

## DevOps 中国调查研究\*

刘博涵, 张贺, 董黎明



(计算机软件新技术国家重点实验室(南京大学), 江苏 南京 210023)

通讯作者: 张贺, E-mail: hezhang@nju.edu.cn

**摘要:** DevOps 已提出近十年,其作为敏捷方法在完整的软件生命周期上的延伸,旨在从文化、自动化、标准化、架构以及工具支持等方面,打破开发与运维之间的壁垒,重塑软件过程,以在保证高质量的前提下,缩短从代码提交到产品上线之间的周期.在竞争日益激烈的市场环境下,用户对于产品服务的稳定性以及更新频率和效率的要求不断提高,DevOps 在学术界和工业界的关注程度因此也不断提高.Puppet Labs 在 2013 年开始了全球 DevOps 现状的问卷调查,迄今已发布了 5 份报告.国内 DevOps 的发展相对滞后,对于国内 DevOps 现状的研究在工业界和学术界均处于空白.2016 年和 2018 年分别进行了两次关于 DevOps 国内现状的问卷调查以填补这一空白,两次调查的受访人数分别为 74 和 66 人.基于两次调查结果,从 DevOps 涵盖的 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这 8 个方面,综合分析了 DevOps 在国内的发展现状与趋势,并与 Puppet Labs 报告的全球现状进行了对比.总体而言,国内 DevOps 虽呈现了稳步发展的态势,但与国际水平相比尚存在明显差距,目前能达到国际高水平 IT 性能的受访团队仅 6%.通过对比,总结了 17 条发现,经过综合分析,获得了 3 个主要结论:(1) 员工素质和人才紧缺是国内 DevOps 滞后、过程成熟度不高的症结;(2) DevOps 化越成熟,员工敬业度和满意度越高;(3) Scrum 敏捷开发和基于主干开发是最普遍采纳的实践.基于分析结果,在未来实践与研究上给出了多项建议.

**关键词:** 软件工程;软件过程;DevOps

**中图法分类号:** TP311

中文引用格式: 刘博涵,张贺,董黎明.DevOps 中国调查研究.软件学报,2019,30(10):3206-3226. <http://www.jos.org.cn/1000-9825/5796.htm>

英文引用格式: Liu BH, Zhang H, Dong LM. Survey on state of DevOps in China. Ruan Jian Xue Bao/Journal of Software, 2019,30(10):3206-3226 (in Chinese). <http://www.jos.org.cn/1000-9825/5796.htm>

### Survey on State of DevOps in China

LIU Bo-Han, ZHANG He, DONG Li-Ming

(State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing 210023, China)

**Abstract:** DevOps has been proposed for nearly a decade. As an extension of the agile approach from development to the entire software lifecycle, DevOps aims to break the barrier between development and operation and reshape the software process from the aspects of culture, automation, standardization, architecture, tool support, etc. The process is to achieve the shortest possible cycle from code submission to product release, while guaranteeing high quality. In an increasingly competitive market environment, users are increasingly demanding the stability of product services and the frequency and efficiency of updates. Therefore, DevOps has attracted a

\* 基金项目: 国家自然科学基金(61572251); 南京大学计算机软件新技术国家重点实验室开放课题(KFKT2017A13)

Foundation item: National Natural Science Foundation of China (61572251); State Key Laboratory for Novel Software Technology (Nanjing University) (KFKT2017A13)

本文由“面向 DevOps 的软件工程新技术”专题特约编辑荣国平、白晓颖、岳涛推荐.

收稿时间: 2018-09-01; 修改时间: 2018-10-31; 采用时间: 2018-12-14; jos 在线出版时间: 2019-05-22

CNKI 网络优先出版: 2019-05-22 15:26:11, <http://kns.cnki.net/kcms/detail/11.2560.TP.20190522.1525.005.html>

large number of researchers and practitioners. Puppet Labs began a global survey of the state of DevOps in 2013 and has published five reports so far. The development of DevOps in China is relatively lagging behind. The research on the state of DevOps in China is in a blank in both industry and academia. Two questionnaire surveys are conducted on the state of DevOps in China in 2016 and 2018 to fill this gap. The numbers of responses received in the two surveys were 74 and 66 respectively. Based on the results of the two surveys, this study performed a comprehensive analysis from eight aspects, including IT performance, organizational culture and related practices, development and operation and maintenance practices, tool support, leadership, work ratio, employee engagement, and satisfaction. The development status and trends of DevOps in China with the global status reported by Puppet Labs are compared. Overall, DevOps has shown a steady development trend in China, but there is still a significant gap compared with the international level. At present, only 6% of the respondents could achieve the international high-level IT performance. 17 findings from the comparison are summarized. Then, three main conclusions are obtained through the synthesis of these findings: (1) The quality of employees and the shortage of talents are the crux of lagging behind and the process maturities of most of organizations are not high; (2) The more mature DevOps leads to the higher degree of the dedication and satisfaction of employees; (3) Scrum agile development and trunk-based development are the most commonly applied practices. Based on the analysis results, several recommendations are provided for practice and research.

**Key words:** software engineering; software process; DevOps

回顾过去十年软件工程的发展,2009年一定是具有里程碑意义的一年,在这一年的 DevOpsDays(<https://www.devopsdays.org/>)上,Debois 首次提出了一个合成词——DevOps,软件工程的新时代就此拉开序幕。

向前追溯到 2001 年,Beck 等人成立敏捷联盟,集结其余 20 位软件工程专家,发起了敏捷宣言(<http://agilemanifesto.org/iso/zhchs/manifesto.html>)。其主要目的是使软件组织适应业务变更日趋频繁的市场环境,充分而有效地响应需求<sup>[1]</sup>。以响应变化、快速交付价值为宗旨的敏捷开发衍生出了各种不同的敏捷开发方法以适应不同的组织、团队和项目的特征,其中,最主要的有 Scrum<sup>[2]</sup>、极限编程(extreme programming)<sup>[3]</sup>、精益开发(lean development)<sup>[4]</sup>、Kanban<sup>[5]</sup>等。

然而,随着(移动)互联网应用的爆发式增长,同质化产品之间的竞争日益激烈,用户需求的多样性及用户对产品稳定性要求的日益增长,敏捷开发也不再能够满足所有业务场景。其中一个重要的原因在于,现有的敏捷方法只关注迭代过程,而忽略了整个软件生命周期中的大多数其他活动,尤其是运维。在传统的软件企业中,往往将开发和运维分设两个部门(有时单独设立的质量保证部门也会被考虑进来),敏捷开发仅仅覆盖了开发部门的主要工作。这导致了开发部门与运维部门仅关注各自的工作与职责,缺乏沟通,且由于工作性质的差异,部门文化也相去甚远,开发部门受到敏捷开发的熏陶在内部形成了敏捷文化,而运维部门的同僚或在堆积如山的各种琐碎的配置、部署任务中应接不暇,或疲惫地奔走于分散各地的客户现场。这种隔阂使得开发和运维的步调难以保持一致,这意味着,即使开发部门能够在敏捷的驱动下快速响应变更,但缺少了运维的支持,依然无法将产品迅速地交付给市场。

一条顺理成章的思路是,让运维团队也敏捷起来。于是,DevOps,开发(development)和运维(operations)的融合诞生了。目前,对于 DevOps 还没有一个一致且明确的定义,但为什么我们需要 DevOps 是明确的,在市场的激励下,软件组织必须在保证质量的前提下尽可能地减少将产品新特性交付到市场的时间。Bass<sup>[6]</sup>以一个架构师的视角将 DevOps 定义为一组旨在减少从提交变更到变更作为产品交付的时间,同时确保高质量的实践。Smeds<sup>[7]</sup>整理了现有关于 DevOps 的不同定义,有些人认为这是一场组织革命,有些人认为这是一次文化运动。关于 DevOps 定义的多样性恰恰体现出了 DevOps 的这种试图涵盖完整的软件生命周期,并影响整个组织及其内部的每一个人的包罗万象的特征,这本身就是对行业未来的一种颠覆;另一方面也体现出了 DevOps 还处于萌芽阶段,无论在学术界还是工业界都没有形成一套完整而又成熟的知识体系。

近两年,DevOps 在国内的热度不断攀升,与 DevOps 相关的会议遍地开花,小到十几人、大到上万人的企业纷纷摩拳擦掌。但即便如此,国内对于 DevOps 的反应还是明显滞后于国外,Puppet Labs 在 2013 年<sup>[8]</sup>即开始了全球 DevOps 现状的问卷调查,彼时,DevOps 在国内还只为很小一部分人所知晓,从 Puppet Labs 的调查结果我们也可以发现,来自亚洲的参与者占比极少。截至 2016 年,国内还依然没有一份关于 DevOps 中国现状的调查面世;截至目前,国内也还没有一篇关于 DevOps 中国发展现状的调查发表。本研究期望填补这一空白,通过近两年对企业中实践 DevOps 的相关从业人员的调查,描绘了 DevOps 在中国的现状及其发展趋势,并通过与国外的现状

与发展进行对比,对 DevOps 在中国的所处阶段进行了定位.我们分别在 2016 年和 2018 年对国内 DevOps 从业人员进行了两次问卷调查.问卷从 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、障碍这 8 个方面进行了调研.结果显示,随着 DevOps 实验经验的生长,团队在 IT 性能表现等方面可以得到显著提高.目前,国内 DevOps 团队在敏捷方法和 DevOps 相关工具的采用实践较好,不同团队在具体方法和工具的选择上呈现出了多样性.然而,在其他方面,尤其是组织投入上,还明显不足,尤其是与国际水平存在明显差距,从各方面来衡量,虽然我们看到了从 2016 年~2018 年呈现出来的一定进步,但目前还处于大约 4 年前的国际水平.

## 1 DevOps

DevOps 是开发和运维这两个领域的合并.它为开发和运维构建桥梁,避免开发和运维之间的脱节,同时让团队以更高效的方式进行合作,实现工作流程上的无缝对接<sup>[9]</sup>.DevOps 通常是指软件行业新兴的专业化运动,是一组文化、流程与工具整合后的统称<sup>[10,11]</sup>.DevOps 通过自动化流程来使得软件构建、测试、发布更加快捷、频繁和可靠.

DevOps 生命周期始于需求定义,覆盖开发、构建、测试、部署和交付反馈等阶段<sup>[10]</sup>.DevOps 是软件开发生命周期从瀑布式到敏捷再到精益的演化(如图 1 所示).瀑布模型的特征是,在进入下一阶段之前,每个阶段目标必须 100%地完成.相对于瀑布开发模式,敏捷开发过程的一个基本原则就是以更快的频率交付最小化可用的软件.在敏捷的目标里,最明显的是在每个 Sprint 的迭代周期末尾,都具备可以交付的能力.DevOps 是敏捷开发的延续,它与敏捷相辅相成,因为它拓展和完善了持续集成和发布流程,因此可以确保代码是生产上可用的,并且确实能给客户带来价值<sup>[10]</sup>.

