



白皮书：企业 DevOps 的成功之路

202302 版本

作者:

Koichiro (Luke) Toda

President of Strategic Staff Services Corporation

Director of TPS Certificate Institution

联系方式: lktoda@ask3s.net

Nobuyuki Mitsui

CTO of Strategic Staff Services Corporation

联系方式: nmitsui@ask3s.net

译者: 刘颀, 史鹏程

审校: EXIN, 刘征

再版修订: 许峰, 王映红, 张楠

www.exinchina.cn

Copyright © EXIN Holding B.V. 2023. All rights reserved.

EXIN® is a registered trademark.

DevOps Master™ is a registered trademark.

No part of this publication may be reproduced, stored, utilized or transmitted in any form or by any means, electronic, mechanical, or otherwise, without the prior written permission from EXIN.

目录

序言	4
什么是企业体系的 DevOps?	6
DevOps 的目标是什么?	6
DevOps 的知识体系	7
DevOps 团队角色	9
推荐文献	18

企业 DevOps 的 成功之路

利用 DevOps 使 IT 服务支持业务

序言

本书作者一直提供敏捷开发和丰田/全面管理系统（TMS）的指导服务，其中 TMS 是丰田生产系统（Toyota Production System, TPS）的核心。基于我们的经验，我们相信 DevOps 能很好的支持业务。本白皮书使用案例研究来帮助解释 DevOps 框架的使用。

DevOps 不仅支持 IT，还可以用于更多领域。DevOps 还能用于支持业务战略并改善业务流程。

关于 DevOps 的书籍非常多。以下三本书对于理解什么是 DevOps 有用：

- 《**凤凰项目：一个 IT 运维的传奇故事**》（Kim et al., 2018）¹是从 IT 经理的角度来编写的。
- 《**持续交付：发布可靠软件的系统方法**》（Humble & Farley, 2010）是从开发和项目经理的角度来编写的。
- 《**DevOps: A Software Architect's Perspective**》（Bass et al., 2015）是从架构师的角度来编写的。

本白皮书从业务流程角度来编写的，因为 DevOps 旨在通过建立软件/IT 服务供应链（Software/IT service supply chain）来支持业务并管理整个流程以达到成熟。

¹ 《凤凰项目：一个 IT 运维的传奇故事》该书中文版已经上市出版。

DevOps 的历程——案例研究

我们的客户提供高档次的网络转换服务：从固定工作站到移动设备。转向 DevOps 的旅程始于 2009 年，就在该客户成功实施了敏捷、Scrum 和 XP 方法之后。

当时客户的 Scrum 团队已能做到更快地开发和发布软件。即使开发时间减少了一半，业务总监依然担忧业务速度并未得到提升。一开始，看似开发过程是瓶颈。经过调查发现开发过程并非瓶颈，相反是业务流程应该被改进。

TMS 这一概念，在从业务战略与规划一直到客户服务的整个业务流程得以实施。这有助于：

- 建立一个精简化 (streamlined) 的业务运营过程
- 并利用 DevOps 概念缩短交付前置时间 (delivery lead-time) 。

该项目于 2012 年成功完成。整个流程从端到端进行了重新调整，并使用可视化控制、单件流 (one-piece flow)²、每周进程同步、每天反馈循环以及 Kaizen³建立起了整个业务的协作。经理、管理员、销售、设计师、程序员、运维和客服形成了一个团队，大家在可视化看板上共享所有业务信息。

实施之后，业绩得到了显著提升：

- 交付前置时间缩短
- 销售量提升
- 利润率提高
- 员工积极性增强

这些是 DevOps 的真正价值。DevOps 框架应该直接支持到业务结果，不仅仅是通过 IT 服务中开发与运维的协同，而是企业使用 IT 服务来支持并提升其业务。

DevOps 的价值应以业务结果来衡量，而不是根据 IT 项目范围和 IT 成果来评判。

² 译者按：One-piece flow 是 70 年代日本的大野耐一在 JIT 生产方式的基础上提出的一种方法，目的是减少单纯依赖大量的在制品和零部件储备来维持均衡生产。

³ 译者按：日语，持续改善，丰田所贯彻的方法建立起了整个业务的协作。请参考术语表。

什么是企业体系的 DevOps?

