5 种 DevOps 基本实践方法

如何制定和发展这些方法



目录

简介 1

根据自身情况选择适当的方法 2

发展量表 4

基本实践和 Cams 7

一切都与共享有关 8

五种基本实践详解 10

监控和警报可由运行服务的团队进行配置 12

测量内容—和测量方法 16

监控真实企业指标的重要性 18

考量标准是 DevOps 的核心 26

重用构建应用程序或服务的部署模式 28

重用用于构建应用程序或服务的测试模式 32

团队对其他团队提供的工具进行改进 36

使用配置管理工具管理配置 38

在您的组织中实施基本实践 42

但是……要根据自身情况开始行动 46

现在，与更多团队共享 48

最好的途径是能够快速显现效果的途径 50

关于作者 53





简介

DevOps 用了不到十年的时间从一个热门的新词变成了被广泛认可的，可以在快速变化、竞争激烈的商业环境中管理技术变化和交付软件的正确方法。但是，尽管世界各地的技术会议、书籍和领先的博客上提供了大量的示例，但到底如何开始采用这种方法，之后又该如何发展仍然会让大多数团队备受折磨。

Puppet 2018 年 DevOps 现状报告今年采用了新的研究方法，以便为团队寻求规范性的指导。我们的调查旨在了解企业在采用 DevOps 的过程，对数据进行分析后，我们发现，成功的企业都经历了特定的阶段。我们的研究还发现了一系列核心实践（我们称之为“基本实践”），这些实践方法在采用 DevOps 的整个过程中对成功起着至关重要的作用。在本文中，我们将为您详细介绍这些基本实践，并根据我们的发现，针对如何以对组织最有利的方式开始采用这一方法为您提供建议。

根据自身情况选择适当的方法

每个组织都应以适合自身的特有方式开始这一过程。各组织都有之前使用的技术方案、已经形成的运行方式、自身特有的企业使命和自己的特定文化。因此，DevOps 转型没有固定的方式；相反，它有着许多可能的发展路径。

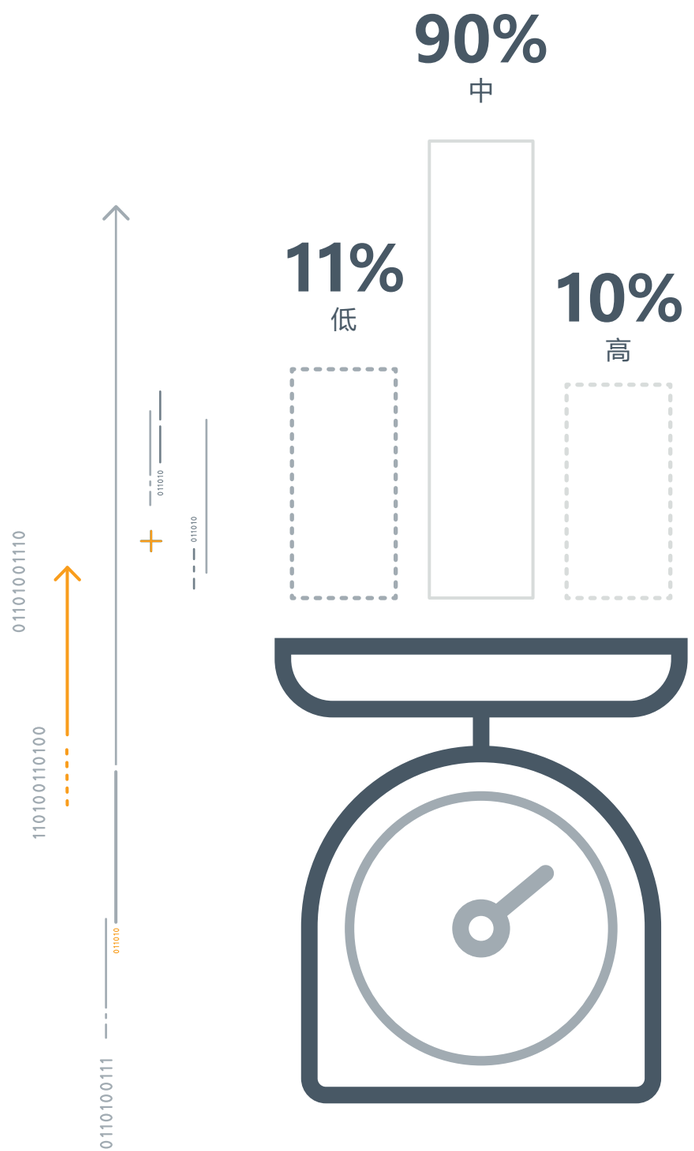
对 2018 年 DevOps 现状调查相关数据进行分析后，我们发现了成功团队所采用的基本实践。这些实践与 DevOps 的成功紧密相关，我们已经确定它们在 DevOps 发展的每个阶段都是必不可少的。换句话说，即使对于那些已经在 DevOps 采用过程中取得最好成绩，并且已经显示出了最大成功的组织，为了进入下一个阶段，在任何给定的阶段必须采用的实践仍然十分重要。

每种基本实践都可以用一句话来概括：

- 监控和警报可由运行服务的团队进行配置。
- 重用用于构建应用程序或服务的**部署模式**。
- 重用用于构建应用程序或服务的**测试模式**。
- **团队对**其他团队提供的工具进行改进。
- 使用配置管理工具**管理配置**。

当我们更加仔细地检查这些实践时，我们发现发展程度较高的组织（见发展量表）比发展程度较低的组织更有可能***总是***在整个发展过程中使用这些实践。我们从发现的结果中可以看出，对 DevOps 而言，上面列出的基本实践是不可或缺的，并且对于 DevOps 的成功也至关重要。





发展量表

在 2018 年 DevOps 现状报告中，我们开发了一个模型，来衡量每个接受调查的组织在发展量表上的位置。我们根据接受调查的组织采取每一种实践方法的频率来评分 (1 = 从不，2 = 很少，3 = 有时，4 = 大部分时间，5 = 总是) 。然后，我们将这些分数相加，得出一个综合分数。根据这个综合分数，我们将这些回答分为三类：低、中和高。所有实践方法使用频率均较高的组织得分较高，或者量表结果为“高”。那些实践采用频率较低的组织量表结果为“低”，仅采用了部分实践的组织量表结果有时可以达到“中”。参见 Puppet 2018 年 DevOps 现状报告，获得完整说明。

90% 接受调查的组织量表结果为“中”。接近 11% 的组织量表结果为“低”，只有不到 10% 的组织量表结果为“高”。

这告诉我们 DevOps 实践已经成为主流，量表得分从“中”到“高”要比从“低”到“中”难得多。由此，我们推断，如果组织集中精力进一步发展他们的 DevOps 实践，那么他们就可以获得较大的竞争优势。

我们怀疑，人们可以更加轻松地达到“中”的量表结果，因为自动化路径的定义相对明确，但量表结果要达到“高”，则需要实现 DevOps 文化和共享，这更难掌握和灌输。

基本实践和 CAMS

DevOps 基本元素的重要性不应该让那些对 DevOps 真正感兴趣的人感到惊讶。其他备受好评的构件基于这些同样的基础而构建。CAMS 模型，[1]是对 DevOps 最早的描述之一，包含这些基本的已知好的模式，从考量结果和共享的重要性到自动化的需要。DevOps 相关论述中其他常见的比喻和方法（诸如左移、团队授权、测试驱动开发等概念）还可以强化这些基本模式和实践。



1.文化、自动化、考量、共享。2010 年，Damon Edwards 和 John Willis 首次定义了 CAMS。更多关于 CAMS 和 DevOps 的讨论，请参见 Puppet 2018 年 DevOps 现状状态中的“CAMS 和 DevOps 发展模型”一章。

一切都与共享有关

通过研究我们的研究发现的基本实践，我们发现它们都依赖共享，都有利于共享。

监控和警报可由运行服务的团队进行配置。监控和警报是共享系统和应用程序运行信息，并使每个人获得相同理解的关键。无论是在一个团队和部门中，还是在多个不同的团队和部门中，这种相同的理解对于进行改进都至关重要。

重用用于构建应用程序或服务的部署模式和测试模式。在不同的应用程序或服务之间共享成功的模式通常意味着在不同的团队之间共享，建立达成共识的工作方式，为进一步的改进提供基础。

