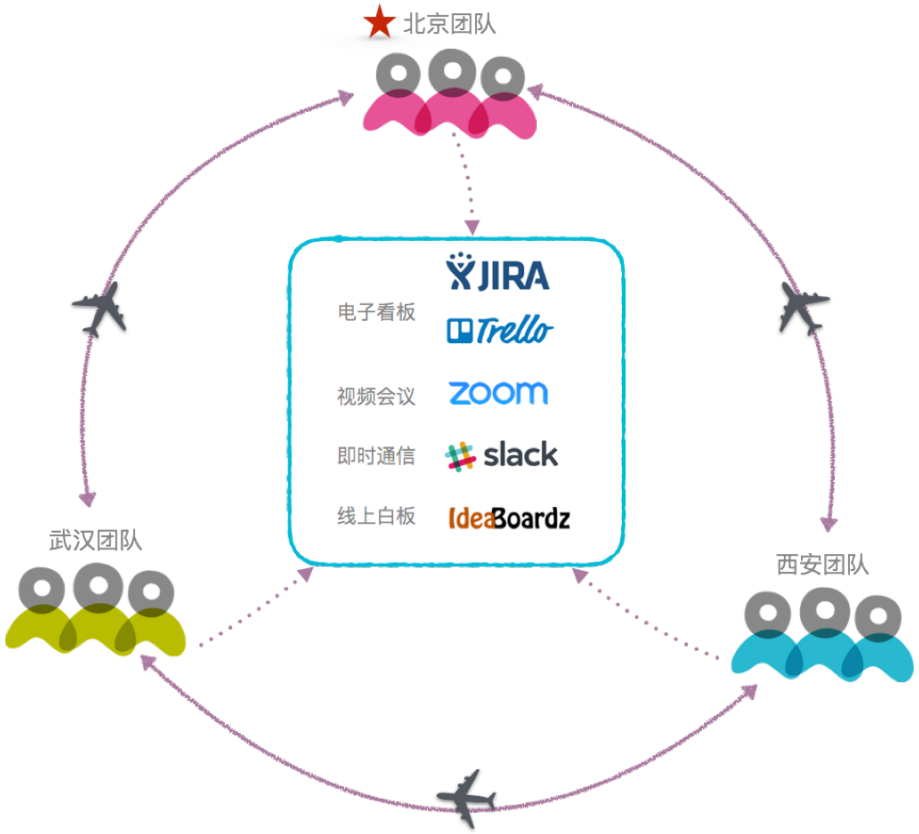
  成员由团队共同选择

- **安全氛围**

  鼓励尝试 免责文化

打破部门间的竖井，打造「面向业务价值」的全功能团队

分布式组织



- 共计 80+ 人
- 北京、武汉和西安三地分布
- 各地域再细分全功能小团队
- 日常工作独立或线上
- 重要事情到客户现场

工作流可视化是分布式团队的核心

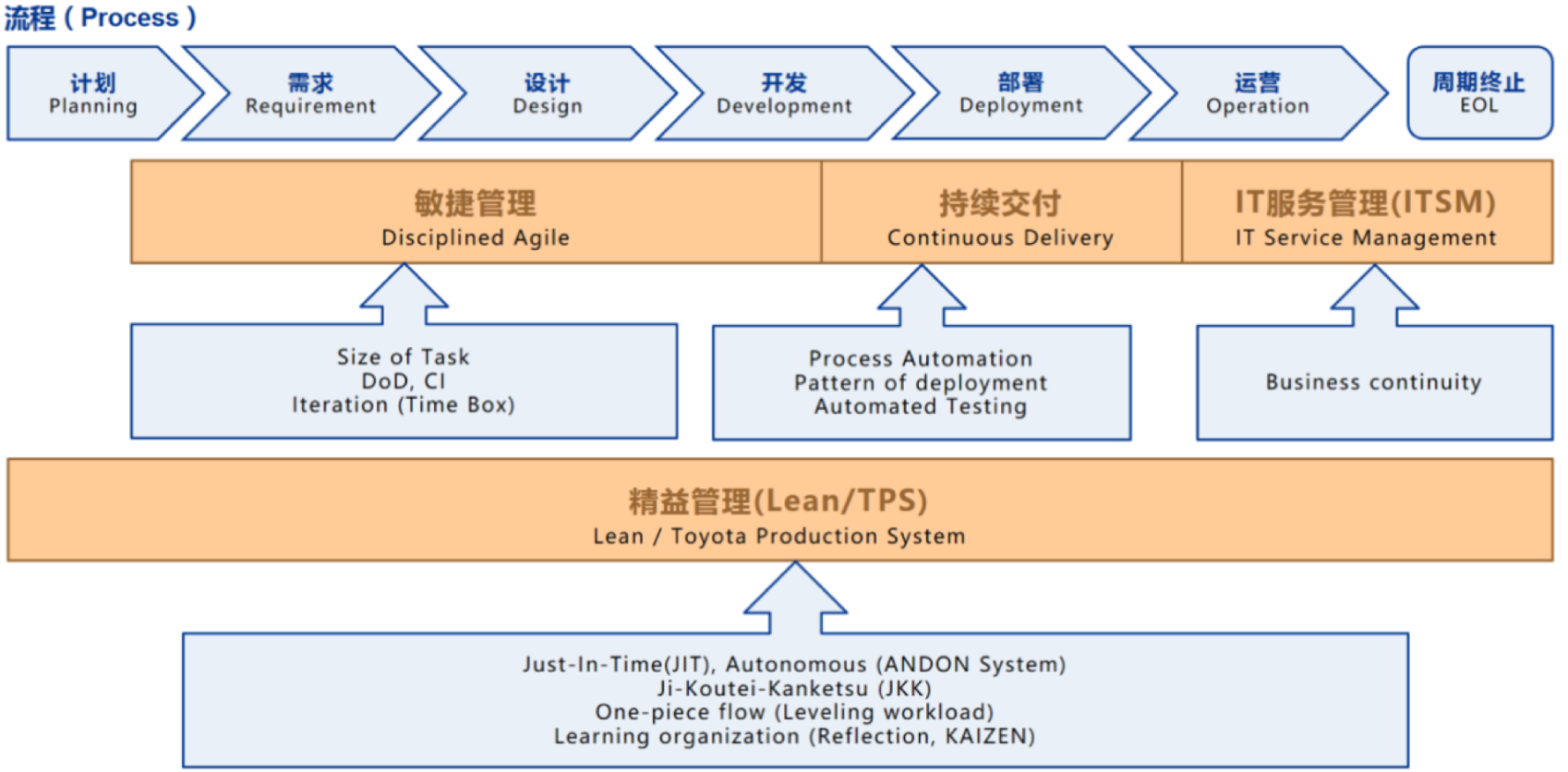
CHINA DEVOPS DAYS

CHINA DEVOPS DAYS | IT大咖说 知识共享平台

方 法

框架 和 流程

DevOps 知识体系



《企业 DevOps 成功之路》
— EXIN

「敏捷管理」

Scrum 框架

迭代和发布的频率需要基于「业务需求」和「团队容量」