关于 DevOps 的书有很多，但不幸的是，大多数都是描述在网站和产品开发中是如何应用 DevOps 的。关于 DevOps 如何用于企业体系的资料非常少。

企业往往同时拥有交互型系统⁴（SoE）和记录型系统（SoR）。

- SoE 关注的是速度。
- SoR 关注的是业务连续性。

而问题在于如何使 SoR 快速适应 SoE 的变更。Gartner 公司把这称为双峰挑战（Bimodal challenge）⁵。

大多数企业的 SoR 在遗留的应用程序或系统上处于困境，可以使用 DevOps 进行修正，及时制（JIT）概念建立精简流程。

DevOps 并非单一的工具、方法、技能或组织结构，DevOps 的框架是结合所有这些元素来建立一个精简流程，使业务更快地运营，并能更快地应对变化。DevOps 还可以通过戴明博士的 Plan-Do-Check-Act（PDCA）来提升其成熟度。

企业级的 DevOps 不仅仅是对敏捷开发和持续交付的增强，同时也是对 IT 服务管理和应用程序管理的提升，从而实现和促进业务增长并保障业务连续性。

DevOps 的目标是什么?

DevOps 的目标是建立精简化的及时制（JIT）的业务流程。DevOps 旨在通过调整及时制（JIT）业务流程将业务成果最大化，例如增加销售和利润、提高业务速度或尽可能降低运营成本。

DevOps 意味着在业务中建立了一条 IT 服务供应链，与其它产品的供应链嵌入业务的方式相同。这是一个从软件交付到提供 IT 服务的模式转变。

从架构的角度来看，DevOps 必须建立一个自动化的快速部署系统。有很多方法和工具可以利用。DevOps 没有统一的实施模板。每个组织都应该考虑并建立自己的 DevOps 流程来提升业务。因此，理解 DevOps 的概念对遵循正确的流程来说是至关重要的。

⁴ 交互型系统中文翻译参考 IBM 文章《解析微服务架构一》。

⁵ 双峰的定义 - Gartner 信息技术术语表。（日期不详）。Gartner.

<https://www.gartner.com/en/information-technology/glossary/bimodal>

DevOps 的知识体系

当实施 DevOps 时，有很多知识源、方法论、实践案例和工具供选择。DevOps 由三大支柱和一个基础组成。

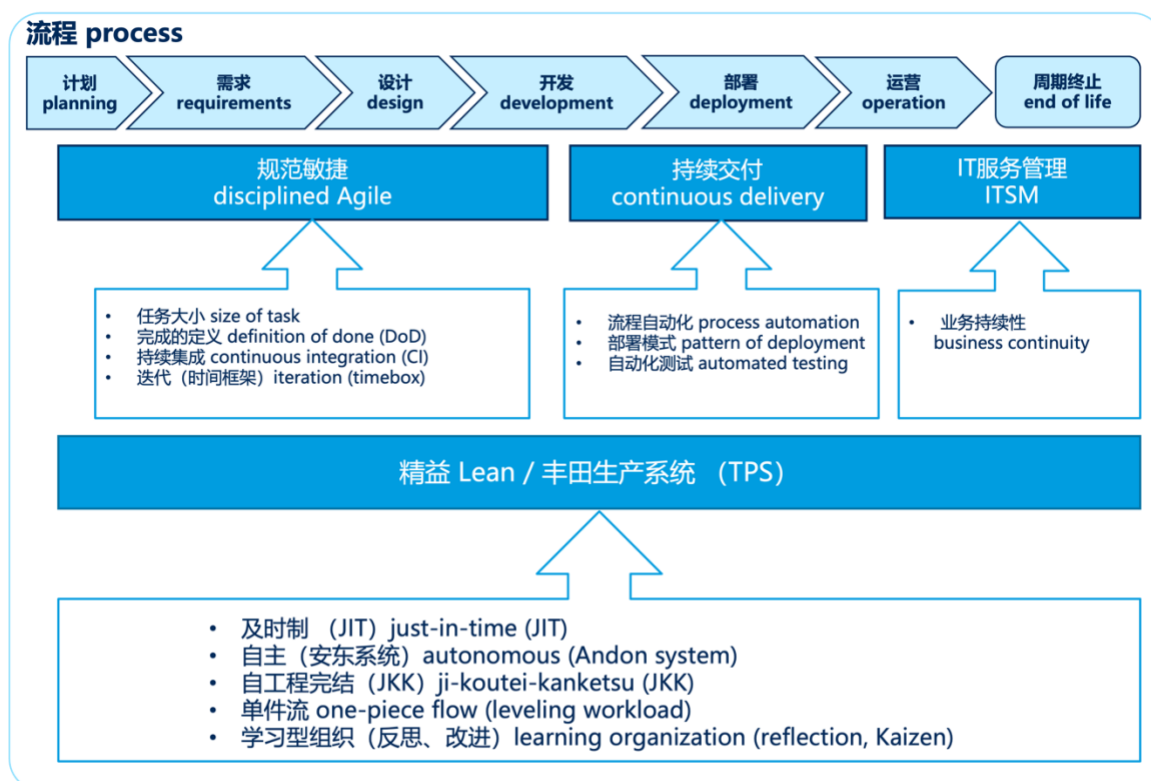
三大支柱为：

- 规范敏捷 (Disciplined Agile)
- 持续交付 (Continuous Delivery)
- IT 服务管理 (IT service management, ITSM)

