

怎样挑战DevOps的实践之旅

刘征 & 张乐

自我介绍



刘征

- Nutanix 高级架构师
- DevOps实践指南 译者
- DevOpsDays中国核心组织者、出品人
- EXIN DevOps Master 首批授权讲师
- EXIN DevOps Professional 官方授权讲师
- 致力于推广DevOps的理念和技术实践
- 喜欢跑步、开源软件和写博客



张乐

- 京东 研发工程效能部 资深解决方案架构师
- 前百度资深敏捷教练、DevOps与持续交付专家
- DevOpsDays中国核心组织者、出品人
- EXIN DevOps Master 官方授权讲师
- EXIN DevOps Professional 全球讲师最高分保持者
- 凤凰项目DevOps沙盘 官方授权教练
- 设计并发布 DevOps道法术器 立体化实施框架
- 设计并实现 端到端持续交付流水线 工具集成解决方案

目录

- 1 DevOps转型的难点和误区
- 2 启动DevOps之旅的正确方法
- 3 DevOps的能力成长模型
- 4 如何持续学习和持续改进
- 5 总结：模型、实践及有用的资源



DevOps is everywhere



DevOps实施的常见反馈：

- DevOps也许在互联网企业推进更合适，传统企业实施阻碍太大
- DevOps的变革成本很高，需要采购一大批工具、调整组织结构
- 其他公司的DevOps实施路径是什么？我们就模仿他们的做法吧



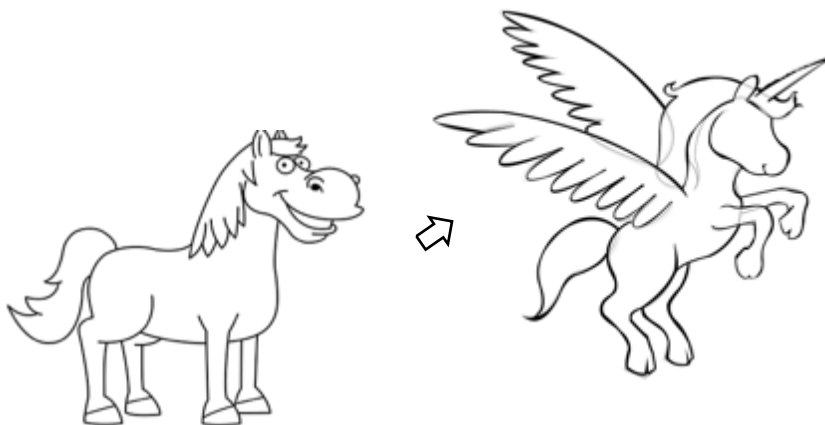
DevOps转型的难点和误区

1

DevOps也许在互联网企业推进更合适，传统类型企业并不适合？



- DevOps实践被互联网公司所倡导
- 谷歌、亚马逊、Facebook、Etsy、LinkedIn ...
- 百度、阿里、腾讯、京东 ...



- 用发展的、演进的思维看待问题
- 正是因为这些公司重构了系统架构、技术实践，改变了文化的氛围，才能走向成功
- DevOps已被证明是普适的方法和技术
 - 银行、保险、电信、零售、甚至政府
 - Web、APP、嵌入式、甚至大型机

美国CapitalOne银行转型数据	2016	2017
# Products deploying multiple times a day	~20	~300
Average #deployments per day	~1	~4
Max #deployments for a product in a single day	~30	~50



