



DevOps Master 白皮书

企业 DevOps 的成功之路

作者：

Koichiro(Luke) Toda

Nobuyuki Mitsui

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

目 录

1. 整体介绍	1
1.1 前言	1
1.2 DevOps 的历程	1
2. 什么是企业体系的 DevOps?	3
3. DevOps 的目标是什么?	4
4. DevOps 的知识体系	5
4.1 规范敏捷 (Disciplined Agile)	5
4.2 持续交付 (Continuous Delivery)	5
4.3 IT 服务管理 (IT service management)	6
4.4 以 TPS (精益管理 Lean) 理念为基础	6
5. DevOps 团队角色	8
6. 组织	10
6.1 适合小型组织的扁平化组织 (Flat organization)	10
6.2 适合大型复杂组织的矩阵架构 (Matrix organization)	10
7. DevOps 的流程	11
7.1 业务战略和规划 (Business Strategy and Planning)	11
7.2 市场营销 (Marketing and sales)	11
7.3 管理 (Administration)	11
7.4 项目规划 (Project Planning)	12
7.5 需求和设计 (Requirements and Design)	12
7.6 开发 (Development)	12
7.7 部署 (Deployment)	13
7.8 运维 (Operation)	13
7.9 维保 (Maintenance)	13
7.10 客户服务 (Customer service)	13
7.11 生命周期终止 (End of life)	13
8. DevOps 的实施	15
8.1 丰田方式 (先进但复杂) (TOYOTA way)	15
8.2 协同方式 (标准) (Collaboration)	15
8.3 持续交付 (基本) (Continuous Delivery)	15
9. 结论	16
10. 推荐文献	17
11. 术语表	20
11.1 白皮书术语解析	20
11.2 EXIN DevOps Master 完整版术语列表	21

1. 整体介绍

1.1 前言

展现在大家面前的这本白皮书，将通过一个企业DevOps案例来呈现DevOps知识框架以及相关应用内容。

我们一直为丰田生产系统（Toyota Production System，TPS）中的核心综合管理系统（Toyota/Total Management System，TMS）提供敏捷开发的指导服务。基于我们的经验，我们相信DevOps能很好的支持业务。DevOps 不仅支持IT，而且还能支持业务战略并改善业务流程。

关于DevOps的书籍非常多。例如《凤凰项目：一个IT运维的传奇故事》是从IT经理的角度来编写的；还有《持续交付:发布可靠软件的系统方法》是从开发和项目经理的角度来编写的。《DevOps A Software Architect 's Perspective》是从架构师的角度来编写的。这些都是帮我们理解什么是DevOps的优秀书籍。



而本问从业务流程角度来编写的，因为DevOps旨在通过建立软件和IT服务的供应链来支持业务并管理整个流程的成熟度。

1.2 DevOps 的历程

DevOps的历程始于2009年，在成功地应用了Agile，Scrum和XP等方法论后，我们的客户能提供很炫酷的web交互服务，可覆盖从传统PC到iPhone移动设备的各种终端。

Scrum团队能够更快地开发和发布软件。即使开发时间减少了一半，但是业务主管仍然对业务速度的提升表示不满。

表面上看起来还是开发过程是瓶颈，但是在调查中发现开发过程并非瓶颈，相反是业务流程应该被改进。

TMS这个概念从业务战略和规划到客户服务的整个业务流程中得以实施。而使用 DevOps的概念将有助于建立一个流水线式的业务运营过程，并缩短交付前置期。

这个项目是在2012年成功完成的。整个过程通过与业务部门的协同，从端到端进行了重组，采用了可视化控制、单件流（One-piece flow）（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来的一种方法，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。）、每周进程同步、每天反馈循环以及KAIZEN（译者按：日语，持续改善，丰田所贯彻的方法）。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。

项目实施后，业绩得到了明显的提升：交付前置时间缩短、销售量提升、利润率和员工积极性也都得到了提升。这一切要归功于DevOps。



DevOps框架应该直接支持到业务产出，不仅仅是为了IT服务中开发与运维的协同，而是要能帮助企业使用IT服务来支持和提升他们的业务。

P2


DevOps的价值应由业务价值的产出来评判，而不是根据IT项目范围和IT成果来评判。

2. 什么是企业体系的 DevOps?

