

DOIS | 2018 · 深圳站  
DevOps 落地，从这里开始

# DevOps 国际峰会

暨 DevOps 金融峰会

指导单位： 云计算开源产业联盟  
Open Source Cloud Alliance for Industry (OSCAR)

主办单位： DevOps时代

 高效运维社区  
GreatOPS Community

时间：2018年11月2日-3日

地址：深圳市南山区圣淘沙大酒店（翡翠店）

# 大型制造业实践DevOps 团队之路

郭家齐 微软最有价值专家

# 我是谁

DOIS



现任 金士顿科技公司 信息处 经理

- 前 全球第三大面板制造商 营运信息处 副理
- 连续六届 微软最有价值专家
- 台湾技术社区讲师
- 2018 DevOps Day Taipei 讲师
- 2017 Microsoft Global Tech Summit 北京 讲师
- 2017 Agile Tour Hsinchu 讲师
- 2017 制造管理新智慧 讲师



A photograph of a road fork under a dark, stormy sky. The road splits into two paths. In the background, there are yellow chevron signs. On the right side of the road, there is a diamond-shaped warning sign with a yellow square in the center, and below it, a square sign with a black arrow indicating a right turn.

# DevOps

制造业与互联网的套路绝对不同



# Dev & Ops

DevOps是制造业的刚需

# 目录



1

DevOps 开端

2

DevOps 团队文化

3

DevOps 流程演进

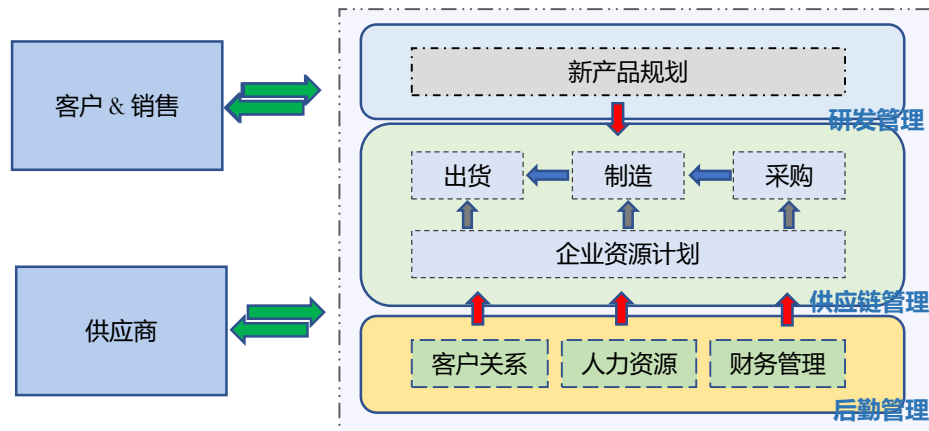
4

DevOps 技术架构

# 极度仰赖IT的产业

## ➤驱动企业数字化转型

- 企业神经中枢，集成企业所有信息
- 联系组织内外的纽带
- 效率是影响企业市场竞争力
- 驱动企业数字化转型的关键
- 敏捷的IT才能替公司获利
- 企业能创新的驱动力





## ➤大部分的制造业现状

- 需求开发、交付的时程，赶不上市场变化
- 修复崩溃的系统时间过长，影响企业获利
- 制造业系统众多、维护成本逐渐升高，影响开发效率
- IT人员有一定比例的编制
- IT人员的质量、能力不一
- 容许「失败」的风险能力较低
- 开发与运维的沟通成本高，拖垮IT部门效能



## ➤制造业的IT是项目开发，非产品开发

- 项目只有交付日期，项目服务没有终止日期
- 常見的场景是交付第一版需求后，用户才能准确说出自己的需求
- 确认完整的需求时程，赶不上市场变化
- 项目优先权，变化莫测
- 项目的范围大小，无法固定

