

开发运维的成功之路

使用机器数据实现 CI/CD

数字经济正在以互联网速度运行。因此，企业必须以前所未有的速度交付和改进新的软件和服务，通过敏捷、迭代的软件开发和交付。通过频繁、较小的更新，开发人员能够快速尝试新功能，并限制破坏关键功能的风险。它们还能让应用程序保持最新，让客户保持最佳状态，并能对威胁和业务变化做出快速响应。

快速应用程序发布周期在数字企业中很常见，在数字企业中，组织正被云服务等技术重塑，这些技术包括软件定义的基础设施、移动应用程序和社交渠道。这些周期催生了新的架构、设计模式、应用程序堆栈和连续交付发布系列。

开发运维是一种基于协作文化的全生命周期方法，正在成为开发、交付和支持应用程序的主要方式。采用开发运维的企业发展更快，使他们能够在不断变化的数字世界中跟上客户的需求。

然而，开发运维爱好者的希望和梦想经常在组织现实的岩石上破灭。开发运维并不总是信守承诺，通常是因为它技术复杂，存在诸多繁文缛节。打造敏捷、响应迅速、数据驱动的文化和支持流程说起来容易做起来难。尽管在这个过程中有许多挑战，但是机器数据的收集和分析是实现卓越开发运维的常见障碍。

机器数据来自应用程序、云、基础设施、容器、移动应用程序、数据库、IoT、流媒体网络数据等等。机器数据不仅仅是日志 - 它包含了所有应用程序、系统、客户、API、工业系统等等的所有活动和性能的确切记录。

开发运维的现实

开发运维将敏捷开发方法引入到软件定义的基础设施和操作中。这是关于不断的迭代。代码被定义、测试、部署、监控和测量。当遇到问题时，代码会被修改，过程再次开始。周而复始。整个过程需要速度、敏捷性和组织心态，包括渐进的改进和快速失效，而不是大爆炸式的软件发布。

成功的开发运维反馈循环的关键是软件度量 - 对使用、性能和错误的测量和分析。在接收和报告任何类型的数据方面，分析平台具有灵活性，这使其非常适合于开发运维流程，该流程总是包含来自各种来源的工具。

复杂性是敏捷性的敌人。典型的开发运维流程包括八个主要步骤，每个阶段使用不同的工具（图 1）。由于这些工具通常是开源或点产品，它们之间很少或没有集成，造成了重叠和低效，影响了整个过程的可见性并阻碍了敏捷性。



图 1：典型开发运维流程的主要步骤

无限符号是开发运维阶段的常见描述，因为它很好地说明了这个过程的两个相互交织的阶段。左边是软件开发和测试周期（“开发”），右边是部署和操作周期（“运维”）。然而，周期中的每一步都代表不同的工具产品市场，在每个阶段通常使用十几个或更多的产品。

这种丰富的选择对开发人员和运营团队来说非常有用，因为每个团队都可以选择适合其需求和偏好的产品。但由此带来的麻烦对于项目经理、开发运维主管和企业高管来说却是一场噩梦。开发运维系统的混合，每个都孤立地消耗数据，掩盖了对整个过程的可见性。

在开发运维世界中盲目飞行意味着，无法判断“修复”是解决了根本问题，还是仅仅提供了一个临时补丁。如果团队对代码开发、发布和使用数据的整个周期缺乏全面的了解，他们就无法验证应用程序的质量、性能和安全性。例如，安全监控软件可以识别漏洞，但是如果有一种方法将问题跟踪回其源软件模块，问题就无法修复。

此外，当组织缺乏对开发运维度量的可见性时，他们无法实现所需的敏捷性来尝试新想法或迭代，以响应不断变化的业务条件。如果没有覆盖整个应用程序交付生命周期的数据反馈循环，组织就很难识别代码开发、测试、质量保证和生产中的错误。结果是质量差，这可能导致客户不满和流失。

如果开发运维构建管道脱节，在整个应用程序交付过程中，会导致多次、耗时的移交，这有引入冗余功能的风险，并会因为耗尽每个人资源的任务切换而降低效率。不连贯的交付工作流还会导致协作受损（团队使用不同的术语，对整个项目有不同的观点，从而造成跨部门的差距）。沟通和信息共享中的摩擦破坏了信任与合作，也违背了开发运维的核心前提。