CHINA DEVOPS DAYS 四大会议 Tips

Sprint Plan Meeting

迭代启动会议

目的:
- 确定目标和优先级

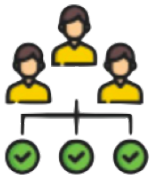
Tips:
- 迭代明确目标
- 团队达成一致

Standup Meeting

每日站会

目的:
- 团队内同步信息

Tips:
- 注意时间控制
- 不深入细节讨论



Review/Show Case

评审会议

目的:
- 及时得到反馈

Tips:
- 演示材料准备充分
- 通知团队所有成员

Retrospective

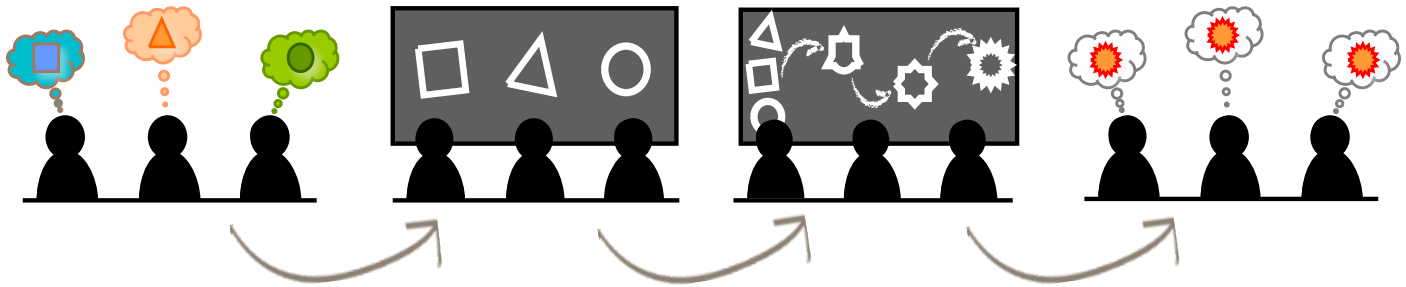
回顾会议

目的:
- 团队持续改进

Tips:
- 认可团队成员的工作
- 改进一定是可执行的

避免形式感　加强仪式感

完成的定义 (DoD)