关于DevOps的书有很多，但不幸的是，大多数都是描述互联网行业的网站和产品开发中如何应用DevOps。很少有相关资料是考虑DevOps如何用于企业体系的。

企业往往同时拥有交互型系统 (SoE) 和记录型系统 (SoR)。SoE系统关注的是速度，SoR系统关注的是业务连续性。问题是当SoE频繁变更的时候，如何保障SoR的业务连续性呢？Gartner公司把这称为[双峰挑战](#) (Bimodal challenge)。

大多数企业的SoR的既有传统的系统与应用还需要维系和使用，使用DevOps建立准时制 (just-in-time, JIT) 概念的流水线过程可以帮到这类系统。

DevOps不能简单认为是一种工具、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流水线的过程，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计划 (戴明环) 来提升其成熟度。企业级的DevOps不仅仅是增强的敏捷开发和持续交付，同时也通过IT服务管理和应用程序管理来实现和促进业务增长并保障业务连续性。

P3

3. DevOps 的目标是什么？

DevOps的目标是建立流水线式的准时制（JIT）的业务流程。DevOps旨在通过合适的准时制业务流程来最大化业务产出，例如增加销售和利润率、提升业务速度，或尽量降低运营成本。

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。这种从提供软件交付到供给IT服务的模式转变是巨大的。

从架构的角度来看，DevOps需要建立一个自动快速部署系统。有很多方法论和工具可以利用。DevOps没有统一的实施模板，每个组织都不得不自己考虑并建立DevOps流程来提高业务。因此，真正理解DevOps的概念，对员工遵循正确的流程有效执行来说是至关重要的。

4. DevOps 的知识体系

当实施DevOps时，我们将从很多知识源、方法论、实践案例和工具中去选择参考。

DevOps主要由以下的三大支柱和一个基础组成。

4.1 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 for change)
- 总是能发布优质的无错误代码 (Always release high quality bug free code)

在IT服务生命周期中，越来越频繁和快速发布的开发速度应取决于业务变更的频度。工作质量是最重要的，需要将工作分割为小任务来进行支持。



Ji-Koutei-Kanketsu (JKK) 概念，认为100%的完成每个条目，是有助于保持高质量工作的。而“做完了” (Done) 与“结束了” (Completion) 的这些概念，对每个人来说都必须定义清楚。

使产品负责人可能改变他/她的任务的，未必一定是对待办项 (Product Backlog) 的管理，也可能是新的IT服务计划，在丰田，这工作是由首席工程师来完成的。

4.2 持续交付 (Continuous Delivery)

持续交付指的是实现自动应用程序的构建、部署、测试和发布的流程。

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。

每个组织都会有各自不同部署流管线 (Pipeline) ，因发布软件的价值流而异。

一个关键的成功因素是为IT服务建立一个单一的部署管线。

4.3 IT 服务管理 (IT service management)

当技术成为大多数业务流程的核心环节时,IT服务的连续性和高可用性是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,只有成功实现服务的连续性才能实现对高层的承诺,并支持组织的所有成员。对于保持有效性而言,持续维护其可恢复能力是最基本的前提条件。服务连续性是服务保障的必要组成部分。如果服务无法按照业务的要求保持连续性或恢复,那么业务将无法实现所承诺的价值。服务将无法被提供,从而失去持续的功效。

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最少必要信息 (Minimum Required Information , MRI) 的,严格聚焦于业务持续性的轻量ITSM。每个组织的MRI设置取决于他们的业务。