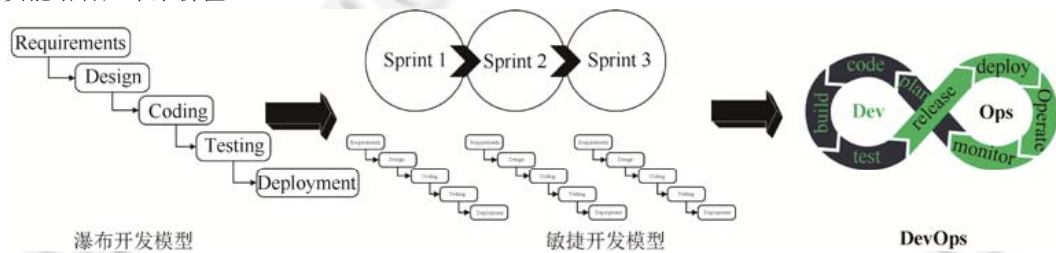


Fig.1 The evolution of software development process model

图 1 软件开发模型演化

DevOps 提倡开发和 IT 运维之间的高度协同,在高频率部署的同时,要保证高质量的完成,高质量即代表要提高生产环境的可靠性、稳定性、弹性和安全性等能力<sup>[10]</sup>.

DevOps 的一个基本点是强调整个系统的稳定性,而非将性能局限于特定的工作领域里,这个领域可以大到一个部门(例如开发和 IT 运维)或者小到一个人贡献者(例如开发者,系统管理员等),由 IT 推动业务价值流.DevOps 通过高部署率快速地把一个想法变成价值交付到客户手中,即“概念落地”.并且在不影响客户的基础上,快速地完成并部署很多小的变更.DevOps 的目标不仅仅是增加变更的频率,而且也支持在不中断和破坏当前服务的基础上,确保功能部署成功,同时也可以快速检测和修复缺陷.

IT 价值流中存在着巨大浪费,这些浪费源于开发和运维之间交接的不顺畅,突如其来的计划外工作,以及各种原因导致的交付期限推迟.在开发和 IT 运维之间,DevOps 不仅聚焦在创建快速和稳定的计划工作流,而且 DevOps 也有一个更全面的方法来系统地消除计划外工作,定义开发弹性准则,并负责管理和减少技术负债.

DevOps 使得开发人员能够通过持续集成和标准化的发布惯例(持续测试的文化)来维持高频率部署,在此基础上,确保质量和信息安全.一种常见的组织级标准是,一旦代码被提交,自动测试便开始运行,且一旦发现问题,则必须马上解决.

企业在应用了 DevOps 之后可以得到以下几个方面的业务优势:产品快速推向市场(比如,缩短开发周期时

间和更高的部署频率),提高质量(比如,提高可用性,提高变更成功率,减少故障)并提高组织的有效性(比如,将时间花在价值增加活动中,减少浪费,同时交付更多的价值至客户手中)。

## 2 问卷调查

2016年末,我们在DevOps中国社区(<http://www.devopschina.org/>)的支持下,开展了首次国内范围的DevOps中国现状的问卷调查。基于对调查结果的整理,2017年,我们发布了中国第1份DevOps年度调查报告——《DevOps中国·2016年度调查报告》(<http://softeng.nju.edu.cn/devops/report2016>)。在前一次的调查和报告过程中,我们积累了丰富的经验,于是紧随其后,在2018年中旬发布了第2份报告——《DevOps中国·2018年度调查报告》(<http://softeng.nju.edu.cn/devops/report2018>)。我们期望通过一线实践者对于DevOps的理解和实践情况,了解DevOps在中国的现状与发展趋势。未来,我们计划每年年初(第1季度)开展基于问卷调查的数据收集活动,并于第2季度发布调查报告。本文将基于对已经完成的两次调查结果的整理与分析加以阐述与讨论。

### 2.1 问卷设计

本研究采用问卷调查的方式进行数据收集,在2016年的问卷(<http://softeng.nju.edu.cn/devops/devops2016cn>)中,为了尽可能地涵盖DevOps的方方面面,我们从基本信息、概念、经验、性能表现、文化、自动化、标准化、质量保证、监控、过程、服务、工具、投入、领导与授权、工作量、员工满意度和障碍共17个方面设计了38个问题。在DevOps所涵盖方面以及文化、自动化、质量保证等具体实践的问题设置上,我们参考了关于DevOps的文献综述的技术报告中所整理出的DevOps推荐实践<sup>[12]</sup>。

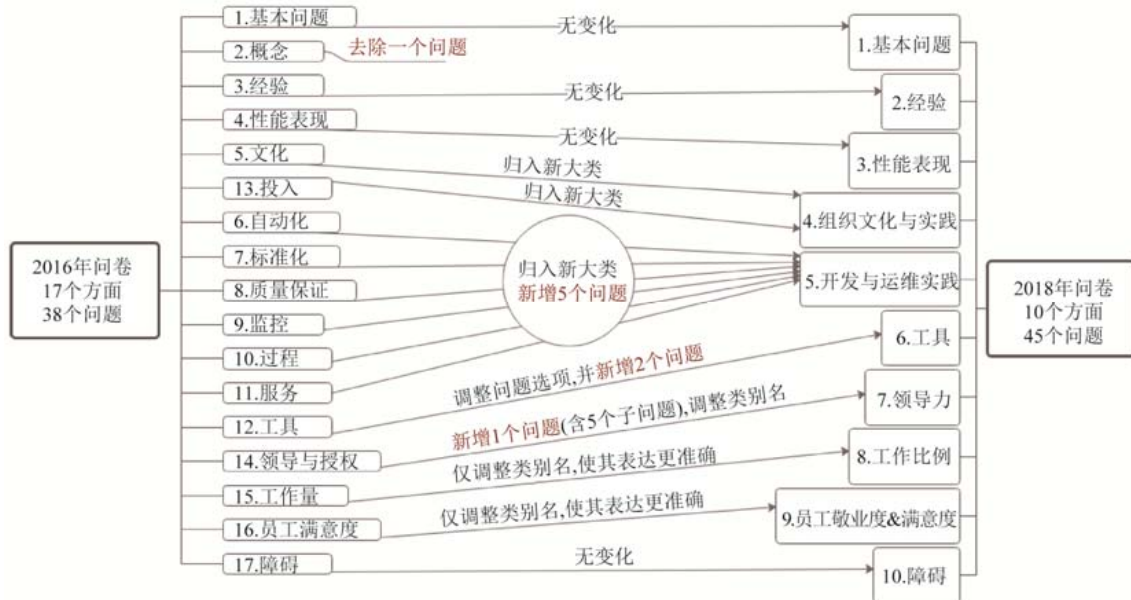


Fig.2 The evolution of the structure and questions of questionnaires

图2 问卷结构及问题演变

此外,考虑到期望与国际现状进行对比,所以我们还大量地参考了Puppet Labs从2013年~2017年已进行的5次全球DevOps现状调查<sup>[8,13-16]</sup>。Puppet Labs于2017年发布的最新报告有超过3000名分别来自全球六大洲(不含南极洲)的IT从业人员参与调查,其中,10%来自于亚洲。除在工业界已经产生了极大的影响力外,其报告的结果在学术界也被广泛地认可和引用<sup>[17,18]</sup>。

• 在2018年的问卷(<http://softeng.nju.edu.cn/devops/devops2018cn>)中,我们以结构更加清晰的方式,将自动化、标准化等实践在分类上合并为开发与运维实践,将文化与投入两类合并为组织文化与实践,去除了概念类

问题.此外,还针对部分问题进行了调整,并在持续交付、架构、工具等方面增加了新的问题,最终共 45 个问题.两次问卷的详细演变过程如图 2 所示,具体来说:在 2016 年我们提出了一个开放性问题,“请说明您如何理解和定义 DevOps?”,从结果来看,受访者对 DevOps 的解释各不相同,值得注意的是,最多被提及的两个概念是“一体化”和“自动化”,然而,由于解释的差异性过大,即便在学术界,目前也还没有对 DevOps 有一个明确且一致的定义,所以结果难以进行分析,我们在 2018 年的问卷中,考虑到新增问题较多,而问卷耗时对问卷的反馈率和反馈质量都有一定影响,最终去掉了这个难以分析的问题以缩短受访者的答题时间.

- 由于在 2016 年忽略了开发实践,所以在 2018 年增加了一个问题“您的团队在开发过程中,满足以下哪些实践?”,问题选项中的实践均来源于 Puppet Labs 报告中推荐的最佳实践.

- 持续交付是 DevOps 的一个核心价值和目标,所以对于自动化部分,我们在 2018 年专门针对持续交付,增加了一个问题,即“在持续交付中,您的团队有过以下哪些实践?”.问题选项中的实践同样来源于 Puppet Labs 报告.

- 在 2016 年,我们没有加入架构的相关问题.但通过参与工业界的相关会议,我们发现,架构反而是工业界对于 DevOps 最关注的主题之一,所以,2018 年我们增加了两个聚焦架构的问题,“以下哪些架构属性与您正在开发的应用/服务的架构相吻合?”,鉴于微服务是当前工业界在 DevOps 上讨论最多的架构,另一个问题是“微服务架构在您企业中的实施状态是?”.

- 2016 年仅针对团队采用的 DevOps 方法进行了调查,2018 年新增了精益生产管理相关实践的调查.

- 在 2016 年,我们基于 Ebert 等人推荐的 DevOps 自动化工具列表<sup>[19]</sup>,设置了工具实践的问题选项,但基于我们的最新研究结果<sup>[20]</sup>,该列表并不全面,且与当前工业界工具使用的现状不完全吻合,所以基于我们的研究结果,对受访者使用了哪些工具的选项进行了更新.

- Puppet Labs 最新的报告<sup>[16]</sup>指出了领导力对于 DevOps 实践有着举足轻重的影响,所以我们在 2018 年也新增了对于领导力的调查.

## 2.2 受访对象

问卷以在线的形式进行发布,通过 DevOps 社区及 InfoQ China(<https://www.infoq.com>)等软件开发领域的技术社区进行扩散式的传播,此外,还在南京(全球)软件大会(NJSD)(<http://www.njsd-china.org/>)等会议上进行宣传以吸引更多的开发者参与.为了尽可能地收集到更多数据,在 2018 年的问卷分发过程中,我们还第一时间将问卷发送给了所有 2016 年的受访者.考虑到受访群体均属于高收入且对于新技术有着浓厚兴趣的群体,在激励参与问卷的方式上,我们没有采用礼品等与金钱相关的激励方式,而是以受访者均能第一时间获取调查报告为激励,采用该激励方式也能够更好地屏蔽掉对于 DevOps 没有兴趣和经验的开发者,收集到的数据能够保证一个较高的质量.最终,2016 年我们共计收到了 74 家组织或个人开发者的有效反馈,2018 年为了开始实现每年一发布,明显缩短了问卷收集时间,所以反馈数量略有减少,共计收到了 66 家组织或个人开发者的有效反馈.目前,我们已建立数据库,收集和存储曾经参与过或下载过我们所发布报告的相关从业者的邮箱,相比其他 IT 从业者,他们对于 DevOps 以及我们的调查更感兴趣,这意味着他们有较高的概率会参与或继续参与到我们未来的调查中.在未来的每次调查中,我们会通过邮件向他们发出邀请,并不断更新和维护邮箱列表,这将为我们收集到更多有效的反馈提供支持.此外,随着与 InfoQ 合作的不断深入,其也将协助我们加大对于年度调查的宣传与推广.

