Google Cloud

DORA
DEVOPS RESEARCH & ASSESSMENT

加速：
2022 年 DevOps 现状报告

赞助方

armory

BROADCOM SOFTWARE

circleci

Deloitte.

GitLab

JET BRAINS

JFrog

Liquibase

Octopus Deploy

sysdig

目录

01

内容提要

Derek DeBellis **Claire Peters**

过去 8 年来，我们每年都会发布《加速：DevOps 现状》报告，在此过程中听取了 33,000 位专业人士的意见和建议。我们的研究重点是，了解各项功能和实践如何影响我们认为对 DevOps 极为重要的成效。

- 软件交付表现 - 软件交付表现的四个关键指标：部署频率、更改前的准备时间、更改失败率和恢复服务所需的时长。
- 运营绩效 - 第五个关键指标，可靠性。
- 组织绩效 - 贵组织在实现绩效和盈利目标方面的表现。

此外，我们还重点关注导致产生其他结果的因素，例如倦怠率和员工推荐所在团队的可能性。

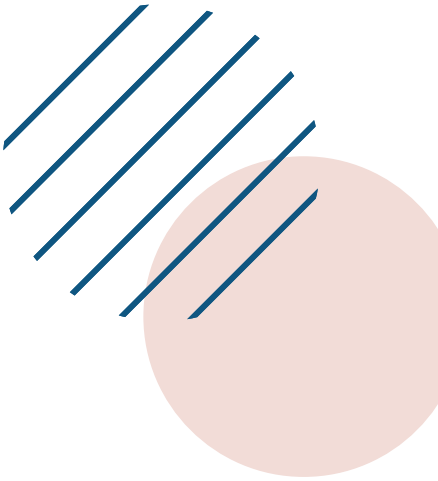
保护软件供应链安全

2021 年，我们发现保护软件供应链安全对于实现许多重要成效都至关重要。

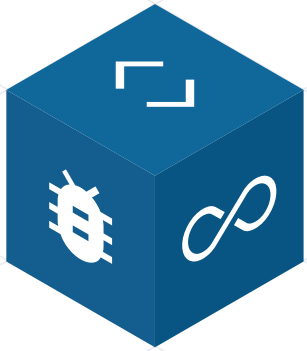
今年，我们对软件供应链安全性进行了更深入的研究，将其作为问卷调查和报告的主要主题。我们利用安全工件的供应链级别 (SLSA) 框架来探索支持软件供应链安全性开发的技术实践。此外，我们还利用美国国家标准与技术研究院的安全软件开发框架 (NIST SSDF) 来探索与保护软件供应链相关的看法、流程和非技术实践。

我们发现，文化对组织的应用开发安全性实践影响最大，而不是技术：信任度较高且不常问责的成效导向型文化更有可能拥有超出平均水平的新兴安全性实践采用率，可能性是信任度较低且经常问责的权力或规则导向型文化的 1.6 倍。我们还发现有早期证据表明，部署前安全扫描可有效找出存在安全漏洞的依赖项，进而减少生产环境中的代码安全漏洞。

采用良好的应用开发安全性实践可带来其他好处。我们发现，在专注于采用这些安全性实践的团队中，开发者较少出现倦怠；与安全性实践采用程度较高的团队相比，采用程度较低的团队中出现高度倦怠的几率达到 1.4 倍[1]。如果团队专注于采用安全性实践，其成员向他人推荐团队的可能性会显著提高。不仅如此，SLSA 相关安全性实践有助于提升组织绩效和软件交付表现，但组织需要具备强大的持续集成能力，才能使这种积极影响充分发挥出来。



[1]我们将此统计数据中的“高”的概念定义为得分（例如安全性）标准差大于或等于 1，“低”的概念定义为得分标准差小于或等于 -1。



组织绩效的驱动因素

除了上述安全性实践以外，影响组织绩效的关键变量往往可分为以下几类：

组织和团队文化

信任度较高且不常问责的文化（根据 Westrum 的定义）往往会推动实现更高的组织绩效。同样地，如果组织能够让团队感受到资金方面和领导阶层的支持，往往会实现更高的组织绩效。此外，团队稳定性以及成员对团队的正面看法（推荐团队的可能性）也有助于实现更高的组织绩效。最后，在工作安排方面灵活多变的公司也更有可能实现较高的组织绩效。

可靠性

与可靠性工程相关联的实践（例如，明确的可靠性目标、重要的可靠性指标等）以及员工认为的其可靠性期望满足程度，对能否实现较高的组织绩效有着极大的影响。

云端

我们发现，云端使用情况与组织绩效密切相关。一开始便构建云端专用软件的公司往往能实现更高的组织绩效。与仅使用本地服务器相比，使用私有云、公有云、混合云或者混合使用多种云可实现更高的组织绩效。使用多个公有云的公司更有可能获得超出平均水平的组织绩效，可能性是未使用的公司的 1.4 倍。

此外，根据我们的数据集，云端使用情况似乎还会通过其他因素影响组织绩效。以供应链安全性为例，我们发现使用公有云的组织采用 SLSA 实践的可能性也更高，原因可能是云服务提供商鼓励采用和提供许多 SLSA 实践的基础组件，例如自动构建和部署[2、3]。从广义上讲，使用云端平台可让团队沿用许多功能和实践，而这最终会转化为更高的组织绩效。”

[2]Jung、Sun Jae。“Introduction to Mediation Analysis and Examples of Its Application to Real-world Data”（中介分析简介及其在实际数据中的应用示例）。《Journal of preventive medicine and public health》（预防医学与公共卫生杂志）= Yebang Uihakhoe chi vol. 54,3（2021 年）：166-172. doi:10.3961/jpmph.21.069

[3] Carrión、Gabriel Cepeda、Christian Nitzl 和 José L. Roldán。“Mediation analyses in partial least squares structural equation modeling: Guidelines and empirical examples”（偏最小二乘结构方程建模中的中介分析：指南和实证示例）。《Partial least squares path modeling》（偏最小二乘路径建模）。Springer、Cham，2017 年。173-195。

背景至关重要

成效依赖于更广泛的团队背景。长久以来，DORA 都考虑到了这一点。我们认为，了解团队的特征（业务流程、强项、限制和目标）以及工作执行环境非常重要。例如，在某个背景下是优势的技术能力在其他背景下却可能是劣势。今年，我们的重点是以互动的形式对这些假设条件进行显式建模；今年的数据支持了其中许多假设：

- 只有在运营绩效同样出色的情况下，出色的软件交付表现才对组织绩效有所助益。如果服务无法满足用户对可靠性的期望，即使拥有快速交付能力也可能无关紧要。
- 持续集成非常成熟时，实现软件供应链安全性控制机制（例如 SLSA 框架建议的控制机制）会对软件交付表现产生积极影响。如果不具备持续集成能力，软件交付表现与安全性控制机制之间可能会存在冲突。
- 站点可靠性工程 (SRE) 实践对团队实现可靠性目标的能力的影响是非线性的。只有当团队的 SRE 成熟度达到特定级别时，SRE 实践才会对可靠性产生积极的影响。在团队的 SRE 成熟度 达到该级别之前，我们并未发现 SRE 与实现可靠性目标之间的关系。然而，当团队的 SRE 采用率提高到拐点时，使用 SRE 便会开始对可靠性产生明显的积极影响。可靠性提高后，就会连带影响组织绩效。
- 各项技术能力是相辅相成的。持续交付和版本控制可强化彼此的能力，进而带动软件交付表现提升。结合使用持续交付、松散耦合架构、版本控制和持续集成后，软件交付表现会比分别使用时的总和更出色。

鉴于交付所依赖的条件以及了解更广泛的团队背景的必要性，我们得出了与 2021 年的这项洞察类似的结论：

“为了做出有意义的改进，团队必须采用持续改进的理念。使用基准来衡量您的当前状态，根据研究调查的能力确定限制条件，并尝试改进以消除这些限制条件。实验可能成功，也可能失败，但无论哪种情况，团队都可以根据经验教训采取有意义的行动。”

事实上，我们今年发现了一个与此总体原则高度契合的现象：与未意识到需要持续改进的团队相比，意识到这一点的团队往往具有更出色的组织绩效。

简而言之，团队需要不断做出调整，尝试各种软件开发实践。

我们了解这一点，是因为从总体来看，这样做的团队具有更出色的组织绩效。这并非放之四海而皆准（适合一个组织的不一定适合另一个组织），但在大多数情况下都适用。在探索和强化适合团队的 DevOps 实践时，难免会遇到失败的情况，建议您对此做好准备。

此外，我们今年还在数据中发现了一些意料之外的现象。请继续阅读，了解具体有哪些现象。

与未意识到需要持续改进的团队相比，意识到这一点的团队往往具有更出色的组织绩效。

02

如何比较?

Derek DeBellis

您是否想知道，相较于业内其他团队，您的团队表现如何？本部分介绍了最新的 DevOps 表现基准评估方式。我们研究团队如何开发、交付和操作软件系统，然后将受访者分为多个组，这些组涵盖最常见的 DevOps 表现组合。

今年，我们添加了两种不同的聚类方法。第一种方法是基于以往惯例。这种聚类方法侧重于根据反映软件交付表现的四个指标来创建组，这些指标分别为：准备时间、部署频率、恢复服务所需的时长以及更改失败率。我们会下文中概要介绍这些指标。这种方法的目标是帮助您量化团队的当前表现，以便您将自己团队的表现与其他团队进行比较。

第二种聚类方法包含第五个指标，即可靠性，我们利用该指标来了解运营绩效。为什么要向组分析添加新的指标？因为该指标持续展现出了它的重要性。事实上，我们有证据表明，如果没有出色的运营绩效，交付表现可能有损于组织绩效。与传统的聚类方法不同，这是一种描述性做法，会尝试呈现团队在交付和运营方面的常见表现。因此，不一定能够看出哪个组的表现更好。

首先，我们来简单看一下用于了解软件交付表现和运营绩效的五个衡量指标。

软件交付表现和运营绩效

为了满足不断变化的行业的需求，组织必须快速可靠地交付和运营软件。您的团队更改软件的速度越快，您就能越快地为客户带来价值、运行实验以及获得有价值的反馈。基于八年的数据收集和研究，我们设置并验证了四个用于衡量软件交付表现的指标。2018 年，我们添加了第五个指标，以涵盖运营能力。在所有五个指标上均表现出色的团队能够实现卓越的组织绩效。我们将这五个指标称为**软件交付表现和运营 (SDO) 绩效**。请注意，这些指标侧重于系统级成效，有助于避免跟踪软件指标的常见问题，而这些问题可能会导致功能相互对立，以及以牺牲总体成效为代价进行局部优化。

关于软件交付表现和运营绩效的五个指标

关于软件交付表现的四个指标可以从吞吐量和稳定性这两个方面来考虑。我们使用**代码更改前的准备时间**（即从代码提交到在生产环境中发布的时间）和**部署频率**来衡量吞吐量，使用突发事件后**恢复服务所需的时长**和**更改失败率**来衡量稳定性。

第五个指标代表**运营绩效**，用来衡量现代运营实践。**运营绩效**根据**可靠性**来评定，即您服务的可用性和性能等方面在多大程度上达到了用户的预期。在过去，我们衡量的是可用性而不是可靠性，但由于可用性是可靠性工程的一个具体关注点，因此我们于 2021 年将可靠性纳入衡量范围，以便更广泛地反映可用性、延迟时间、性能和可伸缩性。具体而言，我们请受访者评估了他们达到或超越其可靠性目标的能力。我们发现，交付表现程度不同的团队若能一并优先考虑运营绩效，将会获得更出色的成效（例如倦怠率较低）。

历史聚类方法：聚类交付表现

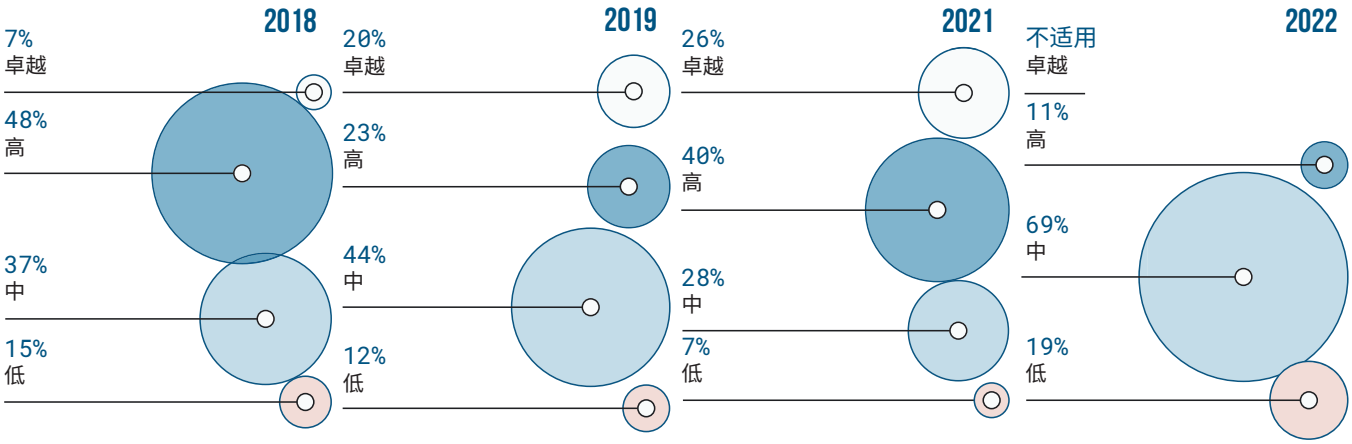
今年，在评估我们自 2018 年起使用的包含四个组的解决方案时，我们注意到，在数据中，一个组的绩效明显较高，一个组的绩效明显较低。但是，传统上我们用来划分中等绩效和高绩效的两个组差异化程度不足，无法确保适当的拆分。此外，我们用来选择合适的聚类解决方案的各种指标无一例外地表明，无论采用何种聚类技术，三个组都能以最佳方式捕获数据。下表介绍了每个组的交付表现特征。

与去年明显不同的是，今年我们没有将任何组划为卓越绩效组。今年的高绩效组是去年的高绩效组和卓越绩效组的组合。我们决定去掉卓越绩效组，因为今年绩效最高的组并不完全符合去年的卓越绩效组的特征。

软件交付表现指标	低	中	高
部署频率 对于您使用的主要应用或主要服务，贵组织将代码部署到生产环境或将其发布给最终用户的频率如何？	介于每月 1 次到每 6 个月 1 次之间	介于每周 1 次到每月 1 次之间	按需（每天多次部署）
更改前的准备时间 对于您使用的主要应用或主要服务，更改前的准备时间是多久（即从代码提交到代码在生产环境中成功运行需要多长时间）？	介于 1 个月到 6 个月之间	介于 1 周到 1 个月之间	介于 1 天到 1 周之间
恢复服务所需的时长 对于您使用的主要应用或主要服务，当发生影响用户的服务突发事件或缺陷（例如计划外服务中断或服务故障）时，通常需要多长时间才能恢复服务？	介于 1 周到 1 个月之间	介于 1 天到 1 周之间	不到 1 天
更改失败率 对于您使用的主要应用或主要服务，生产环境的更改或发布给用户的更改中有百分之多少导致了服务性能下降（例如，导致服务故障或服务中断），并且随后需要进行修复（例如，需要修补程序、回滚、向前修复、补丁程序）？	46% - 60%	16% - 30%	0% - 15%

这表明，此样本并不能代表员工认为自己取得进步的团队或组织。我们目前缺乏数据来支持的一个可能的假设是，软件开发在实践、工具和信息共享方面的创新减少了。原因或许是持续的疫情限制了团队和组织共享知识和实践的能力，减少了他们聚在一起互相学习的机会，这反过来拖慢了创新速度。我们希望更深入地探究此发现结果背后的原因。