4.4 以 TPS (精益管理 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,因为有许多项目要改变现有熟悉的开发周期和方法论,你很有必要观念上做改变。

TPS的概念包括JIT和自动化,TPS可以帮助做到以下环节:

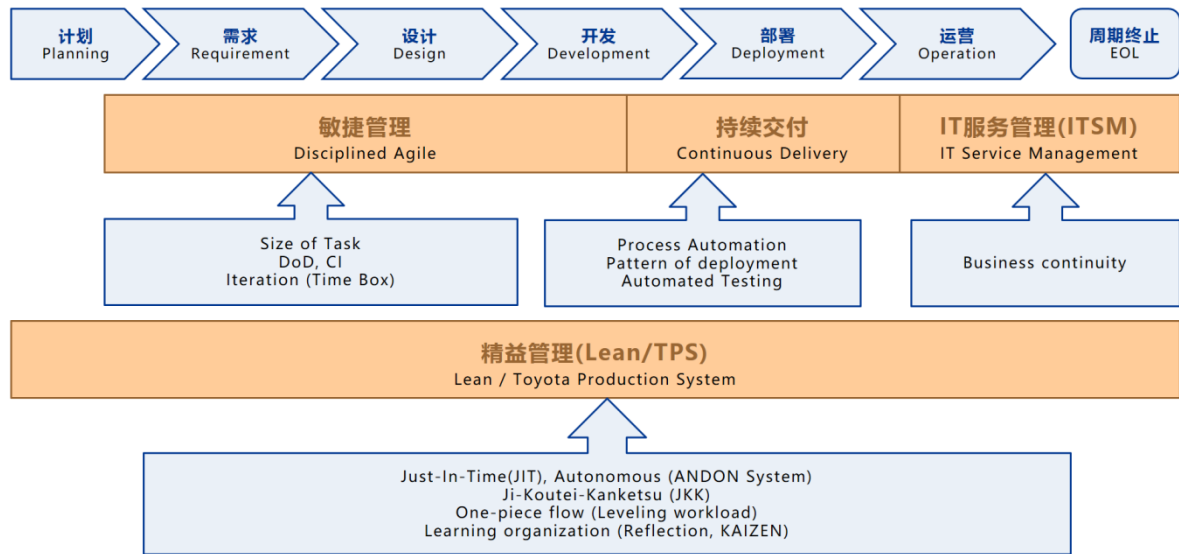
JIT意味着要建立一个流水线式的单件流 (one-piece flow) 的供应链。而自动化意味着尽可能实现自动化并且当生产过程出现缺陷时能停止整个过程。

这个过程需要设计并且员工也需要充分理解这两个概念。

另一个关键问题是开发和运维的生命周期。需要通过敏捷的方法改变工作方式,包括开发和运维之间每周或每天的信息同步。

下图展示了DevOps的知识体系：

流程 (Process)



5. DevOps 团队角色

为了保证IT服务的业务连续性，希望在您的组织中建立DevOps团队。

最好是组建一个小型优质的DevOps团队，根据亚马逊的“两个披萨规则”：团队人数不能多于两个披萨饼还吃不饱的地步。

团队角色描述如下：

流程主管 (Process Master)

领导并促进团队，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，力求建立单件流作业 (one-piece flow) 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，通过看板是否每个人都能很容易的理解当前的情形？”它并不显示状态。但它可以用来表达是否有问题出现。



经验需求：Scrum Master，敏捷项目领导 (Agile Project Leader)。

服务主管 (Service Master)

对提供IT服务及时性 (JIT) 负有全责。

这个角色就类似于Scrum中的产品负责人 (Product Owner)，对待办项 (Product Backlog) 做管理和排序，另外还负责IT服务的成本规划。

经验需求：Scrum 产品负责人 (Scrum Product Owner)、服务负责人 (Service Owner)。

DevOps工程师 (DevOps Engineer)

