



# DevOpsDays

## Shanghai

— 2017.8.18-8.19 —

上海龙之梦酒店（长宁区延安西路1116号）

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





# 链接Dev与Ops的正确姿势-灰度发布

万金 Thoughtworks 高级咨询师

# 关于我



万金 Thoughtworks 高级咨询师，10年+，知名外企与中国企业的IT从业经验，包括IBM，华为，中兴，Thomson. 具有8年云计算相关经验，多系统的研发和运维经验，熟练掌握敏捷和DevOps方法论和实践，对软件研发工具链有一定研究。

《DevOps实施手册 在多级IT企业中使用DevOps》

译者（近期出版）

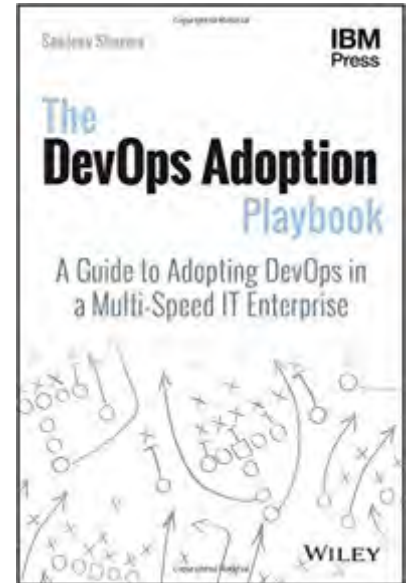
近期发表文章：

Thoughtworks洞见：从集装箱历史看DevOps的发展进程

<http://insights.thoughtworkers.org/process-of-devops-development/>

**InfoQ：当DevOps遇见AI，智能运维的黄金时代**

<http://www.infoq.com/cn/articles/when-devops-meets-ai>



# 目录

- ➔ **1** 数字化转型的趋势与挑战
- 2** 软件发布的各种坑
- 3** 云原生应用如何发布软件
- 4** 下一站智能运维

# 分工协作提高效能达到增长极限

1. 工业时代增长红利的终结
  - 单一产品的竞争优势被行业外颠覆
  - 低价高质不再是客户选择的关键因素
  - 人的延伸价值观的局限



**“我想要雇佣的明明只是一双手，但为什么要让我面对一个人。”**

**--亨利 福特 (Henry Ford)**

# 追求个性化用户体验带来新的增长

## 1. 分工在组织外部进行带来用户体验优化

- 硬件与软件分离（IBM与微软）
- 服务与产品分离（AWS与Netflix）
- 业务与实现分离（企业与IT外包商）
- 使用与拥有分离（用户与共享经济）

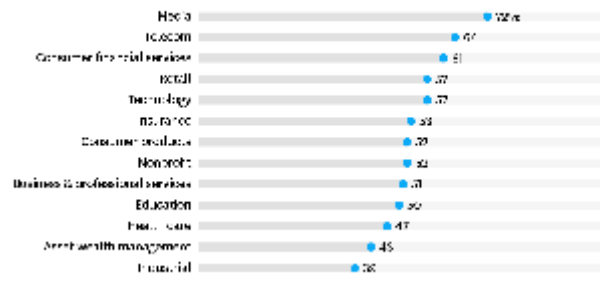


**企业把一部分业务外包出去，满足个性化用户体验，达到产品/服务价值的提升。**

# 传统行业面临数字化转型的两个挑战

1. 提升不同分工的协同效率
  - 如何通过互联网化工具优化协同效率
2. 精准把握最终用户需求
  - 精确：追求用户特定场景下的个性化需求（通过协同网络收集反馈）
  - 准确：从满足标准化需求到挖掘潜在需求。建立产品管道持续反馈，形成特定时间点特性需求

Executives Who Anticipate Moderate or Massive Digital Disruption In the Next 12 Months, by Industry