两次调查的受访者在地域、行业、岗位、所处部门、所处组织规模和 DevOps 经验等方面覆盖面较广,根据调查问卷的发放形式,即通过技术社区及相关技术会议进行分发,参与者质量较高(这一点从反馈中没有无效问卷亦可见一斑),具有一定的代表性.以下将根据 2016 年和 2018 年的受访者的基本信息对其分布进行简要描述.

**地域.**受访者主要来源于上海、北京、深圳、南京、杭州等软件及互联网产业较发达地区,其中,来自上海的受访者在两次调查中均占比最高(两年分别为 23%和 18%).这从一个侧面反映了对于新技术、新文化的接受和实践程度与一个地区的行业发展水平有着密切的关系.在两次调查中,我们均未发现有非一、二线城市的开发者参与到其中,这很大程度上是由大规模或创新型的软件企业在国内的地域分布所决定的.

**行业.**受访者来自于和 DevOps 有关的各行各业,涵盖了科技、互联网、金融、教育等领域,但以科技(45%)和互联网(22%)企业为主.所涉行业之广说明了 DevOps 目前已经为各行各业的软件从业人员所了解,结合近两年各地举办的开发者大会上关于 DevOps 的火热讨论来看,DevOps 在国内的宣传已经初见成效,其对于软件组织及相关从业人员极具吸引力.但考虑到两次调查的参与人数相对有限,又说明了真正在引领国内 DevOps 发展,已经积累了一定 DevOps 实践经验的组织及个人还相对较少,下文将要详细介绍受访者的实践经验.

**岗位.**受访者处于不同工作岗位,包含了在软件生命周期中负责不同阶段工作的工程师以及项目和组织的管理者,如图 3 所示,由于两次调查的选项不同,所以不具有可比性,但可以看到,两年的受访者中 DevOps 工程师的占比均较高.这体现了 DevOps 已经引起了组织中各级人员的关注,对于 DevOps 这样一个旨在打破开发和运维壁垒,重铸企业文化的理念而言,是个好消息.

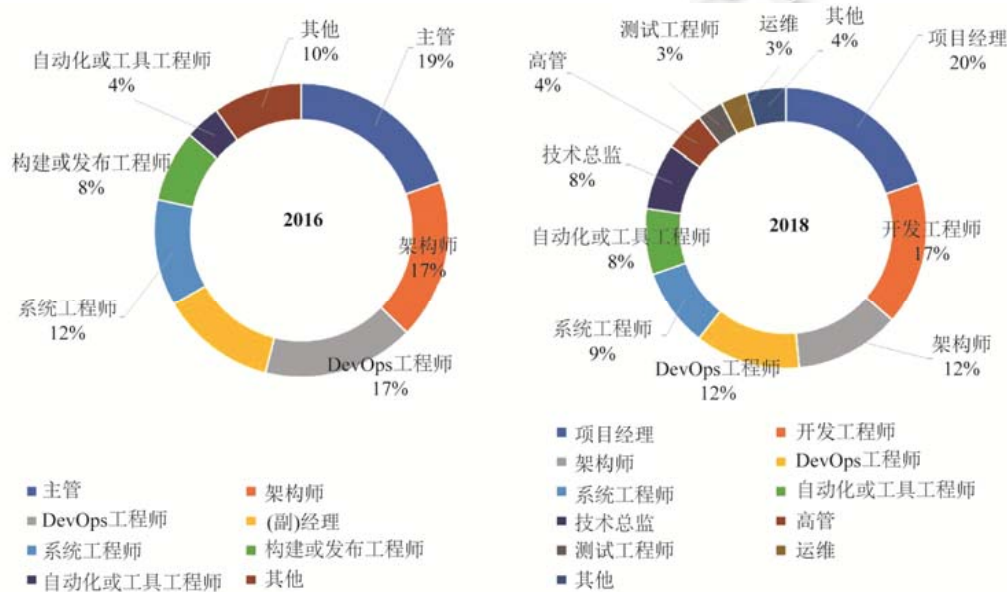


Fig.3 The distribution of respondents' jobs

图 3 受访者岗位分布

**部门.**受访者分属于不同的部门,如图 4 所示,展示了 2016 年和 2018 年两次调查受访者的所属部门分布.其中最值得注意的是 DevOps 部门的设立,且占比相当可观(两年均超过了 10%),这体现了其企业对于 DevOps 的重视,为转型 DevOps 设立了专门的 DevOps 部门.结合受访者所处岗位进行分析,我们发现,2016 年有 14%的受访者为 DevOps 工程师,2018 为 12%,这表明,几乎所有的 DevOps 工程师均来自于 DevOps 部门,企业往往不会在原有部门中增设 DevOps 工程师这一职位,说明不同企业基于 DevOps 对组织结构和人员投入上的差距两极分化较为明显.

**组织规模.**由于部分企业并非仅有软件开发业务,例如电商企业往往还有大量的商业策划、商品的售前、售后等人员.所以,基于人数来考量组织规模是不合理的,于是我们从与企业软件业务能力密切相关的服务器数量上进行考量.对于受访者所在组织的规模,2016 年和 2018 年的差异不明显,从 20 台服务器以内到超过 1 000 台服务器的组织数量分布平均,说明从小型到大型再到超大规模的企业均对 DevOps 有所关注和实践.

**经验.**鉴于 DevOps 还处于萌芽阶段,且中国大部分企业相对起步较晚,经验将是衡量 DevOps 实践水平的一个重要依据,所以我们从 3 个不同的角度了解受访者的经验,包括个人经验、其所在团队的成员平均经验以及其组织经验(即已推行 DevOps 多少年).如图 5 所示,受访者个人经验与其所在团队和组织经验基本持平,且超过半数受访者的经验少于 1 年,同时也不乏一些经验超过 3 年的个人或团队.

性别.两年的结果相近,超过 9 成受访者为男性,且超过 5 成受访者其所在团队的女性成员比例低于 10%,超过 1 成的受访者所在团队中没有女性.整个行业仍以男性为主.

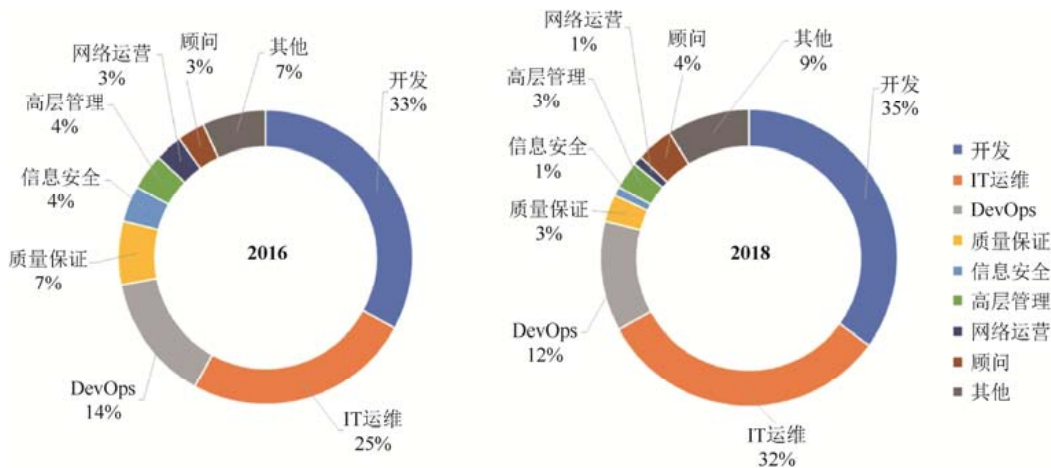


Fig.4 The distribution of respondents' departments

图 4 受访者所属部门分布

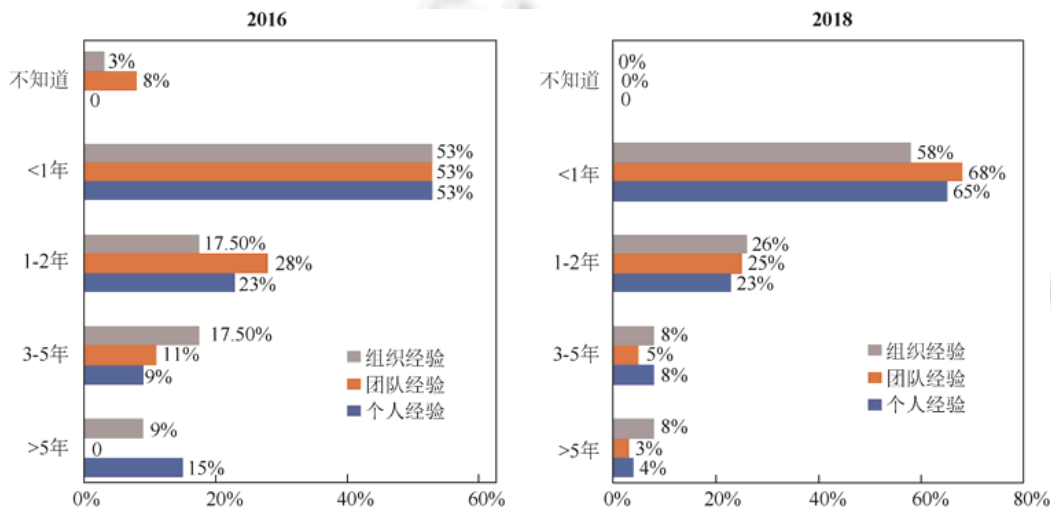


Fig.5 The distribution of respondents' experiences

图 5 受访者经验分布

### 2.3 分析方法

在此次调查中,数量统计是最基本和最常用的方法.但是为了更深入地分析组织间的差异,我们结合组织的 IT 性能来分析不同的实践因素是否对其产生影响,DevOps 的目标是在保证甚至提高质量的前提下,尽可能地缩短从变更提交到变更作为产品一部分交付的时间,所以我们使用两个主要纬度来衡量 IT 性能,即代码吞吐量和系统稳定性.为了更好地衡量 DevOps 对一个组织的生产活动的影响,我们将以上两个方面进行了量化,最终体现在部署频率、交付周期、平均修复时长、变更失败比例这 4 个参数上.DevOps 让软件过程更快,体现在“部署频率”和“交付周期”上.DevOps 让软件过程更稳,体现在“平均修复时长”和“变更失败比例”上,我们又根据收集到的数据分布,基于 IT 性能区分了高性能和低性能的团队进行对比.为了分析 DevOps 如何让开发人员幸福感提升,我们主要分析了员工计划外工作时间、是否恐惧代码部署员工以及净推荐值等几个参数的关系.为了更

进一步地了解高性能团队的驱动因素,主要分析了高性能团队在 DevOps 方面的技术实践,具体分为:(1) 实践活动,包括自动化、标准化、质量保证、敏捷方法、监控手段;(2) 架构,团队正在开发应用/服务架构;(3)工具,DevOps 各个阶段所对应的工具.为了了解团队负责人如何帮助团队获得成功,在 2018 年度调查中,我们使用了 Rafferty 和 Griffin<sup>[21]</sup>于 2004 年提出的变革型领导力模型来衡量团队负责人的工作表现,分别从愿景、智力激发、个人认可、鼓舞人心和支持型领导这 5 个纬度进行分析.另外,我们结合 2016 年度与 2018 年度的调查报告,综合统计出了阻碍 DevOps 推行的各方面因素.