一个基础为：

- 精益 (Lean) / 丰田生产系统 (TPS)

图 1 DevOps 知识体系



Copyright © SSS Corporation 2015

规范敏捷 (Disciplined Agile)

一支训练有素的敏捷开发团队是成功实施 DevOps 的关键。规范敏捷意味着：

- 速度稳定 (Stabilized Velocity)
- 适应变化 (Adaptability to change)
- 发布优质的无错误代码 (Release of high quality, bug free code)

IT 服务需要以更频繁、更快速的发布周期来响应业务变化，这取决于开发速度。工作质量应该是最重要的。这可以通过将工作分割为小任务来实现。

持续交付 (Continuous Delivery)

持续交付是应用程序的构建、部署、测试和发布流程的自动化实施。一个关键的关注点是测试，如验收测试和性能测试等。TPI NEXT (测试流程优化) 可以用于提高这个过程的成熟度。

每个组织部署流水线 (Deployment Pipeline) 的实现都会有差异，因为软件发布的价值流不同。成功的关键因素 (KSF) 在于为 IT 服务构建单一的部署流水线。

IT 服务管理 (IT service management, ITSM)

由于技术是大多数业务流程的核心环节，IT 服务的连续性或高可用性是整个业务存亡的关键因素。这可以通过引入降低风险措施和恢复方案来实现。就像 IT 服务管理所有要素都提及的，成功实现服务的连续性需要得到高层的承诺以及组织中所有成员的支持。要保持恢复方案的有效性，持续维护是至关重要的。服务连续性是服务功效 (warranty, fitness for purpose) 的必要组成部分。如果服务连续性无法按照业务的要求维护和/或恢复，那么业务将无法实现所承诺的价值。

传统的 IT 服务管理 (ITSM) 最佳实践，比如 ITIL，看起来很繁琐，不匹配 DevOps 中所倡导的快速流程。有必要考虑一下如何降低管理工作量。

基于 DevOps 去重新调整 ITSM 是有必要的，创建轻量级的只包含所需最少必要信息 (Minimum Required Information, MRI) 的、严格聚焦于业务持续性的轻量级 ITSM。每个组织的 MRI 设置取决于他们的业务。⁶

以丰田生产系统 (TPS) 与精益 (Lean) 理念为基础

建立一个流水线式的 IT 服务供应链并不容易，因为其包含的元素众多，并且有必要改变我们的思维方式，脱离熟悉的开发周期及其方法论。TPS 的概念包括及时制 (JIT) 和自动化能够起到作用。JIT 意味着以单件流 (one-piece flow) 的方式建立一个精简化的供应链。而自动化意味着尽可能实现自动化，且在出现缺陷时能停止整个流程。

这个过程必须为这两个概念来设计，并对员工进行相应的教育。另一个关键问题是开发和运维的管理周期。这需把工作方式转换至敏捷方法，包括开发与运维之间每周或每天的信息同步。

Ji-Koutei-Kanketsu (JKK) 概念

Ji-koutei-kanketsu (JKK) 概念认为，100%地完成每个条目，是有助于保持高质量工作的。而“完成定义” (Definition of Done, DoD) 或“结束” (Completion)，对每个人都必须清楚定义。

产品负责人需要改变他/她的使命，不单单是管理待办事项列表 (Product Backlog)，也需要规划 IT 服务的运维成本，在丰田，这工作是由首席工程师来完成的。

⁶ 请参考 [DevOps 轻量级 IT 服务管理工作模板](#)

DevOps 团队角色

为了保证 IT 服务的业务连续性，推荐在组织中建立 DevOps 团队。小型的 DevOps 团队可以依据亚马逊的“两个披萨规则”：两个披萨饼就可以喂饱的团队。

团队成员包括：

- 流程主管 (Process Master)
- 服务主管 (Service Master)
- DevOps 工程师 (DevOps engineer)
- 把关人 / 发布协调员 (Gatekeeper / release coordinator)
- 可靠性工程师 (Reliability engineer) (可选)
- 开发团队 (Development team)
- 运维团队 (Operations team)

流程主管 (Process Master)

流程主管领导并引导团队，这个角色类似于在 Scrum 中的 Scrum Master。对整个过程实施可视化管控，并特别注重建立单件流的流水线式的流程。可视化管控意味着“在不需要解释的情况下，通过看板⁷是否每个人都能很容易的理解当前的状况？”该看板并不显示状态。它可以展示是否有问题出现。