《哈佛商业评论》传统企业面临巨大挑战

# 目录

**1** 数字化转型的趋势与挑战

**➔ 2** 软件发布的各种坑

**3** 云原生应用如何发布软件

**4** 下一站智能运维



# 大量手动过程导致低效和事故

## Application PaaS



1. 软件编译复杂
  - 软件编译第三方依赖关系复杂
  - 多技术栈解决方案复杂性高
  - 多分支并行开发策略复杂
2. 测试经历软件测试环境类生产境等迁移
  - 测试环境搭建
  - 测试数据准备
  - 界面测试无法自动化
3. 软件发布过程无法保证不出问题
  - 发布流程与过程时间长
  - 手动或配置过程导致发布失败

# 目录

**1** 数字化转型的趋势与挑战

**2** 软件发布的各种坑

**➔ 3** 云原生应用如何发布软件

**4** 下一站智能运维

# Application Pass项目背景与挑战

## • 项目背景

- 客户是欧洲汽车制造企业，一部分IT项目由外部供应商研发，客户正处于从传统制造企业向数字化转型阶段

## • 项目挑战

- 缺乏新技术新平台的运营能力，某些项目开始自己负责部分运维工作，不规范可靠性低，不想被大型云计算厂商绑定
- 随着外包项目增多，软件运行环境复杂，交付周期长，传统IT部门很难支持产品快速迭代
- 为不同供应商运维软件复杂度高，管理成本居高不下
- 将IT系统运行的运维和运营信息反馈到研发部门，以平台为基础让生产厂商与用户频繁互动，使汽车生产厂家洞察用户需求成为可能。

总结：引入平台降低新技术引入门槛，提升研发和运维效率，通过平台收集信息把握最终用户需求。

# DevOps软件研发实践

- Tools工具
- 1. Infrastructure as code (自动化基础设施)
- 2. Shared version control (在研发与运维角色共享版本信息)
- 3. One step build and deploy (一键式部署)
- 4. Feature flags (特性开关保证随时发布)
- 5. Shared metrics (开放监控信息给研发以方便定位软件问题)
- 6. IRC and IM robots (“IRC“Internet Relay Chat, 通过聊天形式把所有人的上下文统一起来, 今天提到的比较多的就是ChatOps)



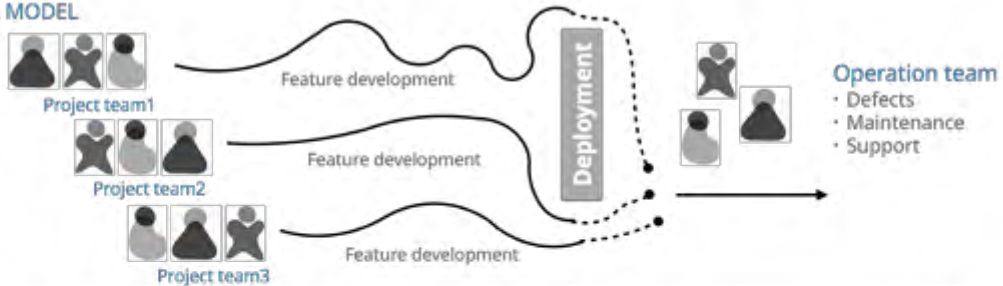
# DevOps 软件研发实践

- Culture 文化
  - 1. Respect (尊重：工程师文化)
  - 2. Trust (信任：责任共担)
  - 3. Healthy attitude about failure (试错：从错误中学习)
  - 4. Avoiding blame (避免问责，而是根因分析避免再次出错)

# 部署与功能分离：从项目到平台

## FROM PROJECT TO PLATFORM

### TRADITIONAL MODEL



### STRATEGIC MODEL

#### Platform team

- Features
- Enhancements
- Defects
- Maintenance
- Support

Long-lived platform team align to business KPIs, with stewardship form a business owner

