



# DevOps中国调查研究

刘博涵



# CONTENTS

**01** DevOps简介

**02** 问卷调查研究

**03** 国内外现状及发展对比

**04** 实践建议及研究问题

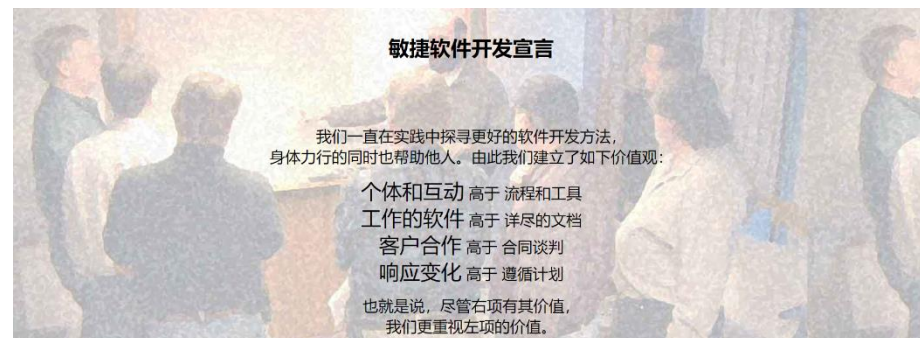
# CONTENTS

## 01 DevOps简介

- 什么是DevOps
- 为什么要DevOps

# 什么是DevOps--诞生

2001



2009

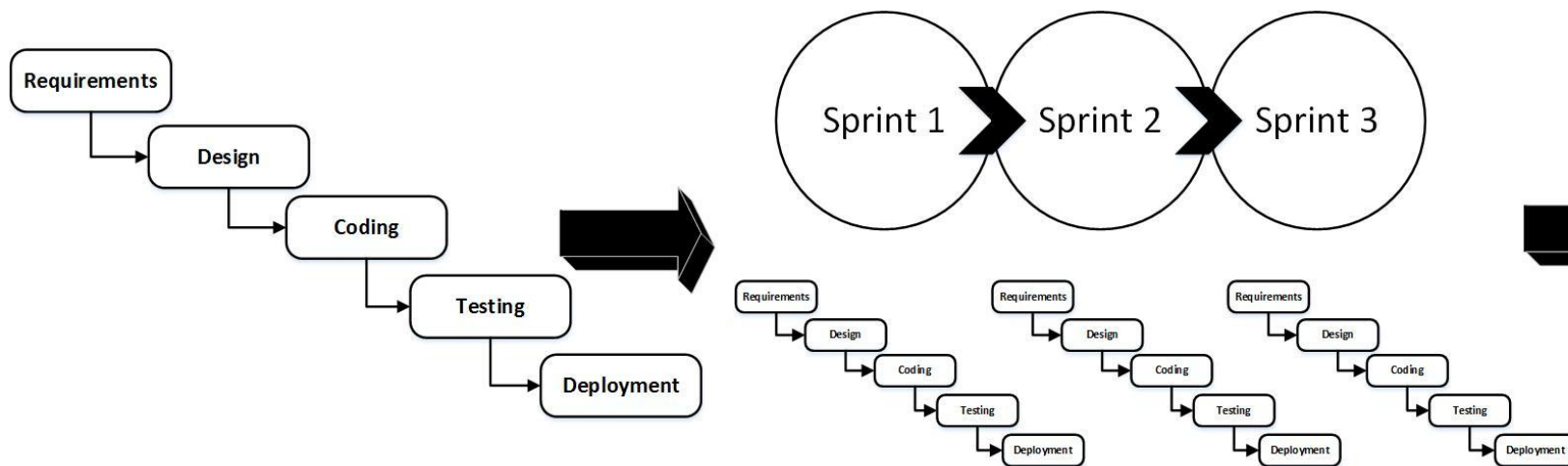


devopsdays



Patrick Debois

# 什么是DevOps--演化

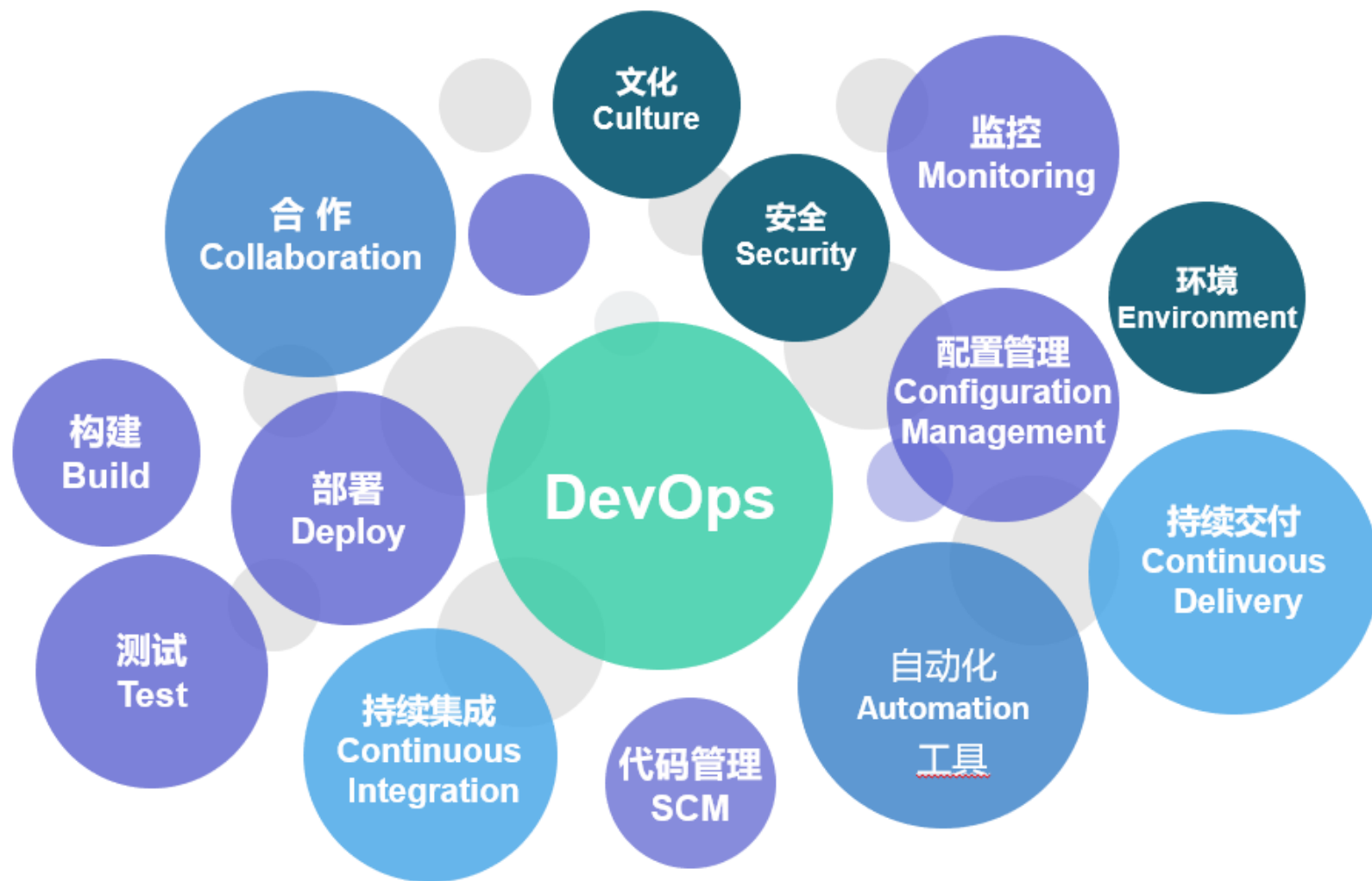


瀑布开发模型

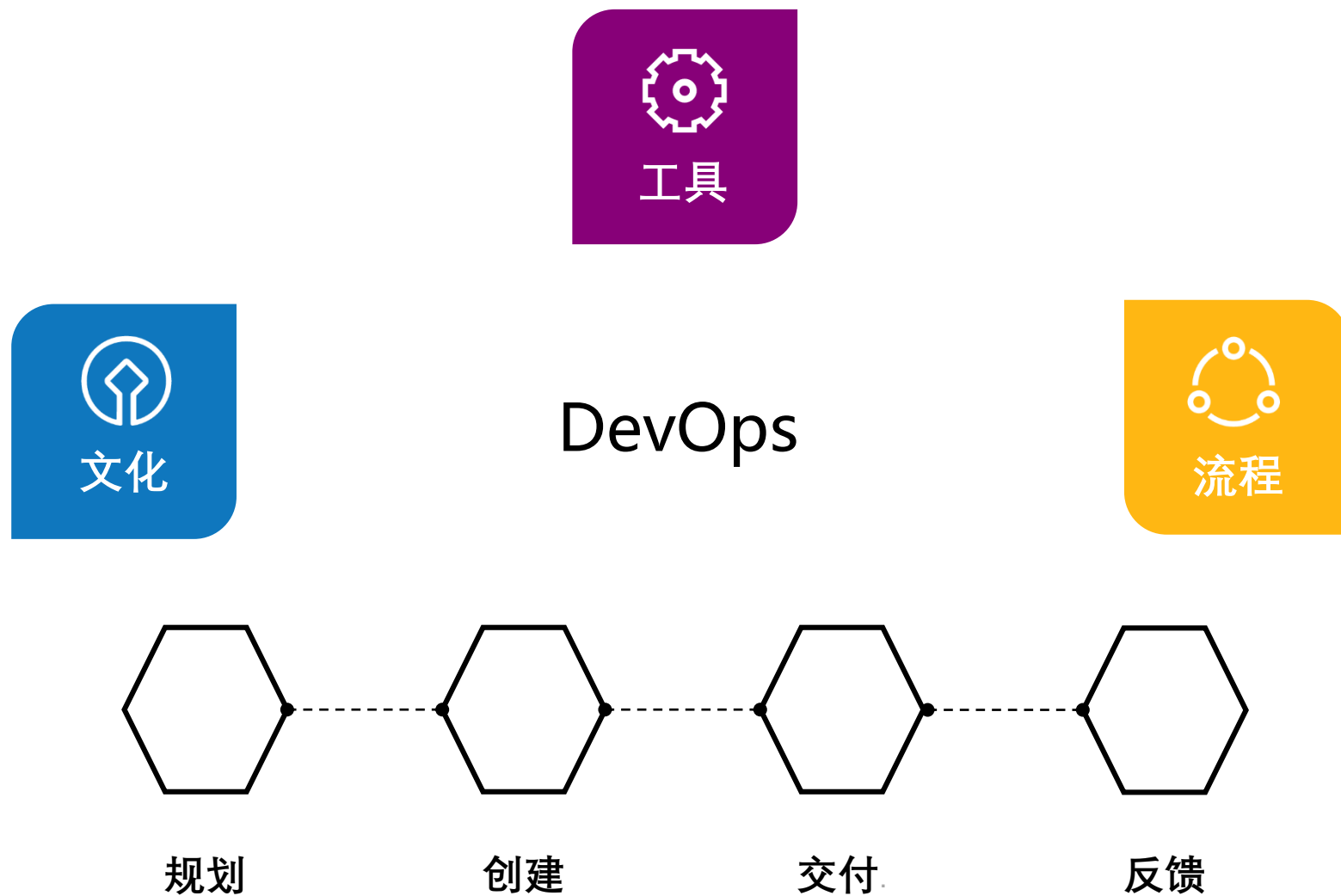
敏捷开发模型

DevOps

# 什么是DevOps--概念



# 为什么要DevOps--关注点更接近商业目标



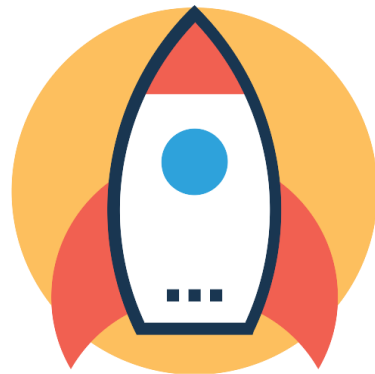
软件生命周期

# 为什么要DevOps--既“快”且“稳”

无论是为了增加业务利润还是创造社会效益，几乎每个组织都依赖于**提高IT性能**去促进这些目标的实现。

越来越多的证据表明DevOps能够帮助组织更好更快地发布。

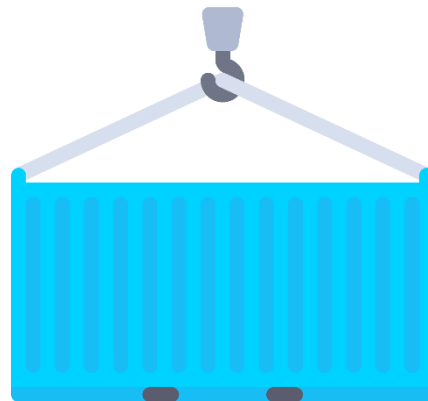
- **代码吞吐量**：吞吐量通过团队所能达到的代码部署频率以及代码从提交到部署的速度进行衡量。代码吞吐量可以衡量“快”。
- **系统稳定性**：稳定性则是通过系统从停机状态恢复的速度以及修改后成功和失败率来衡量的。系统稳定性可以衡量“稳”。



“快”  
Dev  
Throughput

部署频率

交付周期



“稳”  
Ops  
Stability

平均修复时长

变更失败比例

例如：交付周期能相差**70倍**！



# 为什么要DevOps—团队成员更“幸福”



更快更好的交付业务，提高效率



管理人员更有信心



代码体现更多价值



技术人员单位时间创造更多价值

# CONTENTS

## 02 问卷调查研究

- 问卷设计
- 调查过程
- 受访对象

# 动机



## 200% Faster DevOps Lifecycle

From project planning and source code management to CI/CD and monitoring, GitLab is a single application for **the entire DevOps lifecycle**. Only GitLab enables **Concurrent DevOps** to make the software lifecycle 200% faster.



