ISSN 1673-9418 CODEN JKYTA8
Journal of Frontiers of Computer Science and Technology
1673-9418/2000/00(00)-0000-00
doi: 10.3778/j.issn.1673-9418.

E-mail: fcst@vip.163.com
http://www.ceaj.org
Tel: +86-10-89056056

DevOps 自动化支持工具及实践现状经验研究

李杉杉, 李铮, 张贺+, 刘博涵, 荣国平

南京大学 软件学院, 江苏省南京市 210093

Empirical Study on the Practical Status of DevOps Automation and Tools

LI Shanshan, LI Zheng, ZHANG He+, LIU Bohan, RONG Guoping

School of Software, Nanjing University, Nanjing 210093, China
+ Corresponding author: E-mail: hezhang@nju.edu.cn

LI Shanshan, LI Zheng, ZHANG He, et al. Empirical Study on the Practical Status of DevOps Automation and Tools, 2000, 0(0): 1-000.

Abstract: Emerging from the agile culture, DevOps extremely emphasizes automation and heavily relies on tools in practice. Given the rapidly increasing number and diversity of the tools for DevOps, good understanding of the-state-of-art of DevOps-friendly tools will help to improve the automation practice of DevOps. This paper tries to find the answers from two aspects: What supporting tools are available for implementing the automation for DevOps? What issues and problems are associated with the current automation and tooling of DevOps? To collect as much evidence as possible, we employed a mixed empirical study to examine and synthesize academic publications and grey literature respectively, supplemented by a questionnaire-based survey with DevOps practitioners. This paper reports the practical status of DevOps automation and tooling that is worth sharing in the DevOps community together with our concerns for improvement.

Key words: DevOps; automation; tool; empirical study

摘 要: 作为敏捷文化的延伸，DevOps 强调自动化及工具的支持。鉴于 DevOps 的支持工具数量庞大和种类繁多，理解和认识其现状有助于提高 DevOps 自动化实践水平。本文从两个方面来探究 DevOps 自动化实践与工具的现状：DevOps 自动化支持工具有哪些；与自动化支持工具及 DevOps 自动化关键实践相关的问题与挑战是什么。采用了一种混合的经验研究方法，从学术界和工业界尽可能全面地收集证据，数据来源分别是学术文献、灰色文献，并通过调查问卷作为证据的补充和支撑。经过对数据的整合分析，报告了 DevOps 实践及工具的现状及挑战，同时重点揭示了 DevOps 三个关键实践、自动化软件交付过程相关的一系列问题

与挑战，最后针对问题与挑战提出了可能的解决策略。
关键词: DevOps; 自动化;工具；经验研究
文献标志码：A 中图分类号：TP311

1 引言

面对多种多样的用户需求和激烈的市场竞争，如何应对软件行业发展中的挑战、快速实现软件的商业价值是软件企业组织的共同目标。目前该目标的实现通常依赖于软件开发实践中普遍使用的敏捷方法和精益原则，典型的方法如 Scrum 和极限编程 (Extreme Programming, XP)。

DevOps 是一种文化，业界对于该文化的范围尤其是自动化实践有着不同的理解与解释[3-6]。它通过将敏捷文化从开发无缝延伸到运维阶段，并打破传统软件过程中这二者之间的壁垒，简化软件生命周期的各个环节，促进软件的健康、快速、持续交付。自动化在这场文化运动中起着至关重要的作用，因为“短周期内的高质量交付离不开高度的自动化”[1]，同时自动化也是快速获取反馈的关键[2]。

目前 DevOps 的发展尚不成熟，成功的 DevOps 实践需要建立在对 DevOps 充分认识与理解的基础之上，而这一基础在现阶段是非常薄弱的。由于在工业实践中，工具是最终实现 DevOps 自动化甚至一定程度上可视化的重要手段[7]。故为了帮助实践者及研究人员更好地理解 DevOps 自动化实践的现状，推动 DevOps 实践的进一步发展，本文选择部分自动化实践以及相关的 DevOps 自动化工具作为研究对象，探究两个研究问题的答案，即 DevOps 自动化实践都有哪些相关的支持工具、实践过程中存在什么样的挑战。

相对于工业实践，DevOps 新兴文化在学术领域中存在滞后性，故本研究采用了一种混合的经验研究方法来保证所收集证据的全面性。首先采用传统的系统化文献评价方法 (Systematic Literature Review, SLR) 从经过同行评审的学术文献中收集与研究问题相关的证据，继而借助灰色文献评价方法 (Multivocal Literature Review, MLR) 来补充工业实践中的数据，在此基础上，还针对研究问题进行了调研 (Survey)，这为本文的研究提供了更加全面客观的证据支撑。

本文的贡献主要有三点：一是全面探究了 DevOps 自动化相关工具的现状，并详细探讨实践过程中的问题与挑战；二是针对 DevOps 实践的问题与挑战，提出了可能的解决策略；三是通过混合的经验研究方法的使用及结果的分析，侧面说明了在软件工程尤其是经验软件工程领域，灰色文献评价方法对于保证证据全面性有重要的意义。

本文剩余部分的的结构组织如下：第 2 节简要概述 DevOps 背景及相关的研究工作；第 3 节从研究问题、研究方法两个角度介绍研究方案；第 4 节对研究结果进行详细分析；第 5 节针对第 4 节的研究发现讨论可能的解决策略，第 6 节总结本文工作。

2 背景及相关工作

DevOps 的发展可以追溯到敏捷文化的产生，为了应对快速且不可预测的业务需求变动，适应激烈的市场竞争，许多组织开始将敏捷方法 (如 Scrum，XP，DSDM，Lean，Crystal 等) 应用于软件开发实践中[8][9][10]。敏捷的宗旨是快速迭代，它能够加快发展进程 (包括需求分析，设计，编码和测试)，通过持续开发与反馈的迭代过程，使企业及时响应利益相关者的动态需求[11][12]。

敏捷方法只关注开发过程的迭代，而忽略了整个软件生命周期中的其他阶段。传统软件企业中普遍存在开发和运维团队之间的信息鸿沟问题，开发团队希望快速交付，而运维团队则要求更高的质量，这种目标差异导致软件交付过程在开发和运维之间壁垒高筑[13]，严重影响软件产品高质量快速交付的业务目标实现[14]。

