

如何有效突破 DevOps转型的临界点

张 乐

高效运维社区 合伙人

前百度资深敏捷/DevOps专家

自我介绍

ThoughtWorks®

IT大咖说
知识分享平台

张 乐

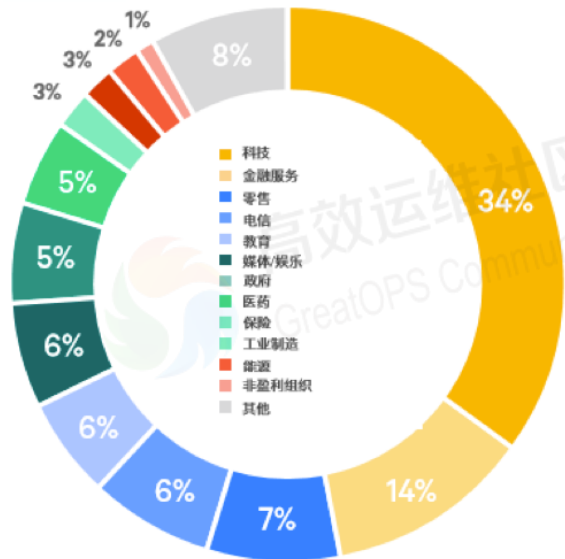
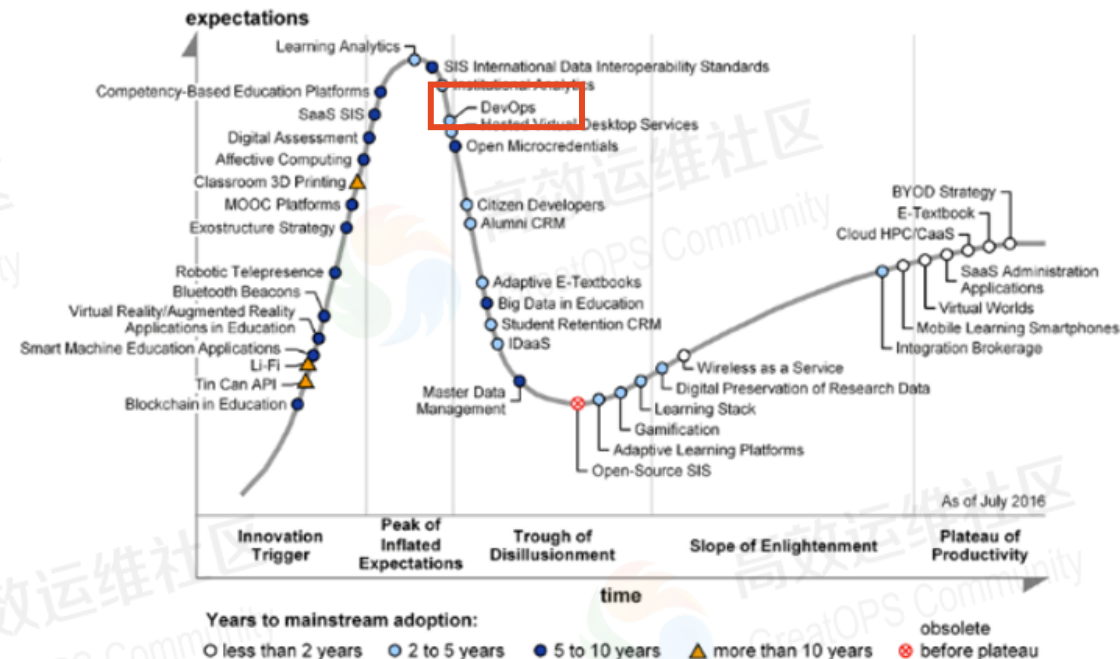
- 高效运维社区合伙人、DevOps时代联合创始人
- 前百度资深敏捷教练、DevOps专家
- 国内首批 Certified DevOps Master
- DevOps Master 授权讲师，凤凰项目沙盘授权教练
- 全球TOP外企，国内一线互联网



- 百度云、百度金融等新技术产品敏捷转型主导者
- DevOpsDays 大会、GOPS全球运维大会金牌讲师



DevOps 已成为发展趋势



DevOps团队比例2014年16%，2015年19%，2016年22%，2017年27%

IT 的技术革新

Development Process

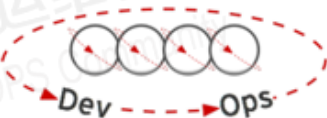
Waterfall



Agile



DevOps



Application Architecture

Monolithic



N-Tier

Microservices



Deployment & Packaging

Physical Servers



Virtual Servers

Containers



Application Infrastructure

Datacenter



Hosted

Cloud



推进 DevOps 转型过程中经常遇到的问题

- 为什么需要DevOps转型？
- 如何明确阐述转型的价值？

- 项目紧急，没空做转型？
- 转型过程中团队成员不支持？
- 转型初期效果不明显怎么办？

方案设计

整体推广

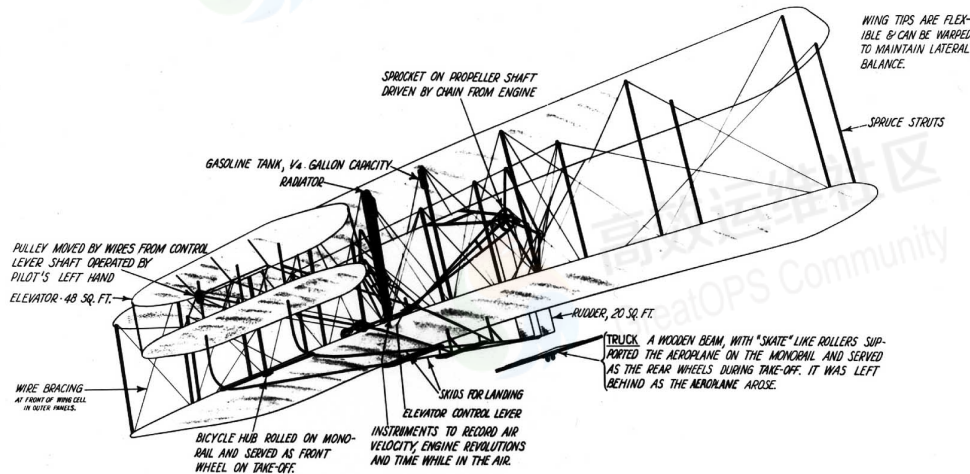
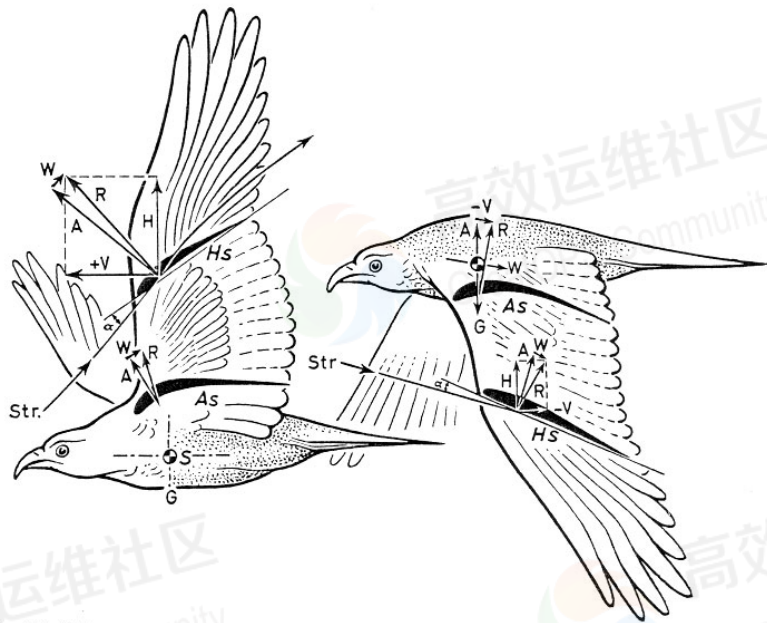
转型启动