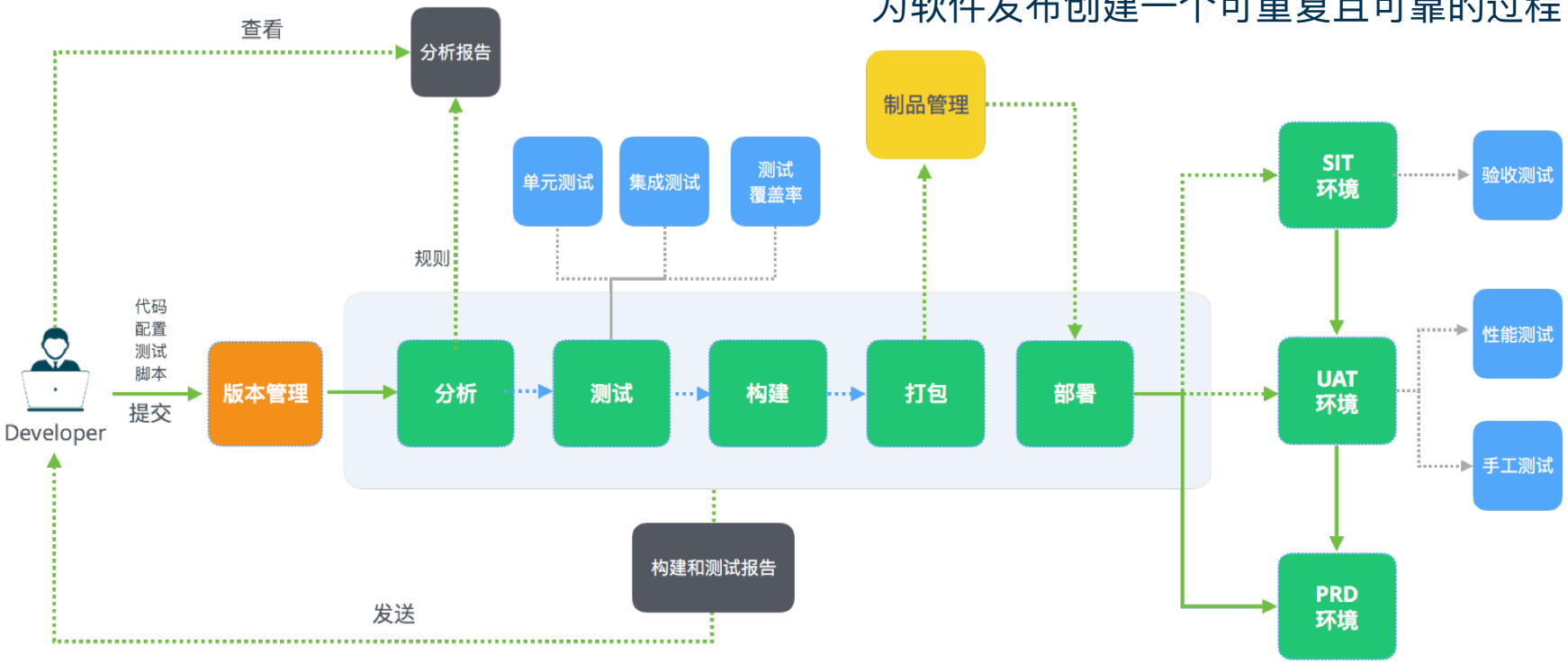


上下游之间高频沟通是为了使整个团队达成一致目标，我们认为在过程中投入的团队成本是值得的

「持续交付」

持续交付工作流

为软件发布创建一个可重复且可靠的过程

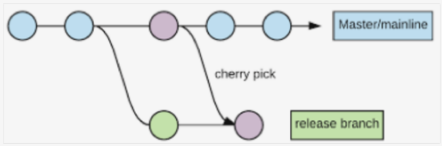


- 将所有内容都纳入到版本控制
- 将所有重复的工作都自动化
- 将质量评估和反馈内嵌到所有环节
- 保证交付件在不同环境部署的一致性
- 频繁提交并保证流程是可工作的

