

A dark, moody photograph of a wooden desk. In the top right, a portion of a silver laptop is visible. To its left is a silver pen and a small, blank, cream-colored notebook. In the bottom center, a pair of black-rimmed glasses lies on the wood. In the bottom right corner, a brown pinecone is partially visible. The overall lighting is low, creating a professional and focused atmosphere.

2016 China·DevOps发展报告

南京大学软件学院

目录

01 | 调查对象情况统计
Who Took the Survey ?

02 | IT 性能表现
IT Performance

03 | 实践活动
Practice Activities

04 | 阻碍
Barriers

05 | 员工敬业度
Employee Engagement

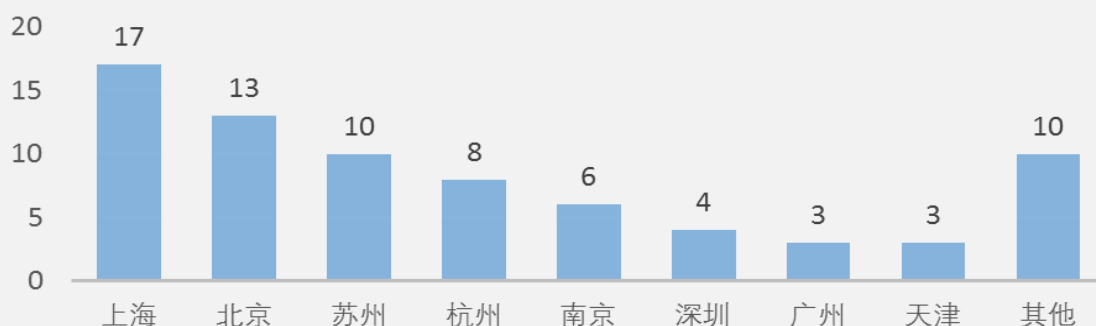
调查对象情况统计

这次调查共计74家组织以及个人开发者参与，这些被调查对象来自于和DevOps有关的各行各业，从金融到政府，从IT运维到销售，这说明DevOps应用范围的广度，同时，考虑到这份问卷发布两个月仅有70余人回复的情况，我们认为DevOps在国内的流行程度不是很高。

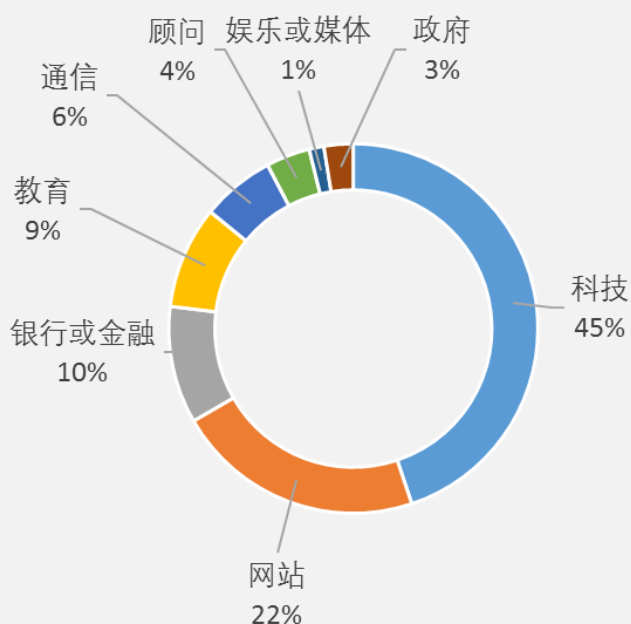
在这次调查中，大部分人员和组织都是在最近一年内才有DevOps相关的实践经验，团队中DevOps工程师占比17%。