DevOps转型的难点和误区

3

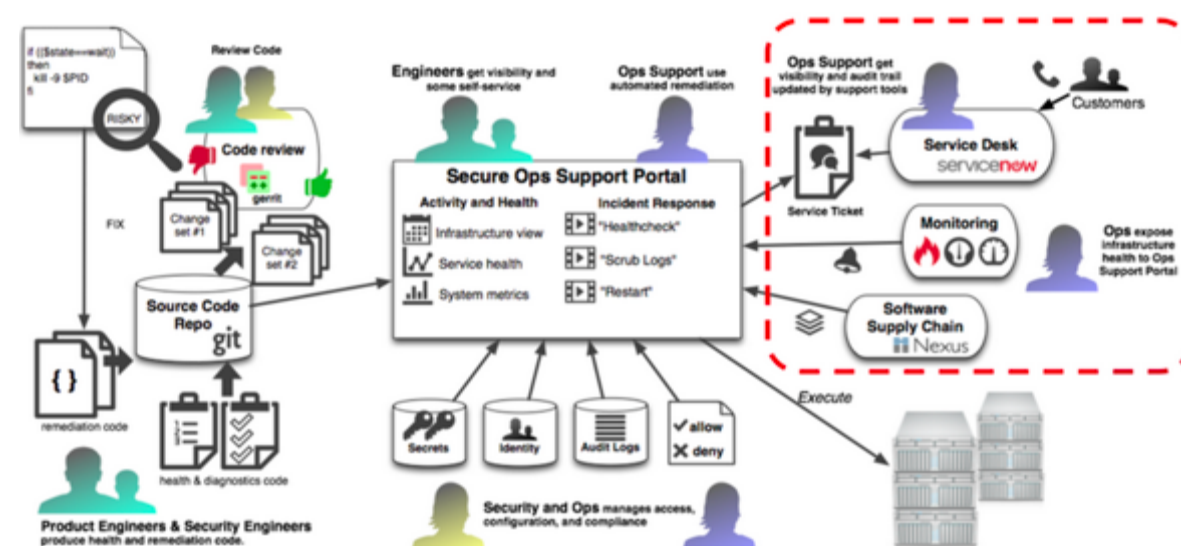
DevOps需要大规模组织结构重组，否则无法实施？



- **职能导向的团队**：强调专业性和集中管理，等级式的组织结构，频繁工作移交、效率低
- **市场导向的团队**：强调快速响应用户需求，跨职能团队，可独立快速交付用户价值

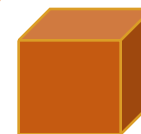


- 但组织结构重组（Re-Org）并不是唯一选择
- 可以让不同角色物理地工作在一起，提升协作效率
- 下游职能团队通过自动化的自服务平台，给上游赋能



DevOps转型的难点和误区

4 在实施DevOps的道路上，未来的那些坑可以预测和避免么？



不可见的复杂系统

Heisenberg Uncertainty Principle



Werner
Heisenberg

"One cannot simultaneously determine both the position and momentum of an electron."

The more certain you are about where the electron is, the less certain you can be about where it is going.

The more certain you are about where the electron is going, the less certain you can be about where it is.

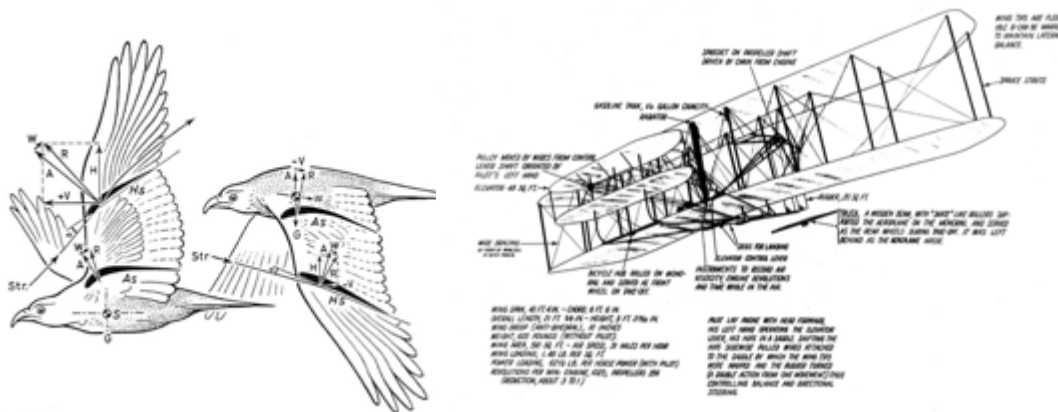
5

其他公司的DevOps实施路径是什么？我们就模仿他们来做吧？



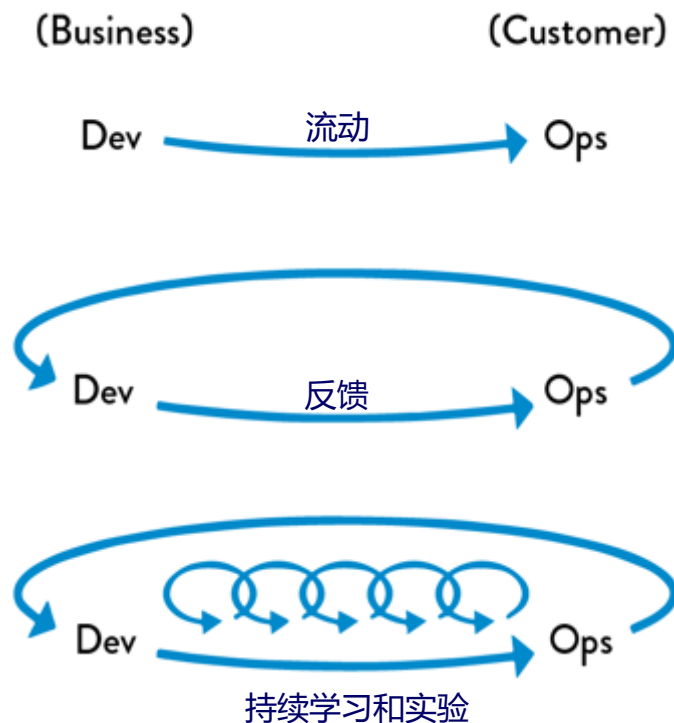
- 鸟飞派 vs 空气动力学派
- 初期的模仿是可以的，但很难照搬其他企业的做法而直接获得成功