所需经验：Scrum Master、敏捷项目领导 (Agile Project Leader)

服务主管 (Service Master)

服务主管对及时地 (JIT) 提供 IT 服务负有全责。这个角色就类似于 Scrum 中的产品负责人 (Product Owner)，对待办事项 (product backlog) 进行管理并对其排序。服务主管还负责 IT 服务的成本规划。

所需经验：Scrum 产品负责人 (Scrum Product Owner)、服务负责人 (Service Owner)

DevOps 工程师 (DevOps engineer)

DevOps 工程师以优化和维护自动化流程为主要使命。工程师将检查整个自动化过程及其工具。DevOps 流程需要很多工具。

所需经验：开发 (Development) 和工具 (Tools)

把关人 / 发布协调员 (Gatekeeper / release coordinator)

负责监控 IT 服务的运维状态和下一次发布的进展。做关于部署是做或不做的决定，需要参照的标准包括安全性、合规性、监管要求、运维团队的成熟度以及他们的流程观念。

所需经验：IT 服务管理 (IT service management)、运维 (Operations)

⁷ “看板”可以是任何一种类型的板：Scrum 板、Kanban 板、数字板、实体卡片墙，或任何其他可视化工作状态的手段。

可靠性工程师 (Reliability engineer) (可选)

监控部署过程中服务的测试质量，处理服务运行中所产生的问题。监控流程状态以确保开发团队严格遵守了 CI（持续集成）和 CD（持续交付）的规则。监控并管理复杂的构建管线的工作流。以提升测试流程为使命。

所需经验：测试 (Testing) , 工具 (Tools) 以及质量保证 (Quality Assurance)

开发团队 (Development team)

DevOps 的关键成功因素之一是建立一个规范的敏捷团队。规范的敏捷团队致力于以可持续的速度来满足发布计划和发布质量。

所需经验：开发 (Development) 和敏捷 (Agile)

运维团队 (Operations team)

采用轻量级的 ITSM 并在总体战略范围内支持对服务的设计、实施、运维与改进。采用 TPS 中的“提前持续改善 (Kaizen in Advance) ” 的实践。

所需经验：运维 (Operations) 和持续改善 (Kaizen)

组织

在服务管理办公室（Service management office）中组建 DevOps 团队来支持服务主管是很有用的。两种类型的组织架构展示。

- 适合小型组织的**扁平化组织（Flat organization）**
 - 这是小型团队的基本架构。
- 适合大型复杂组织的**矩阵架构组织（Matrix organization）**
 - 建立专家池并安排他们作为一个团队给到服务主管。这个矩阵组织的想法来自丰田的首席工程师。

