

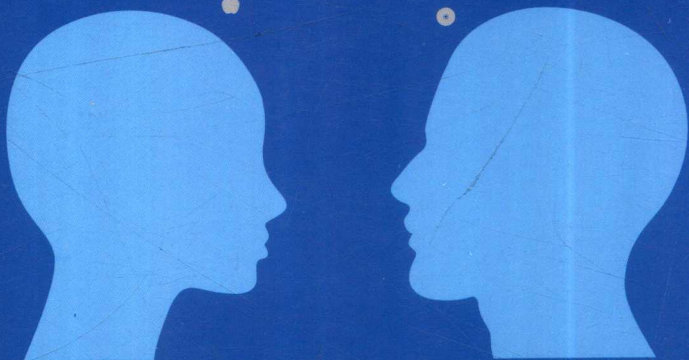
ITSS® TAG
Press

大规模组织

DevOps实践

传统IT的DevOps转型之路

陈能技 付勇 编著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

大规模组织 DevOps 实践

陈能技 付勇 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

DevOps 是开发运维一体化的软件工程思想, DevOps 尝试打破部门墙, 构建一个协同的 IT 建设运行环境, 通过工具链形成数据关联的规范化、规模化的软件持续交付流水线, 从而助力企业业务的敏捷发展。

本书用 9 章内容, 结合项目实战案例和业界主流工具, 从思想、实践方法、工具技术、案例等方面阐述了 DevOps 在传统规模化 IT 组织中实施落地的方法。

本书适合 IT 主管、项目经理、开发、测试、配置管理、运维等 IT 从业人员阅读。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究。

图书在版编目(CIP)数据

大规模组织 DevOps 实践 / 陈能技, 付勇编著. —北京: 电子工业出版社, 2018.11

ISBN 978-7-121-35136-5

I. ①大… II. ①陈… ②付… III. ①软件工程 IV. ①TP311.5

中国版本图书馆 CIP 数据核字 (2018) 第 224855 号

策划编辑: 李 冰

责任编辑: 李 冰 特约编辑: 赵海军 等

印 刷: 三河市鑫金马印装有限公司

装 订: 三河市鑫金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱

邮编 100036

开 本: 787×980 1/16 印张: 20

字数: 384 千字

版 次: 2018 年 11 月第 1 版

印 次: 2018 年 11 月第 1 次印刷

定 价: 68.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 zits@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: libing@phei.com.cn。

推荐序

未来的企业都是软件企业，是面向互联网、物联网、大数据和人工智能的企业，软件作为信息系统的核心部分一直面临着专业化交付的问题，如何才能像工厂生产零部件和整机一样可以流水线作业、批量化生产和交付，是 IT 从业人员一直追求的目标。

从早年的 ISO、CMMI，到近年来的敏捷、精益，再到最近所倡导的 DevOps，其实都在尝试解决软件行业的专业化交付问题。看起来 ISO、CMMI 有点过于保守、笨重、僵化，而敏捷又过于灵活、难以规模化应用，直到最近我们看到 DevOps 所倡导的开发、测试、运维一体化协同，以及自动化流水线的构建，结合新兴的技术（如微服务、容器等），我们认为这是非常有希望解决规模化组织下的专业化交付问题的。

陈能技在软件专业化交付这个领域探索和实践了十几年，将自己的所见、所思、所想、所践、所得集结成书，共享给行业同人，实属可贵、可敬。

润和软件目前正在做软件专业化交付服务的全面转型，本书的内容值得借鉴，借助 DevOps（开发运维一体化）、SAFe（规模化）等新兴的软件交付方法，打造和持续完善我们的专业化交付团队，为国内外客户提供优质的 IT 服务。

陈 斌

润和软件总裁

前言

DevOps 代表了 IT 领域最新的发展趋势，传统金融 IT 部门在尝试 DevOps 转型中碰到各种问题和诸多误区，我们在 DevOps 领域深耕多年，结合金融企业实际问题进行各种探索与实践，提出从流水线设计、自动化、度量等多个维度进行 DevOps 体系建设，从而形成企业 IT “专业化交付”能力的实践方法，构建全生命周期双模软件自动化生产发布与智能化运维的“BTO”蓝图。