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和工具。DevOps 流程需要很多工具。

经验需求：研发 (Development)、工具 (Tools)。

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

负责监控 IT 服务的运行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、运营团队的成熟度以及他们的流程观念。

经验需求：IT 服务管理 (IT service management) 、运维 (Operations) 。

可靠性工程师 (Reliability Engineer) (可选)

监控部署过程中的服务，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。

监视和管理复杂的构建管线的工作流。

有义务提升测试流程。

经验需求：测试 (Testing) ，工具 (Tools) ，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训练有素的敏捷团队。

规范的敏捷团队致力于以可持续的步伐来满足发布计划和发布质量。

经验需求：开发 (Development) ，敏捷 (Agile)

运维团队 (Operation team)

采用轻量级的 ITSM 并在整体战略的环境中支持对服务的设计、实施、运维与改进。

在 TPS 中采用 “提前持续改善 (KAIZEN in Advance) ” 的实践

经验需求：运维 (Operations) ，持续改善 (KAIZEN) 。

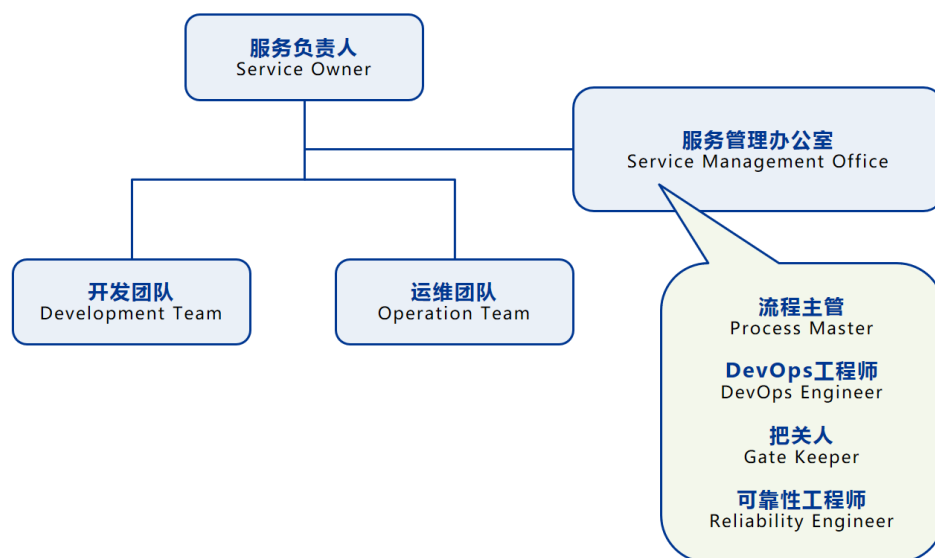
6. 组织

有必要在服务管理办公室 (Service Management Office) 中组建 DevOps 团队来支持服务主管。

有两种类型的组织架构展示如下：

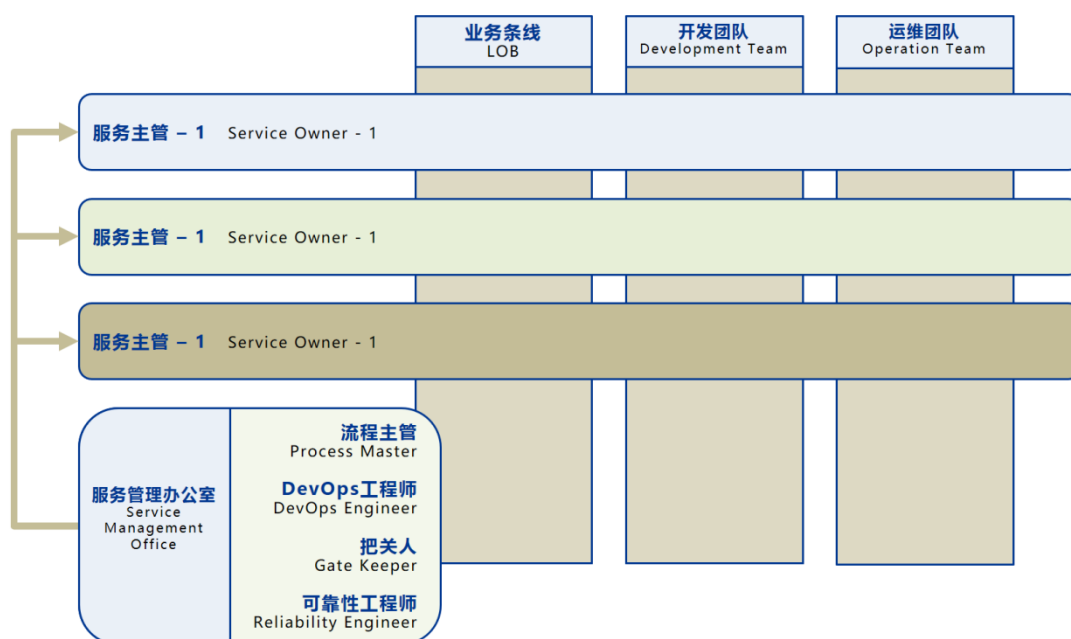
6.1 适合小型组织的扁平化组织 (Flat organization)

下图是小型组织的基本架构：



⇒ **P10**

6.2 适合大型复杂组织的矩阵架构 (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。

7. DevOps 的流程

要建立一个流水线式的流程，采用 JKK 在指导 DevOps 团队是最有效的办法。

JKK 是一种高质量的工作方式，它意味着对目标理解清晰，理解正确的工作方式，使工作正确的 100% 的完成，并在没有检查的情况下维护质量要求。

7.1 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。