尽管如此，与去年相比，今年低、中和高绩效组的交付表现都略有提高。今年各个组的交付表现看起来介于去年的两个组之间。2022 年高绩效组的交付表现介于 2021 年的高绩效组与卓越绩效组之间。2022 年低绩效组的交付表现介于 2021 年的低绩效组与中等绩效组之间。低绩效组的交付表现提升表明，尽管交付表现的上限降低，但下限却有所提高。



上表中的百分比细分数据显示，高绩效者所占百分比处于 4 年来的低点，而低绩效者所占百分比则急剧提高，从 2021 年的 7% 提高到今年的 19%。今年，超过三分之二的受访者划入中等绩效组。高绩效者和卓越绩效者所占百分比明显下降，这可能表明，今年的许多受访者所在的组织或团队尚未建立或正在建立许多现代化团队都有的 DevOps 文化。

我们可能过于关注 2021 年与 2022 年之间的不同，而不是强调相似之处。2021 年与 2022 年的组之间有许多共同特征，包括高绩效者与低绩效者之间的差距很大。例如，高绩效者拥有的部署数量预计[1]是低绩效者的 417 倍。

[1]请参阅“调研方法”部分，了解计算此估算值的方法

聚类交付表现和运营绩效

我们决定对上述五个指标所代表的三个类别进行聚类分析：**吞吐量**（涵盖代码更改前的准备时间和部署频率）、**稳定性**（涵盖恢复服务所需的时长和更改失败率）和**运营绩效**（可靠性）。这么做的原因是，运营绩效在我们的模型中扮演着重要的角色。对于运营绩效不佳的组织，吞吐量和稳定性对组织绩效的影响较小。我们认为，运营绩效 是评估 DevOps 表现的重要一环，如果不将其纳入考虑，将无法全面了解 DevOps 表现。

通过探索数据，我们制定了包含四个组的解决方案。以下是四个组的细分及其名称：

组	稳定性		运营绩效	吞吐量		受访者所占百分比
	恢复服务所需的时长	更改失败率	可靠性	准备时间	部署频率	
启动	介于 1 天到 1 周之间	31% - 45%	有时符合预期	介于 1 周到 1 个月之间	介于每周 1 次到每月 1 次之间	28%
流动	不到 1 小时	0% - 15%	通常符合预期	不到 1 天	按需（每天多次部署）	17%
减速	不到 1 天	0% - 15%	通常符合预期	介于 1 周到 1 个月之间	介于每周 1 次到每月 1 次之间	34%
弃用	介于 1 个月到 6 个月之间	46% - 60%	通常符合预期	介于 1 个月到 6 个月之间	介于每月 1 次到每 6 个月 1 次之间	21%

不过，如果您造访同一个组中的两个团队，可能会发现两者的情况大相径庭，并且我们提供的细分内容中可能并未涵盖您所看到的情况。我们根据与众多团队合作的经验来尝试提供上述每个组的细分内容，以易于理解的方式说明数据中的这些模式。此外，如果您在两个不同的时间点造访同一团队，可能会发现该团队位于不同的组中。造成这种情况的一个可能的原因是该团队进行了改进或有所退步；另一个可能的原因是，该团队进入了更适合其应用或服务的当前状态的部署模式。例如，在应用或服务开发的早期阶段，团队可能一直专注于探索（“启动”组），但当他们开始找到自己的市场定位时，可能会将重心转移到可靠性上（“流动”组或“减速”组）。

每个组都有独特的特征，且在受访者中占据相当大的比例。

启动组在我们的任何维度上的表现都不好不坏。该组中的团队或组织可能处于其产品、功能或服务开发的早期阶段。他们可能不太关注可靠性，因为他们专注于获取反馈，了解其产品与市场的契合度，以及更广泛意义上的探索。

流动组在所有特征上都表现良好：可靠性高、稳定性高、吞吐量高。仅 17% 的受访者处于这种流动状态。

减速组中的受访者不会过于频繁地部署，但如果他们进行部署，很有可能会成功。超过三分之一的受访者划分到该组，这使其成为我们的样本中规模最大且最具代表性的一个组。这种模式可能是正在逐步改进的团队的典型模式（但是并非仅限于这类团队），不过，团队及其客户大多对应用或产品的当前状态感到满意。

最后，**弃用**组代表的团队具有如下特点：在处理仍对他们及其客户有价值的服务或应用，但不再积极开发。

这些组在实践和技术能力方面存在着明显的不同。鉴于**流动**组拥有较为出色的软件交付表现和运营绩效，我们决定研究他们在实践和技术能力方面与其他组有何不同。我们发现，相对于其他组，**流动**组更注重以下几点：

- 松散耦合架构：团队在不依赖其他团队对其系统做出更改的情况下，能大规模更改系统设计的程度
- 提供灵活性：公司在员工工作安排方面的灵活程度如何
- 版本控制：对应用代码、系统配置、应用配置等内容所做的更改是如何管理的
- 持续集成 (CI)：将分支集成到主干中的频率如何
- 持续交付 (CD)：专注于安全、可持续且高效地将更改发布到生产环境的能力

奇怪的是，**流动**组通常不太重视文档。去年，我们发现文档实践对于交付表现和运营绩效 (SDO) 至关重要。**流动**组如何在不太重视文档的情况下获得出色的 SDO 绩效？首先，除了文档之外，还有许多途径可以实现出色的 SDO 绩效。此外，或许**流动**组会不断重构其代码来建立更加完备的自行编制文档流程，因此当我们在调查问卷中提及文档时，他们表现出较低的需求。

减速组在我们受访者中所占比例最高，通常由来自大型组织的受访者组成，采用云原生技术的程度往往比其他组低。假设有一家非常成熟的公司，该公司有一些僵化的流程，但仍然可以为最终用户提供稳定、可靠（以及有价值）的体验。此组展现了以绩效为导向的生成式文化[2]。“减速”组最有趣的特征之一是，它拥有低吞吐量和高度积极的工作文化（根据 Westrum 的定义，这是一种“生成式”工作文化）；这种组合并不常见。较为常见的情况是高吞吐量搭配高度积极的工作文化，或低吞吐量搭配积极性较低的工作文化。我们希望日后进行相关研究，以便更好地了解吞吐量与文化之间的关系。

[2] (Westrum)

此外，我们还会从以下三个方面比较这些不同的组：倦怠率、组织绩效和计划外工作量。发现的情况打破了我们的预期。**弃用**组的组织绩效要优于其他组。从该组的特征（稳定性低，吞吐量低）来看，这似乎与 DORA 之前的大部分发现结果相悖。但是，我们不想将异常情况的发生归咎于随机性（可能性很高），而是想探究一些可能的原因来解释这一发现结果。

在尝试解释这一发现结果时，需要牢记一个重要的补充性发现结果。**弃用**组以高昂的代价实现了较高的组织绩效：该组中的团队的倦怠率最高，看起来最容易出错，且担负的计划外工作量最多。同时，这些结果表明，要实现较高的组织绩效，有可靠性可能就足够了，但如果缺乏速度和稳定性，您的团队可能要付出倦怠率较高和计划外工作量较多的代价。

我们还提出了其他假设来解释为什么**弃用**组的组织绩效高于其他组，尤其是**流动**组。下面简单列出了这些假设。

- 这四个组拥有我们的数据未记录到的其他特征。例如，组织规模可以很好地反映成熟度。**流动**组中主要是小型公司，这可能表示其产品更多地处于形成阶段。
- **流动**组的受访者往往来自小型公司，这些公司可能较少受历史流程和基础架构的束缚，因此拥有更先进的 DevOps 流程。也就是说，数据表明组织的规模与组织绩效呈正相关，原因在很大程度上可能与技术无关。
- **流动**组往往不太重视 Westrum 的生成式文化中描述的原则。我们已经知道，这通常不利于提高组织绩效。
- 每个组中的组织对可靠性的预期及监控方式可能有所不同，并且他们对组织绩效目标的定义也有所不同。

- **弃用**组可能具有较高的短期组织绩效，但我们想知道其长期表现如何。倦怠是否会导致人员流失?他们能否扩展自己的流程?

- 我们在不同的级别提出问题。在团队级别提出技术能力相关问题（例如松散耦合架构）；在组织级别提出组织绩效相关问题。组织通常具有多个团队，受访者可能认为，尽管他们的组织运行状况良好，但其所在团队却并非如此。

此外，**流动**组的组织绩效得分仅次于**弃用**组，并且倦怠率和计划外工作量处于最低水平。正如**流动**和**减速**组所表现出来的那样，当具备可靠性时，DevOps 理念最有效。

我们很高兴能够继续探索新的方式来说明行业的变化。未来，我们希望继续将运营绩效作为了解这些变化的相关维度。我们还希望避免使用高度规范性和评价性组（例如卓越绩效组），并专注于简单的描述性做法，找出常见的 SDO 绩效组合。

03

如何提升?

Derek DeBellis

如何提升多种成效?

《DevOps 现状》报告旨在提供基于证据的指导，帮助您的团队全力采取 DevOps 实践和拓展相关能力，达成您所关注的成效。今年，我们将调查范围扩大到安全性和团队所期望的一系列成效。过去，我们专注于软件交付表现和运营绩效 (SDO) 以及组织绩效。我们会继续这样做，但同时也希望探索倦怠率、推荐团队的可能性、计划外工作量和出错倾向。这不仅是为了提高 SDO 和组织绩效，也是为了了解它们本身。因此，今年我们肯定会提出可能能够影响这些成效的实践和能力。

今年的研究模型经过调整，可更好地反映 DORA 的一个理论：对于 DevOps，并没有放之四海而皆准的方法。在实践中，我们发现，必须深入了解团队的背景，才能提出适当的建议。对一个团队有益的实践对另一个团队来说可能有害。例如，我们一早都有这样的假设：如果能够持续交付，技术能力（例如松散耦合架构、基于主干的开发、版本控制和持续集成）会对软件交付表现产生更明显的积极影响。今年，我们对这项假设和其他互动进行了显式建模，目标是加强我们的理解，从简单的“什么对什么有影响?”到“这些影响在什么条件下会产生、增强或减弱?”事实证明，了解所有这些制约因素是一项复杂且费力的工作，但我们很高兴能与您分享一些早期发现结果。

如需在线查看今年和去年的研究模型，请访问我们的网站。

不局限于 Four Keys 指标

DORA 指标如何提升开发和运营绩效?Liberty Mutual Insurance 的跨职能软件工程师团队利用 DORA 的“Four Keys”指标定期了解绩效。例如，Liberty Mutual 的高级 Scrum Master Jenna Dailey 分享，他们的一个团队利用 DORA 的研究发现瓶颈并改用测试驱动型开发方法，提升了总体绩效。

如需详细了解 Liberty Mutual 如何利用数据和 DORA 指标提升软件的质量和交付表现，请观看他们最近的 Tomorrow Talks 视频。

Eric Maxwell

云端

根据我们过去几年观察到的发展势头，云计算的采用速度会继续加快。事实上，采用公有云（包括多云）的组织所占的百分比从 2021 年的 56% 上升到 76%。完全未使用云端的组织（包括未使用私有云的组织）所占的百分比从去年的 21% 下降到 10.5%。采用多个公有云和混合云的组织所占的百分比分别从 21% 提高到 26%、25% 提高到 42.5%。我们还发现，采用私有云的组织所占的百分比有小幅增长，从去年的 29% 提高到 32.5%。

36%
公有云的采用率增幅

25%
混合云的采用率增幅

采用云计算可对总体组织绩效产生积极的影响。使用云端的受访者超额完成组织绩效目标的**可能性比未使用的受访者高 14%**。

正如我们在往年的报告中所述，采用云计算可对总体组织绩效产生积极的影响，本报告也继续验证了这一点。使用云端的受访者超额完成组织绩效目标的**可能性比未使用的受访者高 14%**。我们的研究表明，云计算可让团队在软件供应链安全性和可靠性等方面拥有出色表现，进而提高组织绩效。

令人惊讶的是，所有类型的云（公有云、私有云、混合云和多云）的采用都与更改失败率呈负相关，也就是说，采用后，更改失败率有所提高。这点有待进一步调查。我们将在未来的研究中展开进一步调查，而不是推测其原因。不过，除了少数例外情况，云原生应用（最初为云端设计和架构的应用）的采用对我们调查的各项内容都有积极影响。

使用任何云计算平台，无论公有还是私有，都有利于提升文化和工作环境成效（例如生成式文化、较低的倦怠率、更高的稳定性以及更高的员工满意度）。就这些文化成效而言，云端用户的得分要高出 16%。

云端采用速度继续加快（年同比）

	2022	变化幅度百分比（与 2021 年相比）
混合云	42.47%	25%
公有云和/或多云	76.08%	36%
私有云	32.55%	12%
未采用云端	10.55%	-50%

使用混合云和多云（以及私有云）可能会对软件交付表现指标（MTTR、准备时间和部署频率）产生**负面**影响，**除非**受访者具有较高的**可靠性**。

混合云和多云有助于提升组织绩效

我们继续观察到，使用混合云和多个公有云明显可为组织带来积极影响。使用多云的从业者的组织绩效是未使用云端的用户的 1.4 倍。不过，使用混合云和多云（以及私有云）可能会对多个软件交付表现指标（MTTR、准备时间和部署频率）产生负面影响，除非受访者同时具有较高的可靠性。这一发现结果进一步说明了强大的 SRE 实践的重要性，以及可靠性在软件交付中所发挥的作用。

2021 年，我们请受访者分享他们采用多个公有云的主要原因；而在 2022 年，我们请参与者分享他们通过采用多个云服务提供商获得的所有好处。提到次数最多的好处是“可用性”，这与业界对可靠性的关注和重视不谋而合 - 如果服务不可用，那么可靠性就无从谈起。超过 50% 的从业者提到了利用不同云服务提供商的独特优势这一好处。

通过采用多个云服务提供商获得的好处

可用性	62.61%
利用各个提供商的独特优势	51.59%
与多个提供商建立信任关系	47.54%
灾难恢复	43.48%
法律法规遵从	37.97%
协商策略或采购要求	19.13%
其他	4.06%

>50%

的受访者提到使用多个云服务提供商。

在提高组织绩效的长长的因果链中，使用云计算的五大特征是重要的开端。

云计算的五大特征

按照我们之前的研究方法，我们不仅要了解参与者是否使用云计算技术，还要了解他们如何使用云计算技术。为此，我们基于美国国家标准与技术研究院 (NIST) 定义的云计算的五大基本特征来提出问题。

按需自助服务 - 消费者可以根据需要自动预配计算资源，无需提供商方面的任何人工交互。

广泛的网络访问 - 功能广泛可用，消费者可以通过手机、平板电脑、笔记本电脑和工作站等多个客户端进行访问。

资源池 - 提供商资源汇集在一个多租户模型中，其中物理资源和虚拟资源可按需动态分配和重新分配。客户通常无法直接控制所提供的资源的确切位置，但可以在更高的抽象级别（例如国家/地区、州/省/自治区/直辖市或数据中心）指定位置。

具有快速调整的弹性 - 您可以灵活地预配和发布功能，以快速按需扩容或缩容。可供预配的消费者能力似乎是无限的，并且可以随时以任意数量配置。

衡量服务用量 - 云系统通过在与服务类型（例如存储、处理、带宽和有效用户帐号）相称的抽象级别使用计量供给功能，自动控制和优化资源的使用。您可以监控、控制和报告资源使用情况，以确保信息透明。

此报告验证了过去三年的 DORA 研究，得出了如下结论：如果组织具有这五个特征，可能会获得出色的软件交付表现和运营绩效。此外，我们还发现，这些特征有助于建立会对组织产生积极影响的动态流程，进而提高组织绩效。在提高组织绩效的漫长旅程中，展现云计算的五大特征是第一步。