### 3 国内现状与进展

基于上一节的结果,从整体来看,两次调查的受访对象分布差异较小,这对于我们基于两年的数据来对比分析国内 DevOps 的发展趋势提供了基础.基于问卷中问题的设置,受访者的基本信息和经验两个方面已在上一节中给出详细描述,本节将从 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和障碍这 8 个方面进行讨论.

#### 3.1 IT性能表现

从商业层面上来说,我们说一个组织是高效的,意味着该组织不仅可以快速地向市场提供具有关键业务的应用,并且具有能够规避那些可能中断其商业价值的服务的能力.

对于 IT 组织,在保持对市场需求的高敏感度的同时,它们还要提供稳定、安全并且可预测的 IT 服务.即:

- 组织是否能够快速地把一个想法变成价值交付到客户手中?
- 是否具备持续的提供有价值的产品的能力?
- 组织的活动是否是有效的?

换言之,组织需要实现既“快”且“稳”.“快”可以通过部署频率和交付周期来体现,而“稳”可以通过平均修复时长和变更失败比例来体现.

高频率部署也意味着快速和持续的部署,这有利于组织对市场需求及时反应,使组织能够通过自动化的构建、测试和部署循环来快速交付高质量的软件.2016 年的调查中有 32.4% 的团队可以实现一天之内进行多次部署,而 2018 年基本持平,有 33.8% 的团队可以实现.结合团队经验数据我们发现,DevOps 经验丰富的团队,部署频率更高.

交付周期是指团队从一次代码提交到成功运行于产品中所需要的时长,主要用于衡量组织应对变更的能力.2016 年的调查中实现交付周期少于一天的团队占比 23.5%,而 2018 年这一比例增长了近 3 倍,有 67.2% 的团队实现了一天之内完成交付.此外,结合团队经验数据我们发现,DevOps 经验丰富的团队,交付周期更短.

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导致服务受损、服务中断,需要热修复,回滚、打补丁等.2016 年有 66.2% 的受访团队可以将变更失败比例控制在 15% 以内,而 2018 年 80% 的团队可以实现这一目标,其中,36.4% 的团队能够将变更失败比例控制在 5% 以内.此外,随着团队 DevOps 经验的增加,变更失败比例 >30% 的可能性会越来越小;变更失败比例在 16%~45% 之间的团队 DevOps 经验均小于 2 年;DevOps 经验在 3~5 年之间的团队均控制在 10% 之间.

平均修复时长刻画了一个团队应对意外故障的能力,它指的是当线上的产品发生故障时,平均需要多长时间能够修复故障并上线.很多组织都无法忍受哪怕是几分钟的故障修复时间,因为这往往意味着数以百万计利润的流失与用户满意度的下降.在这一指标上,2016 年(32.4%)和 2018 年(30.8%)的调查结果基本持平.此外,平均修复时长同样随着 DevOps 经验的增长而缩短.

**发现 1.** DevOps 经验丰富的团队,IT 性能表现更好;从 2016 年~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.

基于这 4 项指标,我们将团队区分为准高性能和准低性能团队,划分标准见表 1,图 6 展示了受访团队中团队性能分布比例,可以看到,2018 年准高性能明显增多,这与发现 1 的结果一致,主要是由于 2016 年大量团队



在交付周期和失败变更比例上难以达到准高性能标准.后文中将基于该划分标准结合团队性能对 DevOps 实践的效果进行分析.

**Table 1** The standard of IT performance for different teams

表 1 团队性能划分		
指标	准高性能	准低性能
部署频率	1 周之内	超过 1 周
交付周期	1 天之内	超过 1 天
平均修复时长	1 天之内	超过 1 天
团队变更失败比例	0%~15%	大于 15%

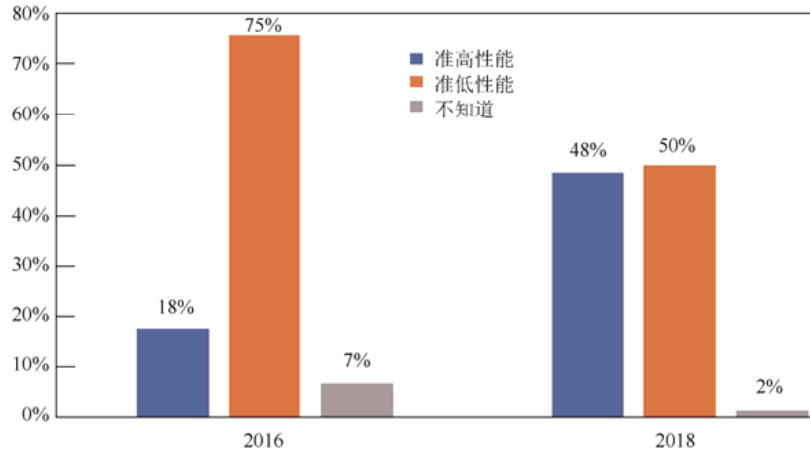


Fig.6 The proportion of quasi-high performance and quasi-low performance teams

图 6 准高和准低性能团队比例

### 3.2 组织文化及相关实践

组织文化是影响组织经济表现的最直接的因素之一.人是组织中最关键的资产,对员工的投资将提高员工的企业认同感和工作积极性.对于文化,借鉴了 Puppet Labs 2016 年的报告,我们主要从组织的氛围、特质、技术共享和投资 4 个角度进行分析.氛围包含了组织对于 DevOps 的支持和促进,以及员工能够从组织氛围中感受到的认同感,比如组织是否鼓励员工在软件的开发方式中使用 DevOps 理论和实践,是否信任并给予员工责任等;技术共享是指组织是否促进开发人员与运维人员之间的沟通来达到技术共享等等.特质指的是组织的结构或规章是否符合 DevOps 文化,例如组织是否淡化部门与团队之间的界限;同一个项目的各个团队是否分散在各地等等.投资指的是组织是否计划或正在投资支持 DevOps 工具的购买或开发;是否投资技术人员的培训或发展等等.

我们对于技术分享和氛围占比最高的两项实践进行分析,相对于准低性能团队的组织,准高性能的组织的 DevOps 文化氛围及相关实践做得更好,它们的员工有更高的企业认同感,且其企业在技术共享上更加出色.具体体现在 4 个方面,如图 7 所示,在文化氛围上体现为员工更能感受到组织在乎自己;组织信任员工并给予员工责任,在技术分享上体现为组织创建开发与运维共同使用的知识管理系统,组织促进开发与运维人员之间的沟通.在特质上,准低和准高团队没有明显差异,两年的结果也基本接近,组织特质主要体现为组织有同时具备开发和运维能力的员工(2016 年占比 68%,2018 年占比 82%),员工的工作不仅仅只是常规职业描述的内容(2016 年占比 34%,2018 年占比 30%).在投资上,如图 8 所示,2016 年准低和准高团队在技术人员和 DevOps 支持工具上的投资没有明显差距,两次调查的准低团队也没有明显差异;而 2018 年的准高团队投资上明显好于 2016 年.然而,无论准高还是准低团队,组织在 DevOps 上还有明显不足,平均低于三成的团队已投资人员培训或支持工具.

发现 2. 准高性能团队在员工的企业认同感和技术共享实践上做得更好,但在企业特质方面,对 DevOps 的投资上,没有发现与准低性能团队的明显差异,虽然 2018 年的准高性能团队相比 2016 年在投资上有明显提高,但仍处于较低水平.

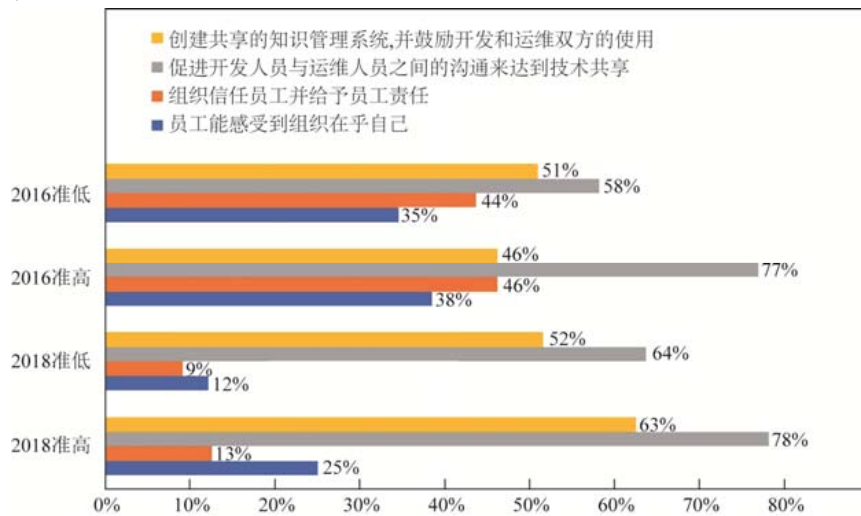


Fig.7 Cultural atmosphere and technology sharing

图 7 文化氛围与技术分享

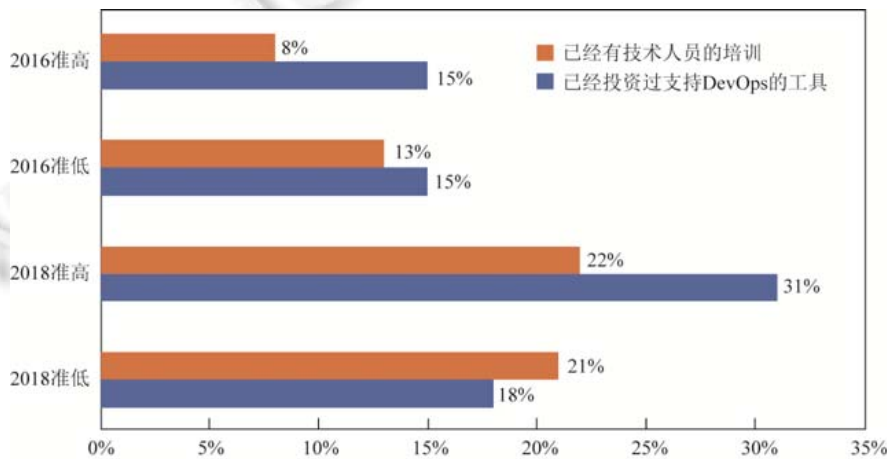


Fig.8 Investment in technicians and supporting tools

图 8 技术人员和支持工具投资

### 3.3 开发与运维实践

开发与运维实践主要体现在 4 个方面,即自动化、标准化、质量保证和开发方法,其中,开发方法主要考虑的是团队是否使用敏捷方法以及使用了哪种敏捷方法.

