

2018

加速度：

# 全球 DevOps 现状调查报告

新经济战略

出品方



黄金赞助商

Deloitte

XebiaLabs  
Enterprise DevOps

pagerduty

redgate



中文版翻译

cloudbees

Datical

sumologic

TRICENTIS



Electric Cloud

Microsoft Azure

Pivotal

aws



# 目录

|    |   |
|----|---|
| 综述 | 3 |
|----|---|

---

|           |   |
|-----------|---|
| 谁参与了本次调研？ | 5 |
|-----------|---|

|        |   |
|--------|---|
| 调查受访对象 | 6 |
|--------|---|

---

|         |    |
|---------|----|
| 如何进行比较？ | 10 |
|---------|----|

|        |    |
|--------|----|
| 软件交付效能 | 11 |
|--------|----|

|                   |    |
|-------------------|----|
| 软件交付与运维效能：新增可用性指标 | 21 |
|-------------------|----|

---

|             |    |
|-------------|----|
| DEVOPS 重要吗？ | 23 |
|-------------|----|

|      |    |
|------|----|
| 组织效能 | 24 |
|------|----|

|      |    |
|------|----|
| 优质产出 | 27 |
|------|----|

---

|       |    |
|-------|----|
| 如何改进？ | 33 |
|-------|----|

|           |    |
|-----------|----|
| 云计算，平台和开源 | 34 |
|-----------|----|

|    |    |
|----|----|
| 外包 | 43 |
|----|----|

|         |    |
|---------|----|
| 精益与敏捷实践 | 49 |
|---------|----|

|      |    |
|------|----|
| 技术实践 | 52 |
|------|----|

|    |    |
|----|----|
| 文化 | 61 |
|----|----|

---

|    |    |
|----|----|
| 结语 | 71 |
|----|----|

---

|    |    |
|----|----|
| 方法 | 72 |
|----|----|

---

|    |    |
|----|----|
| 致谢 | 74 |
|----|----|





# 综述

DevOps加速度现状调查报告是同类研究中规模最大、持续时间最长的。在过去五年里，全球范围内有超过30,000名专业技术人员参加了这项调查。这些成果使我们能够更好地理解哪些实践能带来更高的软件交付效能，从而产生强有力的业务影响。这也是唯一的运用了聚类分析的DevOps报告，可帮助团队将自己与行业中的高效能、中效能或低效能的基准进行比较，并开展预测分析，从而确定团队可以用来提高效能的具体策略。

我们发现这些好处和结果适用于所有的组织，不管他们来自什么行业。

今年我们考察了云的采纳、开源软件的使用、组织实践（包括外包）和文化对软件交付效能的影响。

我们继续地研究了软件交付度量指标（吞吐量和稳定性）对组织绩效的影响作用，并发现了组织习惯性地为了优化稳定性而做出权衡的证据。

今年的调研首次引入了可用性指标，从而扩展了软件交付的效能度量模型。这项指标提高了对组织产出的分析和预测能力，并且形成了从软件开发、交付到运维的更为全面的视图。我们称这种新的构造方法为**软件交付与运维效能，简称SDO效能**。

这种新的分析方法可以使我们更为深入地了解DevOps的转型。

## 关键发现如下：

p21

### SDO效能解锁竞争优势

提升盈利能力、生产力、市场份额、客户满意度，以及实现组织目标和使命的能力

p37

### 如何实施云基础设施很关键

云提高了软件交付的效能。具备云计算所有核心特征的团队，其属于高效能组织的可能性要高出23倍

p42

### 开源软件可以提高效能

高效能组织广泛应用开源软件的频率比其他组织要高1.75倍，并且在未来扩展开源软件使用范围的可能性是其他团队的1.5倍

p43

### 精英效能团队几乎不采用职能外包，因为这会有损于效能

通常外包可以节省成本并提供灵活的人力资源池，然而低效能组织将测试或运维等职能全部外包的比例，至少是高效能组织的4倍

p52

### 关键技术实践驱动高效能

这些实践包括监控与可观察性、持续测试、数据库变更管理，以及尽早在软件开发过程中集成安全性

p11

### 实现软件交付的高效能与行业无关

我们发现在强监管行业和弱监管行业中，都存在着在软件交付方面实现了高效能的组织

# 谁参与了本次调研？

我们的研究给 DevOps 的行业发展状况提供了最为全面的视角，其来源包括了在最近五年的科学研究中，收到的超过30,000份的调查问卷反馈。今年，全球共有近1,900人参与了这项研究。尽管其中女性及非代表性少数族裔的比例反映了这个行业在多元化方面的进步，但离真正实现平等多元还相距甚远。我们今年还首次收到了来自残疾人士的反馈。



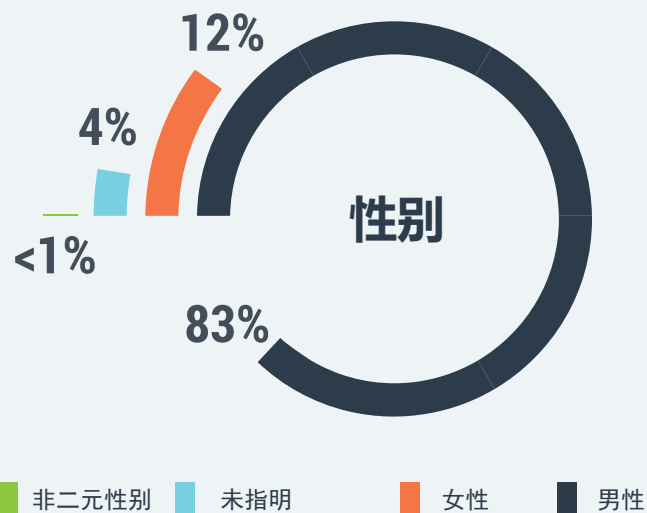
## 调查受访对象

我们看到和前几年相比，今年女性受访者的比例有显著增长。我们相信这一增长体现了行业中性别分布的变化趋势<sup>1</sup>，同时也得益于我们对调研样本多元化的关注。为了更全面地了解女性在技术团队中的比例（即使其中部分人并未参与我们的调查），我们特别询问了受访者，并得知他们团队中25%为女性。需要注意的是，这一受访者统计数据上的变化，并没有推翻以前的调查结果；而是帮助我们再次确认并验证了前几年的结果，对此，我们在报告中都有提及。

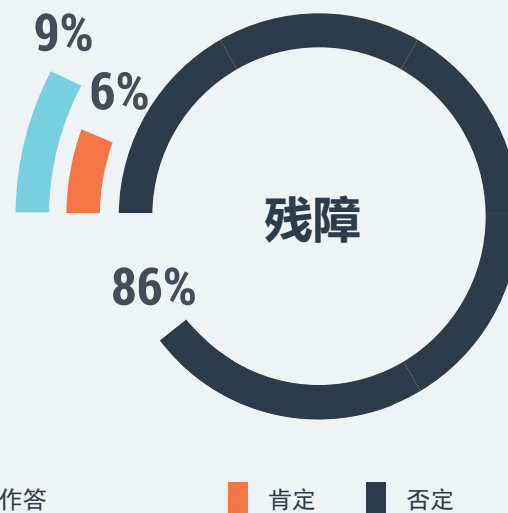
---

<sup>1</sup> 在HackerRank 2018年的报告中，女性在技术领域中的比例为14.2%  
(<https://www.hpcwire.com/2018/03/05/2018-hackerrank-report-shows-progress-challenge-women-coders/>),  
NCWIT研究报告中，女性在技术领域中的比例为25%  
([https://www.ncwit.org/sites/default/files/resources/womenintech\\_facts\\_fullreport\\_05132016.pdf](https://www.ncwit.org/sites/default/files/resources/womenintech_facts_fullreport_05132016.pdf))

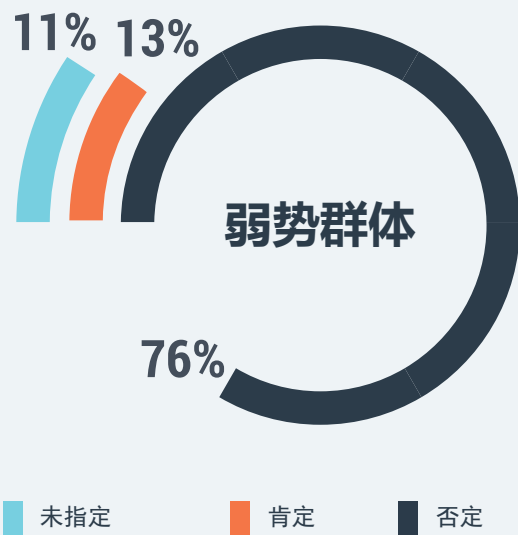
## 受访者统计



受访者表示其团队中25%为女性



这是我们首次对残障情况进行调查，在这个过程中我们遵循了“Washington Group Short Set”指南的六个维度。<sup>2</sup>

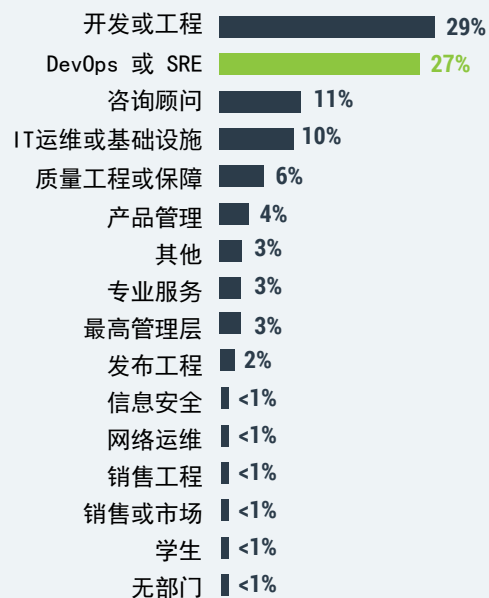


弱势群体的认定标准可以是种族、性别或其他特征。这是我们第二年收集这一数据，并发现了轻微的增长，从2017年的12%上升为2018年的13%。

<sup>2</sup> <http://www.washingtongroup-disability.com/wp-content/uploads/2016/01/The-Washington-Group-Short-Set-of-Questions-on-Disability.pdf>

## 受访组织统计

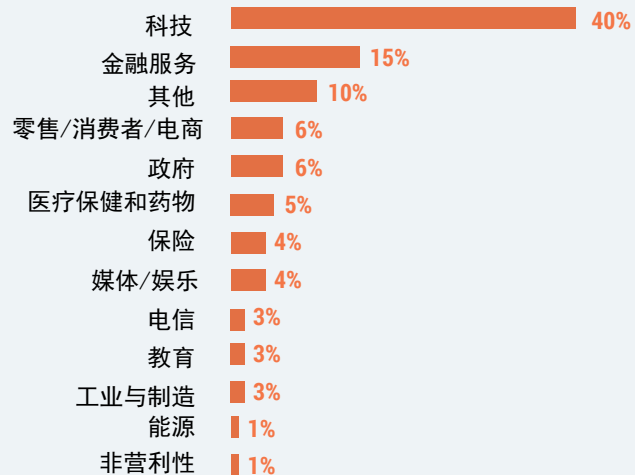
### 部门



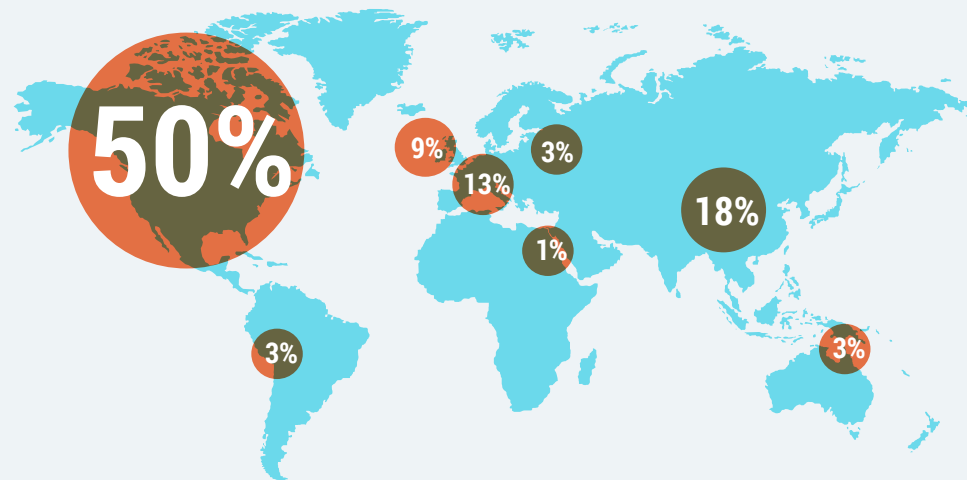
自从我们开始此项研究以来，在DevOps团队里工作的受访者的比例有所增长。

2014年是16%，2015年是19%，2016年是22%，2017年与2018年保持在27%。

### 行业

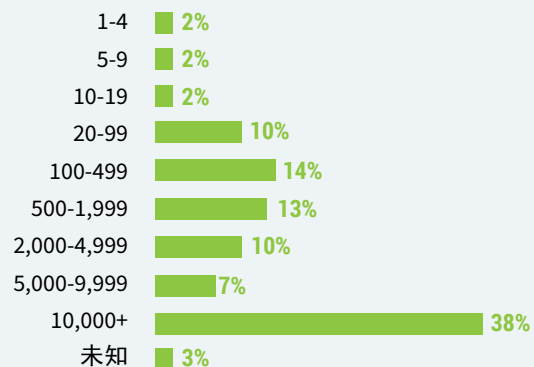


### 地区

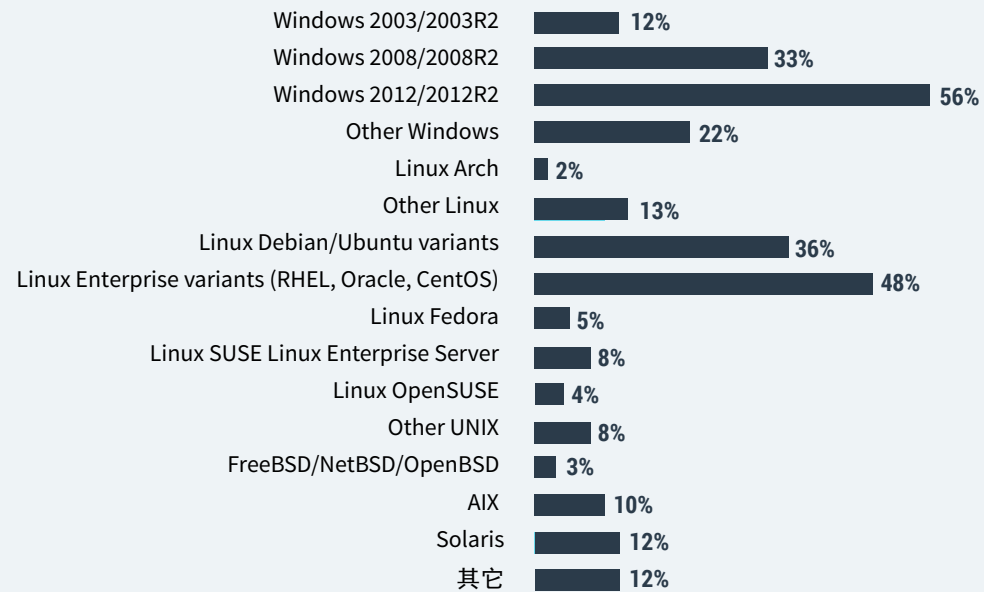


# 受访组织统计

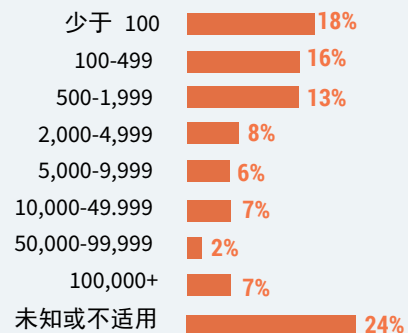
## 员工数量



## 操作系统



## 服务器数量





# 如何进行比较？

你可以将本章视为你的DevOps基准评估。我们以一种统计上有意义的方式，确保我们理解了各个团队是如何进行开发、交付及软件系统运维的。在本报告的多项重要分析中，你可以分别对照高效能、中等效能、低效能基准，了解自己所处的位置。此外，我们还介绍了逐年的发展趋势。