# 问卷设计—参考文献

## 论文及著作

Len Bass, I. Weber, and L. Zhu. **DevOps: A Software Architect's Perspective**. Addison-Wesley Professional, 2015

Shahin, Mojtaba, M. A. Babar, and L. Zhu. " **Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices**. " IEEE Access 5.99(2017):3909-3943.

Rafferty, Alannah E, and M. A. Griffin. " **Dimensions of transformational leadership: Conceptual and empirical extensions**." Leadership Quarterly 15.3(2004):329-354

李杉杉, 李铮, 张贺, 刘博涵 和 荣国平. " **DevOps 自动化支持工具及实践现状经验研究**." 第十六届全国软件与应用学术会议 (NASAC 2017)

## 工业界报告

Puppet Labs "2013 State of DevOps Report."

Puppet Labs "2014 State of DevOps Report."

Puppet Labs "2015 State of DevOps Report."

Puppet Labs "2016 State of DevOps Report."

Puppet Labs "2017 State of DevOps Report."



## 问卷设计--卷首

**调查目的：** 让DevOps在中国获得进一步的推广，让更多的组织和团队了解DevOps，分享DevOps的经验，同时，也为DevOps的每一位实践者提供有效指引和帮助。

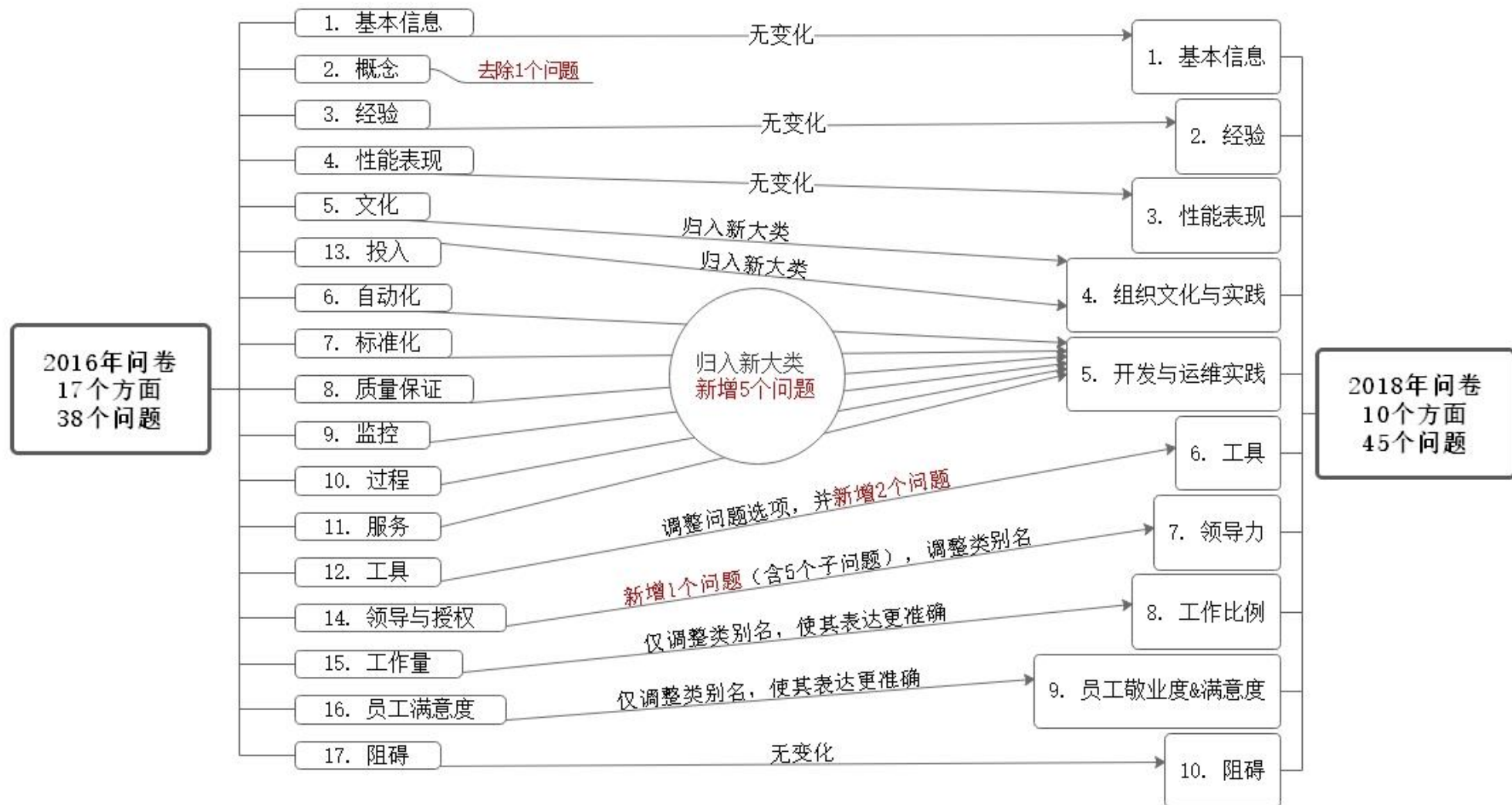
**保密声明：** 这份问卷是匿名的，调查结果只用于统计、分析和生成最终的报告，我们将确保不会泄漏每一位受访者的相关信息。

**目标对象：** 如果您没有任何DevOps相关知识或经验，我们不建议您回答该问卷。

**答题耗时：** 回答这份问卷您将花费大约20分钟。

**激励措施：** 请在下方填写您的邮箱地址，报告将会在完成后通过邮件发送给您

# 问卷设计--问卷结构演化



# 调查过程—分发及反馈

在线多渠道



InfoQ<sup>new</sup>

2018 全球技术周

暨第四届南京（全球）软件大会

收集:2016.9月~2016年12月

发布:2017年8月

历时11个月

74份  
反馈

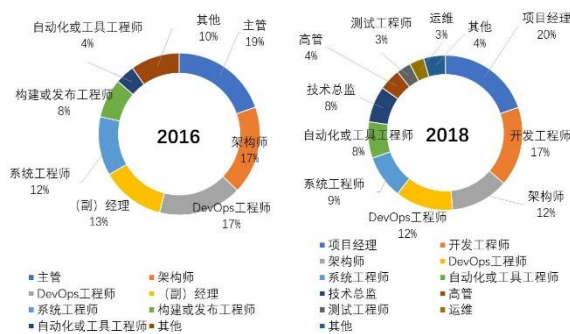
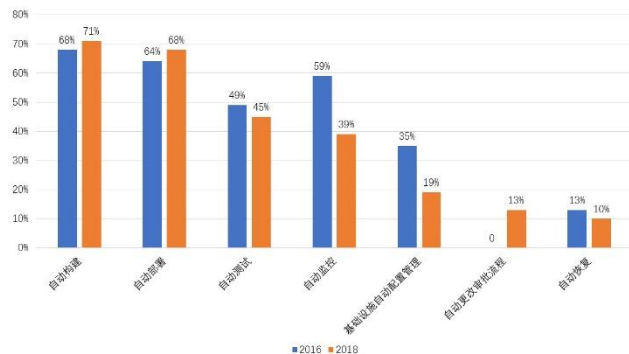
收集: 2017年9月~2018年4月