- 需要深入理解DevOps实施路径背后的原理、原则和实践，从正确的方向入手
- 可以从模仿开始，但最终要超越模仿



启动DevOps之旅的正确方法

遵循的心法和原则：DevOps的三步工作法

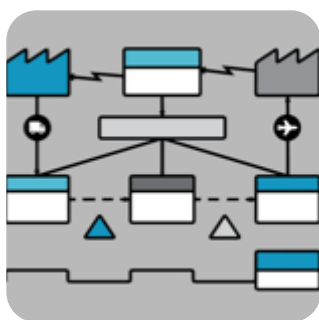


来源：DevOps 实践指南，第一章



启动DevOps之旅的正确方法

方法一：《DevOps实践指南》中的推荐的循序渐进的路径



选择价值流

- 绿地/棕地？
- SOR/SOE？
- 从最赞同和创新的组开始
- 逐步扩大范围



可视化价值流

- 识别相关角色
- 绘制价值流
- 组建专门转型团队
- 配套必要的工具



组建组织和架构

- 组建市场导向团队
- 招募通才团队成员
- 合理设计团队边界
- 保证松耦合和安全性



运维与开发融合

- 建立提高开发效率的自助服务
- 将运维融入开发日常工作
- 在团队中配置运维人员

启动DevOps之旅的正确方法

方法二：《精益思想》中推荐的行动计划



1. 寻找变革代理人



6. 创建精益促进机构

7. ^{KAI}改善^{ZEN} = CONTINUAL IMPROVEMENT
_{CHANGE} _{GOOD}



2. 获取精益知识



5. 根据价值流图重组公司



8. 利用政策为实施保驾护航



3. 寻找危机-险中求胜

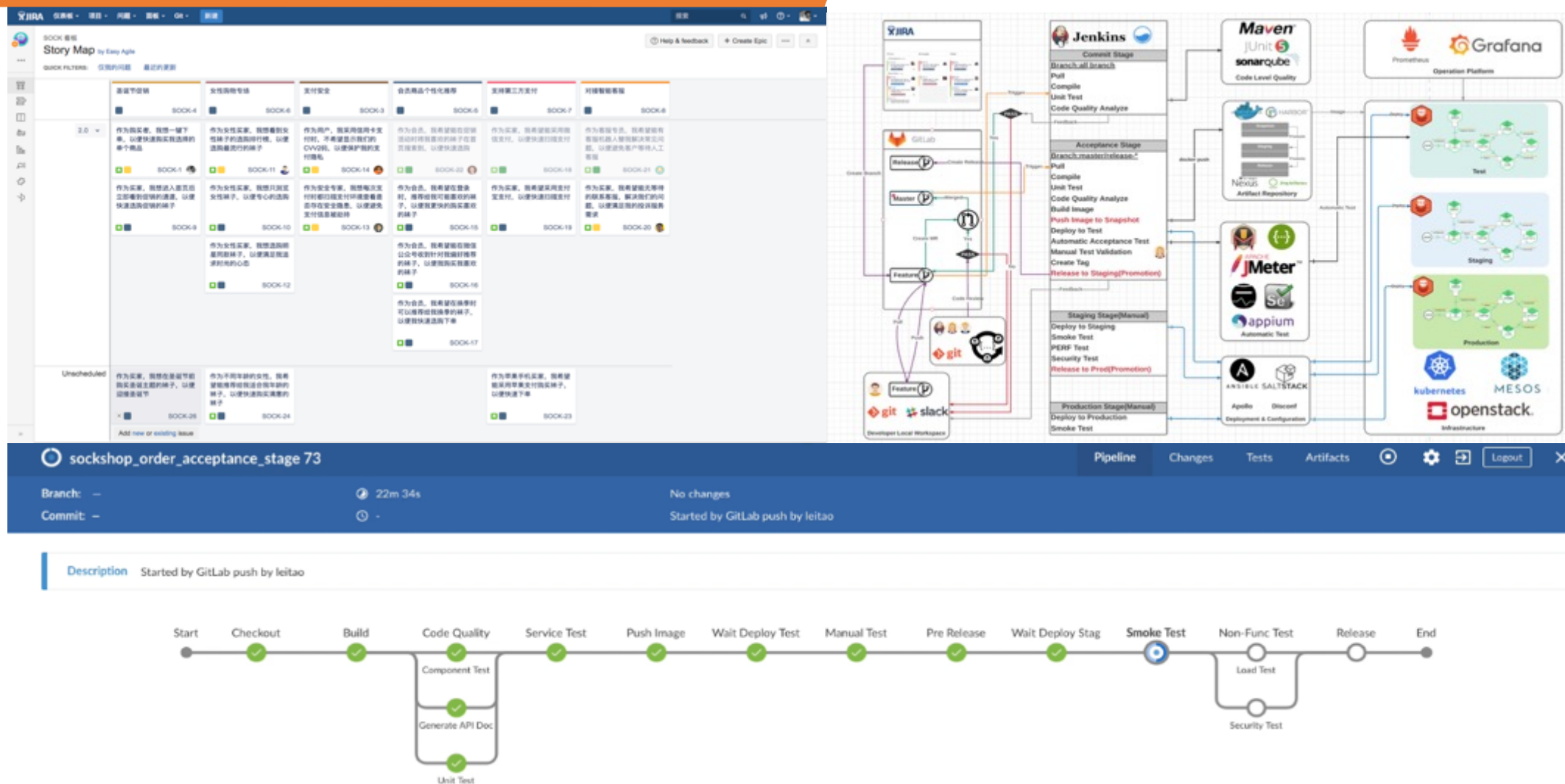


4. 绘制产品的价值流图



9. 同化你的供应商和客户

案例：管理维度 + 工程维度的结合





案例：美国全国保险公司的DevOps登山指南

- 所有开发团队实施前置时间
- 授权团队自行选择路径
- 朝每日、零宕机部署努力



登山装备

New Relic
urban(code)
GitHub
Jenkins
splunk
sonarqube
ROCKET.CHAT
Hand Book

- Handbook是地图
- 用地图规划路径
- 工具乃登山必备
- 工具皆各自有用