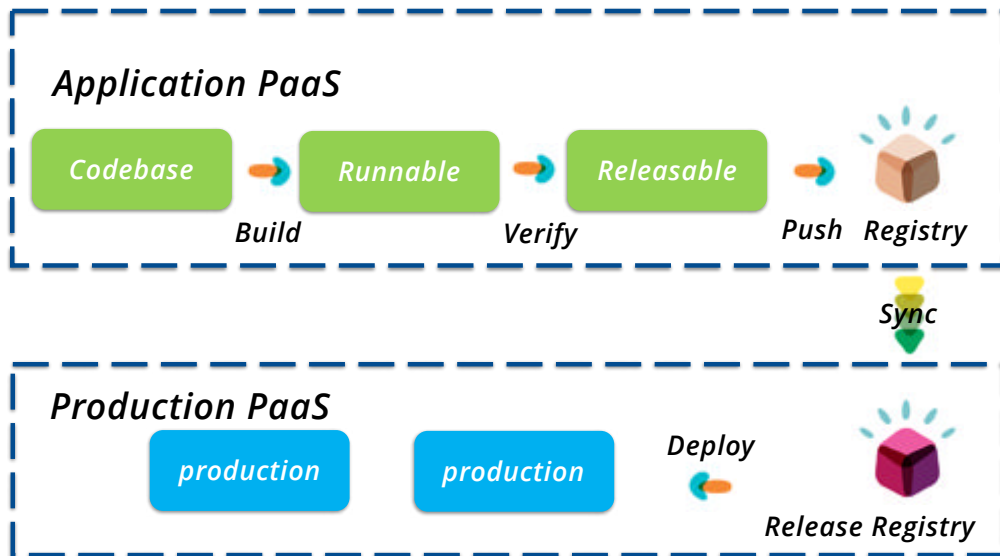


They work from a **Prioritised backlog** Containing features, defects and maintenance

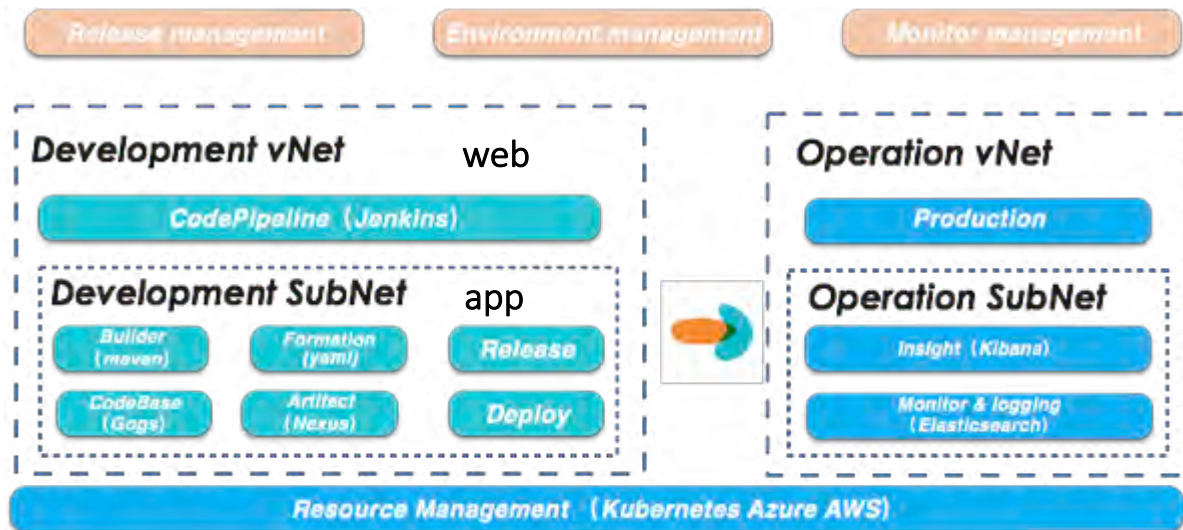
Second-level **Support** is also handled by the platform team. You build it, you support it.

Everyone in the team has a **Clear understanding** of the technical design and business vision.