图 2 扁平组织

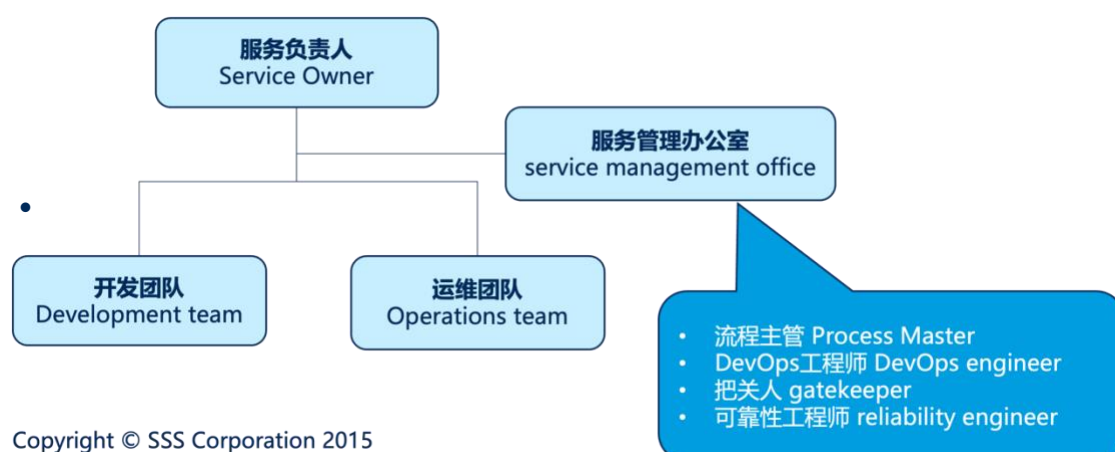
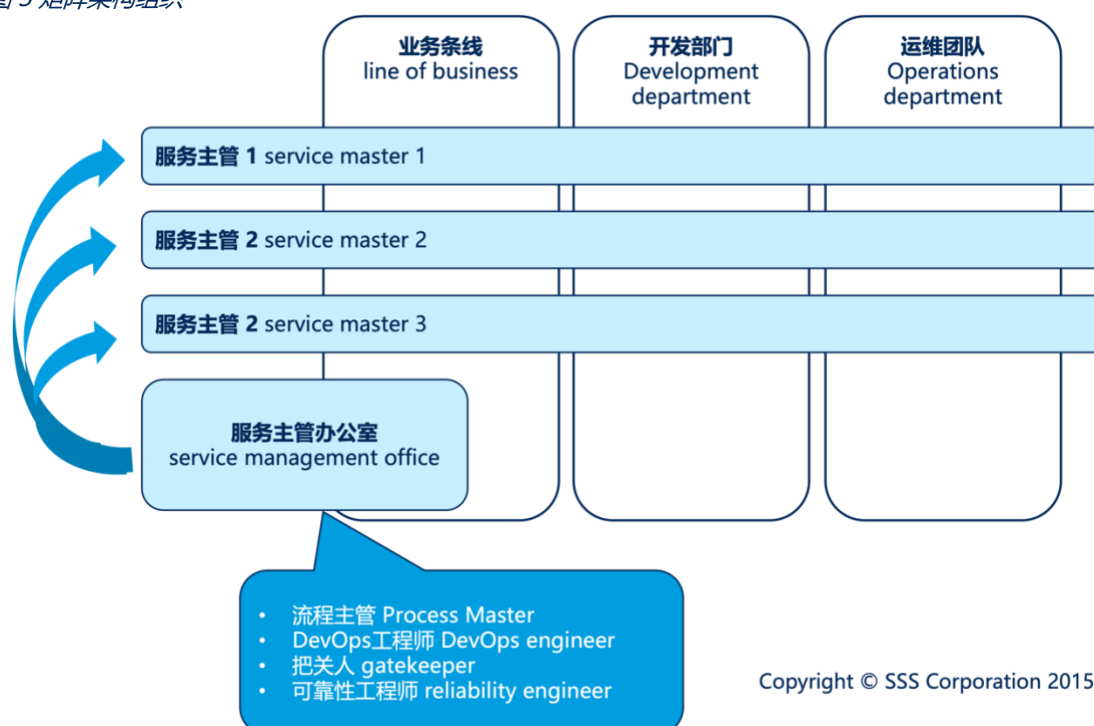


图 3 矩阵架构组织



DevOps 的流程

要建立一个流水线式的流程，采用 ji-koutei-kanketsu (JKK) 来指导 DevOps 团队是最有效的办法。JKK 是一种为高质量而工作的方式，它意味着清晰地理解目标，理解正确的工作方式，使工作百分百地正确完成，并在没有检查的情况下保证质量要求。

业务战略和规划 (Business Strategy and planning)

IT 服务与业务战略和规划有密切的关系。服务的主管应该参加业务规划会议并提出如何通过 IT 服务获得业务优势的建议。

市场营销 (Marketing and sales)

服务主管应该与市场部门讨论如何从 IT 服务中获得优势。

服务主管：

- 确定 IT 服务的客户
- 收集能够产生商业价值的需求
- 确定一个好的时间框架

管理 (Administration)

流程主管探索如何将整个流程实现可视化。一种方法是使用 Obeya，它可以对整个流程进行设置。

Obeya 这个“作战室”具有两个目的：

- 信息管理 (information management)
- 现场决策 (on-the-spot decision making)

这里面有很多可视化管理工具。团队成员可以快速查看项目中的方方面面。

当跨职能团队在一起工作时，Obeya 系统能够：

- 快速、准确做出决策
- 加强沟通
- 保持一致
- 加快信息的收集
- 形成重要的团队集体意识

项目规划 (Project planning)

服务主管组织服务管理办公室 (SMO) 并定义团队的基本规则。服务主管创建愿景和项目目标，决定项目的价值，然后整合 DevOps 的团队成员。

在项目规划阶段，运行时的基础设施运行中得以定义。整体流程的价值流图表 (VSM) 也在此阶段设计出来。

需求和设计 (Requirements and design)

在需求和设计阶段，有以下事项：

- 服务主管定义产品待办事项 (product backlogs) 和并安排优先级。
- DevOps 团队使用产品代办事项 (product backlog) 来定义故事 (story) 。
 - **用户故事**：角色，职能，业务价值/理由，以及运维条件。
 - **测试故事**：验收测试用例和服务验收标准。
 - **运维故事**：设置 IT 服务的优先级的和业务连续性的运维条件。创建服务级别和运维级别协议。
- DevOps 工程师和运维团队定义转换、测试和开发的基础设施。
- 开发团队创建发布和迭代计划。
- 把关人研究 IT 服务的合规性以及 IT 服务的监管要求。
- 可靠性工程师定义测试方法和测试用例。