7.2 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。

服务主管识别 IT 服务的客户，收集具有业务价值的需求，并约定时间范围。

7.3 管理 (Administration)

流程主管很希望了解如何可视化整个过程。一种方法是使用 Obeya (丰田的一种工程合作方式) 为整个流程进行设定。Obeya 这个作战室具有两个目的——信息管理和现场决策 (on-the-spot decision)。这里面有很多可视化管理工具。团队成员可以很快看到他们在过程中的方方面面。

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、保持队形、迅速收集信息、并形成重要的团队意识。

7.4 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、目标和项目的价值，然后整合 DevOps 的团队成员。

在这个阶段，运行中的基础设施被定义。一个整体流程的价值流图表被设计。

7.5 需求和设计 (Requirements and Design)

服务主管定义待办任务 (Product backlogs) 和并安排优先级。

DevOps团队使用待办任务 (Product backlogs) 来定义故事 (Story) 。

- 用户故事：角色，职能，业务价值/理由，以及运营条件。
- 测试故事：验收测试用例和服务验收标准。
- 运营故事：设置 IT 服务的优先级的和业务连续性的运营条件。

创建服务级别和运营级别协议。

DevOps 工程师和运营团队定义转换、测试和开发的基础设施。

开发团队还创建了发布和迭代计划。

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。

7.6 开发 (Development)

Scrum 是这个阶段最适用的方法论。

开发团队必须提交发布计划并使用规范的敏捷方法。

每次迭代 (Sprint) 的周期需要遵循业务的需要。

从质量的角度来看，XP (Extreme Programming ，极限编程) 的实践，例如结对编程 (pair-programming) 、 TDD (Test-Driven Development ，测试驱动开发) 、 重构 (Refactoring) 和十分钟构建 (Ten-Minutes Build) 都是有效的。

7.7 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作业(One-piece flow)方式构建一个单一的自动化部署途径(pipeline)。

可靠性工程师和 DevOps 工程师将共同提升测试流程。

把关人 (Gatekeeper) 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。

7.8 运维 (Operation)