我们将这几年在传统企业落地实施敏捷、DevOps 的经验集结成书，分享给大家，希望能够帮助大家在 DevOps 的转型中少走一些弯路。

在本书即将出版之际，我想谈谈这几年从事 DevOps 领域业务的一些感悟，不管是帮助某运营商企业规划设计 DevOps 体系，还是在某科技公司担任架构师设计 DevOps 工具平台，或者是现在所专注的工作——DevOps 解决方案咨询和售前，我始终在思考一个问题，即如何将看似先进的 DevOps 思想和理念成功地导入国内的传统企业。

不管是不是敏捷、是否 DevOps，始终要解决如下几个核心问题。

1. 流程改进

如何使流程更加精益，以及通过杜绝浪费提高效率。

2. 工具自动化

如何利用工具实现流程的自动化改进，使其具有可重复性及可扩展性，同时减少错误的发生。

3. 平台及环境

如何为从需求到生产上线全过程中各种应用的交付流水线，搭建更具弹性、灵活、可伸缩、可配置的平台及环境。

4. 文化

尤其是如何塑造信任、沟通、协作的文化氛围。

DevOps 的实施一定要有业务目标。通常业务线对 IT 部门的要求包括快速交付、敏捷、创新、优质、低成本。因此，IT 部门需要想办法建立以下能力：

- 价值交付时间。
- 部署速度。
- 成本节约—交付时间。
- 成本节约—测试时间。
- 提高测试覆盖率。
- 提高环境利用率。
- 最大限度地缩短部署所需的停机时间。
- 最大限度地减少部署所花费的时间。
- 尽可能地减少部署应用的回滚。
- 提高再现与缺陷修复能力。
- 最大限度地缩短产品问题的平均修复时间（MTTR）。
- 缩短缺陷周期。

本书分为思想篇、实践方法篇、工具技术篇、案例篇四部分，把传统企业实施 DevOps 会碰到的困惑，在开发、测试、运维三大领域各有哪些 DevOps 最佳实践，以及如何结合开源或商业工具打造交付流水线工具链，分别进行了阐述，希望对你在实施 DevOps 转型过程中实现上述业务目标有所帮助。

本书出版之时恰逢国内 DevOps 领域蓬勃发展之际，DevOps 相关的峰会已经举办了两年，我们一直关注会议的各类议题，发现从前几年的大谈概念为主，到近期大家纷纷

聚焦在落地实践方法、工具链的打造、各种探索实验总结、国内外不同行业的实践案例等，这表明 DevOps 已经开始逐步深入人心，逐步地帮企业解决 IT 中的各类问题，走向更高级的发展阶段。

在出版本书的过程中，得到了所在公司新维数联（北京）科技有限公司（前身为“天维科技”）的大力支持，书中提炼的很多实践经验及工具平台设计理念来源于该公司的项目交付团队及产品团队。第 6 章“双模发布管理平台的设计与应用”的内容主要来自公司的 DevOps 产品线经理付勇两年多的实战经验总结，该产品 AutoChain 已经处于业界领先地位，更获得了传统金融行业（银行、保险）客户的认可，知名度比较高，值得大家参考借鉴其设计理念。另外，也要特别感谢新维数联的 CEO 王勇先生及 VP 王思国先生对本书的出版给予的支持和帮助，他们在 DevOps 领域的业务洞察能力和方向把握能力令我钦佩，王勇把软件系统的建设（Build）、检验（Test）、运营（Operation）提炼成了“BTO”蓝图，并以此为公司业务发展战略落实执行，在本书的成稿过程中也提出了很多建设性的意见。

本书中的“软件工厂”的先进软件工程思想来源于王杰先生，他是一个能深入思考和洞察软件行业发展趋势及传统 IT 行业各类问题所在的人。他一手创办了中国金融 IT 行业最大的软件测试服务提供商——捷科智诚，提出“软件工厂”思想并创办了隆正互联网公司，该公司的负责人吴向东老师也是我非常敬佩的人，在落地实施软件工厂的建设方面不遗余力，我也有幸在两位老师的指导下，基于 DevOps 的理念设计了软件工厂的生产流水线。书中不少实践经验的总结提炼都来源于我所服务的客户，如中国银行、中信银行、安邦保险、青岛银行等，在此也感谢他们给予我这么宝贵的实践机会和项目上的协助，让我在产品设计及理论方法上都能更上一个台阶。