自动化的一大目的是打造完全自动化的 Delivery Pipeline.自动化过程可以使得开发人员更关心软件的逻辑而不用与复杂的配置打交道.自动化也是提高可测试性、一致性、稳定性、部署频率和实现持续交付的核心方法<sup>[22]</sup>.在软件的构建、测试、部署、发布和运维阶段的监控和恢复这些过程中,均能实现自动化过程,所以我们调查了生命周期不同阶段对应的自动化过程的实践情况,如图 9 所示.受访的团队均实现了部分过程的自动化,其中以自动构建和自动部署最为普遍.另外,近半数的团队实现了自动测试以及自动监控,比例最低的为自

动恢复.可见,受访团队关注的重点主要在于构建、部署、测试以及监控.两次调查结果相似,其中,自动更改审批流程为2018年新增项,所以对2016年不适用.差异主要体现在自动监控和基础设施自动配置管理上,2018年的调查结果中这两项实践的占比反而相比2016年有所降低,我们认为这并不意味着2018年反而相比2016年在自动化实践上有退步,例如大量企业选择放弃自动化这种解放生产力的方式是不合常理的,所以我们相信,这种差异是由两次调查的样本之间存在的差异造成的.当然,这一结果也揭示出,自动监控和基础设施自动配置管理这两项实践的普及率可能并不像2016年调查结果中的那么高.

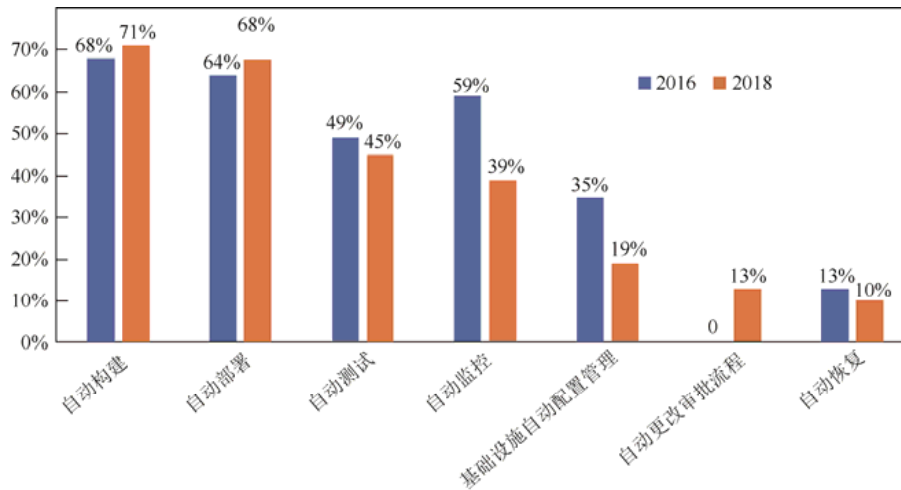


Fig.9 The ratio of process automation practices

图9 自动化过程实践比例

图10展示了2018年受访团队所进行的具体的自动化实践的百分比,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%),2016年的结果与之接近,都显示出受访团队对于网络(实现了网络基础设施自动化)、安全性(在交付过程中有安全团队)方面有所欠缺.这在某种程度上说明,国内团队目前主要关注开发、运维与质量保证三者之间的协作,暂时未将安全考虑在内.

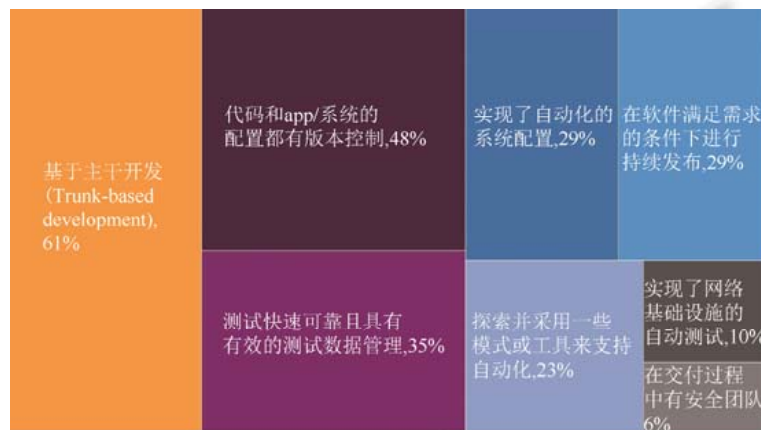


Fig.10 The ratio of specific process automation practices in 2018

图10 2018年具体自动化实践比例

**发现3.** 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.

标准化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<sup>[23]</sup>.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.仅有约 6%的团队没有任何的标准化实践.采用最多的标准化实践是“对 DevOps 的预测都是基于实证角度,除非极具专业性,否则避免使用数学模型”,占比 45%.而“应用 CMM 或 CMMI 以及 ITIL 标准到产品上”的占比最少,仅 29%.有 32%的团队表示会“从企业过程中汲取灵感”.

**发现 4.** 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.

质量保证是质量管理的一部分,致力于使得所提供的产品质量得到客户的信任.质量保证是指为使人们确信产品或服务能够满足质量要求而在质量管理体系中实施并根据需要进行证实的全部有计划和有系统的活动<sup>[24]</sup>.我们主要从 3 个方面考量质量保证实践的程度,如图 11 所示.其中,最为引人注目的就是“使用实时的用户监控并在早期发现问题”这一比例从 2016 年到 2018 年有了极大增长.同时,2016 年有超过三分之一的团队没有任何质量保证实践,而在 2018 年的调查数据中,仅有约 3%的团队没有任何质量保证实践.

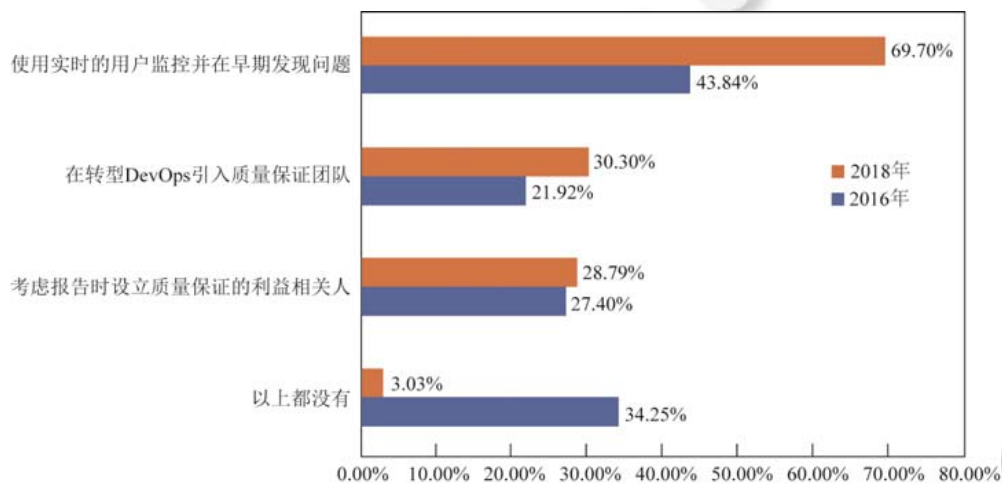


Fig.11 The ratio of different quality assurance practices

图 11 质量保证实践比例

**发现 5.** 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.

敏捷方法对于 DevOps 并不是必须的,虽然 DevOps 是敏捷的一种延伸,且敏捷与 DevOps 在自动构建、自动测试、持续集成与持续交付等方面都具有有一致性.许多业界专家认为将敏捷与 DevOps 结合起来将获得更好的企业效益<sup>[25]</sup>.从 2016 年到 2018 年,没有使用任何敏捷方法的团队比例从 27.4%下降到了 19.7%.在具体方法的使用上,两年差别不大,超过 40%的团队使用 Scrum,12%的团队使用 Kanban,使用其他方法的团队占比均少于 5%.

**发现 6.** 国内团队对于敏捷方法的采用呈现增长趋势,虽然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.

### 3.4 工具支持

在 2016 年,基于 Ebert 等人推荐的 DevOps 自动化工具列表<sup>[19]</sup>,我们调查了 15 种支持 DevOps 实践的工具在受访团队中的使用情况.我们使用相同的工具列表作为选项,分别问了“使用过以下哪些工具?”和“推荐以下哪些工具?”两个问题.图 12 列出了这 15 种工具推荐和使用(推荐+未推荐)情况.将近六成的团队会使用 JenKins (58%),且几乎均推荐该工具.Maven 也被超过一半的团队使用(55%),且其中超过三分之二的受访者推荐该工具.Ant 也有较高的使用率(39%),然而,仅不到一半的使用过的团队推荐该工具.相反地,ReamCityChef、Chef、Bamboo 和 New Relic 虽然使用率不高,但使用者几乎均推荐这些工具.Graylog 未发现有团队使用.

在 2018 年的调查问卷中,基于我们对于 DevOps 自动化支持工具的研究成果<sup>[20]</sup>,选取了 Stack Overflow (<https://stackoverflow.com/>)和现有文献中讨论最多的工具,调整了“使用过以下哪些工具?”问题中的选项.考虑到无法穷尽所有的 DevOps 工具(仅从 Stack Overflow 中,我们就识别到了 178 个不同的工具<sup>[20]</sup>),且当用户需要以文字输入时会更加慎重,我们将“推荐以下哪些工具?”改为了填空式的开放性问题.

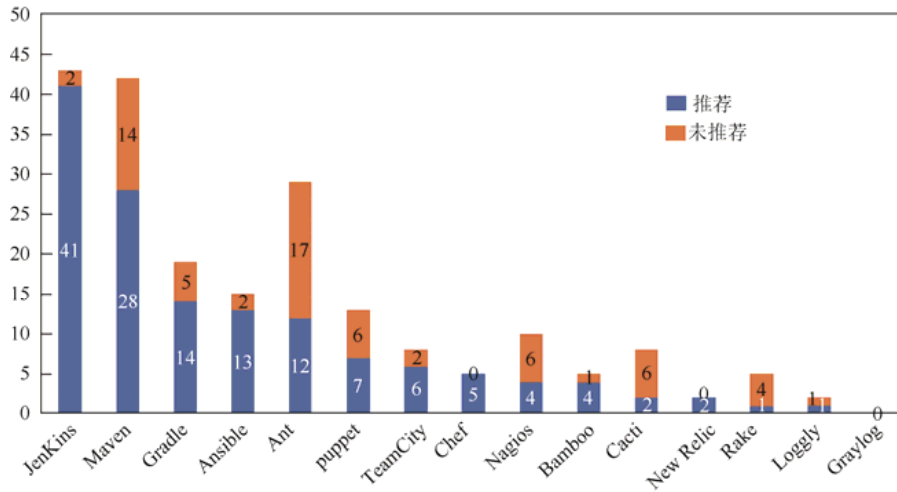


Fig.12 The tools used and recommended by the teams in 2016

图 12 2016 年团队 DevOps 的工具使用与推荐

如图 13 所示,左图统计了准高性能团队工具的使用比例,超过半数的团队使用了 Docker、JenKins 和 Git.比较 2016 年和 2018 年的结果我们可以发现,由于两次问卷所列工具列表存在差异,所以结果存在明显差异,但两次调查都存在一些被普遍使用的工具,这充分说明了 DevOps 工具的多样性以及国内团队工具使用的普遍性.JenKins 和 Ansible 在两次调查中使用率都非常高,同时还有着极高的推荐率,在 2018 年的调查中加入了 Docker,立刻超越 JenKins 成为了使用率最高的工具.图 13 中右图展示了受访者推荐的工具的频次,与 2016 年结果一样,JenKins 排在榜首.