# 软件交付效能

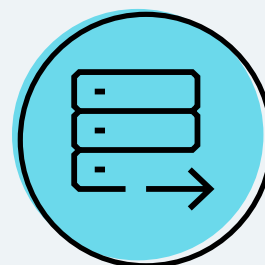
今年的报告反映出了软件和技术行业的一个根本性变化：早前的研究里称为“IT效能”的概念现在已改称为“软件交付效能”，从而区别于IT服务台和其他支持职能工作。运用这种高价值效能的能力是企业的一个关键竞争优势。那些能更快地开发和交付的企业，能更好地开展实验以便提高客户采用率和满意度，在需要的时候改变航向，并紧跟监管及合规要求的变化。

我们的分析显示，无论是哪个行业，无论是哪个细分领域，无论是否需要遵从高强度的监管，任何团队都有能力达到高水平的软件交付效能。我们把不同的团队划分为高效能、中等效能和低效能，并发现在所有组织类型和垂直行业领域中，都可以这样划分。在整个报告中这种分类方式会被频繁引用。

为了了解软件交付效能，我们使用了一些聚焦于全局产出（outcome）而非工作输出（output）的度量指标，以确保不会在不同的团队间产生冲突。和前几年一样，我们使用了4个软件交付效能的度量指标，用来展现软件交付的2个方面，如右图所示。

在吞吐量（throughput）和稳定性（stability）之间进行取舍（权衡）是一种常见的行业实践，尤其是在政府或强监管领域里。但是我们多年的研究得出了一致的结果：最优秀的高效能组织总是能在吞吐量和稳定性上同时达到卓越的水平，而不是在两者中进行取舍。事实上，吞吐量和稳定性是相辅相成的。连续第五年，聚类分析向我们展示了不同效能水平的组织在吞吐量和稳定性两个方面都存在巨大差距。在吞吐量和稳定性的所有方面，最高效能者都表现卓越，而中等效能和低效能者则都落在后面。

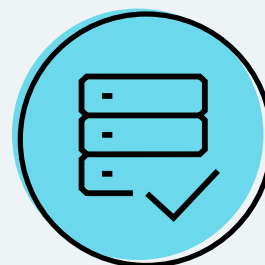
## 软件交付效能



吞吐量

部署频率

变更前置时间



稳定性

服务恢复时间

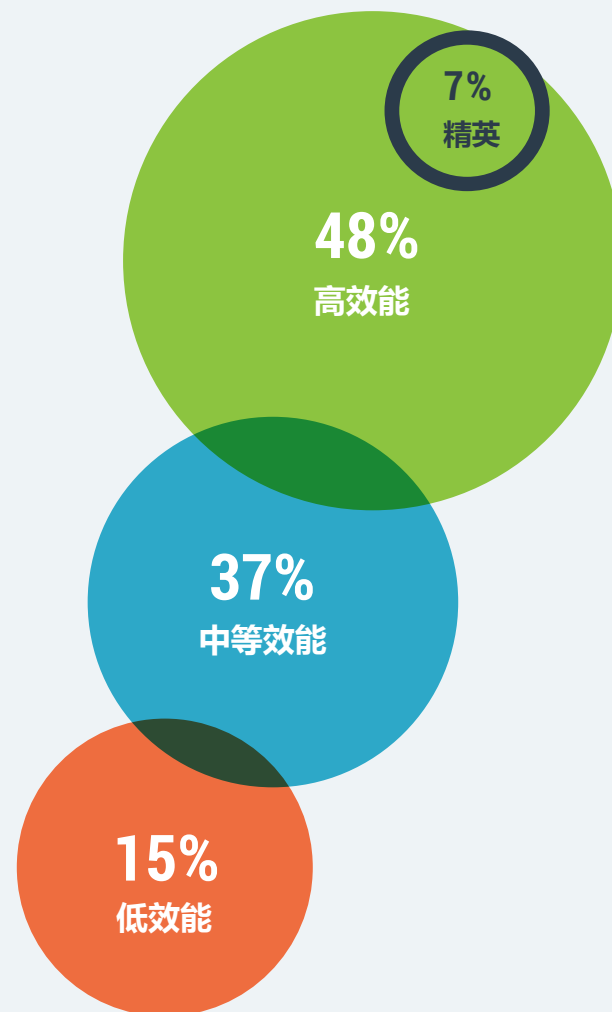
变更失败率

然而，今年我们发现中等效能组织表现出了有趣的行为，并在其中找到了权衡取舍的例证。中等效能组织在稳定性方面表现不错（与高效能组织很接近），然而在速度方面却明显落后。正如预测的那样，低效能组织在各个方面都再次落后于其他群体，几年来一贯如此。

## 新的精英效能组织提高了标准

今年的数据揭示出了第四个高效能群体：精英效能组织。增设这一新的类别有两个原因。首先，我们看到了高效能组织能力的提升和群体的扩大，这表明整个行业的软件开发和交付实践正在改善。这个趋势也标志着行业里的很多团队都可以实现高等效能，这不再是少数性质特殊的团队的专属领域。第二，精英组织体现出整个行业的标杆在不断提升，效能最出色的组织仍在持续优化吞吐量和稳定性。

### 效能特征



高效能组织的比例在逐年增长，这表明了行业本身正在持续优化。正如我们在过去几年中所观察到的那样，我们仍然看到那些效能最优异的组织（比如高效能组织中的部分“精英效能组织”）正在以最高水平开发和交付软件。与此同时，那些低效能组织正在痛苦的追赶，可是差距在不断扩大。

| 软件交付效能的度量指标                                                                            | 精英 <sup>a</sup> | 高效能            | 中等效能                    | 低效能                                    |
|----------------------------------------------------------------------------------------|-----------------|----------------|-------------------------|----------------------------------------|
| <b>部署频率</b><br>对于你负责的主要应用或服务，你的组织部署代码的频率？                                              | 按需（每天多次部署）      | 介于每小时1次和每天1次之间 | 介于每周1次和每月1次之间           | 介于每周1次和每月1次之间                          |
| <b>变更前置时间</b><br>对于你负责的主要应用或服务，变更的前置时间是多长（即从代码提交到代码成功运行在生产环境需要多长时间）？                   | 小于1小时           | 介于1天和1周之间      | 介于1周和1个月之间 <sup>b</sup> | 介于1个月 <sup>b</sup> 和6个月之间 <sup>b</sup> |
| <b>服务恢复时间</b><br>对于你负责的主要应用或服务，发生服务故障时（例如：计划外中断、服务受损），恢复服务通常需要多长时间？                    | 小于1小时           | 小于1天           | 小于1天                    | 介于1周和1个月之间                             |
| <b>变更失败率</b><br>对于你负责的主要应用或服务，有多大比例的变更会导致服务降级或需要事后补救？（例如：导致服务受损或服务中断，需要热修复、回滚、前向修复、补丁） | 0-15%           | 0-15%          | 0-15%                   | 46-60%                                 |

由于不是正态分布，所以使用中位数报告。  
 除非另有说明，否则基于Tukey的事后随机分析，所有差异都有显著差异

<sup>a</sup> 精英分类是高效能分类的子集

<sup>b</sup> 根据Tukey的事后随机分析，平均值没有显著差异；中位数因底层分布而表现出差异。典型的低效能企业的变更前置时间在1个月到6个月之间，典型的中等效能企业的变更前置时间在1个星期到1个月之间；然而，针对显著差异的测试表明，总体而言，当包括所有分类企业的行为差异时，这两个分类在统计上没有太大差异<sup>o</sup>

本报告中文版由DevOpsDays中国社区翻译制作

在对比精英效能组织与低效能组织时，我们发现  
**精英效能组织……**



代码部署频率  
高46倍



从提交到部署的前置时间  
快2555倍



变更失败率  
低7倍  
(变更失败的可能性只有其1/7)



故障恢复时间  
快2604倍

## 被“保守策略”误导的组织

我们经常听说组织更愿意采取保守型的软件开发和交付策略。这些组织向我们及其利益相关方保证，低频次地发布代码是一个有效的策略。因为这样在各次部署之间就有更多的时间用于测试和质量检验，从而将失败/故障的可能性降到最低。

我们今年的聚类分析揭示了一个明显受保守策略误导的效能群体，他们的速度相对较慢（部署频率及变更前置时间），同时变更失败率优于低效能群体。然而，这个群体的服务恢复时间却是最长的。

在日益复杂的系统中开展软件开发困难重重，失败/故障在所难免。然而，大体量的低频变更会给部署环节带来风险。一旦出现了失败/故障，找到问题的根源并恢复服务将非常困难。更糟糕的是，部署还可能会在整个系统里引发一连串其它的失败/故障。

### 被误导的组织

|        |            |
|--------|------------|
| 部署频率   | 每1-6个月部署一次 |
| 变更前置时间 | 1-6个月之间    |
| 服务恢复时间 | 1-6个月之间    |
| 变更失败率  | 16-30%     |

全面恢复这些次生失败/故障需要的时间更是惊人。然而，还是有很多组织依然坚持认为，他们不会遇到这种常见的故障情况。根据数据，我们发现5%的团队正属于这种情况，他们也承担着相应的后果。

乍一看，恢复系统故障耗时1到6个月似乎有些可笑。但是，考虑到由于系统中断而引发一系列的故障和数据损坏，可能还会有多个未知的系统受到牵连而遭殃，完全恢复整个系统耗时数月时间也并非不合理。

客户可能能够在几个小时或几天后恢复使用系统，但系统工程师和运维专家却需要调查故障的所有影响因素，校验并补救数据丢失或不一致的情况，全面将系统恢复到正常运行的状态。

2013年10月healthcare.gov的宕机事件，就是这种情形的一个典型案例。当系统遭到入侵者侵害，调查和补救将需要更长时间。这期间事故处理工作强度会很高。当每年发生多次此类状况时，这将会严重影响团队的工作重心，进而导致非计划工作变成常态。团队会因此而疲于奔命，这是团队和领导者需要慎重考虑的问题。

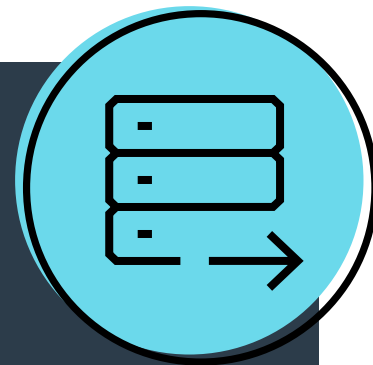
# 吞吐量

## 部署频率

精英组织报告说，他们经常会按需部署，并且每天都会做多次部署。相比之下，低效能组织一周到一个月部署一次；这一数据比去年有所提高。与去年相同的是，高效能组织的标准年度部署次数为1460次（按照一天4次部署\*365天）；而低效能组织的标准年度部署次数为32次。基于这项分析，我们可以看到，精英组织部署代码的频率是低效能团队的46倍。

## 变更前置时间

同样地，精英组织正在优化前置时间，从提交代码到代码成功部署到生产环境的时间不到一个小时；而低效能组织则需要1到6个月的前置时间。如果按照精英组织需要60分钟前置时间（对于“不到一个小时”，这是保守的估算）、低效能团队需要26940分钟前置时间（对1个月43800分钟和6个月262800分钟求平均数而得）来计算，精英组织比低效能组织在变更前置时间方面快2555倍。

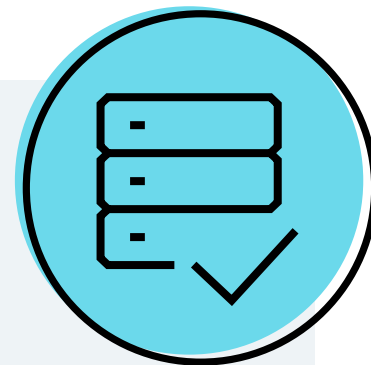


与很多公司的实践相比，每天4次部署是一个保守的估计。例如：[Capital One](#) 每天部署50次，<sup>3</sup>

或者例如Google以及Netflix每天部署几千次（生产环境里的数百项服务的合计值）。

<sup>3</sup> Capital One银行的持续交付：

[https://youtu.be/\\_DnYSQEUtfo](https://youtu.be/_DnYSQEUtfo)



## 稳定性

### 服务恢复时间

根据精英效能组织的反馈，其服务恢复时间一般在1小时以内，而低效能组织则在1周到1个月之间。我们采用了比较保守的计算方法：对精英效能组织取值1小时，对低效能组织取1周（168小时）和1个月（5040小时）的中间值。基于这个计算方法，精英效能组织的服务恢复速度是低效能组织的2604倍。如前所述，在本年度的调查中，低效能组织的服务恢复时间相比去年有所倒退。

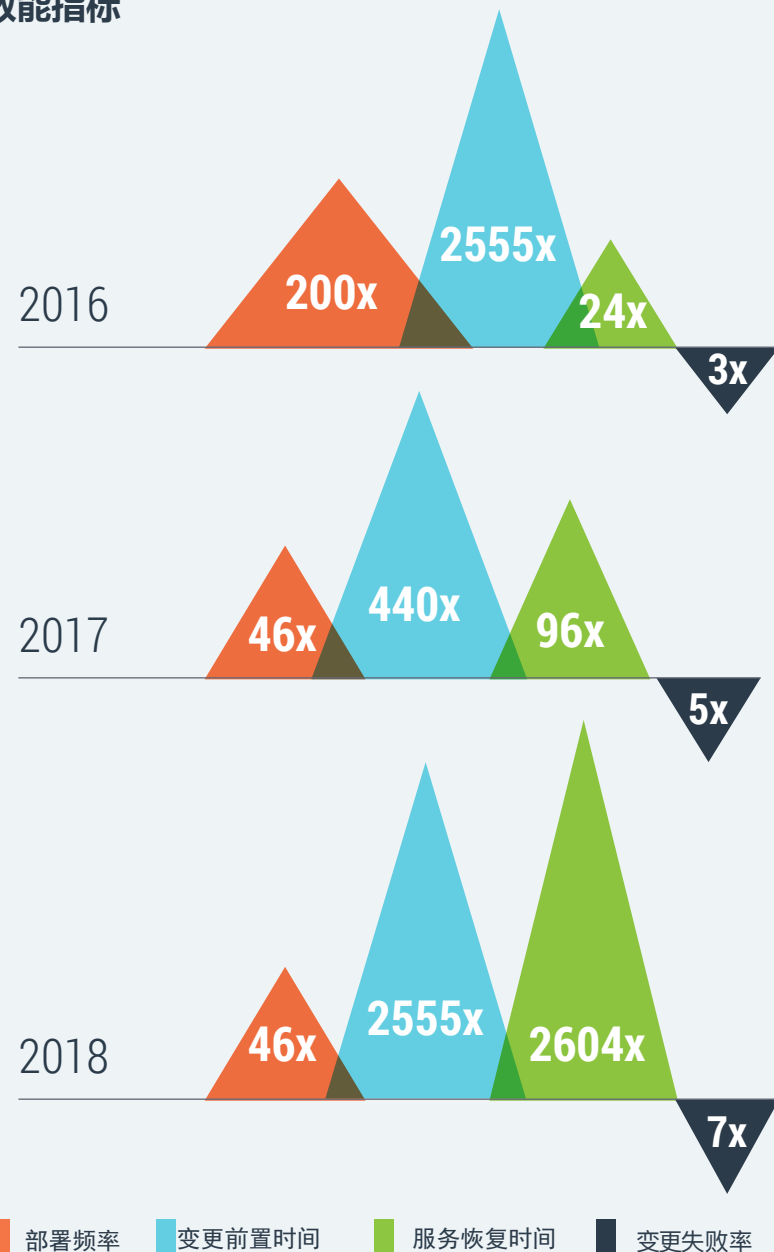
### 变更失败率

根据精英效能组织的反馈，其变更失败率指标在0%到15%之间，而低效能组织则在46%到60%之间。我们根据他们的反馈取中间值的结果为，精英效能组织的变更失败率为7.5%，低效能组织的变更失败率为53%。这一结果说明精英效能组织在这个指标上领先低效能组织7倍。如前所述，在本年度的调查中，低效能组织在变更失败率这个指标上相比去年有所倒退。