由于这是开展调查的第一年，我们无法看出DevOps在近几年的流行趋势，仅能给出对于现状的判断。

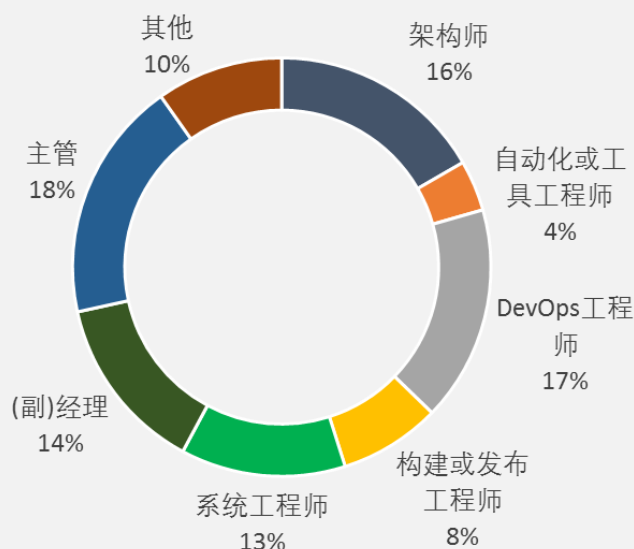
工作地点所在城市



行业

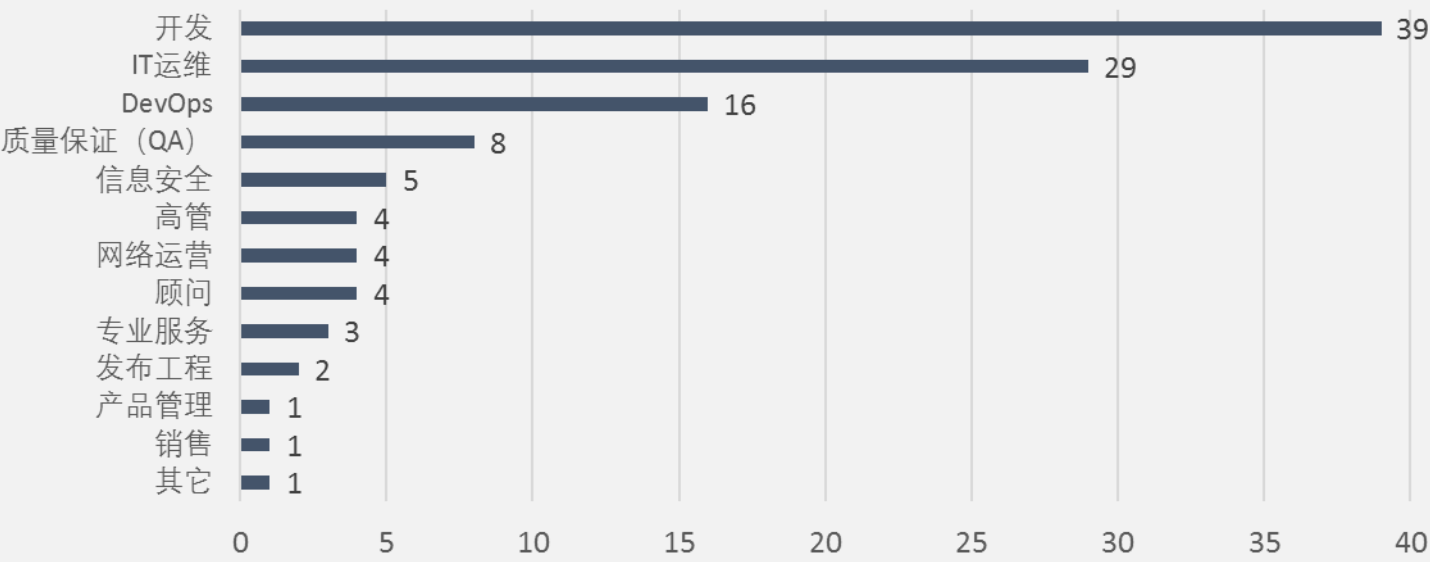


职位

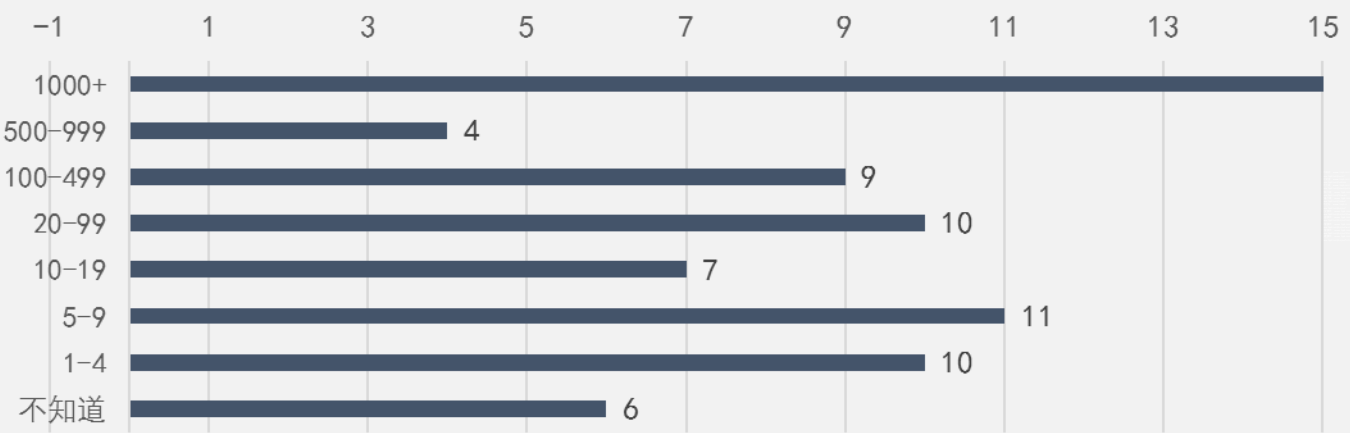


调查对象情况统计

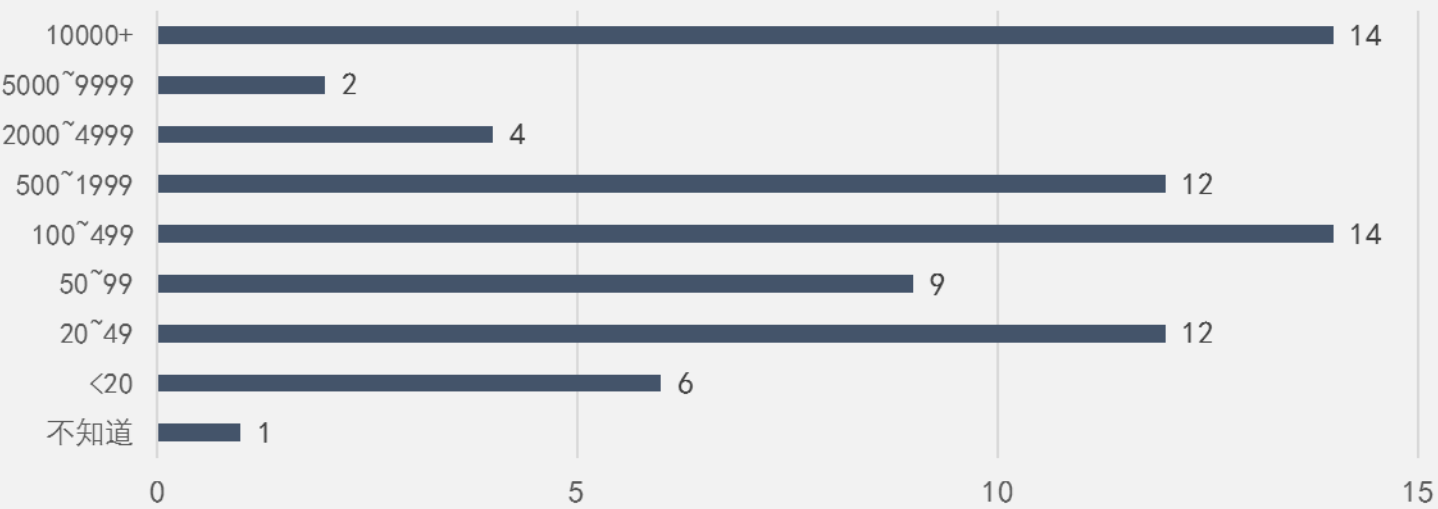
所在部门



所在组织的服务器个数

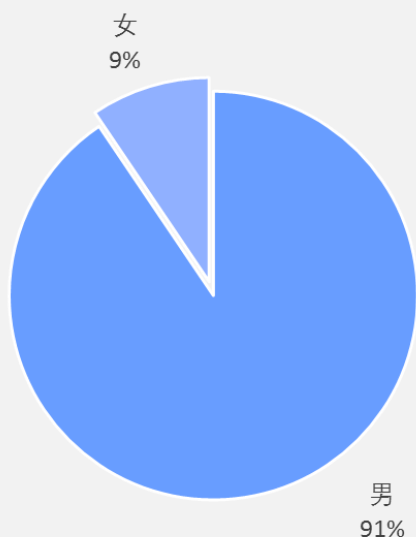


所在组织员工数量

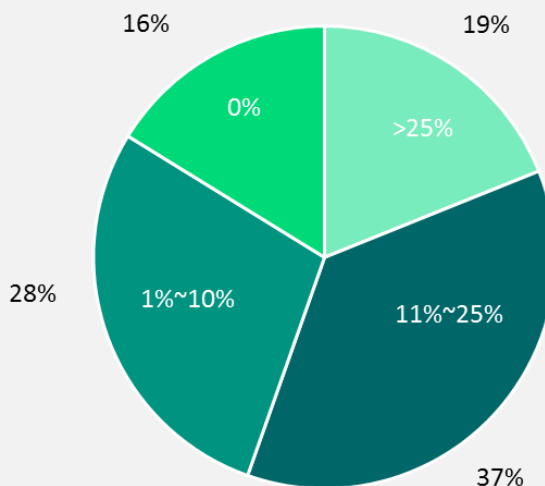


调查对象情况统计

调查对象性别



团队中女性比例



参与调查的人员中，团队中女性人数多于1/4的占19%，可以看出在这个行业依然以男性居多，16%的团队中没有女性的存在



91%

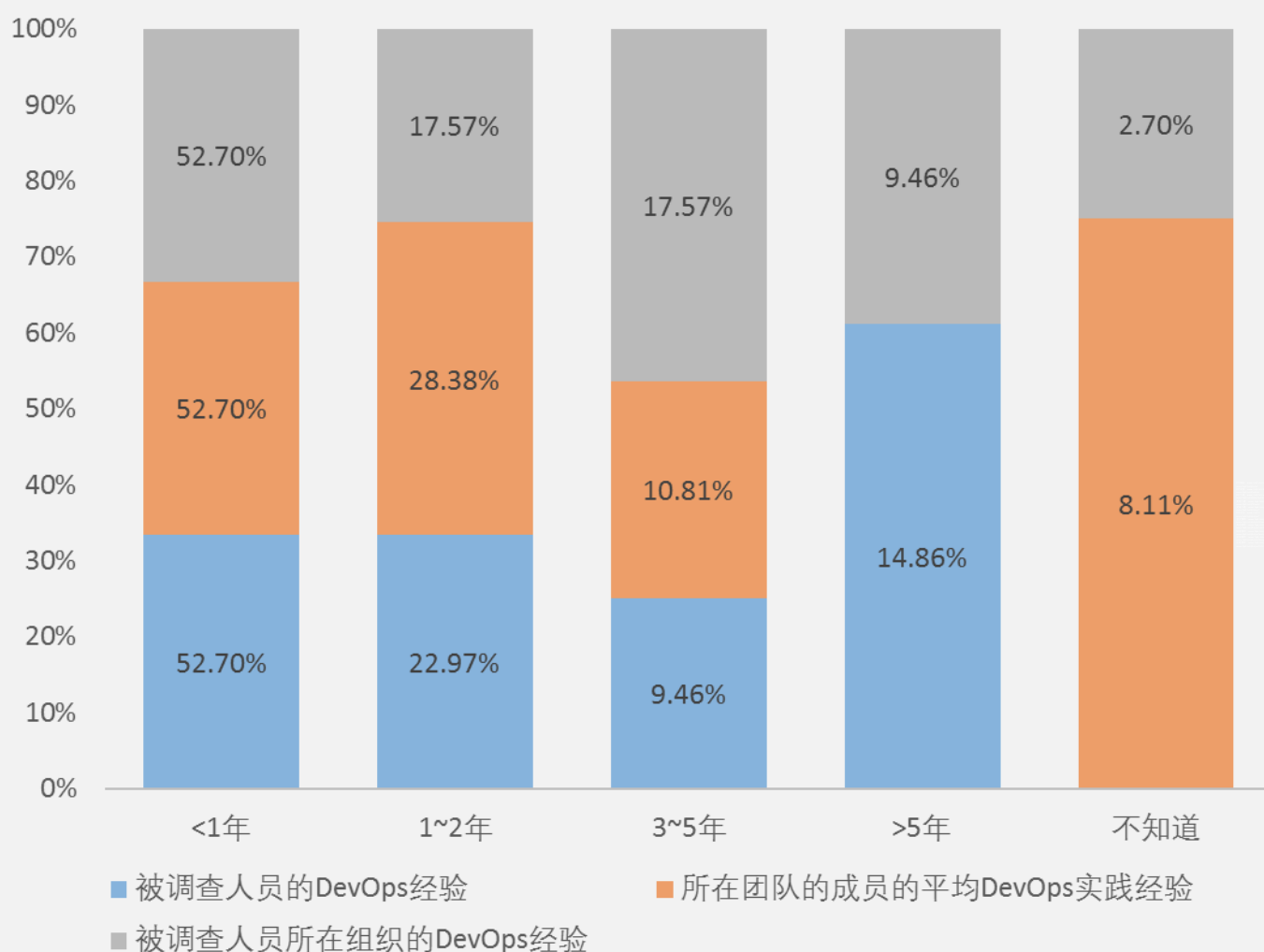


9%

调查对象情况统计

我们对被调查人员以及被调查人员所在团队的成员和被调查人员所在组织的DevOps经验进行了数据采集，发现超过50%的成员和组织的DevOps实践经验都是在最近1年内。

实践经验



接近80%的人员和组织的DevOps经验都在2年之内，说明对于大部分被采访者来说，DevOps是一个新技术。

IT 性能表现



如何定义“高效的组织？”