# 开发与运维双剑齐发

- 「开发」 强化企业获利能力
- 「运维」 稳定企业获利来源
- 「开发」 和 「运维」 失衡对于企业效率有极大影响

**「运维成本」 > > 「开发成本」**

# 目录

1 DevOps 开端

➔ 2 DevOps 团队文化

3 DevOps 流程演进

4 DevOps 技术架构

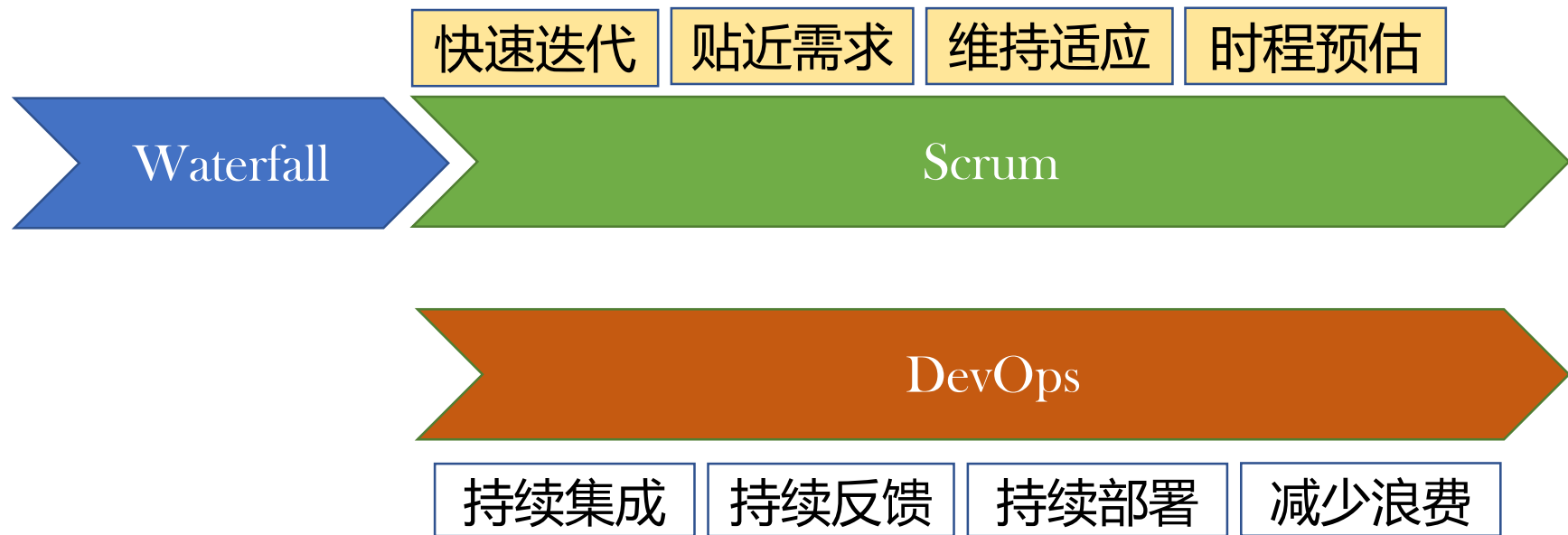


# DevOps 7种习惯



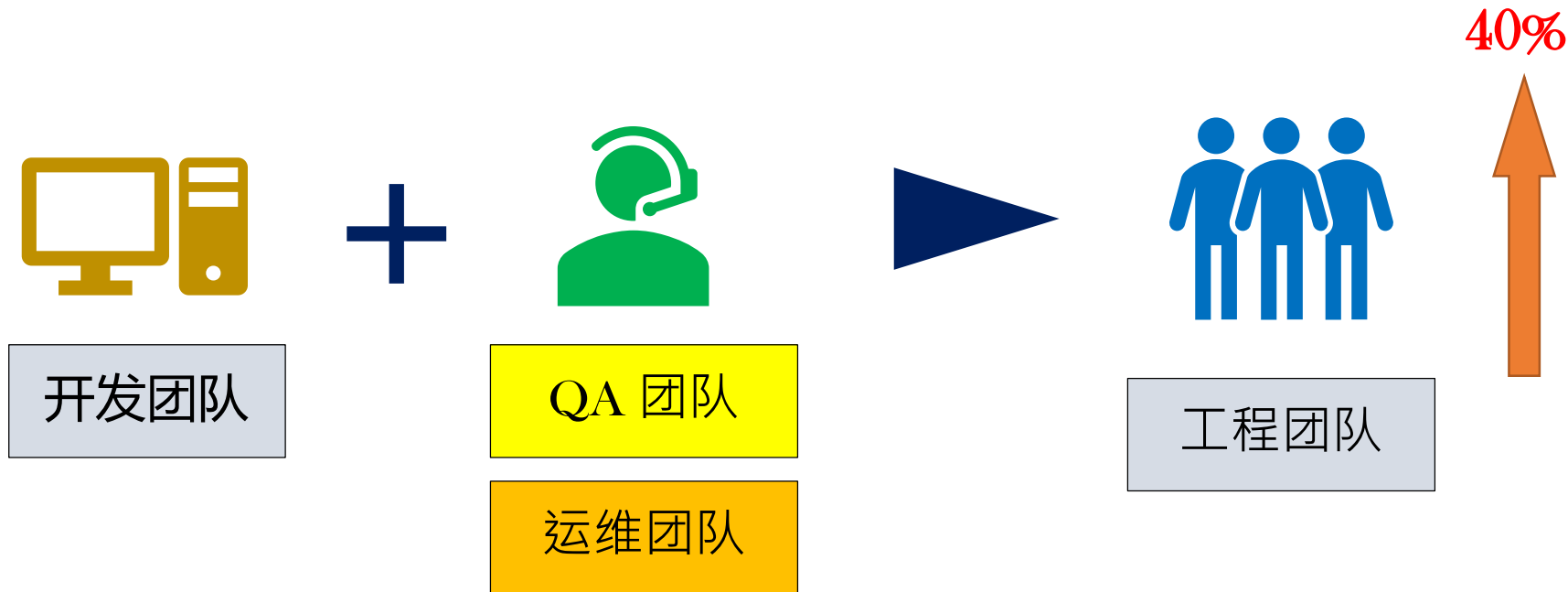
- 自我组织团队的目标须与企业目标一致
- 严格管理技术债
- 专注于客户/用户价值流
- 假设理论，驱动开发
- 搜集(监控)正式环境系统运作信息
- 实战文化
- 管理硬件架构作为灵活资源

# 双轨理论



# 建置团队

DOIS





# 团队DevOps文化



- 团队每个人一起致力让系统能够平稳且有效率
- 信息共享、互助合作共同承担成功与失败
- 团队每个人互相扶持、相信人人都可以改变
- 每个人努力让团队变得更好，不是追求自我成功或是价值