团队对其他团队提供的工具进行改进。这种形式的共享可以促进团队之间围绕工具、过程和考量标准的优先级和进一步改进计划开展更多讨论。

使用配置管理工具管理配置。开发、安全和运维领域以外的其他团队可以通过配置管理工具对系统和应用程序配置进行修改。这使得可操作性和安全性成为整个企业的共同责任。

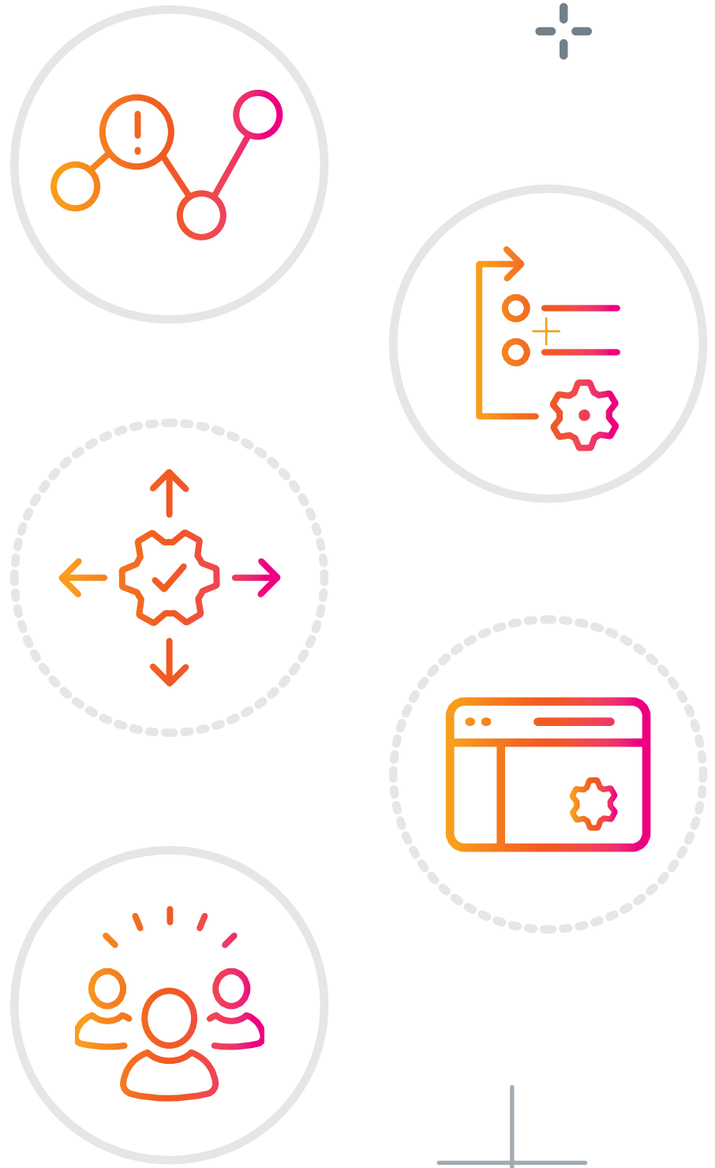
我们发现，所有的基本实践都有助于或依赖于共享，这样我们就知道，扩展 DevOps 成功的关键是采用可以促进共享的实践。

这种情况很有意义：当人们看到事情进展顺利时，他们希望可以复制成功经验，当然人们也希望分享他们的成功。假设您的团队 10 次成功地部署了一个应用程序，并且假设这种类型的部署通常会给您的团队（和其他人）带来很多麻烦。有时候，可能会有人注意到并希望了解你是怎么做到的。这就是 DevOps 实践在多个团队之间开始进行扩展的方式。



五种基本实践详解

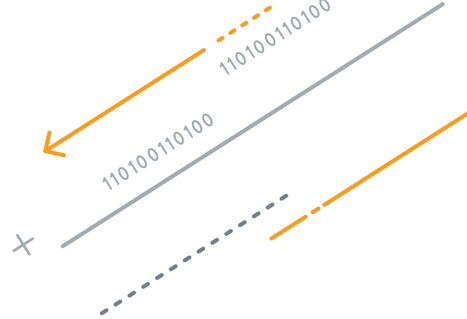
在本部分中，我们将对每一种基本实践进行详细介绍，还会对这些实践对 DevOps 的发展产生的作用进行说明。



监控和警报可由运行服务的团队进行配置。







在采用 DevOps 的过程中，我们需要注意的核心问题是授权和问责的两面性，Amazon CTO Werner Vogels 在他的论述中这样总结："自己建设，自己运行。"[2] 所以我们的研究着眼于，在生产中，有多少运行应用程序和服务的团队（由开发人员、运维人员、软件发布工程师还是其他人员组成）能够定义他们自己的监控和警报标准。

获得授权，在生产环境中运行应用程序和服务的团队可以定义什么是好的服务；如何判断运行是否正常；以及如何发现不正常的情况。这种获得授权的监控方法可以采取多种形式。例如：

- "将监控配置放置在某个位置，我们就可以利用它。"
- "登录这个网页界面来配置你的监控。"
- "在你的基础设施代码中添加一些可监控的输出。"
- "你可以使用这里的 API 将监控配置为代码。"

我们发现，组织开始注意到 DevOps 的推动作用后，47% 的发展水平较高（量表评分：高）的组织能够为生产中的应用和服务定义他们自己的监控和警报标准，而发展水平最低（量表评分：低）的组织只有 2%。评分为"高"的组织采用这种实践方法的可能性要高出 24倍！相反，评分为"低"的组织从不使用这种实践方法的可能性则高出了 23 倍之多。

2.Gray, J., Vogels, W., A Conversation with Werner Vogels, ACM Queue, https://queue.acm.org/detail.cfm?id=1142065, 2006 年 6 月，检索时间：2018 年 8 月。

在我们的调查中，我们问道，“您开始注意到 DevOps 的一些推动作用之后，这些实践方法的使用频率如何？” 右侧是实践答案的详细分布情况，“监控和警报可由运行服务的团队进行配置。”