项目试点

- 采购和引入工具是否足够？
- DevOps体系非常庞大，实施难度较大，应该从何处做起？

- 效率提升了，质量却下降了？
- 试点成果规模化的路径？
- 推广的组织结构是怎样的？

鸟飞派 VS 空气动力学派



WING SPAN, 40 FT. 4 IN. - CHORD, 6 FT. 6 IN.
OVERALL LENGTH, 21 FT. 3/8 IN. - HEIGHT, 9 FT. 3 1/2 IN.
WING DROOP (ANTI-DIHEDRAL), 10 INCHES
WEIGHT, 605 POUNDS (WITHOUT PILOT)
WING AREA, 510 SQ. FT. - AIR SPEED, 31 MILES PER HOUR
WING LOADING, 1.46 LB. PER SQ. FT.
POWER LOADING, 62 1/2 LB. PER HORSE POWER (WITH PILOT)
REVOLUTIONS PER MIN. ENGINE, 1023; PROPELLERS 396
(REDUCTION, ABOUT 3 TO 1)

PILOT LAY PRONE WITH HEAD FORWARD, HIS LEFT HAND OPERATING THE ELEVATOR LEVER, HIS HIPS IN A SADDLE SHIFTING THE HIPS SIDEWISE PULLED WIRES ATTACHED TO THE SADDLE BY WHICH THE WING TIPS WERE WARPED AND THE RUDDER TURNED (A DOUBLE ACTION FROM ONE MOVEMENT) THUS CONTROLLING BALANCE AND DIRECTIONAL STEERING.



应用底层规律，理解问题实质并解决问题

查理·芒格



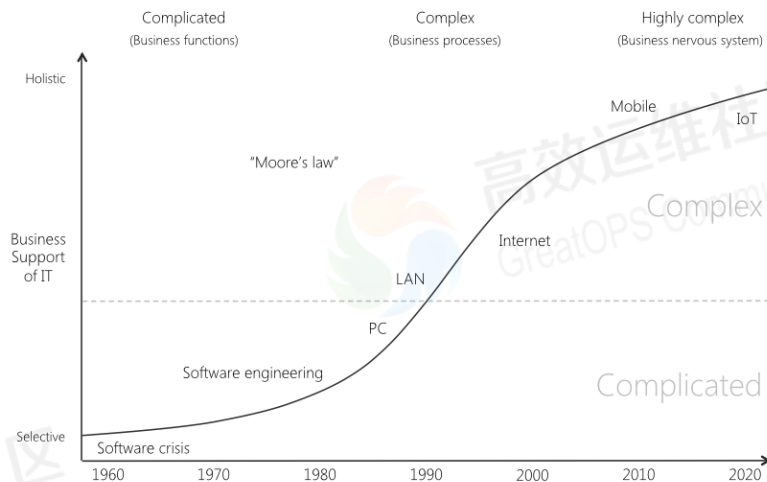
『如果你只是孤立地记住一些事物，试图把它们硬凑起来，那你无法真正理解任何事情……你必须依靠模型组成的框架来安排你的经验。』



为什么需要DevOps 转型？

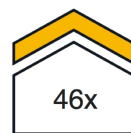
进化论

想要在变化的世界里生存得更好，你得在环境发生变化的时候快速响应

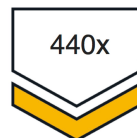


图片来源：《DevOps is not enough》@ufried

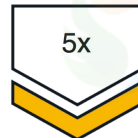
- 传统的软件工程方法
 - 系统的应用工程方法
 - 基于预测性
 - 重管控、结构化
 - 有时避免或拒绝合理变更
- 新的方法应对VUCA挑战
 - 面向业务目标
 - 助力业务成功
 - **快速交付价值**
 - **灵活响应变化**



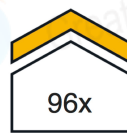
部署频率高



交付时间缩短



变更失败率降低



故障恢复时间

如何明确阐述转型的价值？

DevOps 需要面向业务目标，助力业务成功

Survey questions

High IT performers

Medium IT performers

Low IT performers

部署频率

针对主要应用和服务，你的组织多久部署一次代码？

On demand
(multiple deploys per day)