右侧图表比较了每一年的各项效能指标的最高值和最低值之间的差异，所显示的所有数据均为相对值。如果比较一下过去几年的差异数据，你会发现有些差异在扩大而有些在缩小。从2017年到2018年，这些指标的差异均在扩大或者保持不变。而那些扩大的差异都是由于低效能组织的指标数值下滑，这一表现可能是由于环境复杂度的增加让交付软件变得更为困难而造成的。我们还特别注意到了变更失败率这一指标所揭示的一种趋势：高效能组织的变更失败率持续保持较低水平，而低效能组织则愈发受制于变更对他们系统的影响。这种表现说明构建具备高复原能力的系统变得越发重要；或者按照Richard Cook博士的说法，<sup>4</sup>在构建系统的过程中就将失败的情况纳入考量，这非常重要。

<sup>4</sup> <http://web.mit.edu/2.75/resources/random/How%20Complex%20Systems%20Fail.pdf>

## 效能指标



## 软件交付和运维效能：新增可用性指标

在今年的调查中，我们新增了另一个对于组织非常有价值的软件效能指标：可用性。从较高层面来说，可用性代表着技术团队和组织信守承诺并确保软件产品或服务可用的能力。对于最终用户而言，这意味着软件产品或服务是否能被访问和使用。我们的可用性度量范围包括：团队如何定义可用性目标，如何从服务中断中学习，如何确保反馈环的闭环性。这些用于评估可用性的指标构造了一种坚实可靠的度量方式。

分析表明，可用性指标与软件交付效能呈现强相关；精英效能组织和高效能组织表现出了卓越的可用性，其中精英效能组织具备强有力的可用性实践的比例比其他组织高3.5倍。

基于对业界的观察和研究，以及对这些结果的分析，我们在交付效能模型中增加了可用性指标。<sup>5</sup>

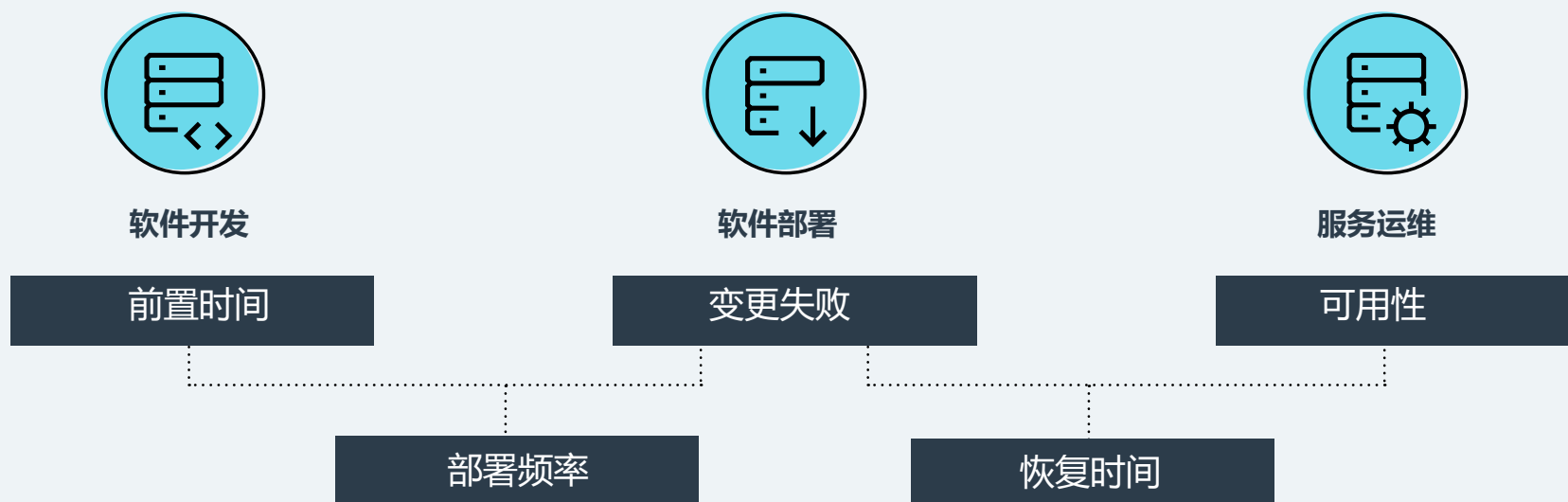
<sup>5</sup> 在我们的软件交付模型中增加可用性指标的原因是，这个指标对于预测组织的整体效能（包括：盈利能力、产能和客户满意度）有所帮助，但我们并没有将其包含在效能分类指标中（也就是说没有包含在我们的分析矩阵中）。这是因为可用性对于那些非服务类解决方案（比如：套装软件和固件）是没有意义的。

在下图中，你可以看到，在原有软件开发和交付指标的基础上引入可用性指标后，我们可以更全面地揭示现今的软件开发、交付和运维状况。在NIST<sup>6</sup>（美国国家标准与技术研究院）的相关文件中，我们也发现了相关佐证，他们对可用性的定义是“确保信息可被及时且可靠访问的能力”。我们将这一新架构称为 软件交付和运维效能（简称SDO效能），同时我们发现它有助于企业提升组织效能。<sup>7</sup>

<sup>6</sup> 来自NIST（美国国家标准与技术研究院）的特别出版物800-12r1：“信息安全概述”

<sup>7</sup> 我们也特意指出，团队将SDO中的S理解为软件（software）或者服务（service）均可。

## 效能度量指标





# DEVOPS 重要吗？

直觉告诉我们DevOps很重要：技术转型可以促进业务产出和质量提升。我们听到很多组织的故事，讲述他们如何利用技术实现效率、利润和客户满意度的改善。但是故事和直觉不足以支撑持续性的投资；我们需要的是证据和数据。

我们的分析表明，在技术转型期间采纳DevOps实践和能力的组织，都在效能和质量成果方面获得了满意的回报。

## 组织效能

在任何行业中，团队和组织的SDO效能都是关键的价值驱动因素和差异化因素，因为它使组织能够利用软件来交付更出色的成果。这些成果通过许多因素来衡量，包括生产力、盈利能力和市场份额，以及效用、效率和客户满意度等非商业衡量指标。我们的分析表明，精英组织达到或超过其组织效能目标的比例要高1.53倍，而高效能组织达到或超过其目标的比例要高1.38倍。

我们的研究连续第五年发现，软件交付效能是组织效能的重要组成部分。我们对组织效能的度量指标参考了学术文献，并把握了达成或超越组织使命目标的两个方面：商业目标和非商业目标。<sup>9</sup>

---

<sup>8</sup> Widener, S. K. (2007). 一个控制框架杠杆的实证分析师。会计、组织和社会家, 32(7-8), 757-788.

<sup>9</sup> Cavalluzzo, K. S., & Ittner, C. D. (2004). 实施效能衡量创新:来自政府的证据。会计、组织和社会家, 29(3-4), 243-267.

商业和非商业的目标包括：

- 盈利能力
- 生产力
- 市场份额
- 客户数量
- 产品或服务数量
- 运营效率
- 客户满意度
- 提供的产品或服务的质量
- 实现组织或使命目标

分析表明，软件交付效能是理解组织效能的重要因素。今年我们在模型中新增了可用性指标，建立了用于预测组织效能的二阶构造。

我们新的软件交付和运维效能二阶构造与单独的软件交付效能或可用性指标相比，能更好地预测组织效能。

## 为什么要关注组织目标而不是具体数值呢？

**这种度量方式让我们能够从各行各业、各种规模的公司收集数据。**

小型创业公司与大型企业集团之间比较具体数值毫无意义，其营业额、利润和收入水平都差异巨大。使用复杂的财务比率可能有助于缩小差异，但不同行业的财务比率也不尽相同。与此相对，基于组织目标的度量方式则可以让我们获得的反馈具备跨行业和组织规模的可比性。

**受访者可能不知道利润或营业额方面的具体数值。**

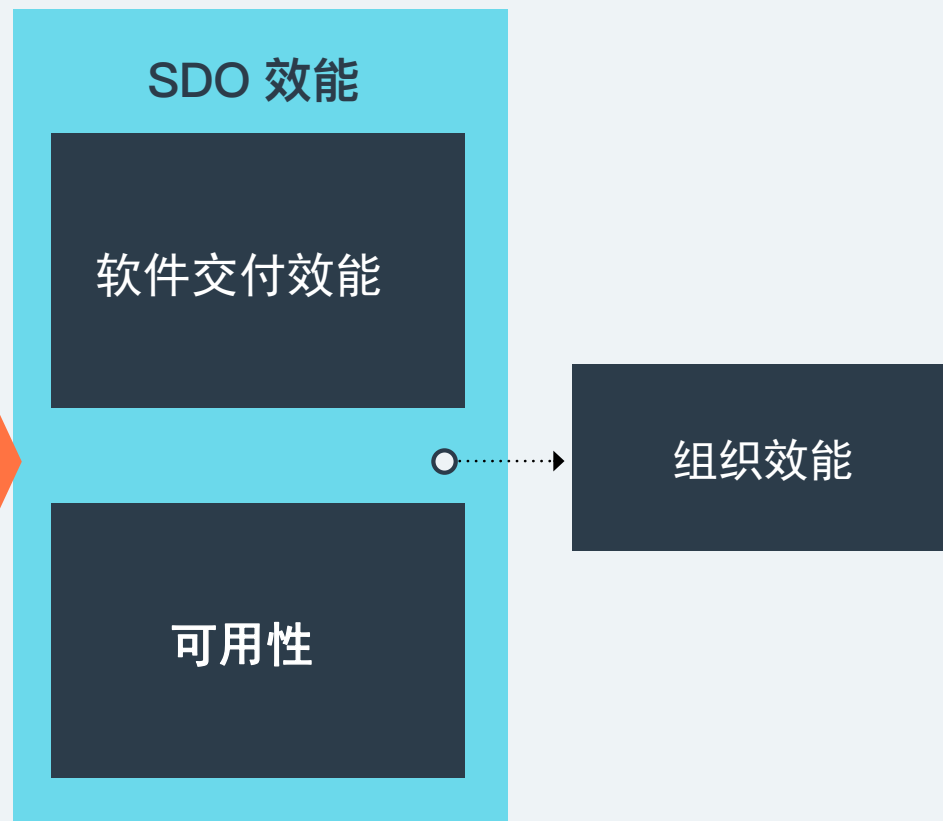
销售目标或客户满意度目标的达成情况通常在各种规模公司里都是人人皆知的信息。

**组织目标达成的程度，体现了领导对市场的了解，以及驾驭业务的水平。**

股票市场会奖励那些达到或超过其收益目标的上市公司，但不认可那些大幅超过其收益目标的公司，因为这表明领导者不了解他们的市场或业务。使用具体数值的度量方式不能体现这一点。

我们的结构化方程模型（SEM）在整个报告中被多次使用，它是用于测试各种关系的预测模型。每个方框代表我们在研究中度量的一种构造，每个箭头代表构造之间的一项关系。内含方框（构造）的较大方框是二阶构造（second - order construct ）。

在解读这个模型时，所有的箭头可以理解为预测、影响、驱动或作用。在此例中，二阶构造“SDO效能”由软件交付效能、可用性这两个构造组成，而这两个因素共同驱动组织绩效。**我们通过粗体标记，表明可用性是今年新加入调研范围的一个构造。**



## 优质产出

开展技术转型的团队和组织，同样也有提高质量的目标。然而，对质量的度量很具挑战性，因为它受制于场景限制，而且很多度量指标都因行业甚至企业不同而异。<sup>10</sup>

尽管制定适合于所有组织的质量指标有难度，我们仍然可以制定出一些跨企业和跨行业也同样有效的间接质量指标。这些指标包括了对时间资源的使用，因为据此可以得知组织用在增值工作和非增值工作上的时间比例。在这项调研中，我们考察了人工操作、计划外工作/返工、安全修补以及客户支持等工作所占据的时间比例，结果非常具有启发性。分析表明，在所有行业中，高效能组织相比低效能同业者，花费在人工操作上的时间明显少很多，投入到新工作的时间更多，用于修复安全问题或缺陷的时间也更少。由于这些组织从一开始就把控好质量，它们用于在下游修复问题的时间更少，节省出了更多时间来做增值工作。

<sup>10</sup> 此概念由软件质量专家Jerry Weinberg在他的《质量软件管理第1卷：系统思考》（Dorset House 出版社，1992年）书中提出。

## 人工操作

通过将重复性任务或可并行处理（因而可以加速）的任务自动化，团队和组织可以提升工作的质量、可重复性和一致性，将员工从低价值任务中解放出来。随着越来越多的工作得到自动化，高效能组织可以释放出更多的技术人员来从事那些可增加真正价值的创新性工作。

但是，我们了解到，估算工作的自动化水平非常困难，而估算仍然依赖人工操作的工作比例会更容易一些。这其实并不奇怪。人工操作很痛苦，所以员工对其有鲜明的切身感知。一旦工作被自动化了，它就不再痛苦，往往也就不再受关注了。

当我们将高效能组织与其低效能同业者进行比较时，我们发现，精英组织和高效能组织的人工工作量水平在所有维度都低于低效能同业且差距显著，而中等效能组织的人工工作量在所有维度都是最高的。