	低	中	高
总是	2%	17%	47%
大部分时间	8%	37%	47%
有时	27%	32%	5%
很少	38%	11%	<1%
从不	23%	3%	—

授权团队定义、管理和共享他们自己的考量结果和警报支持采用 DevOps 过程的多个元素，包括：

- → 共享指标作为促进持续改进的方法
- → 营造并推行持续学习的文化
- → 跨团队协作和团队授权
- → 培养个人和团队的系统思维

正如我们前面讨论的，这些因素是强大 DevOps 文化的核心，所以接受我们调查的，发展水平较高的组织很早就采用了这一实践方法并不奇怪。



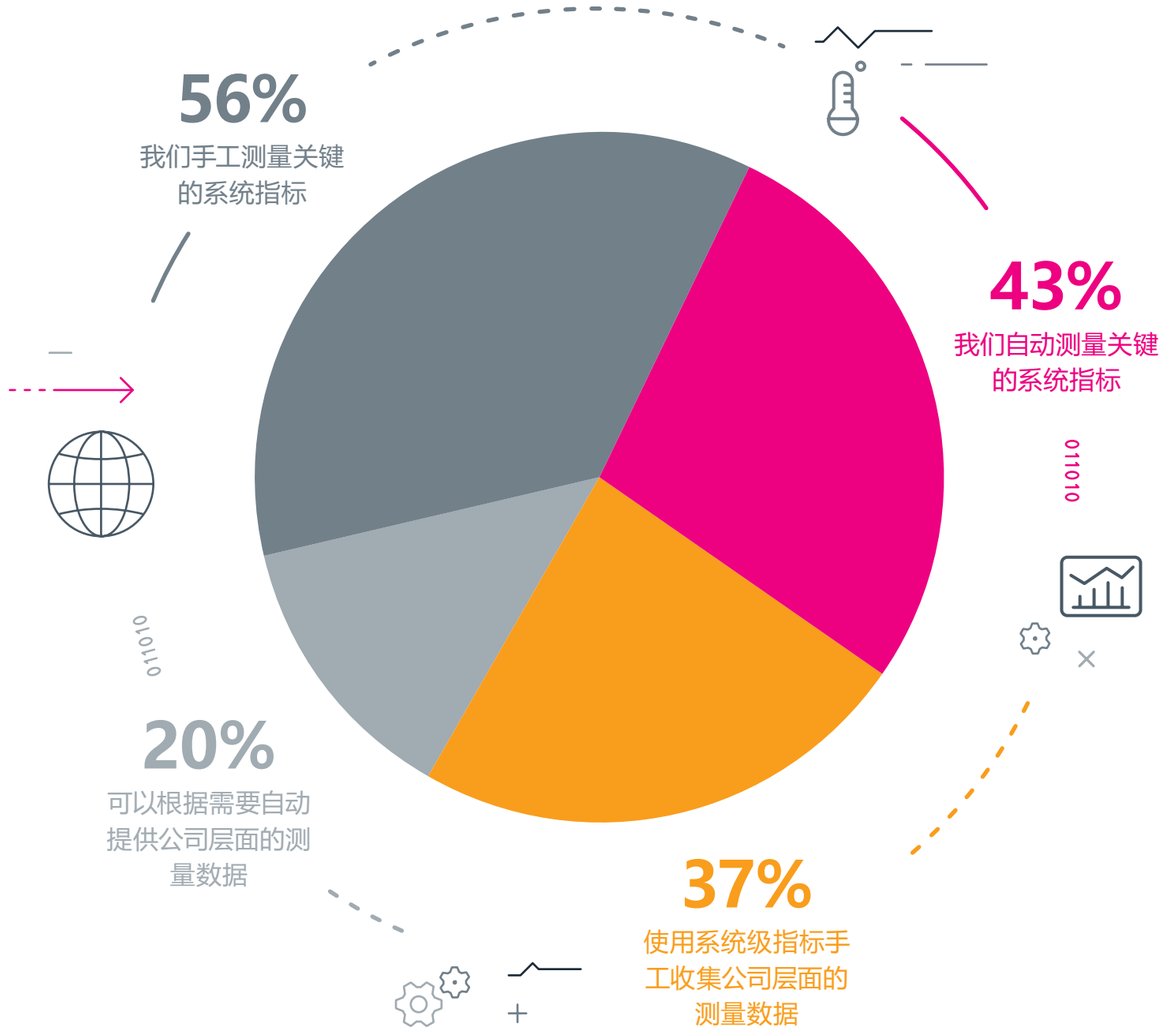


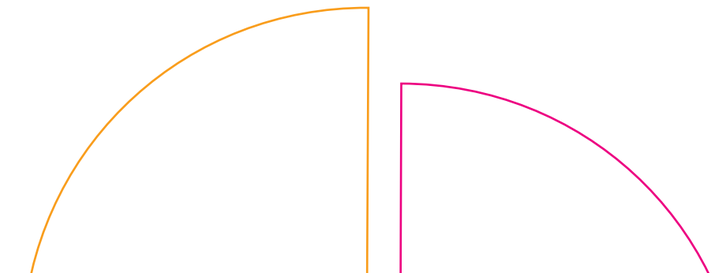
测量内容—和测量方法

我们的研究显示了监控和警报的对象:

- **关键系统指标**, 如延迟、响应时间、资源利用率等。大多数受访组织 (56%) 暴露了这些问题。
- 由**业务目标**产生的系统指标, 例如, 测量响应时间作为客户满意度的一项指标。超过三分之一的受访组织 (37%) 会对这些方面的指标进行监控。
- **按需访问**实际业务指标, 如网站停留时间、应用程序打开次数、客户注册量、营收额等。20% 的受访组织采取了这一做法。

随着组织的发展, 它们越来越有可能在其监控实践做法中采用自动化测量技术。在发展水平最高的受访组织中, 66% 的组织会自动测量系统级指标, 相比之下, DevOps 发展水平较低的组织中, 只有 32% 的组织会这样做。在手工生成系统指标方面, 发展水平较低的组织中, 59% 的组织会这样做, 而发展水平较高的组织中, 只有38% 的组织会这样做。

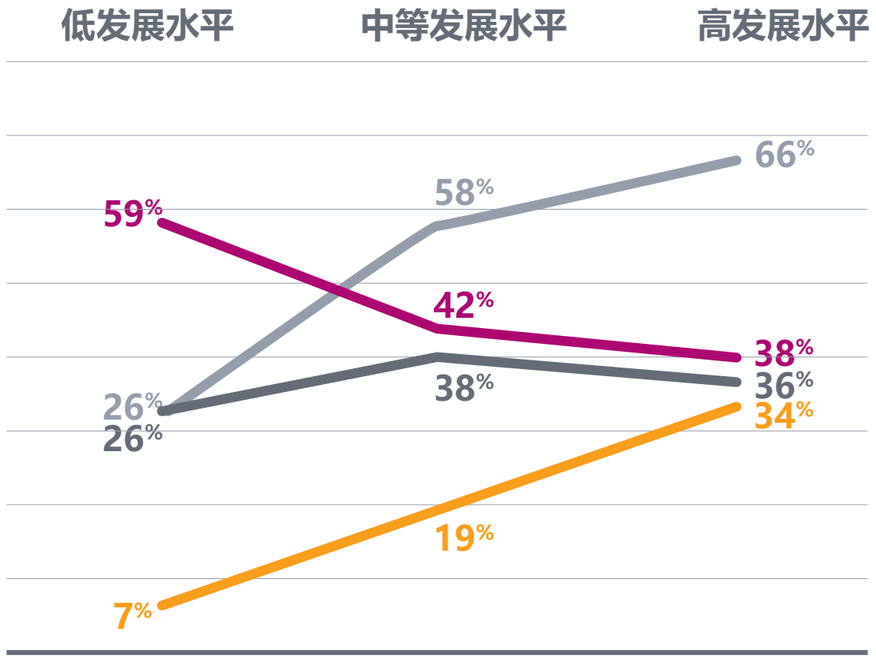




监控真实企业指标的重要性

随着组织的发展，它们监控业务影响的方法也在发展。我们的研究数据显示，DevOps 发展水平较高的组织自动收集业务指标的组织所占比例 (34%) 比发展水平较低的组织 7% 的占比高出近 5 倍。

一些 IT 团队使用业务指标作为 IT 性能的主要考量标准。例如，视频流服务网站的可靠性工程师跟踪“成功视频启动”作为服务运行状况的关键指标，而在线游戏服务开发人员则会跟踪客户注册和取消数量，加上下注金额和支付的金额，作为发行成功与否的考量标准。





业务指标可以回答下列日常服务问题:

客户是否可以购买我们提供的产品或服务?

我们服务的人数有多少人?

我们的收入是多少?

这种模式正常吗?

业务指标对下列长期成功考量标准也很重要:

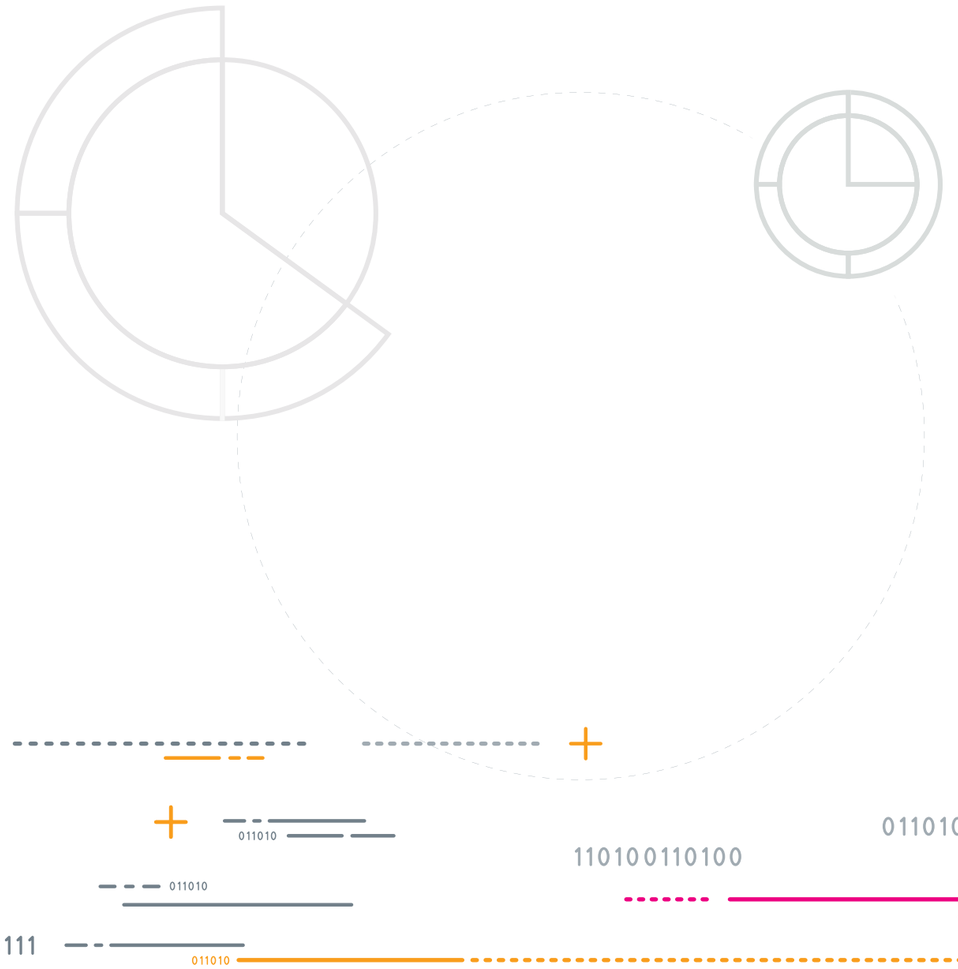
新发布的内容是否可以达到我们的预期?

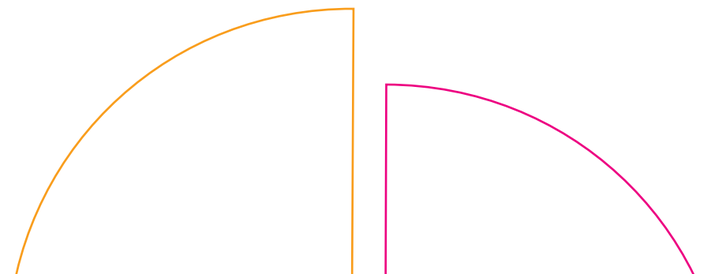
它能带来更好的业务成果吗?