# Dev和Ops需要两个PaaS平台



# Application PaaS架构





# 开源技术选型实现

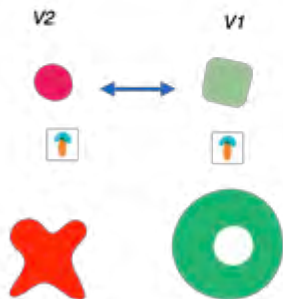
同时本方案还可以使用开源软件的选型实现一定的灵活性，避免技术绑定



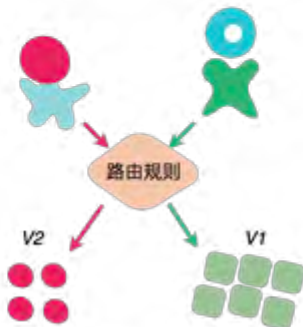
# 试验性发布-灰度发布关键环节

灰度发布是在软件发布阶段将部署与发布分离，采用蓝绿部署、金丝雀发布或特型开关等手段，在生产环境中保持两个或多个版本，基于小范围验证的结果再逐步扩大发布范围，以此控制发布风险。

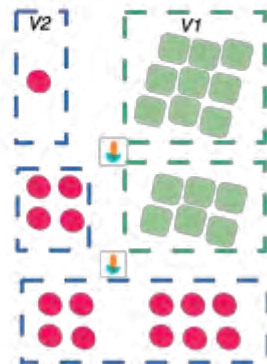
## 应用监控数据



## 用户分流规则

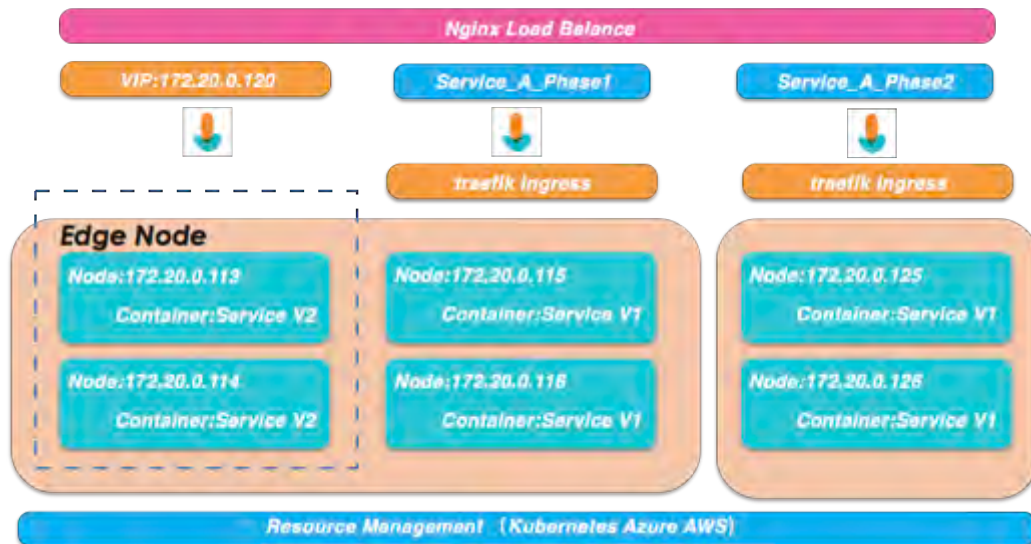


## 递进发布策略



# 用户分流实现：k8s边缘节点（Edge Node）

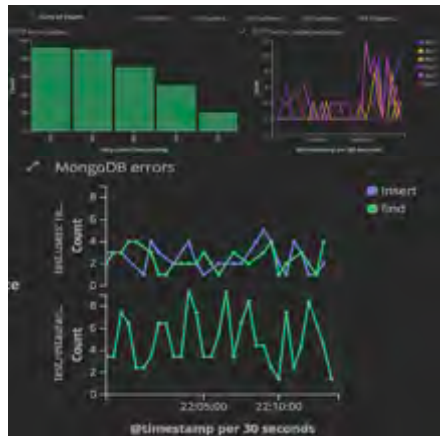
使用边缘节点访问K8s集群内部新版本应用进行部署功能验证，Phase1进行兼容性验证，Phase2集群进行性能压力验证。



看看K8s手册：<https://rootsongjc.gitbooks.io/kubernetes-handbook/content/practice/edge-node-configuration.html?h=%E8%BE%B9%E7%BC%98%E8%8A%82%E7%82%B9>