持续交付 Tips

分支策略

主干开发（Trunk Based Development）

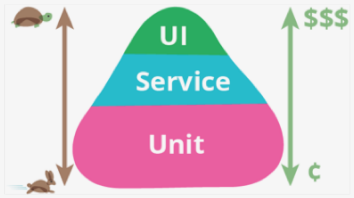


- 主干开发分支发布
- 定期合并分支到主干

- 减少代码合并带来的额外消耗

测试策略

测试驱动开发 （Test Driven Development）

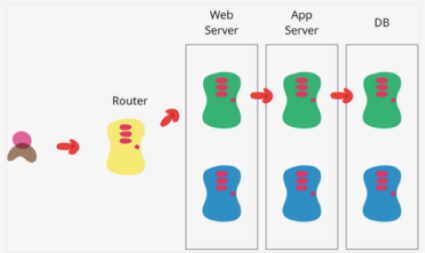


- 测试金字塔
- 耗时与成本

- 加强代码级测试来降低投入并增加产出

部署策略

蓝绿部署 （Blue Green Deployment）



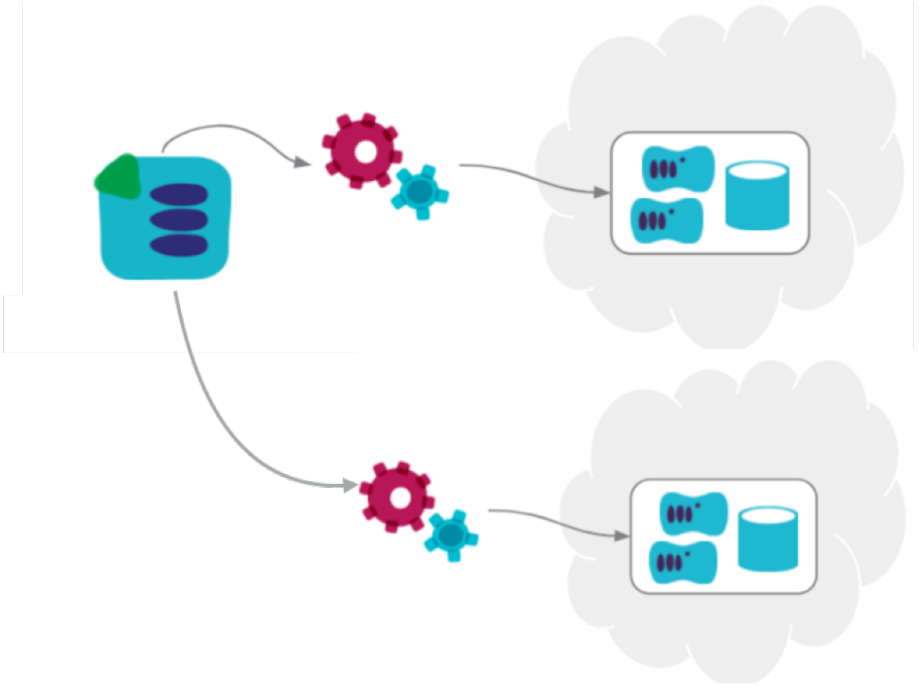
- 服务注册
- 负载均衡

- 保障业务连续性的同时提供回滚机制

「IT 服务管理」

Everything as Code (代码化)

- 应用 Applications
- 基础架构 Infrastructure
- 配置 Configuration
- 测试 Tests
- 流程 Process
- 验收 Acceptance



使用「自动化操作」和「标准化配置」来替代「手工操作」和「文档撰写」等工作

例子: 基础设施即代码

服务器列表.xlsx

```
# This is the inventory file of UAT environments

[back-end-and-infra:children]
back-end-servers
infrastructure

[back-end-servers]
database ansible_host=10.0.0.1 ansible_port=22 ansible_user=postgres
webserver ansible_host=10.0.0.2 ansible_port=22 ansible_user=root

[infrastructure]
ldap ansible_host=10.0.0.100 ansible_port=22 ansible_user=root
```

安装配置手顺.docx

```
main.yaml
# This Playbook Install Docker

- name: Set Hostname
  hostname: name=docker-{{ hostname }}

- name: Install SoftWare For CentOS
  yum: name={{ item }} state=latest
  with_items:
    - yum-utils
    - lvm2
    - device-mapper-persistent-data
  when: ansible_pkg_mgr == "yum"

- name: Install SoftWare For Ubuntu
  apt: name={{ item }} state=latest
  with_items:
    - apt-transport-https
    - ca-certificates
    - gnupg2
    - software-properties-common
  when: ansible_pkg_mgr == "apt"

- name: Upload Docker REPO For CentOS
  copy: src=docker.repo dest=/etc/yum.repos.d/
  when: ansible_pkg_mgr == "yum" and ansible_distribution_major_version == "7"

- name: Upload Docker REPO For Ubuntu stp 01
  apt_key: url=http://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg state=present
  when: ansible_pkg_mgr == "apt" and ansible_distribution_major_version == "16"

- name: Upload Docker REPO For Ubuntu stp 02
  apt_repository: repo='deb [arch=amd64] http://mirrors.aliyun.com/docker-ce/linux/ubuntu xenial stable' filename='docker-ce'
  when: ansible_pkg_mgr == "apt" and ansible_distribution_major_version == "16"
```