与其他版本相比,它对业务的影响如何?

对于使用数据分析、机器学习和人工智能的高级组织来说,业务数据可针对每分钟收入降低和跳出率高出平时水平等可能由未检测到的网站错误导致的变化进行提醒,进而提供早期故障预警。

监控业务指标还可通过企业可以理解的方式演示当前所取得的成功,进而帮助团队为其 DevOps 计划获得更多资源。





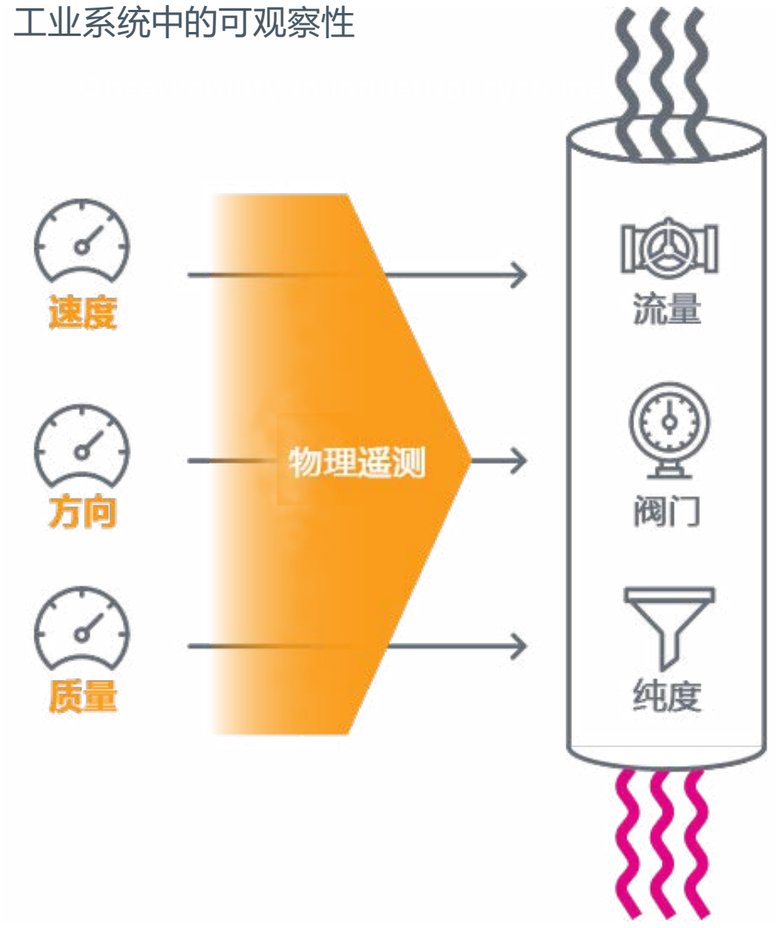
关于可观察性、遥测和语义日志

在发展水平较高的组织中， 51% 的组织总是将日志与应用程序或服务一起部署，以便在生产运营中简化监控和警报。这种做法可以提供“可观察性”：也就是系统的一个方面，我们可以通过了解其外部输出推断其内部状态。

	低	中	高
总是	3%	16%	51%
大部分时间	7%	39%	44%
有时	23%	32%	5%
很少	37%	10%	<1%
从不	30%	4%	<1%

术语“可观察性”用于工业系统，实现可观察性就像在水管中添加一个流量计（即内部测量装置）并将其连接到外部显示装置（即添加遥测装置）一样简单。这样，运维人员就可以通过观察外部输出（安装在管道外的流量计）来观察系统的内部特征（水在管道内的流动速度）。

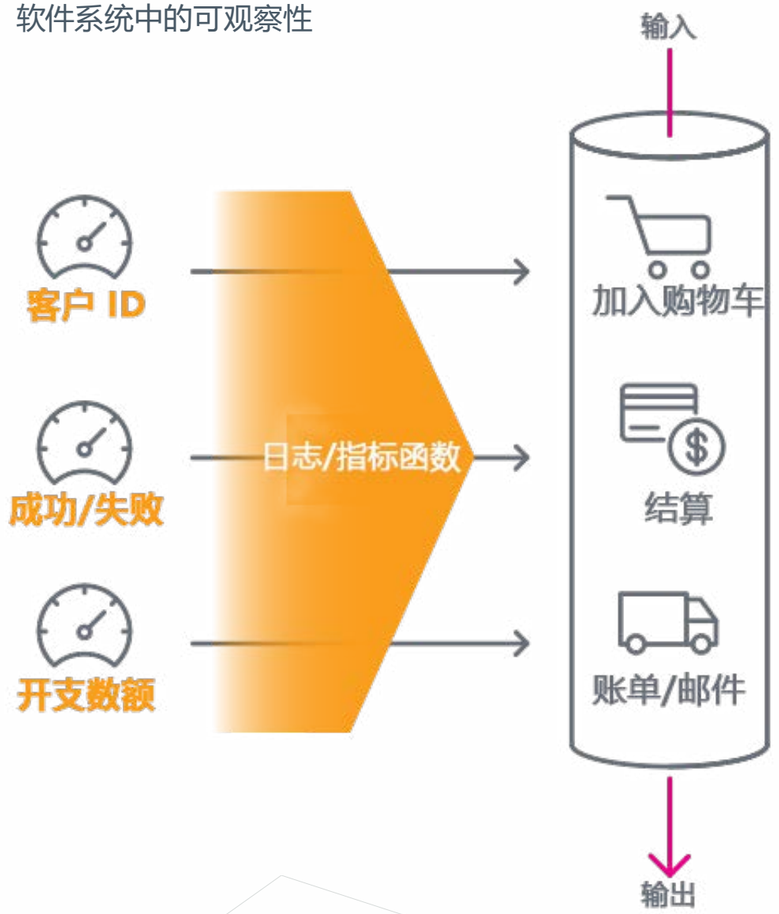
工业系统中的可观察性



在一个应用程序中，可观察性通常通过添加语义日志来描述应用程序的真实活动来实现，可以包括技术指标（如端到端响应时间）和业务指标（如每笔交易的收入）。这使得团队能够使用实际的成功考量标准（包括技术和业务上的成功）来配置他们的监控，而不是从不太可靠的代理和近似值中猜测结果。

可观察性使团队能够使用现代可组合架构，如云、微服务、容器、API 和无服务器基础设施。尽管有这些优势，但这些架构可能无法提供对基础设施性能的可见性，不具备字节码编入功能，也不提供执行综合事务的方法。在这样复杂、不透明的系统中，使用语义日志（以及其他手段）在发布时将可观察性纳入到系统中可以让 DevOps 相关人员处于主导地位，进而具备定义和配置有意义监控的能力。

软件系统中的可观察性



考量标准
是 DevOps 的核心

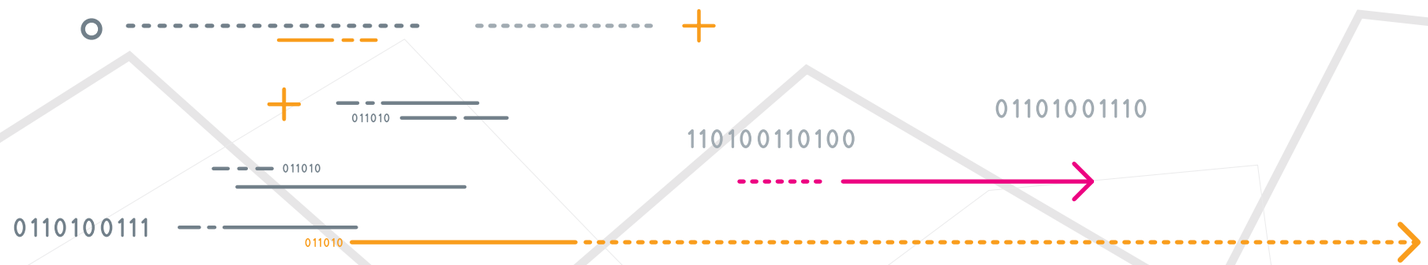
授权的监控功能不仅适用于运维或一些新组建的 DevOps 团队：它可以用于所有与技术相关的团队。所有团队均采用授权的监控强调了 DevOps 中最重要的一点：您不需要创建拥有全新超能力的新团队，而是应该向所有现有团队授权，以新的方式展开合作。

下列人员也可以同样轻松有效地采用自助服务监控和警报：

- 运行自有代码的开发人员。
- 与他们的运维合作方进行合作，以交付可操作应用程序的开发人员团队。
- 作为一个紧密合作的团队，来定义自有监控实践的 DevOps 团队。
- 作为更广泛系统的一部分，由多个团队组成，其中运维专家对开发人员交付的应用程序进行监控的复杂团队。