继瀑布开发、敏捷开发之后，近年来，DevOps 文化成为又一新兴的软件开发理念和愿景。DevOps 是敏捷和精益原则在整个软件交付过程上的延伸[12]，换句话说，它旨在通过一系列文化及技术手段 (尤其是自动化 IT 工具链的打通) 打破开发和运维团队之间的壁垒，改善团队之间的协作关系，实现更加频繁快速、可靠的软件产品交付[15]。目前的 DevOps 已经不单纯是一个口号，很多知名企业已经着手实践 DevOps 并从中获益，如 Facebook, Yahoo 和 Netflix 等。Puppet Labs 在“2016 年 DevOps 报告”[16]中指出，采用 DevOps 的实践者比同行的软件产品部署频率要高 20 倍，错误恢复时间缩短了 24 倍，而产品更

新后的失败率降低了三倍。

DevOps 概念早在 2009 年就被提出来，伴随着越来越多的新技术（如微服务结构、容器）和自动化工具的普及，近两年才逐渐受到企业重视并开始被广泛实践[17]。但目前学术界中对于 DevOps 自动化工具的研究仍然很少，且都存在一定的不足。Akshaya 等人根据开发和运维涉及的阶段划分（构建、测试、监控、日志等），给出了一系列 DevOps 相关工具的简单介绍[18]。 Vaasanthi 等人仅仅对比研究了 DevOps 自动化构建工具。在这些针对 DevOps 某个或多个阶段工具的研究中，都仅仅是从功能和特点上对工具进行简单介绍[19]，缺乏深入的分析。工业界中，XebiaLabs 给出了 DevOps 相关工具的元素周期表[20]，包含一百多种不同类型的工具，但并没有针对工具的类型换分和工具之间关系等问题给出详细介绍。

相比之下，本文从多种数据来源全面探究了 DevOps 自动化的支持工具，以及实践中相关的问题与挑战，帮助更好地理解 DevOps 自动化实践的现状，并提出可能的解决方案来应对挑战。

3 研究方案

3.1 研究问题

DevOps 倡导软件过程中最大限度的自动化，通过自动化工具和技术实践等重塑软件工程的多个方面，尤其是软件的开发和运维。为了描述 DevOps 相关自动化支持工具的现状，更好地理解并促进基于 DevOps 的软件过程自动化发展，本文提出以下研究问题：

研究问题一：目前 DevOps 自动化实践中有哪些支持工具？

该问题旨在收集并形成支持 DevOps 自动化实践的工具集合，以限定后续研究的范围。为了回答该问题，本研究从学术文献、灰色文献以及调查问卷等多个来源进行证据的收集，在这些数据源中搜索、筛选并统计与 DevOps 自动化相关的工具。

研究问题二：与 DevOps 自动化工具相关的问题与挑战是什么？

该问题是为了全面地抽取证据中所体现的 DevOps 工具相关的问题与挑战，并进行整合分析。

研究问题三：与 DevOps 自动化实践相关的问题与挑战是什么？

该问题的目的是全面地抽取证据中所体现的 DevOps 实践及相关的问题与挑战，并进行整合分析。

3.2 经验研究方法

前面提到，DevOps 是一种诞生于技术社区的新兴文化，近年，逐渐吸引了学术界的关注，但相关研究尚且不足。目前，DevOps 的推广以及技术的发展，学术界还明显滞后于工业界。为了从不同视角更全面和客观的研究 DevOps，本文同时采用了系统化评价 (Systematic Literature Review, SLR) 和灰色文献评价 (Multivocal Literature Review, MLR) 两种研究方法，分别从学术界和工业界收集和分析数据。此外，还利用了调研（Survey）的方法作为第三种数据源来为本研究提供更多角度的证据支撑。本文研究方法如图 1 所示。

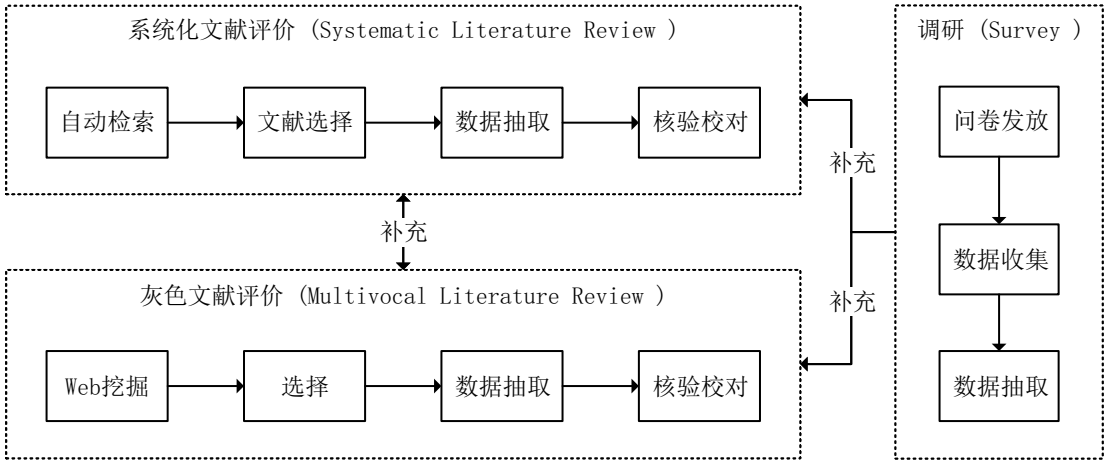


Fig. 1 Research methodology of this study.