代码化 Tips

Application as Code

- 业务代码
- 构建配置
- 单元测试
- 集成测试

Test as Code

- 功能测试脚本
- 性能测试脚本
- 配置文件
- 测试数据

Pipeline as Code

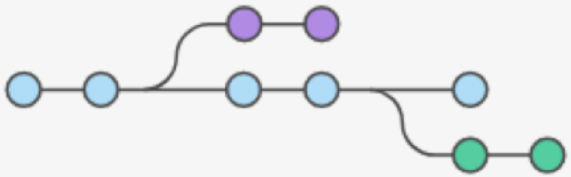
- 构建脚本
- 部署脚本
- 渲染脚本
- 配置模板

Infrastructure as Code

- 主机列表
- 配置模板
- 执行剧本
- 环境变量

分支策略

功能分支（Feature Branch Workflow）



目的

- 确保所有修改都有 Review

方式

- 所有功能开发在专用分支中进行
- 开发完毕之后提交 Merge Request
- 使用 Master 分支进行操作

CHINA DEVOPS DAYS

CHINA DEVOPS DAYS | IT大咖说 知识共享平台

技 术

架构 和 工具

康威定律

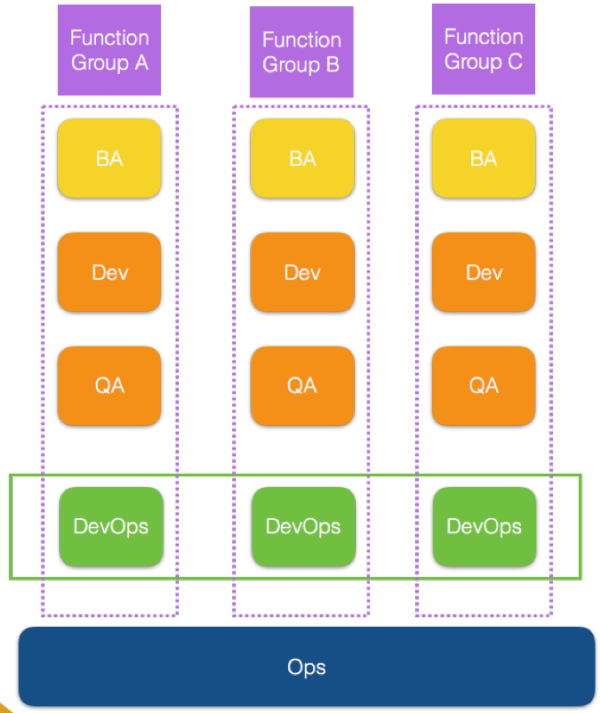
CONWAY'S LAW

"Any organization that designs a system will inevitably produce a design whose structure is a copy of the organization's communication structure."
Melvin E. Conway, 1967

