

# DevOps 助力 IT 组织与软件研发效能的提升 ——以上海财经大学为例

黄 杰

(上海财经大学 信息化办公室,上海 200433)

[摘 要] 阐述了上海财经大学在信息化转型升级过程中遇到的困难和挑战。调查了 DevOps 发展的现状,以及 DevOps 在具体落地实施的建设思路以及对组织文化、组织结构的变化。最后结合上海财经大学一年多的 DevOps 应用实践,证明了其对 IT 组织与软件研发效能的提升。

[关键词] DevOps; 开发运维; 软件交付

doi: 10.3969/j.issn.1673-0194.2019.15.070

[中图分类号] TP315 G420 [文献标识码] A [文章编号] 1673-0194(2019)15-0157-04

## 1 引 言

学校信息化建设转型升级,不仅要快速实现需求,而且要快速发布上线,并且必须保证业务可靠、高效运行。为了满足这些要求,IT 组织需要强有力的流程、技术和人员作为保障。

(1)版本管理不一致,缺少规范和标准

实际应用开发中,项目的管理规范和管理标准是很重要的一个环节,如果没有良好的管理规范,会对项目的上线迭代周期

产生很大的影响。

(2)缺少有效的自动化构建流水线

在实际的应用上线周期中人工占绝大部分,不仅浪费大量项目人员的时间,并且在重复性的工作中,容易产生错误。

(3)无制品仓库,无法有效地进行发布回滚

在项目上线之后,版本回退机制是非常重要的,在碰到某个问题需要回退到某个版本的时候,版本介质存放仓库就会显得非常重要。

[收稿日期] 2019-04-25

查档要求等。

第三类,档案利用指标数据:接待人次、查档卷次、查档人次、打印复印张数、邮件办理人数、补人事档案人次等

有了明确的业务流程,将信息和数据有效地揉进了流程中,使数据和信息在流程中不断地运转。数据驱动流程,流程驱动管理,两者结合,使档案馆向档案利用业务的智能化前进了一步。

## 6 档案利用服务应用成效

(1)管理更加规范。通过管理流程的再造与重构,以信息化的方式将流程落地。

(2)业务更加精细。通过数据指标(KPI)驱动业务流程,持续优化档案利用服务工作的开展,以数据辅助决策。

(3)服务更加人性化。通过档案利用服务系统的实现,为用户提供便捷的业务申请与办理方式,给用户个性化的服务体验。

## 7 结 语

档案工作具有鲜明的政治属性和法律属性,是学校各项工作依法依规开展的重要依据,是推进依法治校工作的重要助手,是支撑学校各项事业健康发展的重要资源。档案馆牢固树立“为党管档、为国守史、为民服务”<sup>[7]</sup>的政治站位,通过数据驱动流程,流程驱动管理的基本思想,着力推进档案工作的规范化、精细化

和人性化,不断加强管理服务创新,更好地服务于广大师生和校友的需求,服务于学校各项事业发展的需要和学校中心工作的开展。

## 主要参考文献

- [1] 国家档案局.全国档案事业发展“十三五”规划纲要[J].中国档案, 2016(5):14-17.
- [2] 周玲.对档案信息化建设的思考[J].档案学通讯, 2003(6):22-25.
- [3] 陶水龙.大数据视野下档案信息化建设的新思考[J].档案学研究, 2017(3):95-101.
- [4] 李江峰.基于流程化管理的高校教学档案工作创新与实践[J].兰台世界, 2018(10).
- [5] 余宏光.关于大数据时代背景下的档案利用服务分析[J].黑龙江史志, 2015(3).
- [6] 刘勇强.关于历史档案散失的回溯与思考[J].上海档案, 2007(7):11-13.
- [7] 黄凤平.扎实开展“两学一做”学习教育,切实履行档案人“为党管档、为国守史、为民服务”的神圣职责[J].云南档案, 2016(4):7-11.