# 应用监控数据：Kibana应用监控

1. 使用Kibana监控灰度升级过程
  - 功能验证阶段，对比应用请求返回值错误比例
  - 兼容性验证阶段，对比服务正确处理情况
  - 性能阶段，对比可用性和资源使用情况
2. 通过业务监控获取用户需求
  - 与运营部门合作，实现关键运营指标监控，比如：转化率，ARPU，用户增长率等
  - 通过与研发共享日志信息为应用性能与用户体验优化提供支持

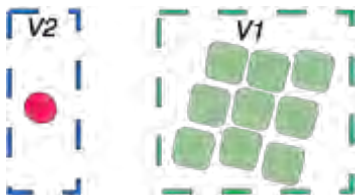


# 递进发布：Kubernetes滚动升级

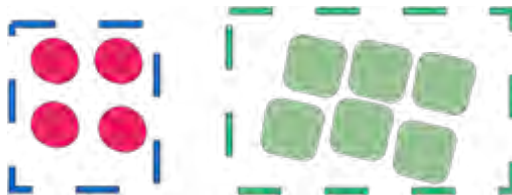
## 持续交付到生产环境边缘节点（Phase-0）

Phase-0监控应用报错信息，当满足发布条件进入灰度发布节点，逐步升级至全网或多数据中心  
集群滚动升级：\$ kubectl rolling-update App-Name-v1 -f App-Name-v2-rc.yaml --update-period=10s

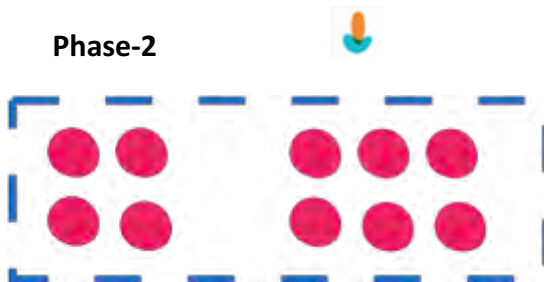
Phase-0



Phase-1



Phase-2



# 通过平台能力开放，从单一产品竞争走向生态竞争



1. 内部：平台+小团队特种部队组织
  - 平台负责效率小团队负责方向  
市场+解决方案+交付对接
2. 外部：以不确定性对抗不确定性
  - 学习谷歌把一件核心的事情做好，其他的交给别人。（one in all）
  - 开放能力给第三方共同完成目标。



# 通过平台能力开放，从单一产品竞



## 某国内大型通讯公司公有云实验性发布方案

本方案是一个分布式公有云方案，主中心在中国，全球主要大洲都有数据中心。发布由主中心发起逐步扩大范围到所有数据中心。



发布包通过功能和性能测试后，会在类生产环境验证部署方式和配置正确性。通过验证后会推送发布包到各个数据中心，同时在主中心进行实验性发布，通过正确性、兼容性、性能等测试后会逐步扩大到其他数据中心。