# 目录

1 DevOps 开端

2 DevOps 团队文化

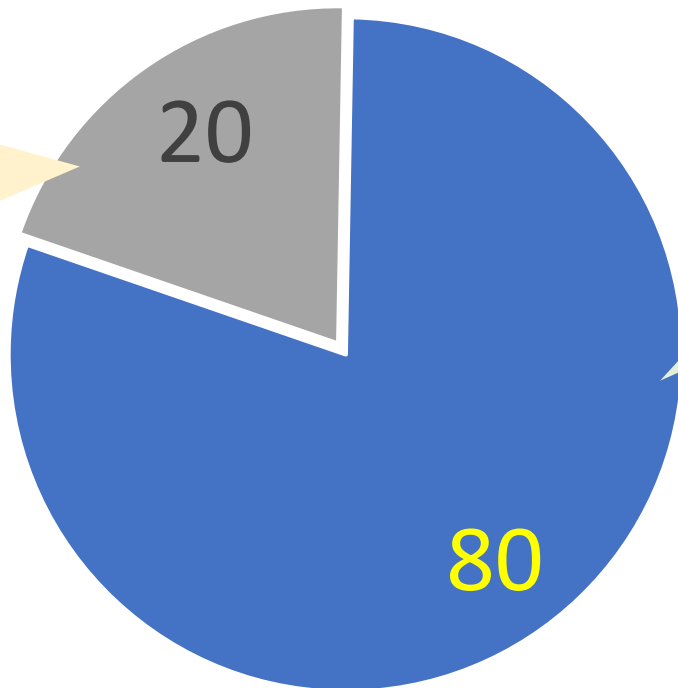
➔ 3 DevOps 流程演进

4 DevOps 技术架构

# 需求

## 工程需求

建构Infrastructure、系统的改善或重构项目。须放在同一个Backlog被追踪和完成，不可以被排除



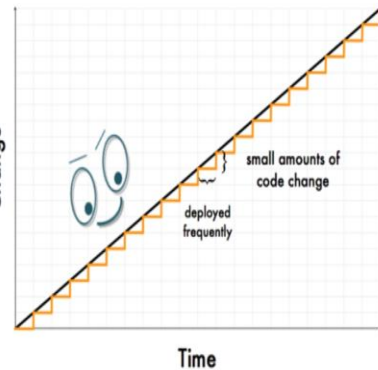
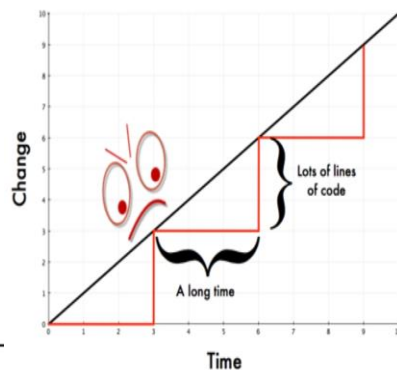
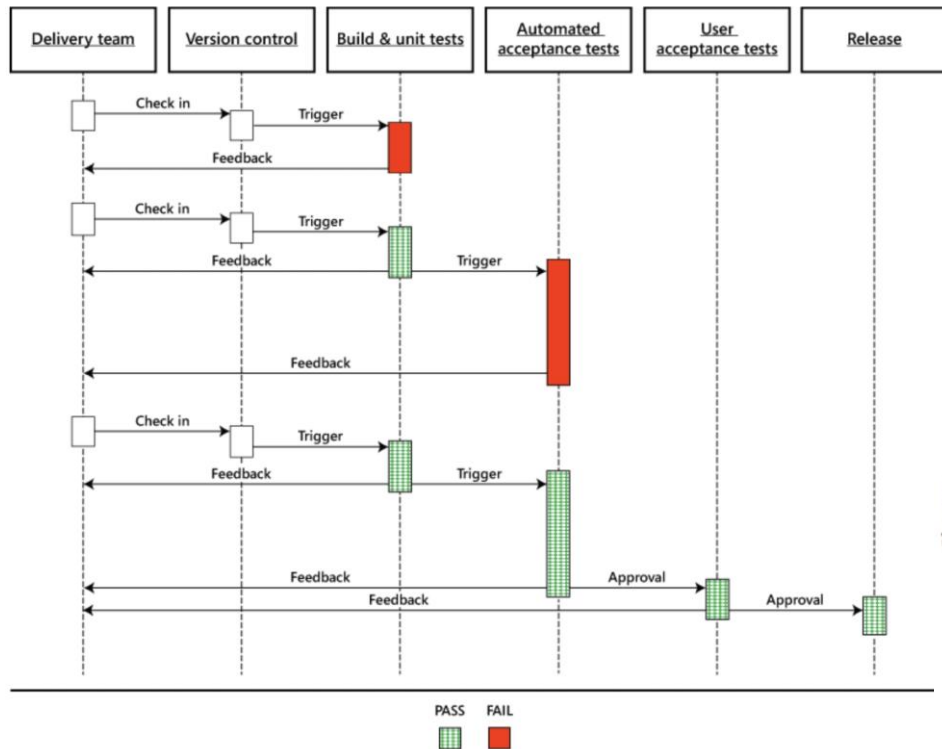
## 用户需求