2022 年，我们发现越来越多的团队开始利用云计算的优势。这是云计算五大特征采用率连续增长的第 4 年。“资源池”的增幅最大，达到 14%；“具有快速调整的弹性”是去年采用率第二高的特征，其增幅最小，为 5%。

NIST	2021	2022	变动百分比
广泛的网络访问	74%	80%	8
具有快速调整的弹性	77%	81%	5
按需自助服务	73%	78%	7
衡量服务用量	78%	83%	7
资源池	73%	83%	14

资源池采用率提高了

14%

Dave Stanke

SRE 和 DevOps

成功的技术团队可为其组织带来诸多贡献，而不仅仅是交付代码，甚至也不仅仅是交付优质代码。他们还应该确保所提供的服务始终具有较高的可用性和出色的性能，且持续符合用户预期。可靠性是一种从多个方面衡量团队的承诺履行情况的指标，今年我们继续探索可靠性这一因素对软件交付表现和运营绩效的影响。

站点可靠性工程 (SRE) 是一种极具影响力的运营方法，最初由 Google 提出，目前在许多组织中得到采用。SRE 优先考虑经验学习、跨职能协作、广泛依赖自动化，以及使用服务等级目标 (SLO) 等衡量技术。其他现代化运营实践采用了类似方法，只是应用的命名惯例不同。因此，为了尽可能客观地评估这些实践的采用程度，我们特别注意调查问卷中的用字遣词，确保受访者看到的是中性的描述性语言。此外，我们还会收集有关可靠性工程成效的数据：团队能够在多大程度上实现其可靠性目标。输入和输出（SRE 实践和可靠性成效）会与其他 DevOps 能力一同反映在我们的预测模型中。

可靠性非常重要

参与我们的问卷调查的团队普遍采用了 SRE：大多数受访者采用了我们问到的一种或多种实践。从众多团队的回复中获得的数据显示，可靠性、软件交付和成效之间存在微妙关系：如果可靠性较差，软件交付表现无法推动组织取得出色成效。但如果可靠性较佳，软件交付表现会对业务成效产生积极影响。

> 如果缺乏可靠性，软件交付表现无法推动组织取得出色成效。

投资于 SRE 可提高可靠性，但前提是采用率要达到一定阈值。

这种现象符合使用 SRE“错误预算”框架时的情形：如果服务不可靠，将代码更快地推送到脆弱的环境中并不会使用户受益。

正如站点可靠性工程师一直以来所声称的那样，对于任何产品来说，可靠性都是最重要的“特性”。我们的研究支持这一观察结果，即遵守对用户的承诺是提升软件交付表现进而使组织受益的必要条件。

今年，我们研究了技术团队的 J 曲线模式：如果团队采用的可靠性工程实践较少，表明他们处于 SRE 采用历程的早期阶段，无法因此获得较为出色的可靠性成效。不过，当团队的 SRE 采用率提高到拐点时，使用 SRE 便会开始对可靠性产生明显的积极影响，而这反过来又有助于提高组织绩效。

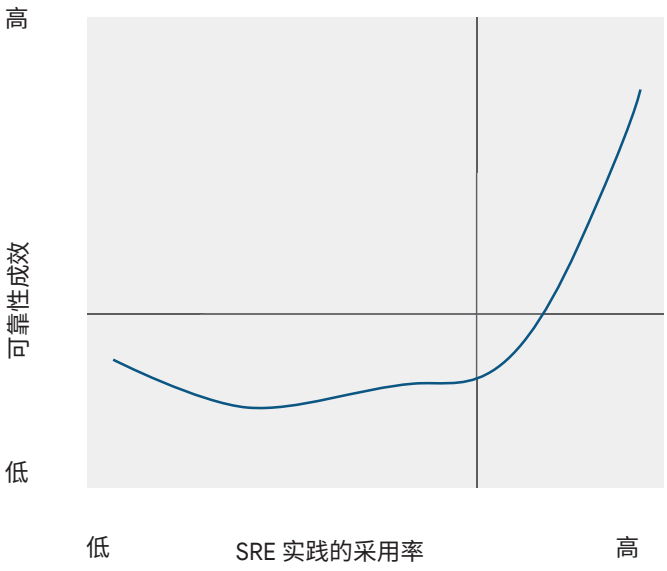
了解 J 曲线

在提高可靠性的过程中，您会遇到哪些挑战？在 O’Reilly 出版的《Enterprise Roadmap to SRE》（企业的 SRE 路线图）[1] 中，DORA 问卷调查贡献者 James Brookbank 和 Steve McGhee 回顾了他们帮助成熟的组织采用 SRE 的经历，并建议“了解变更的 J 曲线”。根据先前的《2018 年 DevOps 现状报告》所述，“J 曲线”是指这样一种现象：组织在转型初期取得成功，随后却进入回报减少 甚至倒退的阶段。但是，一旦组织努力克服这些挑战，通常能够延续成功并取得更高成就。

[1] https://sre.google/resources/practices-and-processes/enterprise-roadmap-to-sre/

可靠的团队提供可靠的服务：生成式团队文化有助于提高可靠性。

处于 SRE 实践采用历程早期阶段的团队应做好随时迎接挑战的准备。这可能会是一段漫长的旅程，因为团队需要根据新的指导原则全面调整文化、流程和工具。不过可以确定的是，只要持续投入时间和资源，就有可能取得成功。



只要团队在完成采用 SRE 的初始步骤后坚持不懈，就能不断提升可靠性成效。

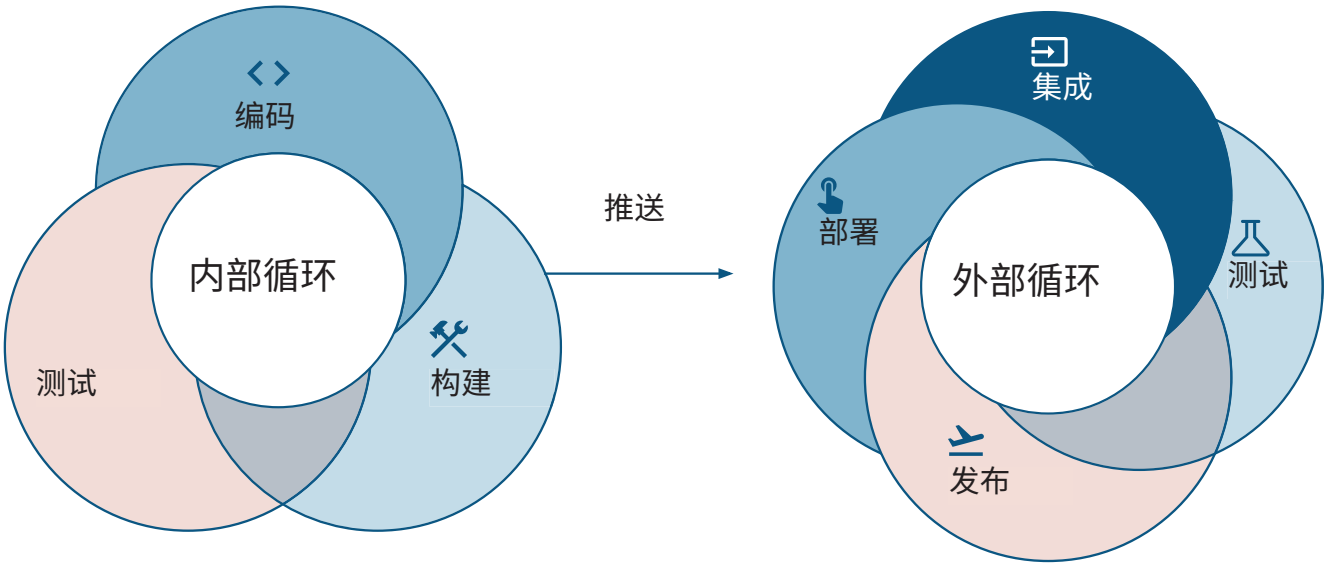
投资于人员、流程和工具

可靠性需要人为付出努力才能实现，SRE 方法从许多方面证明了这一点。SRE 的核心原则之一是，衡量可靠性的真正标准是用户感知，而不是内部监控数据。因此，可靠性由积极的团队动力推动也就不足为奇了。我们发现，拥有“生成式”文化并且成员彼此信任和协作的团队更有可能采用 SRE 实践，取得良好的可靠性成效。此外，稳定的团队意味着人员流动率低，也有助于提高面向用户的服务的可靠性。就像 DevOps 是一个整体一样，如果有流程和工具加持，人为努力将取得事半功倍的效果，进而促进可靠性工程工作。使用云计算和持续集成等实践都有利于提升可靠性成效。

Eric Maxwell

技术 DevOps 能力

今年，我们研究了各种技术能力，以了解不同的技术实践所带来的成效。我们考虑了软件开发的两个主要阶段："内部循环"，包括编码、测试和推送到版本控制等开发者任务；"外部循环"，包括代码合并、自动代码审核、测试作业、部署和发布等活动。



实现可靠性目标的高绩效团队采用 CI 的可能性较高，**是其他团队的 1.4 倍。**

我们的研究显示，在内部循环和外部循环开发方面表现出色的公司能够更快地交付代码，并且具有更高的可靠性。最能带来高绩效的能力包括版本控制、持续集成、持续交付和松散耦合架构。

实现可靠性目标的高绩效团队：

33% 使用版本控制的可能性比其他团队高 33%

39% 采用持续集成的可能性比其他团队高 39%

46% 采用持续交付的可能性比其他团队高 46%

40% 具有基于松散耦合架构的系统的可能性比其他团队高 40%

事实上，上述所有能力的采用率高于平均水平的受访者拥有更高的组织绩效，其组织绩效是未使用这些技术能力的受访者的 **3.8 倍**。

持续集成

持续集成，通常称为“CI”，是外部循环开发流程的一部分，会自动构建工件，并针对每一次代码提交运行一系列自动测试，以评估代码是否可供部署。此流程会自动为开发者提供快速反馈，让他们更有信心地进行操作。CI 是将代码从开发者的工作站推送到生产环境的关键一环。与往年一样，CI 被证明可提升交付表现。**实现可靠性目标的高绩效团队采用 CI 的可能性较高，是其他团队的 1.4 倍。**

今年，我们更深入地探究了外部循环开发流程的另一个部分：持续交付，并会在后面的章节中进行介绍。在此之前，我们先来看一下持续集成的一个补充内容：主干开发。

主干开发

主干开发是指将代码持续合并到主干中并避免使用生命周期较长的功能分支的实践。该实践被认为是对持续集成的补充，并且多年来不断被证明可以加快软件交付速度。

鉴于今年的受访者受众特征在工作年限上的变化，我们可以看出，在采用主干开发实践时，经验至关重要。去年，工作年限不少于 16 年的受访者所占百分比为 40%，而今年仅为 13%。随着不断进行“交付依赖什么”主题验证，我们发现，总体而言，经验较少的受访者使用主干开发时较少取得积极成效，并且会遇到以下情况：

▼ 总体软件交付表现**下降**
▲ 计划外工作量**增加**
▲ 出错倾向**增加**
▲ 更改失败率**提高**

工作年限不少于 16 年的受访者如果使用主干开发，可发挥这项实践的优势，并获得以下成效：

▲ 总体软件交付表现**提升**
▼ 计划外工作量**减少**
▼ 出错倾向**减少**
▼ 更改失败率**下降**

出现这种现象的原因可能是，必须采用额外的实践才能成功实施主干开发。如果团队未针对绝不在主干有恙时离开这一点制定规则并严格执行，或未使用封闭代码分支并自动回滚会破坏主干的代码，那么在尝试进行主干开发时，必然会遇到难题。

但是，主干开发确实会对总体组织绩效产生积极的影响。

Frank Xu

持续交付

持续交付 (CD) 是一种软件开发实践，具有如下优势：

1. 让团队能够随时将软件部署到生产环境或发布给最终用户。
2. 可确保软件在整个生命周期内处于可部署状态，包括在开发新功能时
3. 会建立一个快速反馈环，让团队能够检查系统的质量和可部署性，优先解决阻碍部署的问题。

请注意，持续交付不一定代表持续部署，也就是自动部署每个软件版本的实践。持续交付只要求可随时部署软件版本。

去年，我们研究了哪些技术 DevOps 能力会对团队采用 CD 实践的可能性产生影响，并发现松散耦合架构和持续测试/集成等因素产生的影响最大。今年，除了研究推动 CD 采用的因素外，我们还分析并确定了 CD 本身及该实践与其他 DevOps 能力的互动对开发成效的影响。

CD 助力提升软件交付表现

与往年的发现结果类似，无论是单独使用 CD，还是与其他 DevOps 能力结合使用，都有助于提升软件交付表现。CD 得分较高的团队更有可能更为频繁地向生产环境部署代码，更有可能拥有更短的更改前准备时间和服务恢复时间。

结合使用版本控制和持续交付的团队拥有出色软件交付表现的可能性是仅使用持续交付的团队的 2.5 倍。

此外，如果受访者所在的团队同时采用版本控制实践，则他们拥有出色软件交付表现的可能性会提高到 2.5 倍。

CD 可能会增加计划外工作量

数据显示，持续交付会使开发者将更多时间花费在返工或计划外的工作上。我们针对这一发现提出了一种假设：当反馈环较为紧凑时，开发者更有可能以迭代方式构建应用。因此，他们可能会认为对系统的同一部分做出的反复更改属于计划外的工作。这项工作可能既是计划外的，又是根据对之前部署的反馈进行的。

技术实践和 CD

我们研究一再表明，许多技术能力都支持 CD。今年，我们探索了将其中的部分能力与 CD 搭配使用会发生什么。我们发现，如果将主干开发或松散耦合架构与 CD 搭配使用，可能会对团队绩效产生负面影响。例如，有证据表明，同时采用松散耦合架构和 CD 的团队的出错倾向（即服务出现产品服务中断、安全漏洞和性能显著下降问题）更有可能高于平均水平，可能性比仅采用 CD 的团队高 43%。这些影响有待进一步调查，表明正在改进的团队可能会遇到一些阻碍。这些阻碍可能与转型的 J 曲线有关，即团队在初期顺利提升成效，但随后遇到重重困难。团队必须致力于改进，才能充分发挥潜力。在改进 CD 等任何能力时，请务必留意对团队和总体绩效的影响。

David Farley

松散耦合架构

松散耦合系统对团队和组织的效率而言非常重要。这不仅仅适用于基于云端或微服务的系统，也与组织做出更改的能力有关。组织安全、自信地更改其软件的轻松程度可以反映出软件质量的高低。

采用松散耦合架构的团队可以：

- 在不依赖其他团队对其系统做出更改的情况下，大规模更改系统设计
- 通过独立的按需测试，更快地获取反馈，减少协调费用
- 几乎无需停机即可部署代码

在今年的报告中，我们请受访者分享了他们构建的软件是否基于松散耦合架构。结果非常有趣，显示松散耦合架构的采用与团队在多个维度的表现大多存在正相关性。

松散耦合架构的优势

专注于利用松散耦合架构来构建软件的团队更易于获得出色的稳定性、可靠性和吞吐量成效。这类团队也更有可能向好友或同事推荐他们的工作场所。

部署在云端、采用微服务架构方法来管理数百项服务的团队通常使用松散耦合架构来构建软件。但是，松散耦合不仅仅是一种衡量系统中服务数量的简单方法。松散耦合架构中的组件可以单独进行部署。借助这种独立性，团队可以开发、测试和部署他们的服务，而不必花费精力与其他团队进行协调。

在实际运用中，松散耦合并不局限于一种架构形式；从根本上说，它支持在系统的某一部分中做出更改，而不影响其他部分。这让组织能够对工作进行划分，以便各个团队独立推进工作，而无需与其他团队协调。