不管具体情况如何，自助式监控和警报都是针对开发人员和运维人员在数据仓库中工作的长期反模式的一种对策。只需放开对这些关键指标的访问，就可以实现共享文化，填充反馈回路，实现持续反馈，并推行跨团队持续学习的文化。

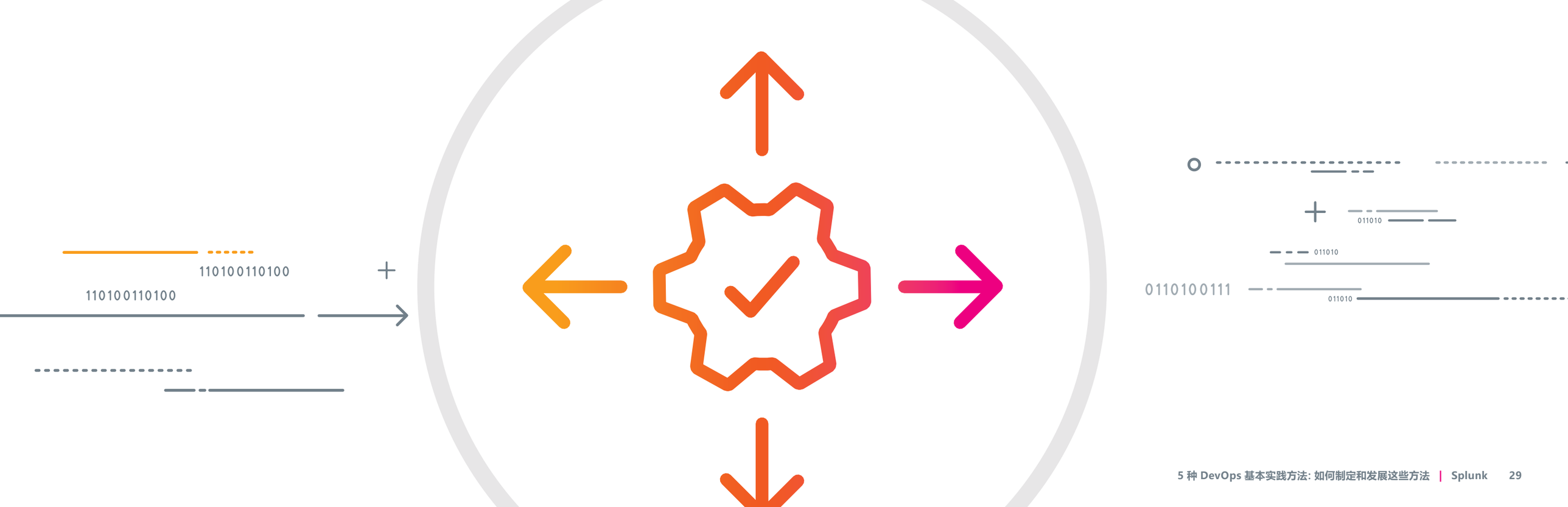
通过授权服务交付团队收集、共享并自定义监控数据来贯彻责任制和问责制是 DevOps 文化变化的基本内容，并可在过程中进一步实现更基本的转变。



重用构建应用程序或服务的部署模式

我们的研究发现了对 DevOps 成功至关重要的第二个基本能力: 重用预先定义的, 用于构建应用程序或整体端到端服务的部署例程、流程、系统和工具的能力。

当我们的调查受访组织通过他们的 DevOps 计划获得一些推动作用时, 对于发展水平较高的组织而言, 有 46% 的组织表示, 他们总是重用构建应用程序或服务的部署模式, 而对于发展水平最低的组织而言, 这一比例只有 2%。所以发展水平较高的团队总是采用这种实践做法的可能性高出 23 倍之多。



我们的调查中有这样一个问题，“您开始注意到 DevOps 的一些推动作用之后，这些实践方法的使用频率如何？” 下面是实践方法“我们重用用于构建应用程序或服务的部署模式。”的回答分解情况。

	低	中	高
总是	2%	14%	46%
大部分时间	7%	44%	47%
有时	34%	33%	6%
很少	40%	8%	1%
从不	18%	1%	—

我们的另一个问题是：“在扩展 DevOps 实践方法时，采用以下实践做法的频率如何？” 下面是实践方法“我们重用用于构建应用程序或服务的部署模式。”的回答分解情况。

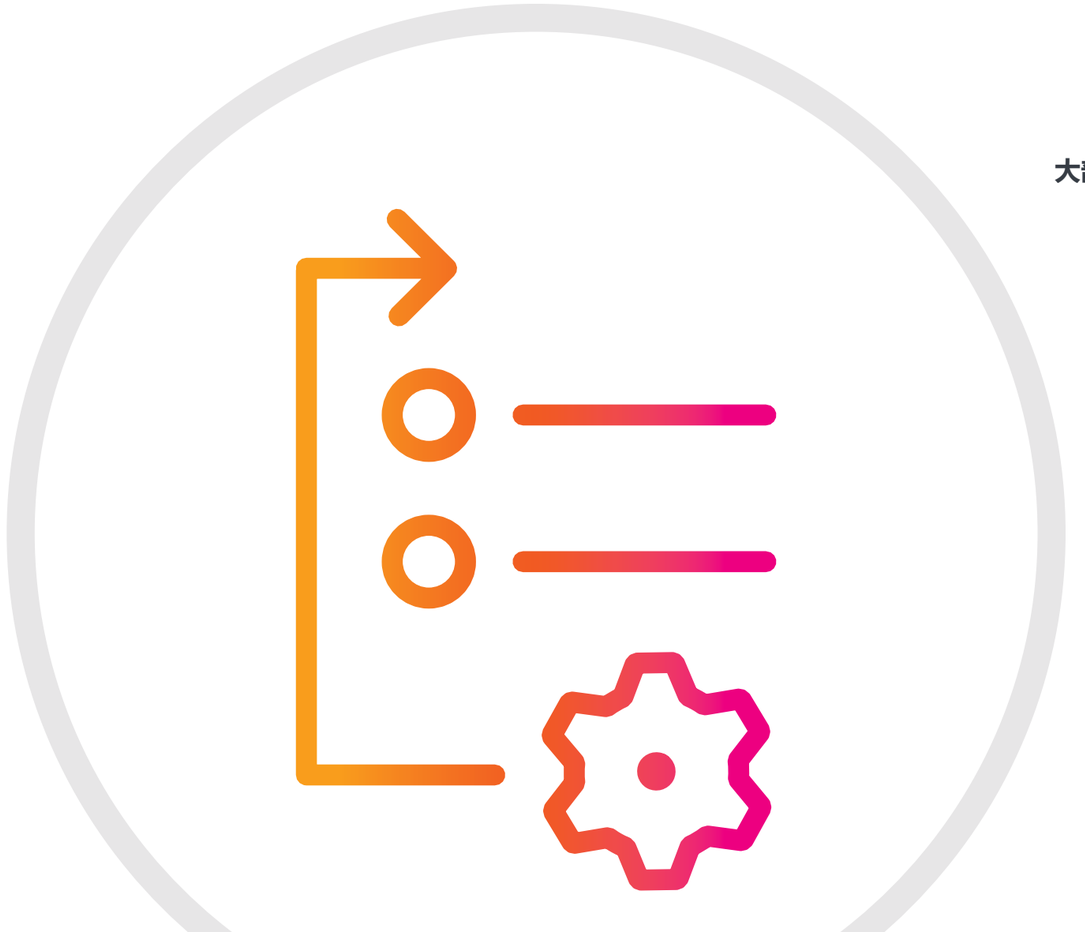
	低	中	高
总是	—	11%	52%
大部分时间	7%	45%	47%
有时	30%	35%	1%
很少	45%	8%	—
从不	18%	1%	—

我们询问了部署模式的重用问题，因为在大多数（如果不是全部的话）组织中，应用程序部署具有特殊性。应用程序部署位于开发和生产之间的边缘地带，是开发和运维任务交叉最密集的领域，也是冲突最集中的位置。因此，改进应用程序部署是 DevOps 的核心，因为它可以在开发和运维的交叉地带调解“混乱之墙”[3]。

使用可重复模式（无论是在内部创建的，还是从外部源采用的）不仅可以减轻部署的压力和混乱情况。它还可以使共享和传播在组织中更广泛地开展部署所需的知识成为可能，使更多的团队和个人能够围绕任何业务的核心竞争力展开合作。

3. http://dev2ops.org/2010/02/what-is-devops/

重用用于构建应用程序或服务的测试模式



就像共享和重用部署模式的功能一样，在 DevOps 发展中取得进展的组织还会使用可重复的测试模式。

当组织开始通过 DevOps 获得一定的推动作用时，44% 的高发展水平组织表示，他们总是使用可重复的测试模式，相比之下，只有不到 1% 的发展水平最低的组织表示他们总是使用可重复的测试模式。这使得发展水平较高的组织使用可重复测试模式的可能性高出 44 倍之多。