运维团队采用轻量级的 ITSM 流程来监控 IT 服务运行的状态。

发生灾难事件时，确保关键服务依然运营是至关重要的。这个团队应该包括可靠性工程师，需要注意两个关键参数：恢复点目标和恢复时间目标 (RPO/PTO)。

7.9 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，它们被作为变更请求添加到待办任务中。

7.10 客户服务 (Customer service)

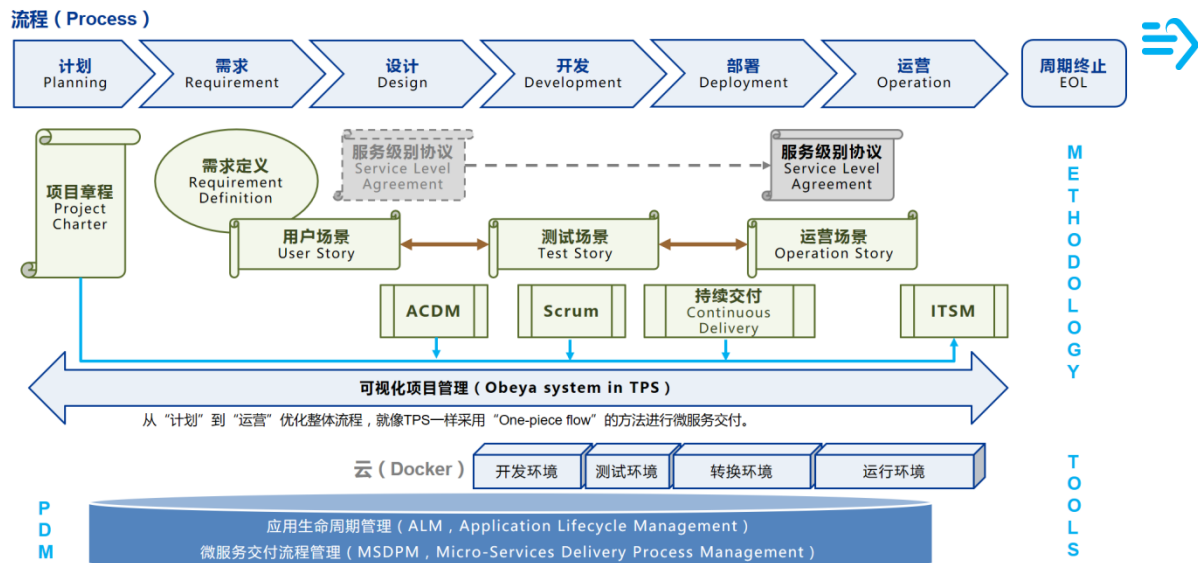
服务主管和可靠性工程师负责收集客户的反馈，例如包括用户体验和质量事件的运营问题。经允许，它们被作为变更请求添加到待办任务中。

7.11 生命周期终止 (End of life)