根据我们的经验，如果团队在部署软件之前，需要进行与其他服务的深度集成测试，以提升对软件的信心，则表示他们尚未实现松散耦合；这类团队要实现松散耦合，需要改进系统之间的接口和隔离。改进接口和隔离的一种有效方式是，提高服务和组件的“可测试性”。如果您的设计允许您单独测试服务，那么根据定义，其接口属于松散耦合。

专注于利用松散耦合架构来构建软件的团队更易于获得出色的稳定性、可靠性和吞吐量成效。

我们还发现，采用松散耦合架构且具有凝聚力的稳定团队更有可能采用鼓励和支持持续改进的软件开发实践。以 SRE 实践为例，设置可靠性目标以确定工作的优先级，或定期进行审核以根据证据修改可靠性目标，均支持松散耦合架构。

此外，借助松散耦合架构，组织还可以更轻松地添加员工，因为独立的团队无需与其他团队协调，可以更自由地自主扩大团队的规模。

简而言之，软件服务的松散耦合不仅会在技术方面产生影响，还会影响软件开发的社会技术方面。康威定律指出，组织的设计系统反映了他们的通信结构，这个定律的核心就是耦合。如果系统的松散耦合程度较高，意味着组织的松散耦合程度较高，且他们采用的开发方法较为分散、可扩展性较强。

令人惊讶的发现

今年的研究显示，松散耦合架构可能会导致团队出现倦怠现象。这是一个令人惊讶的发现，与前些年的发现结果相矛盾。我们的分析显示，在信息可以自由流动的稳定团队中，倦怠率较低。Westrum 的生成式文化和团队稳定性支持松散耦合架构且可降低倦怠率，因此这显然是矛盾的。我们还需要进行更多的研究，才能得出明确的结论。

同时，如果安全性要求由安全性组织统一定义和控制，团队可能更加难以独立开发软件，必须依赖其他团队。这进一步说明了改由应用的主要负责团队处理安全性问题的优势（另请参阅：供应链安全为何至关重要）。这只是组织中耦合的一个较小的表现形式之一，尽管我们收集的是安全性方面的数据，但这可能同样适用于其他集中式功能。允许团队针对安全性和其他经常集中处理的功能自行制定决策，有助于您的组织更快获得采用松散耦合架构的优势。

Daniella Villalba

文化

“嗯，这就是我们这一行的做法。”

在描述所在的组织应对挑战和把握机遇的方法时，各个行业的从业人员可能已经将这句话说过无数次了。

每个组织都有自己独特的文化，我们的研究一再表明，文化是组织迈向成功和确保员工身心健康的基础。

文化也是 DevOps 的一个必要方面，因为从根本上说，DevOps 就是关于工具、实践以及**人员如何协作**来以快速、可靠且安全的方式开发和交付软件的。了解影响组织文化的因素，有助于领导层从正面解决与文化相关的问题。因此，培养健康的文化应该是组织的优先事项。与文化相关的问题如果得不到妥善处理，可能会阻碍 DevOps 实践的采用。

今年，我们继续使用 Westrum 的组织类型学模型来衡量组织文化的健康状态。此外，我们通过衡量团队成员流失率、弹性工作安排、感知到的组织认同感和倦怠率，进一步了解文化。

今年的研究数据支持以往的发现结果，即组织内的文化类型会影响**组织绩效**。具体而言，相较于官僚型和病态型文化，**生成式文化**通常有利于组织获得更出色的组织绩效。

在拥有生成式文化的组织中，员工更有可能制作出更优质的文档，全力投入有意义的工作，其所在团队的流动率也可能较低。

团队成员流失率

通过调查团队成员流失率，我们发现**稳定的团队**（即组成人员在过去 12 个月内没有太大变化的团队）属于高绩效组织的可能性更大。人员不断流失会影响效率和士气，因为新的团队成员需要时间来上手，而现有成员

Westrum 组织文化

病态型 权利导向型	**官僚型** 规则导向型	**生成式** 绩效导向型
缺少合作	合作程度一般	高度合作
阻挠信使	不重视信使	训练信使
逃避责任	各担责任	共担责任
阻碍交流	容忍交流	鼓励交流
失败时寻找替罪羊	失败时公平惩罚	失败时追根溯源
压制新想法	认为新想法会招引麻烦	接纳新想法

可能需要适应工作量和团队动力的变化。此外，我们的研究显示，与成员流失率较高的团队相比，稳定的团队更有可能制作出优质的文档。持续面临变动的团队可能较难持续采用有助于制作出优质文档的实践。

高绩效组织更有可能采用灵活的工作安排。

灵活的工作安排

由于自新型冠状病毒肺炎 (COVID-19) 疫情爆发以来，许多组织都改为采用灵活的工作安排，因此我们调查了让员工能够自由选择远程、现场或混合办公 是否有助于提升组织成效。

结果显示，相较于工作安排僵化的组织，**员工灵活性**较高的组织拥有更出色的组织绩效。这项发现结果印证了，让员工可以根据需要自由调整工作安排，会为组织带来切实而直接的好处。

倦怠

倦怠是指对工作感到畏惧、提不起兴趣以及产生负面情绪。当员工处于倦怠状态时，不仅会缺乏动力，感到疲惫不堪，还更有可能降低对工作的满意度，而这会导致员工离职率上升。倦怠状态可能会导致各种心理和生理健康问题，例如患上抑郁症、焦虑症和心脏病以及产生自杀念头的风险增加[1]。

去年，我们评估了 COVID-19 疫情背景下的倦怠情况，发现在拥有生成式文化的组织中，员工倦怠率较低。

[1] Maslach C、Leiter MP。“Understanding the burnout experience: recent research and its implications for psychiatry”（了解倦怠体验：最近的研究及其精神病学影响）。《World Psychiatry》。2016 年 6 月；15(2):103-11。doi：10.1002/wps.20311。PMID：27265691；PMCID：PMC4911781。

灵活的工作模式有助于降低员工倦怠率，提高员工将所在团队作为适合加入的团队进行推荐的可能性。

今年的问卷调查增进了我们对倦怠现象的了解，而结果同样证明，稳定的团队与灵活的工作安排有助于降低倦怠率。除此之外，今年我们还衡量了团队净推荐值 (NPS)，这个指标会指明员工是否会向好友或同事推荐所在团队。我们发现，团队 NPS 与感知到的领导层认同感有关。与倦怠现象的调查结果一样，我们发现生成式文化、稳定的团队和灵活的工作安排有助于提高员工向他人推荐所在团队的可能性。

员工如何看待所在的组织

最后，我们请受访者预测他们的团队在未来 12 个月会获得多少支持，以调查感知到的领导层认同感。结果显示，如果感知到的领导层认同感较高（例如获得更多资金支持、分配到更多资源以及赞助），组织绩效会很出色。

此外，我们还请受访者预测未来 12 个月出现安全事故或服务完全中断的可能性。结果显示，在高绩效组织工作的员工预测会发生重大错误的可能性较小，他们对自己的组织抱有更积极的看法。同样，我们发现，在软件和交付表现出色的组织工作的员工认为需要改变当前实践来提高业务成效的**可能性较小**。

关于代表性的一些发现

我们的调查结果显示，属于弱势群体的员工花费更多时间来处理计划外工作的可能性较高，无论他们属于高绩效组织还是低绩效组织，都是如此。我们还发现，相较于不属于弱势群体的员工，属于弱势群体的员工的倦怠率更高。团队负责人应注意避免工作量失衡，并确保在团队成员之间公平分配工作。

总而言之，这些发现结果表明，为员工创造健康且具有包容性的组织和团队环境至关重要。

尽管我们不断强调文化的重要性，但我们了解，改变甚至改善组织的文化并非易事。我们建议组织在进行 DevOps 转型的过程中，先了解员工的体验，然后再投入资源来解决与文化相关的问题。

04

供应链安全为何至关重要

John Speed Meyers

Todd Kulesza

2020 年 11 月，少数技术专业人士认为软件供应链安全危机正酝酿当中。Open Source Security Foundation 成立的宗旨是关注开源软件的安全性，它延续了前人的努力，尽管在解决此问题的过程中取得了一些成果，但这一话题并未受到各主流媒体的重视。SolarWinds 遭遇了一次重大攻击，之后一切都发生了变化。当攻击者趁软件更新之际，通过特洛伊木马悄悄入侵数千个大型公司和政府网络时，时代也在迅速变化。

如今，保护软件供应链安全刻不容缓已成为共识，即使人们不在家庭聚餐时谈论，也肯定会在会议室中讨论。

许多计划应运而生，软件行业的大部分公司也都致力于革新自己的软件供应链安全性实践，并提高开源共享的安全性。

在本章节，我们将着重介绍两项计划：软件工件的供应链级别（SLSA，读作“salsa”）以及NIST安全软件开发框架（SSDF）。这两项计划均提供一系列防御措施，以确保攻击者无法篡改软件生产流程以及通过恶意软件更新绕过网络防御程序。

但是，与 SLSA 和 SSDF 相关的软件供应链安全性实践的采用范围有多广？哪些实践需要帮助推动采用？哪些实践已被广泛采用？这些问题至今还没有系统性的答案。通过调查数百位软件专业人士采用供应链安全性相关实践的情况，我们初步得出了一些答案。具体而言，有四项主要发现结果：

01 **已经开始采用**：SLSA 和 SSDF 中包含的软件供应链安全性实践的采用已达到一定规模，但仍有很大的提升空间。

02 **更健康的文化有助于抢占先机**：组织文化是软件开发安全性实践的一个主要驱动因素，与信任度较低的组织文化相比，信任度较高的“不责罚”文化更有可能确立 SLSA 和 SSDF 实践。

03 **存在关键的集成点**：软件供应链安全性的技术实践的采用似乎取决于是否使用 CI/CD，后者通常为许多供应链安全性实践提供集成平台。

04 **提供意想不到的优势**：除了降低安全风险外，较为出色的安全性实践还会带来其他优势，例如降低倦怠率。

目前各公司采取什么措施来避免安全漏洞

为了更好地了解目前各组织采取哪些措施来识别和解决所开发软件中的安全漏洞，我们在今年的调查问卷中新增了二十多个问题。这些问题大致可分为两类：

- 请受访者选择是否赞同某项陈述的问题（例如“我所在的组织能够采用有效的措施来应对安全威胁”或“我可以使用必要的工具来执行安全测试”）。

- 请受访者判断所在组织的安全性实践是否成熟的问题（例如“构建作业通过构建脚本定义，而未采用任何其他方式”或“正式版本使用集中式 CI/CD 系统构建，一律未通过开发者的工作站”。我们之所以使用“成熟/不成熟”量表，是因为早期测试发现，受访者更倾向于赞同一些与安全性相关的问题。不过，对于 SSDF 相关问题，用“赞同/不赞同”的回复量表来评估会更加自然。

在撰写本报告时，**SLSA 框架**版本为 v0.1，该框架描述了一系列与 SLSA“级别”相关的软件供应链完整性实践，其中 SLSA 级别越高，对应的软件供应链安全措施等级也就越高。我们向受访者提出了许多特定的 SLSA 实践相关问题。具体而言，问卷调查询问了如下问题：“就您所开发的主要应用或服务而言，以下实践的成熟度如何?”表 1 列出了调查问卷中所涵盖的 SLSA 相关实践的表述。

SSDF 目前为 v1.1 版，其中的实践侧重于帮助组织减少发布的软件中的漏洞，并最大限度减少其余漏洞的潜在影响。与 SLSA 的“级别”不同，SSDF 实践分为以下四类：让组织做好准备、保护正在开发的软件、开发安全性良好的软件，以及有效响应发现的漏洞。问卷调查请受访者回答在多大程度上赞同（或不赞同）对多种 SSDF 实践的表述；表 2 对这些问题进行了总结。

SLSA 实践	调查问卷定义
集中式 CI/CD	正式版本使用集中式 CI/CD 系统构建，一律未通过开发者的工作站
保留历史记录	修订版本及其更改历史记录将会无限期保留
构建脚本	构建作业完全通过构建脚本定义，而未采用任何其他方式
隔离	构建作业是隔离开来的；不会干扰并行或后续构建作业
构建文本文件	构建定义和配置在版本控制系统中存储的文本文件中定义
参数元数据	关于工件的构建元数据（例如依赖项、构建流程、构建环境）包括所有构建参数
依赖项元数据	关于工件的构建元数据（例如依赖项、构建流程、构建环境）记录了所有依赖项
生成元数据	构建元数据（例如依赖项、构建流程、构建环境）由构建服务或读取构建服务的构建元数据生成器生成
防止输入	运行构建时，防止各个构建步骤动态加载任何构建输入（即预先提取所有必需的资源和依赖项）
用户无法修改	构建服务用户无法修改关于工件的构建元数据（例如依赖项、构建流程、构建环境）
元数据可用	有需要的人员可以通过中央数据库等途径使用构建元数据（例如依赖项、构建流程、构建环境），并且能以可接受的格式将其导出
两人审核	修订版本的历史记录中的每项更改在提交前必须分别由两名值得信赖的人员审核并批准
签署元数据	关于如何制作工件的构建元数据（例如依赖项、构建流程、构建环境）由我的构建服务签署

表 1. 与 SLSA 相关的问卷调查问题

注意：每个问题提供五种可能的回复：完全不成熟、还算成熟、比较成熟、非常成熟和完全成熟。

SSDF 实践	调查问卷定义
安全审核	我会对所开发应用的所有主要功能进行安全审核
持续代码分析/测试	我们会持续对所有受支持的版本执行自动或手动代码分析和测试，以发现或确认是否存在以前未检测到的漏洞
早期安全测试	我或其他团队会在软件开发流程早期运行安全测试
有效应对威胁	我所在的组织能够采用有效的措施来应对安全威胁
与开发团队集成	安全性角色已集成到我们的软件开发团队中
文档要求	我们的组织具有相应流程来识别和记录组织开发或获取的软件（包括第三方和开源软件）的所有安全性要求
定期审核要求	我们会定期审核安全性要求（每年审核一次，或视需要提高频率）
生成元数据	构建元数据（例如依赖项、构建流程、构建环境）由构建服务或读取构建服务的构建元数据生成器生成
与开发周期集成	我的公司已将软件安全协议无缝内置到开发流程中
针对各个项目设置标准流程	我的公司有一套标准化的流程来处理各个项目的软件安全性问题
监控安全性报告	我们持续监控来自公共来源的、有关我们所用软件及其第三方组件中的潜在漏洞的信息
具有必要的工具	我有权访问执行安全测试所需的工具

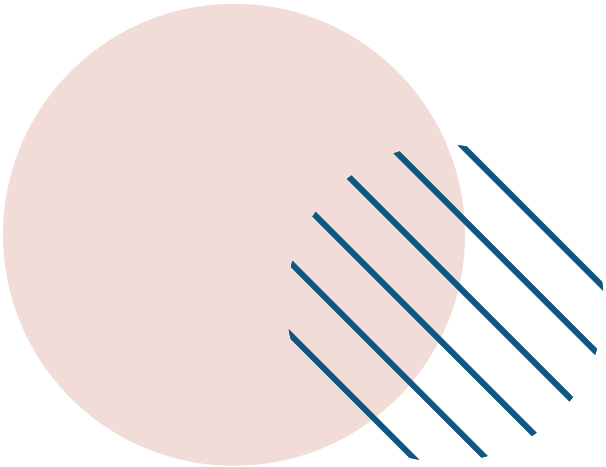
表 2. 与 SSDF 相关的问卷调查问题

注意：每个问题提供七种可能的回复：非常不赞同、不赞同、不太赞同、说不好、有点赞同、赞同和非常赞同。

总体而言，我们发现新兴行业实践得到了相对广泛的采用，但还有很大的发展空间。例如，虽然有 66% 的受访者赞同“我的公司已将软件安全协议无缝内置到开发流程中”这一表述，但只有 18% 的受访者强烈赞同。我们汇整了受访者对安全性问题的回复情况，如图 1 和图 2 所示。