本书关于软件标准化生产、IT CT 化的思想来源于与福建某运营商高管的交流过程中的领悟，获益匪浅，在此表示感谢；本书关于技术债务的处理实践方法来源于浙江某运营商的项目经验，在此表示感谢。该运营商曾在业界率先引入敏捷、DevOps 思想，并建设了相关的 DevOps 平台，我有幸参与该平台的建设过程并负责相关规划设计工作，获益匪浅。

本书关于持续集成的部分案例来自深圳某政府单位的研发过程管控项目的实践，在此表示感谢。从这个项目的实践中我探索了中等规模企业对 DevOps 的需求与实践方法，提炼出不少有用的经验。

本书关于敏捷、规模化敏捷的思考受到光环国际李建昊老师的启发，该公司是国内最大的 PMP、敏捷项目培训、咨询服务公司之一。

本书关于容器技术相关的内容得到了合作伙伴数人云的帮助，他们是国内专注于微服务、容器云、DevOps 领域的公司，感谢公司创始人王璞和售前总监贺洪龙在 DevOps 合作项目过程中给予的帮助，以及对本书内容提出的宝贵建议。

本书关于持续集成过程中整合性能测试的内容得到了合作伙伴臻云科技的帮助，公司的创始人金发华、王凡基于开源的 JMeter 发展了业界领先的性能测试平台 XMeter。

本书关于精准测试的内容得到了合作伙伴星云测试的帮助，精准测试所倡导的通过代码覆盖率度量测试充分度，关联测试用例进行精准回归测试的做法与 DevOps 的协同、自动化理念非常吻合。

本书成书过程中还得到了公司同事的帮助，如付勇、彭菲、陈强、彭伟国、黄凯、李翅展等，他们在各自的领域都非常资深，包括 Scrum、配置管理、自动化测试等，他们在并肩作战的各类项目中提炼总结经验，提供书稿素材，在此表示衷心感谢。

本书将采用敏捷思想持续迭代优化，欢迎大家提出优化修改建议，我将在“IT 运营公社”公众号与读者进行互动。



陈能技

2018年7月

目 录

第一部分 思想篇

第 1 章 软件工厂	2
1.1 软件的生产力	2
1.2 软件工厂——软件的标准化生产	4
1.2.1 标准化生产模式需要一个集成底座——PaaS	4
1.2.2 标准化软件生产流水线	6
第 2 章 DevOps 思想	8
2.1 DevOps 思想与生产流水线	8
2.1.1 DevOps 的起源	8
2.1.2 DevOps 对软件工厂的启发	9
2.1.3 从 DevOps 实践原则看软件生产工艺化水平的提高	10
2.1.4 软件工厂模式下对生产工具发展的促进作用	10
2.2 从瀑布到敏捷	11
2.2.1 传统项目管理问题	12
2.2.2 向互联网企业学习的“敏捷”	13
2.2.3 敏捷的起源	14
2.2.4 瀑布模型	15
2.2.5 传统企业不可能全盘敏捷化	15

2.2.6	从版本上线过程管理看敏捷与瀑布	16
2.2.7	敏捷的前提是“不敏捷”	17
2.3	敏捷与 DevOps 核心思想	18
2.3.1	增量开发与测试	18
2.3.2	持续集成与持续交付	19
2.3.3	自动化	21
2.4	规模化的敏捷	22
2.4.1	从敏捷项目管理到敏捷项目集群管理	22
2.4.2	企业规模化敏捷思想	24
2.4.3	规模化敏捷方法——SAFe	25
2.4.4	规模化敏捷开发的最佳实践	28
2.5	企业规模化敏捷与软件工厂	31
2.5.1	软件生产环境	31
2.5.2	软件工厂生产环境管理——开发测试云	33
2.5.3	整合环境仿真	37
2.5.4	不可忽略的办公环境因素	43
2.5.5	软件工厂质量检测——深度自动化测试装置	45