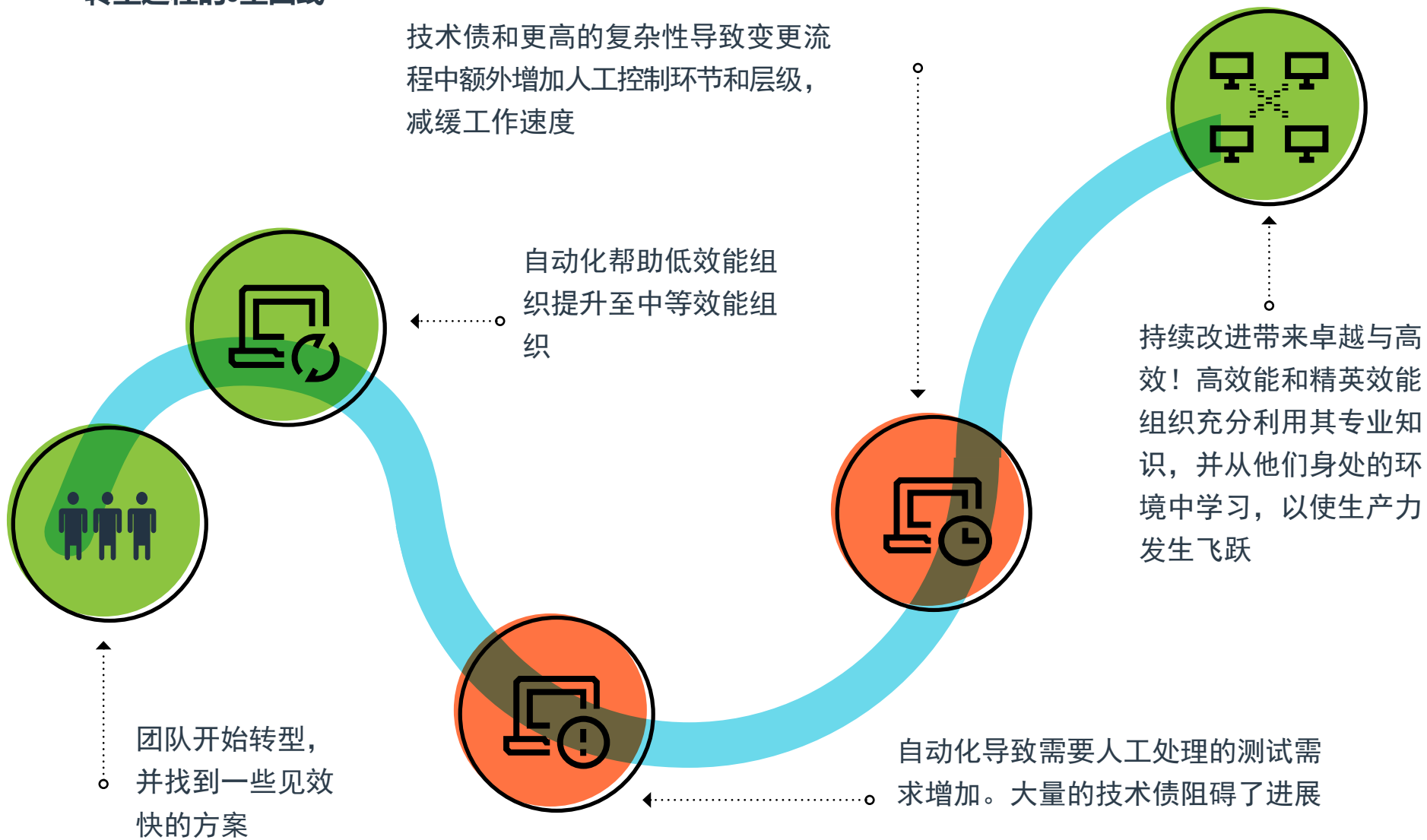
读者可能会惊讶地发现，中等效能组织在测试和变更审批流程方面相比低效能组织做了更多的人工操作，存在着明显差距。然而，我们在去年的数据中也发现了相似的模式，当时是中等效能组织在部署和变更审批流程方面比低效能组织做了更多人工操作（同样存在明显差距）。我们已经多次听说并亲眼目睹了正在转型中的团队确实存在这种情况。下一页上的J型曲线图表介绍了这一过程。

| 人工操作 | 精英  | 高效能 | 中等效能             | 低效能              |
|------|-----|-----|------------------|------------------|
| 配置管理 | 5%  | 10% | 30% <sup>a</sup> | 30% <sup>a</sup> |
| 测试   | 10% | 20% | 50%              | 30%              |
| 部署   | 5%  | 10% | 30% <sup>b</sup> | 30% <sup>b</sup> |
| 变更审批 | 10% | 30% | 75%              | 40%              |

使用了中位数，因为分布不是正态的。

a,b 使用Tukey检验法进行事后方差检验并未发现显著差异。

## 转型过程的J型曲线



## 时间是如何被消耗的

另一个度量工作价值和质量的方式是分析团队的时间使用方式。也就是说，在团队的工作过程中，他们是将时间和精力集中投入到开发新特性和支持基础设施的工作中，还是耗费了大部分的时间来修复缺陷、补救问题、回应缺陷和提供客户支持（也就是说，修复由于没有从一开始就把控好质量而引发的质量问题）？我们将工作分为两类。

第一类是前瞻性工作或新工作，在这类工作中，我们能够以有条理、有成效的方式，设计、构建和处理特性、测试和基础设施，为组织创造价值。

第二类是计划外工作或返工。中断、错误和被动响应推动着此类工作，使人们很难集中注意力完成任何事情。在工作过程中，目标是从一开始就把控好质量，但这很难度量。因此，我们改为寻找质量欠佳的证据；比如，是哪个环节的疏漏导致了错误？时间被消耗在返工、补救和客户支持上是质量不佳的标志，因为我们不得不花费时间去修复质量，而不是在最初就把控好质量。多数情况下，我们希望在第一类工作上花更多的时间，在第二类上花较少的时间。

---

<sup>11</sup> Deming, W. Edwards. *Out of the Crisis*. Cambridge, MA: MIT Press, 2000.

我们调查了受访者的时间分配情况。发现总体而言，精英效能组织在日常工作中做增值性工作的时间最多，非增值性工作的时间最少，其次是高效能和中等效能组织。低效能组织在所有维度上的表现都是最差的。

| 时间消耗        | 精英    | 高效能              | 中等效能             | 低效能              |
|-------------|-------|------------------|------------------|------------------|
| 新工作         | 50%   | 50%              | 40%              | 30%              |
| 计划外工作和返工    | 19.5% | 20% <sup>a</sup> | 20% <sup>a</sup> | 20% <sup>a</sup> |
| 修补安全问题      | 5%    | 5% <sup>b</sup>  | 5% <sup>b</sup>  | 10%              |
| 处理最终用户发现的缺陷 | 10%   | 10% <sup>c</sup> | 10% <sup>c</sup> | 20%              |
| 客户支持工作      | 5%    | 10%              | 10%              | 15%              |

使用了中位数，因为分布不是正态的。

a 当使用Tukey事后分析法测试差异时，差异显著不同

c 当使用Tukey事后分析法测试差异时，差异没有显著不同

# 如何改进？

一旦了解了和同行之间的差距，以及效能改善的作用，下一步就是如何运用这些知识来进行改进了。我们在分析过程中识别出的能力，经统计数据验证，可明显提升软件交付和运维效能。你可以使用这些信息来进行沟通和推动改进举措，从而在效能方面更上一层楼。



# 云计算、平台和开源

Forrester预测<sup>12</sup>2018年全球公有云市场将达到178亿美元，较2017年上涨22%，并且福布斯报告显示<sup>13</sup>到2020年83%的企业工作负载将运行在云上。在我们的调查报告中，67%的受访者表示他们负责的主要应用或服务已经托管在某个云平台上了。今年的报告研究了主流云使用模式对SDO效能的影响，真正重要的是团队如何使用云服务，而不仅仅是使用了云。

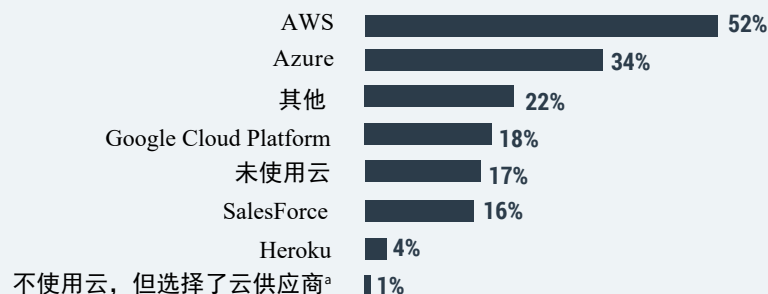
正如你所注意到的，这些百分数加起来已经超过100%。我们询问受访者其团队是否使用了多个云计算供应商以及其原因。<sup>14</sup>

<sup>12</sup> <https://www.forrester.com/report/Predictions+2018+Cloud+Computing+Accelerates+Enterprise+Transformation+Everywhere/-/E-RES139611>

<sup>13</sup> <https://www.forbes.com/sites/louiscolombus/2018/01/07/83-of-enterprise-workloads-will-be-in-the-cloud-by-2020/#173157906261>

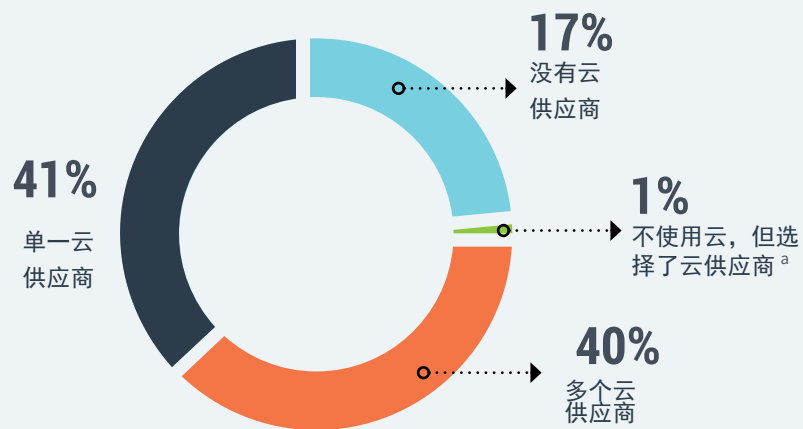
<sup>14</sup> 我们注意到，受访者报告的云使用比例与其他的报告数据很接近，比如2018年的云状况报告（2018 Right Scale State of the Cloud Report）。  
<https://www.rightscale.com/lp/state-of-the-cloud?campaign=7010g0000016JiA>  
还有就是AWS、谷歌云平台、微软Azure云平台的调查也支持我们数据的外部有效性。  
(<https://www.rightscale.com/lp/state-of-the-cloud?campaign=7010g0000016Ji>)

## 云计算供应商占比



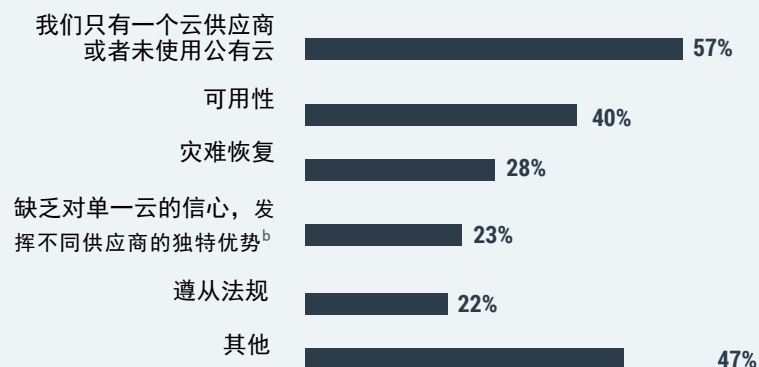
<sup>a</sup> 受访者表示他们不使用云，但又选择了云计算供应商。

## 使用多个云供应商的情况



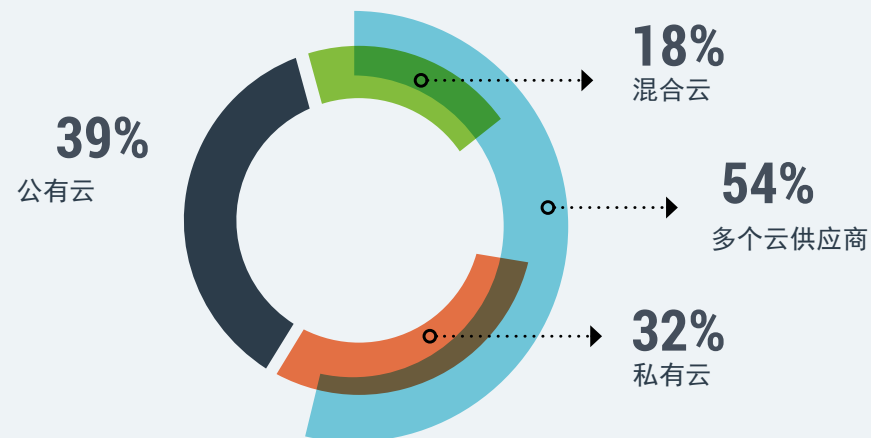
<sup>a</sup> 受访者表示他们不使用云, 但是他们选择了云计算供应商。

## 使用多个云计算供应商的原因



<sup>b</sup> 报告中, 我们错误的将两个选项 (“缺乏对单一云的信心”和“发挥不同供应商的独特优势”) 设置成了一个选项, 因此也只能作为一个选项进行收集统计, 对此我们表示歉意。

## 主要的产品或服务<sup>c</sup>



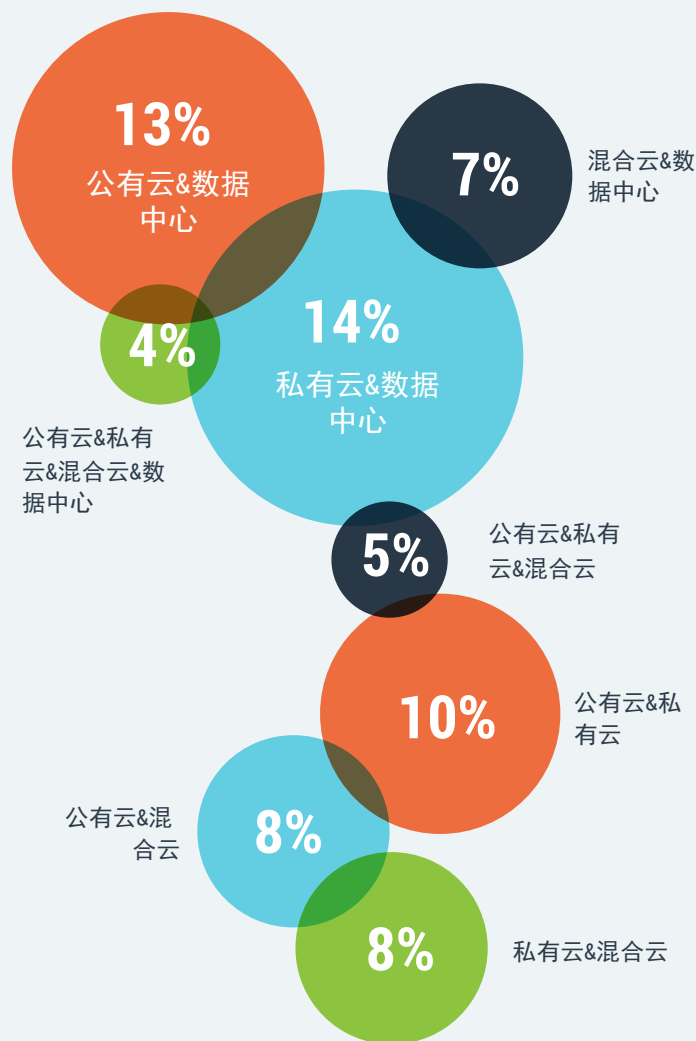
<sup>c</sup> 总和超过100%; 说明一些产品部署到了多个环境, 例如, 13%的受访者表示, 其主要的产品和服务同时运行在公有云和传统数据中心上。

读者可能会注意到一些不一致的地方：如何对受访者的应用进行公有云、私有云和混合云的分类？一项挑战是混合云通常是自行定义的：如果受访者说他们使用了混合云（同时使用了公有云和私有云），那么就是混合云。这些答案也可能反映出受访者同时负责多款应用，其中一部分使用了公有云，一部分使用了私有云，其余使用混合云环境。

我们还注意到数据中心不一定是私有云。组织可以在其数据中心构建私有云，但反过来，数据中心可以按传统方式管理，这种方式不符合下面描述的任何基本的云计算特性。

我们的行业对这些定义的使用方式并不统一，这可以解释为什么报告中的有些内容和我们的经验不同。也就是说，由于定义和度量的差异，我们各自的经验可能存在差异。在下一章节讨论云计算时，我们会专门讨论这个问题。

## 云类型使用范围的重叠



## 如何实施云基础设施很关键

在那些对其团队具备所有云计算核心特征表示同意和非常同意的受访者里，来自精英效能组织的比例比低效能组织大约高出23倍。同样的，使用平台即服务（PaaS）的用户来自精英效能组织的比例高出1.5倍，采用云原生设计实践的受访者来自精英效能组织的比例高出1.8倍。使用基础设施即代码服务来管理其云部署的用户来自精英效能组织的比例高出1.8倍，而使用了容器的受访者来自精英效能组织的比例高出1.5倍。

然而，很多表示他们正在使用云计算的受访者，其实并没有真正应用到那些重要的云计算核心模式，而这会拖他们的后腿。NIST定义了云计算的五个核心特征（参见下一页），但在使用了云计算基础设施的受访者中，只有22%同意或非常同意其团队具备所有这些特征。<sup>15</sup>

<sup>15</sup> NIST 特别刊发 800-145: “NIST的云计算定义”