(4)缺少有效的自动化测试流水线,单元测试、接口测试、集成测试、性能测试

目前对交付应用以手动用例为主,主要依赖人工测试,通常测试人员未参与需求和设计评审,对业务缺乏了解,难以设计合理的测试场景,用例编写时间较长且难以复用。

针对每次全量交付编写新用例的同时,由于无法识别实际更新内容和影响范围,需要执行所有测试用例(包括组件自身和集成用例),用例执行时间长,导致测试效率难以提高。

(5)无法高效地进行代码质量分析

在项目实际开发当中,开发代码的质量是后期维护和功能扩展的关键因素,人为的代码质量检测不仅耗时耗力,还容易出现误判和少判的情况,很难保证高效。

(6)项目管理,集成、测试、交付、运维,各阶段分散,无法串联

目前上海财经大学拥有多套应用运行环境,环境与环境间逻辑上隔离,同一应用在开发环境、测试环境、预发环境和生产环境上架缺乏关联,导致所有环境的部署和上架,都需要运维人员执行部署。未建立以应用为单位的发布计划,每套环境上架时间和资源占用时间,缺乏参考依据,导致最终发布时间难以控制。

(7)业务繁多,要求快速上线,频繁的变更和交付快速性已成为常态

以往项目建设周期通常在三个月以上,而随着微应用建设的推进,应用版本的迭代周期会降低待1个月甚至2周,由于迭代周期变短,采用敏捷开发模式,上线次数会显著增加,测试团队和运维团队的工作压力明显增加,并且很难做到全面回归,导致应用迭代上线的压力非常大。

## 2 DevOps 发展现状

DevOps 作为一个术语,包含了一组概念,虽然这些概念并不全是新的,但 DevOps 已经演化成了一个在整个技术社区迅速蔓延的运动。Ops 是系统工程师,系统管理员,运维人员,发布工程师,DBA,网络工程师,安全专业人员以及其他各种有关的职务的通称。Dev 被用作开发人员的缩写,但实际上它更为广泛,意味着“所有参与开发产品的人”,包括产品,质量保证 QA 和其他相

关工作人员。DevOps 是在整个服务的生命周期中,运维和开发工程师共同参与的,从设计到开发过程到生产支持的完整实践。

人们谈论 DevOps 是“开发和运营协作”,或者将“将代码视为基础设施”,或是“使用自动化”、“使用看板”等许多看似松散相关的内容。所以一定程度上可以理解为 DevOps=人+流程+工具+文化。

乔玮等在《DevOps 发展现状及趋势研究》一文中指出:DevOps 提出将开发与运维结合,其持续部署、持续交付技术能够有效提高产品和服务交付能力,帮助企业提升效能<sup>[1]</sup>。牛晓玲等在《DevOps 发展现状研究》的论文中提到:DevOps 的目标是加速软件发布和部署流程,利用自动化运维工具降低系统出错的概率,并快速消除宕机和错误等的影响,提高企业对业务问题的敏捷性,降低 IT 成本<sup>[2]</sup>。DORA 发布的《2018 DevOps 现状报告》<sup>[3]</sup>,该报告来自全球范围内各行各业的 1800 多名调查者提交了问卷,调查内容涵盖了云基础设施、领导力与学习氛围、交付效能、数据库实践等等。报告显示,在 2014 年时,只有 16%的调查参与者表示自己在 DevOps 团队。而在 2018,这个数字已经增长到 27%。根据《2018 DevOps 现状报告》的软件交付效能基准,团队被划分为三种类型:高效能、中效能与低效能团队,对团队的评价取决于他们的总体产出。发布频率、变更响应时间、服务恢复时间,以及变更故障率等指标是划分的参考标准。2018 年,报告新定义了 DevOps 精英级团队,其中在高效能团队中的 7%可归为精英级团队。精英级执行团队在以下几个方面有着突出的表现:代码发布频率高 46 倍;代码提交至发布的速度快 2 555 倍,变更故障率少 7 倍,事故恢复时间快 2 604 倍。