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的向市场提供具有关键业务的应用，并且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。

组织是否能够快速的把一个想法变成价值交付到客户手中？
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？

在此次调查中，为了更好的衡量DevOps对一个组织的生产活动的影响，我们将这些方面进行了量化，最终归结在部署频率、交付周期、平均修复时长、变更失败比例这四个参数上。

IT 性能表现

衡量敏捷性的参数

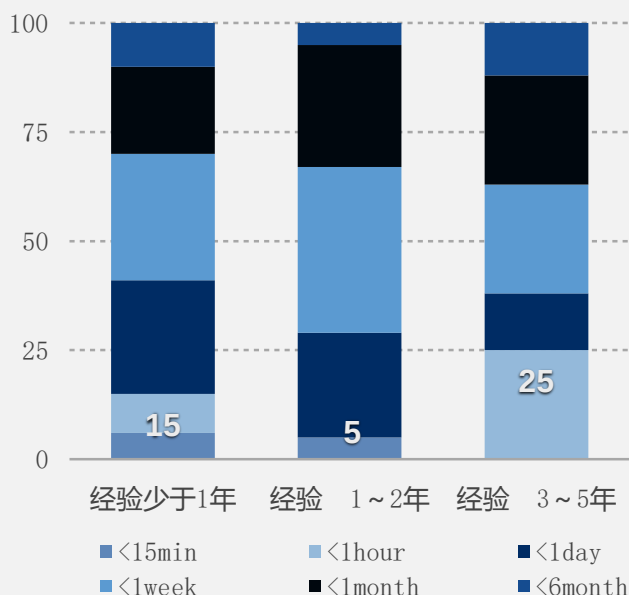
部署频率

维持较高的部署频率（即，固定时间内较多的迭代次数）使得一个组织在快速的将想法变成价值的同时进行多次尝试，加快了商业概念到业务价值的转变。

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。

在这次调查中我们发现，32.4%的团队在一天之内进行多次部署，与部署频率超过一周的团队相比，他们对于市场变化和用户反馈反应更加迅速。

DevOps经验与部署频率



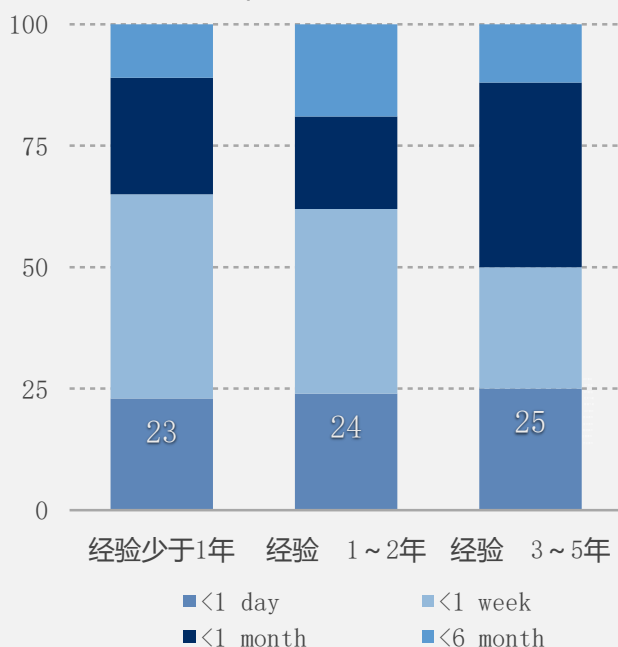
交付周期

交付周期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。

2013年的关于DevOps现状的全球化调查中，已经有超过23%的DevOps实践经验大于一年的团队将交付周期控制在一个小时之内。并且DevOps实践经验和交付周期之间存在显著的正相关性（即实践经验越长的团队中，交付周期少于一小时的团队比重越大）