Between once per week
and once per month

Between once per week
and once per month*

变更前置时间

针对主要应用和服务，变更的前置时间是多长（从代码提交到代码在生产环境中运行成功的时间）？

Less than one hour

Between one week
and one month

Between one week
and one month*

故障恢复时间（MTTR）

针对主要应用和服务，如果发生服务故障，一般多久能恢复服务（比如：计划外宕机，服务损害）？

Less than one hour

Less than one day

Between one day
and one week

变更失败率

针对主要应用和服务，变更结果是回滚或随后修复的比例是多少（比如：导致服务损害，服务中断，需要紧急修复，回滚，向前修复，补丁）？

0-15%

0-15%

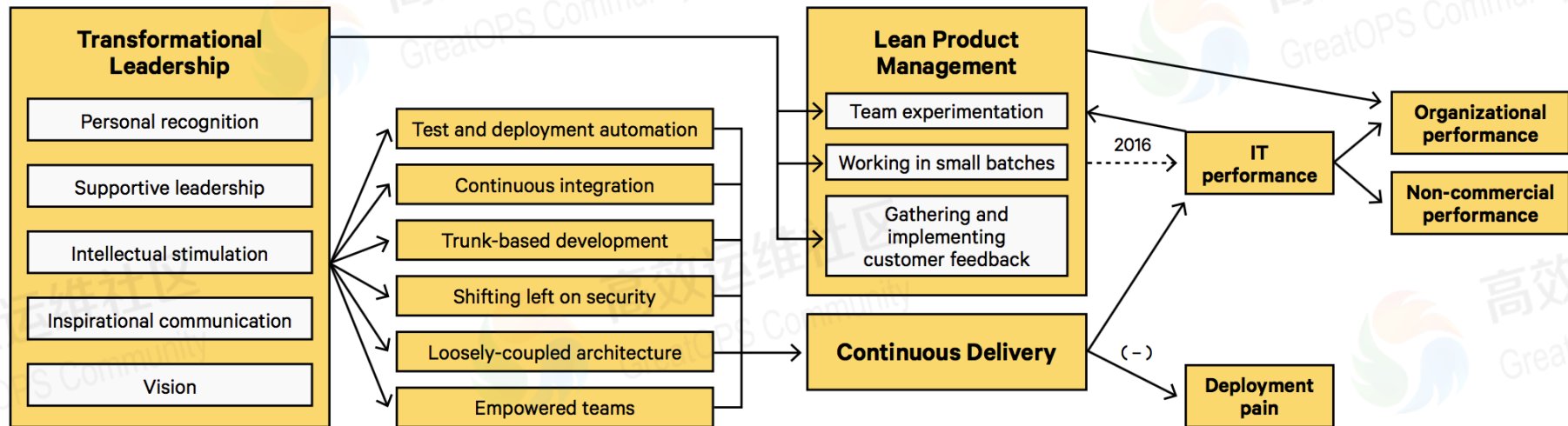
31-45%

采购和引入工具是否足够？

系统思考

系统不是简单的因果关系，而是回路网状关系
进行更深入的系统思考，需要从系统的互动关系入手

Figure 1. Structured equation model showing relationships between constructs



DevOps 道法术器

ThoughtWorks®

IT大咖说
知识分享平台

价值观，对目标
价值的定位

道

VALUE
快速交付价值，灵活响应变化

实现价值观的
战略、方法

法

WHY
全局打通敏捷开发 & 高效运维

战术、技术、
具体的手段

术

HOW
系统应用指导原则、最佳实践

用工具提高效率
复杂问题简单化

器

WHAT
端到端工具链相互联通和整合

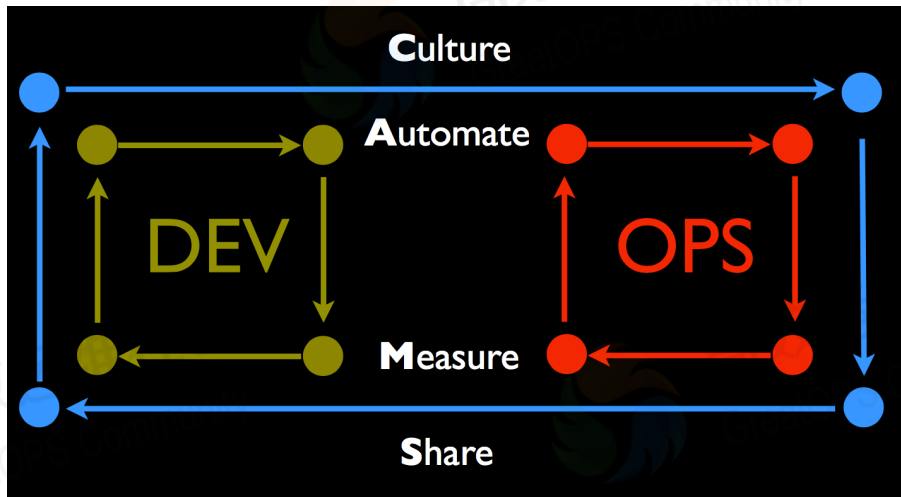
系统思考的层次

解决问题的层次

DevOps 之『道』

快速交付价值，灵活响应变化

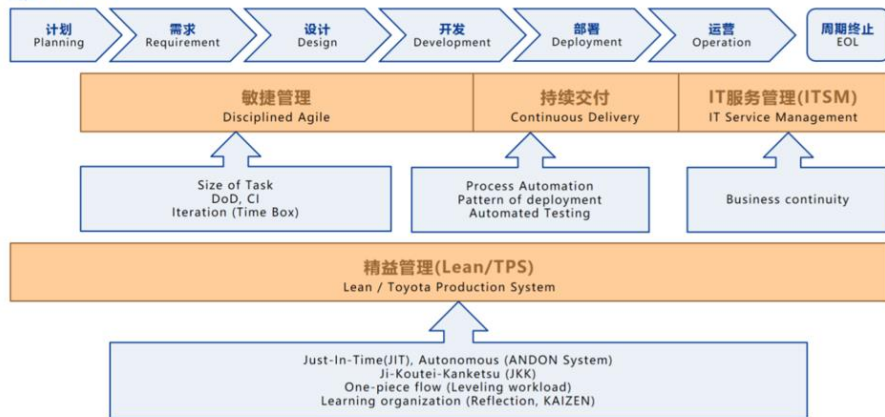
实现 IT 服务的供应链，快速、高质量交付业务价值
有效应对VUCA挑战，让 IT 成为业务发展的竞争优势



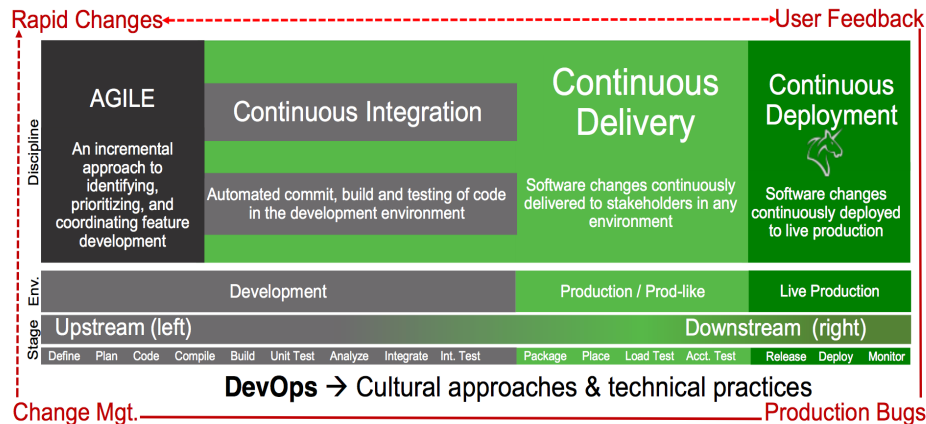
- **Automation:**
Extend delivery to Production (IaC)
- **Measurement:**
Extend Operations feedback to Project (Metrics)
- **Culture :**
Embedded Project knowledges into Operations
(Dev wear pagers, Co-responsible)
- **Sharing:**
Embedded Operations knowledge into Projects
(Monitoring, security stories into project backlog)

DevOps 之『法』

流程 (Process)



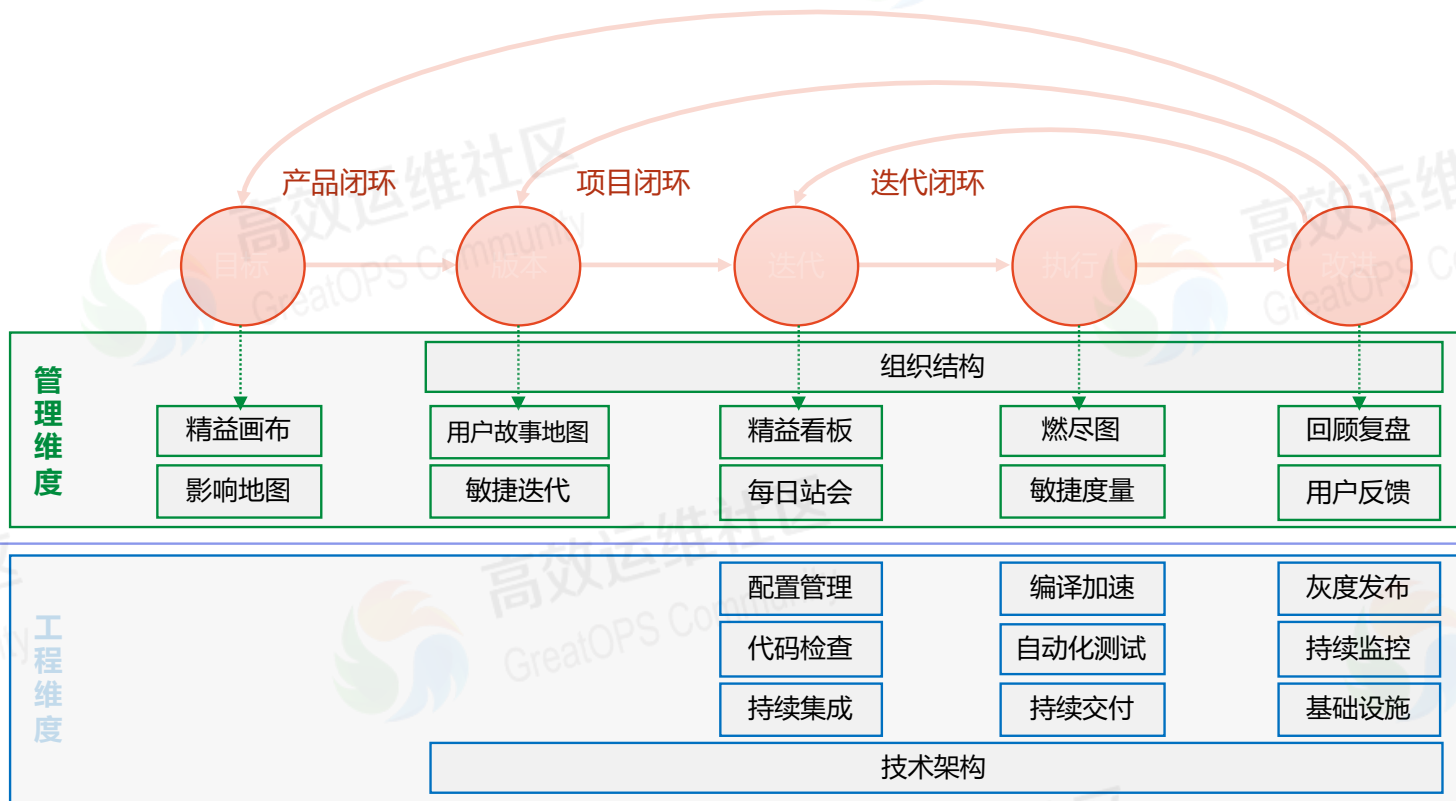
Reference : 《DevOps Master Whitepaper: Success with Enterprise DevOps》