来自商业需要的功能或是用户需求

# 版本控制

- 有效的建立版本控制的分支策略
- 签入的程序代码必须连接到工作项目
- 只要是程序代码都要纳入版控
- 基础结构设置 (IaC), 需求传递版本控制

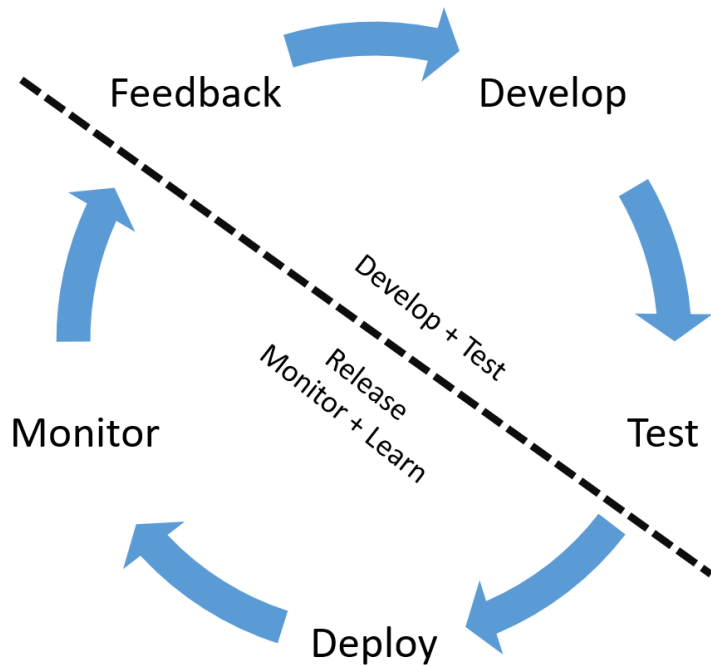
# 持续集成、持续发布



- 自动化是测试的关键
- 单元测试、整合测试、UI测试到负载平衡测试都是属于持续测试一环
- 使用增量和迭代方式，让测试的深度不断前进
- 别期望一开始就写好所有测试案例
- 最好的测试，就是增量测试