图 1 本文研究方法

3.2.1 系统化文献评价

系统化文献评价由 Kitchenham 等人提出，是实证软件工程中最主要的研究方法[21]之一。本研究采用轻量级的系统化文献评价方法[22]，从学术文献中收集与 DevOps 自动化实践相关的证据。

具体来讲，首先使用自动搜索策略在软件工程领域四个主要的电子文献库 (IEEE Xplore, ACM Digital Library, SpringerLink 和 ScienceDirect) 中搜索同时包含 DevOps 和 tool 关键字的文献。为了保证搜索结果的全面性，本研究使用定义的搜索字符串为“DevOps AND tool*”进行全文搜索，其中包含 tool 的同义词 tooling, toolset 和 toolchain 文献结果都符合检索的要求。继而对检索到的文献进行筛选，该阶段中有 117 篇文献被确定为相关文献，分别包括 IEEE Xplore 的 63 篇， ACM Digital Library 的 29 篇， SpringerLink 的 15 篇以及 ScienceDirect 的 10 篇。然后由三名研究人员对所筛选的相关文献进行数据抽取，抽取的内容包括文献基本信息（论文名称、年份）、工具名称、工具特征、详细程度（仅提及、详细介绍）、使用情况（使用、未使用）、问题与挑战。数据抽取过程中，每人负责一个文献库，SpringerLink 和 ScienceDirect 由于相关文献相对较少，由一名研究人员同时负责。为了尽可能减少误差，每人评价的论文都需要由另一名研究人员独立检查。当遇到分歧时要召开会议进行讨论，直到达成一致意见。该阶段结束后，会初步生成学术文献中 DevOps 实践的支持工具集合。

3.2.2 灰色文献评价

学术文献中的信息通常会滞后于灰色文献[23]，尤其是面对从技术社区发展并活跃起来的 DevOps，传统的系统化文献评价方法会因为局限于从学术领域发表的文献中客观地收集、整合分析证据，而忽略工业实践领域关于 DevOps 的声音，继而可能会导致大量未在学术领域应用的 DevOps 工具无法被识别。

基于以上原因，除了系统化文献评价方法之外，本研究还采用了灰色文献评价方法[23]来收集证据，补充对 DevOps 自动化实践的理解。由于灰色文献的数量级过于庞大，为使得研究方法可控，本研究选择该领域典型的企业报告作为灰色文献的主要数据来源。

除此之外，本研究还以 Stack Overflow[1]论坛作为辅助的灰色文献数据源。作为一种基于社区的问答网站（Community based Question Answering，CQA），Stack Overflow 是目前最流行的软件开发者 CQA 网站，其主题主要与软件开发相关，除了其目标人群程序员外，Stack Overflow 还吸引了大批的研究者[24]。Stack Overflow 目前已经成为了一种重要的数据源，近年来在软件工程领域涌现出了一大批挖掘 Stack Overflow 的研究[25-28]。本研究借助 Web 挖掘技术在该论坛上搜索并自动筛选出包含 DevOps 和 tool 及其变型关键词的问题帖，对检索的 1832 篇问题帖中进行人工识别并选择，其中 879 篇讨论了具体的 DevOps 工具。所选择的问题帖平均分配给三名科研人员进行数据抽取，抽取的的内容包括问题帖基本信息（链接、问题标题、问题内容）、回答基本信息（点赞数、回答内容）、工具信息（工具名称、功能、优缺点、与其他工具之间的关系）等。其中抽取是点赞数是为了以计分的方式对工具相关信息进行质量评估。与系统化文献评价方法类似，每人负责的问题帖必须由另一名科研人员进行核对，检查抽取的信息是否有误，在该阶段定期同经验更为丰富的研究人员召开会议，消除数据抽取过程中产生的分歧。最后将灰色文献评价方法产生的 DevOps 工具集合与系统化文献评价方法的结果进行比对和合并，生成新的 DevOps 工具集合。

3.2.3 调研

基于系统化文献评价方法和灰色文献评价方法收集的证据，本研究对 DevOps 自动化及工具有了相对全面的理解认识。为了进一步验证本研究的结果，相关研究人员于 2017 年 4 月参加了一场包含 DevOps 主题的大型工业界会议——软件开发者大会[2]，参会人数 1500+，包括 DevOps 主题论坛的 200 多人次。借助此次与工业实践者面对面交流的机会，研究人员开展了一次关于 DevOps 经验及自动化工具使用的调研，调研采用的是调查问卷的形式。会议过程中总共发放纸质调查问卷 200 份，有效回答问卷 121 份 (召回率 60.5%) 。所设计的问卷包括六个与 DevOps 自动化及工具相关的开放式问题，以

[1] https://stackoverflow.com
[2] http://www.njsd-china.org

及三个基本信息调查类型的封闭式问题[3]。调研的结果作为前两种方法结果的支撑和补充，帮助本研究更好地理解工业界 DevOps 自动化及工具使用的现状。

除了识别并统计 DevOps 自动化实践的支持工具，本文还重点关注并提取了学术、灰色文献以及调研中所暴露出的问题与挑战。下一节将会对研究结果做详细分析。

4 结果分析

4.1 针对研究问题一

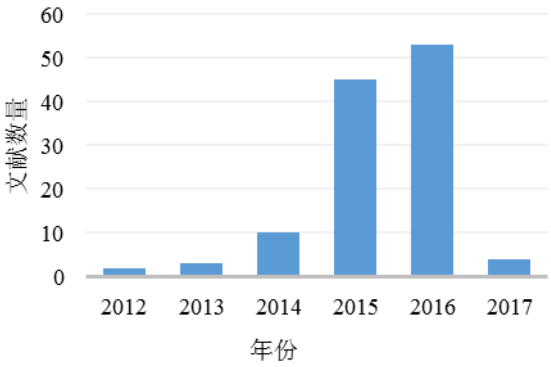


Fig.2 Distribution of the selected academic publications over years.

图 2 学术文献历年分布情况

本研究通过系统化文献评价方法从学术文献中识别出 101 个 DevOps 自动化工具，涉及到编码、配置管理、构建、测试和部署等不同类型；而通过灰色文献评价方法从 Stack Overflow 论坛中识别出 178 个与 DevOps 相关的工具，类型相似。后者相对于前者增长近 80%，一方面反映了 DevOps 相关工具的数量之大，另一方面说明本研究中灰色文献评价方法的使用具备一定的必要性和价值。

图 2 展示的是所筛选的文献 (2012 年至 2017 年) 分布情况。从图中结果可以出，DevOps 从 2014 年后开始广泛流行，2015 年发表的与 DevOps 相关文献数量激增，是 2014 年文献数量的 4 倍以上。而 2017 年文献数量急剧下降，很大程度上因为本研究是在 2017 年第一季度开展的文献收集工作，该数量不代表 2017 年全年的文献数量。