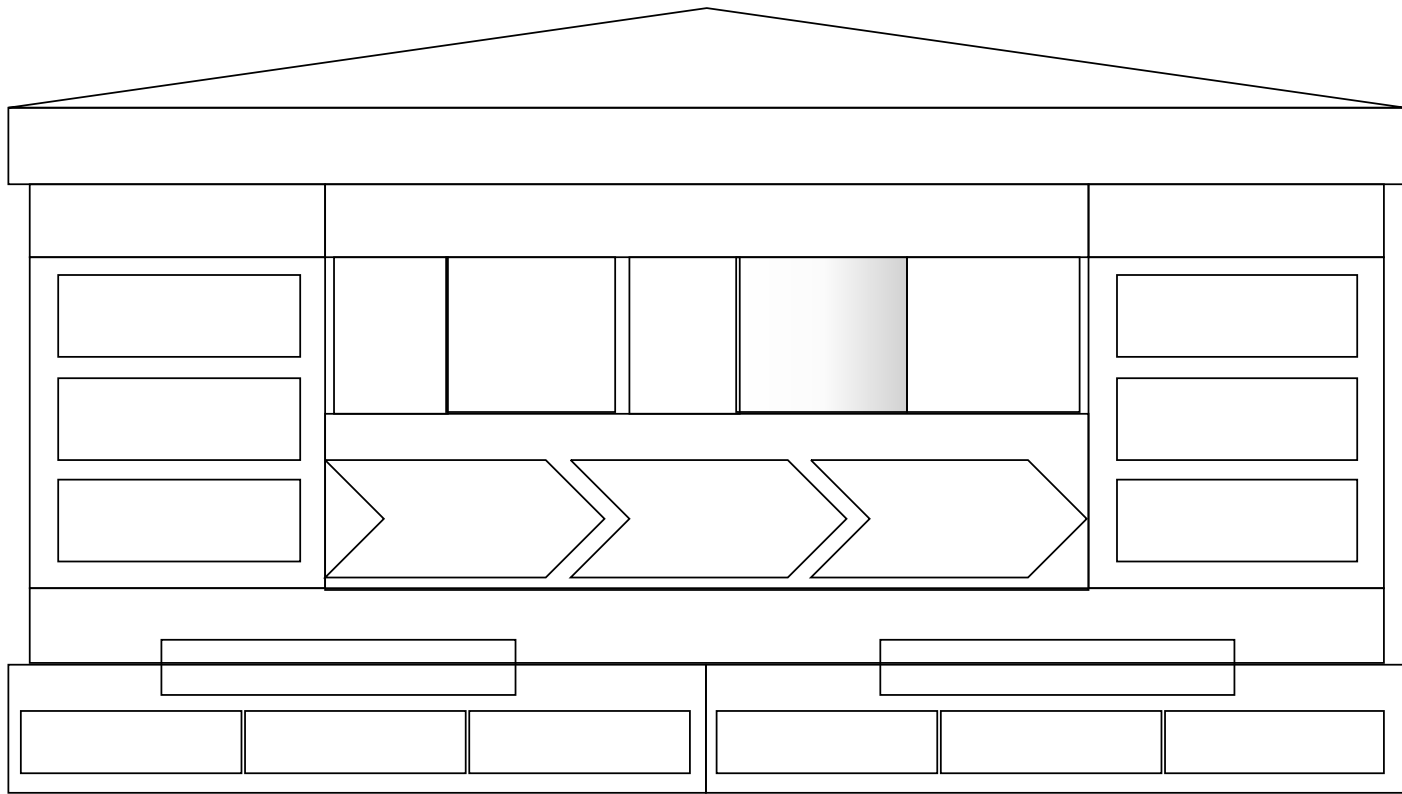


Reference : KK 《Why, What, and How of Continuous Delivery》

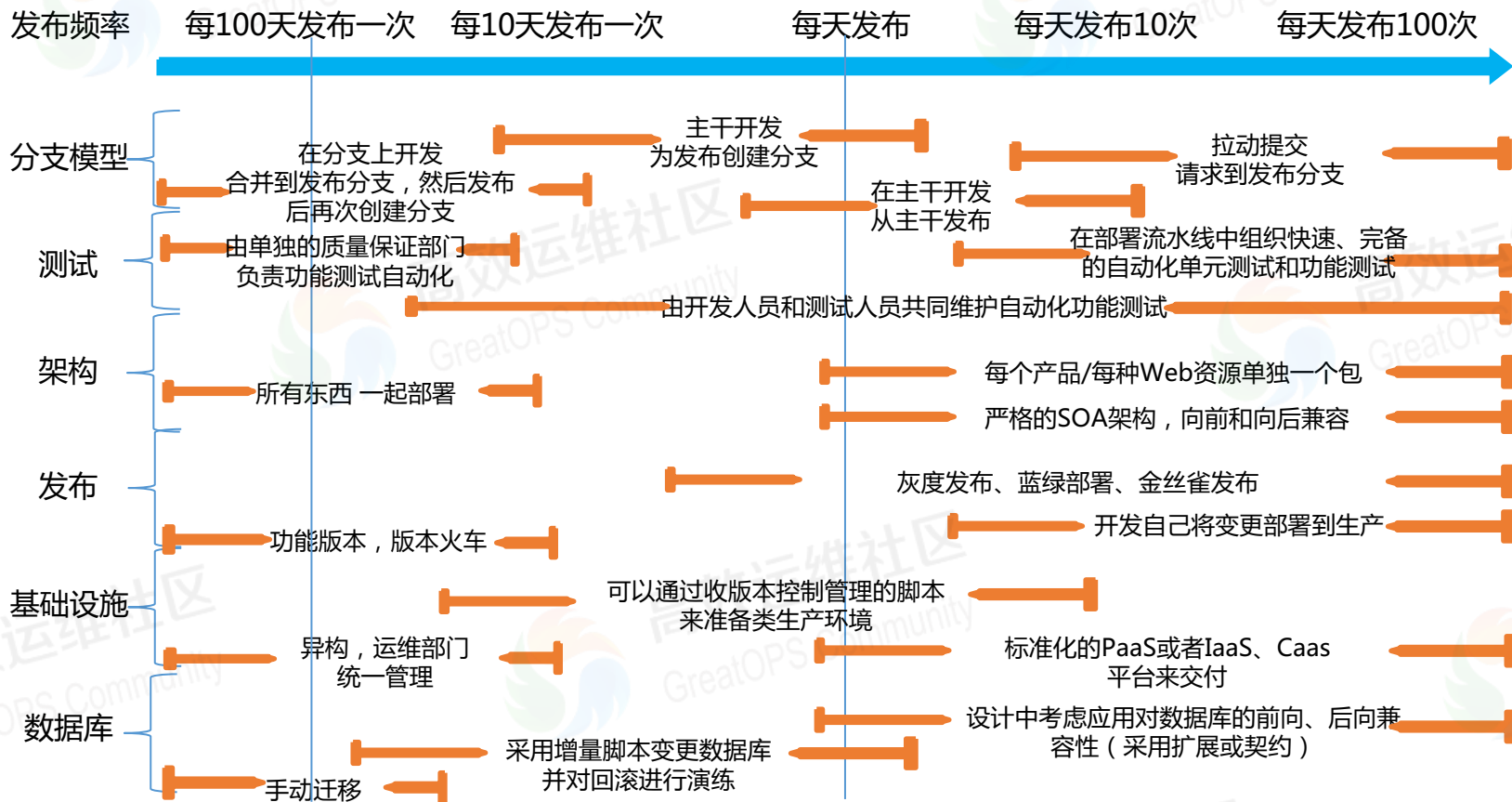
DevOps 之『术』

系统应用指导原则、最佳实践



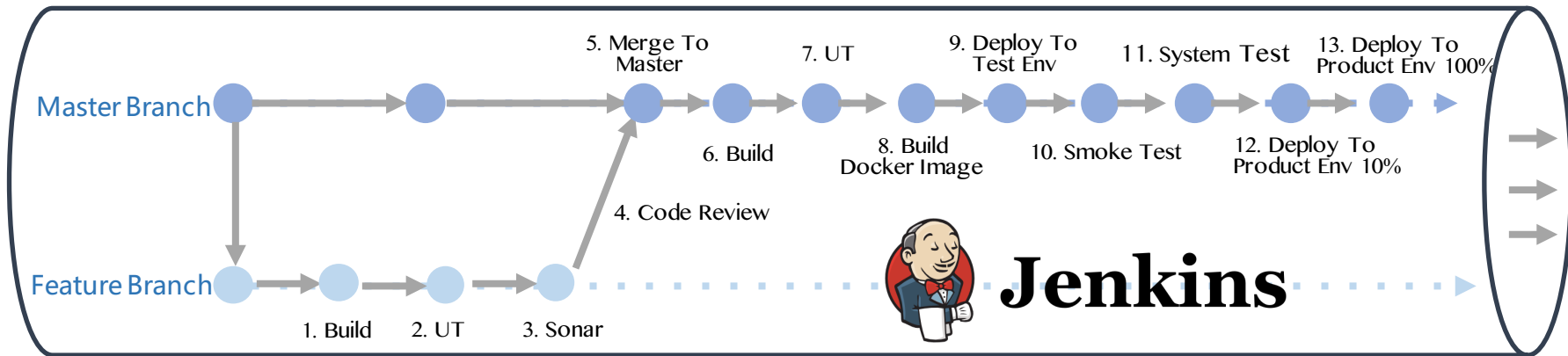


DevOps 之『术』 - 最佳实践集合



DevOps 之『器』

端到端工具链相互联通 & 整合



Maven™