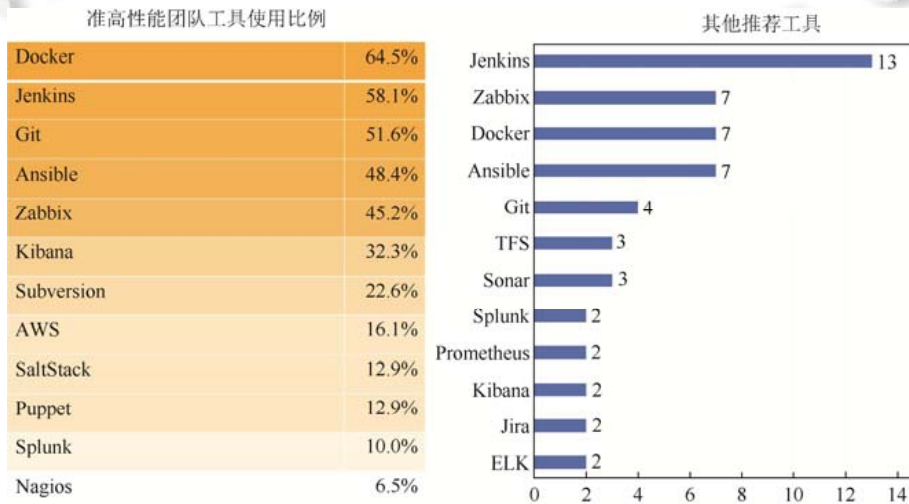


Fig.13 The tools used and recommended by the teams in 2018

图 13 2018 年团队 DevOps 的工具使用与推荐

**发现 7.** 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、Ansible 和 Docker 等.

### 3.5 领导力

团队负责人是组织提升性能、获得经济增长的至关重要的一环.负责人的工作表现直接影响到组织的文化氛围、团队的 IT 表现以及团队对 DevOps 的投资比例等.在 2018 年的调查中,我们使用了 Rafferty 和 Griffin<sup>[22]</sup> 于 2004 年提出的变革型领导力模型来衡量团队负责人的工作表现.该模型由 5 个方面构成.

- 愿景:(1) 理解组织方向;(2) 理解团队方向;(3) 理解团队 5 年内方向.
- 支持型领导:(1) 在行动之前考虑他人的感受;(2) 思考他人的个体需求;(3) 关心个人兴趣.
- 鼓舞人心:(1) 激发团队成员的自豪感;(2) 宣传团队内部的正能量;(3) 激发热情和积极性;(4) 鼓励人们看到这种变化带来的机会.
- 智力激发:(1) 挑战团队现状;(2) 挑战团队不断提出新问题;(3) 挑战团队对工作的基本设想.
- 个人认可:(1) 赞扬高出平均水平的工作;(2) 认可工作质量的提高;(3) 亲自赞美个人的出色表现.

调查结果显示,87.1%的准高性能团队负责人可以有效地帮助团队,而准低性能团队中这一比例为 71.4%.具体来说,准高性能团队在愿景这一项比准低性能团队多出了 13%;鼓舞人心多出了 69%;智力激发多出了 37%;个人认可多出了 39%;在支持型领导上差异不明显.这一结果说明,准高性能团队负责人的领导力更强,而更强的领导力被认为可以有效地帮助团队,帮助直接体现在了团队 IT 性能更高这一点上.反过来说,这种领导力上的显著差距也说明准高性能团队对负责人的领导力更加重视,甚至是有更高的要求.

**发现 8.** 准高性能团队更重视负责人的领导力,且团队负责人的领导力的提升对于团队 IT 性能表现的提升有着切实的好处.

### 3.6 工作比例

对于开发人员而言,计划外的工作,比如代码调试、修复 Bug 等都被认为是没有建设性的工作任务,这样的工作对于开发人员而言没有任何成就感.计划外的工作时间在开发人员中的全部工作时间中所占比例越低,就意味着开发人员在自己的主要工作上的时间和精力投入越多,越少地被外界干扰.

在 2018 年调查的数据中,准高性能团队中,83.33%的受访者表示其用于计划外工作的时间小于 30%,而这一数字在非高性能团队中则为 64.71%.此外,在准高性能团队中,计划外工作时间占比超过 50%的仅占 6.67%,而对准低性能团队而言,这一比例就达到了 17.64%.

**发现 9.** DevOps 化越成熟,即 IT 性能表现较高的团队,其成员的计划外工作时间相对占比也越少.

### 3.7 员工敬业度及满意度

员工敬业度及满意度是衡量员工愿意留在公司并努力为公司服务的程度.至今,全球领先的人力资源咨询公司 Aon Hewitt 进行的全球员工敬业度及满意度调查已覆盖超过 7 000 家公司,其 2017 年的调查结果显示,员工敬业度与公司绩效之间存在密切关系(<http://www.aon.com/engagement17/>).在调查中,我们采用了 Employee Net Promoter Scores(eNPS)来衡量员工的敬业度,其主要体现在以下 3 个问题上.

- 您愿意向您的朋友或同事推荐您团队的产品或服务吗?
- 您愿意向您的朋友或过去的同事推荐您的组织吗?
- 您愿意向您的朋友或同事推荐您的团队吗?

eNPS 采用 10 分制的打分机制,其中,0~6 分为贬损者,7~8 分为被动者,9~10 分为推荐者.2016 年与 2018 年的结果相近,其中,2018 年的结果显示,准高性能团队的成员,相比准低性能团队的成员,在这 3 个问题中推荐者的占比分别高出 1.34、1.35 和 1.21 倍.

**发现 10.** DevOps 化越成熟,即 IT 性能表现较高的团队,其员工敬业度及满意度更高.

### 3.8 障碍

尽管 DevOps 能够带来更频繁的部署频率,构建更有创造性的开发团队,创造更多的商业价值,但企业在实际的转型和自上而下的推行中,会面临各种各样的障碍,我们在问卷中列出了 9 个可能的障碍供受访者多选,在问题选项的设置上,我们参考了 Bass 等人权威著作中的“Barriers”章节<sup>[6]</sup>.综合 2016 年和 2018 年的调查结果,有 5 个最普遍的障碍(被超过四分之一的受访者认为是障碍).

- 组织行业的限制(46%);
- 不同部门的目标不一致(46%);
- 员工还不理解 DevOps 的概念(38%);
- 组织缺少具备 DevOps 经验的专家(33%);
- 缺乏配置使用相关工具的专业知识和人才(25%).

发现 11. 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.

## 4 国内外现状对比

本节将对我们发布的两份国内调查结果与 Puppet Labs 从 2013 年~2017 年(截至 2018 年 7 月,Puppet Labs 还没有发布本年度的调查报告)发布的 5 份 DevOps 全球现状调查报告结果,针对可以量化且具有对比意义的部分进行对比,并对呈现较明显差距的部分进行讨论分析,以期对国内当前发展状况进行定位.

### 4.1 组织结构

在第 2.2 节有关部门这一部分,我们提到目前国内已有部分企业专门设立了 DevOps 部门,但通过与国外的受访者所属部门分布对比可以发现,在《2017 State of DevOps Report》中,DevOps 部门占比高达 27%,而在国内 2018 年的调查中,仅有 12%的受访者是来自 DevOps 部门的工程师,且相比 2016 年的 14%还略有下降.这说明了国内企业对于 DevOps 的重视程度,甚至是转型 DevOps 企业比例还存在不小的差距.Puppet Labs 从 2014 年开始统计 DevOps 部门占比,其经历了从 2014 年的 16%,到 2015 年的 19%,2016 年的 22%,再到 2017 年的 27%的稳定增长.结合 DevOps 部门和 DevOps 工程师岗位两项结果作进一步的对比,可以发现,Puppet 的 2014 年的受访者中 DevOps 工程师的比例已经高达 31%,说明除了改变组织结构,增设 DevOps 部门的企业外,还有相当一部分企业选择变动更小的方式,即增加新的岗位.

发现 12. 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.

### 4.2 IT性能表现

目前国际上对于高性能团队的定义为部署频率为一天多次,交付周期和平均修复时长在 1 小时以内,团队变更失败比例在 0~15%<sup>[15]</sup>.如果按照此标准衡量国内受访团队,我们发现,2016 年的受访团队无一能达到该标准,2018 年仅有 6%的团队可以达到这一标准,这虽然在一定程度上说明了部分国内团队在 IT 性能上有了明显的进步,但总体来看,国内与国际水平还存在明显的差距.针对这 4 个度量对 2018 年数据进行更细化的分析:35%的团队可以在部署频率上达标,甚至有 14%的团队可以将部署频率进一步提高到 1 小时 1 次甚至多次;仅 15%的团队可以在交付周期上达标,如果将这一标准放宽到中等性能团队标准(1 月以内),有 94%的团队可以达标.事实上,68%的团队可以实现 1 天以内完成交付;30%的团队可以在平均修复时长上达标;在变更失败比例上达标的团队高达 82%.由此可以发现,交付周期的控制是当前制约国内团队符合高性能团队标准的最主要因素,而这一指标恰恰反映了国内团队在开发和运维的衔接上还存在明显不足,这正是 DevOps 不成熟的充分表现.

发现 13. 国内团队在变更失败比例的控制上表现出色,但在交付周期上与国际高水平存在明显差距,综合 4 项 IT 性能指标,国内仅有 6%的受访团队能够达到国际上的高 IT 性能标准.

### 4.3 开发与运维实践

比较遗憾的一点是,在组织文化及相关实践、开发与运维实践和工具支持这 3 个对于 DevOps 极其重要的方面,由于 Puppet Labs 缺少对于原始可量化的数据的报告,难以进行对比,但我们还是从中获得了一个较大的发现.

**发现 14.** 国内外对于基于主干开发这一实践备受推崇,这似乎已经成为了一种必备的实践.

### 4.4 工作比例

在工作比例上,国际上同样也呈现了高性能团队比低性能团队的计划外工作比例少的现象.此外,国内准高性能团队的计划外工作比例,与国际上高性能团队的比例接近.

**发现 15.** 在工作比例上,国内的分布情况与国际水平接近,呈现出良性的结果.

### 4.5 部署痛苦程度

关于是否对部署感到痛苦这一问题,我们提供了从“一点也不痛苦”到“害怕部署”4 个不同痛苦程度的选项,结果出乎我们的预料.从 Puppet Labs 在 2017 年的报告结果来看,高性能的团队得益于自动化实践的充分开展,面对部署,比低性能团队感到更轻松.而我们在 2018 年的调查中却得到了一个完全相反的结果,23.3%的准高性能团队受访者表示其和团队都害怕代码部署,而准低性能团队表示害怕的仅为 8.8%.