# 目录

**1** 数字化转型的趋势与挑战

**2** 软件发布的各种坑

**3** 云原生应用如何发布软件

**➔ 4** 下一站智能运维

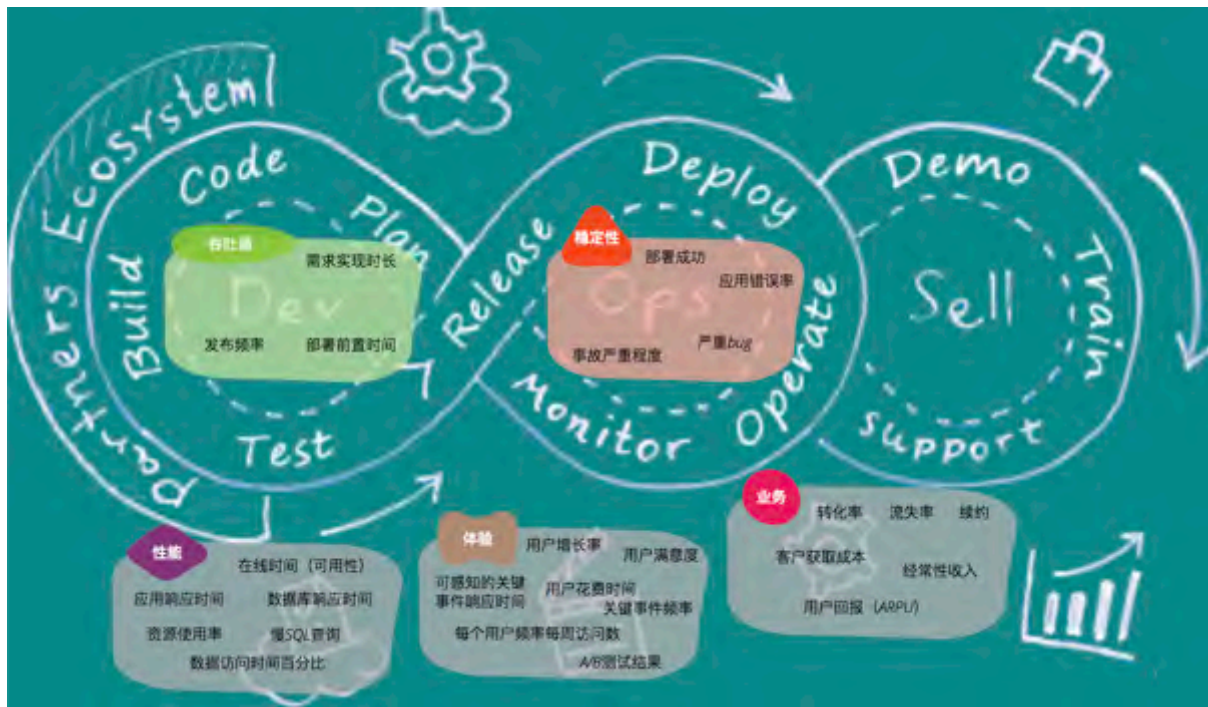


# 当我们说到“智能”我们在谈什么？



1. 系统一不需要思考
  - 判断一个人的情绪
  - 判断一部电影的体验
2. 系统二需要逻辑性思考
  - 数学计算
  - 数据搜索
  - 找到数据相关性

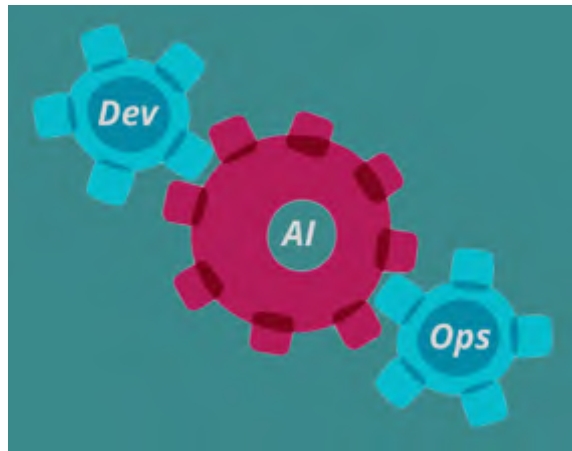
# 软件研发流程数据化个体在线化



# 人工智能辅助研发流程改进解决效率和精准把握最终用户需求

## 1. 软件体验不确定性

- 设计的不确定性
- 软件设计与建筑的差异
- 质量的不可见性
- 软件质量缺陷短期内不易发现
- 价值的不透明性
- 软件研发的流程前后环节传递无法直接体现对最终发布后的价值





高效运维社区  
GreatOPS Community

会议

- 8月18日 DevOpsDays 上海
- 全年 DevOps China 巡回沙龙
- 11月17日 DevOps金融上海

培训

- EXIN DevOps Master 认证培训
- DevOps 企业内训
- DevOps 公开课
- 互联网运维培训

咨询

- 企业DevOps 实践咨询
- 企业运维咨询



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



想第一时间看到  
高效运维社区公众号  
的好文章吗？

请打开高效运维社区公众号，点击右上角小人，如右侧所示设置就好

