

持续交付 2.0

业务引领的DevOps精要

乔梁 / 著

Continuous Delivery 2.0

Business-leading DevOps Essentials



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

持续交付^{2.0}

业务引领的DevOps精要

乔梁 / 著

Continuous Delivery 2.0

Business-leading DevOps Essentials

人民邮电出版社

北京

图书在版编目 (C I P) 数据

持续交付2.0 : 业务引领的DevOps精要 / 乔梁著

— 北京 : 人民邮电出版社, 2019.1

ISBN 978-7-115-50001-4

I. ①持… II. ①乔… III. ①软件工程 IV.
①TP311.5

中国版本图书馆CIP数据核字(2018)第251366号

内 容 提 要

本书“重新定义”了持续交付, 增补了组织管理和架构两个维度, 辅助以真实案例, 对诸多持续交付的原则和实践加以解读, 并对持续交付过程中的取舍原则加以论述。

本书分三个部分: 第一部分作者根据自己近十年的工作及咨询经历, 通过不断总结、提炼和反思, 对原有的持续交付进行了修正, 重新定义持续交付为实现组织战略目标的能力, 并引入持续交付的能力模型; 第二部分阐述组织打造持续交付能力模型所需遵守的原则, 包括基础原则、组织原则和架构原则; 第三部分通过多个互联网公司案例的解读, 阐述如何根据组织的当前状况, 应用相关原则对最佳实践进行取舍, 并快速达到组织能力目标。

本书适合大型互联网公司的技术 VP、技术负责人, 中小型互联网公司的 CTO、技术 VP、研发/测试/运维负责人、主管及骨干, 以及组织变革者阅读。

◆ 著 乔 梁

责任编辑 杨海玲

责任印制 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

大厂聚鑫印刷有限责任公司印刷

◆ 开本: 800×1000 1/16

印张: 21.75

字数: 416 千字

印数: 1—3 000 册

2019 年 1 月第 1 版

2019 年 1 月河北第 1 次印刷



定价: 89.00 元

读者服务热线: (010) 81055410 印装质量热线: (010) 81055316

反盗版热线: (010) 81055315

广告经营许可证: 京东工商广登字 20170147 号

序 一

2007年，我在北京第一次遇到乔梁，并与之合作，共同开发一款软件产品，名为 go.cd，这是第一个支持部署流水线的CI/CD工具。我们在构建这款产品过程中的思想碰撞与经验总结为《持续交付》一书提供了非常关键的素材。

在过去10年中，这些内容已经从最前沿的想法变为业界公认的智慧。每个追求卓越的科技公司都希望能够随时随地发布，而无须工程师在晚上或周末进行部署。能够快速、频繁且安全地发布软件，并实现小批量交付，意味着我们可以快速获得对我们的想法的反馈。我们可以构建原型并使用真实用户对其进行测试，从而避免开发那些对用户没有任何价值的功能。反过来，这也意味着产品更好，客户更满意，员工更快乐。这些能力对需要这种工作方式的每个组织来说，都是非常关键的。

然而，获得这种能力并不是一件容易的事情。组织需要对软件系统架构进行不断演进，使其支持尽快且有效的测试，以及快速部署，同时，还需要培养快速试验的文化。文化因素对于成功实施持续交付和通过持续交付实现产品管理实践至关重要。

乔梁曾与中国的各类组织合作，帮助它们实施持续交付并实现其效益。我想不出比乔梁更合适的人选，来写一本关于如何根据实际经验实现这些想法的书。希望本书的每位读者都能在提高软件交付能力的不断尝试中取得圆满成功，并利用这种能力来构建更好的产品和服务，以及更快乐、更高效的团队。

Jez Humble

《持续交付》一书作者，DORA联合创始人兼CTO

序 二

我们处在移动互联网时代的下半场。这意味着，对于主流人群来说，头部的基本需求或许已经被满足或者说“饱和”，而未来的方向或许是现有模式的颠覆，或许是长尾个性化需求的充分满足，又或许是小众人群的挖掘。无论如何，我相信未来创新的成本和创新失败的可能性有很大的机会持续走高。

在这种大背景下，如何通过极致的产品研发运营效率，快速发现新的机会，降低创新成本就变得尤为重要。乔梁老师是软件工程领域的大师，同时作为腾讯的敏捷咨询顾问，也和我们合作多年，亲历了移动互联网发展的全过程。乔梁老师对产品研发运营体系的种种问题可谓洞若观火，对于效率的追求可谓孜孜不倦。

