DevOps 调研

摘要：随着软件产品和服务生命周期迭代间隔的不断缩短,把开发和运维割裂开的瀑布模型已无法满足快速交付的需求，开发端和运维端之间的信息鸿沟越来越大。DevOps 的出现有效解决了开发与运维团队之间的阻抗失衡，使交付变得更加快速，更加高效，并最终做到了持续交付，得到广泛应用。本文综述 DevOps 方法，对提出的背景，所力求解决的问题，主要开发过程，阶段划分以及关键技术方法进行概述。

关键词：DevOps;持续集成;持续部署;

DevOps（Development 和 Operations 的组合词）是一组过程、方法与系统的统称，用于促进开发（应用程序/软件工程）、技术运营和质量保障（QA）部门之间的沟通、协作与整合。它是一种重视“软件开发人员（Dev）”和“IT 运维技术人员（Ops）”之间沟通合作的文化、运动或惯例。[1]通过自动化“软件交付”和“架构变更”的流程，来使得构建、测试、发布软件能够更加地快捷、频繁和可靠。

（一）方法提出的背景

随着市场需求的快速变化，快速交付的能力成为企业的核心竞争力。代码的短迭代快速开发，增强了开发与运维团队之间更好沟通与协作的需求。但在 DevOps 方法提出之前，从业人员主要使用瀑布模型或敏捷开发模型进行软件项目开发。瀑布模型的线性和敏捷开发的跨功能性都无法确保快速、连续地交付无缺陷的软件应用程序。

为了按时交付软件产品和服务，开发和运维工作必须紧密合作。在 20090’Reilly 敏捷大会上被约翰·阿尔斯帕瓦(JohnAllspaw)和保罗·哈蒙德(PaulHammond)提出 DevOps，这个名称由此开始流行起来并引发极大关注。

新技术和新研发工程实践的成熟为 DevOps 方法的提出提供了基础。

（二）力求解决的问题

开发人员与运维人员在职责中天然的矛盾：开发人员喜欢频繁的发布新代码，运维人员却认为系统变更是充满风险的，要求稳定可靠。两者目标的不匹配，就造成了开发与运营部门的鸿沟，从而减慢了 IT 交付业务价值的速度[2]。

2.1 传统研发和运维及测试定位的问题

由于传统研发和运维及测试定位的问题，即便在同一个企业中，由于工作性质和考核的方式的不同,研发、运维和测试之间的部门墙对效率的提高有着很大阻碍。此时即便研发可以通过敏捷高效运作起来，但糟糕的交付过程依然会将业务交付整体拉回到低效的状态，甚至造成业务失败，而这也是 DevOps 所要解决的问题。

2.2 开发和生产环境不一致

随着敏捷原则和精益思想在软件开发实践的应用，解决了客户快速开发软件的需要，但是在互联网高速发展的过程中,软件开发成为业务价值流运营过程中的一环，开发团队阶段性完成软件开发工作后,并不能从总体上为业务立即带来任何价值，软件必须经过后续测试阶段，直到被部署到生产环境并真正带动业务运转起来后,整个软件研发工作才算交付了价值。[5]在很多采用了敏捷开发方法的组织中，新功能开发所需的时间已经从几年缩短至几个月，但是部署到生产环境仍然需要几周甚至数月，而且部署过程中还总是伴随着大量不可预知的状况。

这是由于软件开发环境与正式运行环境在配置、环境上存在差异，所以需要确保构建产物一直处在运行环境当中。

部署对系统的变更，质量很重要，保证上生产前通过全部测试。所以 DevOps 是软件开发、运维、质量保证之间的高度协同，从而在完成频繁部署的同时，提高生产环境的可靠性、稳定性和安全性。