所以,DevOps 能够实现交付标准化、自动化、智能化。通过工具链与持续集成、交付、反馈与优化进行端到端整合,完成无缝的跨团队、跨系统的高效协作。

## 3 DevOps 建设思路

### 3.1 持续交付全流程管理

将 DevOps 生态工具按照持续交付流程与 DaoCloud Services 集成,实现持续集成、持续测试、持续部署、持续运维、持续反馈、持续改进的全流程管理(如图 1 所示)。

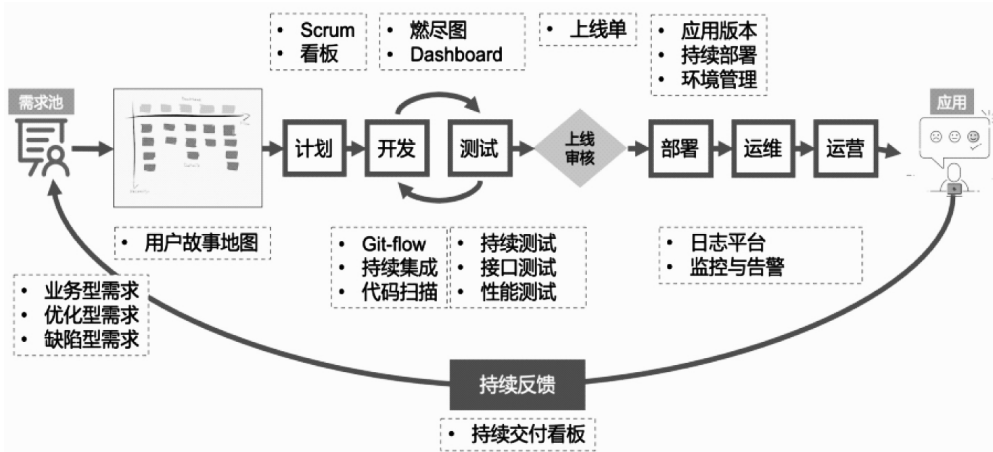


图 1 持续交付全流程管理

3.2 支持多种模式流水线触发,多种类型流水线

流水线支持代码源变更触发、定时触发,上游流水线触发,实例部署变更触发,满足多种条件下执行流水线的需求,同时支持开发类和测试类流水线,满足不同用户的使用场景。

3.3 多租户,多应用,多项目,多角色

多研发团队、多服务应用、多项目、多角色管理,更灵活的支撑学校传统矩阵式项目管理模式向 DevOps 跨职能管理模式转型。

3.4 多集群/环境统一管理

对接物理机、虚拟化、K8s、DCE 等多种类型的集群,并打上相应环境的标签,实现代码从开发环境、测试环境、预生产环境到生产环境的交付。

3.5 可扩展的流水线

可视化配置,根据校方需求自行定义,结合业界最佳实践,提供自动构建、单元测试、静态代码扫描、代码安全扫描、镜像推送等一站式服务,助力 DevOps 快速落地。

3.6 应用上线流程管理

通过应用的版本管理,记录每一次变更的相关制品版本和

Release Notes,同时通过发布单,配合学校内部上线流程,保障应用每一次上线有章可循。

3.7 DevOps 生态工具集成

集成了 Bitbucket,SVN,JIRA,Crowd,Jenkins,SonarQube,Nexus,Robot Framework,JMeter,Maven,Gradle,Splunk,上财认认等工具,有效实现 DevOps 持续交付全流程。

3.8 可视化看板,持续改进交付流程

提供租户下,跨应用、多维度、多层次的 KPI 统计,实现应用发布和流水线执行情况可视化管理,持续改进交付流程。

3.9 DevOps 生态工具标准化配置

集成 Bitbucket,SVN,JIRA,Crowd,Jenkins,SonarQube,Nexus,Robot Framework,JMeter,Maven,Gradle,Splunk,上财认认等工具的标准化配置,最大化工具的价值。