发布: 2018年5月

历时9个月

66份  
反馈

# 调查过程—分析方法



数据分类：按IT性能分为准高和准低性能团队

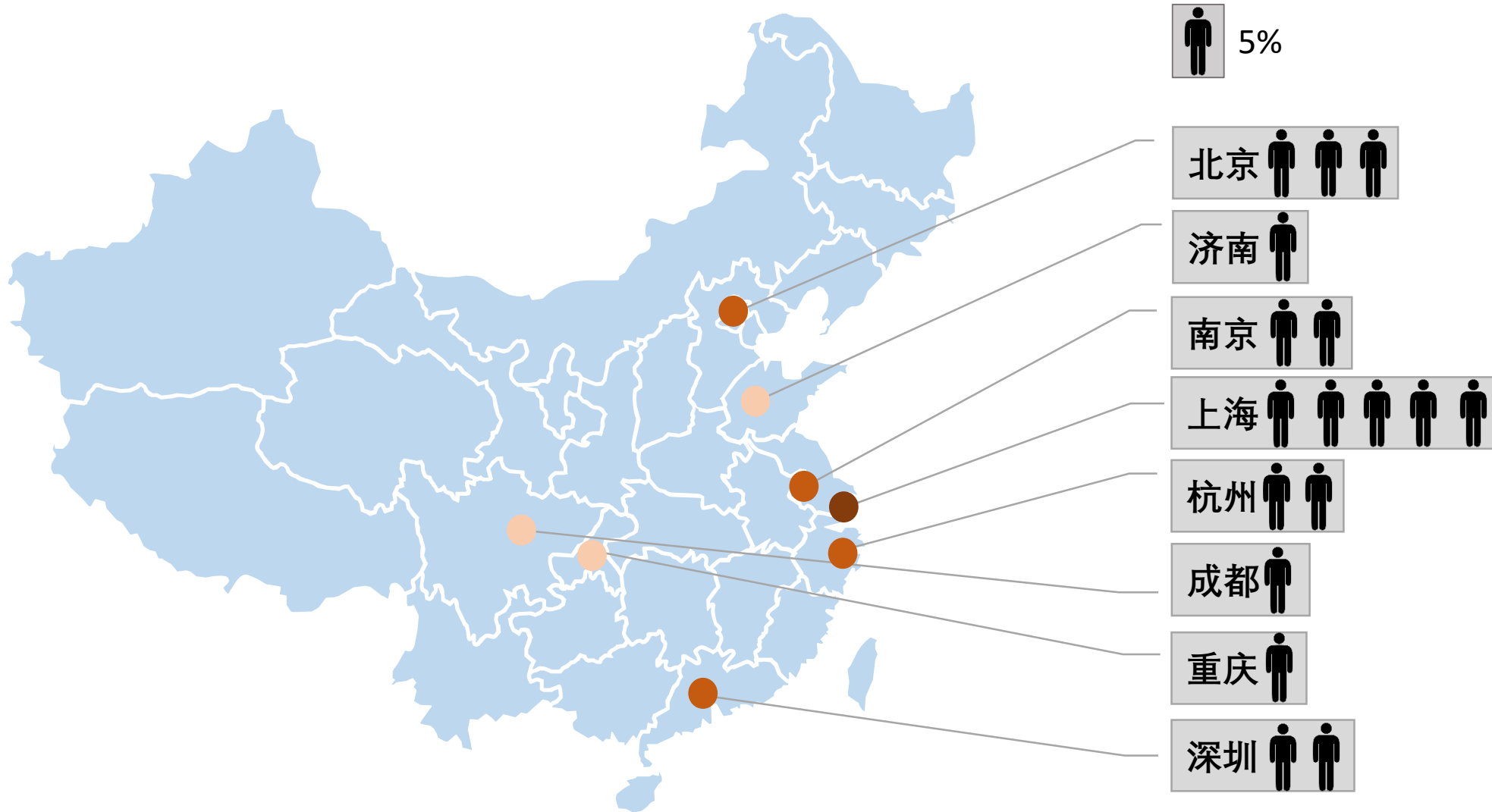
数量统计：多类型图表



# 调查过程—年度报告



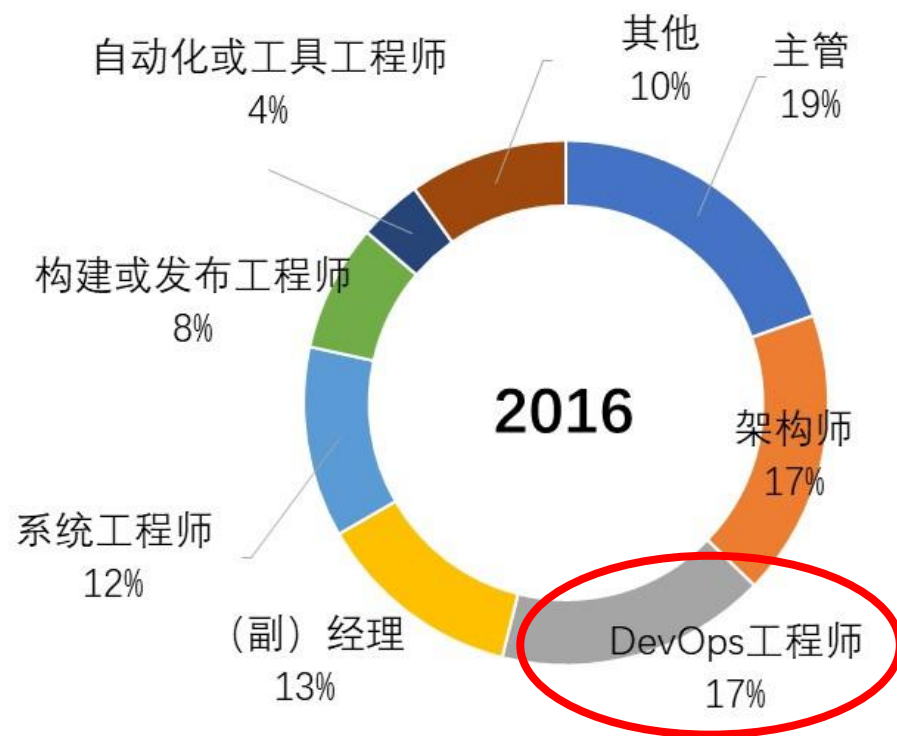
# 受访对象--所在城市分布



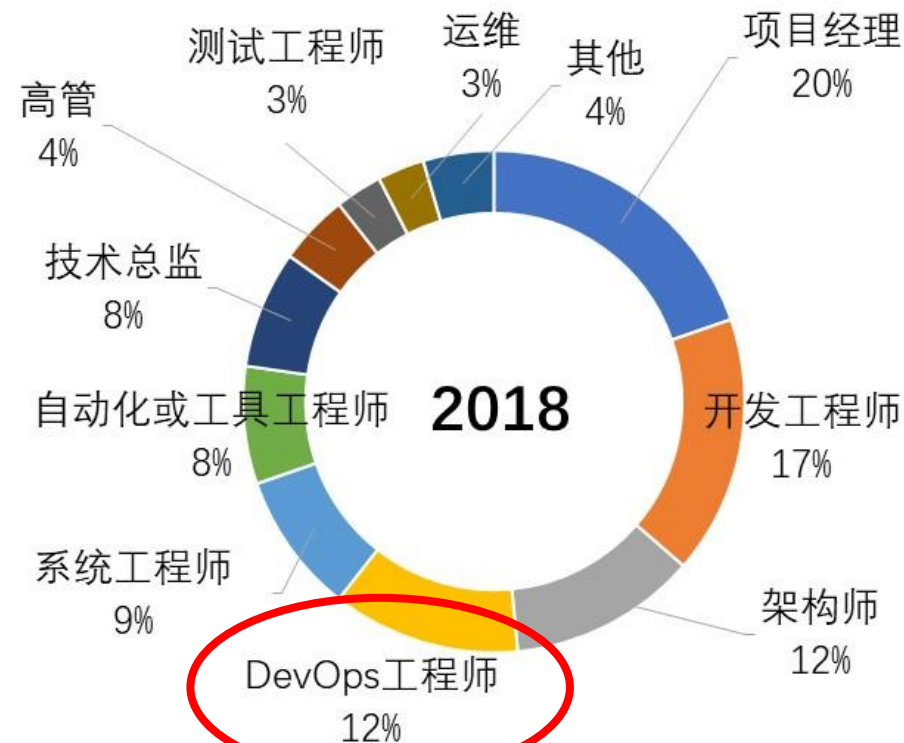
受访者均来自一、二线城市

## 受访对象--岗位

部分组织已经有专门的  
**DevOps工程师**



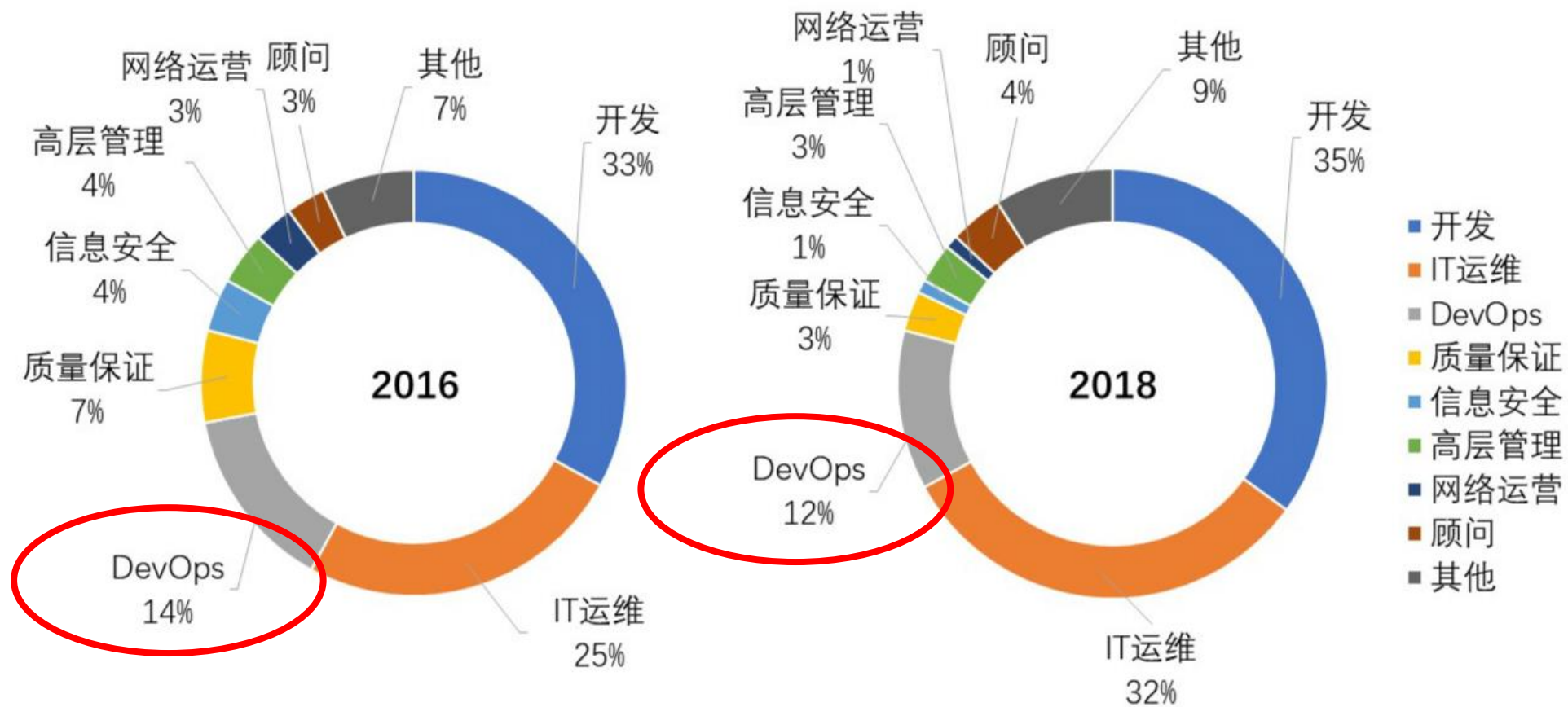
- 主管
- DevOps工程师
- 系统工程师
- 自动化或工具工程师
- 架构师
- (副) 经理
- 构建或发布工程师
- 其他