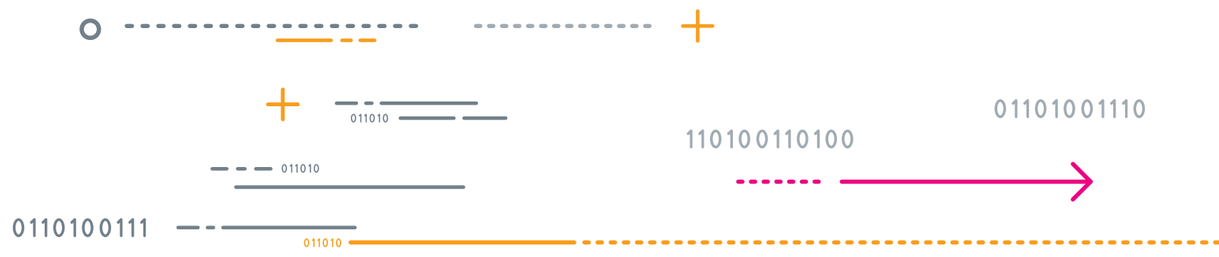
我们的调查中有这样一个问题，“您开始注意到 DevOps 的一些推动作用之后，这些实践方法的使用频率如何？” 下面是“我们会重用构建应用程序或服务的测试模式”的回答分解情况。

	低	中	高
总是	<1%	10%	44%
大部分时间	6%	38%	48%
有时	32%	39%	7%
很少	40%	11%	<1%
从不	21%	2%	—

根据组织的结构和复杂性，**自动化测试和重用测试模式**可能是一个更难解决的挑战。尽管我们确实看到，发展水平较高的组织在 DevOps 发展的早期阶段就开始采用了这种实践做法，但它可能不是你处理的第一个问题。

以下是优先考虑这个实践做法的一些注意事项：

- **如果质量团队与开发团队和运维团队完全脱节，你可能需要暂停一下，并在稍后将他们集成到 DevOps 计划中。**首先关注在您自己的团队中建立良好的测试模式。对于运维团队来说，这可能意味着在部署到生产环境之前，会有一个测试基础设施变更的流程。对于开发团队来说，这可能意味着将测试驱动的开发 (TDD) 或其他方法作为敏捷工作流程的一部分进行实施。
- **对于团队来说，最接近生产的活动 (如配置、监控、警报等) 通常具有更高的优先级，因为在那里会出现更多问题。**首先解决您的部署难题，以获得您可以用来改进测试实践做法的回顾时间。
- **测试模式的可重用性可能不如部署模式**，因为测试处理的是应用程序或服务的细节，而且还涵盖了静态/白盒或动态/黑盒环境中的许多不同流程（冒烟测试、单元测试、功能测试、合规性测试、复杂性测试）。
- **在生产中进行测试比在生产前进行测试更加困难和复杂，因为**其目标不同于生产前测试。质量团队会在生产前测试合规性、稳定性、安全性、客户满意度和其他核心目标。通过持续的交付，团队可以在生产中进行试验，以测试新的想法 (例如，通过蓝/绿或金丝雀发布)。这很重要，但又与生产中的测试不同，后者关注的是质量、功能、弹性、稳定性等等。
- **我们的结论是，采用可重用的测试过程是一种基本的实践方法**，但在很好地建立部署模式之后，倾向于在发展过程的后期推出。如果必须确定优先顺序，我们建议您先关注其他的实践方法。然而，一旦您采用了这种实践方法，确保测试模式是可共享的就很重要了。例如，您可以将可重用的测试通过编码的形式添加到自动化测试工具中，并在所有涉众中共享这些工具的访问权限以及结果报表或仪表板。



团队对其他团队提供的工具进行改进



对其他团队提供的工具进行改进的能力是一项重要的基础能力。

随着组织对其 DevOps 实践方法进行扩展，44% 的高发展水平组织表示，团队总是可以对其他团队的工具做出改进，相比之下，发展水平最低的组织中，只有不到 1% 的组织这样说。换句话说，发展水平较高的组织采用这种实践做法的可能性要高出 44 倍之多。

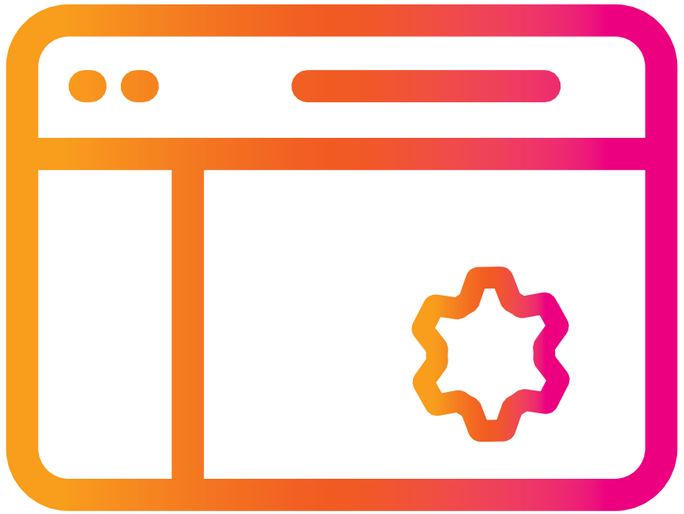
我们的另一个问题是："在扩展 DevOps 实践方法时，采用以下实践做法的频率如何？" 以下是对"团队对其他团队提供的工具进行改进"回答的分解情况。

	低	中	高
总是	<1%	11%	44%
大部分时间	4%	31%	45%
有时	26%	40%	11%
很少	46%	15%	—
从不	23%	3%	<1%

对工具的改进通常是手工的和随机的，并且在单个团队中独立完成，直到某些变更导致需要向其他团队开放。这种变化可能是整个部门的文化或组织变化；语义日志等新的跨边界数据源；或者在功能边界上跨团队进行新的协作，比如配置或发布自动化。

因为跨团队对工具进行处理的实践方法要看团队是否已经首先将自己的问题处理好，所以我们相信这种实践方法的采用可以留到后期，成功的机会是相同的。

使用配置管理工具管理配置



组织开始看到 DevOps 发展产生的一些推动作用时，使用配置管理工具管理配置的实践方法就会迅速扎根。53% 的高发展水平组织表示总是使用配置管理，相比之下，发展水平最低的组织则只有 2%。发展水平较高的组织总是使用配置管理工具的可能性高出 27 倍之多。

我们的调查中有这样一个问题，“您开始注意到 DevOps 的一些推动作用之后，这些实践方法的使用频率如何？” 以下是“使用配置管理工具管理配置”的回答分解情况。

	低	中	高
总是	2%	19%	53%
大部分时间	11%	37%	40%
有时	28%	31%	8%
很少	36%	10%	—
从不	24%	3%	—

对于长期关注 DevOps 的爱好者而言，将这种实践做法视为成功的基准并不奇怪。自动化配置管理多年来一直是 DevOps 的主要推动因素，尤其是在将基础设施视为代码的最初时期，而 DevOps 的发展在很大程度上都是围绕着自动化配置管理最初时期的创新因素而展开的。

观察早期 DevOps 推动因素就可以了解配置管理的重要性。

随着 IT 消费化推动前所未有的速度和服务需求，开发人员无法为开发和测试提供和配置关键资源（更不用说生产了）成为快速提供高质量软件，满足客户和业务需求的重大障碍。

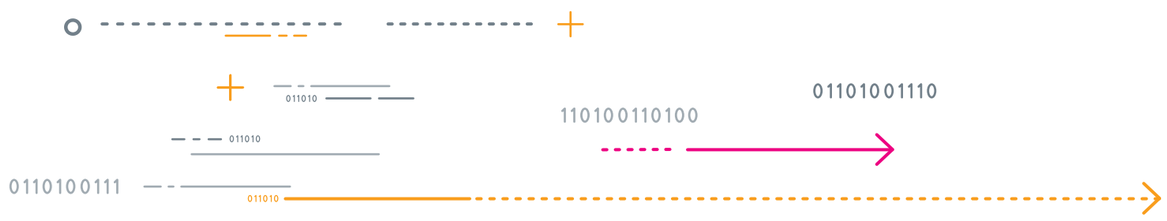
与此同时，IT 运维团队也因为无法对他们正在运行的应用程序和服务做出任何实质性的更改而感到沮丧，这些更改可以让这些应用程序和服务更稳定、更高效、更可操作。

云计算的出现让开发人员只需拿出一张信用卡就可以满足自己的需求。但这种新的能力也给运维人员带来了更多的问题和麻烦。由于开发人员采用了影子 IT，于是运维团队在生产中面临着大量特有配置、难以管理、非常脆弱且容易出错的 snowßake 算法。

当开发人员和运维人员开始共同努力，解决配置方面的这些困难时，许多 DevOps 计划很自然地会从配置和资源提供的自动化开始。这是一种可以让开发人员和运维人员皆大欢喜的解决方案，它使开发人员能够以市场速度前进，同时封装已知的理想配置并进行编码，以减少运维人员的工作负担。

