

加快应用程序开发的七个DevOps建议

DevOps的优点是速度：更快的应用程序开发、更快的更新和持续开发，以及更快的交付，所有这些都会缩短系统开发生命周期。最近的研究表明，高绩效的DevOps团队从停机中恢复的速度加快了96倍，变更失败率降低了5倍，部署代码的频率提高了46倍。简而言之，成功的DevOps团队可以在几个小时而不是几周完成代码部署和恢复。

毫不奇怪，几乎所有规模、所有垂直市场和全球地理位置的组织都接受了DevOps。另一项调查发现，全球接受调查的组织中，近四分之三已经采用了DevOps。然而，所有这些有满腔热情和DevOps潜力都受到了现实的挑战的磨砺。这项调查还发现，消除流程瓶颈以加快发布速度，已跃居DevOps所面临挑战的首位。简化协作模式并选择合适的自动化工具套件，才能更好的迎接挑战。

本文列出了高效团队为最大化他们的DevOps成效和投资所遵循的七大DevOps建议。

1 DevOps文化很重要，尤其是从基层建立起来。

年度DevOps现状调查（现在已经第七年）的作者写道：“对我们来说，最重要的主题一直是增强团队力量，让他们把工作做到最好，克服开发团队和运维团队之间的文化差异。”² 对本次调查的历史数据进行的广泛分析表明，DevOps的成功往往是从小规模基层工作开始的。接下来，开发人员需要一个轻量级平台，以便他们可以安全地与其他团队分享早期成功经验和证实有效的实践。然后这种共享传播到单个部门内的多个团队，并最终传播到其他部门。换句话说，使用DevOps工作要循序渐进，过犹不及。这正是全球最大的商业软件制造商和全球物流领导者所遵循的做法。

2 从一开始就将安全性深深嵌入DevOps中。

这意味着在可能的最大限度内自动化安全性配置。随着DevOps组织从基层开始发展，安全策略成为了运维的一个固有部分，而不是IT部门为了证明自己符合审计要求而仓促加入的东西。作为全球领先的软件开发平台，GitHub提供符合《通用数据保护条例》的解决方案，对传输中的所有数据进行加密，并通过云安全联盟CSA-CAIQ评估支持行业领先的控制考虑事项。组织还可以选择使用自托管产品GitHub企业服务器在自己的基础设施上托管代码，GitHub企业服务器旨在支持严格的安全性、审计和合规性要求。该解决方案由现有的本地信息安全控制来管理，包括防火墙、虚拟专用网络、身份识别与访问管理和监控系统。这种本地解决方案有助于避免基于云的解决方案中常见的许多监管合规性问题。因此，关注重点可以从远离审计人员转移到保护组织及其客户的数据安全上。最重要的是，团队和开发人员可以更快地构建更安全的代码。

要实现高效的整体过程，DevOps组织和团队需要将关注重点从审计和合规性的细节转移到开发安全代码。

3 尽可能自动化。

关于DevOps自动化的讨论通常集中在系统配置、工作流和系统准备方面的自动化。这种基础设施自动化解决了一个与软件部署能力直接相关的常见难题，即保持开发人员的工作进度，而不是超前于运维能力。自动化的另一个关键功能是在开发的后期阶段创建更广泛的自助服务，显而易见，这会带来更高的效率。GitHub使用一系列模板，使用户能够自动化配置工作流，以便使项目板卡片的状态与相关议题和拉取卡片保持同步。用户可以根据触发事件自动化操作，从而消除管理项目板时一些耗时的手动任务。用户可以复制GitHub项目板，以便在以后类似的项目中重用其自定义内容。

4 利用开放平台提供透明性的好处。

协作是DevOps效率的主要驱动力，是开源的核心，在开源中，软件是在高度协作、责任共享和公开的过程中开发的。近年来，随着各种类型的组织积极地将开源构建到他们的DevOps方法中，全球开源收入急剧增长。基于Web的软件平台允许用户跟踪和跟随大量其他开发人员的操作，无论他们在哪里或为谁工作。开放式平台还为开发人员和团队提供在团队内部和外部的协作机会。这样，他们不仅可以利用开源代码，还可以利用安全漏洞更新来更快地构建更安全的代码。这种开放性将有可能实现最佳透明度，因而可以从根本上改善在高度复杂的软件开发业务中的协作和学习。