在我们的样本范围内，交付周期少于一天的占比21%，国内的交付周期与DevOps实践经验无明显联系。

DevOps经验与交付周期



IT 性能表现

衡量可靠性的参数

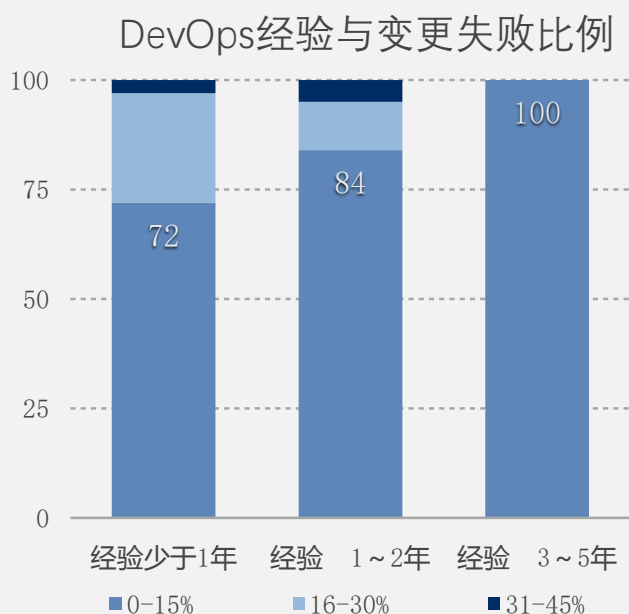
平均修复时长

高效的团队在应对一些意外故障时，能够将恢复时间控制在几分钟之内，是低效团队（耗费一小时甚至更多的时间）恢复速度的30倍。

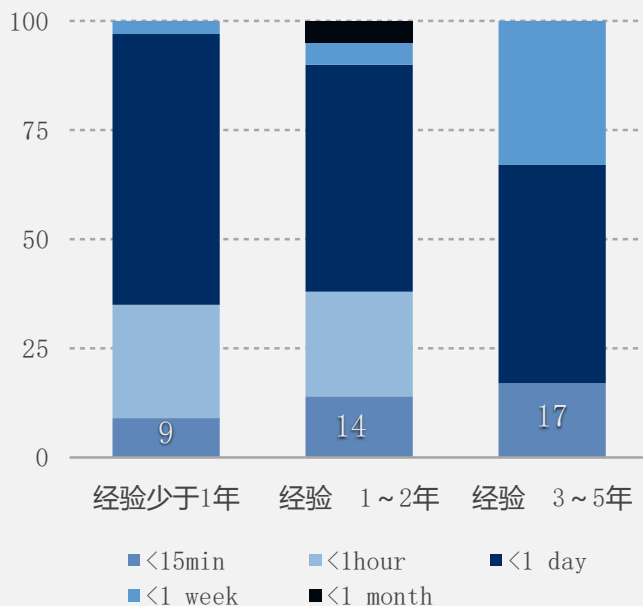
很多组织都无法忍受哪怕是几分钟的故障修复时间，这意味着数以百万计的利润的流失与用户满意度的下降。

在2013年的全球性调查中，平均修复时长小于一小时的组织占比为31.5%。

在这次调查中，13.5%的组织能够将平均修复时长控制在15分钟之内，32.9%的团队能够在一小时之内完成修复，考虑到DevOps在我国还处于流行的初级阶段，这个比例已经相当不错了。



DevOps经验与平均修复时长



变更失败比例

变更失败比例主要考虑变更导致的服务降级或随后的修复，比如导致服务受损、服务中断、需要热修复、回滚、打补丁等情况。变更失败的主要原因是开发、测试环境与生产环境的不匹配。

当一次变更失败后，组织需要花费时间和资源对其进行修复，可能对开发、发布、更新产品等阶段造成不同程度的恶劣影响。

此次调查中我们看到，66.2%的团队能够将比例控制在0-15%中，同时我们发现，变更失败比例超过15%的团队DevOps实践经验均小于两年。

IT 性能表现

综合考虑，我们给出对于高性能和低性能的衡量标准。

	高性能	低性能
部署频率	一天之内	超过一周
交付周期	一天之内	超过一天
平均修复时长	15分钟之内	超过一天
团队变更失败比例	0%-15%	大于15%

遗憾的是，在被调查团队中，只有6.7%的团队能够达到我们所规定的高性能指标。尽管综合表现非常出色的团队较少，但是在部署频率与变更失败比例这两个参数上均有超过半数的团队表现相当出色。

在部署频率项上，有32.4%的团队可以实现平均一天实现多次部署，这表明有接近三分之一的团队在部署频率方面能达到极高性能。

而在交付周期上，能在1天之内交付的比例为22.9%，说明大部分团队在交付周期此项指标上性能表现不佳。

在平均修复时长方面，83.78%的团队可以在1天之内修复故障，而将此标准提高到1小时之内的话，只有32.4%的团队能够满足要求。说明大部分用户能达到较高水平，但是只有接近三分之一的用户能够达到极高性能的修复水平。

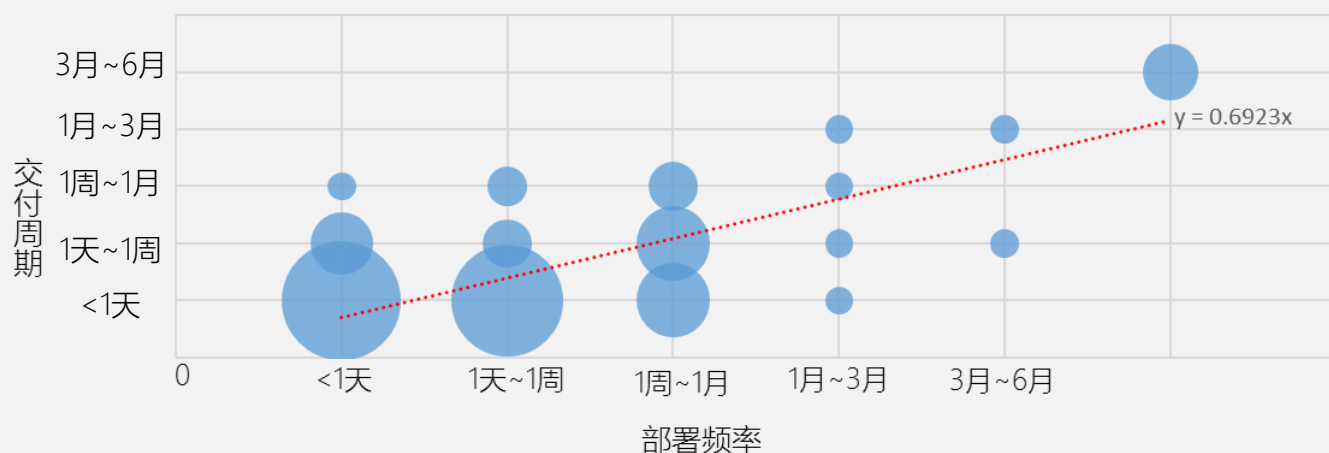
在变更导致失败的稳定性因素上，66.2%的团队将变更导致失败的比例控制在0-15%这一区间，仅有2.7%的团队比例在31~45%区间

- 变更失败比例在16~45%之间的团队DevOps经验均小于两年
- DevOps经验在3~5年之间的团队，除去不清楚团队变更失败比例的，其余均控制在0~15%之间，均达到高性能的变更失败比例

IT 性能表现

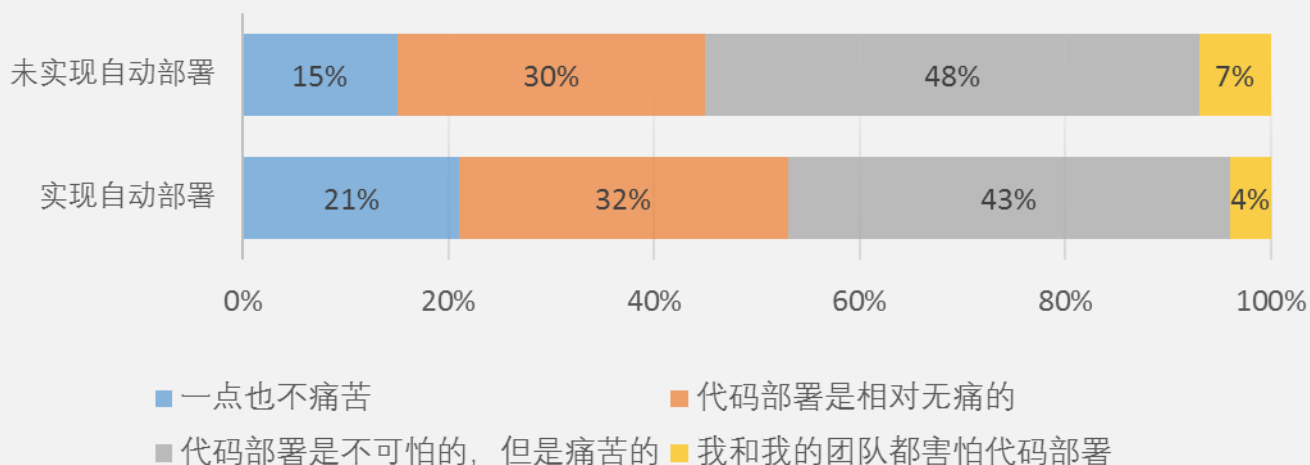
这次调查中，**超过三分之一的公司实现了一天多次部署**，尽管这个指标与DevOps经验并无显著关系，我们发现部署频率与交付周期直接存在明显的正相关，即部署频率越高的团队在交付周期的表现上越好，这说明部署频率的提高确实有助于提高产品对市场需求变化作出快速变更的可能性。