随着 DevOps 的发展和扩展，以及从自动资源配置中解放出来的开发人员逐渐向连续交付的方向发展，运维人员在维持生产中的正常运行时间、性能和可用性方面就会面临更大的压力。随着组织过程的成熟，还会出现可审核性问题。因此，用于生产的自动化配置和资源提供与用于开发和测试的资源提供一样重要。通过配置管理实现可重复性可确保稳定、可靠和可审核的生产环境，还可实现后期的能力（例如，自我服务），这已经成为了 DevOps 计划的新目标。

最后，自动资源提供和配置的重要性不仅仅在于速度。它还可杜绝针对 DevOps 最早产生的一个管理上的担忧：在没有追索机制，没有审计和控制的情况下，获得授权的开发人员对生产系统的处置方式可能会过分随意。将已知理想的流程进行编码处理并通过预先设置的例程来执行它们可以减轻这些担忧，同时仍然能够使开发和运维团队以更加敏捷的方式开展工作。特别是对于较大的组织和那些医疗、金融和政府等受监管行业的组织来说，自动化可提供保持记录的更大优势，可以显示谁接触了任何系统，他们做了什么，采取行动的时间，通常还包括采取这种行动的原因。



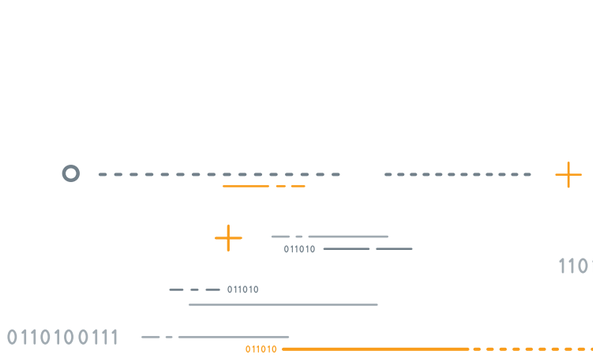
在您的组织中实施基本实践

有许多证据表明，实施这五种基本实践做法就可以获得成功，我们知道我们的读者会希望获得实施这些做法的指导。幸运的是，我们的研究确实提供了一些从哪些实践开始做起的指导。

总的来说，研究建议组织应该从基准开始，这些基准可以为未来的增长和发展奠定基础，同时还可以帮助快速实现价值。从哪些实践开始在很大程度上取决于组织在开始时所处的状况，以及它最迫切的需求是什么。但请注意：发展水平较高的组织总是会在早期就采用每一种基本实践。







大多数发展水平较高的组织看到他们的 DevOps 计划可以产生一些推动作用时，他们总是会积极采用**三种实践做法**：

- → **重用部署模式。**
- → **使用配置管理工具。**
- → **允许团队为其运营的服务配置监控和警报。**

虽然这三种实践做法是基本的能力，并且可为更高级别的 DevOps 发展奠定基准，但是采用这些做法的顺序并不重要。从这些做法开始，但不用担心先采取哪种做法。你很可能会意识到，自己的组织需要优先考虑某个特定的实践做法。

我们需要注意一个问题：根据我们的研究，我们可以很明显地看到，组织在开始显现效果前需要有更多的基本基准。因此，争取采用以上三种实践做法，然后尽快将它们扩展到其他团队中。

其余的两个基本实践 是稳定确立其他三个实践之后，我们推荐的优先实践做法：

- → **重用用于构建应用程序或服务的测试模式。**
- → **授权团队对其他团队的工具进行改进。**

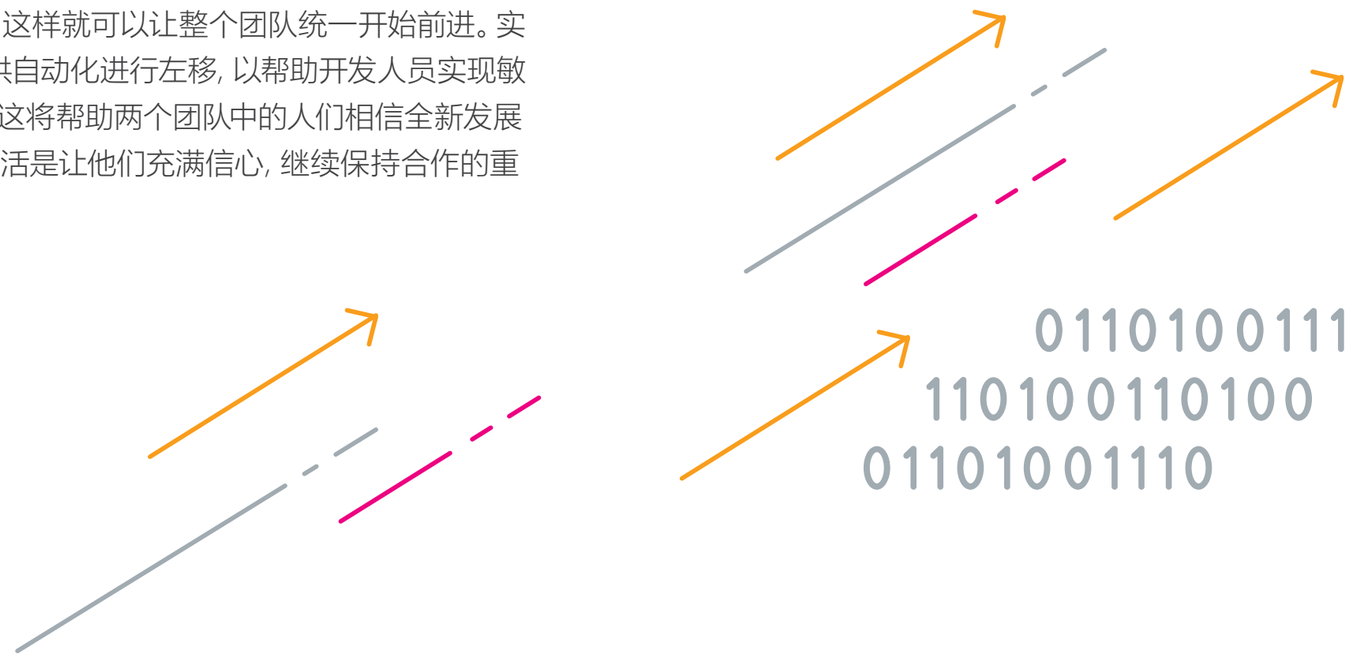


但是……**要根据自身情况开始行动**

每个组织都是不同的。要根据自身情况开始行动。确定组织中的现有条件，并选择能在组织最需要的领域尽早产生最为积极的影响的做法。由于基本实践可以提供更好的监控、更好的配置控制、更好的部署模式、更好的质量控制和协作工具，因此有很大的选择余地。

在开始时为系统、客户和业务指标建立并共享基准考量标准，这样每个人都知道从哪里开始，知道需要改进什么，并知道他们的工作对端到端系统的影响。这将帮助团队中的每个人开始接受 DevOps 概念，比如持续反馈和持续改进。

尽早共享工作实践方法和流程，这样就可以让整个团队统一开始前进。实现自动化并通过配置和资源提供自动化进行左移，以帮助开发人员实现敏捷性，运维人员实现可预测性。这将帮助两个团队中的人们相信全新发展计划的价值。改善人们的工作生活是让他们充满信心，继续保持合作的重要途径。