- 项目经理
- 开发工程师
- 架构师
- DevOps工程师
- 系统工程师
- 自动化或工具工程师
- 技术总监
- 高管
- 测试工程师
- 运维
- 其他

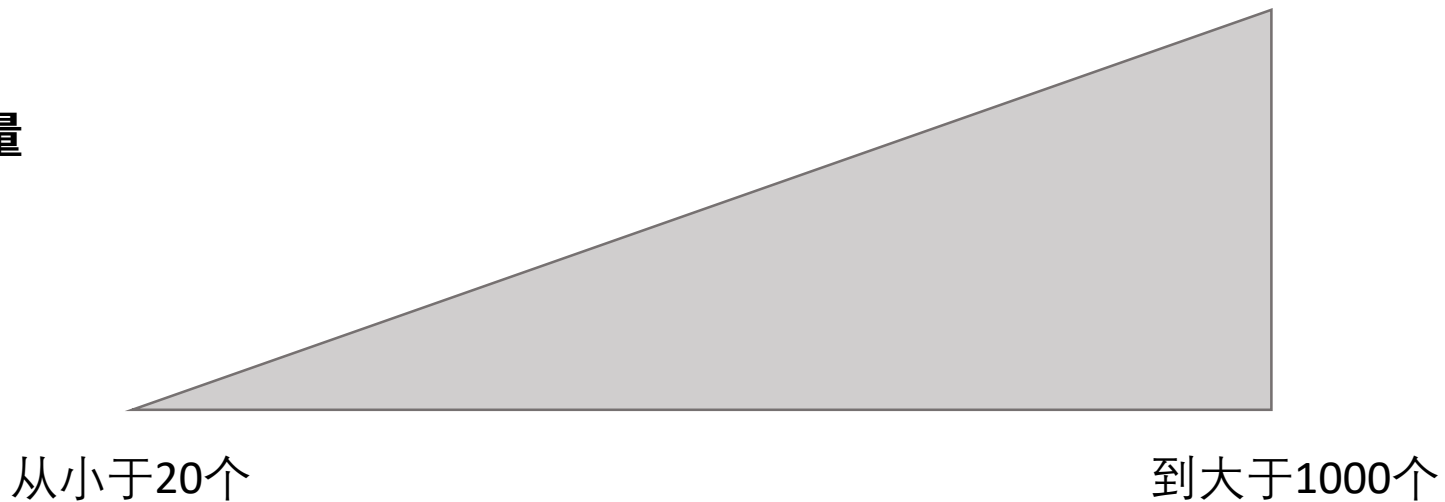
## 受访对象--部门

部分组织已经有专门的  
**DevOps**部门

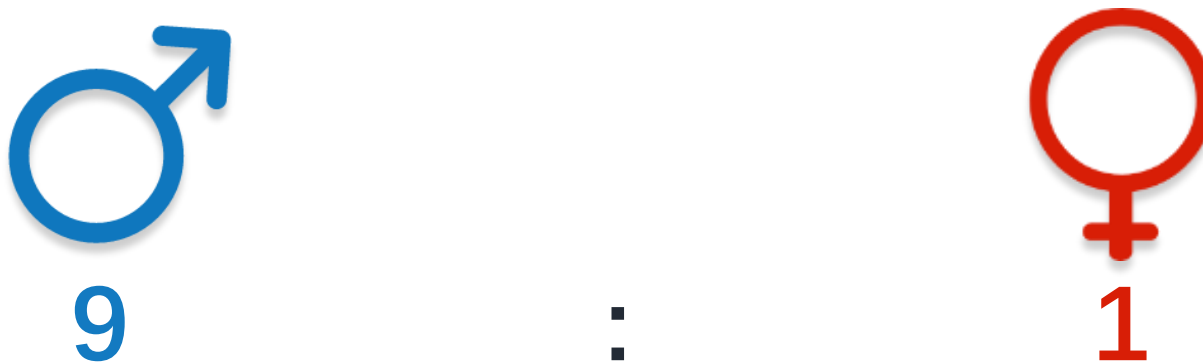


# 受访对象--组织及团队组成

## 组织服务器数量

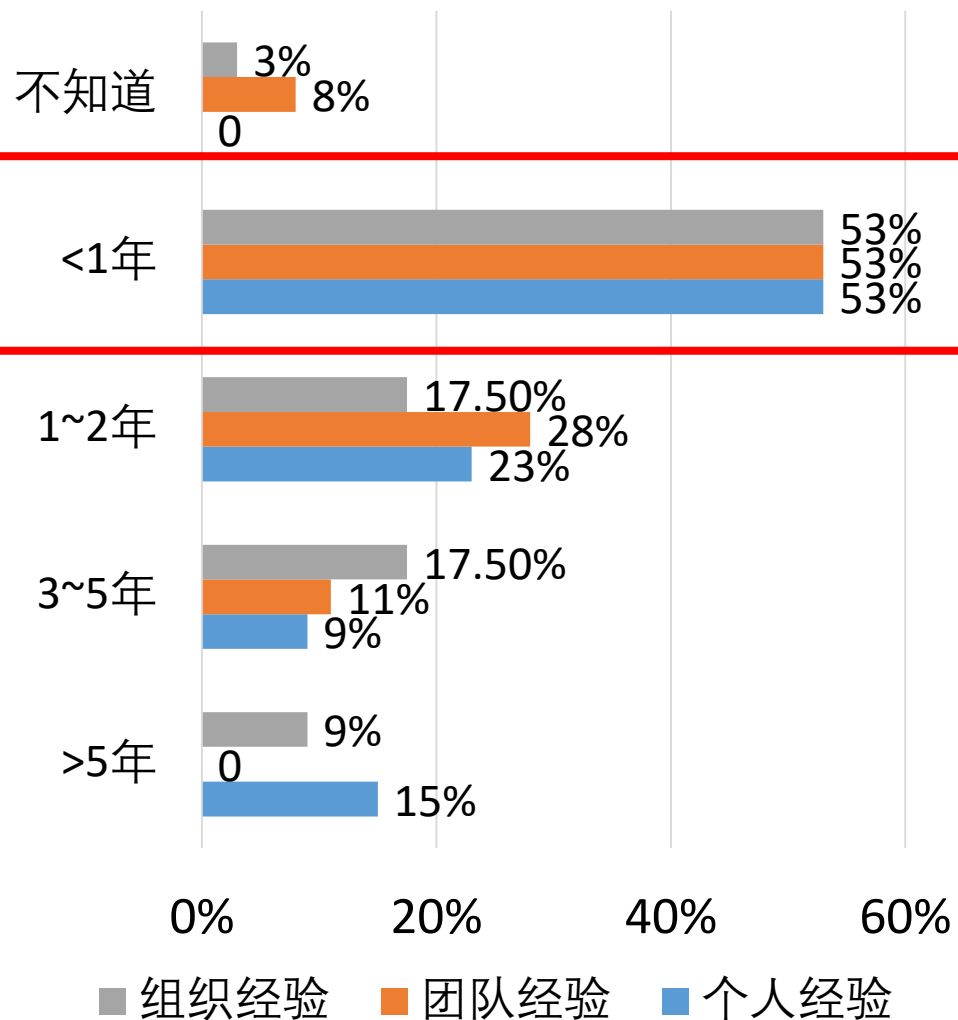


## 团队性别比例

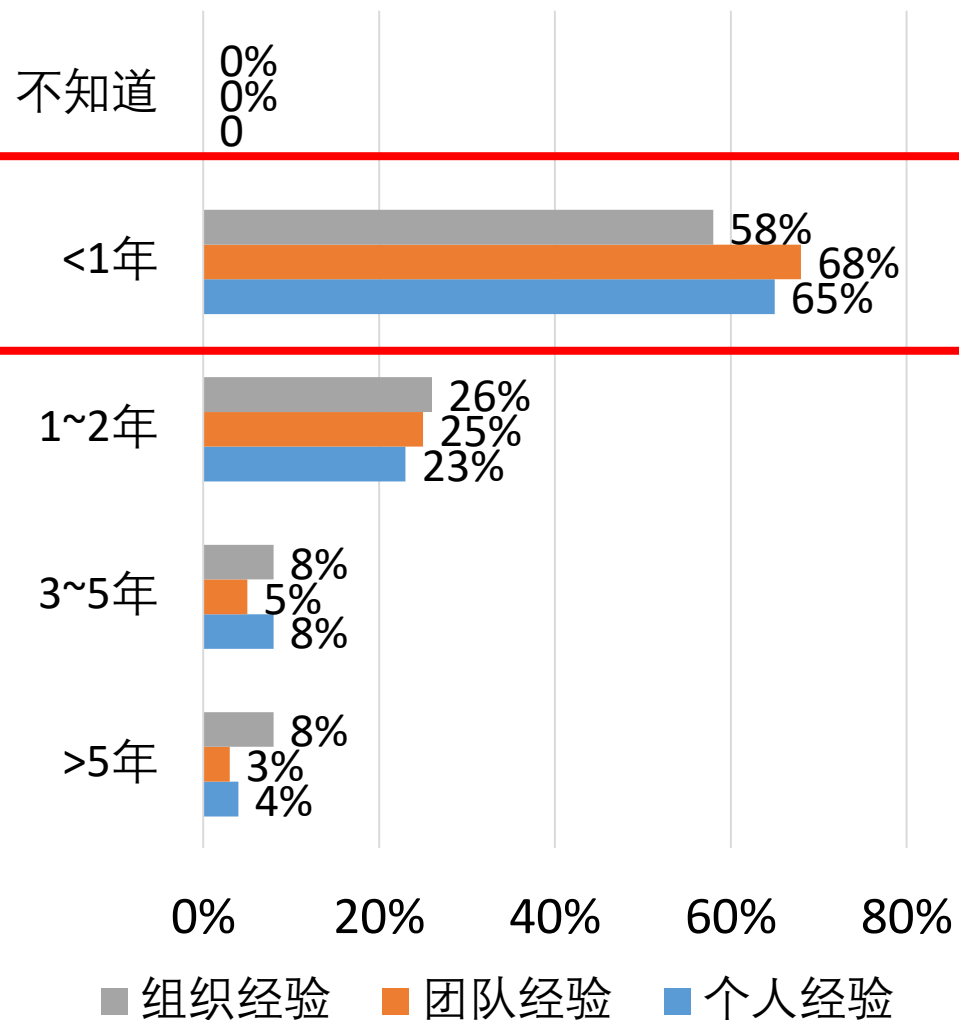


# 受访对象--经验

2016



2018



# CONTENTS

## 03 国内外现状及发展对比

- 组织结构
- IT性能表现
- 组织文化及相关实践