本文对学术文献和灰色文献（Stack Overflow）中提到和使用的工具集合进行了整合处理，分别计算每个工具在学术文献和灰色文献中各自所占的比例和，将不同数量级的工具集相结合并生成工具的频率排名，由于篇幅限制，在此只展示排名靠前的二十个工具，如图 3 所示。从图中可以看出，Docker 和 Jenkins 是所列的 20 个 DevOps 支持工具中使用最为普遍的三个。Jenkins 作为开源的 DevOps 自动化支持工具，提供了上千个插件，用于串联各种类型的构建、测试、部署自动化任务。而 Docker 作为开源的应用容器引擎，为应用和依赖包提供可移植的运行环境，因为其启动快、资源占用小而被广泛使用。同时，Jenkins 也提供插件来支持 Docker 的快速配置与启动。

关于 DevOps 中三个关键实践，持续集成 (Continuous Intergination)、持续交付 (Continuous Delivery)、持续部署 (Continuous Deployment) [17] 的工具调研结果如图 4 所示，从图 4 中可以发现一个比较明显的现象，即自研工具在企业的 DevOps 自动化实践中占据重要地位，甚至在持续交付（图 4 (b)）和持续部署（图 4 (c)）中使用频数居首位。调研结果也进一步证实了 Jenkins 和 Docker 的流行程度，如 Jenkins 是持续集成中使用最多的工具（图 4 (a)）。

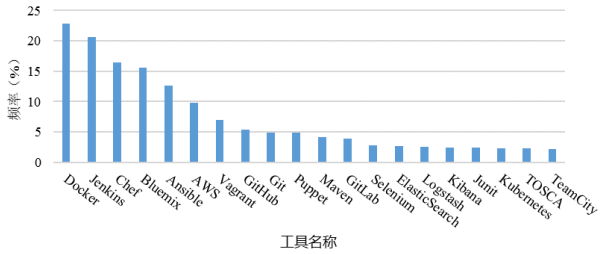
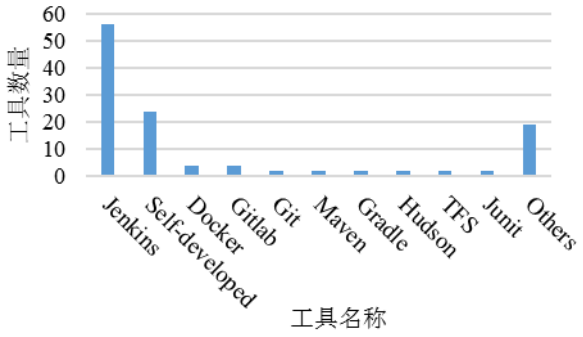


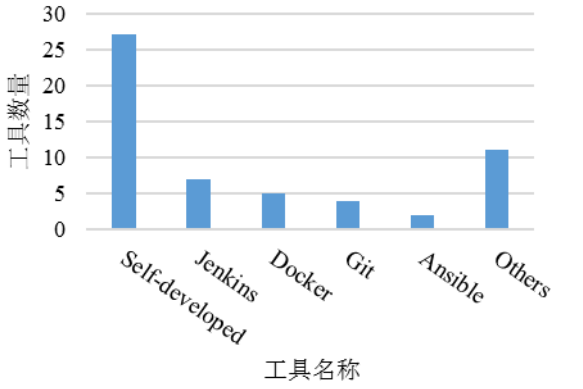
Fig.3 Integrated rank of tools related to DevOps (Papers and Stack Overflow).

图 3 DevOps 相关工具整合排名（文献和 Stack Overflow）

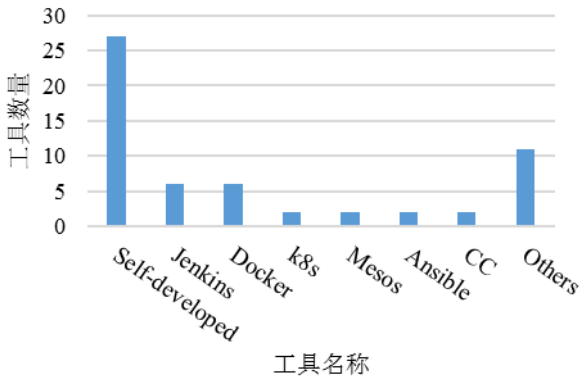
[3] http://softeng.nju.edu.cn/devops2016/devopstool2016



(a) Result of Continuous Integration.
(a) 持续集成的结果



(b) Result of Continuous Delivery.
(b) 持续交付的结果



(c) Result of Continuous Deployment.
(c) 持续部署的结果

Fig.4 Results of questionnaire.
图 4 调查问卷结果

4.2 针对研究问题二

DevOps 自动化实践涉及到软件开发和运维的各个过程阶段，相关的支持工具具有类型广、数量多的特点，同时也存在功能重叠、功能难以满足实践者需求等问题。

4.2.1 DevOps 支持工具数量多

4.1 节中提到，本研究从学术文献和灰色文献中检索到与 DevOps 相关的将近两百个工具，包括配置管理、构建、测试、集成、部署等不同类型(由 DevOps 涉及范围决定)。面对数量众多的工具集合，横向来看，很多工具在传统软件开发和运维中已使用多年，如 Puppet、Chef 和 Jenkins 等，DevOps 作为线索将其串联起来，同时近年来仍然不断有新的工具向 DevOps 靠拢，如 Bitbucket [40]和 IBM Bluemix[41]纷纷推出流水线产品来支持 DevOps 自动化；纵向来看，诸多原本属于 DevOps 范围之内的工具也持续在功能上进行扩展，以 Jenkins 为例，由于 Stack Overflow 论坛中多次提到 Jenkins 不同类型的插件，故本研究对 Jenkins 所有插件及其产生年份通过自动化手段进行了统计，统计结果如图 5 所示，从图中可以看出其插件的数量从 2007 年的几十个以惊人的速度增长到 2017 年的 1300 多个。

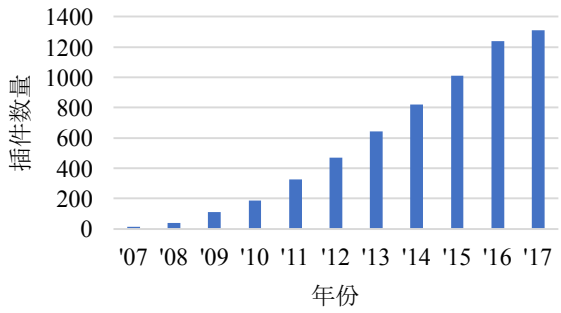


Fig.5 Amount of overall Jenkins's plugins at each year from 2007 to present.
图 5 DevOps 历年来插件数量变化 (2007-2017)