我们发现，利用持续集成/持续交付 (CI/CD) 系统来发布正式版本是最常见且已趋成熟的实践，63% 的受访者表示这种实践“非常”或“完全”成熟。CI/CD 实践成熟度居于榜首的结果印证了以往的一项安全性研究，该研究发现，大多数组织会在其 CI/CD 流程中进行应用级安全扫描。此外，一系列以安全性为主题的单独定性访谈表明，在开发过程中，大多数开发者无法在本地运行此类工具。SLSA 框架同样建立在 CI 系统之上，是供应链安全性的中央集成点。我们的模型分析发现，组织采用 CI 将有助于提高其安全性实践成熟度，我们将在下一部分中对此进行介绍。因此，我们认为，如果没有这一关键的基础架构部分，组织很难确保针对所创建的软件工件一致地运行一组扫描程序、linter 和测试。

除了 CI/CD 之外，其他常见且已趋成熟的实践包括：无限期保留代码历史记录 (60%)；仅通过脚本定义构建作业 (58%)；隔离各个构建作业 (57%)；以及将版本定义存储在源代码控制系统中 (56%)。相较之下，最不常见且最不成熟的两个实践分别是：需要安排两个或更多审核人员批准每项代码更改 (45%)，以及签署构建元数据以防止/检测篡改行为 (41%)。



除了实践成熟度相关问题之外，我们还提供了一系列关于其所在组织的安全性的表述，请受访者选择是否赞同。赞同比例最高的表述是："我们持续监控来自公共来源的、有关我们所用软件及其第三方组件中的潜在漏洞的信息"，有 81% 的受访者表示赞同。另一方面，赞同比例最低的表述是："公司的软件安全性流程拖慢了我经手的应用的开发流程"，56% 的受访者赞同安全性实践对软件开发造成负面影响。

尽管此表述的赞同比例最低这一点令人欣喜，但大多数受访者认为当前的安全性流程拖慢了开发速度这一事实表明，安全性工具和方法还有很大的改进空间。我们的模型分析也支持此解读，指出安全性流程对软件交付表现的影响是多方面的（尽管影响很小）。

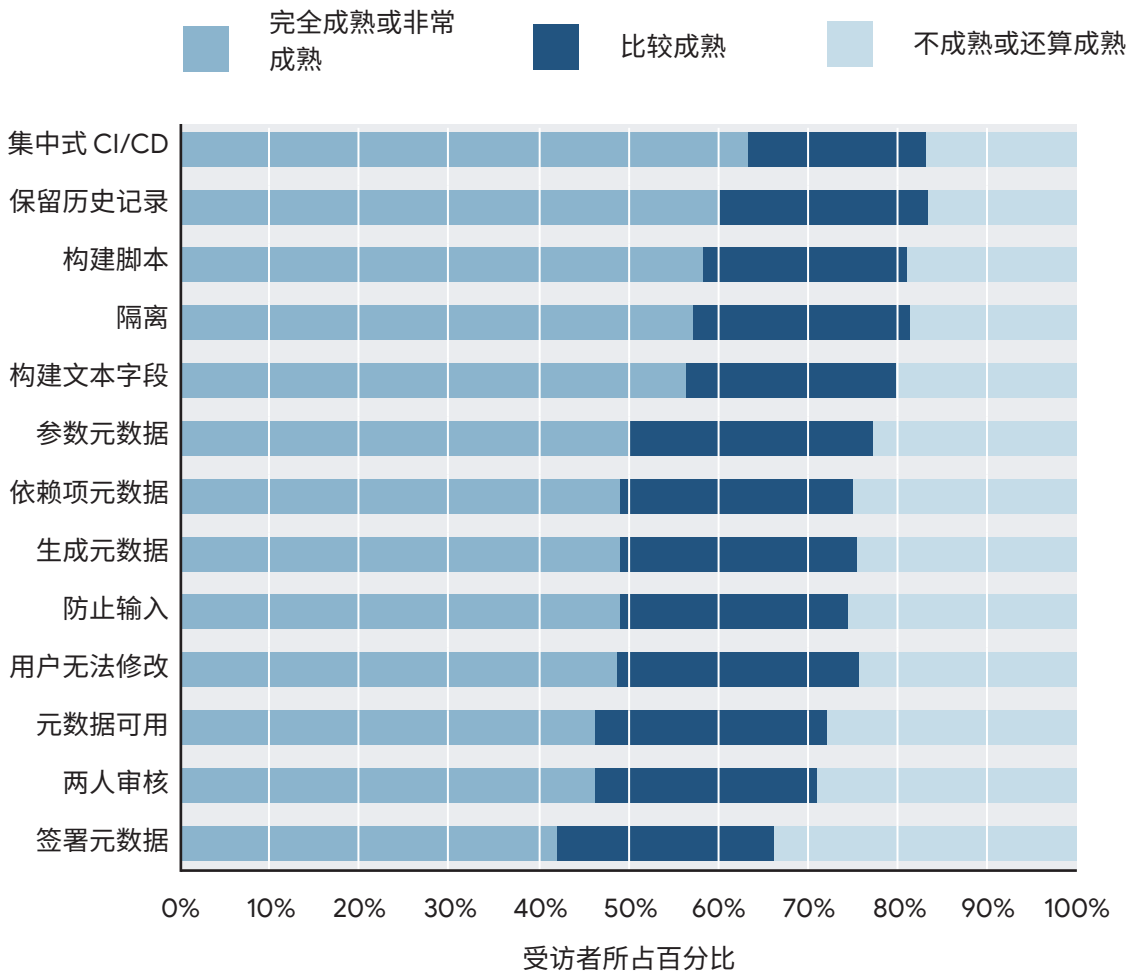


图 1. SLSA 实践成熟度

与 SLSA 实践成熟度有关的调查问卷回复。大多数受访者表示上述所有实践已达到一定成熟度，但相对较少的受访者表示已“完全”成熟。

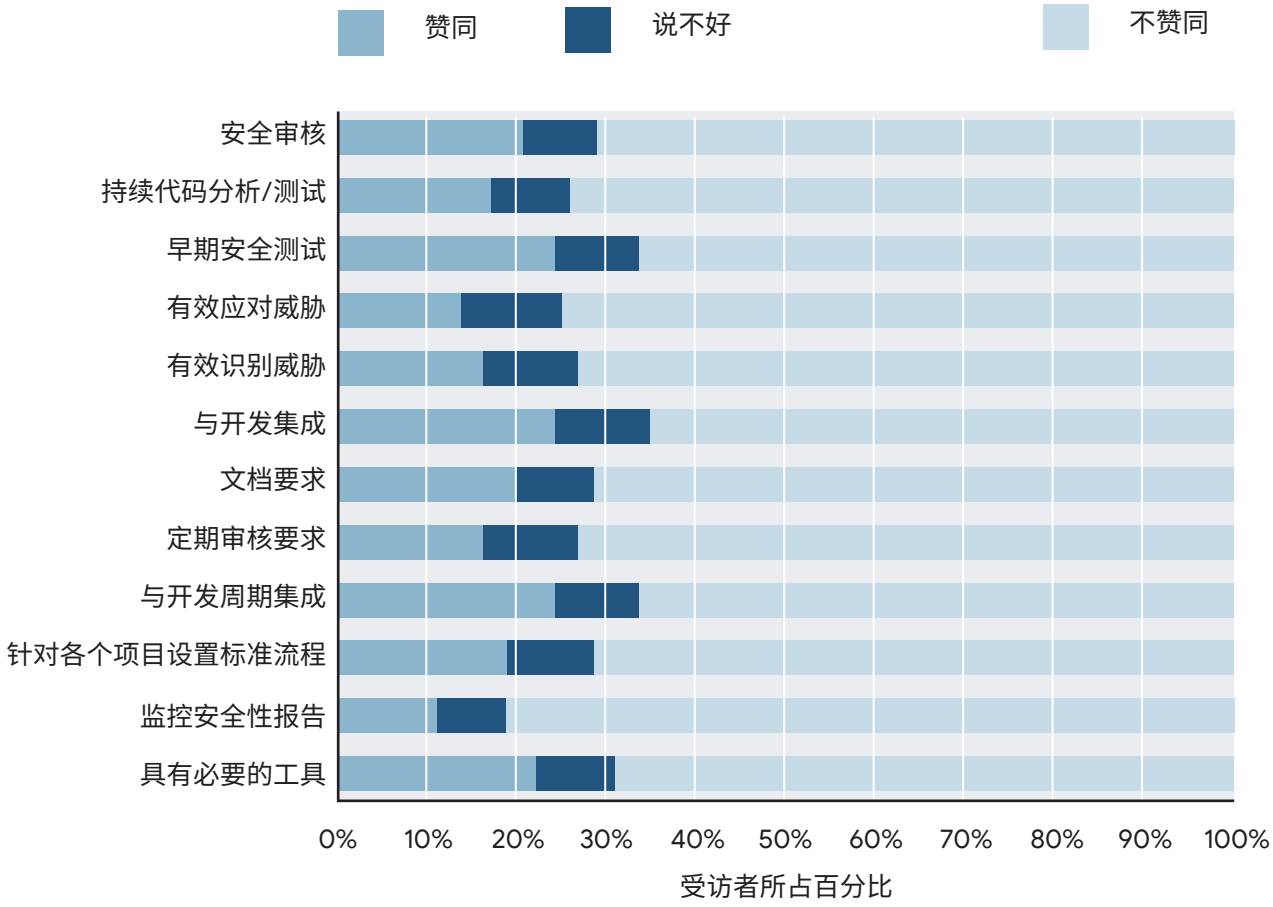


图 2. SSDF 实践成熟度

与 SSDF 实践成熟度有关的调查问卷回复。与 SLSA 实践成熟度调查结果类似，大多数受访者都赞同他们的组织采用了所有这些实践。

哪些因素有助于公司采用良好的安全性实践？

在软件开发过程中，应用安全保障只是开发者需要顾及的一个方面，因此他们在相关实践上投入的时间和精力有限。如果安全性实践的采用阻力较大，可能会让开发者感到头疼，并且总体而言是无效的，因为人们总是会尝试避开阻碍。例如，根据与专业软件工程师的一系列研究访谈，他们仅在项目开始或结束时与安全团队接触，而且很难与这些团队互动。用一位受访者的话来说："我们有应用安全保障团队，但我从来没有让他们审核过我的代码…和大多数工程师一样，我通常会避开他们。"

提升软件安全性的一种方法是，减少采用安全性实践的障碍。受访的开发者希望采用安全性实践，并且多次提到，他们总是需要优先考虑发布功能和修复，而无法顾及潜在安全性问题，这一点令人沮丧。例如，在另一项安全性相关问卷调查中，一位受访者表示："最大的安全性挑战就是将安全性视为第一要务。它并不有趣，也无法提高产品销量，[而且]在成为问题之前都不是问题。"

我们的问卷调查数据显示，有几个因素可让开发者更轻松地"采用安全性实践"。

我们发现，最大的因素不是技术，而是文化：**文化与 Westrum"生成式"文化最接近的组织明显更有可能表示已广泛确立安全性实践**（如 SLSA 框架所定义）[1]。生成式文化涵盖的方面包括：高度合作，共担风险和责任，以及从过去的错误中吸取教训。我们假设这些特质能以多种方式提升安全性实践的运作成效，例如：鼓励软件工程师更加积极主动地关注供应链安全；奖励那些在提升安全性方面付出努力的人员，无论其职务如何；降低报告潜在安全问题的感知风险。

从技术上讲，有助于提升安全性的三个最重要因素都与基础架构有关。这当然是有道理的：如果您的基础架构可让工程师更轻松地执行漏洞扫描或手动代码审核等任务，他们就更有可能使用该基础架构。具体而言，我们发现，**如果组织具有源代码控制、持续集成和**

[1] 有趣的是，在回答 NIST SSDF 相关问题时，同一组受访者表示赞同的可能性并没有更高。尽管 SLSA 和 SSDF 探讨的是应用开发安全性的不同方面，但我们预计这几组问题存在重叠。如前所述，由于 SSDF 的回复量表偏向于得出"赞同"回复，因此会出现此差异。

持续交付系统，往往也拥有更成熟的 SLSA 实践。借助这些系统提高安全性的关键在于，开发者何时注意到安全问题，另一项问卷调查发现，主要是在 CI 期间。通常情况下，CI 流程早于代码审核，在此流程中，开发者会运行漏洞扫描程序和其他代码分析工具，以确保所有代码提交均符合相同的安全要求。如果缺少集中式构建系统，会更加难以执行这种一致的扫描；而如果缺少源代码控制，反过来又难以从一开始就拥有集中式构建系统。

然而，对软件工程师而言，将安全性扫描纳入 CI/CD 中可能还不够早。在一组安全性相关访谈中，受访的应用开发者一致表示，在自己的开发工作站进行安全扫描有助于节省时间和精力。受访者经常提到以下两种情况：1) 希望提前了解自己是否基于具有已知漏洞的依赖项进行构建，以便在基于该依赖项进行构建之前重新评估是否要使用它；2) 避免较长的 CI 等待时间，有时等待时间长达数小时，却只是为了确认当前所做更改是否已解决某个安全问题。软件工程师表示，虽然 CI 是必要的“后盾”，但在这两种情况下，能够在本地运行相同的安全工具有助于提升工作的速度和效率。

上文所述的文化和技术因素能够最有效地提升安全性，但并不是仅有的驱动因素。其他重要的因素包括：

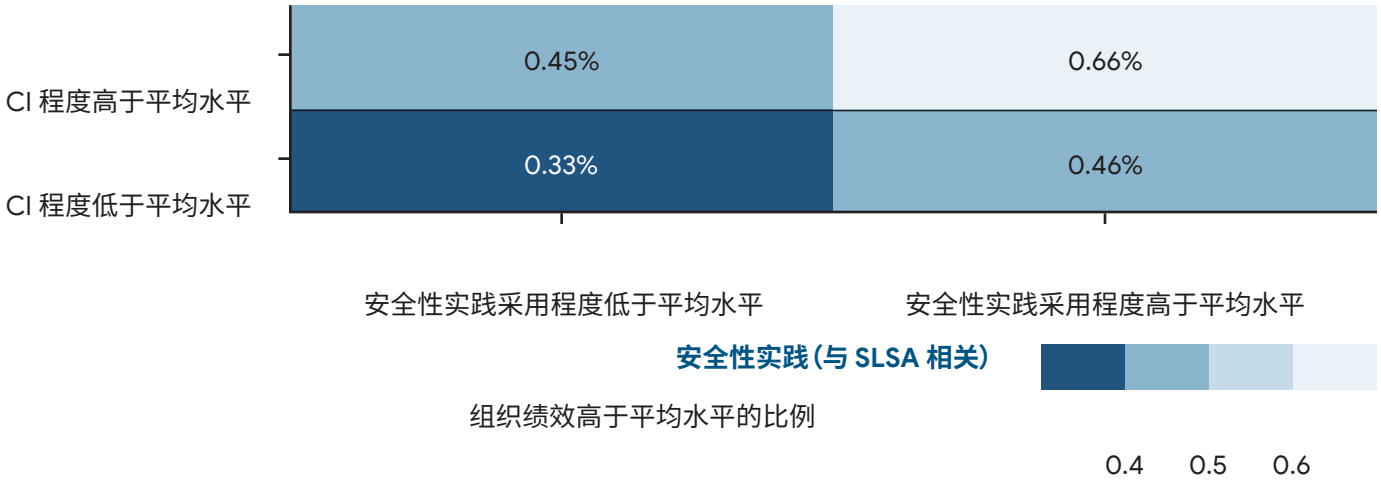
- 工作安排灵活（例如，组织是否支持在家办公?）
- 使用云端（公有云或私有云）
- 使用“云原生”应用或服务
- 感受到企业对团队的重视和投资
- 团队成员流动率较低
- 组织规模（组织规模越大，安全性得分就越高）

不过，这些因素似乎大多与 Westrum 的生成式文化（例如，工作安排灵活、感觉受到组织的重视或团队成员流动率较低）或 CI/CD 的使用（例如，使用云原生应用或在大型组织工作）相关联。这些数据使我们相信，组织文化和现代开发流程（例如持续集成）是组织应用开发安全性的最大驱动因素，同时也是希望改善安全状况的组织的最佳起点。

良好的安全性实践会带来哪些成效?