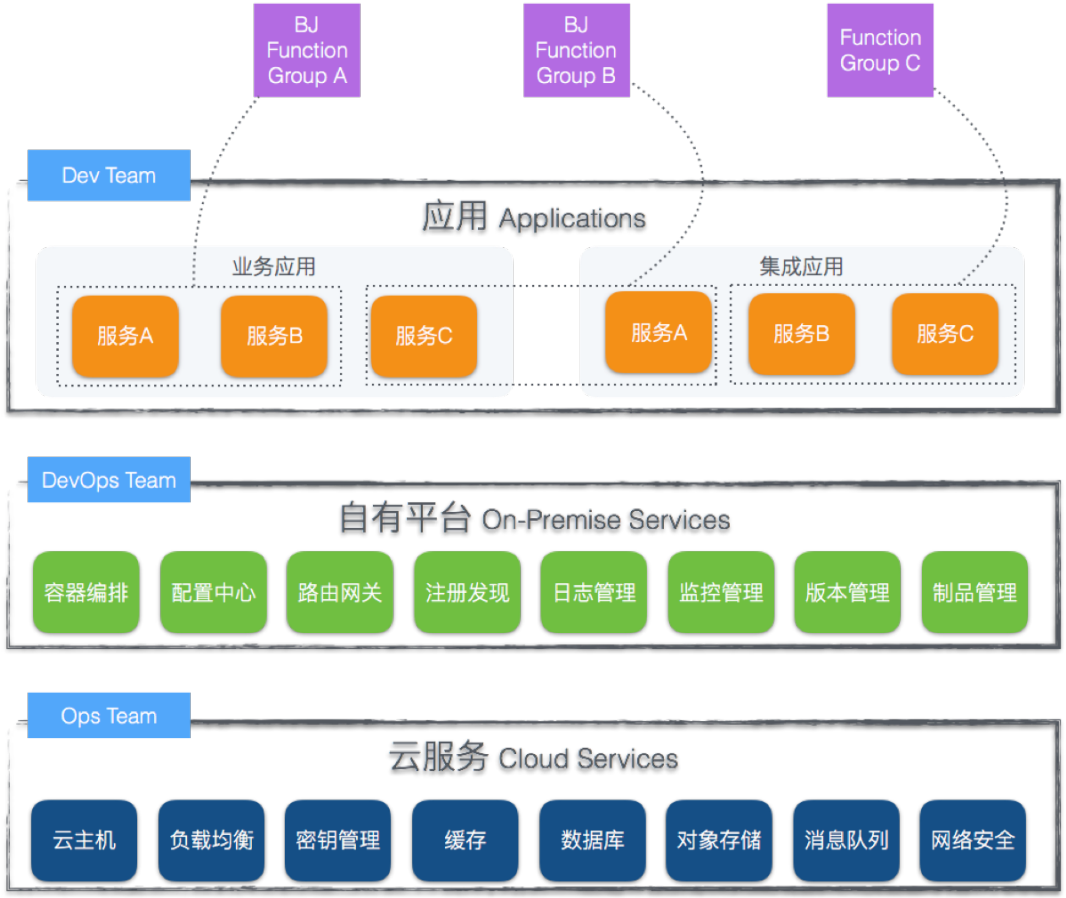
组织结构决定系统架构

CHINA DEVOPS DAYS 整体架构

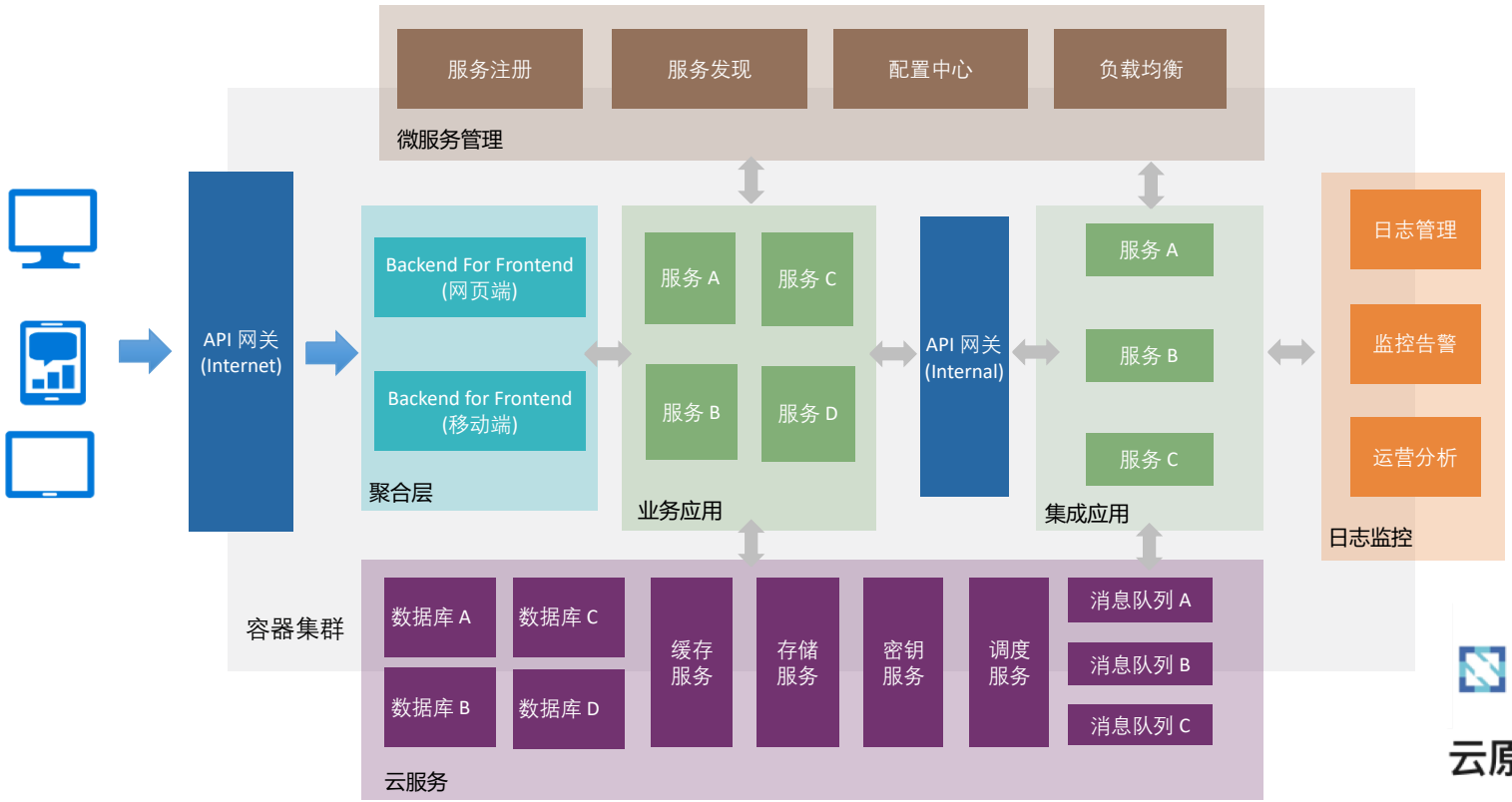