（三）主要开发过程及其阶段划分

DevOps 开发流程包含:管理与计划,集成与测试,交付与部署,监控与运维,分析与计划，这是一个循环的过程。

3.1 管理与计划

功能负责人对设计过程进行讨论,并且制定详尽的计划,从实际角度出发,对项目开发的可行性进行研究。项目整体计划和任务情况,被创建在管理平台上,并且对所有人可见。

3.2 集成与测试

应开发实践应用方法,确保代码与产品结合。每一次代码的改动均需要在模拟环境中完成,并且对设计结果进行严格的自动化检测,以期确保相关服务和功能得到实现。

3.3 交付与部署

当模拟环境中的代码通过自动化测试,它们将自动被部署到真实的产品环境中(即生产环境)。由于部署脚本与测试用例都是在开发前有着充分的沟通,且部署和测试都是通过自动化实现,投产过程出问题的可能性将大大减少,且即使出现,大家也会在一起找到原因并解决。[3]

3.4 监测与运维

在此阶段，开发人员记录有关应用程序使用的数据，并持续监测着每个功能。

持续监测保持应用程序中服务的可用性。它还能确认重复出现的系统错误的威胁和根本原因。在这个阶段，安全问题可以得到解决，还能自动检测和修复缺陷。

持续运维使发布的应用程序和后续更新的过程自动化。

3.5 分析与计划

由于整个开发流程进度及具体任务对所有人可见,每人在每天登录管理平台都可以具体感受到目前功能的进度。因此每天的站立会中每人都可以提出对产品功能及整体计划的分析和建议。[3]

（四）关键技术和方法概述

Devops 关键技术和方法在 Devops 更快更简单地落地都起着至关重要的作用,帮助开发人员缩短软件开发交付时间，提高交付质量，简化工作。

常见的版本控制工具 GitHub、GitLab 等，持续集成工具 Jenkins 等，配置管理工具 Chef、Puppet 等，容器平台 Docker 等，监控工具 Nagios 等。

4.1Jenkins

持续集成(ContinousIntegration)是个简单重复劳动，由人来操作费时费力，使用自动化集成技术能够快速地构建代码,自动地进行测试，从而提高代码的效率和质量，节约大量的人力。比如开发人员提交代码到 Git，自动进行 Maven 构建，部署或是打入 Docker 容器后发布，这一系列自动化过程需要一名协调者，Jenkin 就是这名协调者。

Jenkins 是一个开源的持续集成软件，扩展性强，可以和大部分软件相结合，比如 Maven,Git,Docker。实现无须太多人工干预，减少重复工作，节省时间，保障在任意时间点都可以自动编译、部署、测试和发布软件。[4]

持续集成的目标是为了任何时候都能从 Jenkins 中获取最近构建成功的程序包，并可随时发布到服务器上。[4]如果能把程序包封装到 Docker 容器中，自动的发布到服务器上，将更大程度上提高交付效率。

4.2Docker

不同的应用会依赖一个不同的运行环境，比如 JDK 版本不同，有的开发语言也不一样，服务器环境也有限制。运维人员要为每一个应用都提供一个运行环境这是很麻烦，很耗费时间的事情。所以我们有必要将服务及其运行环境封装在 Docker 镜像中，需要发布服务只需根据镜像启动容器即可，交付方式由应有程序转为 Docker 镜像，更加简洁方便。

Docker 是一个开源的容器引擎,Docker 容器是与 DevOps 密切相关的关键技术。Jenkins 可以结合 Docker 实现将服务镜像自动化发布。[4]

（五）参考文献

[1]耿泉峰,李曦,葛维,葛云龙,卢潇潇.基于 DevOps 的软件开发管理模式[J].软件,2019,40(01):93-96.

[2]DevOps[M].机械工业出版社,荣国平,2017

[3]蔡建军,任女尔,魏金津.DevOps 在软件开发项目中的实践[J].电脑知识与技术,2019,15(29):242-244.

[4]法晓宇.基于 DevOps 的软件开发与项目管理[J].电子元器件与信息技术,2020,4(08):132-133.

[5]小特雷弗A.罗伯茨等.DevOps 实践[M].北京:机械工业出版社，2016:3-10.