随着组织改善软件开发的安全性实践,他们有望获得哪些优势?我们的问卷调查数据证实,受访者预计,**随着公司的供应链安全性实践成熟度的提高,安全事故、服务中断和性能下降的可能性会降低**。同样,我们在 2022 年上半年进行的另外一项研究的结果显示,在 CI 期间运行漏洞扫描程序等工具可显著提高识别软件依赖项中漏洞的概率:使用此类工具的受访者在自己的代码或某个依赖项中发现安全漏洞的可能性是其他受访者的近两倍。简而言之:SLSA 和 SSDF 实践的效用一如预期。我们并没有声称这些实践可以消除安全威胁,但我们的证据表明它们确实可以降低组织的安全风险。

安全性实践也有助于提升基于绩效的成效,但有一个前提:CI 扮演着重要的角色。如果未采用 CI,则安全性实践不会对软件交付表现产生任何影响。但如果采用了 CI,安全性实践将会对软件交付表现产生明显的积极影响。这实际上意味着,必须采用 CI,安全性实践才有助于提升软件交付表现。此外,安全性实践通常会对组织绩效产生积极影响,如果 CI 非常成熟,这种影响将会放大。下图试着直观呈现了这种影响。



除了感知到的安全风险降低之外，受访者还表示，团队成员的**倦怠率降低**，且他们将所在团队作为**值得加入的团队**进行推荐的意愿提高。这两项发现结果均表明，对于软件工程师来说，安全性是在原本的众多工作之外需要完成的又一项工作。各种工具和流程可帮助开发者将安全性实践整合到现有的开发流程中，而不是在发现威胁时执行计划外工作或“消防演习”，从而提供了一种降低安全风险和提高开发者幸福感的机制。

总而言之，我们的证据表明，**健康状况良好的高绩效团队也往往会广泛确立良好的安全性实践**（不过，如前所述，仍有改进的空间）。仅凭 SLSA 或 SSDF 框架等方法可能无法提升我们衡量的所有文化和绩效指标的表现，但很显然，追求安全性不需要以牺牲其他开发优先事项为代价。

各种工具和流程可帮助开发者将安全性实践整合到现有的开发流程中，而不是在发现威胁时执行计划外工作或“消防演习”，从而提供了一种降低安全风险**和**提高开发者幸福感的机制。

05

意料之外的现象

Derek DeBellis

虽然每年的报告都会重点分析当年的调查问卷回复，但我们会尽可能根据《DevOps 现状》报告的完整目录架构以及相关研究（例如倦怠率和文化相关研究）来理解这些发现结果。本研究计划的核心原则是，通过复制性研究来测试这些影响的可靠性。这使我们有机会根据数据调整看法，了解不断变化或新兴的趋势。

今年，我们遇到了一些意想不到的现象。造成这种结果的潜在原因有很多。首先，与之前报告中的样本相比，今年的样本发生了变化，包含更多处于职业生涯早期阶段的人。一种解读是，我们获得的回复更多地来自直接负责采用技术实践和能力的人员，而不是可能负责监督或指导这些实践的采用的人员。另一种可能性是，行业或世界发生了变化；过去行之有效的东西不一定适用于未来。一些因素，例如宏观经济力量以及 2021 年的大部分时间继续笼罩在 COVID-19 疫情的阴影下，可能改变了 DevOps 的实质特性。最后，我们模型中所包含因素的细微变化可能改变了各个变量之间的关系。[1]

[1] Judea Pearl 的《Book of Why》（因果革命）和 Robert McElreath 的《Statistical Rethinking》（统计反思）通过一些绝佳的示例，阐述了在统计模型中包含和未包含的因素如何影响模型的输出结果。

获得出乎意料或与假设相反的发现结果会让研究人员在撰写报告时陷入困境。鉴于这些发现结果可能是错误的结果，或者至少尚未获得多项研究的实验证据支持（甚至与这些证据相矛盾），负责任的做法是进行后续研究，尝试重现这些结果并了解原因[2]。然而，重点关注意料之外的发现结果，也可能会使研究人员忽视多年来的可靠研究成果。对于每份《DevOps 现状》报告，我们都会以统计方式探索超过一百种分析路径，这可能会使我们偶然得出错误的发现结果，因此，我们每年都会进行复制性研究，以抵消这种影响。

另一方面，不记录这类发现结果可能会造成文件抽屉效应[3]，即符合预期或可接受的有效结果将广为人知，而不符合预期或难以接受的结果将不会公诸于世。我们致力于在这两者之间取得平衡：虽然不想过度强调新的发现结果，但也认为有必要分享它们。以下是最让我们感到意外的发现结果，以及我们认为这些结果所代表的含义：

01 我们一再观察到，主干开发实践有助于提升软件交付表现。事实上，自 2014 年以来，每年的研究结果都证实了这一点。今年，主干开发能力的行为出乎我们的意料。举例来说，主干开发能力会对软件交付表现产生负面影响。这与以往的研究结果完全相反。鉴于此发现结果非常异常，我们迫切地想要看看它在接下来的研究中是否会重现，并了解社区对此是否有任何解释。

02 我们发现，只有当运营绩效也较高时，软件交付表现才会对组织绩效产生积极影响，然而许多受访者的运营绩效并不高。此结果与我们往年的研究相悖，在先前的每一次研究中，软件交付表现与组织绩效之间的联系都更为明显。

[2] Kerr, N. L.（1998 年）。“HARKing: Hypothesizing after the results are known”（HARK 法：知道结果后再假设）。《Personality and social psychology review》（人格与社会心理学评论），2(3)，196-217。
[3] Rosenthal, R.（1979 年）。“The file drawer problem and tolerance for null results”（文件抽屉问题和对 null 结果的容忍度）。《Psychological bulletin》（心理学公报），86(3)，638。

03 文档实践会对软件交付表现产生负面影响。这与之前的报告相悖。我们对此所做的一种假设是，文档实践的自动化程度越来越高，尤其是在高绩效团队中。不过我们掌握的证据不足以支持或反驳这个假设，必须收集更多数据。

04 部分技术能力（即主干开发、松散耦合架构、CI、CD）可能会造成倦怠。如上所述，与往年相比，今年的样本中有许多明显处于职业生涯早期阶段的受访者。因此，我们收到的回复大多来自负责实施这些能力的人员，而不是负责制定或监督该计划的人员。实施流程明显比监督工作更具挑战性。我们希望进行进一步的研究，以便更好地了解此发现结果。

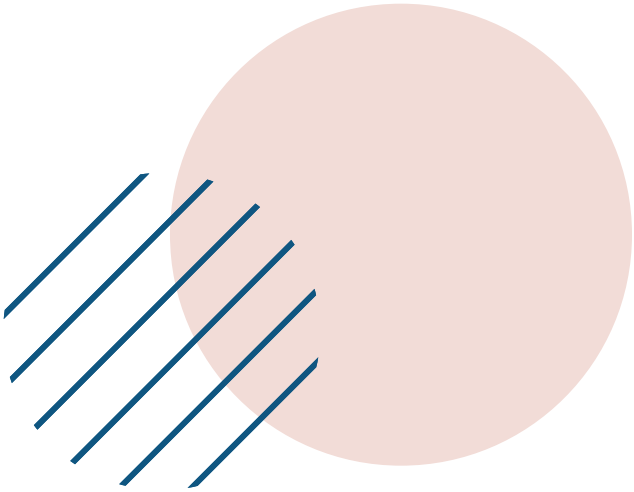
05 可靠性工程实践会对软件交付表现产生负面影响。一种解释是，这两者之间不一定有因果关系。今年，我们在一项新的聚类分析（请参阅“如何比较”一节）中发现，一部分组似乎专注于可靠性，而忽略了软件交付表现。我们认为这两者并不挂钩，您可以只关注一个，而忽视另外一个，但如果要让软件交付表现带动组织绩效的提升，终究需具有较高的可靠性。

06 我们新增了与 SLSA 相关的实践，以了解团队是否正在采用这些方法来维护软件供应链安全。虽然我们预计安全性实践的采用与绩效之间有一定关联（例如，使用技术能力、提升软件交付表现、提升组织绩效），但我们惊讶地发现，技术能力实际上是通过安全性实践这一机制影响软件交付表现和组织绩效。

由此看来，持续集成、版本控制和持续交付对软件交付表现和组织绩效的大部分影响可能都与 SLSA 相关实践的采用有关。换句话说，我们从数据中发现了一条因果链，即许多技术能力都对 SLSA 相关实践有积极影响，然后通过对 SLSA 相关实践的这种积极影响，推动软件交付表现和组织绩效提升。我们通过中介分析发掘出此结果[4][5]。这促使我们探索对 SLSA 相关实践的衡量是否跟踪团队的其他特征（例如一般绩效），以及安全性实践以何种方式提升软件交付表现和组织绩效。

我们期望明年再次研究这些影响，看看我们是否可以重现和解释这些新模式，或者我们是否应该将其视为离群值而不予考量（但也应该尝试进行解释）。与以往一样，欢迎社区成员提供反馈。

加入 DORA 社区 (http://dora.community)，继续讨论这些意料之外的现象以及今年报告中的其他发现！



[4] Jung、Sun Jae。“Introduction to Mediation Analysis and Examples of Its Application to Real-world Data”（中介分析简介及其在实际数据中的应用示例）。《Journal of preventive medicine and public health》（预防医学与公共卫生杂志）= Yebang Uihakhoe chi vol. 54,3（2021 年）：166-172. doi:10.3961/jpmph.21.069

[5] Carrión、Gabriel Cepeda、Christian Nitzl 和 José L. Roldán。“Mediation analyses in partial least squares structural equation modeling: Guidelines and empirical examples”（偏最小二乘结构方程建模中的中介分析：指南和实证示例）。《Partial least squares path modeling》（偏最小二乘路径建模）。Springer、Cham，2017 年。173-195。

06

受众特征和企业特征

Derek DeBellis

感谢诸位对我们的研究以及整个行业的贡献！

哪些人参与了问卷调查？

《DevOps 现状》报告整合了八年的研究心血和超过 33,000 条业界专业人士的问卷调查回复，归纳出可使团队和组织取得出色成效的软件开发和 DevOps 实践。今年，来自全球各行各业的超过 1,350 名专业人士分享了他们的经验，帮助我们更好地了解了有助于提高绩效的因素。感谢诸位对我们的研究以及整个行业的贡献！总而言之，受众特征和企业特征衡量的取样比例仍保持着高度的一致性。

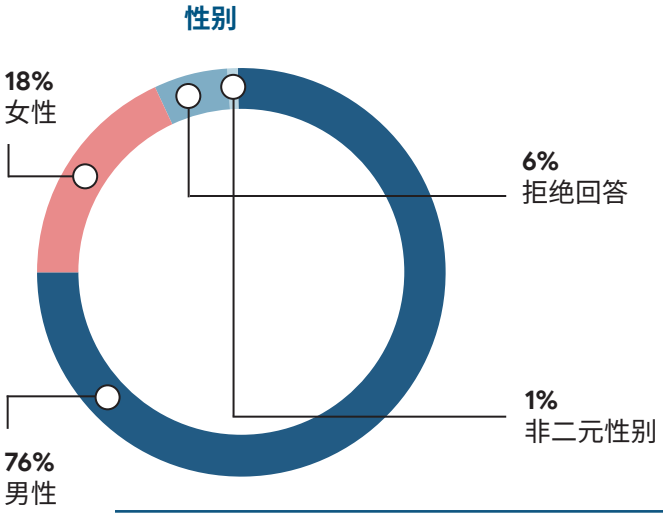
与往年类似，我们收集了每一位调查问卷回复者的受众特征信息。类别包括性别、残障人士和弱势群体。

今年，公司规模、行业和区域等企业特征类别的取样比例和之前的报告一致。其中超过 60% 的受访者是工程师或经理，且三分之一来自科技行业。此外，还有来自金融服务、零售和工业/制造公司的业界代表。

受众特征

性别

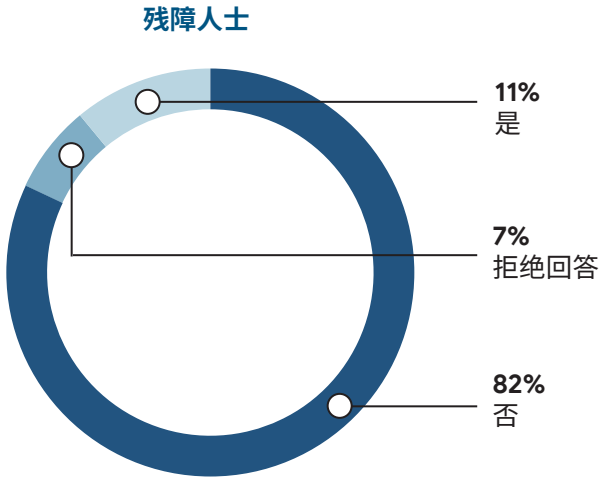
在今年的样本中，女性受访者所占的比例 (18%) 高于 2021 年 (12%)。男性受访者所占的比例 (76%) 低于 2021 年 (83%)。受访者表示，女性在其团队中所占比例为 25%，这与 2021 年 (25%) 相同。



女性所占百分比：**25%（中位数）**

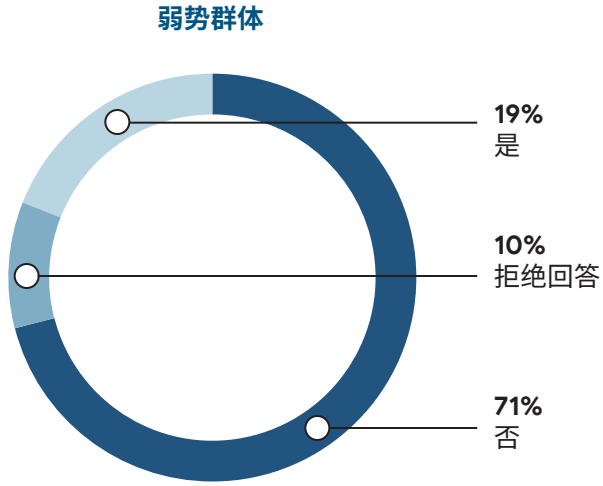
残障人士

我们根据 Washington Group Short Set（华盛顿小组简短问卷），从 6 个维度来确定残障情况。这是我们第四年询问残障人士相关信息。残障人士所占比例为 11%，与我们 2021 年的报告一致。