开发 (Development)

下一个阶段便是服务的实际开发。在这个阶段，Scrum 是最适用的框架。开发团队必须承诺发布计划，然后使用规范化的敏捷方法工作。每个迭代（冲刺）的周期是根据业务需求来决定的。

从质量的角度来看，极限编程 (XP) 实践是有效的：

- 结对编程 (pair-programming)
- 测试驱动的开发 (TDD)
- 重构 (refactoring)
- 10 分钟构建 (10 minute builds)

开发阶段也从持续集成中受益。

部署 (Deployment)

在完成持续集成之后，自动化流程开始进行验收测试、性能测试和部署。

- DevOps 工程师应该建立单件流 (one-piece flow) 方式构建一个单一的自动化部署流水线 (pipeline) 。
- 可靠性工程师和 DevOps 工程师将共同改进测试流程。
- 把关人监控整个过程的进度，决定是否上线。
- 运维团队研究如何保持业务连续性。

运维 (Operations)

运营团队负责利用轻量级的 ITSM，在运营期间监控 IT 服务的状态。

发生灾难事件时，保持重要服务的运行是至关重要的。运营团队应该让可靠性工程师参与进来，并注意两个关键参数。

- 恢复点目标：系统可以恢复到之前的哪个时间点？
- 恢复时间目标：恢复系统需要多长时间？

维保 (Maintenance)

服务主管和可靠性工程师决定是否允许进行维保。经允许，维保活动将作为变更请求 (RFC) 添加到待办任务中。

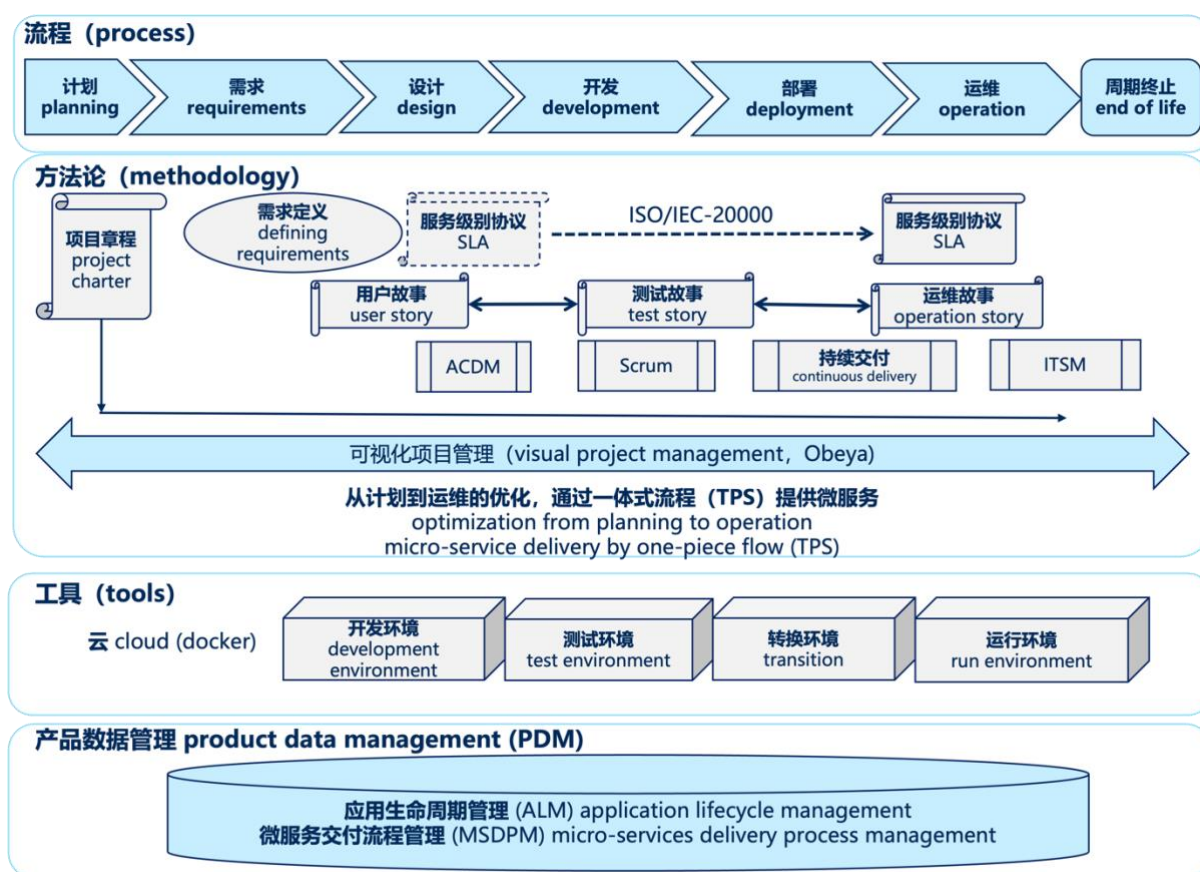
客户服务 (Customer service)