# 23 倍

具备所有云计算核心特征的团队来自精英效能组织的可能性要高23倍。

在对采用云计算的真正内涵进行定义的时候，这些特征至关重要。我们的调研表明它们会影响软件交付效能。

那些同意或非常同意其团队具备所有云计算核心特征的受访者，其来自精英效能组织的可能性比低效能组织高出23倍。

低门槛的网络接入和按需自助服务经常被忽视，但却特别重要，因为它们直接影响到消费方的效能产出。

例如，在一些组织的云实施方案中，用户仍然需要先提交申请表才能访问一些关键资源并完成工作，或者不能从其设备轻松访问云系统。从消费方的角度看，这与使用传统的数据中心也没什么差别。这个巨大障碍会阻碍团队实现交付流程效率的提升，并从而成为更高效能的团队。

## 云计算的五个核心特征



## 平台即服务PaaS

另一种为应用开发者提供更优服务的方法是实施平台即服务（PaaS），即“消费方不用管理或控制底层的云基础设施，包括网络、服务器、操作系统或存储空间，而是控制所部署的应用，及使用托管环境所提供的配置和设置”<sup>16</sup>。

PaaS的例子包括Heroku、RedHat OpenShift、Azure App Service、Google App Engine、AWS Elastic Beanstalk 和 Cloud Foundry。

只有24%的受访者表示正在使用PaaS。但是，在PaaS上完成大部分工作的受访者来自精英效能组织的可能性要高出1.5倍。

这些受访者同意或非常同意其团队使用由PaaS定义的库和基础设施作为其应用的基础，可以按需将其应用一键部署到云上，而且可以对其应用所需的数据库和其他服务按需进行自助变更。

<sup>16</sup> NIST 特别刊发 800-145：“NIST的云计算定义”

## 基础设施即代码

DevOps运动的一项重要创新是“基础设施即代码”的理念。在这个范式中，我们以自动化的方式从版本控制库里提取用来重建和变更环境状态的代码，而不是人工配置基础设施。

这种工作方式非常适用于云基础设施，可以通过API制备和配置资源。Terraform之类的工具使用声明式、版本控制的配置，可轻松地制备和持续更新云基础设施。通过这种方式，制备测试和生产环境变得快速而可靠，从而改善了云基础设施的管理员和用户的产出。类似的技术可用于自动化部署应用程序。

在我们的研究中，44%的云采用者同意或非常同意其团队只使用存储在版本控制库中的脚本和信息进行环境的配置和部署，而不需要人工操作环节（除了审批）。

使用基础设施即代码范式的受访者，其来自精英效能组织的可能性要高出1.8倍。



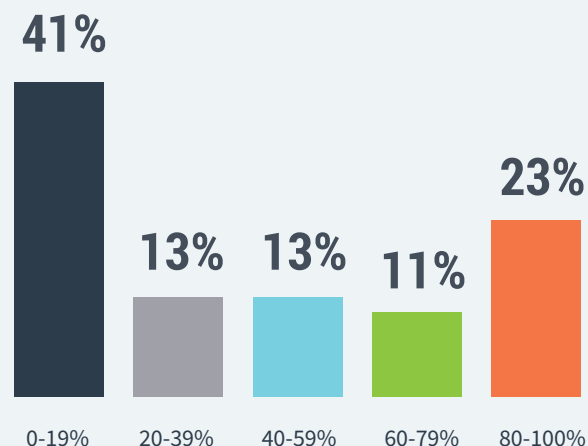
## 云原生

根据云系统的固有限制而专门设计的应用程序被称为云原生。这些应用在几个关键的方面都不同于那些面向传统数据中心而设计的应用。<sup>17</sup>重要的是，云中的系统假定运行在不可靠的底层基础设施上，并且必须进行容错设计。这意味着云原生应用必须具有复原能力，能够动态地响应工作负载的变化（即弹性），并且易于部署和按需管理。

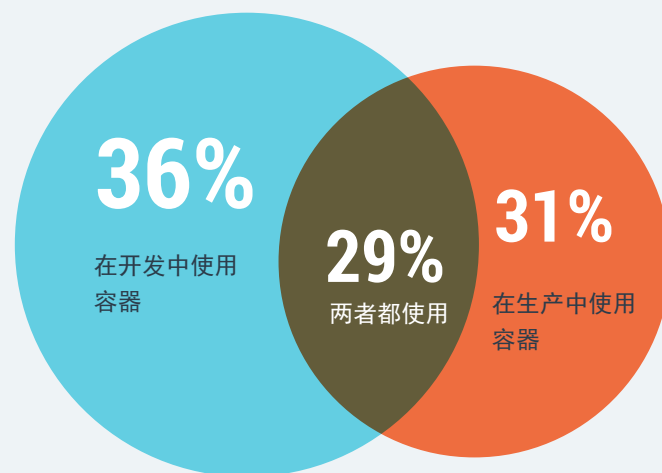
在将应用部署到云端的受访者中，47%从一开始就是为运行在云端而设计和架构的。虽然不是所有的高效能团队都运营云原生应用，但这样做的团队来自精英效能组织的可能性要高1.8倍。最后，我们询问受访者是否在使用容器。在生产环境中使用容器的受访者来自精英效能组织的可能性要高1.3倍。

<sup>17</sup> 以下网站提供了一份广泛流传的云原生应用设计指南 <https://12factor.net/>

在云上运行的产品/服务是云原生的比例



容器的使用



## 开源

在过去的20年间，开源软件被越来越多的团队所广泛应用。在这次调查中，58%的受访者表示他们所在的团队已经应用了大量开源的组件、库和平台，还有超过50%的受访者表示，团队正计划引入和应用更多开源软件。

报告显示，精英效能团队应用开源组件、库和平台的比例是低效能团队的1.75倍，计划引入并应用更多开源软件的比例是低效能团队的1.5倍。

## Capital One的开源经验

Capital One(COFC, Capital One Financial Corp, 美国第一资本投资国际集团)是美国十大银行之一，以其为本土及全球客户所提供的优质专业的定制化产品和创新的服务方式而闻名。这家公司的专业技术人员之所以能够更好地工作并交付更好的产品和服务给客户，能够成功转型为更有效率的高效能组织，都完全得益于他们在很早的时候就采纳了敏捷和DevOps的实践。

曾经和公司众多工程团队深入共事的Capital One资深工程研究员Tapabrata Pal博士说：“在Capital One，我们秉承‘开源优先’的原则。我们在开源基础上构建和运行软件，并积极为开源社区做贡献。我们也发布了自己的开源项目，其中最值得一提的是我们屡获殊荣的DevOps仪表板项目——Hygieia。我们认为必须要接受、采用和拥抱开源，因为这是一个必然趋势。对于我们来说，这不仅仅是开源的软件，它更是一种文化——在组织中协作构建软件的文化。得益于这种开源运用方式和理念，Capital One得以更快速、更可靠、更高质量地向客户交付软件，使我们能够更好地为客户服务。”

## 外包

我们在传统上视“外包”为一种快速获取能力和资源的途径。除了能够解决短期项目或难以招聘到合适人员的项目的需求，外包还有助于降低成本；它减少了对内部员工的需求，并提供了技术劳动力的弹性。将某个独立的职能（例如：应用程序开发、测试和质量保证、服务运营）委托给外部供应商是一种流行的外包模式。然而，这种模式增加了职能部门之间额外的交接需求和潜在的协作阻力。职能部门的职责划分可能会有损于敏捷性：一旦签订了合同，就很难再变更与外部实体（筒仓）之间的细节要求。

许多外包模式导致的交接和筒仓问题都被视为高效能的阻碍，受到了敏捷和DevOps社区的批评。今年，我们关注了外包实践对软件交付和运维（SDO）效能的影响。我们调查了受访者在应用程序开发、测试和质量保证，以及IT运维等方面工作外包的比例。

分析表明，低效能团队使用职能外包的比例（总的来说）是精英效能团队的3.9倍，并且将以下任一职能的外包的比例是精英效能团队的3.2倍：应用程序开发、IT运维、测试和质量保证。这表明，精英效能组织很少采用职能外包。

让我们再深入了解一下职能外包的影响。根据我们掌握的精英效能团队和低效能团队的特征，我们可以量化和估算其对业务绩效的影响。首先，我们知道精英组织通常每天多次交付软件，而低效能组织则是每1到6个月才能交付1次软件。

## 外包对被误导组织的影响

让我们回顾一下在软件交付效能章节中提到的“被误导组织”：该群体的部署频率和变更前置时间比低效能组织更慢，或者不分伯仲。但他们的变更失败率好于低效能组织。但是报告也显示出他们的故障恢复时间也是最长的，平均宕机时间在1到6个月之间。

被误导组织的数据表明，他们使用外包的比例是最高的，这不仅拖累了他们的效能，而且从宕机事件中恢复的速度也非常慢。当在外包环境下工作时，为了修复由代码引起的问题，可能需要花费几个月时间来实施、测试和部署。

外包往往会导致批量化地处理工作，从而造成过长的前置时间。因为在外包组织中，从开发到测试再到运维的事务处理成本相当高。当工作按项目或版本分为不同批次时，高价值和低价值的特性会混合在每次发布中，这意味着所有工作——无论是高价值的或是低价值的——都是以相同的速度交付。

我们想强调这一点：重要的和关键的特性被迫等待那些低价值的工作，因为它们都被集成到了同一次发布中。我们确信，有许多专业人士已经在实践中看到了这一点：在大多数项目待办事项列表中，有些特性具有不对称的高价值。延迟这些高价值特性的成本（由于所有特性一起发布）通常也非常高昂。在许多情况下，这种延迟成本可能会超过外包所节省的成本。

让我们来看看世界上最大的集装箱海运公司马士基的实际案例。<sup>18</sup>在一个项目中，团队由于无法按时交付待办事项清单中的特性，每周都会给公司带来巨大的损失。

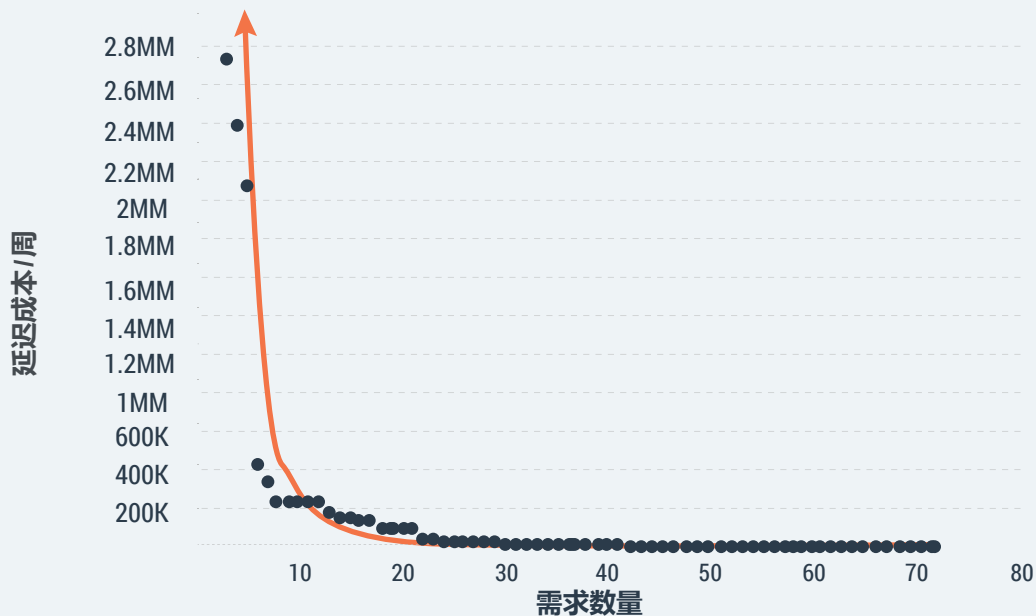
---

<sup>18</sup> 该案例及下一页的图示，源自Joshua Arnold和Özlem Yüce的“BlackSwanFarming使用延迟成本”，<https://blackswanfarming.com/experience-report-maersk-line/>

我们来看看马士基计算延迟成本的一个例子，并假设我们正在一个背景相似的精英组织中工作。

提醒一下，这意味着我们能够按需交付已开发完成的特性，而非几个月才能发布一批特性。仅图中的前三个特性，每周的延迟成本就高达约700万美元，即每月约3,000万美元。按数据所示，如果采用职能外包，并导致需要数月的前置时间才能交付特性，那么延迟成本完全有可能远远超出外包所节省的费用的。

马士基延迟成本幂律曲线



该幂律曲线<sup>19</sup>可以说是典型的产品待办事项列表

<sup>19</sup> 幂率分布展现了少数项目占资源的95%的现象。

参见<http://www.statisticshowto.com/power-law/>

当然，这不是提升SDO效能的唯一好处。高效能组织还可以在发生中断时更快地恢复服务、发布更稳定的版本、实现更出色的可用性，并产出缺陷更少且质量更高的软件。

我们对外包的研究结果中有一些值得注意的要点。这里提出的论点针对的是基于职能（开发、测试、质量保证、运营）的批量外包。这些研究结果证实了我们在行业内听到的许多关于外包导致不良结果的故事。（这些论点也适用于具有内部职能孤岛的组织）

但值得注意的是，我们的研究结果不一定能类推到其他涉及多个实体的工作模型。例如，这些论点不适用于委托合作伙伴或供应商来开发整个系统，并将从设计到运维的整个流程全都外包的情况。在这种情况下，成功的关键是服务之间的技术依赖关系是否得到了妥善管理<sup>20</sup>，且不会造成高价值特性的交付出现延迟。

---

<sup>20</sup> 如需更多详细信息，建议您参阅2017年DevOps报告中关于松散耦合架构重要性的内容。

<https://devops-research.com/assets/state-of-devops-2017.pdf>

此分析中未涉及的其他常见模型包括地理位置上分散的团队和所谓的“嵌入式”承包商模型。这些模型的一个主要区别在于“其他”团队（无论是内部分散的团队还是由承包和咨询公司所提供的额外员工）的运转和行事方式都等同于主体组织内部的跨职能产品或技术团队；如果能保持软件开发和交付的节奏，其结果也很可能得到保障。

事实上，我们从数据中看到了对该论点的佐证：高效能组织通过单个跨职能团队开发和交付软件的可能性高达两倍，这是我们在下一章节讨论的一项关键实践。



## 精益和敏捷实践

在组织想要提高效能时，通常会聚焦于采购（工具、供应商和方法形式的）解决方案。但是真正重要的不是解决方案本身，而是它能导入什么能力。过去四年中，我们致力于识别有哪些能力对我们所关注的产出有显著影响。今年，我们不但确认精益的产品管理方式对软件交付效能影响显著，还具体研究了团队的组织方式。

### 跨职能团队的重要性

跨职能团队的概念是许多敏捷方法的核心。根据Scrum指南描述，“Scrum团队由产品负责人、开发团队和Scrum Master组成。Scrum团队是自组织和跨职能的……跨职能团队拥有完成工作所需的所有能力，而不需要依赖团队之外的其他人”。<sup>21</sup>

在《解析极限编程：拥抱变化》（第二版）中，Kent Beck和Cynthia Andres写道：“让团队成员拥有项目成功所需的所有技能

---

<sup>21</sup> <https://www.scrumguides.org/scrum-guide.html#team>

和视角。”（第38页）事实上，我们发现低效能组织以孤立的筒仓团队的方式开发和交付软件的比例是精英效能组织的两倍。

## 精益产品管理

过去几年，我们研究了精益和敏捷产品管理实践对软件交付效能和组织效能的影响。这些实践在许多情境下都取得过成功，然而它们并不总是被采用。例如，以下情况依然普遍：在开始工作之前花费数月的时间来做预算、分析和收集需求；将工作批量打包成大项目并集中地发布；软件交付团队在如何完成其工作方面缺乏必要的指导；客户反馈不能真正融入开发流程。