一项调查表明，在这样的社会发展环境中，用户从这种网络活动中获得了一系列丰富且互惠互利的联系和社会推断，比如在编辑代码时推断出其他人的技术目标和愿景，或者更准确地猜测哪些项目从长远来看最有可能成功。然后，用户将这些推断组合成更有效的策略，以协调工作并提升自己的技术技能。³

5 与尽可能广泛的工具集成。

在最新版本的企业DevOps部署年度调查中，有关实现更有效的DevOps流程所面临的两大挑战都是以人为导向的（消除流程瓶颈和简化开发人员协作）。但在这个列表中排在第三位的是选择和实施应用程序监控、管理和自动化工具。将DevOps工作与尽可能广泛的工具集成的能力是DevOps成功的基石。无论是促进持续集成，还是代码审查或代码编辑，很重要的一点是，开发人员都必须能够为开发过程的每个步骤找到合适的工具，然后立即开始使用它们。GitHub提供了数百种工具，这些工具能够帮助开发团队更好地沟通、自动化他们的工作以及开发更好的软件。有时候，特定的任务只需要定制开发的工具。GitHub使用户能够使用GitHub GraphQL API创建这样的定制工具，来更方便地访问数据，该API与公司用来构建GitHub的API相同。

GitHub是人们构建软件的方式。GitHub得到了超过4000万开发人员和财富500强中一半企业的信任，它可以帮助各种规模的DevOps团队安全进行协作，并更快地提供更好的客户体验。

6 利用内部开源(Innersource)最大限度地提高开源效率。

越来越多的组织正在接纳和使用内部开源，这是一种利用大型开源项目中最佳实践的开发方法。这类高度复杂的多开发团队工作需要数百名甚至数千名开发人员和团队之间的协作。因此，内部开源被证明是应对大型DevOps项目带来的诸多挑战的强大武器，也是通过代码复用提高效率的一个好方法。公司可以访问和利用内部开源防火墙后面的开源最佳实践。GitHub是世界上最大的开源社区。GitHub平台上的开发人员和团队可以出于自己选择的查看、修改和分发项目，这是由开源许可证实施的。GitHub提供了一系列广泛的自助指南，比如说，帮助您的项目寻找用户，建设友好社区，维护和管理，和开源度量。

7 促进持续的反馈、集成、变更管理和部署。

发现DevOps中的常规实践和行业标准是一回事。更重要的是将它们集成并深深嵌入到一个流程中，这个流程跨越了从开发到部署再到运维的数据和信息孤岛。为此，持续集成要建立在测试自动化的基础上，以确保在新代码被添加到应用程序中时，持续集成能够重点关注在集成新的或修改的代码时的应用层质量。变更管理可以让运维参与其中，提供有关哪些其他系统可能受到影响，以及变更可能在更广泛的层面上暴露哪些后果或机会的意见。增量式部署策略经常受制于运维限制性因素，这些运维限制性因素可能会限制或停止进行中的连续部署。这是人的问题，而不是技术问题。补救措施是让所有利益相关者参与应用程序开发、运维和支持。这种持续的部门内投入将极大地提高一个优秀的应用程序满足用户需求和期望的可能性。

GitHub是人们构建软件的方式。GitHub得到了超过4000万开发人员和财富500强中一半企业的信任，它可以帮助各种规模的DevOps团队安全进行协作，并更快地提供更好的客户体验。要开始免费试用或了解关于GitHub的本地和SaaS解决方案的更多信息，请访问 <https://github.com/business>。

1 《2017年DevOps现状报告》，Puppet，2017年。

2 《2018年DevOps现状报告：DevOps发展的实务指引》，Puppet博客，2018年9月。

3 《GitHub社交编码：开放软件存储库中的透明性和协作》，jsntsay.com，2012年。