服务主管和可靠性工程师负责收集客户的反馈，例如包括用户体验和质量事件的运维问题。经允许，这些问题将作为变更请求 (RFC) 添加到待办任务中。

生命周期终止 (End of life)

服务主管将决定 IT 服务生命周期的终止条件，包括决定何时以及如何进行生命周期的终止。

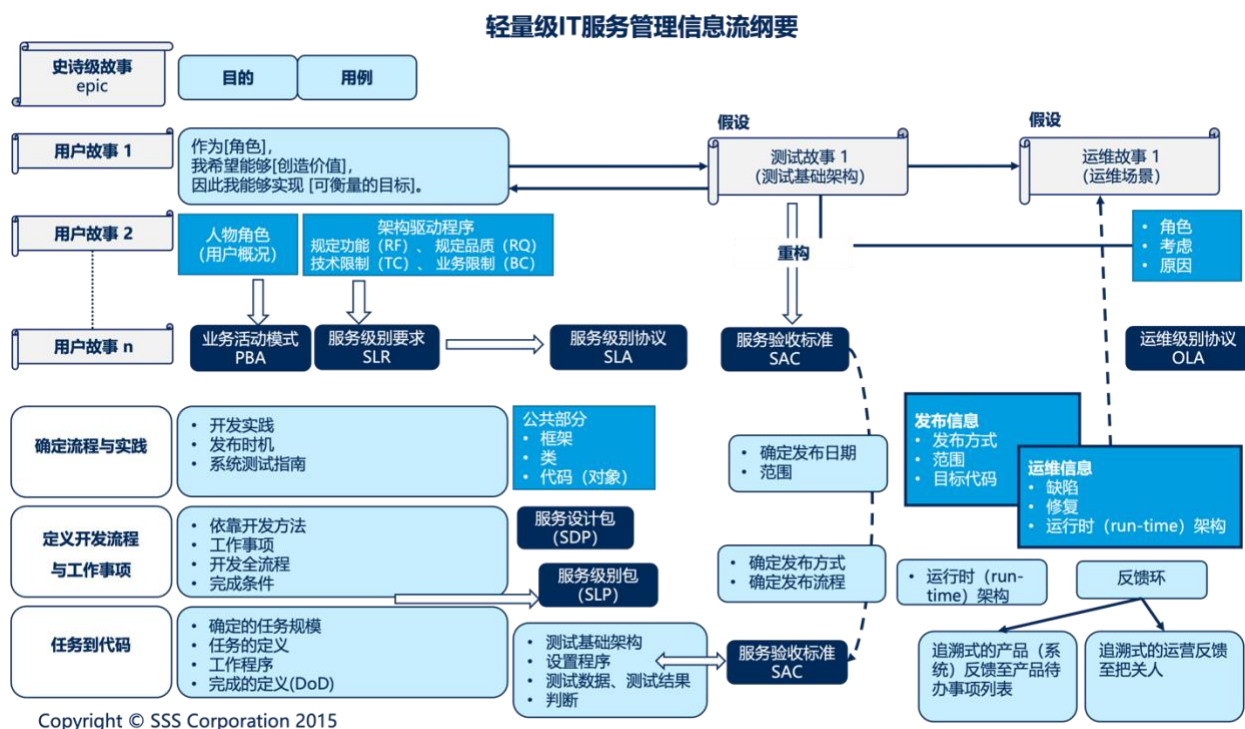
图 4 DevOps 的一个配置示例



Copyright © SSS Corporation 2015

译注：ACDM 为 Architecture-Centric Design Method，一种演进式软件架构设计方法。详见附录推荐文献。

图 5 轻量级的 IT 服务管理 (ITSM)



DevOps 的实施

DevOps 有三种实施方式。可以根据企业的业务模式进行选择。

- 丰田方式 (Toyota way) (先进而复杂)
- 协同方式 (Collaboration) (标准)
- 持续交付 (Continuous delivery) (基本)

丰田方式 (Toyota way)

这种方式重点在于关注 IT 服务战略，并给予业务的战略优势。这需要由业务负责人或服务主管来领导。在大型企业最好选择矩阵式管理组织架构，并且在 IT 战略和业务战略之间保持密切的关系。

这个结构很适合 IT 服务提供商 (IT Service Provider) 。

协同方式 (Collaboration)