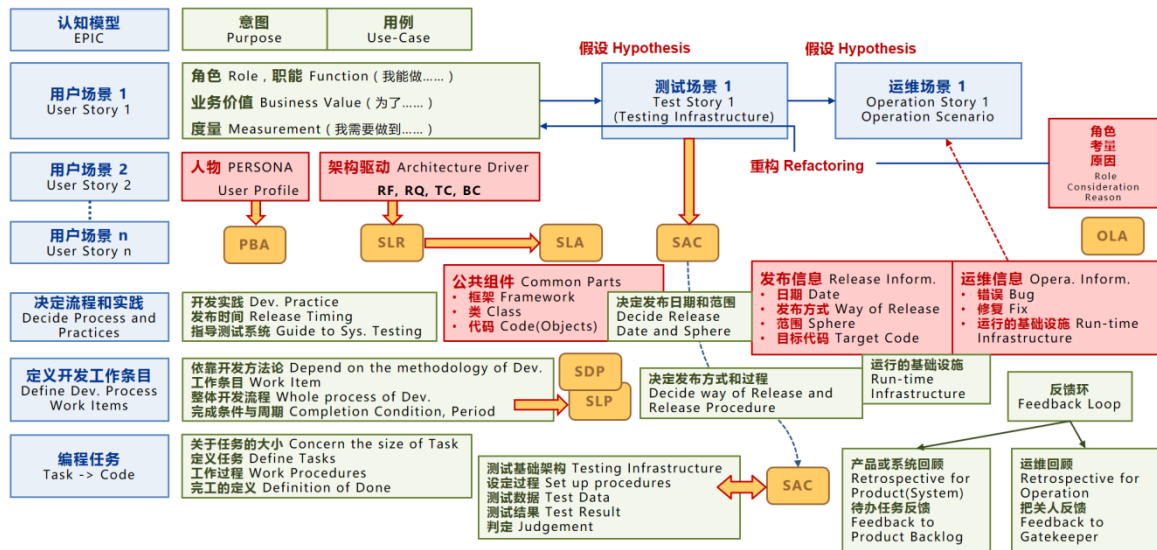
服务主管将决定 IT 服务生命周期的终止条件，包括发生事件，以及如何发生。

(接下页)

下图显示了 DevOps 的一个配置示例。



轻量级的 ITSM 图表如下图所示：



8. 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。

8.1 丰田方式（先进但复杂）(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）（译者按：一般 IT 服务提供商分为内部和外部，内部为企业的 IT 团队，外部可能是一家独立的 IT 公司）。

8.2 协同方式（标准）(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，一般由服务主管来主导。



这种方式尤其适合交互型系统（SoE）和记录型系统（SoR）共存。

8.3 持续交付（基本）(Continuous Delivery)

这种方式侧重于快速和频繁的软件发布，可以由产品负责人主导。

它最适合数码产品提供商（译者按：许多新型的互联网企业都属于此列）。

9. 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，将有助于您理解 DevOps 并从中受益。

10. 推荐文献

- **The Phoenix Project** - A Novel about IT, DevOps, and helping your business

作者 : Gene Kim, Kevin Behr, George Spafford.

书号 : ISBN9784822285357

推荐人群 : 通用阅读

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

书号 : ISBN 9787-7-115-40365-0

- **Continuous Delivery** - Reliable Software Releases through Build, Test, and Deployment automation



P17

作者 : Jez Humble, David Farley

书号 : ISBN9780321601919

推荐人群 : DevOps工程师, 可靠性工程师, 开发团队, 流程主管

注 : 该文献目前已有中文版图书《**持续交付:发布可靠软件的系统方法**》

译者 : 乔梁

书号 : ISBN 9787115264596

- **Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale**

作者 : Jennifer Davis, Katherine Daniels

书号 : ISBN9781491926307

EXIN® is a Registered Trademark of EXIN Holding B.V.

推荐人群：流程主管，开发团队

- **DevOps: A Software Architect' s perspective** (SEI Series in Software Engineering)

作者：Len Bass, Ingo Weber, Liming Zhu.

书号：ISBN9780134049847

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

- **The DevOps 2.0 Toolkit** - Automating the Continuous Deployment Pipeline with Containerized Microservices.

作者：Viktor Fracic

P18

书号：ISBN9781523917440

推荐人群：DevOps工程师、开发团队，流程主管



- **Architecting Software Intensive Systems** (ACDM) - A Practitioner' s Guide

作者：Anthony J. Lattanze.

书号：ISBN9781420045697

推荐人群：开发团队，流程主管

- **DevOps Automation Cookbook**

作者：Michael Duffy.

书号：ISBN9781784392826

EXIN® is a Registered Trademark of EXIN Holding B.V.

推荐人群：DevOps工程师，运维团队，流程主管

- **The Visible Ops Handbook** - Implementing ITIL in 4 Practical and Auditable steps

作者：Kevin Behr, Gene Kim, George Spafford.