# 监控与学习

- 提升质量
- 提高安全
- 增加可靠性
- 持续提供价值给用户
- 了解用户如何使用系统





# 目录

1 DevOps 开端

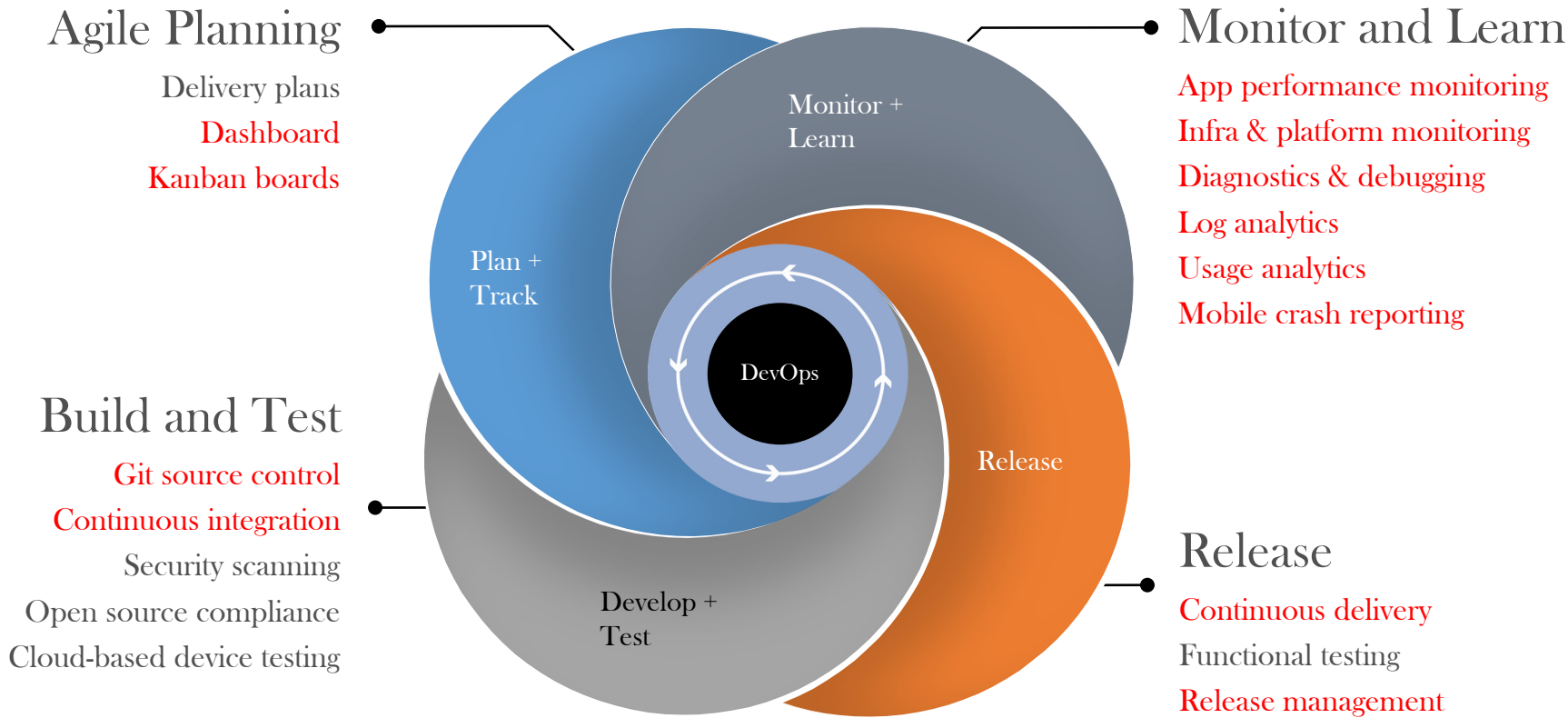
2 DevOps 团队文化

3 DevOps 流程演进

➔ 4 DevOps 技术架构

# DevOps技术栈

DOIS



- 企业组织需要确保不同的团队，使用的工具可以整合。
- 工具链不配套，会导致许多团队发生碰撞。
- 有效的工具整合是支持开发和运维方面最有帮助的技术。
- 降低试错风险成本与推进时效
- 减少人员开发与运维成本
- 实时因应市场变化，实时应对。
- 时间花在目的，而不是过程