4.2.2 DevOps 支持工具功能重叠

在识别的将近两百个 DevOps 相关工具中，不仅工具之间会有常见的功能重叠的情况。DevOps 自动化所涉及的每一个软件过程，都有诸多类似的工具来支持，如所识别的工具中与配置管理相关的有 10 个，具有集成功能的工具有 8 个，监控工具有 11 个。与此同时，不同的功能也可能会重叠在一个工具中，即一个工具具备多种功能，如 Bamboo、Jenkins、IBM Bluemix 等工具都同时支持 DevOps 中构建、测试和部署等功能。

4.3.3 现有 DevOps 支持工具无法满足实际需求

数量和功能庞大的支持工具为 DevOps 实践者的选择提供了更多的可能性，其中 4.1 节中已经提到，

Docker、Jenkins 和 Chef 是工具选择中被使用或考虑最多的三个工具。尽管如此，仍然有很多实践者选择独立研发自有工具。在基于问卷的调研中，将近30%的实践者在持续集成中使用了自研工具，其中该问题的有效回答是 82；而持续交付和持续部署中自研工具的使用比例都超过 50%，这两个问题的有效回答分别是 52 和 55。部分实践者陈述选择自研工具的原因主要有两个，一个是对开源工具如 Jenkins 的安全性存在疑虑；另一个是现有的工具无法满足实践者的实际需求，比如某些对 Jenkins 进行二次开发的实践者提到，二次开发的原因正是 Jenkins 不足以支持其在 DevOps 实践中的所有需求。

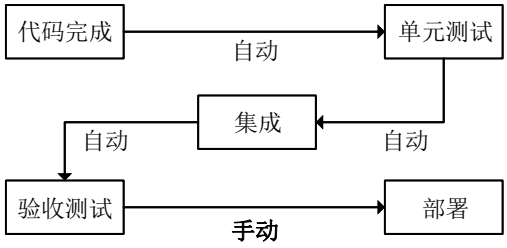
4.3 针对研究问题三

除了对学术文献、灰色文献以及调研中所出现的工具进行统计分析之外，本研究还通过对抽取的数据的分析整合，总结出两种与 DevOps 自动化实践相关的问题与挑战。

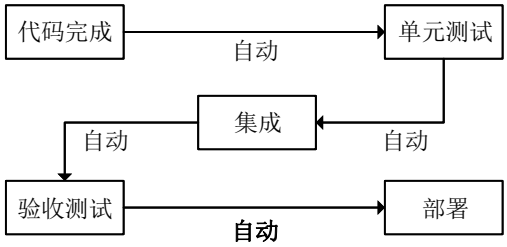
4.3.1 DevOps 关键实践理解分歧

实现 DevOps 自动化的三个关键实践分别是持续集成、持续交付和持续部署[17]。目前学术界和工业界对于它们的理解存在一定的分歧。本研究所检索的灰色文献[29]（同样也是目前学术文献关于持续集成的主要参考来源）中提到，持续集成实践指的是软件开发过程中，项目团队成员不断快速集成所完成的工作，每次集成都需经过自动化构建（包括测试）活动，目的是尽快发现集成后的错误。

目前学术界和业界对于持续集成实践的解释不存在争议，但是对于持续交付和持续部署过程的理解存在一定的分歧。如图 5 所示，灰色文献[30]对于持续交付和持续部署的解释是：持续交付实践是为了保证软件产品已经通过所有环境的测试（包括开发环境、测试环境、类生产环境）并可部署到生产环境中去，最后产品的部署操作取决于商业因素而非技术因素，通过手动实现（图 6 (a)）；与持续交付实践过程类似，持续部署实践也需要经过代码完成、自动化构建、测试、集成等阶段，不同的是通过了所有阶段测试的产品将被自动部署到生产环境中，而非手动实现（图 6 (b)）。学术界中有研究者如 Sundman 等人[31]持相同观点，但同时也存在不同的见解，如 Fitzgerald 等人[32]认为持续部署是将产品手动部署到生产环境中，而持续交付实现的是自动化部署，持续部署是持续交付的前提。



(a) Continuous Delivery.
(a) 持续交付



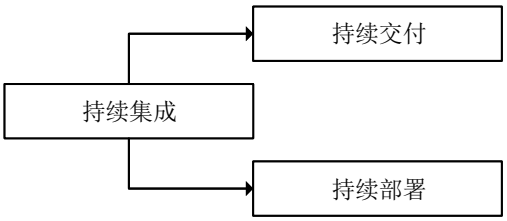
(b) Continuous Deployment.
(b) 持续部署

Fig.6 The difference between Continuous Delivery and Continuous Deployment[29].
图 6 持续交付和持续部署的差异[29]

学术界和工业界对于持续交付和持续部署过程理解的分歧隐含了二者之间两种不同的关系，一种是灰色文献明确声称的包含关系，另一种是学术文献体现的并列关系。如图 7 (a)所示，灰色文献[33]认为以持续集成为基础，持续集成、持续交付和持续部署是逐级包含的关系（学术文献[32]隐含的包含关系却相反，即持续交付包含持续部署）；如图 7 (b)所示，学术文献[34]认为，持续交付和持续部署之间唯一的不同是最后一步到生产环境的部署方式（手动或自动），不存在附属关系，是继持续集成之后相互并列的两个实践过程。

DevOps 自动化中对于三个关键实践过程和关系理解的分歧，可能会导致产业界实践的困惑。比如，在 Stack Overflow 论坛上，求助持续集成、持续部署和持续交付三者之间区别的问题帖[35]中的回答中，点赞数（204）最多的回答指出：针对这持续交付也有不同的观点，其中一种是将持续部署看成是持续交付的同义词，且在实践过程中，持续部署和持续交付存在理解冲突、术语混用的现象。

(a) The embedded relationship [33].
(a) 包含关系[33]



(b) The parallel relationship [34].
(b) 并列关系[34]

Fig.7 Different understanding of the relationship between three key practices of DevOps.
图 7 对 DevOps 三个关键实践关系的两种理解