4 DevOps 应用成效与价值

4.1 DevOps 应用成效

整个 DevOps 平台包括四个部分。分别是项目管理、持续集成与持续测试、持续交付、持续运维,其平台全景如图 2 所示。

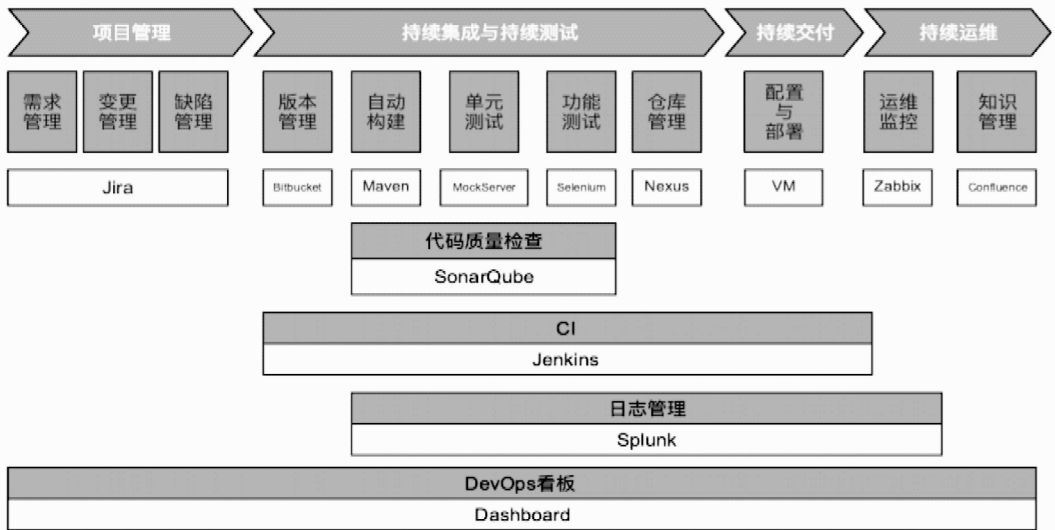


图 2 DevOps 平台全景

项目管理采用 Jira 工具来进行需求管理、变更管理以及缺陷管理,适用敏捷的项目 kanban 进行项目开发协作。

持续集成与持续测试有版本管理驱动,自动化的编译与构建,代码质量扫描、自动化的单元测试、功能测试,生成制品包上传到制品仓库。

持续部署集成了 Jenkins 软件发布的能力,将制品高效的发布到三套运行环境中(集成测试环境、预发布环境、生产环境)。

持续运维与反馈,使用 Zabbix 来监控基础架构与业务的健康度,使用 Splunk 收集业务日志,配合服务台和服务知识库进行知识管理,给用户持续的优良体验。

DevOps 看板给领导层与各类管理层查看项目情况提供了实时的 KPI 数据,比如项目的故事点数量,已投入的人天,团队

速率,需求数量,按时完成率,构建成功率,部署成功率,构建时长,部署时长,单元测试覆盖率,缺陷密度,发现 bug 数量,技术债天数,遗漏 bug 数量,故事燃尽图,bug 燃尽图,版本迭代信息等。

4.2 DevOps 价值

4.2.1 组织文化

交付是每个人的事,经过 DevOps 项目的建设,强调团队之间的沟通、协作与尊重,在组织与文化方面,让所有人就目标达成一致,一切都以更快更好地交付有价值的服务为目标。在 DevOps 模式下,开发、测试、技术运营三种角色需要像一支团队一样紧密地协作。每个迭代都是一个完整的交付周期,每次完整的开发、测试、受控、生产才算一次发布。

# 校园网由 IPv4 到 IPv6 的过渡技术浅析

常伟鹏, 袁 泉

(中国药科大学 图书与信息中心, 南京 211198)

[摘 要] IPv6 作为下一代互联网协议, 具有地址空间大、数据包转发效率高、可靠性高、安全性高等特点。常见的 IPv4 到 IPv6 的过渡技术有双栈、隧道、协议转换等类型。三种过渡技术各有特点, 分别适用于校园网的不同场景。校园网的 IPv6 过渡要依据网络建设阶段和校园网发展水平, 组合选择多种技术来实现校园网的安全、平稳过渡。