这种方式将专注如何快速和频繁的提供 IT 服务并保障可靠运行，由服务主管来主导。

这种方式尤其适合需要将交互型系统 (SoE) 和记录型系统 (SoR) 联结在一起的企业。

持续交付 (Continuous Delivery)

这种方式侧重于快速和频繁的软件发布，由产品负责人主导。

它最适合数码产品提供商。

结论

显然 DevOps 对从大多数 IT 经验来说是一个整体的模式重大模式的转变。因此，关于 DevOps 的培训对员工来说十分重要。

这是您 DevOps 旅程的开始。Exin DevOps Master™这项认证培训，将有助于您理解 DevOps 并使您所在组织从中受益。

术语表

此术语表描述了我们在该白皮书行文中使用到的一些术语。

持续改善 (Kaizen)

持续改进意味着按照周、日循环进行 PDCA。为了找到根本原因问五次“为什么” (5-Why)。

问题需要通过数据的方式来定义和支持。是否每个人都能清晰地认识到问题？设置一个你发现的问题的假设，然后基于对策来思考并验证你的假设。

对策必须在基于日常活动进行定义，也需要设定每周的关键技术指标 (KPI)，这样人们可以感觉到一种成就感。

提前改善 (Kaizen in advance)

当下游环节的人员意识到有些问题可能来自上游，最优的方式是他们站在整体流程的角度来为解决问题创造可能的解决方案。然后他们可以向上游部门提出期望。这给问题提供了一个反馈循环。

Ji-koutei-kanketsu (JKK)

JKK 的概念是一种完美状态：

- 不要在你的过程步骤中做低质量的工作。
- 不要接受前一个流程步骤的错误产出。
- 不要把你的不良产出带给下一个过程步骤。

工作标准要求能够采用正确的方法去完成工作，这也意味着要定义一个衡量方法去决定是否继续进行下一步。

TPI Next

TPI Next 指的是业务驱动测试流程改进。

推荐文献

通用阅读

Kim, G., Behr, K., & Spafford, G. (2018). *The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win*. IT Revolution.

注：该文献目前已有中文版图书《凤凰项目：一个运维的传奇故事》译者：成小留。

推荐人群：流程主管、开发团队

Davis, J., & Daniels, R. (2016). *Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale*. Van Duuren Media.

推荐人群：开发团队、流程主管、服务主管、DevOps 工程师、可靠性工程师、把关人

Bass, L., Weber, I. M., & Zhu, L. (2015). *DevOps: A Software Architect's Perspective*. Addison-Wesley.

推荐人群：DevOps 工程师、开发团队、流程主管

Farcic, V. (2016). *The DevOps 2.0 Toolkit: Automating the Continuous Deployment Pipeline with Containerized Microservices* (1st ed.). CreateSpace Independent Publishing Platform.

推荐人群：DevOps 工程师、流程主管

Lattanze, A. (2009). *Architecting Software Intensive Systems: A Practitioners Guide. Software Development, Software Engineering, and Project Management*. CRC Press.

推荐人群：DevOps 工程师、运维团队、流程主管

Duffy, M. (2015). *DevOps Automation Cookbook*. Van Haren Publishing.

推荐人群：运维团队、把关人、流程主管

Behr, K., Kim, G., & Spafford, G. (2005). *The Visible Ops Handbook: Implementing ITIL in 4 Practical and Auditable Steps*. Information Technology Process Institute.

推荐人群：DevOps 工程师、可靠性工程师、开发团队、流程主管

Ewijk, A. van, Linker, B., Oosterwijk, M. van, Visser, B., van Oosterwijk, M., & van Ewijk, A. (2013). *TPI next: business driven test process improvement*. Kleine Uil, Uitgeverij.

推荐人群：通用阅读，执行层、经理、服务主管、流程主管

Liker, J. (2020). *The Toyota Way, Second Edition: 14 Management Principles from the World's Greatest Manufacturer* (2nd ed.). McGraw Hill.

备考 EXIN DevOps Foundation 认证的参考文献

Skrynnik, O. (2018). *DevOps: A Business Perspective* (1st ed.). Van Haren Publishing.

备考 EXIN DevOps Professional 认证的参考文献

Kim, G., Humble, J., Debois, P., Willis, J., & Forsgren, N. (2021). *The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations* (2nd ed.). IT Revolution Press.

备考 EXIN DevOps Master™认证的参考文献

Best, B. D., & Hemmen, L. V. (2022). *DevOps Continuous Assessment: Publication in the Continuous Everything series*. Leonon Media.

Humble, J., & Farley, D. (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Pearson Education.

PhD, F. N., Humble, J., & Kim, G. (2018). *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations* (1st ed.). IT Revolution Press



Driving Professional Growth

联系 EXIN

www.exinchina.cn

info.china@exin.com

WeChat ID: EXINCH