“在快速变化的互联网环境下，软件产品团队如何在质量与速度上取得平衡？如何让软件产品以最快的速度抵达用户手中，使团队得到最快速的反馈？”我与乔梁老师在多年前就曾针对这些问题有过深入的探讨。“质量与速度根本不存在平衡，只有在产品能够承受的一定质量水准基础上，追求交付的速度才有意义。”他的观点如此鲜明，没有咨询师的“口头禅”——It depends（视情况而定），却道出了产品整个生命周期中的一个不变法则。虽然不同的产品阶段对产品质量要求有不同的定义，但在同一产品阶段中，质量要求却几乎是稳定的。本书中所讨论的内容全部围绕“如何在满足质量要求的前提下快速交付产品价值”这一问题展开。

为了快速发现新的机会，必须加快产品迭代速度，而迭代速度的加快也会让一些固定成本，例如为达到发布质量所需的测试成本，以及产品发布流程所需承担的成本，在每个迭代中所占的比例突显。这种事务性成本在很大程度上阻碍了产品的迭代速度。乔梁老师在书中对持续交付“价值探索环”与“快速验证环”中每个步骤的详细拆解，一方面会让你开阔对业务问题的思考维度与角度，另一方面也能让你发现，在日常工作中就有很多细节可以优化，从而降低迭代中的固定成本，有效提升产品创新效率。与此同时，也会在潜移默化中提高团队的整体战斗力。

《持续交付2.0》是乔梁老师在大量实战项目中总结出来的理论框架，可谓经历千锤百炼。本书也许没有太多高深抽象的概念，但每一项原则和方法的后面，都有足够多的正反

例支持，和乔老师的为人一样，低调、务实、简单，值得每一位钻研和应用高效产品研发运营体系的读者阅读体会。我相信，在互联网行业，效率代表未来。

曾宇
腾讯副总裁

自序

2002年，我偶然得到一本书，名为《解析极限编程》。书中介绍的软件开发方法与现实中使用的工作方法截然不同。书中的很多实践看上去都不现实，如测试驱动开发、持续集成、结对编程、用户故事等，这让我感到很新奇，怎么会有团队这么做呢？但看上去这些方法的确很诱人，于是我带着“怀疑”的态度，在实际工作中引入了其中一些方法，但执行上还是打了一些折扣。例如，我没有做测试驱动开发，而只是增加一些单元测试；没有做结对编程，而是要求代码评审（code review）；没有做持续集成，而是每日构建。一段时间后，虽然能感受到一些收益，但并没有那么显著。

直到2005年，我的一个朋友向我展示了他们如何使用这种开发方式交付真实的软件项目，和真实的编写代码的过程。每一次修改代码，都编写并执行一系列的自动化测试用例；每次提交都会进行持续集成。这是一种从未有过的编码体验，开发工程师很少需要启动程序，通过单步调试来找出代码中的问题。这使我真正相信，的确存在按照这种敏捷方式工作的团队，而且离我并不遥远。

2007年，我加入ThoughtWorks，希望能体验敏捷软件开发方式。作为一名需求分析师和交付经理，我加入了持续交付平台GoCD的产品研发，我的搭档就是Jez Humber（该产品的产品经理），他也是《持续交付：发布可靠软件的系统方法》的作者之一，书中很多实践都来自我们团队自己的软件产品研发过程。从想法的诞生到产品上架，我经历了一个完整的产品研发过程，也真正认识了敏捷开发方法，掌握了持续交付实践。

2009年以后，我作为外部顾问或内部教练，开始为国内外很多企业提供相关的组织敏捷与精益转型咨询服务，客户既有PC互联网时代的巨头，也有传统IT企业；既有国内知名大企业，也有高速成长的移动互联网创业公司。在与客户合作的过程中，我对“持续交付”有了更深刻的理解，也对如何帮助组织实现“持续交付价值”有了全新的认识。

2007年，我认为包括极限编程在内的众多敏捷开发实践是快速高质量软件交付的法宝；2010年之后，我发现实践本身虽然非常重要，但更重要的是支撑实践的组织管理方法、工作思路与理念。于是，我的口头禅成了“别提敏捷，只解决问题！”。2012年后，更多的软件开发方法与敏捷流派在国内开始盛行，但其背后的核心理念与主要工作原则并没有根本性的变化。无论什么样的方法，都应该以“解决问题”为出发点，而“解决问题”的一