4.3.2 DevOps 自动化软件交付过程多样化

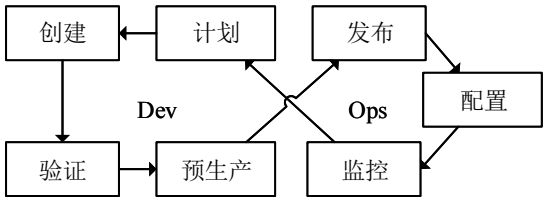
DevOps 旨在通过自动化等关键实践打破软件交付过程中开发和运维的壁垒，实现软件产品的高效、高质量交付。目前工业界产生了一系列为了支持 DevOps 自动化软件过程的工具链 (Toolchain) 或流水线 (Pipeline)， 它们都包含由多个结点组成的工作流 (Workflow)，且类型多样，每种类型的工作流中结点也存在多样性。由此一来在实践中的一大挑战就是如何理解及应对自动化工作流的多样性。

本研究对来自灰色文献的结果进行整合分析后，以工具链或流水线的结点层次为标准，将结果分为两类：阶段导向的工作流 (Stage-oriented Workflow) 、活动导向的工作流 (Activity-oriented Workflow)。阶段的层次要高于活动，一个阶段可能会包括多个活动。

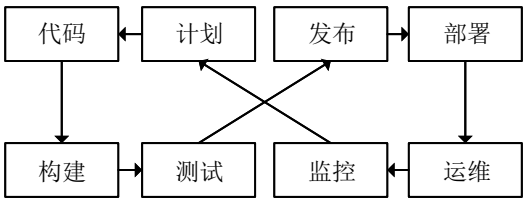
1) 阶段导向的工作流

典型的阶段导向的工作流来自由 Gartner 提出的工具链，如图 8 (a) 所示，它定义了 DevOps 工作流的七个阶段，分别是计划 (Plan)，创建 (Create)，验证 (Verify)，预生产 (Preprod)， 发布 (Release)，配置 (Configure) 和 监控 (Monitor)。其中每个阶段包括多个活动，如创建阶段有编码 (Code)、构建 (Build) 等活动；验证阶段包括自动化测试 (Test Automation)、静态分析 (Static Analysis) 等活动。

图 8 (b) 是从灰色文献[36]的工具链中识别出的另一种基于阶段的工作流，包括计划 (Plan)，编码 (Code)，构建 (Build)，测试 (Test)， 发布 (Release)，部署 (Deploy)， 运维 (Operate) 和监控 (Monitor)。通过对比可以看出 ，该工作流中除了计划、发布和监控这三个阶段，其他阶段都有别于 Gartner 所提出的工具链中的工作流。



(a) DevOps stage-oriented automation workflow defined by Gartner.
(a) Gartner 定义的阶段导向 DevOps 自动化工作流



(b) Another DevOps stage-oriented automation workflow [36].
(b) 另一种阶段导向的 DevOps 自动化工作[36]

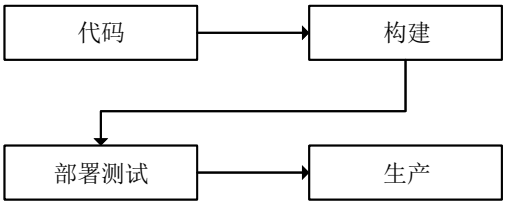
Fig.8 Different DevOps stage-oriented automation workflows.
图 8 不同的阶段导向 DevOps 自动化工作流

2) 活动导向的工作流

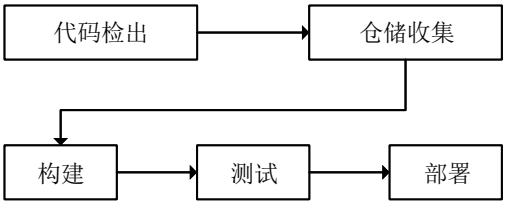
除了工作流结点是阶段层次的工具链，业界也提出了很多流水线产品来支持 DevOps 自动化软件交付过程。部分流水线产品的工作流结点属于活动级别，要低于前面所提到的阶段层次，如主流云服务提供商 AWS[37]和典型持续集成工 Jenkins [38] (前文提到) 所推出的支持 DevOps 持续交付的流水线 (图 9 (a) 和 (b))。需要注意的是，这里的结点不属于阶段，而仅仅是单一的活动结点。从图中可以看出，两个工作流所包含的活动存在一定的差异。这种差异往往是由工具本身所支持的功能决定的。如 Jenkins 本身支持代码检出 (SCM Checkout) 和收集相关仓储库 (Collect Dependent Repository) 的活动，而在 AWS 的 CodePipeline 工作流中却没有体现。

尽管 DevOps 工具链或流水线的工作流存在多样性，仍然有实践者设计适合自身需求的 DevOps

自动化软件过程工具链或流水线，如 Cisco IT [39]，甚至有不少企业也在进行自研工具的开发（如图 4 所示）。



(a) AWS activity-oriented workflow.
(a) AWS 活动导向工作流



(b) Jenkins activity-oriented workflow.
(b) Jenkins 活动导向工作流

Fig.9 Different DevOps activity-oriented automation workflows.
图 9 不同的活动导向 DevOps 自动化工作流

5 讨论

尽管 Devops 实践过程中面临诸多问题挑战，但本文认为，这只是现阶段 DevOps 发展不够成熟、实践者对 DevOps 认识不足的必然体现，理论上存在可能的策略来应对。经过对前文所提到的挑战和问题深入分析后，本文提出了三个可能的应对策略。

5.1 DevOps 生态系统术语词典

目前DevOps没有明确的业界所公认的定义[7]，DevOps 实践中的各种分歧表明，对整个 DevOps 生态系统中特定的术语，概念及其关系的讨论与研究显得尤为必要。事实上，似乎正是因为缺少这样的统一认识，才造成了前文所提到的关于 DevOps 实践的其他问题，如 DevOps 自动化交互过程中对工作流定义不统一。考虑到建立良好的术语词典将有助于在相应的研究领域促进理解和沟通[42]，本文认为，DevOps 生态系统的术语词典将有助于解决自动化实践中的理解分歧问题，统一实践者的认识。

5.2 DevOps 自动化工具关系图谱

毫无疑问，类型多样、数量众多的自动化工具在 DevOps 生态系统中有着至关重要的地位。然而，对数量和类型过多的工具缺乏认识的标准又可能在一定程度上影响 DevOps 自动化的实践。

本文建议设立 DevOps 自动化工具的关系图谱，以帮助识别现有 DevOps 相关工具并通过图谱的形式对工具之间的关系进行展示。尤其是对于 DevOps 及 DevOps 工具的确切定义有一定的必要性，这可以帮助研究人员更好地区分工具。