- 开发与运维实践
- 工具支持
- 领导力
- 工作比例
- 员工敬业度及满意度
- 阻碍

# 组织结构--DevOps

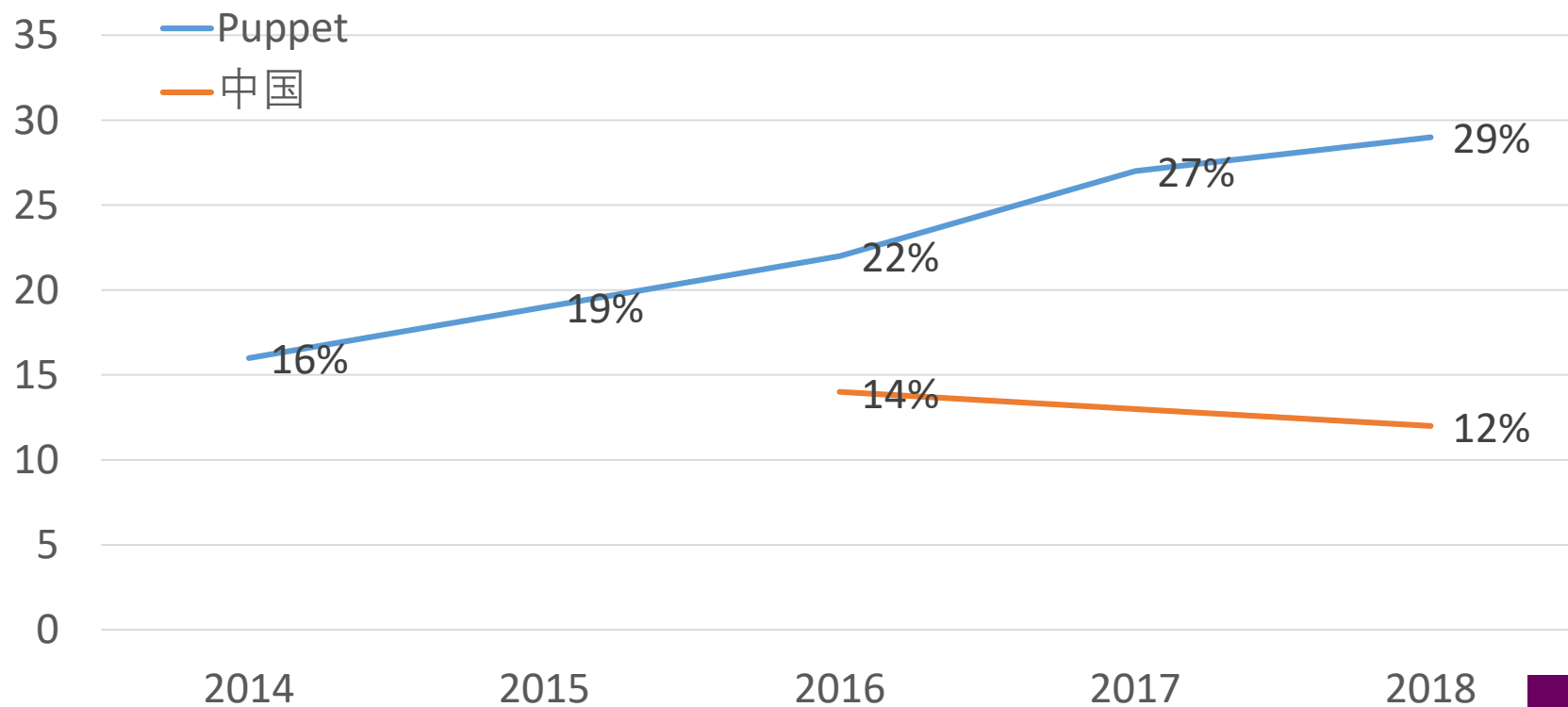
## DevOps工程师比例

Puppet 2014年: 31%

接近3倍

中国 2018年: 12%

## DevOps部门比例





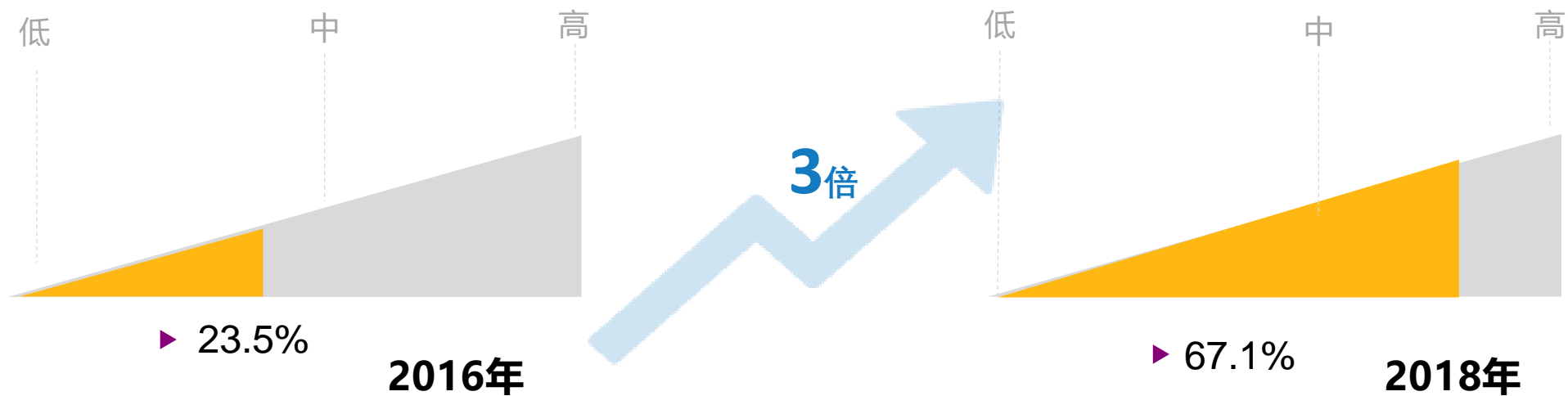


## 组织结构--发现

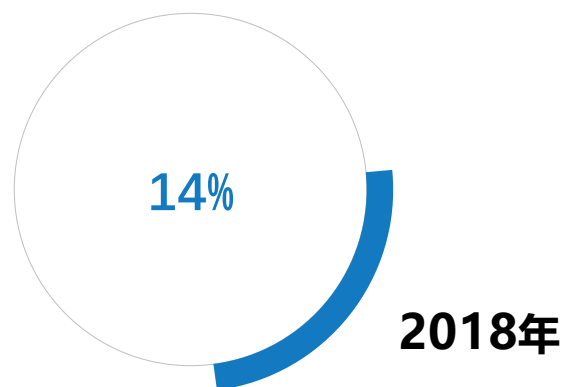
**发现：**国内在组织结构上针对 DevOps 的改进，还处于全球 2014 年甚至之前的水平，且**两极分化**严重，鲜有选择折中方案，即不增加部门只增加新岗位的企业。在 DevOps 工程师这一新岗位的资源投入上，即使与 2014 年的国际水平相比，也还普遍存在**明显差距**。