JUnit



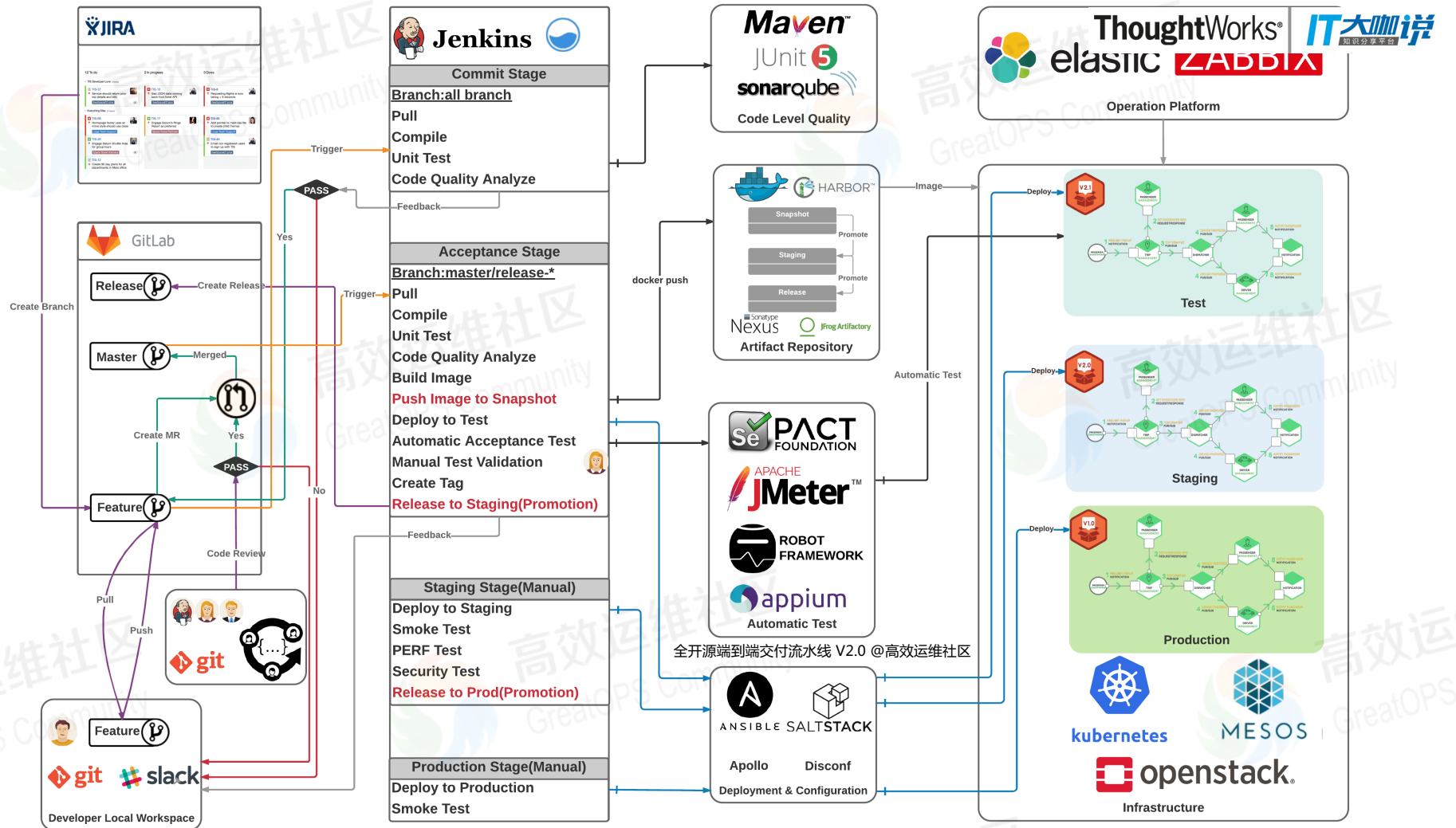
GitLab

Nexus

sonarqube



kubernetes
elastic



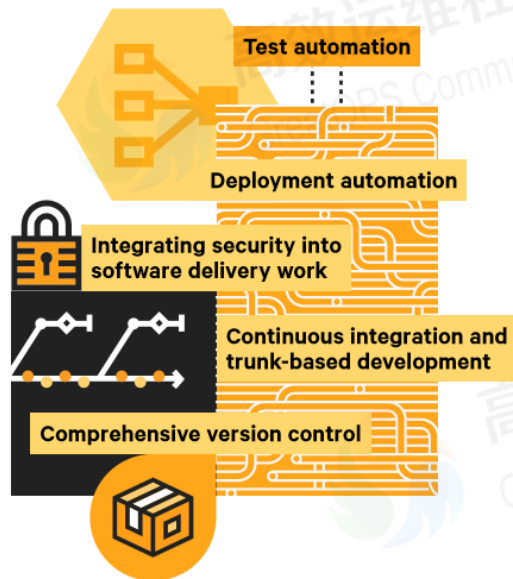
DevOps体系非常庞大，应该从何处做起？

二八法则

在资源有限的情况下，需要找到实现高绩效的突破口
把80%的资源，用到能产出关键效益的20%的工作上



Factors that positively contribute to continuous delivery:



自动化是高效能团队的显著特征

- 配置管理
- 自动化测试
- 自动化部署
- 变更审批流程

投入效果：在创新和快速反馈上能投入更多时间

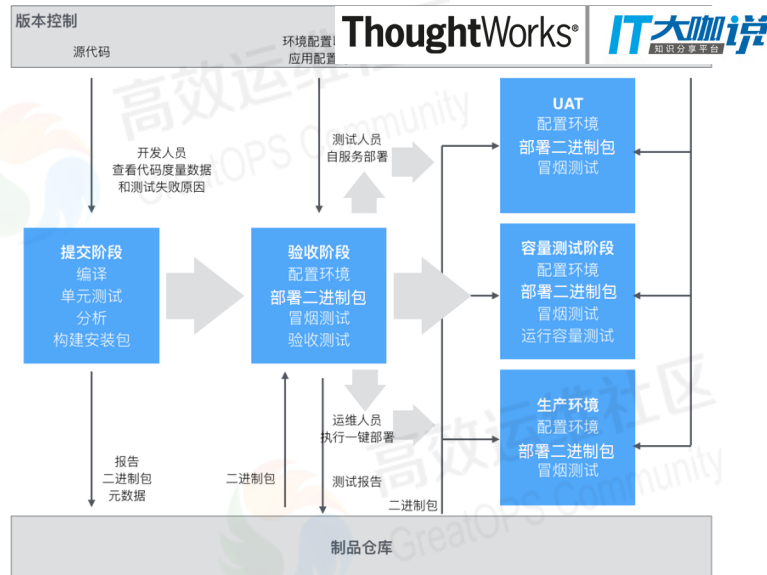


Table 3: Percentage of work that is done manually, by performance group.

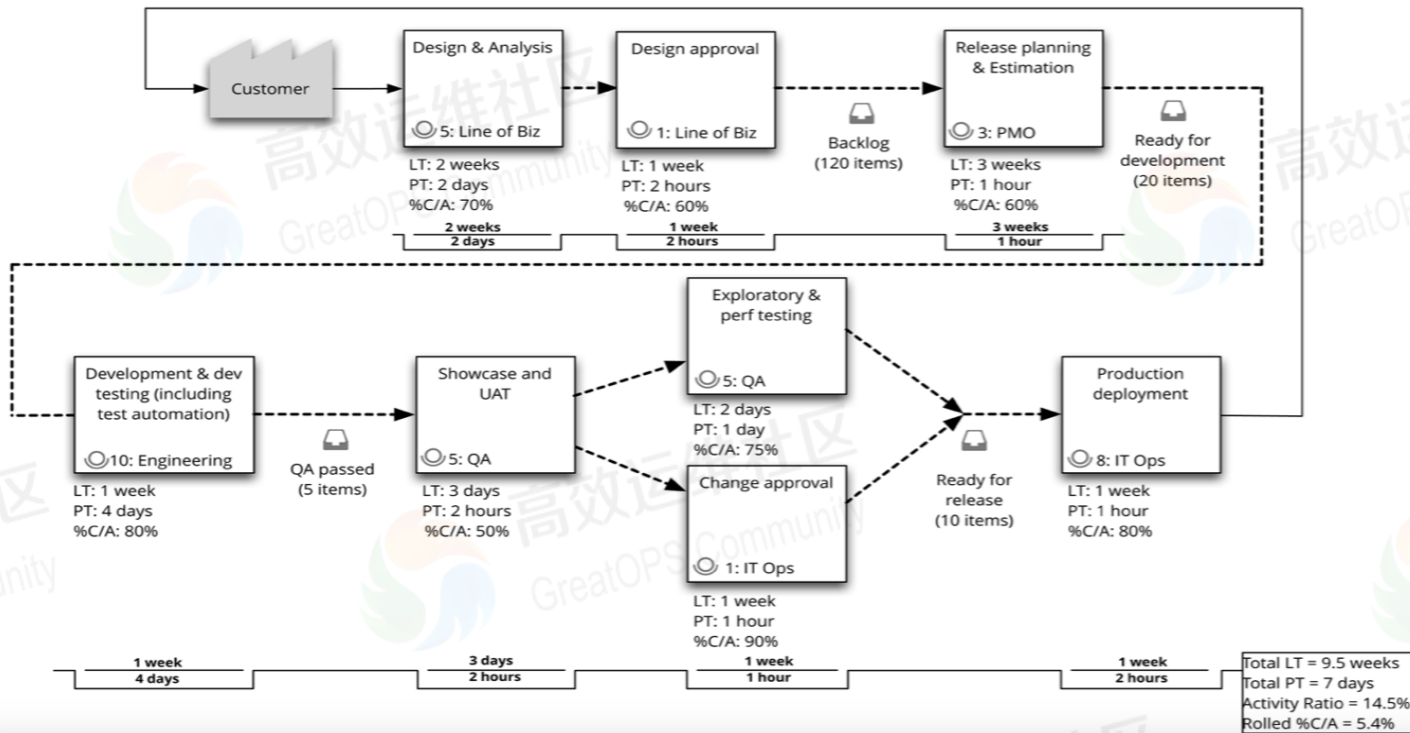
All percentages significantly different among High, Medium, and Low IT performers, except where otherwise noted.

	High performers	Medium performers	Low performers
Configuration management	28%	47% ^a	46% ^a
Testing	35%	51% ^b	49% ^b
Deployments	26%	47%	43%
Change approval processes	48%	67%	59%