弱势群体

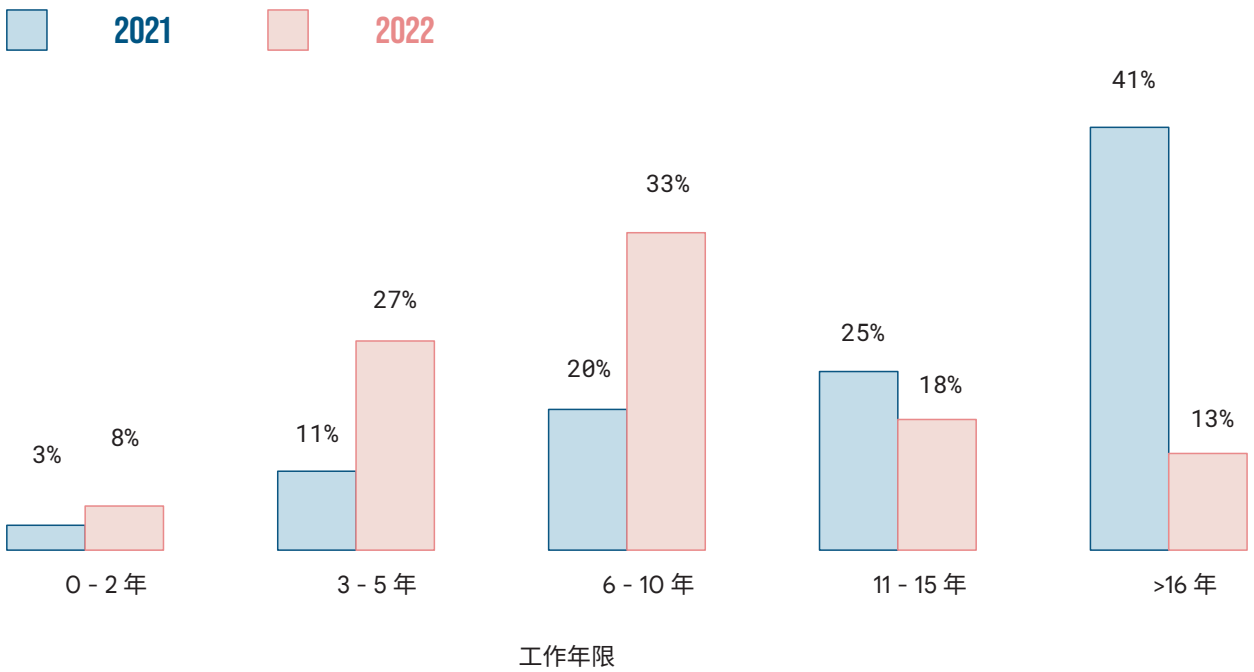
受访者可能会因种族、性别或其他特征而被认定为属于弱势群体。这是我们第五年询问弱势群体相关信息。属于弱势群体的受访者所占比例从 2021 年的 17% 小幅增加到 2022 年的 19%。



工作年限

今年，拥有 5 年或以下工作经验的受访者所占比例 (35%) 明显高于 2021 年 (14%)。此外，拥有 16 年以上工作经验的受访者所占比例 (13%) 要比 2021 年 (41%) 低上许多，这一点或许并不令人意外。

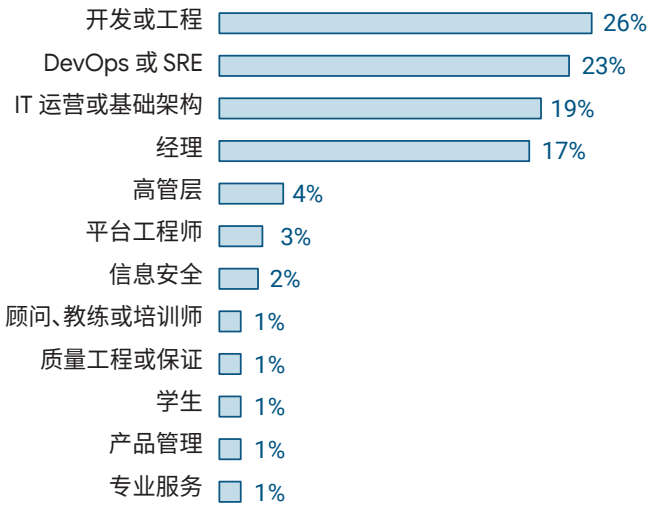
这种变化或许可以解释数据中出现的一些模式，我们认为在解读结果时牢记这一点非常重要，尤其是在与去年进行比较时。



企业特征

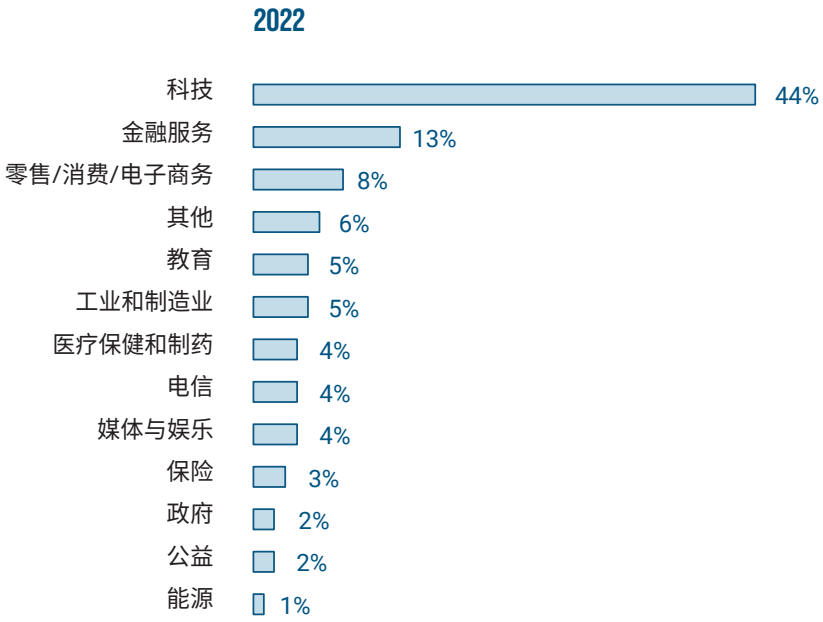
职务

85% 的受访者来自开发或工程团队 (26%)、DevOps 或 SRE 团队 (23%)、IT 运营或基础架构团队 (19%)，或担任经理职务 (17%)。来自 IT 运营或基础架构团队的受访者所占比例 (19%) 是去年 (9%) 的两倍多。与去年相比，有两个职务的受访者所占比例下降明显：高级管理层（从 2021 年的 9% 下降到 4%）和专业服务团队（从 2021 年的 4% 下降到 1%）。



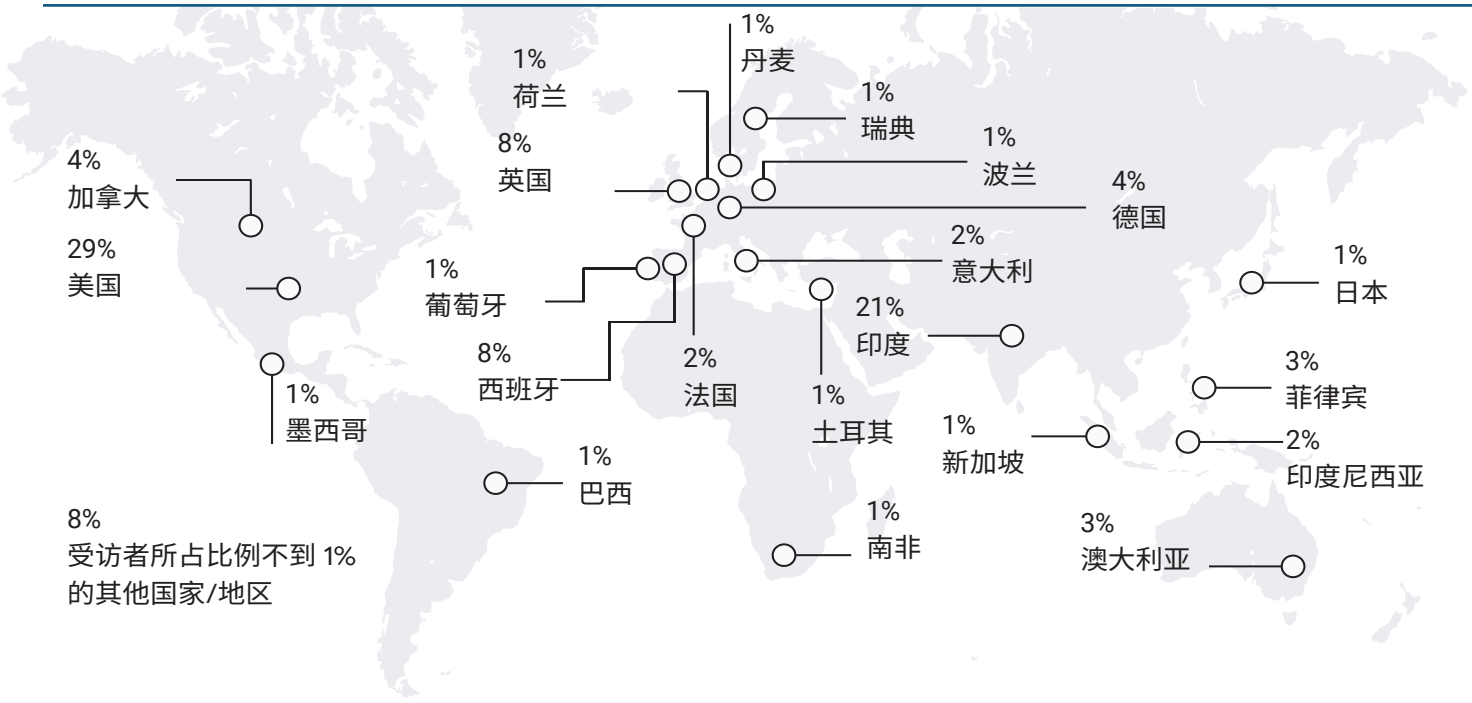
行业

与以往的《DevOps 现状》报告一样，我们发现来自科技行业的受访者人数最多，其次是金融服务行业、其他行业和零售行业。



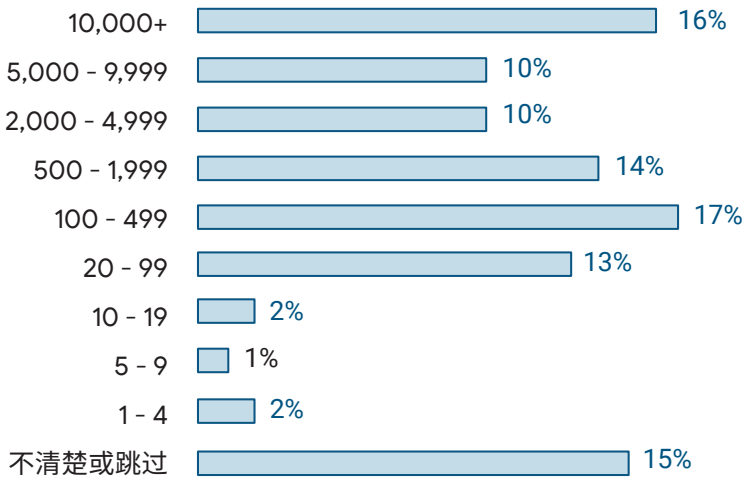
区域

今年，我们请受访者选择他们所在的国家/地区，而不是区域。区域通常代表的是某个大洲，这种分类方式似乎过于笼统，无法帮助我们了解受访者的组成结构。我们收到超过 70 个国家/地区的受访者提供的回复；89% 的受访者来自 22 个国家/地区。



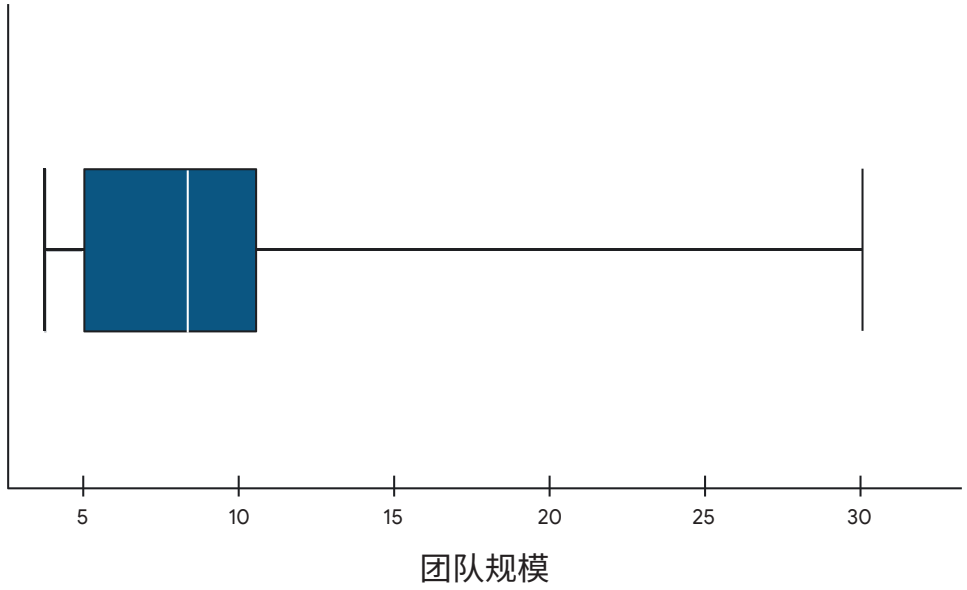
员工数

与以往的《DevOps 现状》问卷调查一致，受访者来自各种规模的组织。22% 的受访者来自员工人数超过 10,000 的公司，7% 的受访者来自员工人数介于 5,000 - 9,999 之间的公司，另有 15% 的受访者来自员工人数介于 2,000 - 4,999 之间的组织。此外，我们还看到，有相当一部分受访者 (13%) 来自员工人数介于 500 - 1,999 之间的组织，有 15% 的受访者来自员工人数介于 100 - 499 之间的组织，最后，有 15% 的受访者来自员工人数介于 20 - 99 之间的组织。今年，我们还提供了“不清楚”组织规模的选项；有 15% 的受访者表示不清楚或选择跳过此问题。



团队规模

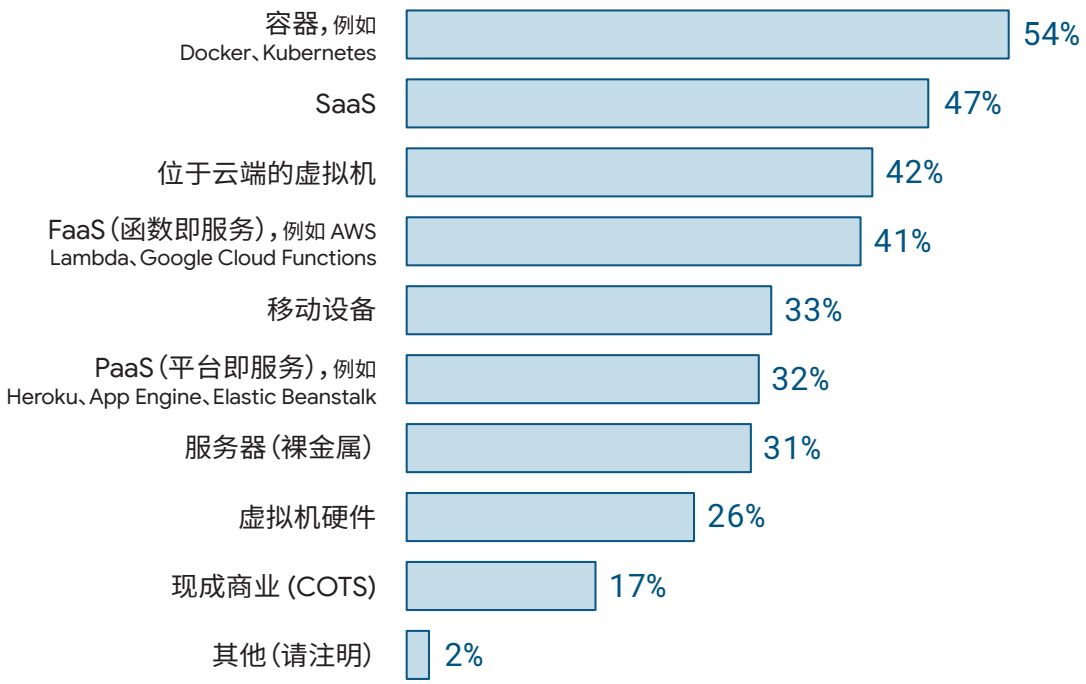
今年，我们请受访者指明其所在团队的大概人数。25% 的受访者来自成员不超过 5 人的团队，50% 的受访者来自成员不超过 8 人的团队，75% 的受访者来自成员不超过 12 人的团队。