# IT性能表现--交付周期

## 一天内完成交付



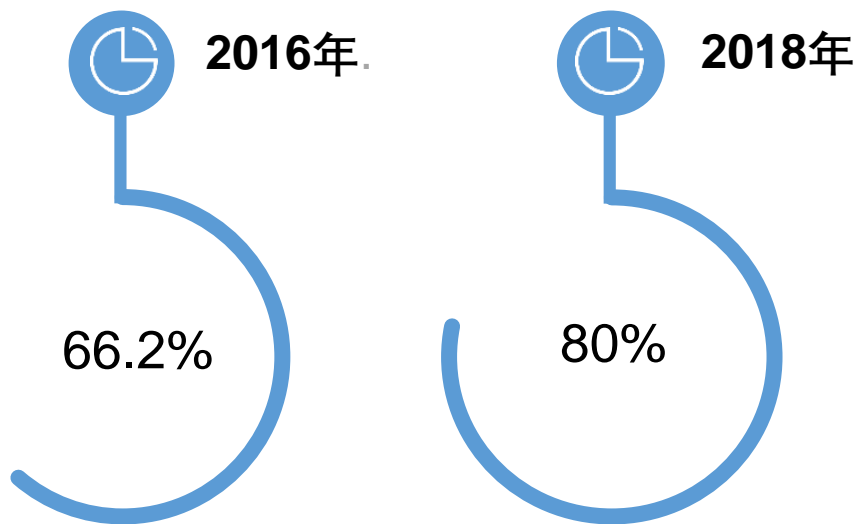
## 一小时内完成交付



# IT性能表现--变更失败比例

变更失败比例也就是在对于应用或服务进行变更时，导致了服务的降级或随后修复的比例情况，所谓变更失败就是在变更之后致服务受损，服务中断，需要热修复，回滚，打补丁等。

## 变更失败比例在0-15%的团队



- 2018年**
- ▶ 控制在10%以内的团队 **73.8%**
  - ▶ 控制在5%以内的团队 **36.4%**

# IT性能表现--团队分类

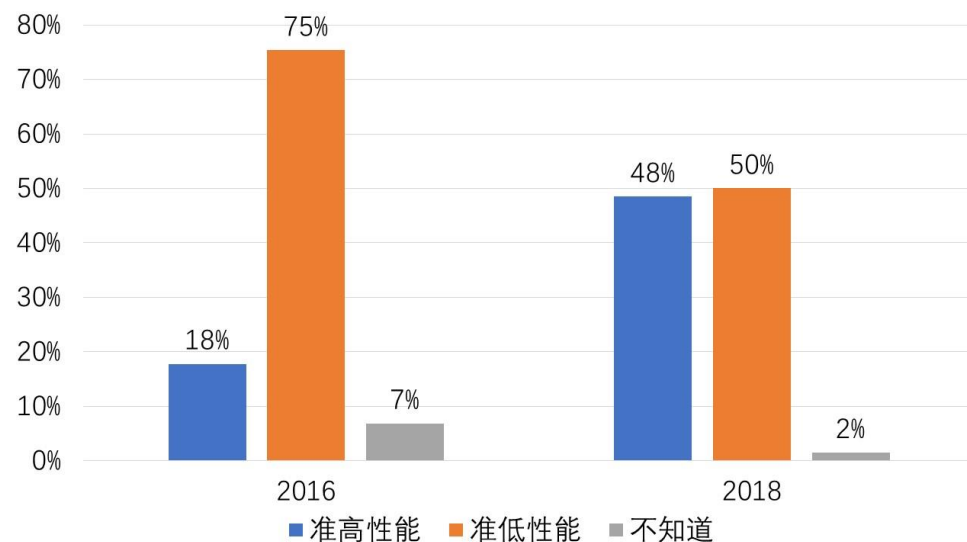
## 国际上高低性能标准

参数	高性能	低性能
部署频率	一天之内	超过一周
交付周期	一小时之内	超过一天
平均修复时长	一小时之内	超过一天
变更失败比例	0%-15%	大于15%

按照以上的衡量标准，2018年受访团队中仅有**6.0%**能够达到高性能指标。

## 我们定义的准高低性能标准

参数	准高性能	准低性能
部署频率	一周之内	超过一周
交付周期	一天之内	超过一天
平均修复时长	一天之内	超过一天
变更失败比例	0%-15%	大于15%





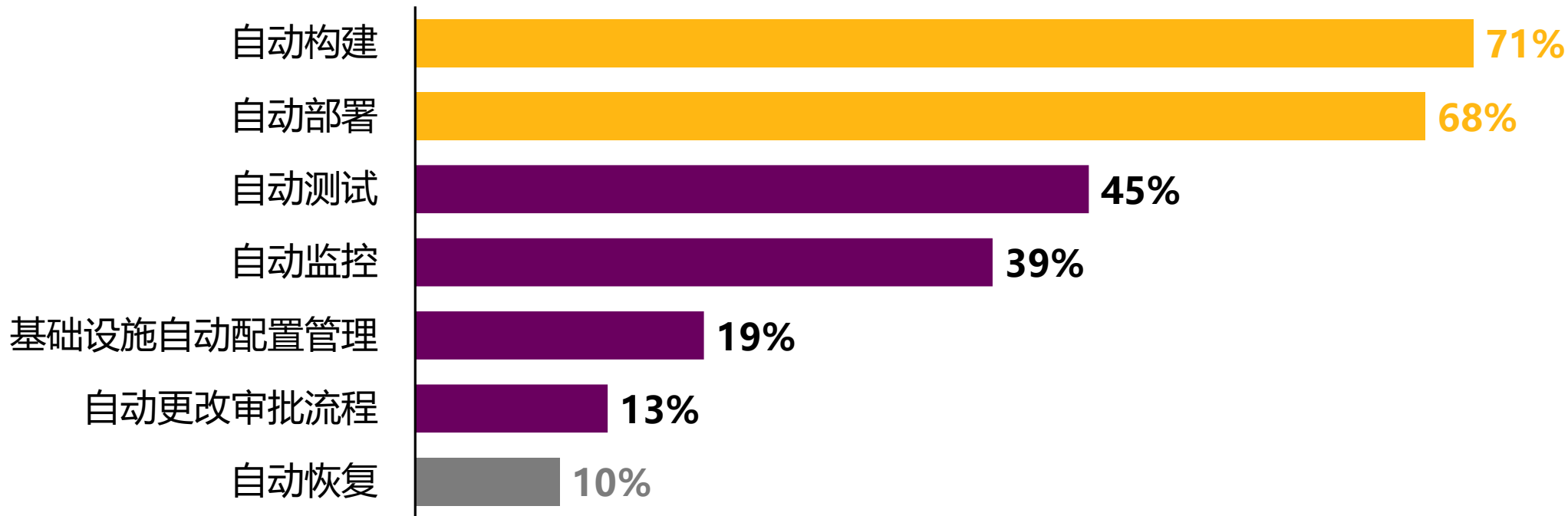
# IT性能表现--发现

**发现1：** DevOps 经验丰富的团队， IT 性能表现更好；从 2016 年到 2018 年，**交付周期**和**变更失败比例**性能有着明显提升，但部署频率和平均修复时长基本不变。

**发现2：** 除**变更失败比例**外，国内其他三项指标与国际高水平均存在差距

# 开发与运维实践--自动化

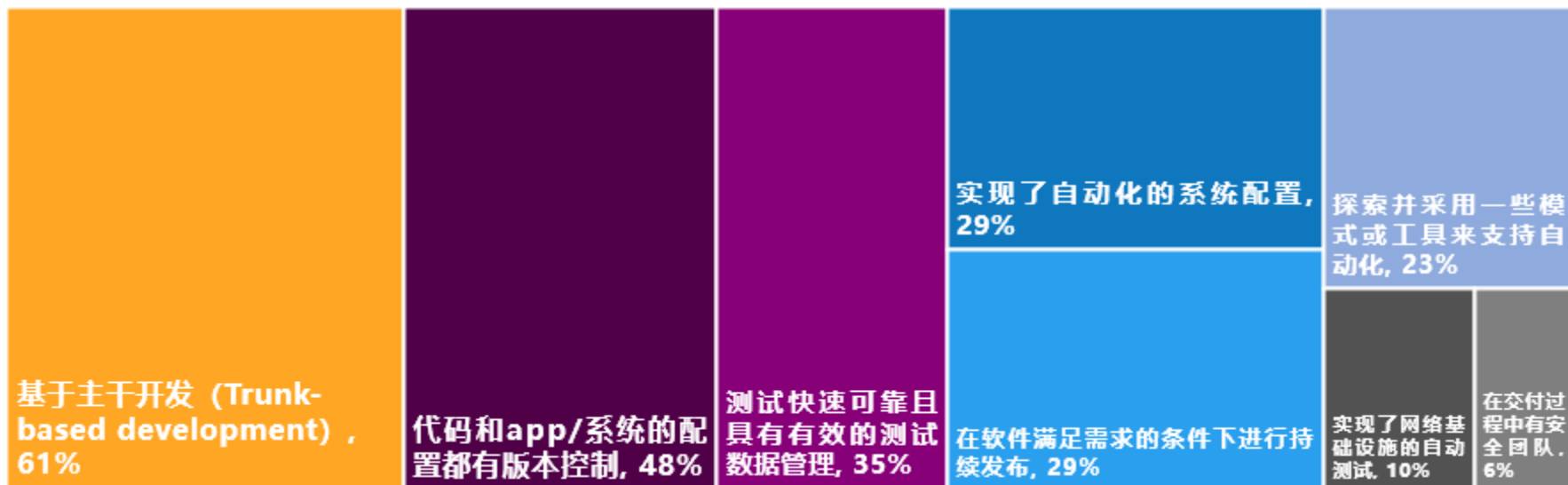
## 团队的过程自动化



受访的**准高性能团队**均实现了部分过程的自动化，其中以**自动构建（71%）**和**自动部署（68%）**为最，另外近半数的团队实现了自动测试（45%）以及自动监控（39%）。比例最低的为自动恢复，仅有10%。

# 开发与运维实践--自动化

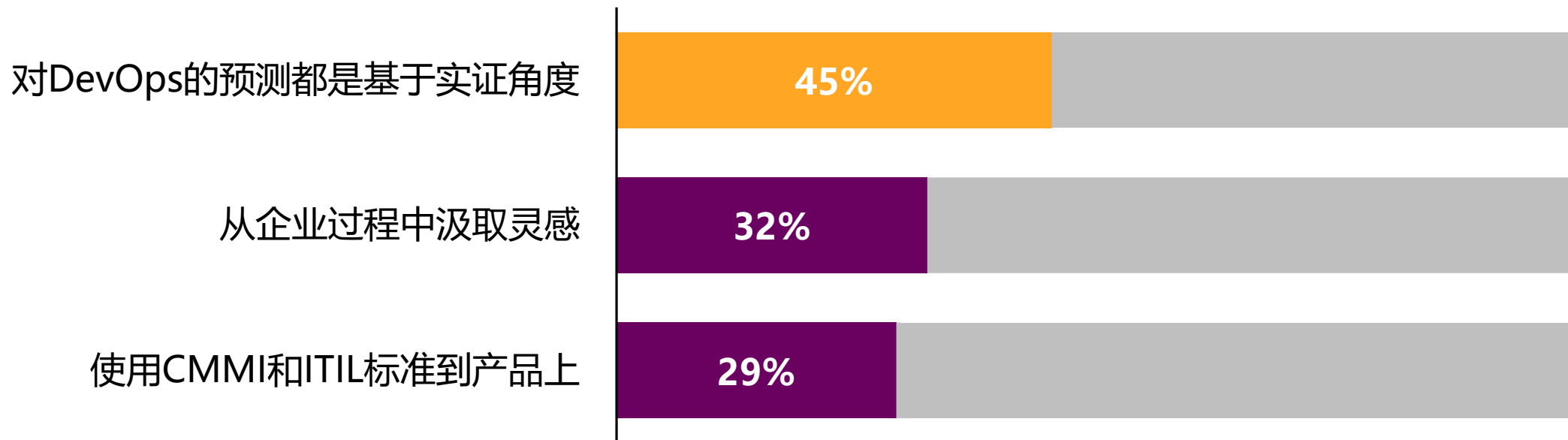
## 团队的自动化实践



受访的**准高性能团队**在自动化实践方面均有所涉猎，其中**基于主干开发**占比最大（61%）， “代码和app/系统的配置都有版本控制” 紧跟其后（48%）。