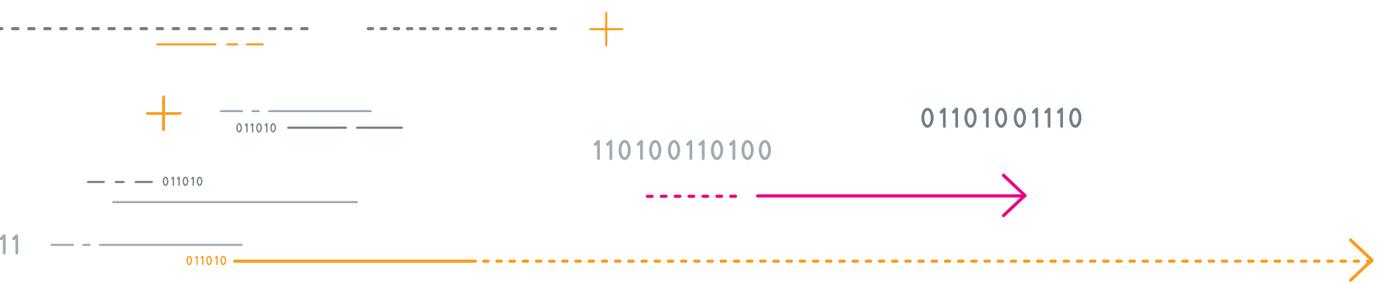


现在，**与更多团队共享**

我们的一个关键研究假设是 (就像对文化的假设一样) 自动化从团队内部开始，为自己所用。随着组织的发展，他们会越来越多地与组织内部的其他团队共享已知的良好实践。

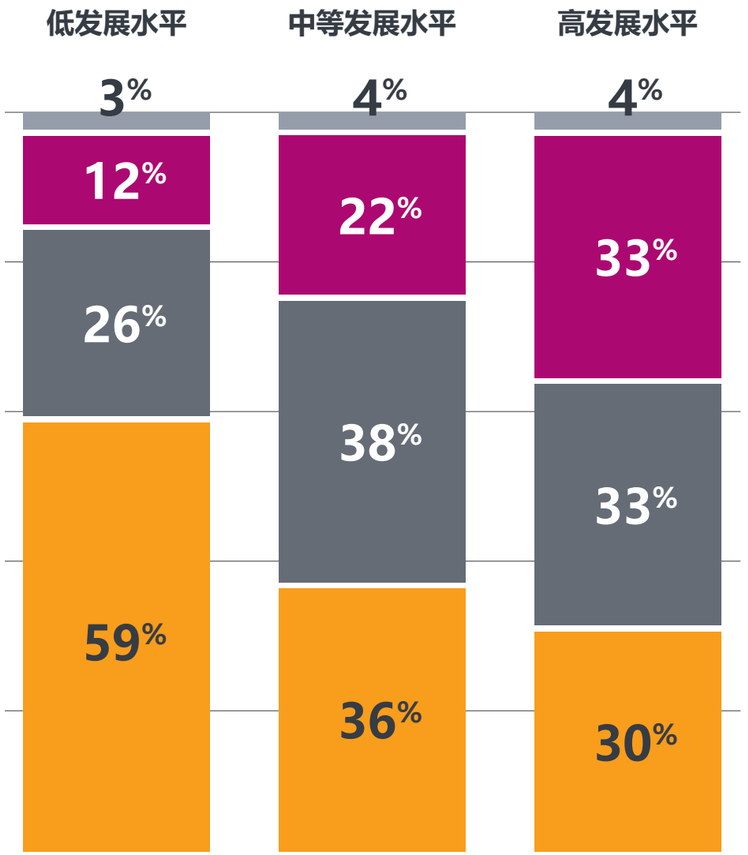
我们的研究证实了这一假设。在 DevOps 发展的最低层面上，很高比例的调查参与者 (59%) 只在他们的团队中共享最佳实践。只有 26% 的参与者会与其他团队共享，只有 12% 的参与者会在整个组织内部共享。但随着组织发展到 DevOps 成熟度的最高级别，33% 的参与者会与其他团队或整个组织共享最佳实践。只有 29% 的参与者将这种协作限制在自己的团队中。

这些发现显示了一个清晰的进程，从在 DevOps 发展的最低层次上的更紧密的孤立采用方式到发展程度很高的组织中更广泛的共享。这表明，随着 DevOps 的发展，DevOps 实践从单个团队扩展到多个团队，然后扩展到整个组织。



模式和最佳实践在……

- ……组织外共享
- ……组织内共享
- ……不同团队间共享
- ……团队内的不同人员之间共享



最好的途径是能够快速显现效果的途径

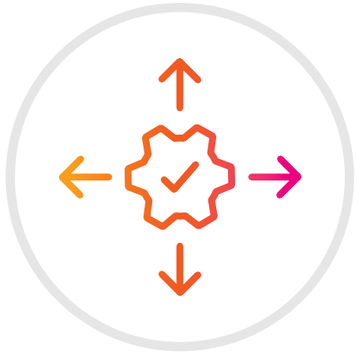
每个组织都是不同的，所以要从你关心的人最关心的事情开始。选择一种能够展现成功的工作方法，这种成功将会赢得组织领导者的信心。这将为你赢得更多时间、更多资源、更多预算，以便获得发展并早日获得成功。

我们希望了解更多关于 DevOps 的故事。请告诉我们您是如何实施这些基本实践做法的，以及您是如何衡量成功的。我们定当洗耳恭听：devops@splunk.com.

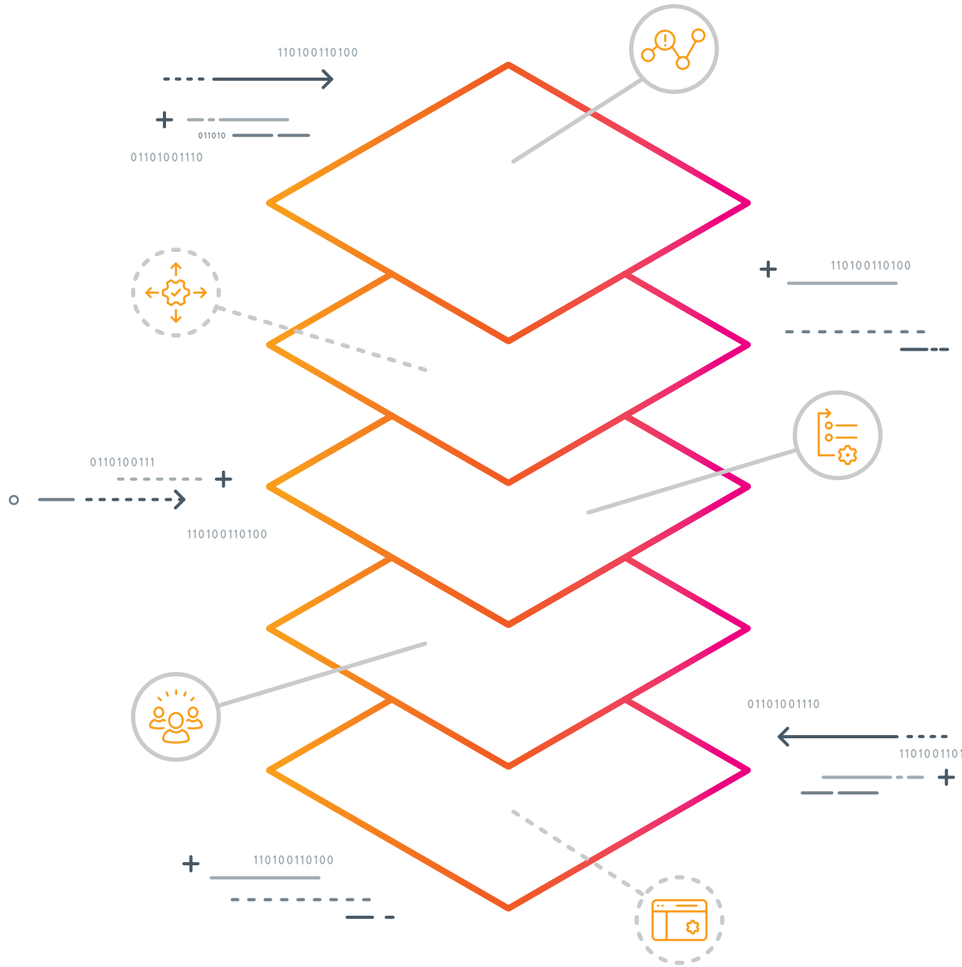














关于作者

Andi Mann•是 Splunk 的首席技术倡导者，也是一位颇有成就的数字业务领导者，在全球范围内拥有丰富的战略、技术、创新和交流经验。在超过 30 年的从业生涯中，Andi 作为领先的研究分析师和顾问，在五大洲与财富 500 强企业、供应商、各地政府展开了成功的合作。Andi 还是一位备受瞩目的商业技术评论员。曾他在《今日美国》、《纽约时报》、《福布斯》、《CIO》和《华尔街日报》等多家杂志和媒体发表文章；在 Gartner ITxpo、VMworld、CA World、Interop、Cloud Expo、DevOps 峰会上做过精彩的演讲报告；并多次参与并主持广播、电视、网络广播、播客和现场访谈节目。

splunk>

puppet

关于 Splunk

Splunk Inc. (纳斯达克: SPLK) 将计算机数据转化为答案。企业使用市场领先的 Splunk 解决方案和机器学习来解决最棘手的IT、物联网和安全挑战。立即加入数百万充满热情的用户大军，通过 Splunk 提供的方案解决"各种疑难问题": splunk.com。

关于 Puppet

Puppet 正在推动一场全球性的无约束软件变革。它所推出的革命性平台是软件交付和运维自动化的行业标准，可为我们身边的所有业务提供强大功能。超过 4 万家公司 (包括财富 100 强中 75% 以上的公司) 使用 Puppet 的开源商业解决方案来采用 DevOps 实践，实现态势感知，并信心十足地推动软件改革。Puppet 总部位于美国俄勒冈州波特兰市，是一家私人控股公司，在全球拥有 500 多名员工。详细了解 puppet.com。

了解更多信息

Splunk、Splunk>、Data-to-Everything、D2E 和 Turn Data Into Doing 是 Splunk Inc. 在美国和其他国家/地区的商标和注册商标。所有其他品牌名称、产品名称或商标均属于其各自所有者。© 2020 Splunk Inc.版权所有。

20-33892-SPLK-5 Foundational DevOps Practices-EB-113

splunk> turn data into doing