# DevOps架构推进

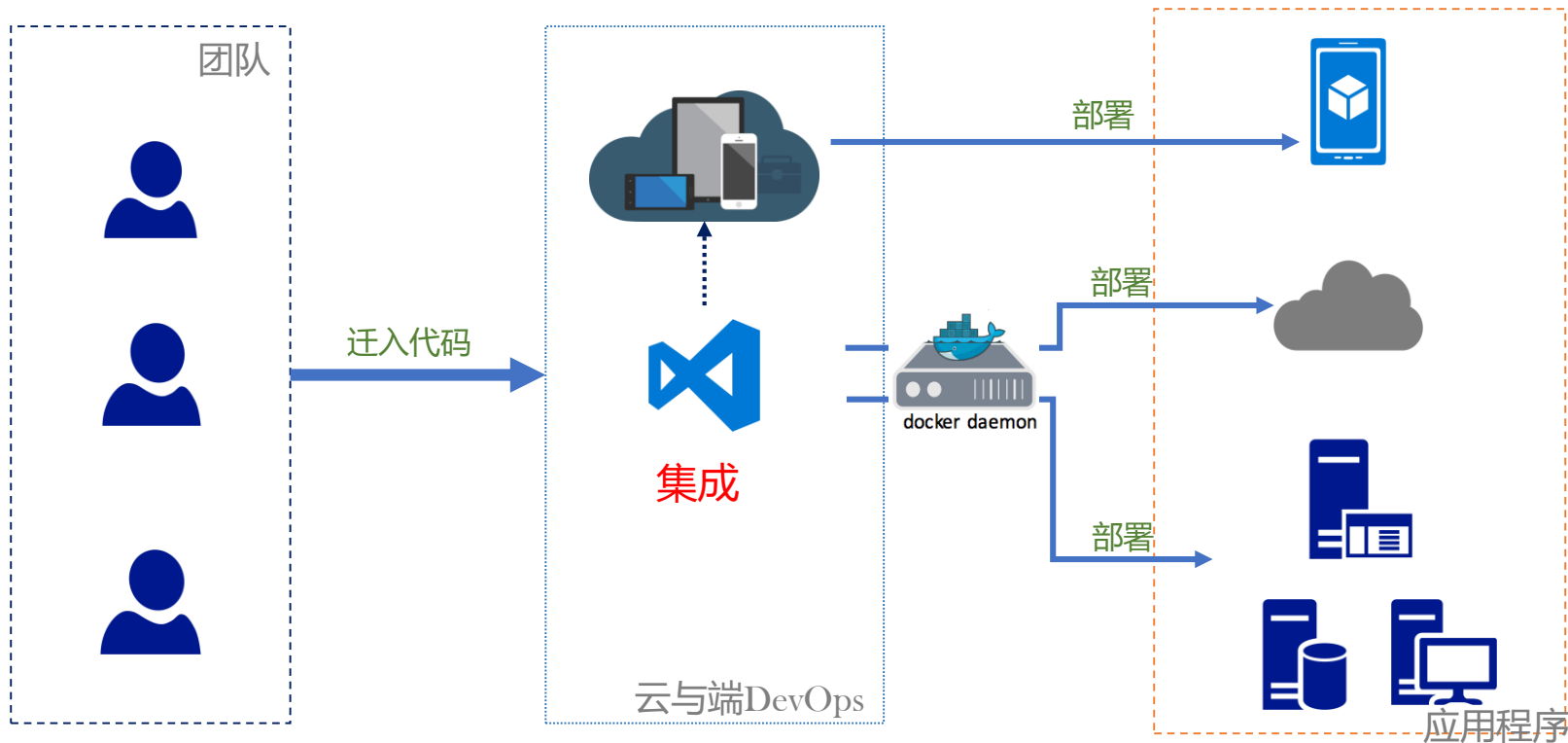


- 先从新项目、新系统开始导入DevOps思维
- 系统架构必须是低耦，所以，系统架构必须重新设计
- 80% 自动化, 20% 人工作业
- 善用云技术

# 自动化好处

- 让人可以不必执行重复性且单调的工作，减少人工错误
- 让人有更多时间，创造公司的商业价值
- 自动化比手工速度更快
- 自动化可以提高发布的质量。 因为部署的步骤标准化。 并在确保的环境与流程中，预期后续会产生结果
- 从中长期来看，自动化是可以降低成本。