部署频率与交付周期的关系



当团队实现自动部署时，部署的主观感受整体优于未实现自动部署团队。

部署感受



由于问卷调查基数较小，调查结果只能代表部分中国团队DevOps的现状。

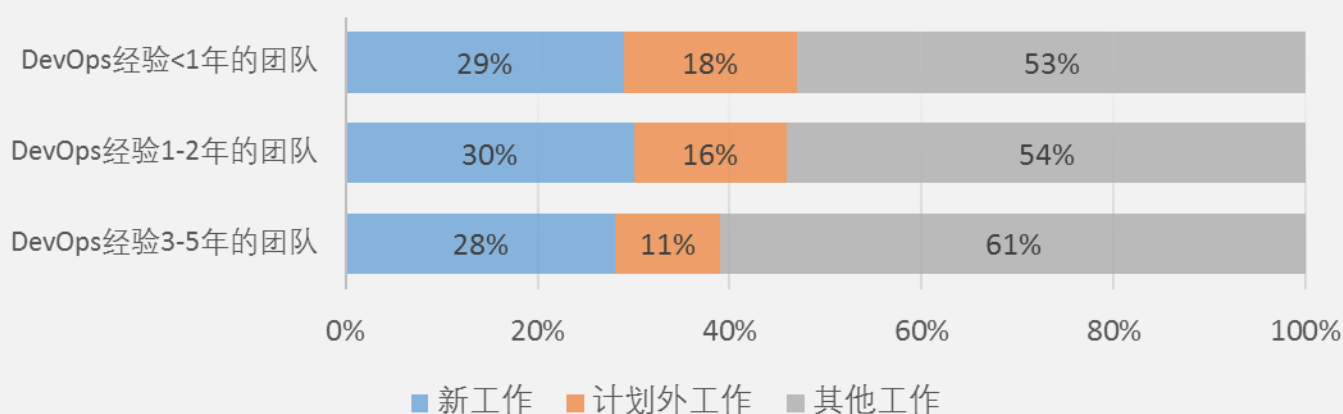
IT 性能表现

DevOps提倡开发和运维之间的高度协同，从而在完成高频率部署的同时，提高生产环境的可靠性、稳定性、弹性和安全性。典型的价值流是在业务（定义需求）和客户（交付价值）之间产生的，越来越多的组织认识到，缩短价值产生的时间，会在很大程度上提高产品的商业价值，而这需要开发和运维人员的高度配合。

传统的软件开发过程是线性的，如瀑布模型，功能测试和安全测试发生在软件发布之后，在这个阶段修复错误需要付出非常昂贵的时间和金钱代价，可能需要重构整个架构。DevOps聚焦在创建快速和稳定的计划工作流，尽最大可能的消除系统的计划外工作。因此DevOps要求组织加强开发人员与QA和测试人员的交流与沟通，在开发的同时更加关注系统是否满足质量属性，这样能够更早的监测并处理开发中发生的错误，减少计划外工作。因此，组织花费在计划外工作上的时间可以反应其在软件开发过程中对质量属性的满足程度。

2013年的全球性DevOps调查报告中显示，高效的组织在计划外工作上花费的时间更少。我们对参与调查的组织花费在计划外工作的时间进行了统计，计划外工作主要包括紧急部署，紧急补丁或返工的工作等，新工作包括设计或构建新的功能等，其他工作包括会议、常规维护等。

不同类型工作的占比



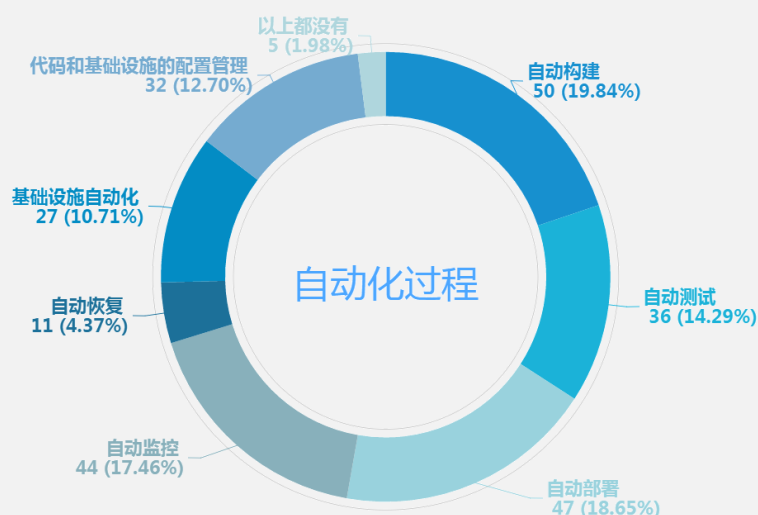
- 团队在新工作上花费的时间平均为30%左右
- 团队在新工作与计划外工作上花费的时间比例约为2:1
- 经验丰富的团队在计划外的工作或返工中花费的时间会更少
- 团队用在其他工作的时间超过了50%

实践活动

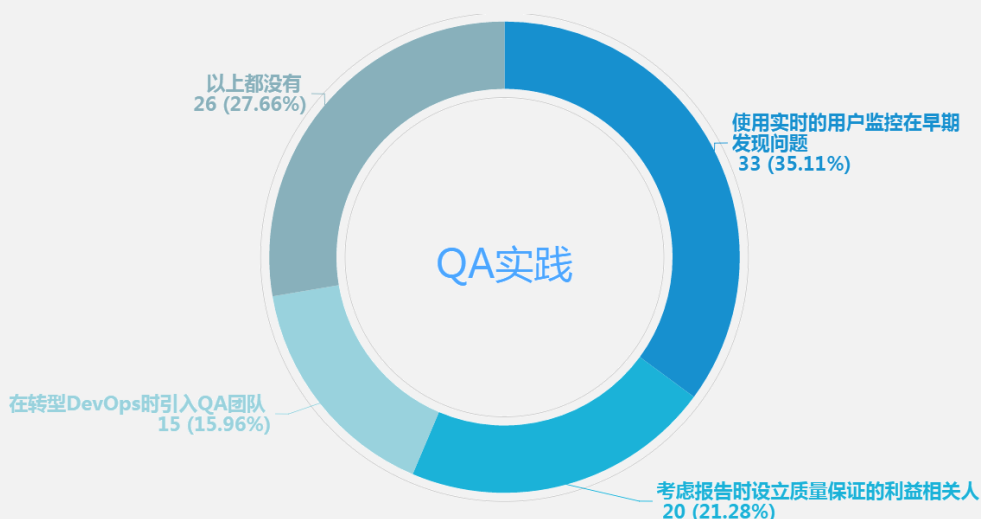
为了更进一步了解高性能团队的驱动因素，我们进行了团队实践活动的调查，主要关注DevOps及其他开发工程中比较关键的五个方面的实践活动，以此来分析实践活动与团队性能之间的联系。

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。DevOps的一大目的是打造完全自动化的Delivery Pipeline，自动化也是提高可测试性、一致性、稳定性、部署频率和达到持续交付核心。