一个重要前提是“能够正确定义问题，并达成共识”。

我当然不是思想无用论的支持者。相反，我认为思想对每个人对事物的认知和理解至关重要。但咨询经历告诉我，对事物的正确理解，并不能确保正确的思想和理念在现实中落地，也不能确保对企业有大的和直接的帮助。对方法应用者而言，其目标是通过对其思想理念的认知，能够尽早解决自己（或者客户）所面临的棘手问题。

正如企业经营管理一样，软件工程发展的历程也是各种方法论不断出现与发展的过程。从20世纪60年代“软件工程”这一术语的诞生，到20世纪70年代提出瀑布软件开发模型，以及1985年提出的迭代增量开发和1986年Barry Boehm在“A Spiral Model of Software Development and Enhancement”一文中提出的喷泉模型，20世纪90年代的软件能力成熟度模型集成（Capability Maturity Model Integration, CMMI）的产生和多种轻量级软件开发方法，21世纪初敏捷宣言的正式发表，再到精益软件开发方法、看板方法，以及持续交付和DevOps运动。所有这一切变化，既反映出该领域的快速变化，也反映出没有哪一种理论或方法能够完全解决这个领域面临的所有问题。

本书希望能够让读者在了解“持续交付”全貌的基础上，当遇到与IT组织效能相关的问题时，能够以适当的思考方式和背景知识来应对，让你在今后的工作中少走一些弯路，至少遇到相似问题时，可以有所参考。

前 言

从“软件工程”这一名词诞生以来，“质量”和“效率”就是它的目标。IT组织大都在这条路上探寻，从最初的瀑布模型，到CMMI，很多组织曾经尊其为软件开发过程的“圣经”。而当“敏捷运动”兴起时，他们想要“做”敏捷；当听说“持续交付”，他们想要“做”持续交付。现在，DevOps也来了！在各种各样的交流大会里，不断传来DevOps胜利的凯歌，各种媒体也在报道它的好处。很多公司又想要“DevOps”了……

我们的确听到过一些美妙的“故事”，但它们可能都不属于我们自己。在自己身边，就连“如何让大家对这些理念或实践达成共识”都成了一大困难，这令你感到无比困扰。就像走在一团迷雾之中，耳边一直听到美妙的音乐响起，也隐约看到远处的点点亮光，然而脚下的“路”却忽明忽暗。

多年工作经历让我对这一领域有了新的认识，并进行总结与反思。“持续交付”是一个非常有吸引力的名字，总会让人浮想联翩，业务人员似乎看到了一丝希望“所有的需求，上午提出来，下午就能拿到手”。然而，太多的企业低估了自己所面临的困难。这些困难一部分是显性的，如没有自动化测试，也做不到自动化部署，主干开发更是不可想象；还有一部分困难是隐性的，例如，职能部门之间的“墙”存在已久。业务人员嫌开发团队的软件交付速度慢，开发团队嫌业务人员提出的需求不靠谱。这很可能归因于每个人的价值思考方式。

本书的目标是希望企业中所有角色转换价值思考的角度，改进软件服务端到端的商业价值交付方式，提升相关人员之间的协作效率，最终达到以安全可靠的方式快速验证想法，缩短实现真正商业价值的时间。也就是说，本书不仅关注“从需求列表到可运行的软件”这一过程，还提出“价值探索—快速验证”双闭环，如图0-1所示，这也是本书的书名“持续交付2.0”的由来。

事实证明，没有放之四海皆准的企业管理解决方案，能够完美解决每个企业遇到的问题。但是，管理者只有从整体视角出发，抵住局部优化的诱惑，才能在资源有限的情况下，引领企业创造更大的价值。本书提供了一个整体框架，给出了这个框架中各节点所涉及的原则与相关的实践方法，同时介绍了它们的优势与约束。