BASE CAMP 大本营

- 从运用敏捷开发开始
- 应用各种DevOps工具
- 成立DevOps Handbook读书俱乐部
- 秉持持续改进程序员体验的观念

NORTH CAMP 北坡营地

- 实施代码和数据的自动化部署
- 实施暗发布模式
- 在代码就绪/校验之后正式对外发布
- 推送流水线状态/健康度至Rocket.Chat
- 通过聊天机器人最小化上下文切换
- 发展多元化技能
- 实施去中心化的性能测试、安全测试和部署
- 实现生产环境的自动化校验和反馈
- 通过价值流映射的方式定位约束点
- 使用前置时间指标驱动持续改进

SUMMIT 顶峰

- 实现每日发布的频率
- 实施ChatOps
- 业务伙伴创建MMP并根据反馈指标而调整
- 实现单件流

SHERPAS 夏尔巴人

- 技术顾问
- DevOps教练

COMING SOON

- 沉浸式配对
- 每月DevOps道场
- 交付流水线支持

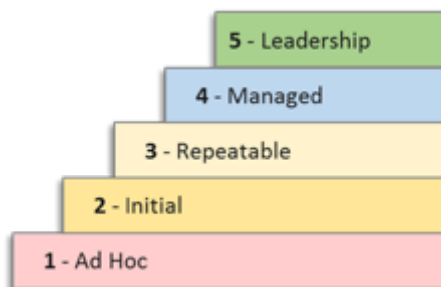
支持模型

DevOps的实施能力和效果如何评估？

DevOps的实施能力和效果应该如何评估？有没有通用的成熟度模型？

成熟度模型的特征：

- 阶梯式，统一的成熟度，所谓的标准？
- 定义技术、过程和组织能力的静态等级
- 组织的目标是“通过”某个级别



成熟度模型的局限：

- 企业环境、约束不同，级别定义都是相同的？
- IT行业每年都在发生着剧变，但模型不变化？
- 仅度量技术和工具本身，如何体现对结果影响？
- 达到了某级就会失去之前的资源和预算？
- 锚定和满足感阻挡了探索和尝试？
- **还有没有更好的方法？**

Maturity models are for CHUMPS

- 2005年，角色级别上限升至70，重新设置角色职业，增加新专业技能、副本、战场和种族，可以变更角色
- 2008年，角色级别上限升至80，增加死亡骑士职业，开放更多副本和新的装备
- 2010年，角色级别上限升至85，新增两个种族，游戏的内容在改头换面的艾泽拉斯大陆进行，增加新专业技能