[关键词] 校园网; 过渡技术; IPv6 过渡

doi: 10.3969/j.issn.1673-0194.2019.15.071

[中图分类号] TP393.1 [文献标识码] A [文章编号] 1673-0194(2019)15-0160-04

## 1 引 言

随着物联网、5G 通信、大数据等新一代信息技术的蓬勃发展, 人们对下一代互联网的网络规模和服务质量都提出了一系列相当高的要求。据预测, 到 2025 年物联网的连接数将超过 270

亿<sup>[1]</sup>, 而全球可供分配的 IPv4 地址已于 2011 年 2 月由互联网地址分配机构 (IANA) 分配完毕<sup>[2]</sup>, 从现实来看, 只有 IPv6 才能满足新技术对网络互联的要求。在现阶段开展互联网协议从 IPv4 向 IPv6 过渡, 推进 IPv6 在互联网中的应用与普及, 在当前的形势下显得尤为必要和迫切了。

## 2 IPv6 及其特点

IPv6 即互联网协议第六版 (Internet Protocol Version 6), 是由

[收稿日期] 2019-04-25

[基金项目] 2019 年中央高校基本科研业务费专项资金资助项目 (2632019PY16)。

### 4.2.2 高度自动化

自动化是一切 DevOps 项目的核心思维, 开发、测试、运维需要一起优化交付过程, 自动化构建、自动化部署、自动化测试、自动化运维。从虚拟机创建、环境安装 (EOS、Tomcat、JDK)、应用构建打包、应用发布、代码质量扫描到后续运维, 一切皆自动。

### 4.2.3 持续交付能力

自动化是持续交付的基础能力, 目标是加速代码提交到部署上线的过程, 主要包括如下几方面的自动化: 构建、环境管理、应用部署、测试。持续交付的源头是配置管理, 源代码、依赖、应用、环境都应该实现配置管理。不仅仅使用了 Bitbucket 版本控制工具, 像更加反向倒逼了 Jira 项目管理的规范性, 自建学校独立的依赖库 Nexus、创建了应用制品仓库、统一了 Maven 构建环境, 使得试点项目能够根据版本定位 Jira 问题单、关联源代码、准确知道线上环境版本号, 统一的版本库 统一的构建环境 统一的制品仓库 统一的依赖管理 统一的代码质量管理 统一的虚拟机资源管理 统一的工具链自动化流水线。最终实现了 Code as a Service。

### 4.2.4 规范化

制定和落实代码管理规范: 目录、版本、分支、提交等 Jira 项目管理驱动整个持续交付能力的运行, 倒逼项目管理更加规范有序。

流程规范的改进, 在原有的 ITIL 基础上, 增加轻量化的

ITSM 实践, 明确项目、版本、组件、代码库等维护流程和相应规范, 支撑自动化流水线的运行, 并使系统都可以快速升级及回滚。

直观的度量指标, 对开发运维过程进行度量, 内建质量。包括构建通过率、代码质量扫描、发布成功率等, 对每一个环节“完成”的标准进行量化, 减少后续环节产生问题的概率。

## 5 结 语

整个平台还有很多需要增强和优化的地方: 自动化测试方面还需要提升, 更高的单元测试覆盖率, 补充自动化的 UI 测试; 应用安全方面需要补充, 代码的安全白盒扫描, 应用的安全风险评估等。

通过一年多 DevOps 平台的建设, 上海财经大学初步实现交付标准化、自动化、智能化。通过工具链与持续集成、交付、反馈与优化进行端到端整合, 完成无缝的跨团队、跨系统协作。

### 主要参考文献

- [1] 乔玮, 赵文瑞. DevOps 发展现状及趋势研究[J]. 数字技术与应用, 2018(4): 40.
- [2] 牛晓玲, 吴蕾. DevOps 发展现状研究[J]. 电信网技术, 2017(10): 48-51.