## 精益产品开发方式的三个特征

1

团队在多大程度上将产品和功能分解成可在一周内完成的较小批量并做到频繁发布，其中包括最小可行产品（MVP）实践。

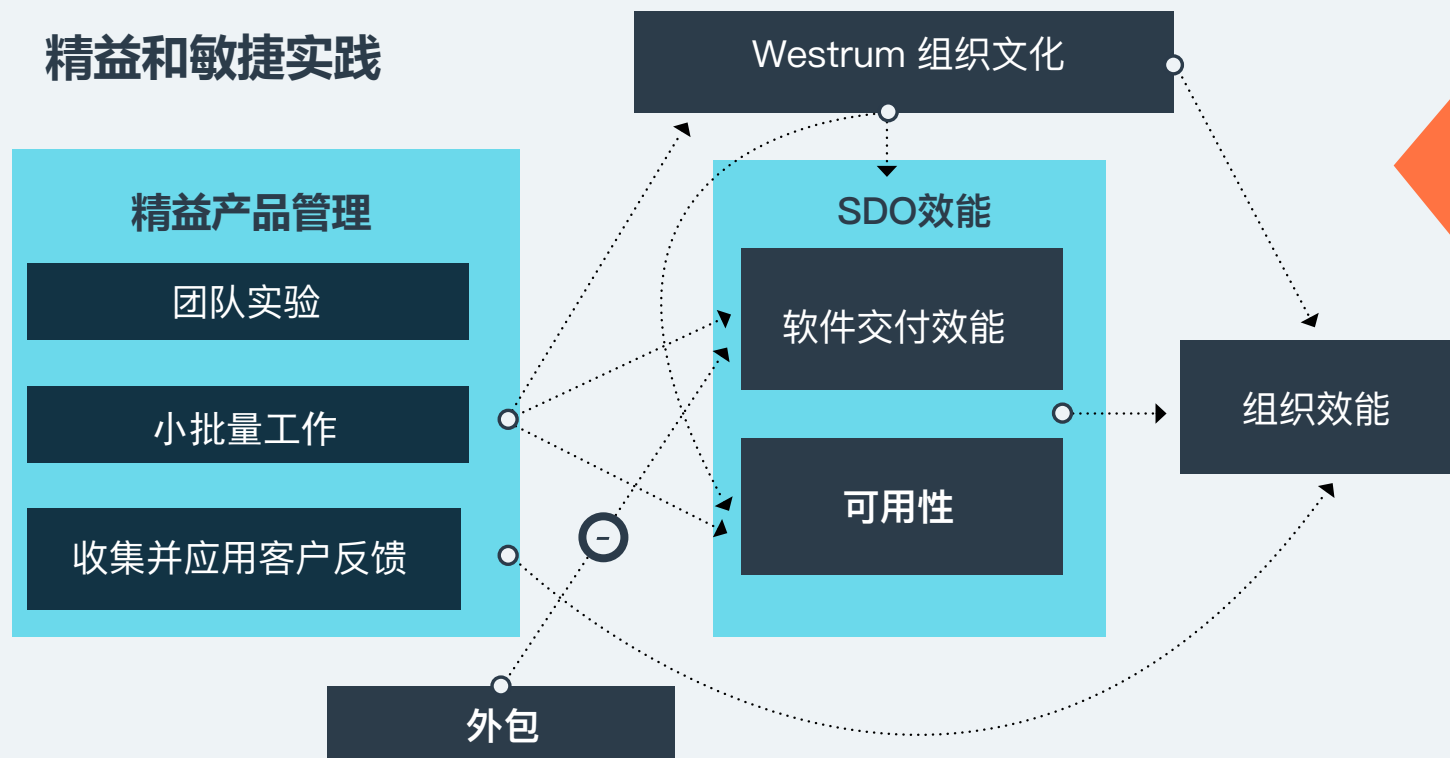
2

组织是否积极和定期地寻求客户反馈，并运用反馈来指导产品设计。

3

作为开发流程的一部分，开发团队是否被授权在无需额外审批的情况下新增和变更产品要求。

我们的研究验证了之前的研究结果：精益产品管理能力对软件交付效能、组织文化和绩效有积极影响。我们今年的研究还得出新的结论：外包对软件交付效能有负面影响，精益产品管理方式对可用性有正面影响。



方框是基本构造，内含方框的更大方框是二阶构造，箭头表示关联关系。以粗体标记的是今年新增加的构造，之前研究的其他构造相互之间的关系今年得到了再次确认。

\* Westrum是一个组织文化的度量方法，它特别适合DevOps。我们在第62页文化部分详细讨论了这一点。

## 技术实践

我们的研究也表明很多实践对成功技术转型至关重要。这些实践包括版本控制、部署自动化、持续集成、主干开发以及松耦合架构。

今年，我们还发现，采用监控和可观察性解决方案、持续测试、将数据库变更集成到软件交付流程中，以及安全检查左移等实践都有助于持续交付。

### 持续交付

那些在部署和交付中能够降低发布风险和成本的技术实践，我们统称其为持续交付技术实践。它们是实现高效软件交付的关键。请注意，如果您希望组织取得成功，那么为了持续交付而持续交付是不够的。

## 我们如何衡量持续交付

1

在整个软件交付生命周期中，团队能够按需将软件部署到生产环境或交到最终用户手中。

2

团队中的每个人都可以获得有关系统质量和可部署性的快速反馈，并且根据这些反馈采取行动是团队成员的最首要任务。

与往年一样，我们证实了下述能力对持续交付有积极影响的假设：自动化部署、持续集成、主干开发、松耦合架构，以及对所有生产制品进行版本控制。我们今年还发现并验证了一些对持续交付有积极影响的新能力：监控和可观察性、持续测试、将数据和数据库集成到部署流水线中，以及将安全验证集成到软件交付工作中。我们今年还确认了持续交付的影响，并发现它显然有助于减少部署工作给员工带来的痛苦和疲惫，并可提升SDO效能。

## 监控与可观察性

良好的监控是高效能团队的永恒主题。在过去几年里，我们发现，主动监控应用程序和基础架构并利用所得信息做出业务决策，也与软件交付效能密切相关。

### 监控

它是一种工具或技术解决方案，允许团队观察和了解软件系统的状态。监控以收集预定义的指标或日志为基础。

### 可观察性

它是一种工具或技术解决方案，让团队可以主动调试他们的系统，并探索他们未预先定义的特征和模式。

系统复杂性的不断增加激发了关于监测和可观测性的更多讨论，因此我们将这些主题纳入了今年的研究中。我们发现，全面的监控和可观测性解决方案对持续交付有积极影响。此外，具备良好监测和可观测能力的团队，其属于精英效能团队的可能性要高1.3倍。

为了理解团队如何在工作中利用监控和可观测性，我们在确定调查度量方式时假设监控和可观测性是两种不同的能力或实践维度。但是，当我们对数据进行统计验证时发现，调查受访者认为这些做法是一回事。因此，我们将监控和可观测性结合在了一起，对其统一进行分析。

## 度量方式的构造

为了细致地了解 and 度量这些能力，我们在研究过程中采用了如下严谨方法<sup>\*</sup>

1. 定义每种能力（即构造）
2. 根据精心编制的定义来拟订调查问卷的问题（即调查项），并交由领域专家审核
3. 收集数据
4. 通过统计学测试，对构造进行验证

我们尽可能使用多个调查项。因为有始终存在某个调查项被误解的可能性，此时就需要在后续分析中舍弃相应数据在统计验证过程中（步骤4），我们会对调查项目进行分析，以验证它们所度量的就是想要度量的目标并且没有度量那些我们不想度量的内容，同时，调查受访者还能够对它们有一致的理解。

这样的流程让我们能够采取谨慎和系统的方法进行调查研究，以便能够度量和研究新的领域，并验证其对结果的影响。

<sup>\*</sup>有关此过程的更多详细信息，请参阅《加速度：精益软件与DevOps的科学原理》一书的第二部分。

那么，监控和可观测性到底是一回事吗？行业权威强烈认为它们是不一样的两个维度。所以请允许我们澄清一下，当我们说把监控和可观测性放在一起时，我们的意思是今年的受访者认为这两组实践和能力是基本上是一回事。以下是几种可能的解释。

首先，可观测能力这个市场还相对比较新，还没能展现出让整个市场认同的差异，这也意味着在差异化和信息传递方面还有潜在机会。还有一种可能性是监测和可观测性的市场差异性只对部分细分用户群来说是显而易见或有意义的，而我们的调查针对的是各种各样岗位的DevOps从业者，而不仅仅是那些能注意到监测和可观测性之间差异的人。

## 持续测试

在过去几年中，我们发现测试自动化对持续交付有重大影响。今年，我们在前几年的研究基础上，发现持续测试对持续交付也有积极影响。

近年来，测试社区中的一些人一直对持续交付持怀疑态度，因此我们想调查一下测试所发挥作用的演变是否对前面所述的持续交付结果有所影响。



但什么是持续测试？它与自动化测试有什么不同？

在过去几年中，我们发现自动化测试对于持续交付是非常重要的。我们度量自动化测试的方面包括：主要由开发人员创建和维护快速、可靠的自动化测试套件；另外开发人员应该可以轻松重现自动化测试并在测试失败时使用自己的开发环境进行修复，而技术专业人员应该可以轻松使用测试数据来进行测试。

### 在此基础上，持续测试还包括以下重要实践：

- 持续验证和优化测试套件，以更好地发现缺陷并管控复杂度和成本
- 允许测试人员在整个软件开发和交付过程中与开发人员一起工作
- 在整个交付过程中开展探索性测试、可用性测试和验收测试等人工测试活动
- 让开发人员在代码库做任何变更时，先写单元测试再写生产代码，以实现测试驱动型开发
- 无论是在本地工作站还是CI服务器端，都可以在十分钟之内得自动化测试的反馈结果

持续测试可能是一笔重大的投资，但是由于团队在开发和交付生命周期中越来越重视测试，我们确实看到了这种方法的强大优势。

## 管理数据库变更

在执行部署时，数据库的变更经常是风险和延期问题的主要根源。我们想调查哪些与数据库相关的实践对持续交付有帮助，从而提高软件交付的效能和可用性。

我们的分析发现，把数据库工作集成到软件交付流程中确实有助于持续交付<sup>22</sup>，但是团队如何才能在持续交付中提高他们的数据库交付水平呢？有一些实践能够有所裨益。我们发现良好的沟通和把数据库包含在内的全面配置管理。在持续交付方面表现出色的团队把数据库的变更做成脚本存储在版本控制系统中，并在管理数据库变更时采用与管理生产应用变更相同的方式。

此外，当应用程序的变更需要数据库也进行变更时，这些团队会与负责生产数据库的人员讨论这些变更，以确保工程团队可以看到目标数据库的变更进度。当团队遵循这些实践时，数据库的变更就不会降低他们的速度，也不会在部署代码时导致问题。

---

<sup>22</sup> 有关将数据库工作集成到软件交付流水线中的详细讨论和提示，我们建议您阅读Laine Campbell和Charity Majors的Database Reliability Engineering。

## 安全性和安全效能

在这个安全威胁无处不在、持续不断，甚至有时是国家在背后支持的时代，信息安全至关重要。许多组织还受到PCIDSS、HIPAA或Sarbanes-Oxley等法规的约束，这些法规要求将信息安全管理（infosec）作为软件交付生命周期的一部分来实施。在我们的调查中，87%的受访者表示他们受到法规遵从要求的约束。

但是，与其他技术职能团队相比，信息安全团队往往人员配备不足。Signal Sciences 的研究主管James Wickett 指出了比例：大公司中往往每100个研发人员对应10个基础设施人员并对应1名信息安全人员。此外他指出，信息安全人员通常只在软件交付生命周期的最后阶段才参与进来，这时候再实施信息安全方面的改进，往往是痛苦而代价昂贵的。

在软件开发过程中内建安全性可以提高SDO效能和安全质量。**低效能组织需要数周时间来开展安全审查并完成所需变更。相比之下，精英效能组织内建安全性，可以在短短几天内完成安全审查和变更。**



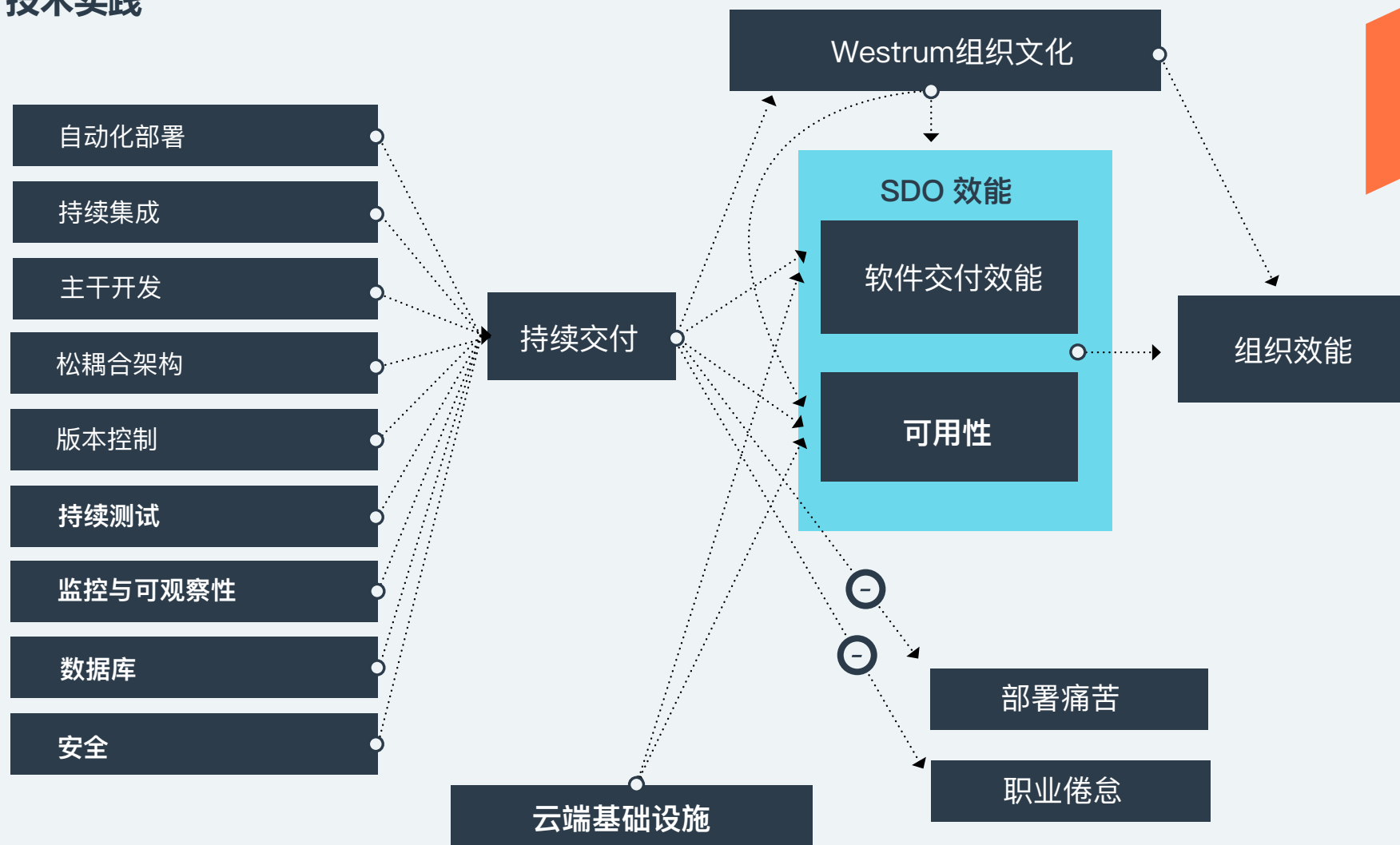
我们的研究表明，信息安全人员应该在整个开发过程中与团队合作，并参与应用的设计（包括对所有主要功能进行安全性评审）。换句话说，我们应该采用持续的方式来交付安全的系统。在表现良好的团队中，安全评审不会减慢开发过程。

安全风险前移可促进持续交付。为了建立安全性，团队应在整个软件开发过程中执行测试来发现安全问题，使团队可以轻松用上已通过安全评估的库、软件包和工具链，并为团队提供可以参考的安全流程模板。