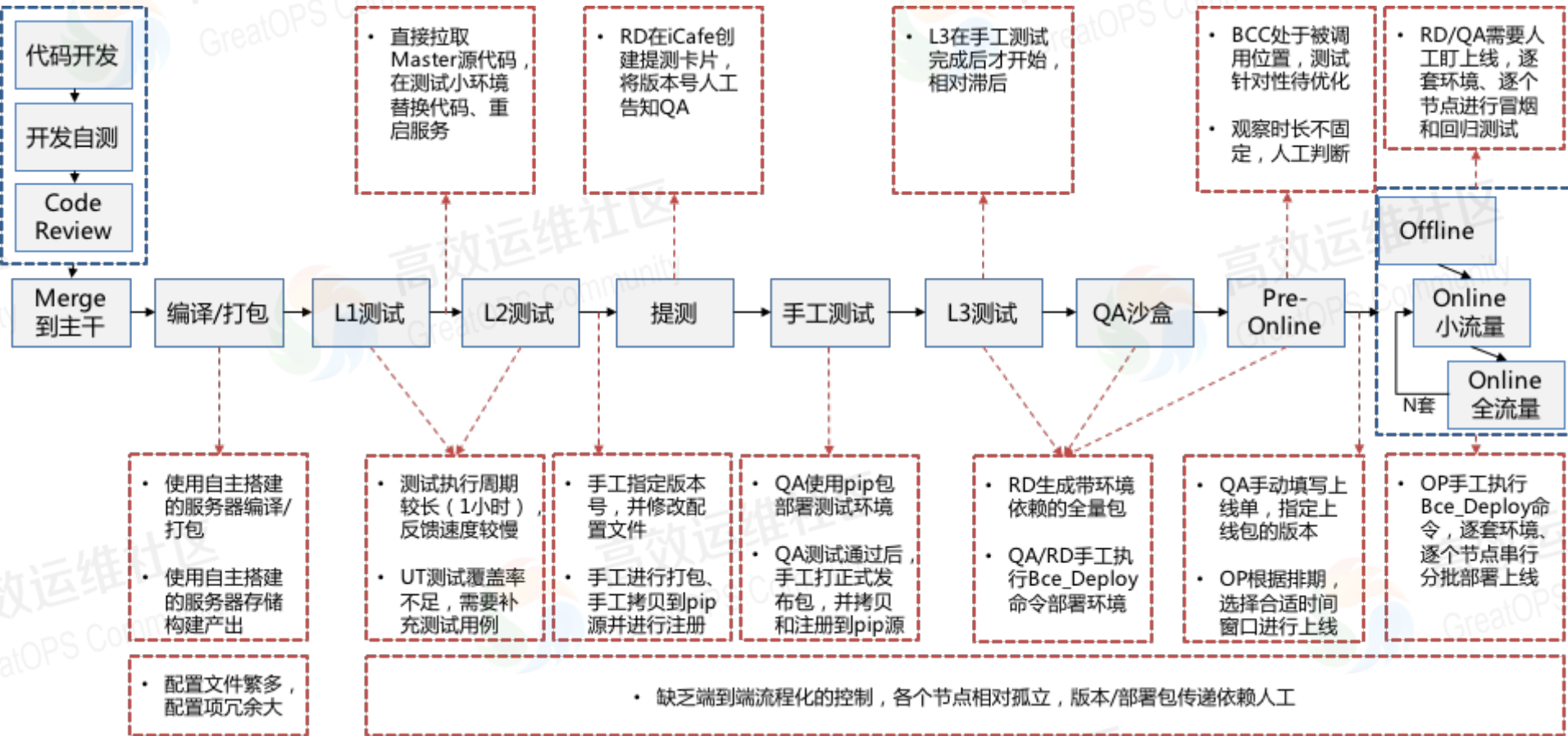
DevOps体系非常庞大，应该从何处做起？

约束理论

明确过程中影响绩效的瓶颈，努力来解决它们
聚焦于瓶颈的改善，达到系统各环节同步、整体改善的目标



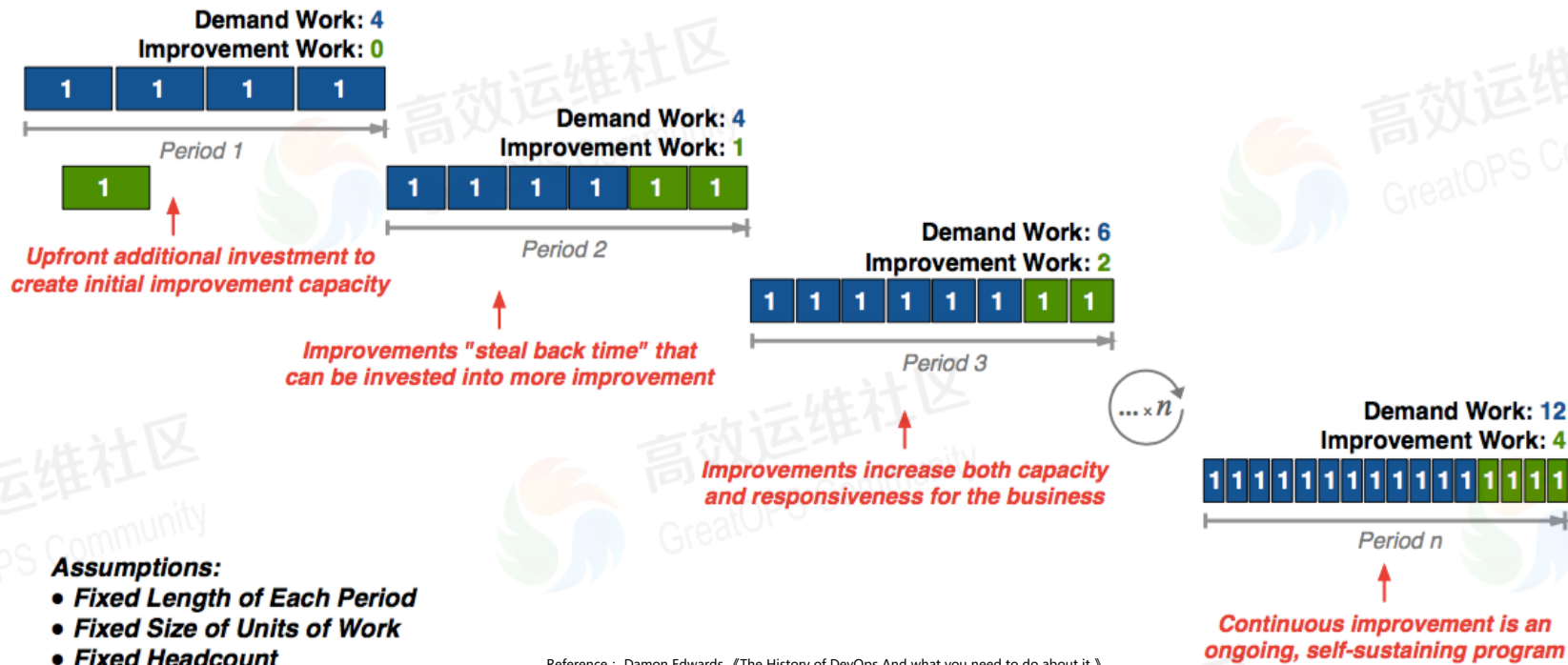
聚焦于瓶颈的改善



项目紧急，没空做转型？

复利效应

做事情A，会导致结果B，而结果B又会加强A，不断循环
需要长期坚持，会达到临界点，实现跨越式发展



转型过程中团队成员不支持？

黄金思维圈

深入思考WHY，抓住问题的关键
抓住WHY的本质，激发HOW的创意

《第五项修炼》：每个人不能只囿于自己的岗位，觉得只做好分内之职就可以了，要对职位之间相互关联产生的结果负有一定责任



- **责任共担**：鼓励合作的发生
- **自动化**：DevOps运动及促进合作的基石
- **内建质量**：频繁和低风险发布的基础
- **反馈**：为了实现持续改进
- **推倒部门墙**：早期参与、构建合作文化
- **自组织团队**：改变风险管理方式、建立对失败的宽容环境
- **营造DevOps文化**：奖励『改进冒险』的『行为』

转型初期效果不明显？

概率论

在不确定性的世界中，选择不断投入成功概率最大的事情中
从长期来看，一直投入最大赔率的事情，终会有回报

TABLE 2 Yearly Returns Possible from Cost of Unnecessary Rework Avoided

$$\text{Technical Staff Size} \times \text{Average Salary} \times \text{Benefits Multiplier} \times \text{Percentage of Time Spent on Unnecessary Rework} = \text{Cost of Unnecessary Rework Avoided per Year}$$

TABLE 4 Returns Possible from Cost of Downtime Avoided

$$\text{Deployment Frequency} \times \text{Change Fail Rate Percentage} \times \text{Mean Time To Restore (MTTR)} \times \text{Outage Cost} = \text{Cost of Downtime per Year}$$

TABLE 3 Potential Value Added from Reinvestment in New Features¹

$$\text{Time Recovered and Reinvested in New Features} \times \text{Revenue Generating Features} = \text{Potential Revenue from Reinvestment}$$

$$\left(\text{Frequency of Experiments per Line of Business} \right) \times \left(\text{Lines of Business in the Organization} \right) \times \left(\text{Idea Success Rate} \right) \times \left(\text{Idea Impact} \right) \times \left(\text{Product Business Size} \right)$$