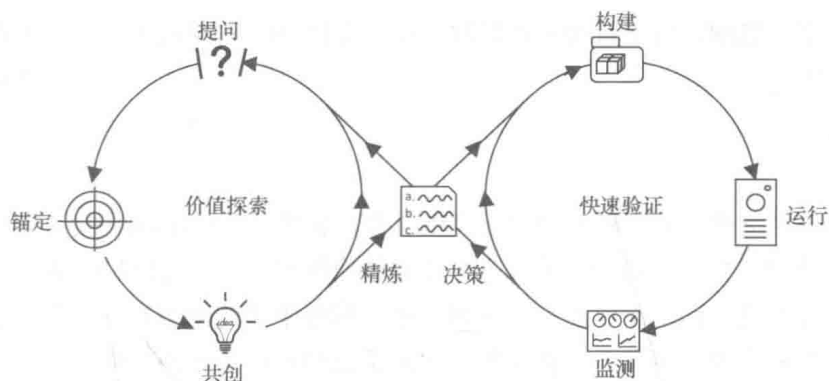


图0-1 持续交付双闭环模型

如果你将“持续交付2.0双环模型”应用到整个企业范围，就是一种企业级的组织管理变革指引；如果你将它引入某一个团队，对这个团队来说，就是团队工作模式的改进套路。既然“持续交付2.0”是一个管理框架，企业势必要根据自己的实际情况来进行定制。因此，书中列举了很多实际案例，告诉你，其他企业或团队如何应用这些实践方法，达到它们的目标。这些案例也说明，解决方案与实施路径很难在企业之间进行复制，企业必须应用书中的原则，结合自身的实际情况（产品形态及所处的商业竞争阶段、团队的规模与人员技能水平、软件系统架构，以及组织管理机制与文化等），逐步探索出自己的道路。

读者对象

本书主要服务于那些身处IT业务一线的管理者，或即将成为一线管理者的骨干技术人员，当然也包括那些从事软件产品项目管理和软件过程改进的人们。对新公司的创立者或高速发展的成长期公司技术高管也有参考作用。它并不是为那些已经成熟的大型软件企业的高层管理者服务的。当然，如果他们也认为本书有所帮助，那我也非常高兴。

对于产品经理、开发经理、测试经理和运维经理，可以从本书中获得更全面的工作视角，发现自己领域之外更多的信息，以及如何与其他角色协作共赢。

对于过程改进者或者变革者，可以从本书中了解其他团队或公司的做法，希望能给你带来工作上的灵感。

对于新公司的创立者与高速成长公司的高管，希望可以从本书中找到一些管理方式与高效方法，使得公司在成立和发展之初，就能够以尽可能少的成本，支持你的事业持续高速发展。

书中有很多案例，用于帮助读者理解“持续交付2.0”的双环模型，与其中各环节应用的不同实践。

内容简介

在进入正文之前，先谈谈本书的结构和内容。全书共包括3部分内容。第一部分介绍了“持续交付2.0”的双环模型；第二部分主要讨论使用“持续交付2.0”框架中可能遇到的问题，以及改进过程中需要遵循的原则。第三部分主要是案例分享，目的是让读者体会在持续改善过程中，不同企业和团队的实施重点与解决方案的不同。书中具体内容如下所示。

第一部分主要讲述“持续交付2.0”双环模型（即持续交付“8”字环）和“持续交付2.0”的4个工作原则，还会介绍两个闭环“价值探索”与“快速验证”的执行步骤与相关原则。

- 第1章讨论持续交付的发展必然性，并介绍“持续交付2.0双环模型”及其4个基本原则。
- 第2章讨论“价值探索环”（简称“探索环”）的4个核心环节，以及每个环节的指导原则与实践方法。只有业务方能够以“精益”方式思考，持续交付才能更显威力，否则很可能退缩成为持续交付1.0的单环模式，即只有“快速验证环”。
- 第3章简单阐述“快速验证环”（简称“验证环”）中各环节的主要活动，并给出各环节的工作方法。

第二部分主要阐述“持续交付2.0”的实施七巧板中，三大主要板块的工作原则与实践方法。这三大板块包括组织机制、软件架构与基础设施。其中组织机制是一个复杂课题，本书仅讨论持续交付所需的文化，以及建立文化的四步法。组织架构、人才结构、激励机制等内容将在本书的续篇中专题讨论。基础设施部分是产品研发过程中最基础的工作。这部分首先讨论持续交付部署流水线及其工具设计原则，然后分别介绍部署流水线的建立与优化必须关注的五大领域，也就是说，业务需求协作流程、分支与配置管理、构建与环境管理、自动化测试管理，以及部署发布与监控管理。

- 第4章讨论持续交付能力的提升需要企业具有信任、安全和持续改善的组织文化，并介绍丰田公司和谷歌公司用过的改善组织文化四步法。
- 第5章讨论软件架构对实现持续交付快速验证能力的重要性、有利于持续交付的软件架构特征，以及软件架构改造的3种方式，即“拆迁”“绞杀”和“修缮”。
- 第6章讨论如何利用约束理论和精益思想，发现流程中的瓶颈，使各角色之间的业务需求协作更加顺畅，提升需求流动速度。
- 第7章讨论快速验证环所依赖的部署流水线的设计原则与工具链建设草案。