对于这一现象的一个可能的解释是,准高性能团队内部对于部署有着更高的要求,这一方面促成了他们实现更快更稳的部署,另一方面对他们可能是更大的挑战和考验.由第 4.2 节我们可以发现,对比国际上高性能团队的标准,国内团队的 IT 性能从 2016 年到 2018 年有明显提升,但还存在巨大的差距,一方面说明了国内生产力的滞后,另一方面也说明了国内水平在逐渐提高,在发展中阶段感受到痛苦是正常现象.国内准高性能团队和准低性能团队的差距可能是源于两方面的,一方面是准高性能团队进步更快,另一方面是准高性能团队在商业目标上对于 IT 性能相对要求更高,这两方面最终体现在组织对准高性能团队在 IT 性能上不断提出更高的要求.

**发现 16.** 国内性能高的团队反而更害怕部署,与国际上报告出来的结果相反.这实际上是,标准逐渐与国际接轨,而生产力却发展滞后的体现.

### 4.6 障 碍

在 Puppet 的 2013 年的报告中,识别出了采用 DevOps 所面临的 6 个最大的障碍.

- 团队不理解 DevOps 的价值(48%);
- 开发和运维团队没有相同的管理结构(43%);
- 没有工具支持(33%);
- 没有时间尝试 DevOps(31%);
- 缺乏成功转型的支持和保障(19%);
- 尝试转型 DevOps 的成本过高(5%).

这 6 条障碍与我们所收集到的制约国内企业实践 DevOps 面临的障碍非常相似.主要体现在固有的组织文化和结构与 DevOps 的要求差异较大,为了实现变革所需的工具、人力等的资源配置不足以扭转差异.企业的最终目标是实现商业价值的最大化,所以 DevOps 的转型与否,投入多少进行转型的根本问题在于,转型所带来的回报是否大于转型所需的开销.这一问题很难给出一致的标准答案,DevOps 所能带来的比如交付周期上的变化并不对于所有业务都是必须的,对于电商而言,一分钟的宕机意味着数以万计的财富的销售额的流失,而对于一个有着充分技术壁垒的非在线的软件产品开发的企业而言,交付周期、部署频率等指标可能并不需要追求极致.

**发现 17.** 国内企业当前在实践 DevOps 所面临的障碍与国际上 2013 年所面临的障碍相似,其根本问题在企业对于 DevOps 的态度和需要的程度.此外,还有一个突出的问题在于,国内面临着独有的 DevOps 人才紧缺的问题.



### 5 讨论

通过对两次调查研究结果的分析与对比,以及与 Puppet Labs 的报告的对,我们共总结出了 17 条发现.综合分析这 17 条发现,梳理出了 3 条综合发现,进一步地,基于 3 条综合发现,我们在实践和研究这两个方面提出了多项建议,以思维导图的形式展现出来,如图 14 所示.

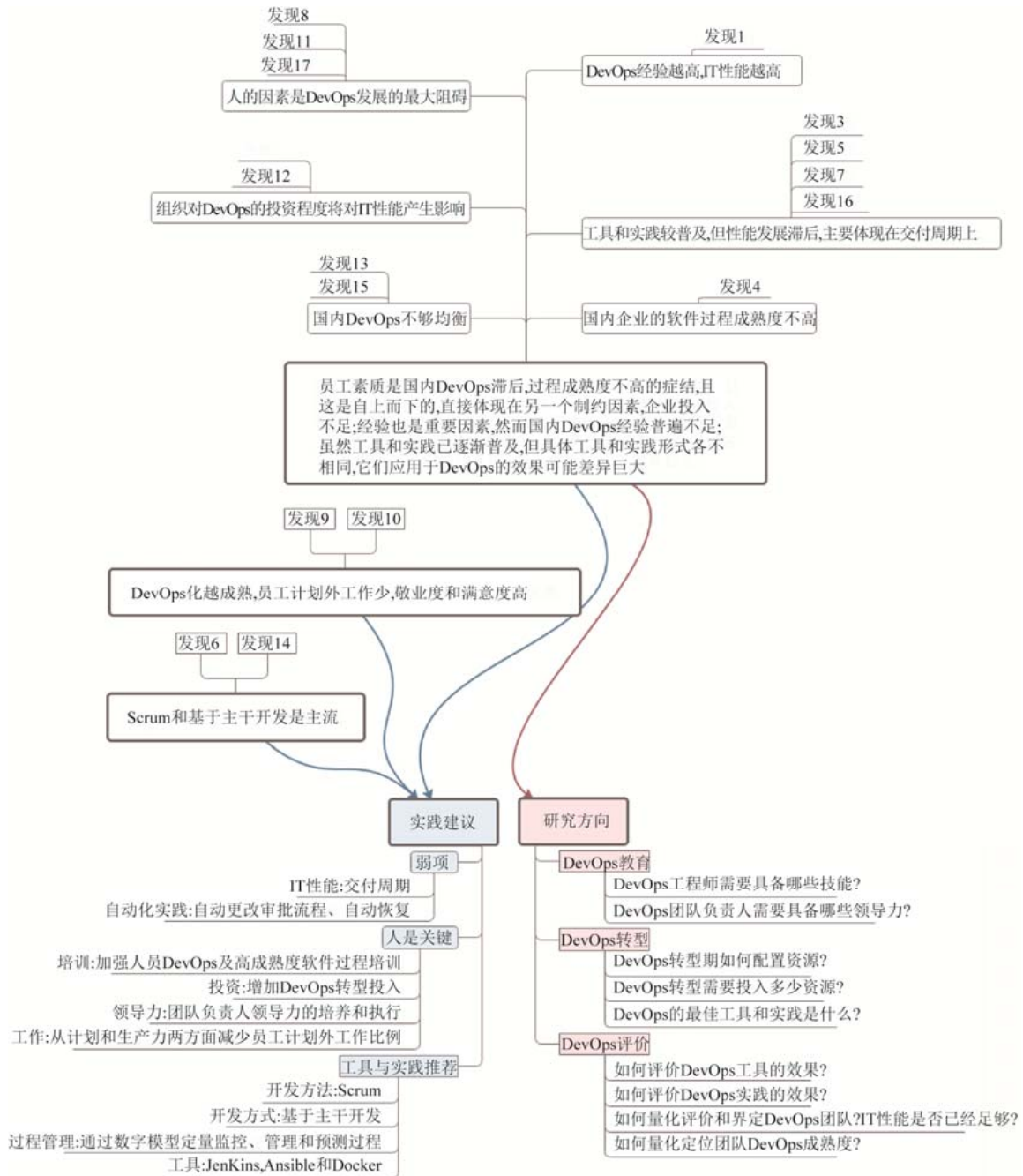


Fig.14 Recommendations for practice and research based on survey findings

图 14 基于调查发现的实践与研究方向建议

## 5.1 综合分析

**综合发现 1.** 基于发现 3、5、7、16,我们发现虽然国内的工具和 DevOps 实践在各个团队中已逐渐普及,且普及程度较高,但与国际水平相比,IT 性能表现依然不够理想,发展不够均衡(发现 13,15),变更失败比例虽有良好表现,但交付周期还有明显不足.基于发现 8、11 和 17,可以发现人的因素是 DevOps 发展的最大障碍,体现在领导力不足、人才紧缺、缺乏 DevOps 知识和技能 3 个方面,这与组织对人员培训的投资不足有着密切的关系(发现 2、12).大部分国内组织 DevOps 起步较晚,经验不足也是原因之一,DevOps 的成熟需要以经验为基础,循序渐进(发现 1).此外,我们还发现,大量组织的过程成熟度还不够高,例如,虽然组织普遍实现了标准化,但鲜有能够进一步构建数学模型定量管理和改进的过程(发现 4),以 CMMI<sup>[26]</sup>成熟度模型来评价,这些组织的过程成熟度是难以达到 4 级的,这可能也是制约组织推行 DevOps 的一个较大障碍.

**综合发现 2.** 我们可以看到 DevOps 给组织及员工带来的好处,随着 DevOps 的逐渐成熟,员工的计划外工作将减少(发现 9),这有利于软件过程的可管理性和易预测性.这样的组织,员工将有更高的敬业度和满意度(发现 10),这有利于组织的可持续性发展.

**综合发现 3.** 发现 6、14 说明 Scrum 是目前主流过程方法,而在开发方式上,基于主干开发在产业界是主流.

## 5.2 实践建议

以 17 项发现作为证据,3 项综合分析作为过程结果,我们从弱项、人、工具及实践这 3 方面给出实践建议.

- 从调查结果来看,国内组织 IT 性能的主要弱项是交付周期;自动化实践中的自动更改审批流程和自动恢复相较于其他自动化实践应用较少.这些弱项可以作为组织未来改进的着眼点.

- 在工具和培训上加大投资,强化员工的知识和技能以及相关的工具支持.同时,关注和提高团队负责人的领导力,从而提高组织的软件过程成熟度,这些是 DevOps 化的基础.项目及工作计划会因此更加准确,团队生产力会有所提高,员工计划外工作会减少,从而,员工的敬业度和满意度会提高,形成良性循环.

- 工具和实践的推荐基于受访者采用的比例.开发过程推荐 Scrum 方法,开发方式推荐基于主干开发,工具推荐 JenKins、Ansible 和 Docker.虽然通过对过程构建数学模型进行定量管理的组织较少,但这是组织寻求突破的必要手段.

## 5.3 研究方向

基于调查中发现的问题,对于研究方向从教育、组织转型、评价 3 方面,我们提出了多个研究问题.

DevOps 教育问题.

- DevOps 工程师需要具备哪些技能?前文我们解释了人的重要性,国内 DevOps 人才紧缺.一名 DevOps 工程师具体需要哪些技能,如何培养一名 DevOps 工程师,目前仍然没有标准答案.

- DevOps 团队负责人需要具备哪些领导力?除了培养 DevOps 工程师以外,提升团队负责人的领导力也是必要的,虽然对于领导力等已有大量相关研究,但在 DevOps 背景下,对团队负责人有哪些新的要求仍是未知的.

DevOps 组织转型问题.

- DevOps 转型期如何配置资源?组织如何转型,在转型中如何配置资源是一个庞大的且众说纷纭的命题,采用经验研究方法,可能可以为该问题寻求到一个合理的答案,为更多组织科学地实现 DevOps 化提供基础.

- DevOps 转型需要投入多少资源?转型必然意味着资源的投入,我们观察到有大量组织对 DevOps 还持观望态度,一方面不知道组织当前是否需要 DevOps 化,另一方面对需要多少投入心里没底.

- DevOps 的最佳工具和实践是什么?工具和实践是 DevOps 化落地的基础,为组织从纷杂的工具集和实践集中选择最合适的集合,可以让组织少走弯路.

DevOps 评价问题.

- 如何评价 DevOps 工具的效果?根据我们的不完全统计<sup>[20]</sup>,目前有超过百种支持 DevOps 的工具,建立工具评价的统一标准可以为工具选择提供支持.