没有平台来提供数据驱动的反馈，那么开发运维不太可能成功。组织若要避免这一陷阱，唯一方法是使用数据来衡量、报告和展示开发运维的成功，同时使用数据驱动的见解来随着时间的推移对其进行优化和改进。

实现开发运维的全部价值

对大多数高管来说，开发运维是一个时髦词，不熟悉通常会导致不切实际的期望。他们对它能实现的目标抱有很高的期望，但是因为对原则、过程和挑战缺乏正确的理解，往往会使他们感到失望。

没有平台来提供数据驱动的反馈，那么开发运维不太可能成功。组织若要避免这一陷阱，唯一方法是使用数据来衡量、报告和展示开发运维的成功，同时使用数据驱动的见解来随着时间的推移对其进行优化和改进。

实践中的开发运维

行动敏捷的公司能够适应客户需求。进行调整需要一个数据驱动的平台，该平台提供持续的反馈来帮助组织的所有领域：IT、应用程序开发人员、业务部门高管、安全团队、审计人员和其他人员。通过提供基于事实的见解和决策支持，开发运维数据平台使企业能够迅速采纳好的想法并消除坏的想法。

为了正确做到这一点，开发运维需要数据、工具和流程，为所有利益相关者提供对必要开发运维工作流程的持续见解，以便运行业务单元、应用程序开发和 IT 运营。机器数据为这些见解和决策提供了原材料，如果使用得当，会带来许多好处（表 1）。

表 1. 将机器数据用于开发运维分析的好处

好处	对业务的影响
更好的决策	IT 部门需要采取有组织的方式来解决和管理潜在漏洞，以及安全漏洞或攻击的后果，以便限制损害并缩短恢复时间和成本。
更容易访问数据	使用上下文监控用户活动对查明违规行为并发现滥用行为至关重要。对特权用户进行监控是合规性报告的共同要求。
增强安全性和合规性	威胁情报可以帮助 IT 部门识别异常活动，评估业务风险并优先处理响应。
延长正常运行时间	分析是从海量数据中获得见解的关键，机器学习可以自动执行此分析以识别隐藏的威胁。
加快开发时间	安全专业人员需要专门的工具来监控、分析和检测杀伤链中的威胁。

在实践中，若能成功地将开发运维与全面的数据驱动分析相结合，将获得三个关键好处：更快的应用程序交付速度、更好的应用质量和更好的业务质量。

使用数据分析更快地交付应用程序

要实现敏捷性和客户满意度，开发运维团队需要具备四项关键能力：

- **监控**：开发运维构建和交付工作流的每个组件都必须使用收集到主存储库中的数据进行检测，以便进一步分析。如果没有数据，就不可能理解和补救应用程序的交付问题。
- **迭代**：必须使用整个工具链中的相关数据来快速识别、分类和开发代码修复和改进，以提供更深入的应用程序见解。
- **合作**：若要实现快速交付，开发团队必须在同一页面上，使用相同的数据，并基于相同的度量采取行动。
- **优化**：开发运维经理必须不断努力通过做出基于事实的决策来改进流程。在对流程进行优化时，需要对以下问题作出数据驱动的回答，例如：
 - 我们的“想法转换为现金”的比率是多少，我们的端到端吞吐量转换为盈利能力的比率是多少？
 - 我们交付管道的每个阶段有多长？
 - 团队花多少时间编写、测试或审查代码？
 - 哪些开发团队效率最高？

数据驱动的开发运维平台的优势直接体现在底线上 - 通过更高的效率和开发人员生产率、更快的应用程序交付速度、更低的成本、更高的客户满意度和更高的收入来改善业务。

利用实时见解创建更好的应用程序

来自整个开发运维工具链的机器数据提供了信息，使得能够在开发和测试周期的早期对问题做出主动响应。有了机器数据，开发人员和运营团队就可以在客户和用户体验到功能崩溃和应用崩溃的挫败感之前发现并解决问题。这种类型的主动响应需要使用通用的数据平台，开发运维团队可以将该平台作为事实的单一版本，与源代码控制系统使开发人员能够整合和共享工作成果的方式相当。