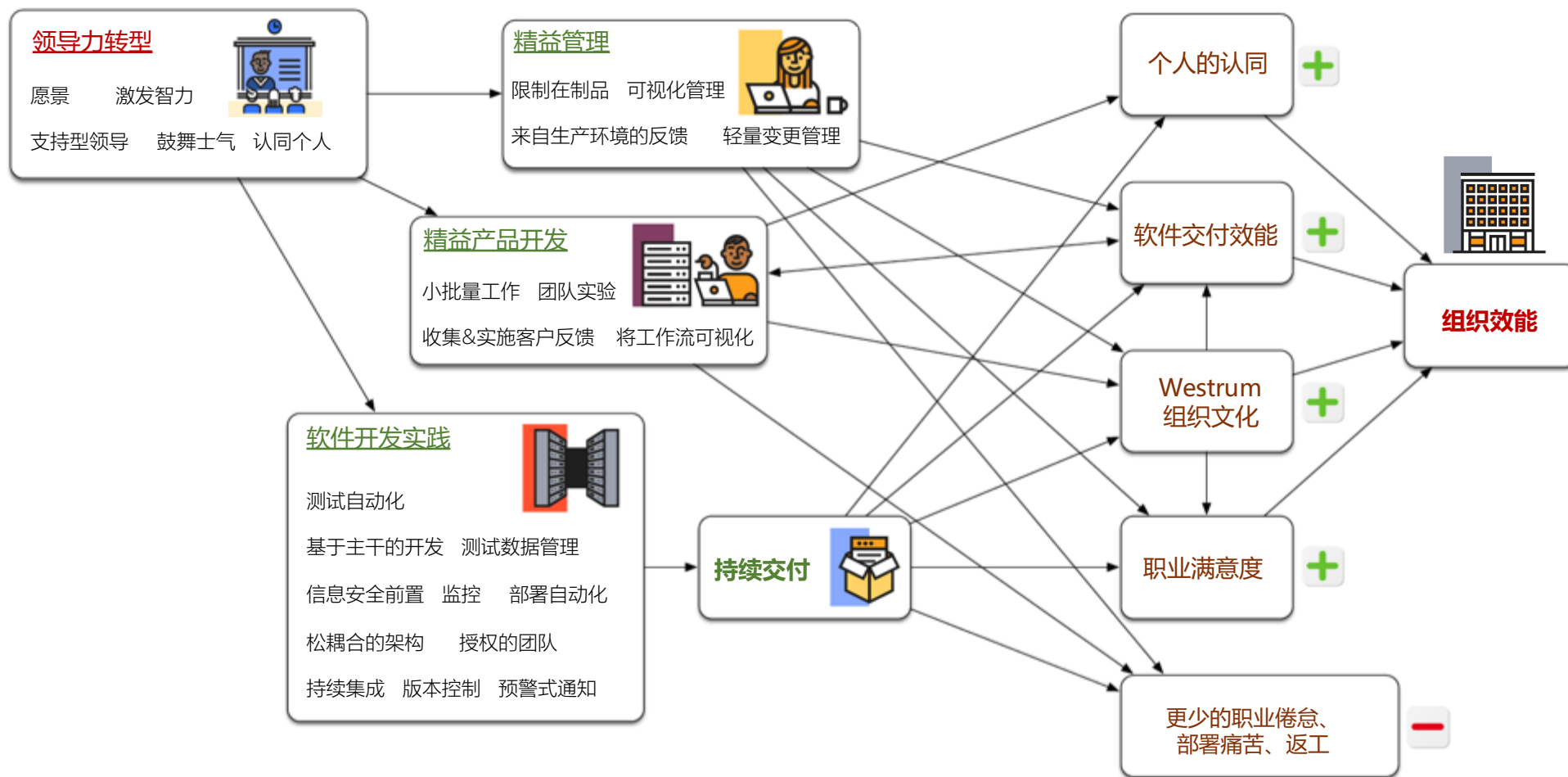


Maturity models are for CHUMPS

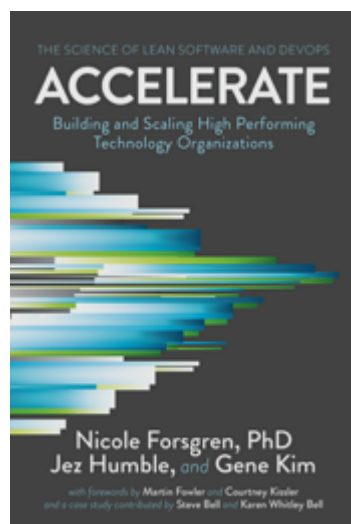
- 2011年，角色级别上限升至90
- 2013年，角色级别上限升至100，新增新的大陆德拉诺
- 2017年，角色级别上限升至120，部落和联盟接在海上找到了新的盟友

业务和技术环境每年都在变化

DevOps的能力成长模型



- 适用于本来能力和起点就参差不齐的不同团队
- 符合每个团队的优劣势和业务上下文的不同
- 每个团队聚焦在不同的、各自的约束点/弱点上
- 根据行业基线设置下一步的目标



Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations

Link: <http://a.co/eo5FyQj>



分类	DevOps能力
持续交付	1 对生产环境的所有工件进行自动化部署
	2 运用自动化部署流程
	3 实施持续集成
	4 运用基于主干的开发方法
	5 实施测试自动化
	6 进行测试数据管理
	7 前置信息安全管理
	8 实施持续交付
系统架构	9 应用松耦合的架构设计
	10 授权团队进行架构重构的决策
产品和流程	11 收集和实施客户反馈
	12 运用价值流模式可视化 workflows
	13 运用小批量的工作模式
	14 培养和赋能团队进行试验
精益管理和监控	15 应用轻量变更审批流程
	16 业务决策得到从应用到基础架构的全堆栈支持
	17 前瞻性地监控系统的状况
	18 应用WIP来进行价值流的工作管理
	19 通过可视化来监控团队工作质量和进行沟通
企业文化	20 发展并支持生机型企业文化
	21 鼓励和支持学习
	22 支持和辅助团队间的协作
	23 提供工作所需要的资源和工具
	24 支持和落实领导力转型