在软件开发开发阶段的构建、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



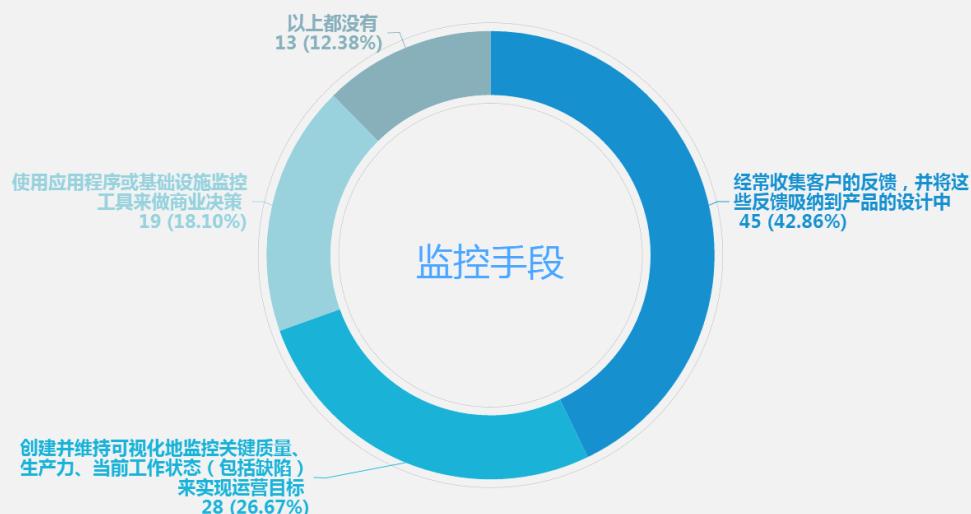
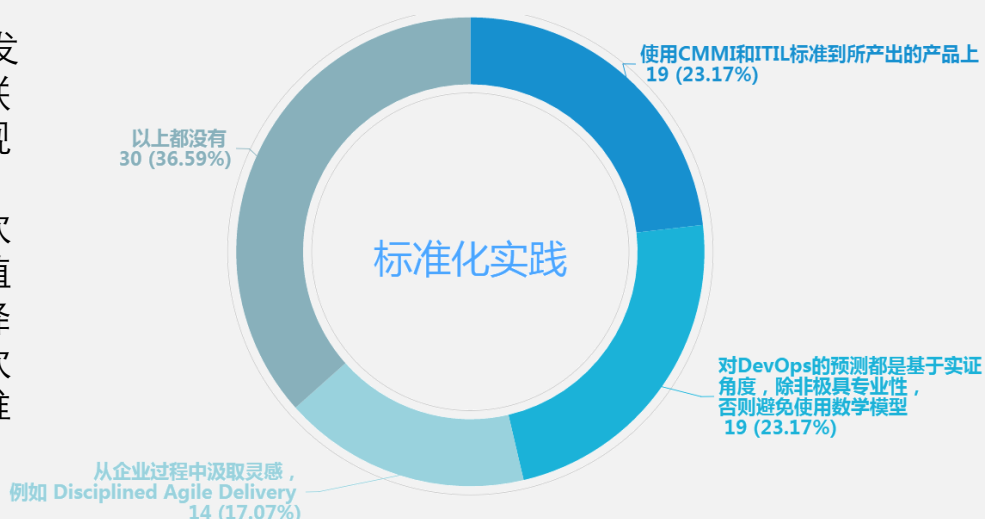
QA包括了一系列对软件系统进行检测来保证质量的手段，但是DevOps的高部署频率通常会给QA带来很大的压力，传统的QA往往跟不上DevOps的部署频率，或者使得服务质量下降，或者使得发布频率降低。如何将QA引入整个DevOps workflow是一大难题。



实践活动

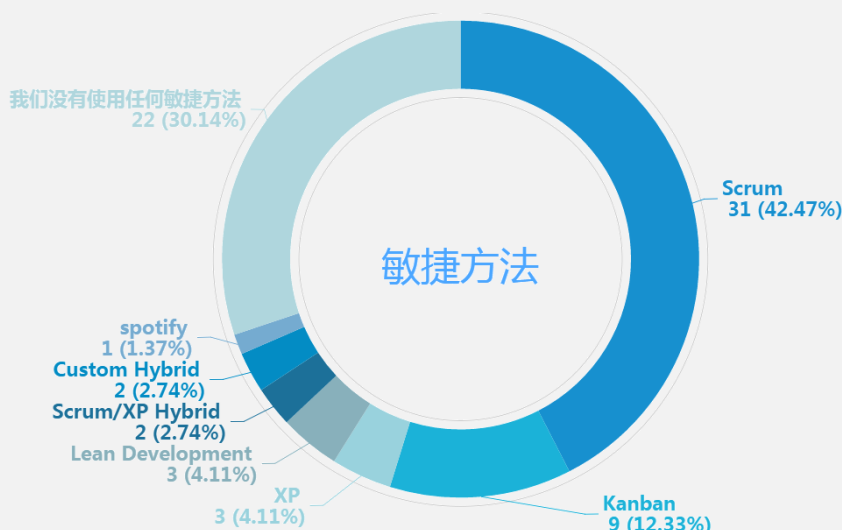
软件标准是为了协调开发项目的各个阶段和各个部分联系和衔接而做出的的约束和规定。

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。



为了更好、更有效的保障服务的稳定运作、服务产生的效绩。对于服务器信息（如工作状态等）与服务信息（如服务使用情况等）等都必须有一个可靠和可持续的监控机制，所产生的高频率的反馈和缺陷也正好能在DevOps的高部署频率上得到很好的满足与解决。

DevOps不等于传统的敏捷，但是不可否认的是，敏捷方法中提倡的自动构建、持续发布等特点与DevOps是一致的。我们将会考察敏捷方法对促进DevOps的开展与实施是否有明显作用。