第二部分 实践方法篇

第 3 章	DevOps 体系的建立	49
3.1	构建 DevOps 流水线，打通开发—测试—运维持续交付通道	49
3.2	敏捷开发知识体系	50
3.3	Scrum 框架	51
3.3.1	Scrum 的 3 类角色	52
3.3.2	Scrum 的 3 份工件	52
3.3.3	Scrum 的 5 项活动	53

3.3.4 Scrum 的 5 个价值观.....	54
3.4 敏捷、DevOps 与协作	54
3.5 开发域 DevOps 实践	55
3.5.1 敏捷需求.....	55
3.5.2 传统需求分析.....	55
3.5.3 敏捷需求方法.....	57
3.5.4 面向敏捷的架构设计	61
3.5.5 向微服务架构迁移的基本做法	63
3.5.6 向微服务架构迁移的低风险演进方法	64
3.5.7 微服务架构设计模式.....	74
3.5.8 开发敏捷.....	77
3.5.9 分支策略.....	77
3.5.10 依赖包管理.....	79
3.5.11 技术债务.....	80
3.5.12 技术债务的形成.....	80
3.5.13 技术债务的分类.....	83
3.5.14 技术债务与质量投资.....	84
3.5.15 技术债务处理方法.....	85
3.5.16 持续集成.....	88
3.5.17 持续集成最佳实践.....	91
3.5.18 持续集成最佳实践——持续编译.....	92
3.5.19 持续集成最佳实践——持续代码检查.....	92
3.5.20 持续集成最佳实践——持续测试.....	93
3.5.21 持续集成最佳实践——持续部署.....	94
3.5.22 持续集成最佳实践——持续报告.....	94
3.5.23 企业持续集成实施方法.....	95

3.5.24	持续集成实施案例介绍	96
3.6	测试域 DevOps 实践	96
3.6.1	敏捷测试环境管理	96
3.6.2	业界关于测试环境管理的实践	98
3.6.3	测试如何纳入持续集成体系中	100
3.6.4	敏捷测试方法与传统测试方法的不同之处	105
3.6.5	敏捷测试人员的定义	106
3.6.6	敏捷测试过程管理	107
3.6.7	敏捷测试团队管理	112
3.7	运维域 DevOps 实践	113
3.7.1	敏捷基础设施	113
3.7.2	自动化部署	113
3.7.3	自动化部署的要素	114
3.7.4	部署常见场景及问题	115
3.7.5	部署服务工具链	116
3.7.6	资源部署	117
3.7.7	自动化资源部署	117
3.7.8	应用部署	119
3.7.9	应用交付容器化	120
3.7.10	应用容器化改造	121
3.7.11	不中断服务的部署方法	121
3.7.12	敏捷运维	125
第 4 章	DevOps 的有效实践	129
4.1	敏捷模式与传统企业研发模式的结合	129
4.2	敏捷模式与传统 ISO、CMMI 标准规范的兼容	131
4.2.1	需求获取	131

4.2.2	方案制定（或技术评估）	132
4.2.3	生产任务排期	135
4.2.4	Sprint 0——迭代前准备	135
4.2.5	Sprint 0——基础框架搭建	143
4.2.6	迭代实施	143
4.2.7	验收测试	154
4.2.8	投产	155
4.3	敏捷与 DevOps 基础实践	155
4.3.1	SRE 与发布工程	155
4.3.2	配置管理——版本管理	161
4.3.3	配置管理——发布包制作规范	165
4.3.4	缺陷管理	167
4.3.5	缺陷预防	168
4.3.6	迭代周期的时间	169
4.4	敏捷与 DevOps 反模式	170
4.4.1	急功近利	170
4.4.2	跨部门协作的浪费	172
4.4.3	持续集成“坏味道”	176
4.4.4	手工完成所有部署	178
4.4.5	开发与运维各一套自动化部署系统	178
4.4.6	重建数据库比较困难	179