- 如何评价 DevOps 实践的效果?相比于评价工具,对 DevOps 实践的评价难得多,但却是必要的。
- 如何量化评价和界定 DevOps 团队?IT 性能是否已经足够?目前,无论是产业界还是学术界,对于 DevOps 都还没有明晰的界定,对于 DevOps 化的程度也还没有类似 CMMI 的科学的、公认的评价模型.借鉴 Puppet Labs 的工作,本研究采用 IT 性能来量化度量组织的能力,这些是与 DevOps 最相关的性能,但其与组织的 DevOps 程度没有直接的量化的因果关系,且其不可直接用于对软件过程的评价。
- 如何量化定位团队 DevOps 的成熟度?如果能够量化地评价组织的 DevOps 化程度,下一步可以对组织的成熟度进行科学的分级,这应该是基于证据和最佳实践的,而不是单纯地基于一些数值。

## 6 有效性威胁

本研究采用了问卷调查的经验研究方法,结合 2016 年和 2018 年两次问卷调查结果分析 DevOps 在中国的发展现状与趋势.本节将从反馈数量、数据质量和统计分析 3 个方面分析有效性威胁以及我们采取的削弱威胁的措施.

**反馈数量.**为了尽可能地收集到更多的反馈,我们两次问卷都是采用在线问卷的形式,问卷通过技术社区、产业界及学术界会议等渠道分发,例如在会议上展出含问卷二维码的海报,所以实际分发了多少问卷是无法统计的,这导致了度量问卷调查效率的重要指标——反馈率的值是未知的.为了提高反馈数量,我们还将报告结果作为激励吸引更多人参与,目前已经有 DevOps 经验的从业人员,往往收入水平较高,奖品和金钱对于他们缺乏吸引力,在 DevOps 刚刚兴起时即有经验的从业人员往往对于技术有较高的求知欲,一份能够让他们了解到自己和其他从业者所处的水平以及遇到的瓶颈的调查报告,将更具吸引力,所以我们没有采用奖品或金钱的激励方式.

**数据质量.**为了确保反馈数据的质量,我们采取了一系列措施.首先,在卷首注明了回答问卷需要的时间,使受访者有心理准备.此外,我们还在卷首声明“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”,在问卷设计上也避免了问及姓名、公司名等私密信息,以此减少受访者的顾虑.在问卷问题的设计上,我们主要采用的是封闭式问题,且提供“其他”“不知道”或“以上都没有”等选项,避免受访者在不理解的情况下胡乱填写.我们的问卷分发渠道面向专业人士,激励方式与调查本身相关,这也为数据质量提供了一定保障.从结果来看,确实没有收到无效反馈.

**统计分析.**虽然我们采取了多项提高反馈数量的策略,然而,2016 年和 2018 年的反馈数量分别仅为 74 和 66,这使得结果在统计上显著性不足,所以在分析上我们尽可能地比较两年的结果,以此减少统计结果对于样本差异的敏感性.从最终的分析结果来看,例如质量保证实践等,虽然在占比量上两年存在差异,但在实践占比的排序上,两年差异较小,支撑了结果的统计意义.在问题选项的设置上,我们尽可能地细粒度,例如对于“团队平均多长时间部署一次代码?”,我们从“<15 分钟”到“>6 个月”共设置了 9 个选项,但在统计分析上,我们基于数据的分布,对选项进行了合并,通过小于/大于 1 周作为准低性能和准高性能团队的分水岭,由于受访企业的能力在调查前是未知的,设置细粒度的选项可以减少用于分析的数据在分布上受问卷设计者的主观偏见,在具体分析时合并选项可以确保各个类别的数量,使我们分析时具备统计意义.

## 7 结束语

本研究基于两次中国 DevOps 现状调查,结果显示,DevOps 经验丰富的团队在 IT 性能表现上明显更佳,高低性能的团队在 IT 性能上会存在数十倍的差距.这反映出 DevOps 在支持和提升中国软件行业效率和质量上有着明显效果,为 DevOps 运动在中国的进一步推广提供了充足的极具参考价值的证据.从另一角度来看,调查报告反映的内容能够帮助学术/科研领域真实、全面地了解 DevOps 在国内软件产业的发展的整体状态,进一步发现和解决这其中具有实践意义和价值的问题,使软件工程研究落地服务于软件产业的最新发展.

具体而言,我们从受访团队的 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、障碍这 8 个方面,对 DevOps 在中国近两年的现状与发展进行了分析,并将部

分结果与 Puppet Labs 发布的全球 DevOps 现状报告进行了对比。目前,国内团队的 IT 性能表现在交付周期上还存在明显不足,从趋势上看,随着 DevOps 经验的提高,IT 性能表现也会随之提高,IT 性能表现高的团队,在文化、自动化等实践上明显优于较低性能的团队。这体现了 IT 性能、实践、经验三者之间相辅相成、循序渐进的过程。值得警惕的一点是,当前国内组织在转型 DevOps 上正面临着国际上 5 年前识别出来的障碍,且在人才资源上面临着更严峻的挑战。相应地,国内组织在 DevOps 上的投入和力度虽然平均已到达国际上 5 年前的水准,却存在明显的两极分化现象,整体上发展不够良性,这将制约 DevOps 在中国的进一步发展和推广。此外,国内 DevOps 实践更佳的团队,其成员反而更觉得部署是一件痛苦的事,这反映了国内一部分高标准的企业正处于 DevOps 转型的阵痛期,而另一部分企业发展较缓。本文结合两年的问卷调查研究和 Puppet Labs 的调查报告,分析获得了 17 条发现,综合分析这些发现,我们在实践和研究两方面分别给出了建议。本研究最大的局限在于,虽然竭力推广,但收到的问卷反馈数量相对有限,这也从侧面反映了国内企业或个人对于 DevOps 的态度还不足够积极。此外,参与到调查中的受访者更多的是对 DevOps 有着浓厚兴趣的从业人员,从他们的反馈数据分析出来的结果,虽已然体现出了中国 DevOps 发展的滞后,但真实情况,可能更不容乐观。

## References:

- [1] Cohen D, Lindvall M, Costa P. An introduction to agile methods. *Advances in Computers*, 2004,62:1-66.
- [2] Schwaber K, Beedle M. *Agile Software Development with Scrum*. Upper Saddle River: Prentice Hall, 2001.
- [3] Beck K. *Extreme Programming Explained: Embrace Change*. Addison-Wesley, 2000.
- [4] Poppendieck M, Poppendieck T. *Lean Software Development: An Agile Toolkit for Software Development Managers*. Boston: Addison-Wesley, 2003.
- [5] Ahmad MO, Markkula J, Ovio M. Kanban in software development: A systematic literature review. In: *Proc. of the 39th Euromicro Conf. on Software Engineering and Advanced Applications*. IEEE, 2013. 9-16.
- [6] Bass L, Weber I, Zhu L. *DevOps: A Software Architect's Perspective*. Addison-Wesley Professional, 2015.
- [7] Smeds J, Nybom K, Porres I. DevOps: A definition and perceived adoption impediments. In: *Proc. of the Int'l Conf. on Agile Software Development*. Springer-Verlag, 2015. 166-177.
- [8] Puppet Labs. 2013 State of DevOps Report. 2013. <https://puppet.com/resources/whitepaper/>
- [9] Lwakatare LE, Kuvaja P, Oivo M. Dimensions of DevOps. In: *Proc. of the Int'l Conf. on Agile Software Development*. Springer-Verlag, 2015. 212-217.
- [10] Kim G, Humble J, Debois P, Willis J. *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*, 2016.
- [11] Jabbari R, Ali N, Petersen K, Tanveer B. What Is DevOps? A systematic mapping study on definitions and practices. In: *Proc. of the Scientific Workshop of XP*. ACM, 2016. 1-11.
- [12] Erich F, Amrit C, Daneva M. Report: Devops literature review. Technical Report, University of Twente, 2014.
- [13] Puppet Labs. 2014 State of DevOps Report. 2014. <https://puppet.com/resources/whitepaper/>
- [14] Puppet Labs. 2015 State of DevOps Report. 2015. <https://puppet.com/resources/whitepaper/>
- [15] Puppet Labs. 2016 State of DevOps Report. 2016. <https://puppet.com/resources/whitepaper/>
- [16] Puppet Labs. 2017 State of DevOps Report. 2017. <https://puppet.com/resources/whitepaper/>
- [17] Shahin M, Babar MA, Zhu L. Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices. *IEEE Access*, 2017,5:3909-3943. [doi: 10.1109/ACCESS.2017.2685629]
- [18] Ur Rahman AA, Helms E, Williams L, Parnin C. Synthesizing continuous deployment practices used in software development. In: *Proc. of the Agile Conf*. IEEE, 2015. 1-10.
- [19] Ebert C, Gallardo G, Hernantes J, Serrano N. DevOps. *IEEE Software*, 2016,33(3):94-100.
- [20] Li SS, Li Z, Zhang H, Liu B, Rong GP. Empirical study on the practical status of DevOps automation and tools. In: *Proc. of the NASAC 2017*. 2017 (in Chinese with English abstract).
- [21] Rafferty AE, Griffin MA. Dimensions of transformational leadership: Conceptual and empirical extensions. *Leadership Quarterly*, 2004,(15):329-354.

- [22] Condo C, Giudice DL. Use DevOps and supply chain principles to automate application delivery governance. In: Forrester Research. 2016. 1–19.
- [23] Li G, Dong HM, Yang ZJ, Han HQ. Overview and analysis on software engineering standard. Journal of SiChuan University (Engineering Science Edition), 2007,39:303–307(in Chinese with English abstract).
- [24] Hanssen GK, Haugset B, Stålhane TT, Myklebust T, Kulbrandstad I. Quality assurance in Scrum applied to safety critical software. In: Proc. of the Int'l Conf. on Agile Software Development. Springer-Verlag, 2016. 92–103.
- [25] Elberzhager F, Arif T, Naab M, Süß I, Koban S. From agile development to DevOps: Going towards faster releases at high quality —Experiences from an industrial context. In: Complexity and Challenges of Software Engineering in Emerging Technologies. Springer-Verlag, 2017. 33–44.
- [26] Team CP. CMMI® for Development. Version 1.3, Improving processes for developing better products and services. Technical Report, CMU/SEI-2010-TR-033, Software Engineering Institute, 2010.

#### 附中文参考文献:

- [20] 李杉杉,李铮,张贺,刘博涵,荣国平.DevOps 自动化支持工具及实践现状经验研究.见:第 16 届全国软件与应用学术会议.2017.
- [23] 李刚,董火民,杨子江,韩红强.软件工程标准化现状与分析.四川大学学报(工程科学与技术),2007,39:303–307.



刘博涵(1991—),男,湖北武汉人,博士生,CCF 学生会员,主要研究领域为软件过程挖掘,软件过程度量.



董黎明(1994—),女,博士生,主要研究领域为软件过程挖掘.



张贺(1971—),男,博士,教授,博士生导师,CCF 专业会员,主要研究领域为软件过程,软件体系结构,服务计算,经验软件工程领域的科研和实践.