5.3 DevOps 自动化成熟度模型

作为前面提到的分歧的延续性问题，DevOps 自动化软件交付过程中，从所谓工具链和流水线中提取的工作流存在不一致性。例如，当描述自动化工作流时，现有文献在宏观层面上的阶段导向节点和微观层面的活动导向节点之间并没有做很好的区分。因此，本文建议通过 DevOps 自动化成熟度模型来处理此问题。理论上可以通过成熟度模型的映射可以轻松地显示自动化工作流的范围，并且能够衡量工具链对 DevOps 自动化的支持程度。请注意，类似地，DevOps 自动化成熟度模型可以被视为理想的 DevOps 成熟度模型的子集。

6 结束语

DevOps 作为一场新的软件工程变革，发展尚不成熟，学术领域研究不多，业界丰富的实践经验及多样化的支持工具为理解和认识 DevOps 现状提供了可能。

为了了解 DevOps 自动化及工具现状，并保证支撑证据的全面性，本文采用了一种混合的经验研究方法，在传统的系统化文献评价方法基础上，补充使用灰色文献评价方法收集的工业界的相关证据，以弥补学术研究的滞后性。另外本研究还开展了基于调查问卷的调研，调研结果作为证据的进一步补充与支撑。

通过对研究结果的整合分析，从两方面揭示了 DevOps 自动化及工具的现状：DevOps 相关工具数量众多、类型广泛；DevOps 支持工具、自动化实践、自动化软件交付过程面临一系列的问题与挑战，如需求无法满足、理解分歧、过程定义不一致等。针对产生的问题，本文从建立 DevOps 生态系统词典、DevOps 自动化工具的关系图谱、DevOps 成熟度模型等这三个角度提出可能的应对策略，同时这也将是本文未来的研究方向。

References:

[1] Ebert C, Gallardo G, Hernantes J, et al. DevOps[J]. IEEE Software, 2016, 33(3):94-100.

[2] Httermann M. DevOps for Developers[M]. ser. Expert's Voice in Web Development. New York: Apress, 11 September 2012.

[3] Dyck A, Penners R, Lichter H. Towards Definitions for Release Engineering and DevOps[C]// IEEE/ACM, International Workshop on Release Engineering, Florence, Italy, May 2015. IEEE, 2015:3-3.

[4] França B B N D, Jeronimo H, Travassos G H. Characterizing DevOps by Hearing Multiple Voices[C]// Brazilian Symposium on Software Engineering, Maringa´, Brazil, September 2016. ACM, 2016:53-62.

[5] Jabbari R, Ali N B, Kai P, et al. What is DevOps?: A Systematic Mapping Study on Definitions and Practices[C]// Scientific Workshop Proceedings of Xp, Edinburgh, Scotland, UK, May 2016. ACM, 2016:12.

[6] 张贝贝. DevOps:推倒隔阂之墙[J]. 软件和信息服务，2013(8):56-57.

[7] Plancius A, DevOps a definition please?[OL]. [2017-07-20].https://www.linkedin.com/pulse/devops-definition-please-anne-plancius.

[8] 邹筱菁，柯林. 基于 DevOps 的软件开发管理模式[J]. 数字技术与应用，2016(11):184-187.

[9] Farroha B S, Farroha D L. A Framework for Managing Mission Needs, Compliance, and Trust in the DevOps Environment[C]// Military Communications Conference. IEEE, Baltimore, MD, USA, October 2014. IEEE, 2014:288-293.

[10] Virmani M. Understanding DevOps & bridging the gap from continuous integration to continuous delivery[C]// Fifth International Conference on Innovative Computing Technology, Pontevedra, Spain, May 2015. IEEE, 2015:78-82.

[11] Kaur K, Jajoo A, Manisha. Applying Agile Methodologies in Industry Projects: Benefits and Challenges[C]// International Conference on Computing Communication Control and Automation, Pune, India, February 2015. IEEE, 2015:832-836.

[12] Olszewska M, Waldén M. DevOps meets formal modelling in high-criticality complex systems[C]//Proceedings of the 1st International Workshop on Quality-Aware DevOps, Bergamo, Italy, September 2015. ACM, 2015: 7-12.

[13] Waller J, Ehmke N C, Hasselbring W. Including performance benchmarks into continuous integration to enable DevOps[J]. ACM SIGSOFT Software Engineering Notes, 2015, 40(2): 1-4.

[14] Erich F, Amrit C, Daneva M. A mapping study on cooperation between information system development and operations[C]//International Conference on Product-Focused Software Process Improvement, Helsinki, Finland, December 2014. Springer, Cham, 2014: 277-280.

[15] Lwakatare L E, Kuvaja P, Oivo M. Dimensions of devops[C]//International Conference on Agile Software Development, Helsinki, Finland, May 2015. Springer, Cham, 2015: 212-217.

[16] Puppet, 2016 state of DevOps report[OL]. [2017-07-20]. https://puppet.com/resources/ whitepaper/2016-state-of-devops-report.

[17] Loukides M. What is DevOps?[M]. " O'Reilly Media, Inc.", 2012.

[18] Akshaya H L, Vidya J, Veena K. A Basic Introduction to DevOps Tools[J]. International Journal of Computer Science and Information Technologies, 2015, 6 (3): 2349-2353.

[19] Vaasanthi R, Kingston P S and Kumari P V. Comparative Study of DevOps Build Automation Tools[J]. International Journal of Computer Applications, 2017, 170 (7):1-8.

[20] XebiaLabs, Periodic table of DevOps tools[OL]. [2017-07-20].https://xebialabs.com/periodic-table-of-devops-tools/.

[21] Kitchenham B A, Dyba T, Jorgensen M. Evidence-based software engineering[C]//Proceedings of the 26th international conference on software engineering, Edinburgh, UK, May 2004. IEEE Computer Society, 2004: 273-281.

[22] Zhang H, Babar M A, Tell P. Identifying relevant studies in software engineering[J]. Information and Software Technology, 2011, 53(6): 625-637.

[23] Garousi V, Felderer M, Mäntylä M V. The need for multivocal literature reviews in software engineering: complementing systematic literature reviews with grey literature[C]//Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering, Limerick, Ireland, June 2016. ACM, 2016: 26.