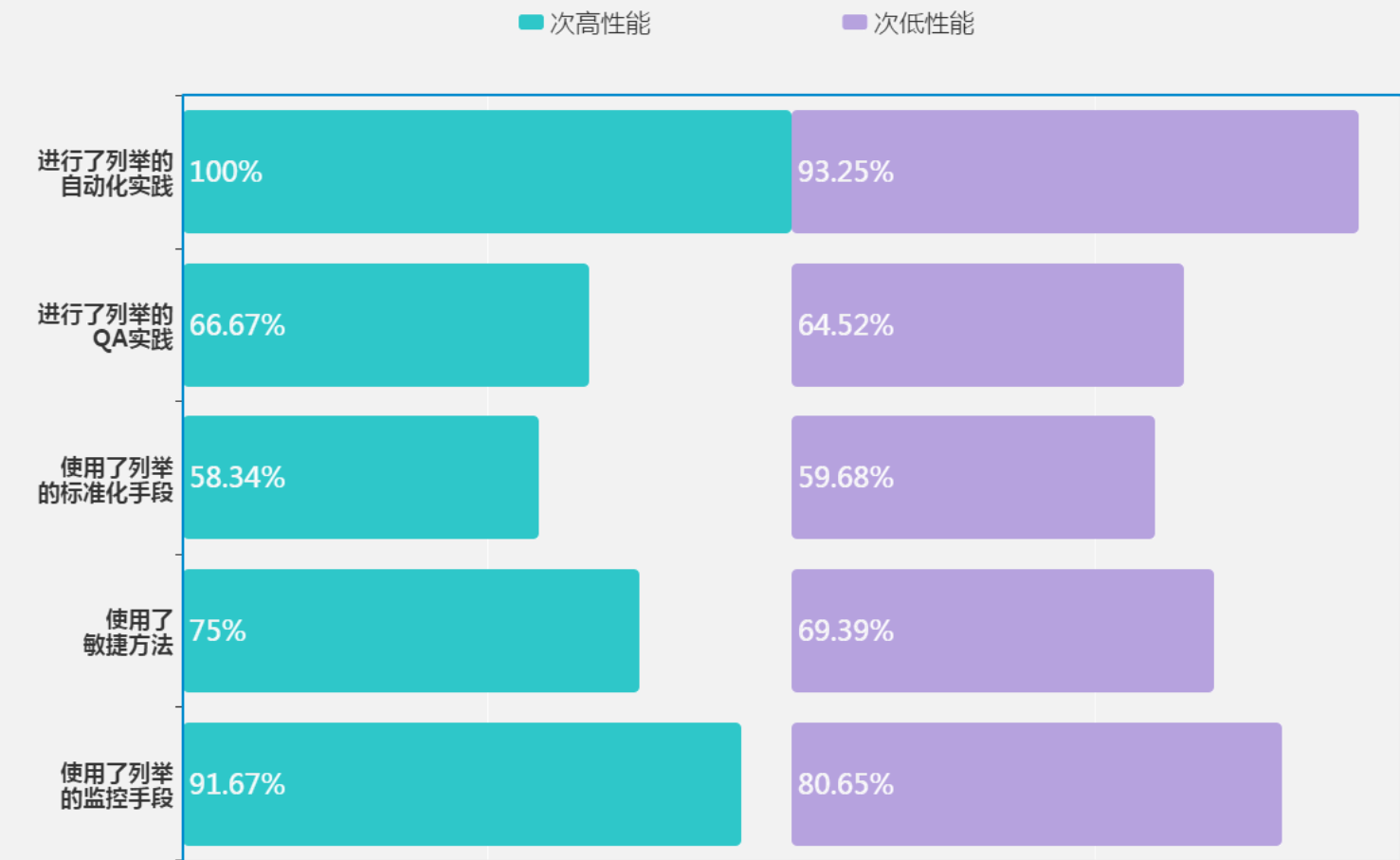
实践活动

由于高性能团队样本数过少，无法得出其与团队实践活动之间准确的关系，我们为了分析出较为准确与严格的结果，对部分指标进行放大，定义了一个新的次高性能，来分析其与团队实践活动之间的关系。

按照这样的标准，我们得到了16.2%的次高性能团队，其余83.8%的团队我们称之为次低性能团队。我们统计了未进行上述五种实践团队的数量分别在这两个团队中的占比，以期望得出那种实践活动对实现性能有较大的影响。

	次高性能
部署频率	一周之内
交付周期	一天之内
平均修复时长	一天以内
团队变更失败比例	0%-15%

团队性能与实践活动



实践活动

我们发现三个较为明显的结果：

*100%次高性能团队进行了列举的自动化过程，而次低性能为93.25%
91.6%次高性能团队使用了列举的监控手段，而次低性能为80.65%
其余实践活动比率上，两类团队大致相同*

自动化能为团队性能带来几个重大的影响。首先自动构建、测试、部署可以减少在不同环境中造成的配置上花费的时间和精力，也有助于提高部署的成功率。其次自动化能对大量重复的人工活动进行简化，使得团队的人员能够更加关注系统的关键业务逻辑。同时自动化还能让团队人员及时跟踪变化，在变化需要团队花费更大的精力应对它之前，及时让软件对变化进行相应的变更。

使用有效、可靠的监控手段对系统进行持续的监控能够提高系统的可靠性。将系统的关键服务、服务设施纳入到监控的范围中，能够为团队提供一个良好的错误、变更、系统状态的视图，使得系统对于错误和变更能够即时地进行应对，降低了因为错误和变更带来的损失；对服务的监控也为软件决策者做出相应的决策提供了可靠的数据支持。

从统计结果来看，实践活动对团队性能的影响主要体现在自动化过程和监控手段上。但是一个值得注意的问题是，即使这两个实践活动在两个团队间相对差别最大，但是也只有6.75%与10.35%的差距，且这两项实践活动都处于一个较高的使用水平（>80%）。所以我们认为国内进行的实践活动情况，从总体情况来看，对团队性能没有太大的影响；换句话说就是

国内软件开发团队的团队性能与其进行的实践活动没有明显的相关性。

这个结果有点出乎意料，但是考虑到调查样本数量的限制，以及调查人员对各个实践活动的理解，这只能反映部分团队的情况。

阻碍



应用DevOps模式中最常见的阻碍有：

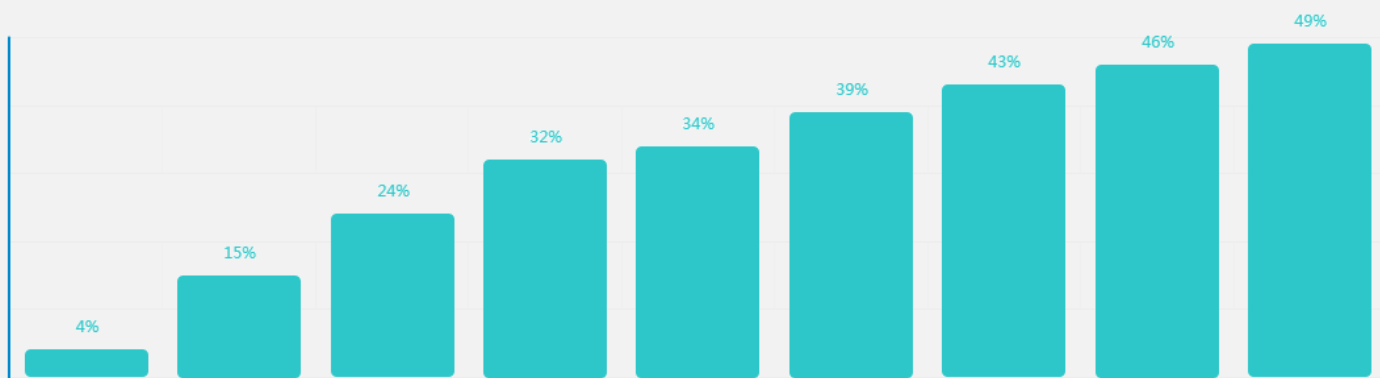
- 不同部门的目标不一致，开发人员关注的是新的功能的实现，而运维人员关注的是已有功能的成功运行
- 员工还不理解DevOps的概念
- 组织缺少具备DevOps经验的专家，导致推进缓慢甚至无从下手
- 缺乏配置使用相关工具的专业知识或缺少相关人才

阻碍

DevOps模式的应用有助于更快的构建可靠性更高、质量更好的软件，但仅靠这些实践不足以实现组织上的成功。

我们罗列了一些通过调查得到的在实现DevOps过程中常见的阻碍：

- 不同部门的目标不一致，开发人员关注的是新的功能的实现，而运维人员关注的是已有功能的成功运行（49%）
- 员工还不理解DevOps的概念（46%）
- 组织缺少具备DevOps经验的专家，导致推进缓慢甚至无从下手（43%）
- 缺乏配置使用相关工具的专业知识或缺少相关人才（39%）
- 组织行业的限制，比如金融行业，运维可能必须遵循某些规则（34%）
- 即使不同部门有着相同的目标，即使组织成功，个人往往优先效忠于自己的部门，然后是自己的组织（32%）
- 开发人员工作负担已经很重，额外的运维任务可能会使情况恶化（24%）
- 开发人员比运维人员的工资高大概50%，组织必须从收支的角度分析采纳哪些DevOps过程，哪些不采纳（15%）
- 必须为不同的开发团队定义相同的过程（4%）



阻碍

Cloud Technology Partners 公司的副总裁兼首席架构师Mike Kavis是这样描述DevOps的：

DevOps的关注点是从SDLC中移除浪费。通常情况下，发现浪费或者瓶颈的形式包括：不一致的环境，人工的构建和部署流程，差的质量和测试实践，IT部门之间缺少沟通和理解，频繁的中断和失败的协定以及那些需要珍贵的资源、花费重要的时间和金钱才能保持系统运行的全套问题。

我们注意到，有些常见的障碍在DevOps模式下理应是可以得到一定程度解决的：

- 不同部门的目标不一致，开发人员关注的是新的功能的实现，而运维人员关注的是已有功能的成功运行（49%）
- 即使不同部门有着相同的目标，即使组织成功，个人往往优先效忠于自己的部门，然后是自己的组织（32%）
- 开发人员工作负担已经很重，额外的运维任务可能会使情况恶化（24%）