- 第8章讨论代码仓库的分支方式对持续交付的重大影响，以及不同分支方式下部署流水线的建设方案。最后介绍代码分支的数量对发布策略的影响。
- 第9章回顾持续集成的历史，并讲解如何判断团队是否在实践中“持续集成”，还给出企业实施持续集成的五大步骤。
- 第10章讨论软件发布以前制订自动化测试策略需要考虑的因素，还介绍持续集成对自动化测试用例的编写与运行要求。最后，为了提高自动化测试的投资回报率，团队如何为遗留系统编写和增加自动化测试用例。
- 第11章讨论软件配置管理，它是持续交付快速验证环的基础。对代码、配置、环境、数据做好配置管理，最终实现一键部署和一键测试，让各角色在协作过程中能够全部实现自动化服务，并且互不影响，解放人的大脑和双手，做更有价值的事情，而让机器做它擅长的事——不断地重复。
- 第12章讨论降低生产部署与发布风险的技术与方法。
- 第13章讨论软件在运行时，数据收集与分析的重要性，以及衡量数据监测环节的衡量指标，包括正确性、完整性和及时性，此外，还介绍测试扁平化趋势，以及生产环境上的质量巡检与演习。

第三部分主要是实战案例的分析。它们分别代表不同类型的公司、不同大小的团队以及不同的软件产品特点。我将带你深入案例现场，了解当时状况，分析问题，并提出解决思路。

- 第14章介绍一个百人工程师的互联网产品团队历经一年时间，如何打造快速运转的“持续交付2.0双环模型”，并且做到可持续运转。
- 第15章介绍在无法“测试右移”的情况下，一个大项目中的小团队如何改变团队协作模式，从“死亡行军”转变为“无缺陷交付”。
- 第16章介绍一个微服务化开发团队如何在项目运行的过程中，通过逐步对基础设施板块中各模块进行改造，提升交付质量与频率，并推动运维人员也做出改变，真正成为一个“DevOps团队”。

阅读方法

如何阅读本书取决于读者的目标是什么。你可以按照章节从头到尾读下来。这样，你可以了解我在工作中应用到的知识体系，最终了解为什么“持续交付2.0是一种组织能力”，如何衡量这种组织能力，以及如何在具体的工作场景中运用相关原则解决具体的问题。

当然，读者也可以根据自己的兴趣以及工作中遇到的问题，选取不同的章节。例如，后面3章的具体案例分析包括小团队的改进案例、大团队的变革案例，甚至超大公司所用

的一些工作方法与工具。这样，读者也许可以为明天将要进行的艰难说服工作找到一丝破局的思路。

无论怎样，我都非常重视你的反馈。让我知道你对这本书有什么想法——你喜欢什么？你还希望知道哪些内容？你可以扫描下面的二维码，关注“持续交付2.0”的微信公众号，或者访问www.continuousdelivery20.com网站查看相关的延伸阅读内容。



致 谢

很多人为本书做出了贡献。首先要感谢编辑团队给予我的大力支持，尤其是杨海玲主编的敬业精神打动了我，让我有勇气将自己多年咨询过程中曾经使用的方法与实践写下来，与更多人分享。尽管它们并不完美，但也算是对自己过去工作的总结。其次，感谢我的朋友任发科对本书诸多章节的审校，以及提出的宝贵意见。

另外，我要感谢我在ThoughtWorks工作时的同事，尤其是GoCD团队和Jez Humble。从他们那里，我真正领会了敏捷与精益思想，并在后来的工作中熟练应用持续交付领域的诸多技术实践，帮助我的客户解决实际问题。还要感谢我的朋友王鹏超，他就是让我认识到极限编程独特魅力的那个人，他目前在Facebook公司工作，负责Cassandra的开发。

最后，感谢我的妻子霞和儿子天，我的挚爱。我一度想放弃本书的写作，是他们的鼓励才让我走出低谷。而且，霞不厌其烦地帮助我修订文稿，提出改进建议，帮助我提炼总结，使得本书结构更加合理。

服务与支持

本书由异步社区出品，社区 (<https://www.epubit.com/>) 为您提供相关资源和后续服务。

提交勘误

作者和编辑尽最大努力来确保书中内容的准确性，但难免会存在疏漏。欢迎您将发现的问题反馈给我们，帮助我们提升图书的质量。

当您发现错误时，请登录异步社区，按书名搜索，进入本书页面，点击“提交勘误”，输入勘误信息，点击“提交”按钮即可。本书的作者和编辑会对您提交的勘误进行审核，确认并接受后，您将获赠异步社区的100积分。积分可用于在异步社区兑换优惠券、样书或奖品。