# DevOps技术架构



## ➤透过YAML格式，建立持续集成流水线

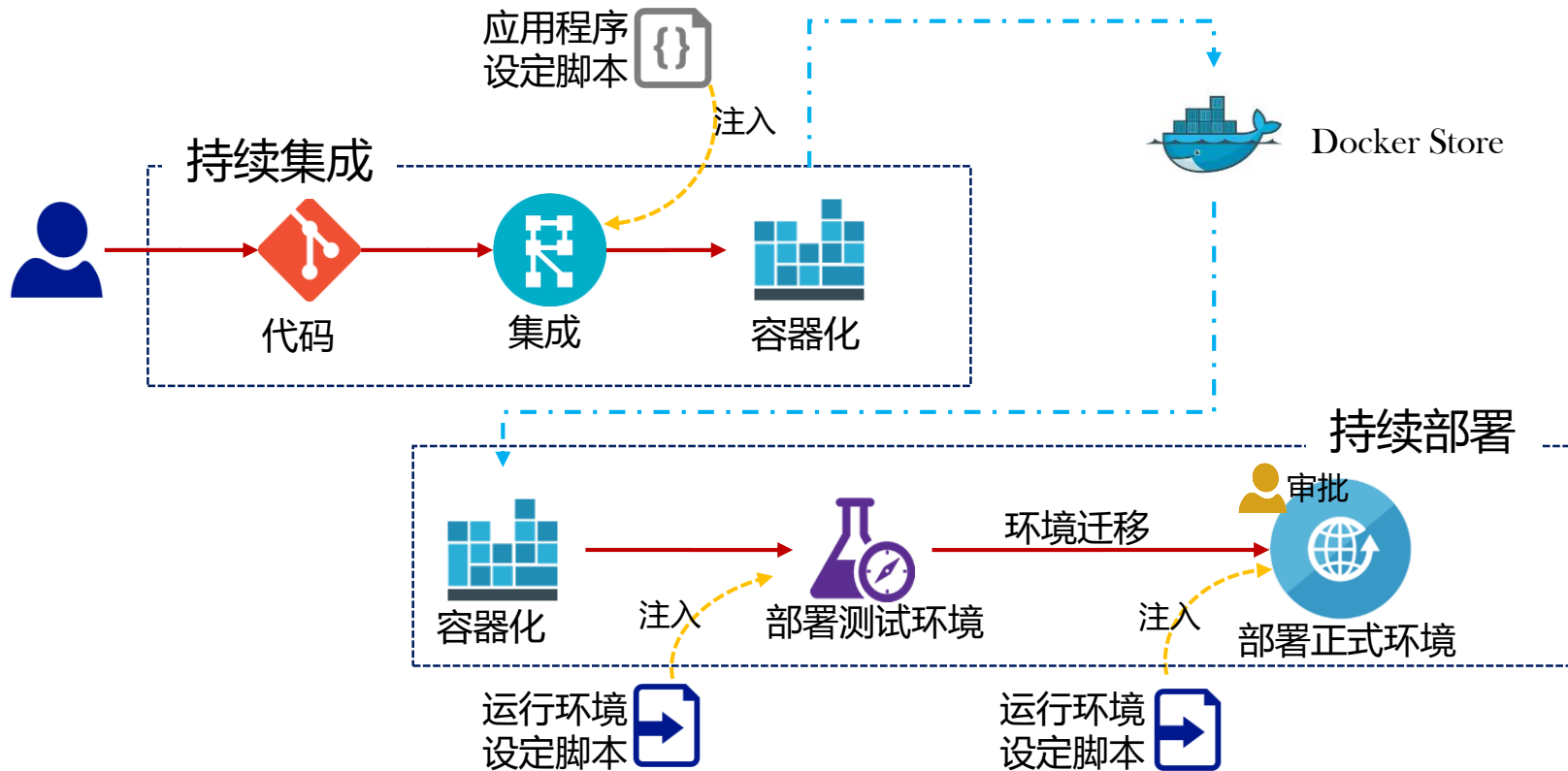
- 版本管控
- 移转、重复利用率高
- 具有更多的弹性
- 开发人员可以直接撰写流水线



## ➤导入容器化技术

- 减少程序部署造成的服务停止时间
- 确保测试环境与上线环境设定是标准化
- 降低系统内部的耦合程度
- 可以快速发布新功能

# 容器化的持续集成



# 「监控」是DevOps重要的一环



## ➤制造业推行DevOps中，监控是最需要优先被导入

- 各种装置皆可取得运行问题实时反馈
- 开发与维运人员可以线上实时共同讨论解决方案
- 讯息必须公开透明

## ➤「监控」要點

- 系统任何错误信息皆须反馈
- 系统效能监控
- 节点进出信息必须监控



# 循序推进

从小团队开始推进、解决问题思路优先

# DevOps需持续迭代

DevOps没有唯一标准，找出合适团队方法





# End

我的微信



# Thanks

DevOps 时代社区 荣誉出品



想第一时间看到高效运维社区的  
最新动态吗?