我们发现，大部分团队对于DevOps的概念理解上还有偏差，实践中也受到成本和资源的限制，没有办法较好的进行DevOps实践。

在DevOps Cookbook里，DevOps模式被分成四个领域，其中：

领域三，开发嵌入到运维中——包括开发投入到整个生产问题处理链，分配开发资源用于生产问题管理，并协助退回技术债务，而且开发为IT运维提供交叉培训，增加IT运维处理问题的能力，从而降低升级问题的数量。

领域四，将IT运维嵌入至开发——包括嵌入和联络IT运维资源至开发，帮助开发创建为IT运维（部署，生产代码的管理等）使用的可重用的用户故事，定义一些可以被所有项目共用的非功能性需求。

克服这些阻碍的最佳办法是交流。你可以邀请其他团队的成员共进午餐，或者开些程序员的小玩笑来打开话题。和对方一起探讨彼此遇到的问题和挑战，很有可能在互相的交流中碰撞出一些技巧或方法，可以用来共同解决某个具体问题。创建这些开放的沟通渠道可以增强同理心，并有助于打破开发团队和运维团队之间的壁垒。

员工敬业度



员工敬业度反映员工对公司投入的智慧、感情和承诺的程度，行为表现为乐意宣传，乐意留下，全力付出，因此在这里我们将敬业度等价的看作是员工对公司的忠诚度。

员工的忠诚度会体现在他们的工作中：忠诚度高的员工参与工作的程度会更高，往往会以更高的标准要求自己，以提供更好的客户体验，这反过来会推动公司的绩效

在此次调查中，我们使用了员工净推荐值（eNPS）来了解员工在DevOps工作环境中的忠诚度。我们发现相比IT性能高的团队，在IT性能低的团队中忠诚度低的员工占比更多。

这是有一定道理的，因为忠诚度高的员工更愿意推荐他们的朋友来自己的团队工作，一个高效率的工作环境也会促进员工的参与度与敬业度。

员工敬业度

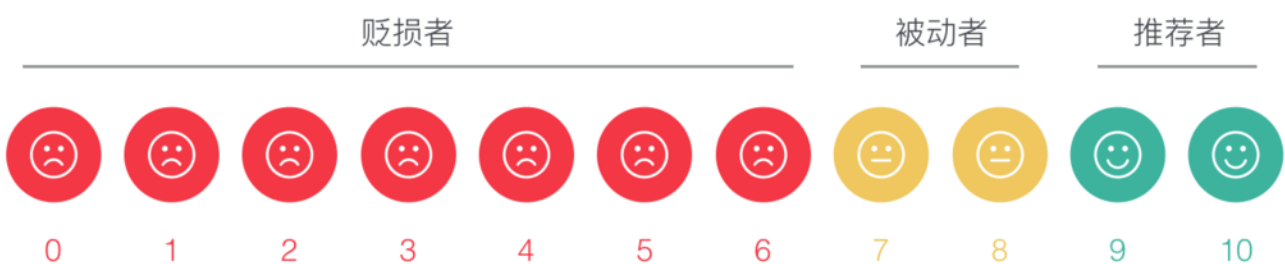
为了了解员工在DevOps环境中的忠诚度，我们使用了一个在分析忠诚度上非常流行的指标——净推荐值（Net Promoter Score，以下简称NPS）。

NPS的计算基于一个直截了当的问题：

“您愿意向您的朋友或同事推荐我们的公司/产品/服务吗”

根据愿意推荐的程度让员工在0 - 10之间来打分，然后根据得分情况来建立忠诚度的3个范畴：

- **推荐者（得分在9 - 10之间）**：推荐者是忠诚度较高的人，他们会为公司创造更大的价值。他们乐于宣传所在公司，并希望留在组织之中，会付出额外的努力并致力于那些能够促成经营成功的工作。
- **被动者（得分在7 - 8之间）**：被动者总体满意，但并不像推荐者一样具有高度热情。他们很少会称赞自己的公司，会去考虑其他竞争对手。
- **贬损者（得分在0 - 6之间）**：贬损者对公司并不满意或者没有忠诚度。公司需要付出更高的代价获得或留住他们，他们传播的消极的口碑可能会伤害公司业务。



$$\text{NPS} = \text{推荐者\%} - \text{贬损者\%}$$



NPS的一些解释

净推荐值，也叫口碑，是在许多行业中被广泛使用来测量消费者对生产者的忠诚度的指标。生产者可以是进行NPS调查的公司、雇主或其他任何实体，消费者则是进行打分的客户或员工。

员工敬业度

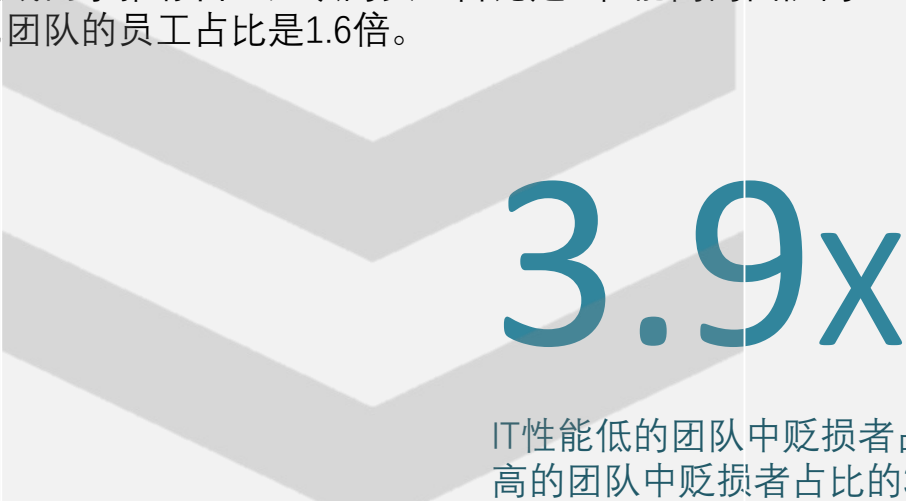
我们向被访者询问了三个问题：

您愿意向您的朋友或同事推荐我们的公司/产品/服务吗？

您愿意向您的朋友或过去的同事推荐，“我的组织不错，来我的组织工作吧”？

您愿意向您的朋友或同事推荐，“我的团队不错，来我的团队工作吧”？

我们比较了IT性能高的团队和IT性能低的团队中的贬损者的比例，发现在第一个问题中，后者是前者的3.9倍。在第二个与第三个问题中，IT性能低的团队中不愿意向朋友或同事推荐自己组织的员工占比是IT性能高的团队的1.9倍，不愿意推荐朋友去自己团队的员工占比是1.6倍。



3.9x

IT性能低的团队中贬损者占比是IT性能表现高的团队中贬损者占比的3.9倍

至于我们没有比较推荐者占比的原因，是因为在IT性能低的团队中没有推荐者，愿意推荐给朋友自己的组织或者团队的员工也寥寥无几。不过，在IT性能高的团队中愿意推荐的情况也并不多见，通过我们所收集到的数据计算得到的DevOps公司净推荐值为-14.86%——看上去情况并不乐观。

通过进一步分析，我们发现员工敬业度与以下因素显著相关：

- **团队的负责人可以有效的帮助团队：**当团队负责人对团队产生了积极影响时，例如可以做到明确的分工、合理的规划等，使员工有一个可预期的输出，对待工作也会更加投入。
- **促进开发人员与运营人员之间的沟通来达到技术共享：**当开发人员和运维人员之间的矛盾被解决，员工就可以更加专注于自己的工作而不用频繁返工。
- **经常收集客户的反馈，并将这些反馈吸纳到产品的设计中：**当员工看到他们所做的工作与对客户的积极影响之间的联系时，他们会更认同公司的目标。