The screenshot shows a web form for submitting勘误 (勘误). At the top, there are three tabs: '详细信息' (Detailed Information), '写书评' (Write a Review), and '提交勘误' (Submit勘误), with the last one being active. Below the tabs, there are three input fields: '页码:' (Page Number), '页码位置 (行数):' (Page Position (Line Number)), and '勘误内容:' (勘误内容). Below these fields is a rich text editor with a toolbar containing icons for bold (B), italic (I), underline (U), strikethrough (ABC), bulleted list, numbered list, link, and unlink. At the bottom right of the form, there is a '提交' (Submit) button.

扫码关注本书

扫描下方二维码，您将会在异步社区微信服务号中看到本书信息及相关的服务提示。



与我们联系

我们的联系邮箱是contact@epubit.com.cn。

如果您对本书有任何疑问或建议，请您发邮件给我们，并在邮件标题中注明本书书名，以便我们更高效地做出反馈。

如果您有兴趣出版图书、录制教学视频，或者参与图书翻译、技术审校等工作，可以发邮件给我们；有意出版图书的作者也可以到异步社区在线提交投稿（直接访问www.epubit.com/selfpublish/submission即可）。

如果您是学校、培训机构或企业，想批量购买本书或异步社区出版的其他图书，也可以发邮件给我们。

如果您在网上发现有针对异步社区出品图书的各种形式的盗版行为，包括对图书全部或部分内容的非授权传播，请您将怀疑有侵权行为的链接发邮件给我们。您的这一举动是对作者权益的保护，也是我们持续为您提供有价值内容的动力之源。

关于异步社区和异步图书

“异步社区”是人民邮电出版社旗下IT专业图书社区，致力于出版精品IT技术图书和相关学习产品，为译者提供优质出版服务。异步社区创办于2015年8月，提供大量精品IT技术图书和电子书，以及高品质技术文章和视频课程。更多详情请访问异步社区官网<https://www.epubit.com>。

“异步图书”是由异步社区编辑团队策划出版的精品IT专业图书的品牌，依托于人民邮电出版社近30年的计算机图书出版积累和专业编辑团队，相关图书在封面上印有异步图书的LOGO。异步图书的出版领域包括软件开发、大数据、AI、测试、前端、网络技术等。



异步社区



微信服务号

目 录

第 1 章 持续交付 2.0	1	2.4.6 最小可行产品法	32
1.1 软件工程发展概述	1	2.5 实施注意事项	32
1.1.1 瀑布软件开发方法	1	2.6 小结	35
1.1.2 敏捷软件开发方法	2	第 3 章 快速验证环	36
1.1.3 DevOps 运动	3	3.1 验证环的目标	36
1.1.4 持续交付 1.0	4	3.2 验证环的 4 个关键环节	37
1.2 持续交付 2.0	7	3.2.1 构建	37
1.2.1 精益思想	8	3.2.2 运行	38
1.2.2 双环模型	9	3.2.3 监测	39
1.2.3 4 个核心原则	11	3.2.4 决策	39
1.2.4 持续交付七巧板	12	3.3 工作原则	39
1.3 小结	13	3.3.1 质量内建	39
第 2 章 价值探索环	14	3.3.2 消除等待	40
2.1 探索环的意义	14	3.3.3 重复事务自动化	43
2.2 探索环的 4 个关键环节	15	3.3.4 监测一切	43
2.2.1 提问	16	3.4 小结	44
2.2.2 锚定	17	第 4 章 持续交付 2.0 的组织文化	45
2.2.3 共创	19	4.1 安全、信任与持续改善	45
2.2.4 精炼	22	4.1.1 失败是安全的	45
2.3 工作原则	24	4.1.2 相互信任	45
2.3.1 分解并快速试错	24	4.1.3 持续改善	46
2.3.2 一次只验证一点	25	4.2 文化塑造四步法	46
2.3.3 允许失败	26	4.2.1 行为决定文化	46
2.4 共创与精炼的常用方法	27	4.2.2 谷歌的工程师质量文化	48
2.4.1 装饰窗方法	27	4.2.3 Etsy 的持续试验文化	49
2.4.2 最小可行特性法	29	4.3 行动原则	50
2.4.3 特区法	30	4.3.1 价值导向	51
2.4.4 定向探索法	30	4.3.2 快速验证	51
2.4.5 稻草人法	31	4.3.3 持续学习	51