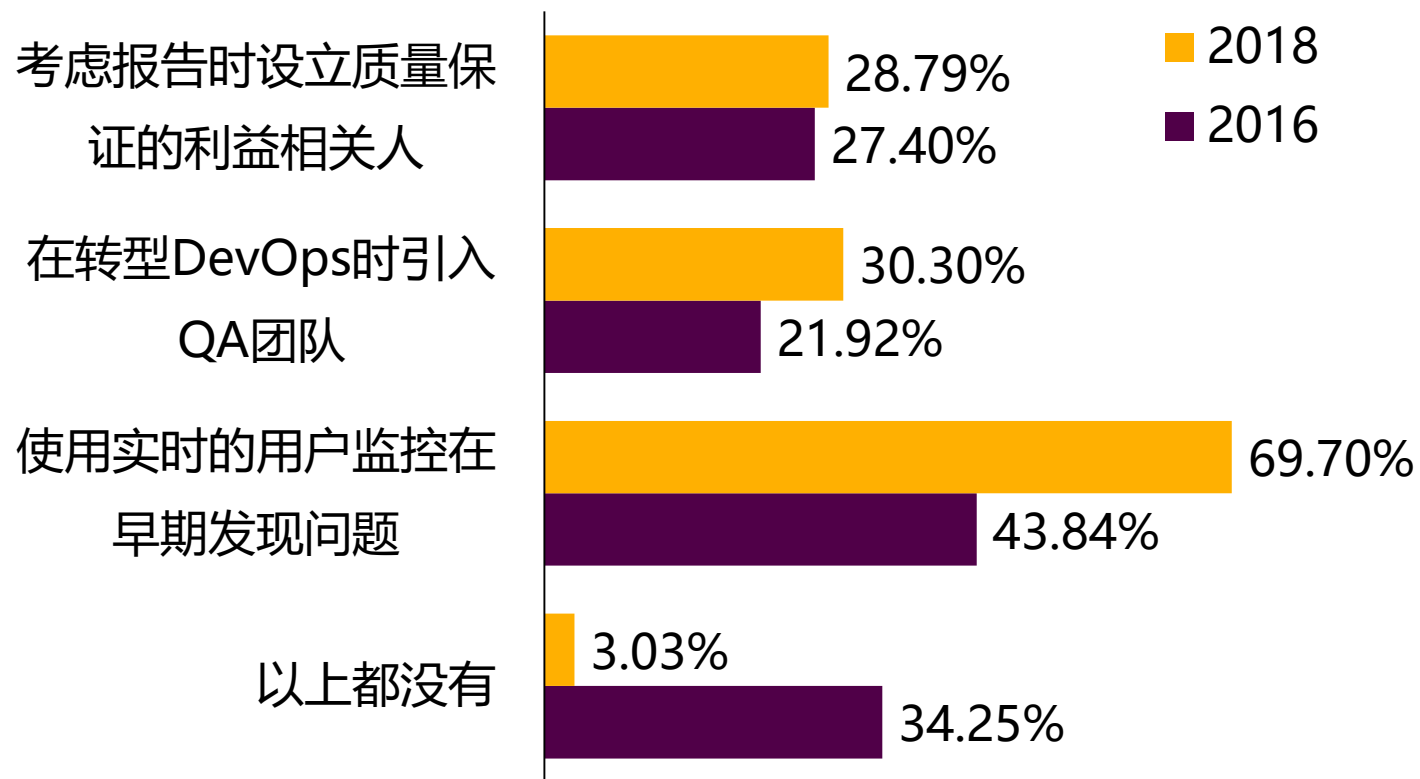
# 开发与运维实践--标准化

**软件标准**是为了协调开发项目的各个阶段和各个部分联系和衔接而做出的**约束和规定**。良好的标准化实践可以提高软件可靠性、可维护性和可移植性，缩短软件的开发周期和降低运行维护成本。然而由于软件开发的特性，标准化往往难以得到严格的实施。





## 开发与运维实践--质量保证



**2018年**三种主要的QA实践所使用的比例都有所增长。 2016年有**超过30%**的团队没有任何QA实践，而在2018年的调查数据中，**仅有约3%的团队没有任何QA实践。**

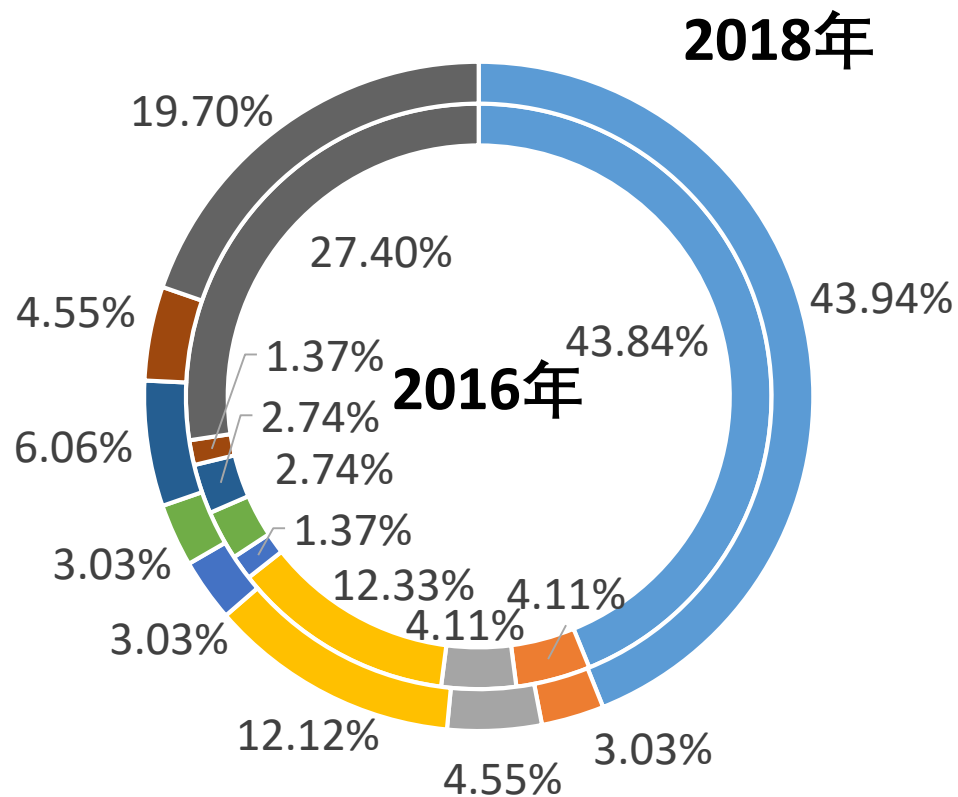
0实践比例下降10倍

# 开发与运维实践--敏捷方法

## 敏捷方法

2016年及2018年敏捷方法的使用情况

- Scrum
- XP
- Lean Development
- Kanban
- Scrumban
- Scrum/XP Hybrid
- Custom Hybrid (multiple methodologies)



相比2016年，没有使用任何敏捷方法的团队减少了**7.7%**

**Scrum**占比超过**4成**



# 开发与运维实践--发现

**发现1：**国内团队**自动化**实践已基本普及，尤其是在构建、部署、测试和监控等阶段。但在自动恢复上还明显重视不足。此外，国内还鲜有充分将安全和网络基础设施自动化考虑进来的团队。

**发现2：****基于主干开发**是最流行的自动化实践。

**发现3：**受访团队已普遍实现了**标准化**，但更多的是基于实证的角度，缺少量化的**数学模型**的构建。

**发现 4：**国内团队在**质量保证**实践上有着明显的进步，且目前几乎所有团队均有质量保证实践。

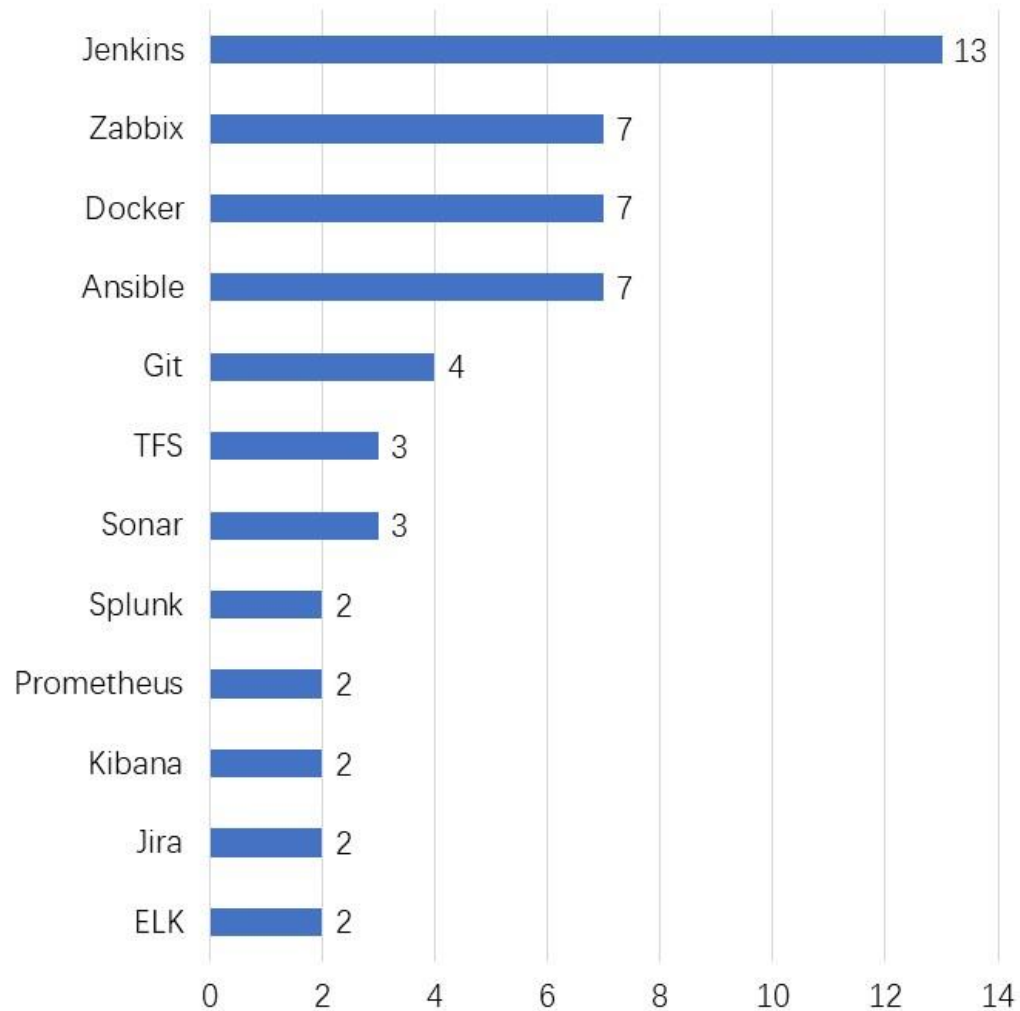
**发现 5：**国内团队对于**敏捷方法**的采用呈现增长趋势，虽各种敏捷方法均有团队使用，但 **Scrum** 始终表现出一枝独秀的态势。