我们还调查了由谁来负责安全性。在下面的词云中，我们可以看到答案中各角色的相对频率，这表明在被调查的受访者中，往往是由运维和基础设施团队来完成许多安全工作，其比例甚至高于信息安全专业人员。



## 技术实践



方框是基本构造，内含方框的更大方框是二阶构造，箭头表示关联关系。以粗体标记的是今年新增加的构造，之前研究的其他构造相互之间的关系今年得到了再次确认。

# 文化

文化一直都是DevOps、敏捷和精益运动中至关重要的组成部分。然而，文化是无形的，并且不可直接度量。2014年，我们应用并验证了社会学家Ron Westrum提出的组织文化模型，它可以驱动软件交付效能和组织效能。在过去的几年间，我们发现了一些可以影响文化的管理和技术方面的能力，可以通过改变组织中的工作方式来实现组织文化的改变。

今年，我们重新验证了之前几年的一些成果，并调查研究了如何通过领导力实践和学习文化来影响文化。

## Westrum组织文化

在过去的几年里，我们的研究已经证实业界多年流传的一种说法：文化是DevOps和技术转型的重要组成部分。我们发现技术和管理实践塑造了文化，而文化反过来又有助于提高效能产出。

为了衡量组织文化，我们参考了发现根据组织文化可以预测安全和效能产出的社会学家Ron Westrum开创组织类型学。Westrum组织文化的模型包括三种类型的组织：<sup>23</sup>

<sup>23</sup> Westrum, Ron. “组织文化类型学” Quality and Safety in Health Care 13, no. suppl 2 (2004): ii22–ii27.

| 病态型组织<br>(权力导向) | 官僚型组织<br>(规则导向) | 开拓型组织<br>(效能导向) |
|-----------------|-----------------|-----------------|
| 协作程度低           | 协作程度中等          | 协作程度高           |
| 消灭信使            | 不重视信使           | 训练信使            |
| 逃避责任            | 各自担责            | 共担责任            |
| 阻碍交流            | 容忍交流            | 鼓励交流            |
| 失败时相互推诿         | 失败时公平惩罚         | 失败时追根溯源         |
| 压制新想法           | 认为新想法会造成麻烦      | 接纳新想法           |

他对文化的定义涉及到了在谈论此话题时，我们听说过的关于DevOps团队的许多重要方面：协作、揭露问题（训练信使给我们带来坏消息，以便我们能够发现并修正问题）、打破孤岛（鼓励交流），事后回顾（失败时追根溯源），并通过坚持不懈的尝试来促成改进（接纳新想法）。

其他调查的结果也印证了这一点，团队的氛围对于团队效率来说，要比团队成员具备的特定技能更加重要。研究员对Google内部的180多个工程团队进行研究后发现，预测高效能团队时最重要的因素是心理安全，即在团队里承担风险时所感受到的安全感。其次是可靠性、工作的结构和明确性、意义以及影响力。<sup>24</sup> 当团队拥有良好的氛围时，他们的工作就会在技术和组织层面受益。我们多年的研究证实了这一点，并且我们提醒组织不要忽视他们的员工和文化在技术转型中的重要性。

---

<sup>24</sup> <https://rework.withgoogle.com/blog/five-keys-to-a-successful-google-team/>

但是我们如何去影响文化呢？我们已经看到了管理和技术实践（也就是我们的工作方式）可以到影响文化。我们还可以做些什么呢？

## 通过领导力和自主权影响文化

今年，我们研究了领导者的角色对于文化的影响。我们发现当领导者在工作中给予团队自主权时，信任感和发言权随之而来。信任反映了一个人有多相信他们的领导和经理们是真诚的，有好的动机和意图，并且公平对待他们。发言权是指某个人感觉自己及团队发表意见的能力有多强，尤其是在遇到冲突期间（例如，当团队成员的意见不统一时，当系统故障或者遭遇风险时，以及提出工作改进建议时）。信任和发言权，会反过来对组织文化产生积极影响。

## 员工净推荐值 (eNPS)

员工净推荐值eNPS是一个衡量员工参与程度，以及他们向同行推荐团队或组织的可能性。在很多行业里，该数值和公司的成长相关，<sup>25</sup> 其他研究也表明它和更优异的业务成果相关。<sup>26</sup>

我们的研究也发现，Westrum组织文化与（eNPS）<sup>27</sup>高度相关，而精英效能团队的成员将其团队作为理想工作场所推荐给其他人的比例要高1.8倍。

---

<sup>25</sup> <https://hbr.org/2003/12/the-one-number-you-need-to-grow/ar/1>

<sup>26</sup> Azzarello, Domenico, Frédéric Debruyne, and Ludovica Mottura. “热情的化学.” Bain. 2012年5月4日. <http://www.bain.com/publications/articles/the-chemistry-of-enthusiasm.aspx>

<sup>27</sup> [https://en.wikipedia.org/wiki/Net\\_Promoter](https://en.wikipedia.org/wiki/Net_Promoter)

那领导者如何才能最有效率地帮助他们的团队在工作中获得自主权呢？ 重要的因素是：

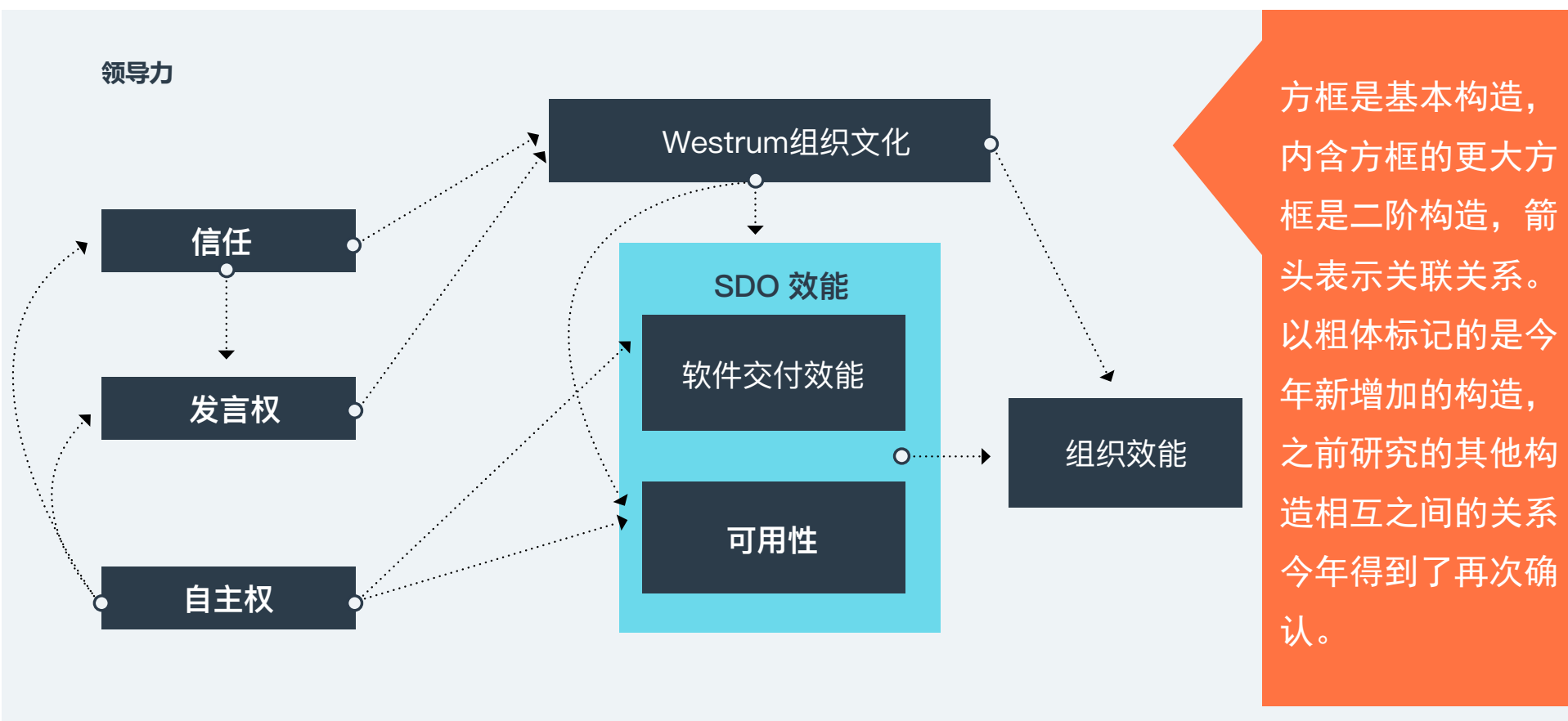
- 树立并传达目标，但是让团队自己决定该如何完成工作
- 通过简化规则消除障碍
- 如果规则阻碍了目标的实现，允许团队更改规则
- 允许团队按给客户带去的价值来安排工作优先级，即使这意味着要在一定程度上违背规则

我们可以看到这种向团队明确地传达成果和目标的做法是至关重要的，它甚至优于强劲的领导力。一旦团队理解了目标，一个优秀的领导者会相信：团队成员能凭借他们的专业知识实现目标。当然，我们的研究发现给予团队更多自主权有助于树立对领导者的信任——也就是说，团队相信他们的领导者是公正、诚实且值得信赖的。这种对领导力的信任会有助于营造更强有力的组织文化。



自主权带来的额外好处是，它可鼓励团队发表各自对工作、团队的意见，并提出改进工作的建议。这种透明的沟通也有助于改善组织文化。

下图展示了这种模型。



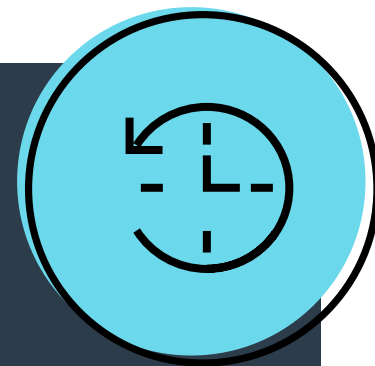
## 通过学习影响文化

影响组织文化的另一种方式就是学习。在DevOps和工程领域，这往往是通过回顾（也称为复盘）来完成的。在运维领域，经常在突发事件结束后进行复盘，以便了解如何改进系统以防止类似事件再次发生。这种做法有时被称为“事后回顾”。

不管怎样，目标都是一样的：学习如何改进。无论是在敏捷领域还是运维领域，很重要的一点是经常强调“不要指责”。例如：在Norm Kerth 2001年写的《项目回顾：团队复盘手册》<sup>28</sup>中，他提出了“回顾的首要原则”，建议参与者在开始回顾时先念：“不管我们发现了什么，我们都必须理解并且相信基于我们当时所具有的资源、技术、能力和条件，团队中的每个人都尽了最大的努力。”

<sup>28</sup> J.Paul Reed在《无可指责的回顾》一书中做了重要补充，他写了“回顾存在的指责现象”。

(<https://techbeacon.com/blameless-postmortems-dont-work-heres-what-does>) 建议读者要意识到人类倾向于指责，在组织回顾要避免这个问题。



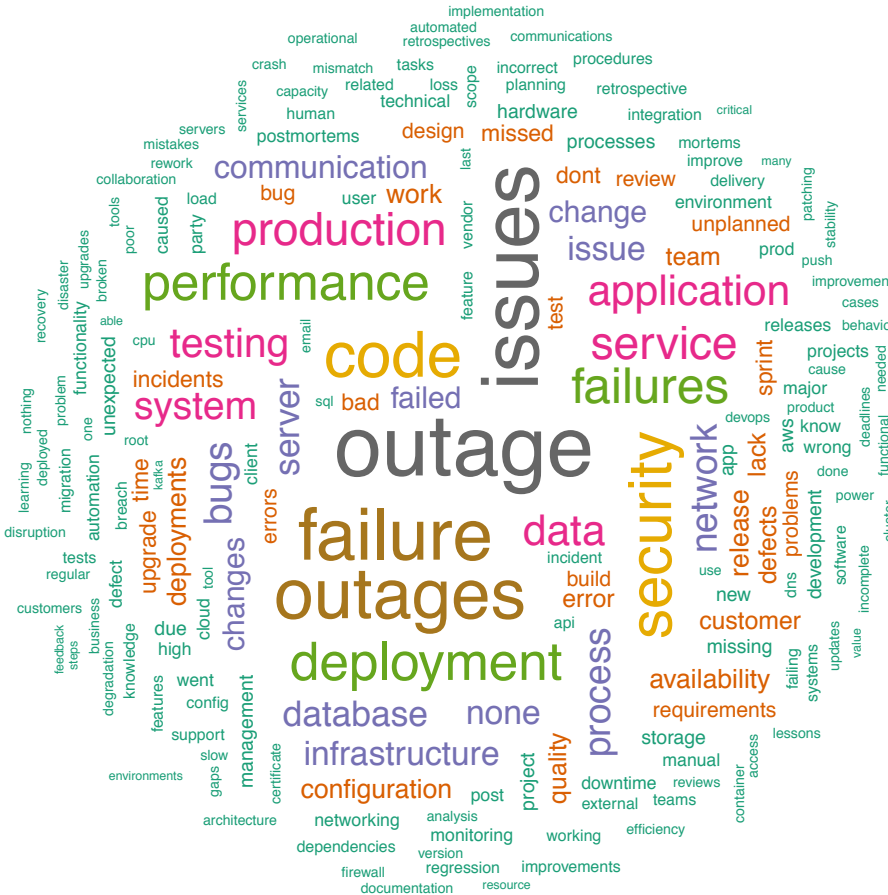
## 回顾

敏捷宣言的第十二条原则：“团队定期思考如何变得更加有效，然后相应地改变和调整其行为。”

我们发现复盘有利于营造学习氛围，也会影响到组织文化。重要的是团队从错误和失败中汲取教训，然后将其转化为改进工作方式的机会。特别是，利用回顾总结来实现对工具、过程或流程的改进，成效最为显著。

通过分析我们发现，精英效能组织坚持开展回顾并藉此改进工作方式的比例要高1.5倍。当我们询问受访者他们近期的回顾情况时，往往会发现一些常见的主题，例如系统中断、故障、性能和部署问题。

## 回顾主题



## 学习氛围

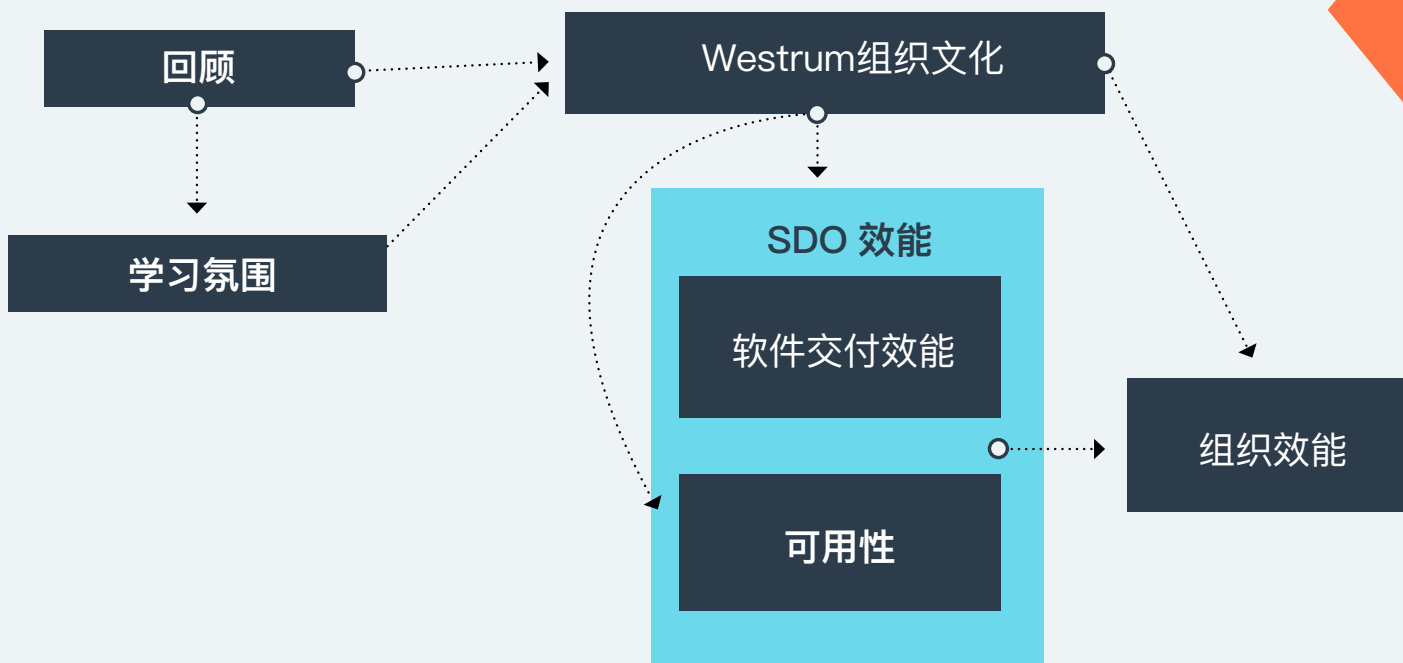
具有学习氛围的组织会把学习看做是一种投资。这种投资是成长所必须的。这样的组织不会把学习看成苦差，非到万不得已就不去学习。在金融等其他领域开展的一些研究表明这种学习氛围产生的效能收益是可以预期的。今年的调查确认了之前调查的结果，即学习氛围对组织文化具有积极的影响。