对整个开发运维过程进行检测和分析可以展示实际性能、使用情况和错误数据，这些对于改进最终产品和整个过程至关重要。通用数据平台使信息能够跨工具和基础设施进行关联，并能够主动对问题发出警报。例如，将来自代码检查的数据与性能监控系统相关联，可以在用户提交错误报告之前揭示问题。

整合来自整个开发运维交付周期的数据需要一个平台，该平台能够从前面描述的八阶段周期中使用的各种工具中实时接收数据。开发工具在不断变化，数据也是如此。

数据是开发运维度量的原材料，但是它必须来自客观的度量标准，可以量化代码是否满足功能和操作规范以及质量 SLA。使用通用的数据存储库来分析整个开发运维流程有助于实现：

- 代码审查和解决时间的量化测量
- 用于 bug 分析和项目跟踪工具的单一数据存储库
- 跨开发团队和代码发布的一致的、可测量的和可跟踪的 bug 率基准
- 提高测试和软件质量保证指标的可见性，允许在产品发布前识别问题

- 将安全性更紧密地集成到开发过程中，或者越来越多的被称为**开发运维**中。通过在构建周期的早期收集应用程序和安全情报分析，开发团队可以通过及早发现和消除漏洞来交付更加安全和符合规范的代码。通用的开发运维数据平台意味着，安全审计数据可以在一个地方收集和轻松获得。借助来自应用程序构建管道的实时、数据驱动的意见，安全团队能够在部署应用程序之前进行沟通、发出警报并避免新的潜在威胁。

获得更好的业务成果

数据驱动的开发运维平台的优势直接体现在底线上 - 通过更高的效率和开发人员生产率、更快的应用程序交付速度、更低的成本、更高的客户满意度和更高的收入来改善业务。将综合数据分析平台（例如 **Splunk**）的使用从传统 IT 运营角色扩展到开发运维后，组织可以预期：

- 实时了解新应用程序版本的使用情况、性能、可靠性、错误和安全事件，而不像使用临时报告时那样延迟数小时或数天
- 大幅降低 MTTR，生产速度提高 70%，生产前问题解决速度提高 40%
- 通过自动化数据接收和分析提高效率。开发人员和运营团队可以专注于满足业务需求，而不是在构建和维护 监控工具上浪费时间

若要确保开发运维团队与业务保持一致，必须具备持续的交付过程，该过程需要频繁的发布，并与实际的业务结果进行衡量和关联。开发运维分析平台可以从云中或内部的各种来源获取与业务相关的数据。通过整合的通用数据索引和关联，不仅可

以了解关键功能的交付状态，而且有助于在开发运维活动和相关的客户和业务指标之间建立联系。

事实驱动的开发运维通过提供更好的性能和更有用的代码来改善客户体验。更好的代码意味着客户更满意，客户越满意会越忠诚。通过分析像应用程序使用和销售情况这样的指标，使用来自开发运维构建管道的数据意味着，客户满意度可以直接与代码发布联系在一起。与开发运维相关的其他常见业务指标包括：

- 客户注册率、下载率和收入变化率，用于寻找与产品发布相关的波峰和波谷
- 客户参与度和销售或购物车放弃情况的变化

分析量化并增强开发运维的优势

若要在数字世界取得成功，需要具备敏捷性。这通过高效的开发运维发布过程来实现，并且需要数据分析来充分利用开发运维程序。

开发运维从业者必须不断提高响应能力、协作能力、安全性和法规遵从性，所有这些都可以通过快速开发创新产品来直接提高商业信誉和客户满意度。然而，采用开发运维也带来了许多挑战，这些挑战通常是复杂过程和工具链的结果。企业高管、IT 领导者和开发团队不仅必须了解开发运维的好处，还必须了解如何成功实现这些好处。

开发运维成功的秘诀是量化验证。格言“如果你不能测量它，你就不能改进它”仍然适用，整合的数据分析平台为测量和评估提供了理想的系统。

了解与 [Splunk for DevOps](#) 有关的更多信息，或[免费试用](#) Splunk 软件。