第三部分 工具技术篇

第 5 章	精准测试技术	182
5.1	什么是精准测试技术	182
5.1.1	传统测试方法面临大型软件时的问题与瓶颈	182

5.1.2	传统白盒测试方法的问题	183
5.1.3	开发团队和测试团队协同工作难	183
5.1.4	精准测试方法解决的问题	184
5.1.5	精准测试的工作模式与工作原理	185
5.2	精准测试云平台	186
5.2.1	测试复杂度分析	186
5.2.2	测试复杂度种类	187
5.2.3	覆盖率分析	188
5.2.4	软件内部逻辑结构可视化	192
5.2.5	函数列表功能说明	196
5.2.6	覆盖率可视化	197
5.2.7	动态实时测试监控	199
5.2.8	缺陷管理系统	199
5.2.9	Bug 与崩溃代码级捕获	201
5.2.10	双向追溯	202
5.2.11	多版本与累计覆盖率	205
5.2.12	智能的回归测试用例选取分析算法	206
5.2.13	测试用例的聚类分析算法	207
5.3	精准测试与度量分析	208
5.3.1	获取度量信息	208
5.3.2	项目度量指标	208
5.3.3	项目汇总	211
5.3.4	测试用例——按日趋势图	213
5.3.5	测试用例——测试用例列表	215
5.3.6	测试人、机	218
5.3.7	测试缺陷——Bug 信息汇总	219

5.3.8	测试缺陷——Bug 详细列表	220
5.3.9	覆盖率——按日增长趋势图	221
5.3.10	函数、类、文件覆盖率统计	223
5.3.11	函数、类、文件复杂度和覆盖率关系图	223
5.3.12	覆盖率列表	224
5.3.13	函数、类、包复杂度统计	226
5.3.14	测试覆盖率漏洞的智能分析	227
第 6 章	双模发布管理平台的设计与应用	229
6.1	产生背景	229
6.1.1	传统企业数字化转型浪潮下的双模挑战	229
6.1.2	双模发布管理平台	230
6.2	双模发布管理平台设计	231
6.2.1	设计思路	231
6.2.2	架构设计	233
6.3	案例及功能说明	234
6.3.1	案例背景介绍	235
6.3.2	痛点诊断及建设目标	236
6.3.3	功能说明	236
第 7 章	整合环境仿真技术	252
7.1	服务虚拟化技术	252
7.2	Lisa 解决方案特色	253
7.3	采用 Lisa 方案的预期效果	254
7.4	Lisa 实施和服务模式举例	256
第 8 章	容器技术	259
8.1	容器与 Docker	259

8.2	应用容器化	261
8.3	Dockerfile 命令详解	266
8.4	Dockerfile 使用建议	270
8.5	应用上云容器	271
8.5.1	上云应用的场景	271
8.5.2	软件和硬件的限制	272
8.5.3	应用架构的限制	272
8.5.4	应用上云建议	274
8.6	容器化应用开发规范	275
8.6.1	应用预编译	275
8.6.2	热点数据缓存	277
8.6.3	应用无状态化	277
8.6.4	应用上配置的定时任务需与应用分离	278
8.6.5	应用使用固定的服务监听端口	278
8.6.6	主机名和服务端口获取	278
8.6.7	数据库连接池	279
8.6.8	应用日志	279
8.6.9	应用中间件选型	279
8.6.10	应用环境配置规范	280
8.6.11	硬件负载	280
8.6.12	不要在容器中存储数据	281
8.6.13	不要发布两个应用	281
8.6.14	清除不必要的包和文件	281
8.6.15	不要在容器中运行多个进程	281
8.6.16	不要在镜像中存储凭据、使用环境变量	281
8.6.17	使用非 root 用户运行	281

8.6.18 不要依赖 IP 地址	282
8.6.19 服务间接口调用使用 TCP 和 HTTP	282
8.7 容器化应用迁移实施方案	282
8.7.1 应用迁移的流程	282
8.7.2 信息收集和业务分析	284
8.7.3 容器镜像制作	289
8.7.4 功能验证	291

第四部分 案例篇

第 9 章 持续集成实践案例	294
9.1 项目背景	294
9.2 解决方案	295
9.2.1 持续集成简介	295
9.2.2 应用持续集成解决某局的开发项目问题	296
9.2.3 持续集成基础技术框架	297
9.2.4 方案优势	302