同样重要的是组织期望员工在什么条件下进行学习。在工作时间内有没有学习和成长的机会？相对地，是不是员工想学习但得不到支持，不得已只能利用晚上和周末的时间来学习？这样的环境最终会导致员工怠倦，并对少数群体和非传统生活方式的人产生不同程度的负面影响。例如，如果你需要照顾小孩或生病的父母（这些事情往往由女性来承担），抽出时间来学习就更加困难，导致完全无法跟那些能利用业余时间来学习和培训的人相比。

具有浓厚的学习氛围，这对团队和企业来讲都是一项战略优势。随着需求的不断改变，我们的工作环境变得越来越复杂，一个积极迎接改变并且不放过任何一个机会来学习新知识的组织文化，最终会取得领先地位。无论是技术转型、组织架构变化，还是客户需求和市场迅速转变，这样的团队都能在变革中茁壮成长。

考虑到这一点，组织如何营造这样的学习氛围呢？首先，他们尽可能提供各种学习机会和资源。例如为参加培训和研讨会提供正式的预算，此外还可以组织骇客日、员工聚会、午餐研讨会，这些都是很好的选择。

### 学习氛围



方框是基本构造，内含方框的更大方框是二阶构造，箭头表示关联关系。以粗体标记的是今年新增加的构造，之前研究的其他构造相互之间的关系今年得到了再次确认。



# 结语

在五年严格的科学分析中，我们一再看到软件交付对各种组织的重要性。我们采用独特的研究方法，它既能够提供数据来帮助您对自己的团队进行基准评估，也能够预测分析您可以借力于哪些关键能力来实现数字化转型。我们真诚地感谢所有参与调查的人。

我们希望当您尝试新的工作方法时，我们的研究有助于激励您和您的组织。我们期待听到您的故事。

## 方法

我们严谨的调研方法并不止步于原始数据，而且还关注软件交付和运维效能、组织效能、技术实践、文化规范和管理方式之间的预测关系。

在本节中，我们将介绍我们的分析方法，说明我们如何招募调查受访者以及我们如何设计我们的问题、模型和结论。

有关更多详细信息，您可以参阅我们出版的《加速度：精益软件与DevOps的科学原理》一书的第二章。如果对我们的研究方法有任何问题，欢迎利用以下邮箱联系我们[data@devops-research.com](mailto:data@devops-research.com)。

## 研究设计

本项研究采用代表性、理论性的设计方式。这种被称为推理式预测的设计方式是当今商业和技术研究中最常用的方式之一，适用于不可能实现纯粹的实验设计并且更倾向于现场实验的情况。例如，在商业领域，当数据收集发生在复杂的组织中，而不是在无菌的实验室环境中时，公司不会牺牲利润来适应由研究小组定义的对照组要求。

## 目标人群和采样方法

我们这次调查的目标人群是从事技术工作和转型工作或与之密切合作的从业者和领导者，尤其是熟悉DevOps的人员。因为我们没有这些人的完整名单——我们可以描述他们，但是我们并不确切地知道他们在哪里，如何找到他们，或者他们有多少人——我们使用雪球取样来募集受访者。这意味着我们通过电子邮件列表，线上推广和社交媒体推广调查，并要求人们与他们的人际网络分享调查，像雪球一样增加样本量。我们的样本可能仅限于熟悉DevOps的组织 and 团队，因此他们可能本身就负责一些DevOps工作。克服雪球采样限制的关键是拥有多样化的初始样本。我们通过利用我们自己的联系人列表以及我们的赞助商的联系人列表来实现这一点，从而实现受访者及其公司的特征与行业趋势大致相符。

## 创建潜在的构造

我们尽可能使用以前验证过的构造。当我们需要创建新的构造时，我们会根据理论、定义和专家意见来编写它们。然后，我们采取了额外步骤来澄清意图和措辞，以确保从最终调查中收集的数据很可能是可靠和有效的。<sup>29</sup> 我们使用Likert型<sup>30</sup>问题进行概念度量，从而可以执行更高级的分析。

<sup>29</sup> 我们使用了Churchill's 方法论：Churchill Jr, G. A. “开发更好的营销结构衡量标准的范例，” Journal of Marketing Research 16:1, (1979), 64–73.

<sup>30</sup> McLeod, S. A. (2008). Likert scale. 来自 [www.simplypsychology.org/likert-scale.html](http://www.simplypsychology.org/likert-scale.html)

## 统计分析方法

### 聚类分析

我们使用聚类分析来确定我们的软件交付效能分组。在这种方法中，同组会展现同样的基于吞吐量和稳定性的效能行为：如部署频率、前置时间、服务恢复时间和变更失败率等。我们使用了基于Ward方法<sup>31</sup>的解决方案，基于（a）融合系数的变化，（b）每个群集中的个体数量（包括具有少数个体的群集的解决方案被排除）和（c）单变量F统计量来选择。<sup>32</sup>我们使用了层次聚类分析方法因为它具有很强的解释力（让我们了解集群中的父子关系），并且因为我们没有任何行业或理论上的理由来拥有预定数量的集群。也就是说，我们希望数据确定我们应该拥有的集群数量。最后，我们的数据集不是太大（层次聚类不适合超大型数据集）。

<sup>31</sup> Ward, J. H. “分层分组以优化目标函数。” 美国统计学协会记者。 58 (1963): 236—244.

<sup>32</sup> Irich, D., and B. McKelvey. “一般组织分类:美国和日本电子工业的实证检验。” 组织科学. 1, no. 1 (1990): 99—118.

<sup>33</sup> Straub, D., M.-C. Boudreau, and D. Gefen. “IS实证研究的验证指南。” Communications of the AIS 13 (2004): 380—427.

<sup>34</sup> Nunnally, J. C. 心理测量学理论. New York: McGraw-Hill, 1978.

<sup>35</sup> Chin, Wynne W. “如何撰写和报告PLS分析。” In: V. Esposito Vinzi, W. W. Chin, J. Henseler, and H. Wang (eds.), Handbook of Partial Least Squares. Berlin: Springer (2010): 655 – 690.

<sup>36</sup> <http://www.socialresearchmethods.net/kb/convdisc.htm>

<sup>37</sup> <http://www.socialresearchmethods.net/kb/reliable.php>

<sup>38</sup> 这些方法论和考虑说到了Chin的支持 (1998), Chin, W. W. (1998). 关于结构方程建模的问题和看法. MIS Quarterly, 22(2), vii-xvi. Gefen et. al (2011) Gefen, D., Straub, D. W., & Rigdon, E. E. (2011). 行政和社会科学研究的SEM指南的更新和扩展. MIS Quarterly, 35(2), iii-xiv. and Hulland (1999). Hulland, J. (1999). Use of partial least squares (PLS) in strategic management research: 最近四项研究综述. Strategic Management Journal, 20(2), 195-204.

## 测量模型

在进行分析之前，使用探索性因子分析确认构建，使用varimax旋转进行主成分分析。<sup>33</sup>

使用平均方差提取（AVE）、相关性、cronbach's alpha<sup>34</sup>和复合可靠性<sup>35</sup>确认了收敛和发散有效性<sup>36</sup>和可靠性<sup>37</sup>的统计检验。构建通过了这些测试，因此表现出良好的心理测量特性。

## 结构方程模型

使用PLS分析测试结构化方程模型（SEM），这是基于相关性的SEM。我们利用PLS进行分析有以下几个原因：它不需要假设数据的正态性，它非常适合探索性和增量式研究，并且分析优化了因变量的预测（与模型拟合的测试相比）数据。<sup>38</sup>使用SmartPLS 3.2.6。

在控制行业时，未发现显著效果。SEM图中显示的所有路径均为 $p < .001$ ，除了以下内容， $p < 0.05$ ：数据库→持续交付监控和可观察性→持续交付，Westrum→SDO效能（在技术实践模型中）和Westrum→组织效能（在精益和敏捷实践模型中）。



## 致谢

作者感谢众多人员对于今年的报告所做的贡献和指导。所有的致谢按照字母顺序分类列出。

特别感谢**George Miranda**和**Xavier Velasquez**为今年的调查和报告提供了撰写和分析方面的支持。这个项目很大程度受益于你们今年的参与。

感谢**Sam Guckenheimer**和**Bryan Kirschner**在报告整体上的指导。

对于话题评审和度量条目的贡献，我们感谢**Silvia Botros**、**Stephanie Herr**、**Liz Fong-Jones**、**Bridget Kromhout**、**Cheryl Kwinn**、**Aruna Ravichandran**、**J. Paul Reed**、**Mary Robbins**、**Robert Reeves**、**Dustin Smith**、**Claire Taylor**和**Seth Vargo**。

作者感谢**Cheryl Coupé**在编辑今年的报告时所做的认真和细致的工作。

报告的排版和设计由**Siobhán Doyle**完成。

# 作者个人简介



**Nicole Forsgren**博士是DevOps研究和评估组织（DORA）的共同创立者、CEO和首席和科学家，也是《加速度：精益软件与DevOps的科学原理》一书的合著者。她最为人知的是在测量技术流程方面和作为迄今规模最大的DevOps研究的首席调查员所做的工作。她曾经是一位教授、系统管理员和性能工程师。**Nicole的作品**发表在一些经同行评审的期刊上。Nicole取得了亚利桑那大学的管理信息系统博士学位，她也是克莱姆森大学和佛罗里达国际大学的研究人员。

**Jez Humble**是《加速度：精益软件与DevOps的科学原理》、《DevOps实践手册》、《精益企业》和获Jolt生产力大奖的《持续交付》的合著者。他的职业生涯包括开发、基础设施、在3个大洲的不同大小的公司做产品开发、最近在为美国政府的18F项目工作。他目前研究如何在其初创公司建立一个高效能团队、参与DevOps研究和评估，以及在加州大学伯克利分校任教。



**Gene Kim**是一位曾获奖项的CTO、研究者和作家。他是Tripwire的创建者并担任了13年的CTO，他是《凤凰项目：一个关于IT、DevOps和有助于商业成功的小说》、《DevOps实践指南》，以及最新发布的《加速度：精益软件与DevOps的科学原理》的合著者。从2014年开始，他担任了DevOps企业峰会的组织者，研究大型、复杂组织的技术转型。



## 关于DevOps研究和评估组织

DevOps研究和评估组织（DORA）是由Nicole Forsgren博士、Jez Humble和Gene Kim建立的，对于软件开发中的高性能的实现及其影响因素进行研究。DORA经过四年多的研究，以及三万多个数据点，为一套基于证据的工具提供了基础，这些工具用于技术组织的评估和基准测试，并确定加速其技术转换过程的关键能力。

详情请看[devops-research.com](https://devops-research.com).

### 《加速度：2018 全球DevOps现状调查报告：新经济战略》中文版致谢

我们特别感谢中国DevOpsDays社区的组织者、行业专家、志愿者们，对本年度报告中文版翻译和校对工作所付出和努力。

总策划：孙振鹏 总编辑：张乐、刘征

翻译委员会：（按姓氏笔画排列，排名不分先后）

万 金 王一男 王立杰 卢梦纯 乔 梁 许 峰 李伟光 吴 非 吴 穹 何 勉 汪 珺  
张 晔 张笑梅 林伟丹 周一行 周纪海 姚 冬 徐 磊 徐 毅 龚明杰 董沙沙 瞿俊龙

校对委员会：（按姓氏笔画排列，排名不分先后）

王 磊 严惟安 杜静娴 李 旭 张 洁 张彬彬 鲁逸丁 熊小龙



## 我们的赞助商

我们要感谢我们的赞助商。除Google Cloud之外，《加速度：2018年DevOps的状况-新经济的策略》这份报告因这些全球知名的公司的支持才成为可能，这些公司支持我们出版高质量的、供应商中立的研究。他们的支持使得行业标准的研究可以帮助技术型的组织理解如何加速向高效能发展。

### Amazon Web Services

Amazon Web Services提供超过120个全球基于云的服务，包括计算、存储、数据库、分析、网络、移动、开发工具、管理工具、IoT、安全、机器学习和企业应用。

### CA Technologies

从计划到开发到管理到安全，CA创建的软件为在应用经济的公司助力转型。

### CloudBees

CloudBees助力持续经济，通过构建世界第一个端到端系统，自动化持续交付并提供治理、可视化和洞察，优化效率并控制新风险。

### Datical

Datical的使命是改变商业构建软件的方式。我们的解决方案交付自动化能力，企业需要移除作为交付应用创新的障碍的数据库部署。

### Deloitte

在德勤，我们从战略到运维，帮助客户拥抱精益、敏捷和DevOps能力。我们的专家团队建立和部署适合的解决方案，可以促进和支持敏捷。

### Electric Cloud

Electric Cloud的适应性发布编排和DevOps分析平台，可以让类似于E\*TRADE、GM、Intel和Lockheed Martin的组织有信心地发布新的应用，并且适应对业务需求的变更。

### GitLab

GitLab是被十万个组织使用的独立应用，使得产品、开发、质量、安全和运营团队，达到200%更快的DevOps生命周期。

### IT Revolution

IT Revolution通过出版、活动和研究聚集把技术领导和从业者。我们寻找提升工作状态、量化未达最佳绩效的成本，并且改善技术工作者的生活。

### Microsoft

我们的使命是让地球上的每个人和每个组织能够成就更多。  
包括Azure云、DevOps工具，以及Visual Studio和VS代码开发环境。

### Sumo Logic

Sumo Logic是领先的原生云、机器数据分析平台，可以在应用生命周期中交付实时的、连续的智能服务。

### PagerDuty

PagerDuty是组织领先的数字化运营管理平台。全球由超过一万个企业和中小组织使用PagerDuty，来提升数字化运营。

### Tricentis

业界第一的持续测试平台。  
Tricentis被认为彻底改造了软件测试工作，即把DevOps转型测试从一个障碍变成一个创新的催化剂。

### Pivotal

Pivotal软件公司集成了我们的原生云平台、开发工具和独特的方法论，帮助世界上最大的公司转型他们最重要的软件应用的构建和运行。

### XebiaLabs

XebiaLabs开发企业级的持续交付和DevOps软件，提供给公司可视化、自动化和控制的能力，使软件交付更快而且更少风险。

### Redgate

领先的Microsoft SQL Server工具供应商，Redgate设计高可用和可靠的软件，完美地解决了开发者和数据库管理员每天要面对的问题，帮助他们实施数据库DevOps。

### 中国DevOpsDays社区

中国DevOpsDays社区隶属于国际DevOpsDays核心组织，是中国大陆各城市承办DevOpsDays大会的官方组织。