如何持续学习和持续改进

在复杂系统中实施DevOps仍然是很有挑战的，如何能够更快的进步？

- 实现飞行的关键技术是升力、动力和控制
- 莱特兄弟进行上千次的风洞实验，不急于试飞
- 思维的飞跃：摆脱模仿，从原理出发进行发明



- DevOps实施过程的实验和测试非常重要
- 需要一整套持续改进的机制或方法
- 建立实验的组织文化
 - Google的灾难恢复测试
 - Etsy的"三只袖毛衣"奖励
 - Netflix的Chaos Monkey和Simian Army



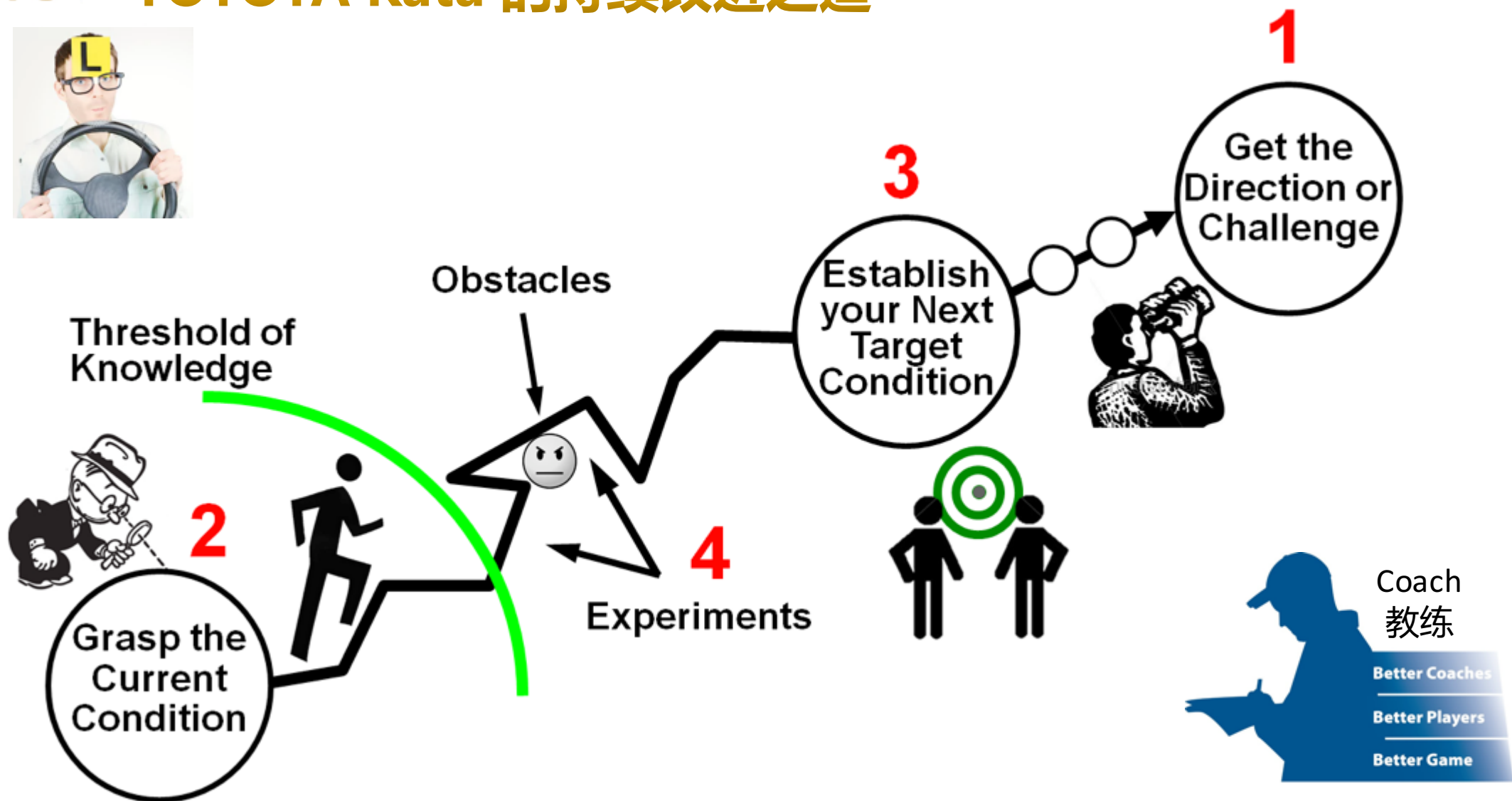


CHINA
DEVOPS
DAYS

Learner
学习者



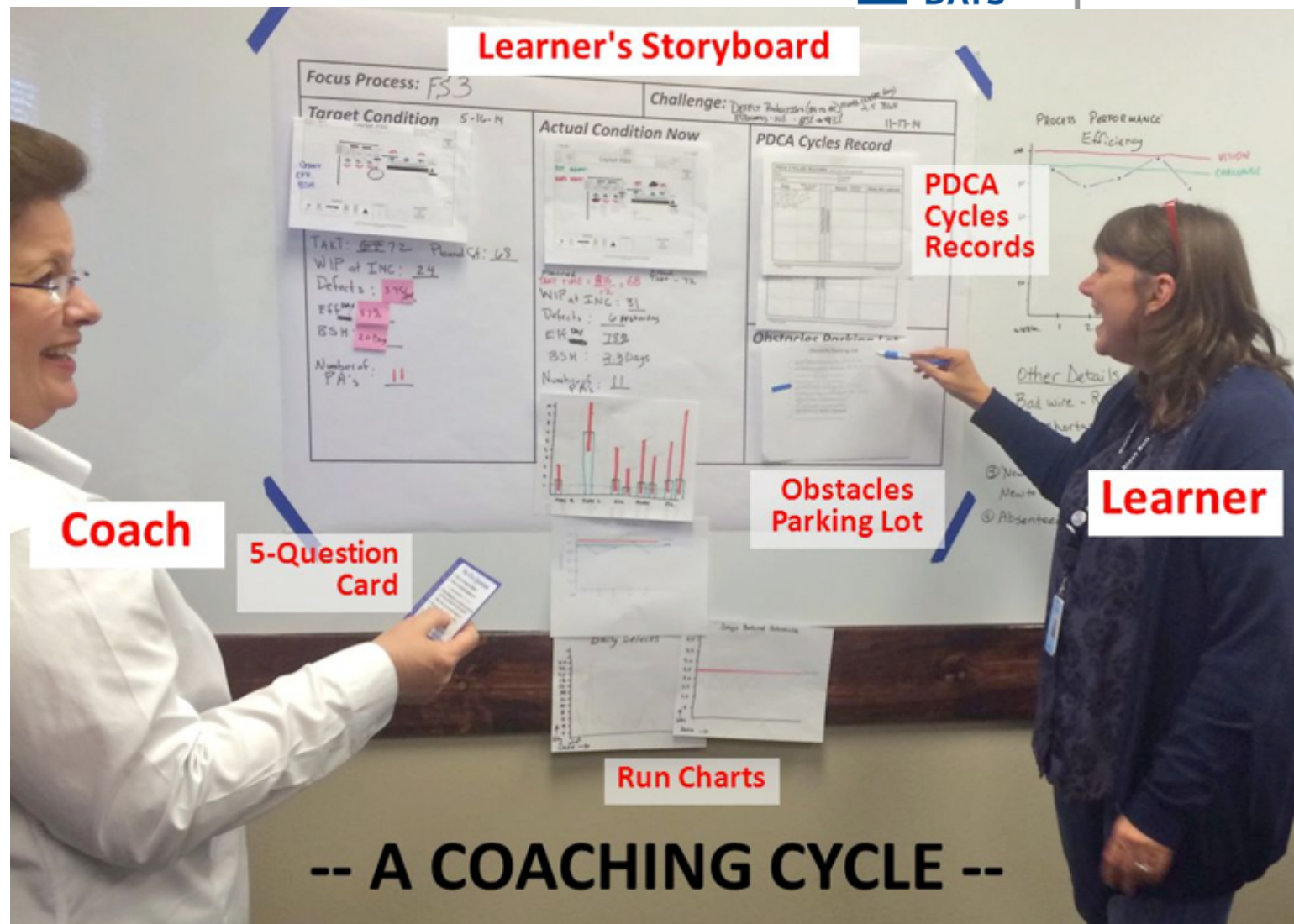
TOYOTA Kata 的持续改进之道



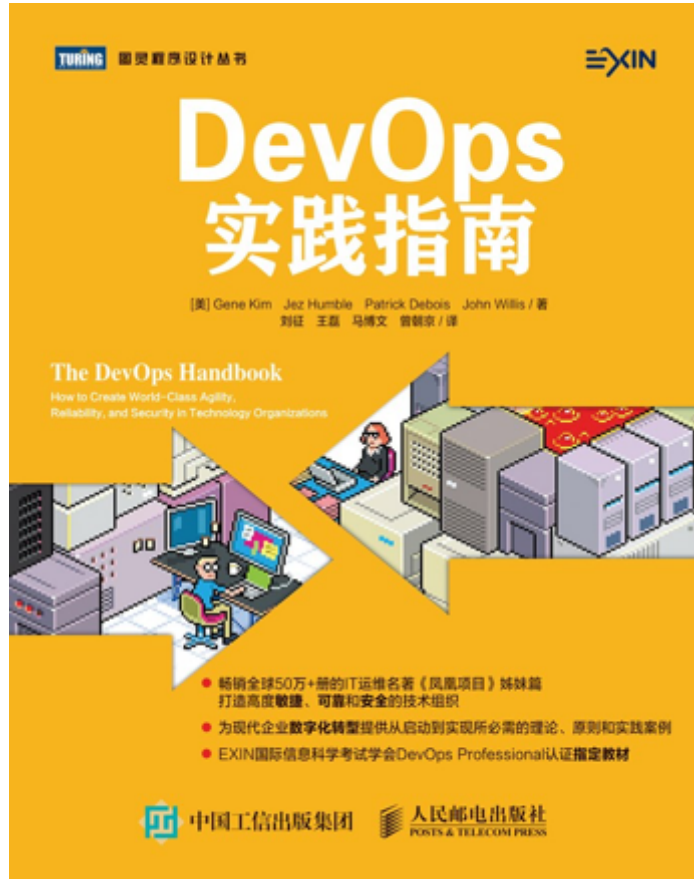
The Five Questions

- 1) What is the **Target Condition**?
- 2) What is the **Actual Condition** now?
-----*(Turn Card Over)*----->