部署目标

在 2021 年，我们首次决定了解受访者将开发的主要服务或应用部署在何处。令我们感到惊讶的是，排在第一位的部署目标是容器。今年的结果也一样，但其所占比例 (54%) 低于去年 (64%)。

此外，我们还添加了更多选项，希望为受访者提供更多方式来反映他们的部署位置。



07

最后总结

Derek DeBellis

每年制作这份报告时，我们都会尽量以严谨的方式说明实践和能力如何有助于提升重要的业务成效，例如组织绩效。我们评估了之前的报告中许多影响的可复制性，并扩展工作范围，探讨了 DevOps 领域内新出现的优先事项。今年，我们调整了问卷调查和分析方法，以深入探究安全性实践，并且改变了统计建模方法，以发掘特定影响的条件性或依赖性。此外，我们还探索了描述软件交付表现和运营绩效的新方法。

今年实际成文的论述在许多方面都与前几年类似：技术能力相辅相成，可提升绩效；使用云端这一做法本身就有许多好处；企业文化和工作灵活性有助于提升组织绩效；员工倦怠会阻碍组织达成目标。我们明确添加到模型中的互动分析可帮助我们了解 特定影响产生的条件。

例如，只有当运营绩效（可靠性）也较高时，软件交付表现才会对组织绩效产生积极影响，由此我们可以得出以下结论：必须同时兼顾这两者，才能推动组织蓬勃发展。此外，我们还专门规划了一个章节，介绍了意料之外的现象。

我们衷心感谢为今年的问卷调查做出贡献的所有人。希望我们的研究能够帮助您和您所在的组织建设更优秀的团队，构建更出色的软件，同时保持工作与生活的平衡。

08

致谢

今年的报告能顺利完成，离不开众多热心人士的无私贡献。我们的同事为帮助完成这一庞大计划付出了巨大的心血，调查问卷的问题设计、分析、撰写、编辑和报告设计只是其所做工作一部分。全体作者由衷感谢所有人对今年报告的贡献和指导。致谢对象名单按字母顺序列出。

Scott Aucoin
Alex Barrett
James Brookbank
Kim Castillo
Lolly Chessie
Jenna Dailey
Derek DeBellis
Rob Edwards
Dave Farley
Christopher Grant
Mahshad Haeri
Nathen Harvey
Damith Karunaratne
Todd Kulesza
Amanda Lewis
Ian Lewis
Eric Maxwell
John Speed Meyers
Steve McGhee
Jacinda Mein
Alison Milligan
Pablo Pérez Villanueva
Claire Peters
Connor Poske
Dave Stanke
Dustin Smith
Seth Vargo
Daniella Villalba
Brenna Washington
Kaiyuan “Frank” Xu
Nicola Yap

09

作者

Claire Peters

Claire Peters 是 Google 的用户体验研究员，她的研究涵盖 DORA 在应用环境下的各个方面和表现形式。她研究 Google 的云应用现代化改造计划 (CAMP)、由 DORA 驱动的客户互动度以及与 Four Keys 项目相关的工具，旨在帮助团队和个人在日常工作中更有效地应用 DORA 原则。Claire 还是 DORA 核心调研团队的成员，负责制作年度 DORA 调查问卷和《DevOps 现状》报告。她拥有哥本哈根大学应用文化分析硕士学位。

Dave Farley

Dave Farley 是 Continuous Delivery Ltd 的执行董事兼创始人，也是 YouTube 频道“Continuous Delivery”的创作者。Dave 是畅销书《Continuous Delivery》（持续交付）的联合作者，并且还是畅销书《Modern Software Engineering: Doing What Works to Build Better Software Faster》（现代软件工程：采用有效实践，加快构建更出色的软件）的作者。同时，他与人合著了《Reactive Manifesto》（反应式宣言），并凭借开源 LMAX Disruptor 项目荣获爱丁堡公爵奖。Dave 不仅是持续交付领域的先行者，还是 CD、DevOps、TDD 和软件设计方面的思想引领者和专家，在建设高绩效团队、打造成功的组织和开发出色软件方面拥有丰富的经验。如需详细了解 Dave，请访问他的 Twitter 主页、YouTube 频道、博客和网站。

Daniella Villalba

Daniella Villalba 是专门负责 DORA 项目的用户体验研究员。她专注于研究使开发者保有幸福感并提高效率的因素。在加入 Google 之前，Daniella 研究过冥想训练的益处、影响大学生经历的心理社会因素、目击者记忆和虚假供认。她拥有佛罗里达国际大学的实验心理学博士学位。

Dave Stanke

Dave Stanke 是 Google 开发技术推广部门的一名工程师，负责为客户提供有关采用 DevOps 和 SRE 的最佳实践建议。在他的职业生涯中，他担任过各种职务，包括初创公司首席技术官、产品经理、客服人员、软件开发者、系统管理员和平面设计师。他拥有哥伦比亚大学技术管理硕士学位。

Derek DeBellis

Derek DeBellis 是 Google 的定量用户体验研究员。在 Google，Derek 专门负责问卷调查研究、日志分析，以及找出产品开发核心概念的衡量方式。Derek 在近期发表的文章中探讨了人工-AI 交互、COVID-19 确诊对戒烟的影响、NLP 错误设计以及用户体验在隐私保护议题中所扮演的角色。

Eric Maxwell

Eric Maxwell 领导着 Google 的 DevOps 数字化转型实践，他为全球的顶级公司提供建议，帮助他们一步一步持续取得发展。在职业生涯的前半段，Eric 是一名一线工程师，他建立了自动化程序来处理所有工作，并设身处地为其他从业者着想。Eric 参与制定了 Google 的云应用现代化改造计划 (CAMP)，是 DORA 核心团队的成员，也是《DevOps Enterprise Guidebook》(DevOps 企业指南) 的作者。在加入 Google 之前，Eric 任职于 Chef Software 并与同事度过了一段愉快的合作时光。

John Speed Meyers

John Speed Meyers 是软件供应链安全领域的初创公司 Chainguard 的一名安全数据科学家。John 的研究项目涵盖的主题包括软件供应链安全性、开源软件安全性以及全球对中国军事力量不断增强的反应。John 之前曾就职于 In-Q-Tel、RAND Corporation 以及战略和预算评估中心。John 拥有兰德公司帕迪研究生院政策分析博士学位、普林斯顿大学公共与国际事务学院公共事务硕士学位 (MPA) 以及塔夫茨大学国际关系学士学位。

Kaiyuan “Frank” Xu

Kaiyuan “Frank” Xu 是 Google 的定量用户体验研究员。他负责分析日志和问卷调查数据，了解 Google Cloud 产品的使用模式和用户反馈，以便为开发者打造更加卓越的产品。在加入 Google 前，Kaiyuan 任职于 Microsoft，对 Azure 和 Power Platform 产品进行了数年的定性和定量用户调查。他拥有华盛顿大学以人为本的设计与工程学硕士学位。

Nathen Harvey

Nathen Harvey 是 Google 开发技术推广部门的一名工程师，致力于帮助团队充分发挥自身潜力，同时根据业务成效来调整技术。Nathen 曾有幸与一些优秀的团队和开源社区合作，帮助他们应用 DevOps 和 SRE 的原则和实践。Nathen 曾参与编辑 O'Reilly 2020 年出版的《97 Things Every Cloud Engineer Should Know》（所有云端工程师都应知道的 97 件事），贡献一己之力。

Todd Kulesza

Todd Kulesza 是一名 Google 用户体验研究员，负责研究软件工程师现今的工作方式，以及他们未来如何提升工作效率。他拥有俄勒冈州立大学计算机科学博士学位。

10

调研方法

研究设计

这项研究采用了基于理论的跨行业设计，称为**推断式预测**，这是当今商业和技术研究领域最常见的设计类型之一。如果无法或难以使用纯实验设计，便可以使用推断式预测设计。

目标人群和采样

此次问卷调查的目标人群是实际运用技术和参与转型或有密切关联的从业者和领导层，尤其是熟悉 DevOps 的人员。我们通过以下方式来宣传此次问卷调查：电子邮件名单、在线促销、在线样本组、社交媒体以及请用户在自己的人际网络中分享调查问卷（即滚雪球抽样）。

建立潜在构念

我们尽可能根据之前经过验证的构念来准备我们的假设和构念。我们基于理论、定义和专家意见建立全新的构念。然后，我们执行额外的步骤来阐明意图，以确保通过调查问卷收集的数据可靠且有效的可能性较高[1]。

计算低绩效者与高绩效者之间的差异

在“如何比较”部分，我们使用四项交付表现指标来比较低绩效者与高绩效者之间的差异。此方法非常简单，我们以部署频率为例。高绩效组会按需进行部署（即每天部署多次）。如果他们平均每天部署 4 次，则一年共部署 1,460 次 (4 * 365)。相比之下，低绩效者的部署频率介于每月 1 次到每 6 个月 1 次之间，平均每 3.5 个月部署 1 次，一年部署约 3.4 次 (12/3.5)。由此可以计算出高绩效者的部署频率是低绩效者的 417 倍 (1,460 / 3.4)。此方法同样适用于其他开发绩效指标。

[1] Churchill Jr, G. A. “A paradigm for developing better measures of marketing constructs”（改进营销构念衡量的范式），《Journal of Marketing Research》（营销研究杂志）16:1, (1979), 64–73。

统计分析方法

聚类分析

对于“如何比较”部分中所述的两种聚类解决方案，我们使用 Ward 的凝聚式层次聚类方法[2]来评估不同的聚类解决方案与数据的适配程度。

对于第一种聚类结果，我们基于有关部署频率、准备时间、恢复服务所需的时长和更改失败率的回复来查找组。今年，我们使用 30 种不同的指标来确定组的数量，在评估 14 种不同的层次聚类解决方案后找到了三个组。[3]

我们提出的第二种聚类分析方法与第一种相同，但部署在 不同的数据维度上。我们希望基于吞吐量（涵盖部署频率和准备时间）、运营绩效（可靠性）和稳定性（涵盖恢复服务所需的时长和更改失败率）查找常见回复模式（即组）。此外，我们还探索了不同的聚类算法，以了解结果对方法的敏感度。尽管没有既定的方式来量化我们所知的敏感度，但显现的组通常具有类似特征。

衡量模型

在进行分析之前，我们通过探索性因素分析和使用方差极大旋转的主成分分析来确定构念[4]。我们使用平均方差提取值 (AVE)、相关性、克隆巴赫系数[5]、rhoA[6]$_A$、异质-单质比率[5 7]和组合信度，确认收敛和发散有效性和可靠性的统计测试。

[2] Murtagh、Fionn 和 Pierre Legendre。“Ward’s hierarchical agglomerative clustering method: which algorithms implement Ward’s criterion?”（Ward 的凝聚式层次聚类方法：哪些算法实现了 Ward 的标准?）。《Journal of classification》（分类杂志）31.3（2014 年）：274-295。

[3] Charrad M.、Ghazzali N.、Boiteau V.、Niknafs A（2014 年）。“NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set”（NbClust：用于确定数据集中相关组数量的 R 包），《Journal of Statistical Software》（统计软件杂志），61(6)，1-36。网址：http://www.jstatsoft.org/v61/i06/

[4] Straub, D.、Boudreau, M. C. 和 Gefen, D.（2004 年）。IS 实证研究的验证指南。《Communications of the Association for Informa-tion systems》（信息系统协会通讯），13(1)，24。

[5] Nunnally, J.C. 心理测量理论。纽约：McGraw-Hill，1978 年

[6] Hair Jr、Joseph F. 等。“Partial least squares structural equation modeling (PLS-SEM) using R: A workbook”（使用 R 的偏最小二乘结构方程建模 [PLS-SEM]：手册）（2021 年）：197。

[7] Brown、Timothy A. 和 Michael T. Moore。“Confirmatory factor analysis”（验证性因素分析）。《Handbook of structural equation modeling》（结构方程建模手册）361（2012 年）：379。

结构方程建模

我们使用偏最小二乘 (PLS) 分析[8]来测试结构方程模型 (SEM)；偏最小二乘是一种基于相关性的结构方程建模方式。

第二个组模型分析

为了解组成员资格的影响因素，我们采用了多项逻辑回归[9]。之所以采用这种方法，是因为我们尝试影响组成员资格，而在此例中，这是具有超过两个层级的无序分类数据。为了解组成员资格影响的成效，我们使用了线性回归来分析每项成效（倦怠率、计划外工作量和组织绩效）。

[8] Hair Jr, J. F.、Hult, G. T. M.、Ringle, C. M. 和 Sarstedt, M. （2021 年）。《A primer on partial least squares structural equation modeling (PLS-SEM)》（偏最小二乘结构方程建模 [PLS-SEM] 入门）。Sage publications。

[9] Ripley、Brian、William Venables 和 Maintainer Brian Ripley。“Package ‘nnet’” （“nnet”软件包）。R 软件包版本 7.3-12（2016 年）：700。

11

补充阅读材料

加入 DORA 社区，讨论、学习和协作，以改进软件交付表现和运营绩效。
http://dora.community

详细了解 Four Keys 指标
https://goo.gle/four-keys

详细了解 DevOps 能力。
https://goo.gle/devops-capabilities

如需详细了解如何在组织内采用 DORA 实践，请参阅我们的《企业指南》
https://goo.gle/enterprise-guidebook

查找有关站点可靠性工程 (SRE) 的资源
https://sre.google
https://goo.gle/enterprise-roadmap-sre

执行 DevOps 快速检查
https://goo.gle/devops-quickcheck

浏览 DevOps 研究项目
https://goo.gle/devops-research

阅读 Google Cloud DevOps Awards 获奖者电子书，向采用 DORA 实践的其他公司学习经验
https://goo.gle/devops-awards

了解 Google Cloud 应用现代化改造计划 (CAMP)。
https://goo.gle/3daLa9s

利用数据和 DORA 指标革新技术流程
https://goo.gle/3Doh8Km

阅读白皮书：《The ROI of DevOps Transformation: How to quantify the impact of your modernization initiatives》（DevOps 转型的投资回报率：如何量化现代化改造计划的影响），作者：Forsgren, N.、Humble, J. 和 Kim, G.（2018 年）。
https://goo.gle/3qECIlh

阅读图书：《Accelerate: *The science behind devops: Building and scaling high performing technology organizations*》（加速：*DevOps* 背后的科学：建立和壮大高绩效技术组织）。IT Revolution。
https://itrevolution.com/book/accelerate

详细了解安全工件的供应链级别 (SLSA) 框架
https://slsa.dev

详细了解安全软件开发框架 (NIST SSDF)
https://goo.gle/3qBXLWk

详细了解 DevOps 文化：Westrum 组织文化
https://goo.gle/3xq7KBV

详细了解 Open Source Security Foundation
https://openssf.org/

详细了解 in-toto
https://in-toto.io/

详细了解 NTIA.gov 的软件物料清单
https://www.ntia.gov/SBOM

信息安全：联邦政府对 SolarWinds 和 Microsoft Exchange 事件的回应
https://www.gao.gov/products/gao-22-104746

详细了解在 CI/CD 流程中进行应用级安全扫描
https://go.dev/blog/survey2022-q2-results#security

最后但同样重要的是，查看往年的《DevOps 现状》报告。访问以下网址可找到所有报告：https://goo.gle/dora-sodrs：
《加速：2014 年 DevOps 现状》报告
《加速：2015 年 DevOps 现状》报告
《加速：2016 年 DevOps 现状》报告
《加速：2017 年 DevOps 现状》报告
《加速：2018 年 DevOps 现状》报告
《加速：2019 年 DevOps 现状》报告
《加速：2021 年 DevOps 现状》报告