组织结构



系统架构



应用架构



CLOUD NATIVE COMPUTING FOUNDATION

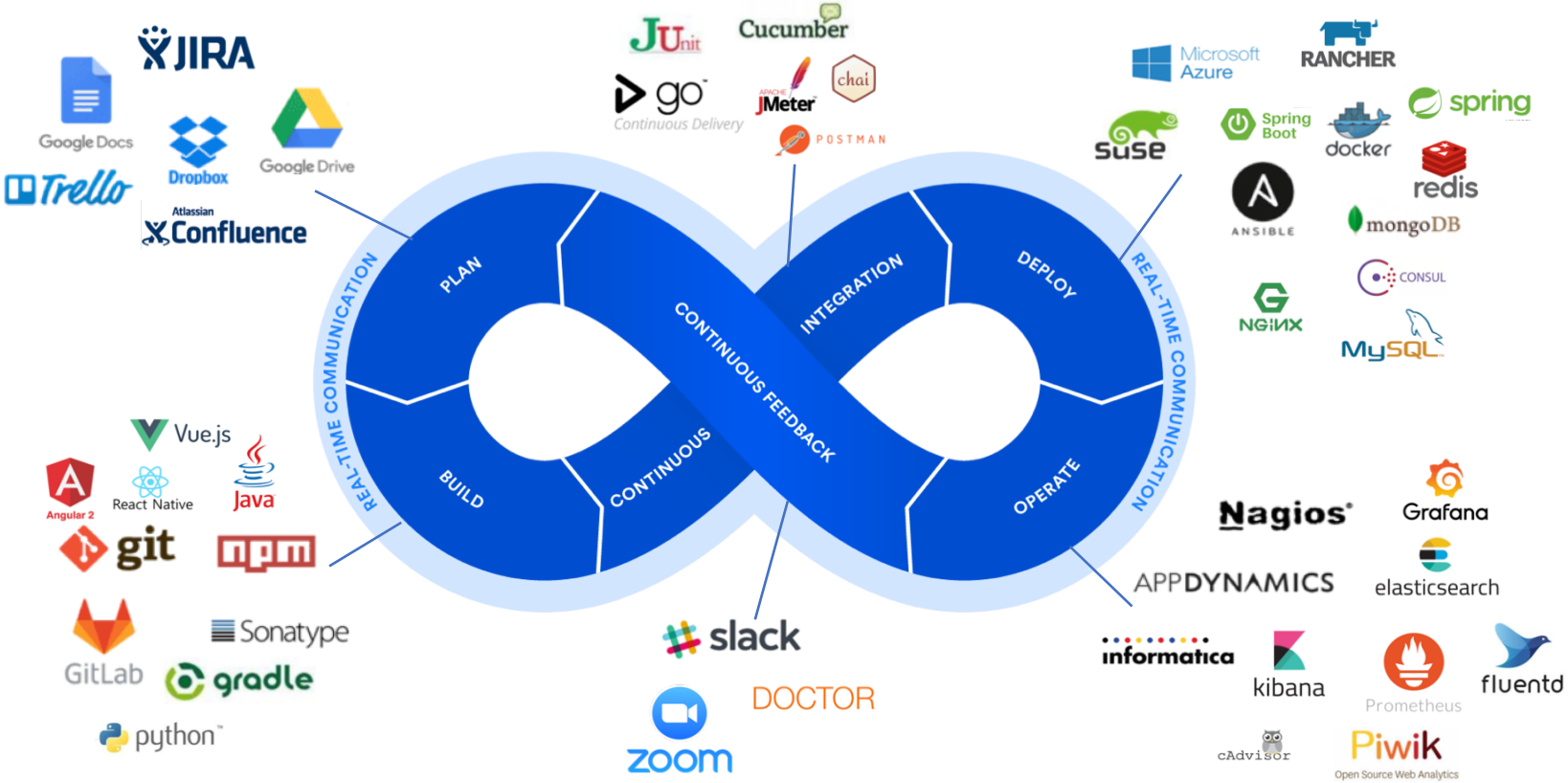
云原生应用:

(a) 容器化封装

(b) 自动化管理

(c) 面向微服务

工具链

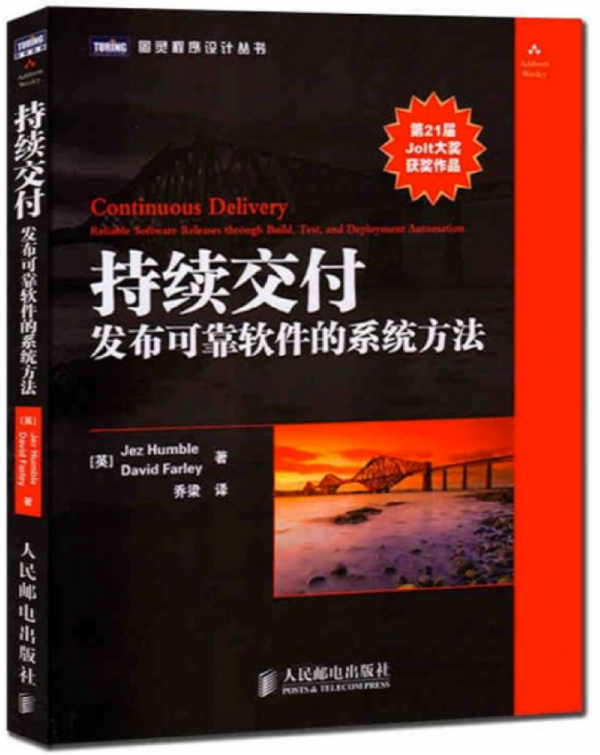

JIRA
Google Docs
Dropbox
Google Drive
Trello
Atlassian Confluence
JUnit
Cucumber
go
Continuous Delivery
JMeter
chai
POSTMAN
Microsoft Azure
RANCHER
SUSE
Spring Boot
spring
docker
redis
ANSIBLE
mongoDB
CONSUL
NGINX
MySQL
PLAN
CODE
BUILD
CONTINUOUS INTEGRATION
TEST
DEPLOY
RELEASE
CONFIGURE
OPERATE
CONTINUOUS FEEDBACK
REAL-TIME COMMUNICATION
REAL-TIME COMMUNICATION
Vue.js
Angular 2
React Native
Java
git
npm
GitLab
Sonatype
gradle
python
slack
DOCTOR
zoom
Nagios
Grafana
APPDYNAMICS
elasticsearch
informatica
kibana
Prometheus
fluentd
cAdvisor
Piwik
Open Source Web Analytics

CHINA DEVOPS DAYS

GoCD

GoCD 是一个专用于高级工作流建模和依赖关系管理的开源持续交付服务器。其价值流示意图能让从提交到部署的代码变更的跟踪工作变得一目了然，为完整的工作流提供卓越的可视化程度。

- 复杂流水线支持
- 可视化工作流
- 构建对比分析
- 按需构建部署
- 插件系统
- 社区生态



没有最好的工具，只有最合适的工具

改 善

持续改进

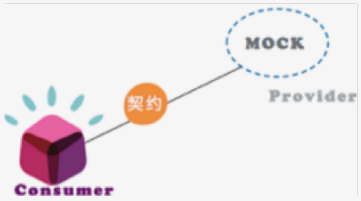
DevOps 看板

可视化工作流

Dev看板和Ops看板是割裂的

契约测试

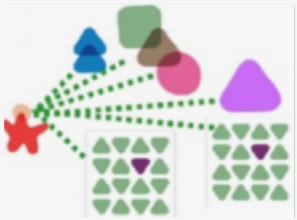
降低测试成本



端到端测试并不太适合微服务

监控分级

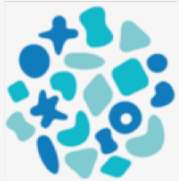
快速定位问题



告警触发后无法找到故障点

文档库

知识积累传递



新成员难以快速投入工作

CHINA DEVOPS DAYS

CHINA DEVOPS DAYS | IT大咖说 知识共享平台

DevOps 不是目标 而是探索

CHINA DEVOPS DAYS

CHINA DEVOPS DAYS | IT大咖说 知识共享平台

DEVOPS DAYS BEIJING 2018

THANKS