- 3) What **Obstacles** do you think are preventing you from reaching the target condition?
Which ***one*** are you addressing now?
- 4) What is your **Next Step**? (next PDCA / experiment) What do you expect?
- 5) When can we go and see what we **Have Learned** from taking that step?

*You'll often work on the same obstacle for several PDCA cycles



总结：模型、实践及有用的资源



Part I 三步工作法

- 敏捷，持续交付与三步工作法
- 第一步：流动原则
- 第二步：反馈原则
- 第三步：持续学习与实验原则



Part II 从哪里开始入手

- 选择价值流
- 理解当前价值流工作，可视化，并在组织扩展
- 依据康威定律构建组织和架构
- 将运维嵌入研发日常工作环节，改善产出



Part III 工作法第一步：流水线原则

- 打造持续部署流水线的基础
- 快速可靠的自动化测试
- 启动和实践持续集成
- 自动化并确保低风险发布
- 低风险发布的架构



Part VI 集成安全，变更管理和合规的技术实践

- 信息安全嵌入日常工作
- 保护部署流水线，集成变更管理和安全合规性控制



Part V 工作法第三步：持续学习和实验原则

- 注入学习到日常工作
- 局部发现转变为组织改进
- 预留组织学习改进时间



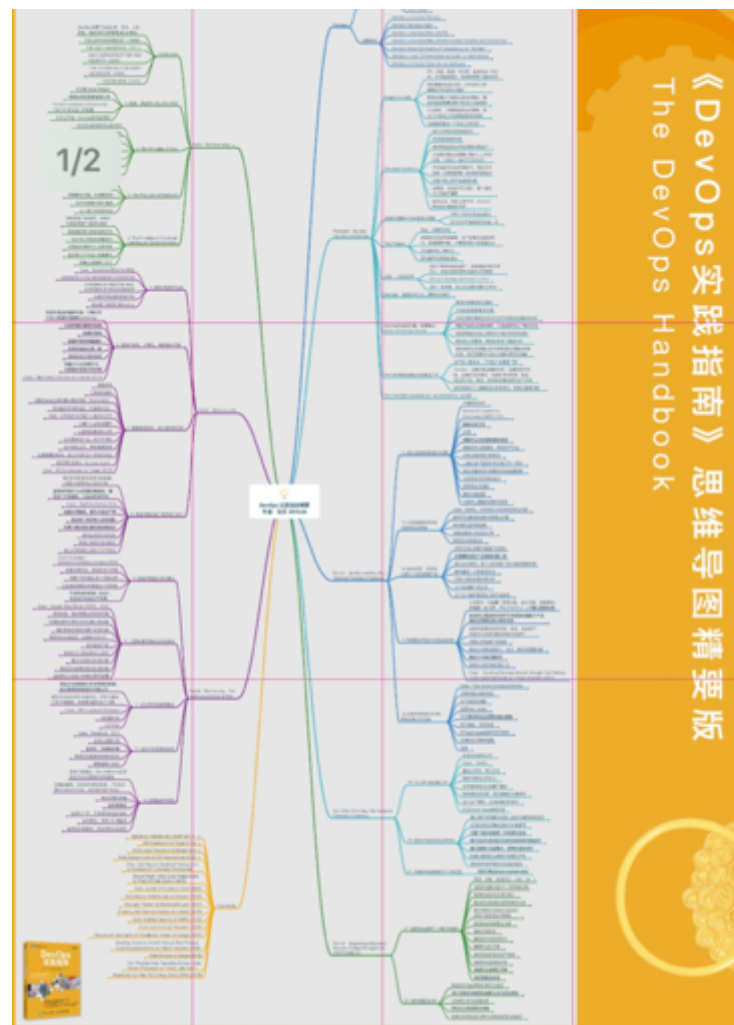
Part IV 工作法第二步：反馈原则

- 建立监控发现和解决问题
- 分析监控预测问题
- 开启反馈让开发和运维可安全部署代码
- 集成假设驱动的开发和AB测试到日常工作
- 建立Review和协调流程提升当前工作质量



CHINA
DEVOPS
DAYS

总结：模型、实践及有用的资源



欢迎大家参加：《DevOps实践指南》线上读书会

发起：DevOpsDays中国社区

组织者：刘征、张乐、许峰等，欢迎业内大咖加入

Jesse's Rule:

***Don't Fight Stupid,
Make More Awesome***



THANKS