# 工具支持--推荐

准高性能团队工具使用比例

Docker	64.5%
Jenkins	58.1%
Git	51.6%
Ansible	48.4%
Zabbix	45.2%
Kibana	32.3%
Subversion	22.6%
AWS	16.1%
SaltStack	12.9%
Puppet	12.9%
Splunk	10.0%
Nagios	6.5%

其他推荐工具



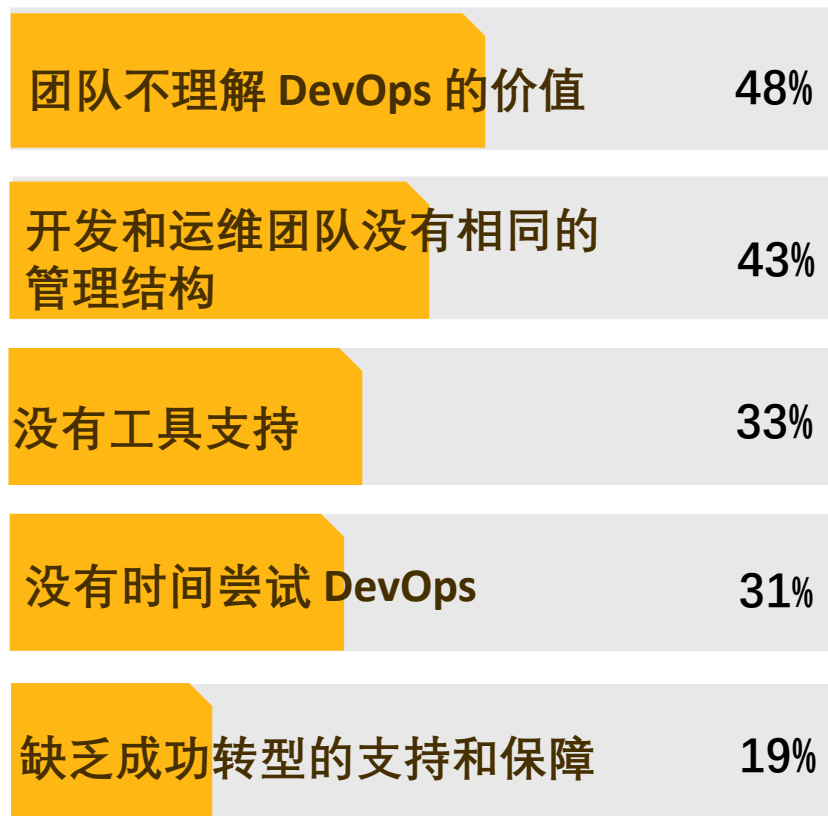


## 工具支持--发现

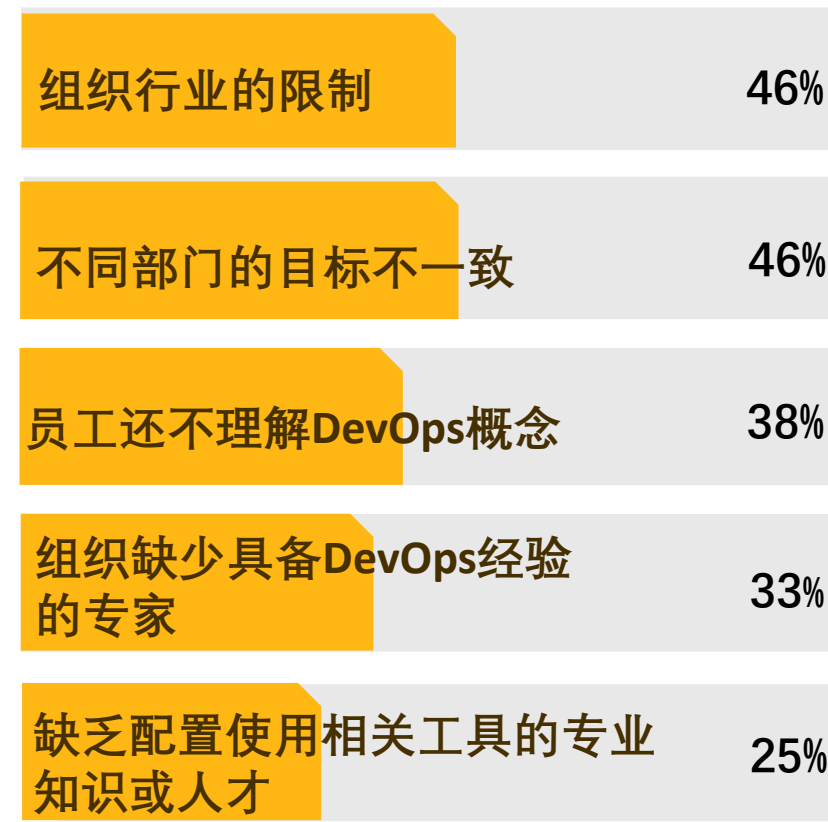
**发现：**国内 DevOps 团队对工具的使用较为普遍，且在工具选择上呈现了多样性；在多样性中也突显出了一些被广泛使用和值得推荐的工具，如 **JenKins**（持续集成），**Ansible**（配置管理）和 **Docker**（容器）等

## 阻碍--TOP5

### | 国外阻碍（2013年）：



### | 国内阻碍：





## 阻碍--发现

**发现1：**阻碍 DevOps 推行的五大因素中与“人”相关的因素高达80%，仅组织行业一条与“人”无关。

**发现2：**国内企业当前在实践 DevOps 所面临的阻碍与国际上 2013 年所面临的阻碍相似，其根本问题在企业对于 DevOps 的态度和需要的程度。此外，还有一个突出问题在于，国内面临着独有的 DevOps 人才紧缺的问题。

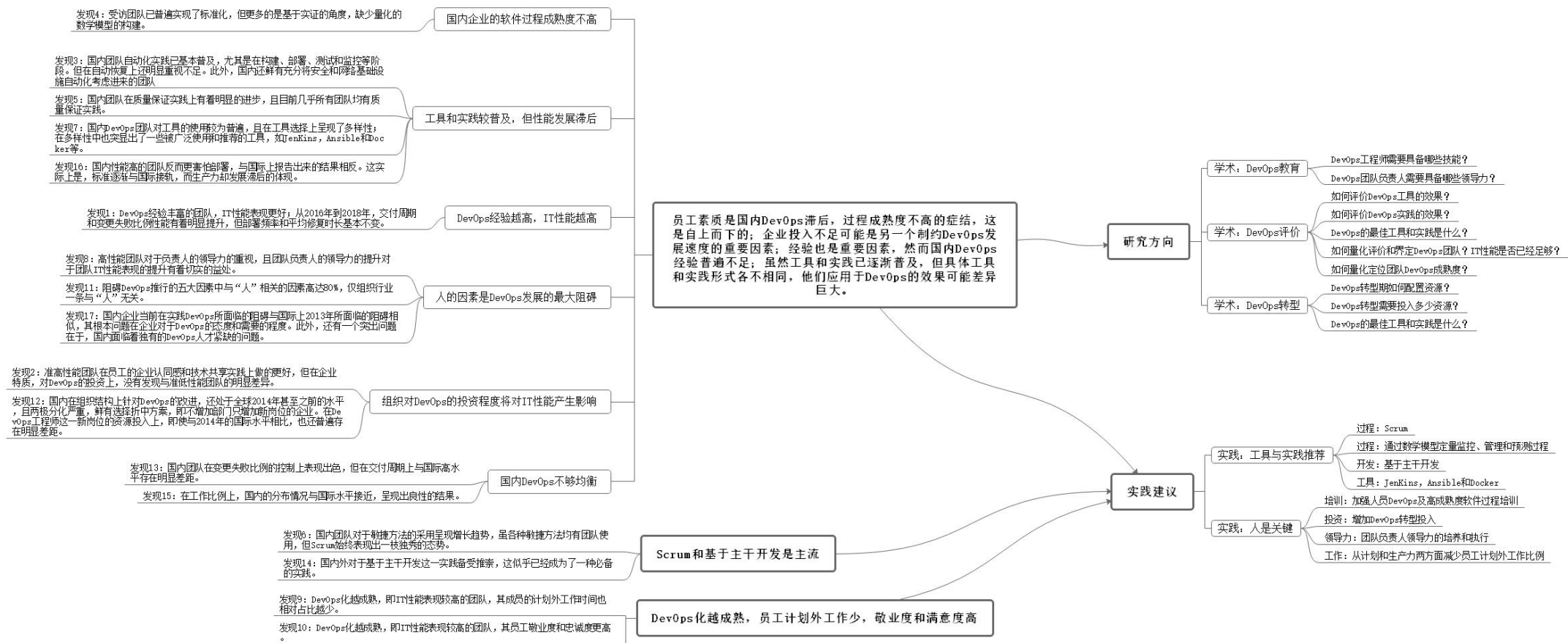
# CONTENTS

## 04 实践建议及研究问题

- 实践建议
- 研究方向



# 综合分析



# 实践建议

## 工具与实践推荐

开发方法：Scrum

开发方式：基于主干开发

过程管理：通过数学模型定量监控、管理和预测过程

工具：JenKins, Ansible和Docker

## 人是关键

培训：加强人员DevOps及高成熟度软件过程培训

投资：增加DevOps转型投入

领导力：团队负责人领导力的培养和执行

工作：从计划和生产力两方面减少员工计划外工作比例

## 弱项

IT性能：交付周期

自动化实践：自动更改审批流程、自动恢复

# 研究方向

## DevOps教育

DevOps工程师需要具备哪些技能？

DevOps团队负责人需要具备哪些领导力？

## DevOps转型

DevOps转型期如何配置资源？

DevOps转型需要投入多少资源？

DevOps的最佳工具和实践是什么？

## DevOps评价

如何评价DevOps工具的效果？

如何评价DevOps实践的效果？

如何量化评价和界定DevOps团队？IT性能是否已经足够？

如何量化定位团队DevOps成熟度？



## DevOps中国调查研究

刘博涵

# THANKS

Email: [hifi\\_lbh@163.com](mailto:hifi_lbh@163.com)



南京大學  
NANJING UNIVERSITY



南京大學软件学院  
NANJING UNIVERSITY · SOFTWARE INSTITUTE