[24] Correa D, Sureka A. Chaff from the wheat: characterization and modeling of deleted questions on stack overflow[C]//Proceedings of the 23rd international conference on World wide web. ACM, 2014: 631-642.

[25] Allamanis M, Sutton C. Why, when, and what: analyzing stack overflow questions by topic, type, and code[C]//Proceedings of the 10th Working Conference on Mining Software Repositories. IEEE Press, 2013: 53-56.

[26] Calefato F, Lanubile F, Marasciulo M C, et al. Mining successful answers in stack overflow[C]//Mining Software Repositories (MSR), 2015 IEEE/ACM 12th Working Conference on. IEEE, 2015: 430-433.

[27] Marder A. Stack overflow badges and user behavior: an econometric approach[C]//Proceedings of the 12th Working Conference on Mining Software Repositories. IEEE Press, 2015: 450-453.

[28] Ye D, Xing Z, Kapre N. The structure and dynamics of knowledge network in domain-specific Q&A sites: a case study of stack overflow[J]. Empirical Software Engineering, 2017, 22(1): 375-406.
[29] M. Fowler, Continuous Integration[OL]. (2006-05-01) [2017-07-20].https://www.martinfowler.com/articles/continuousIntegration.html.
[30] S. Prince, The Product Managers' Guide to Continuous Delivery and DevOps[OL]. (2016-02-11) [2017-07-20]. http://www.mindtheproduct.com/2016/02/what- the- hell-are- ci- cd- and- devops- a- cheatsheet- for- the- rest- of-us/.
[31] Y. Sundman, Continuous Delivery vs Continuous Deployment[OL]. (2013-02-05) [2017-07-20]. http://blog.crisp.se/2013/02/05/yassalsundman/continuous - delivery- vs- continuous-deployment.
[32] Fitzgerald B, Stol K J. Continuous software engineering and beyond: trends and challenges[C]//Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering. ACM, 2014: 1-9.
[33] A. Tiwari, Continuous integration vs continuous delivery vs continuous deployment[OL]. [2017-07-20]. http://www.saviantconsulting.com/blog/difference- between- continuous- integration- continuous- delivery-and- continuous- deployment.aspx.
[34] Shahin M, Babar M A, Zhu L. Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices[J]. IEEE Access, 2017, 5: 3909-3943.
[35] Stack Overflow, Continuous integration vs. continuous delivery vs. continuous deployment[OL]. [2017-07-20]. http://stackoverflow.com/questions/28608015/ continuous- integration- vs- continuous- delivery- vs- continuous-deployment.
[36] A. Bridgwater, Inside DevOps: How it really works[OL]. (2015-06-01) [2017-07-20]. http://www.computerweekly.com/feature/Inside- DevOps-How- it- really-works.
[37] AWS, AWS codePipeline[OL]. [2017-07-20]. https://aws.amazon.com/cn/codepipeline/
[38] Jenkins, Jenkins Pipeline[OL]. [2017-07-20]. https://jenkins.io/doc/book/pipeline/
[39] Cisco, DevOps toolchain for continuous application delivery[OL]. [2017-07-20]. http://www.cisco.com/c/en/us/solutions/collateral/enterprise/cisco- on-cisco/i- db- 12232016- continuous- delivery-toolchain.html, 10 May 2017.
[40] Stack Overflow, Is there a ci service for bitbucket.org which allow managing build commands in a VCS file? [OL]. [2017-07-20]. http://stackoverflow.com/questions/35254434/is-there-a-ci- service- for- bitbucket- org- which- allow-managing- build- commands- in- a.
[41] Stack Overflow, How to integrate private GitHub repo with Bluemix DevOps service[OL]. [2017-07-20]. http://stackoverflow.com/questions/41850433/how-to-integrate- private- github- repo- with- bluemix-devops-service.
[42] Li Z, OBrien L, Cai R, et al. Towards a taxonomy of performance evaluation of commercial Cloud services[C]//Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on, Honolulu, Hawaii, USA, June 2012. IEEE, 2012: 344-351.

1 寸数字照片须为证件照，不能提供生活照，不能低于 300 像素

LI Shanshan was born in 1990. She is a Ph.D. candidate at Nanjing university. Her research interests include Software Architecture, DevOps and Container, Empirical Software Engineering, etc.

李杉杉(1990–)，女，南京大学博士研究生，主要研究领域为软件体系结构，开发运维与容器技术，经验软件工程等。

1 寸数字照片须为证件照，不能提供生活照，不能低于 300 像素

LI Zheng received the Ph.D. degree in Computer Science from Australian National University in 2015. He is a visiting researcher at Nanjing University. His research interests include Cloud Computing, Performance Engineering, Empirical Software Engineering, etc.

李铮，男，2015 年于澳洲国立大学获得博士学位，目前是南京大学软件学院访问学者，主要研究领域为云计算、性能工程、经验软件工程等。目前发表了 37 篇国际期刊/会议/章节论文、1 部个人专著、及 7 篇中文期刊论文，其中 SCI 索引 7 篇。

1寸数字照片须为证件照，不能提供生活照，不能低于300像素

ZHANG He received the Ph.D. degree from The University of New South Wales. He is a Full Professor of Software Engineering at the Nanjing University and a Senior Member of China Computer Federation (CCF). He has published over 100 peer-reviewed papers in high quality international conferences and journals, and won 9 Best Paper Awards from several prestigious international conferences (e.g., ICSE, ESEM, ICSP/ICSSP, APSEC, and EASE). His research interests include Software & Systems Engineering Process, Software Architecture, DevOps and Container, Empirical Software Engineering, etc.

张贺，于新南威尔士大学获得博士学位，现为南京大学软件工程教授，CCF 高级会员。近年来，著有英文专著两部，并在包括国际软件工程大会（ICSE）在内的国际重要软件工程期刊和会议上发表论文100余篇，其中9篇会议长文获最佳论文奖。主要研究领域为软件及系统工程过程，软件架构，开发运维与容器技术，经验软件工程等。

1寸数字照片须为证件照，不能提供生活照，不能低于300像素

LIU Bohan was born in 1991. He is a Ph.D. candidate at Nanjing university. His research interests include Software Process, Empirical Software Engineering, etc.

刘博涵(1991-)，男，南京大学博士研究生，主要研究领域为软件过程，实证软件工程等。