ITEM	SPEND AMOUNT
Consulting: assessment and roadmap development for technology transformation initiative	\$200,000
Cloud subscription services	\$65,000
Automation software	\$1,000,000
SREs and DevOps engineers to augment team (5 x \$180,000 x 1.5 benefits multiplier ⁿ)	\$1,350,000
Training and DevOps/Kanban/agile coaching for teams	\$200,000
Dedicated time and resources of existing workforce (equivalent to 18 FTE x \$105,000 x 1.5 benefits multiplier)	\$2,835,000 ^o

RETURN

\$75.7M, Rounded

$$\text{ROI} = \frac{\text{Return} - \text{Investment}}{\text{Investment}}$$

$$= \frac{\$75,741,666.67 - \$5,600,000}{\$5,600,000}$$

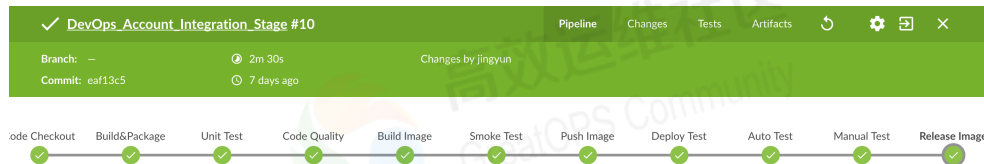
$$= 12.5$$

效率提升了，质量却下降了？

安全空间

小概率事件会造成极端严重的后果
构建安全空间，保证复利效应持续起作用

方案一：增强分级测试验证，减少出错概率



- 在追求速度提升的同时，不能以牺牲质量为代价
- 如果在内建质量上投入不足，会造成更多的故障

Table 1: Changes in IT performance of high performers, 2016 to 2017

IT performance metrics	2016	2017
部署频率	200x more frequent	46x more frequent
变更前置时间	2,555x faster	440x faster
故障恢复时间 (MTTR)	24x faster	96x faster
变更失败率	3x lower (1/3 as likely)	5x lower (1/5 as likely)

方案二：提升系统的反脆弱性，增强容错能力

Survey questions

故障恢复时间 (MTTR)

针对主要应用和服务，如果发生服务故障，一般多久能恢复服务（比如：计划外宕机，服务损害）？

High
IT performers

Medium
IT performers

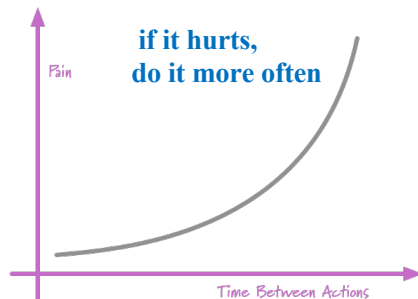
Low
IT performers

Less than one hour

Less than one day

Between one day
and one week

- 改进交付流水线的效率，能够快速进行故障恢复
- 主动发现弱点：『避免失败的最好办法是经常失败』



试点成果规模化的路径？

第一种路径

- 首先是从组织级认识到DevOps的重要性，并开始立项做相关的准备
- 然后在工作组级开始启动相关建设，当在团队级通过采用一些实践取得成功后，再把这些成果扩展到工作组级
- 最终向组织级做规模化的推广

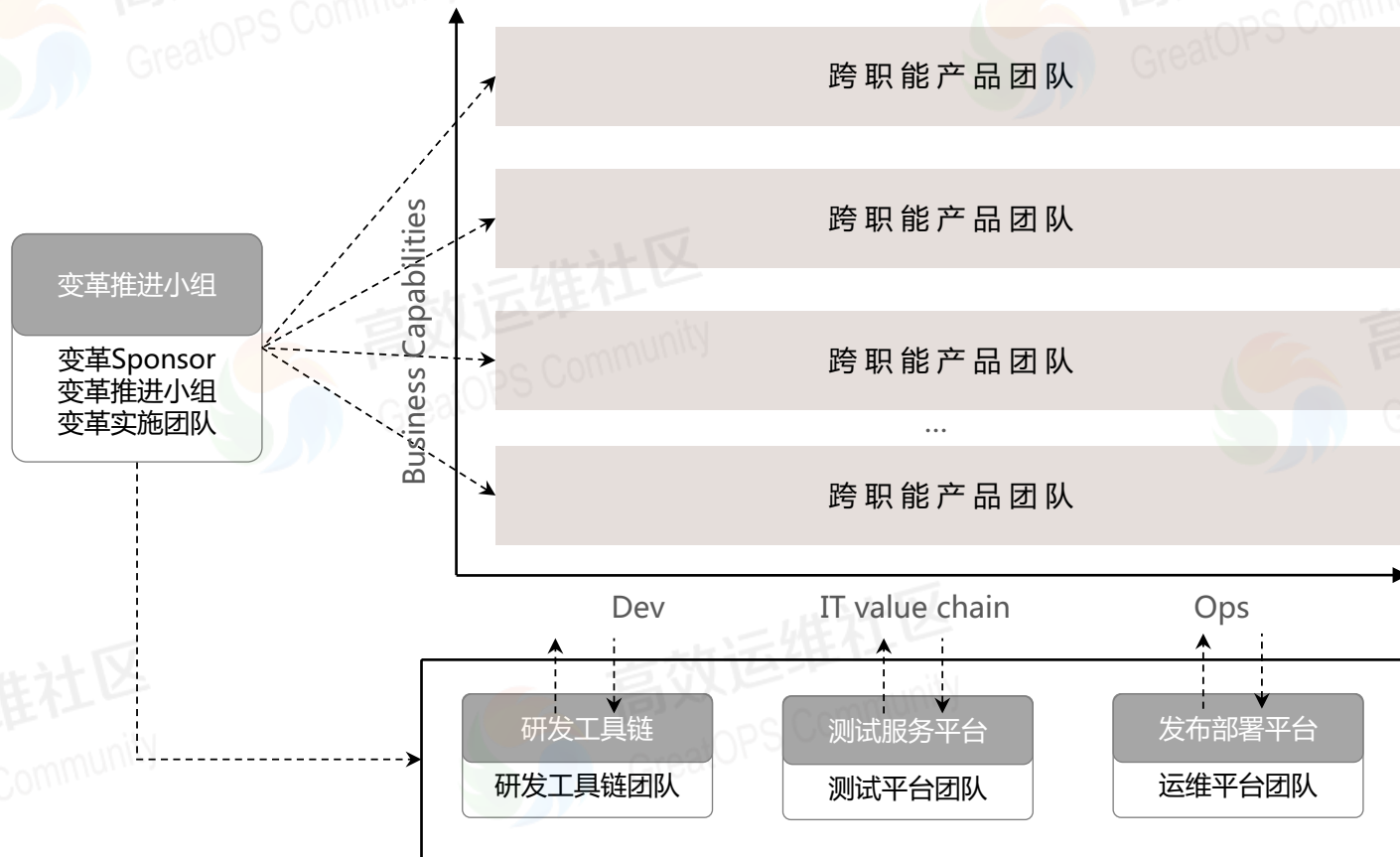


第二种路径

- 首先从团队级别做小范围的改进和建设，然后让领导层清晰的了解我们在做DevOps的尝试
- 直到在团队级通过采用一些实践取得成功后，再把这些成果扩展到工作组级
- 在获得管理层的支持后，最终向组织级做规模化的推广



推广的组织结构是怎样的？



- Overall Goal
- T-shaped
- Co-located

We choose to go to the Moon

We choose to go to the moon in this decade and do the other things, **not because they are easy, but because they are hard**, because that goal will serve to organize and measure the best of our energies and skills, **because that challenge is one that we are willing to accept, one we are unwilling to postpone, and one which we intend to win**, and the others, too.

U.S. President John F. Kennedy





Q & A