书号：ISBN9780975568613

推荐人群：运维团队、把关人、流程主管

- **TPI NEXT: Business Driven Test Process Improvement**

作者：Alexander van Ewijk, Bert Linker, Marcel van Oosterwijk, Ben Visser, Gerrit de Vries, Loek Wilhelmus, Tik Marselis.

P19

书号：ISBN9789072194978

推荐人群：DevOps工程师，可靠性工程师，开发团队，流程主管



- **The TOYOTA Way**

作者：Jeffrey K. Liker

书号：ISBN9780071392310

推荐人群：通用阅读，执行层，经理，服务主管，流程主管

11.术语表

11.1 白皮书术语解析

此术语表描述了我们在白皮书行文中使用到的一些术语。完整术语表参见EXIN官方文档

KAIZEN (持续改善)

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”。

问题需要通过数据的方式来定义和支持。是否大家都能清晰地认识到问题？设置一个你发现的问题的假设，然后基于防范方式来思考并验证你的假设。

防范措施必须基于活动在日常进行定义，也需要设定每周的KPI，这样人们可以感觉到一种成就感。

KAIZEN in Advance (提前改善)

P20

当下游的环节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假设。然后他们可以向上游部门提出期望。这是一个反馈闭环的问题。



Ji-Koutei-Kanketsu (JKK)

JKK的概念是一种完美状态：在你所处在工作流程中不要做低质量的工作，不接受流程早期就出现错误的输出，不把糟糕的情形输出到下一个流程。

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度量方法去决定是否继续进行下一步。

TPI NEXT:

TPI-Test Process Improvement 测试流程优化是一种业务驱动测试过程改进。TPI是一套来自欧洲的软件测试最佳实践方法论，EXIN 提供 TMAP 和 TPI 相关的认证。

TPI NEXT® is a Registered Trade Mark of Sogeti Nederland B.V.

ITIL® is a Registered Trademark of AXELOS Limited

EXIN® is a Registered Trademark of EXIN Holding B.V.

11.2 EXIN DevOps Master 完整版术语列表

ENGLISH	CHINESE
Affinity (<i>in DevOps</i>)	亲和
Agile	敏捷
Application Deployment	应用部署
Artifact Management	构件库管理
Automation	自动化
Blamelessness	免责
Cloud Computing	云计算
Collaboration (<i>in DevOps</i>)	协作
Configuration Management	配置管理
Containers	容器
Continuous Delivery	持续交付
Continuous Deployment	持续部署
Continuous Integration	持续集成
Definition of Done (<i>in Agile Scrum</i>)	完成的定义
Infrastructure Automation	基础架构自动化
Iteration	迭代
ITSM (IT Service Management)	IT 服务管理
Ji-Kotei-Kanketsu (JKK)	质量检查 (JKK)
Just-in-Time (JiT)	准时
Kaizen (<i>in Lean</i>)	持续改善
Lean	精益
Micro-service	微服务
Minimum Viable Product	最小可用产品

ENGLISH	CHINESE
One-piece-flow	单件流
Operations Story	运维场景
Organizational Learning	组织级进修
Plan-Do-Check-Act cycle (PDCA cycle)	计划-实施-检查-改进 （戴明环）
Post-Mortem	事后剖析
Product Owner (<i>in Agile Scrum</i>)	产品负责人
Retrospective	回顾
Rhythm (<i>in Lean</i>)	节奏
Scaling (<i>of DevOps or Agile Scrum</i>)	规模化
Scrum	Scrum
Scrum Master (<i>in Agile Scrum</i>)	Scrum Master
Service Deployment	服务部署
Service Level Agreement (SLA)	服务级别协议(SLA)
Sprint	迭代 (sprint)
Test Story	测试场景
Test-Driven Development	测试驱动开发
Tools	工具
Toyota Production System (TPS)	丰田生产系统
User Story	用户故事
Velocity (<i>in Agile Scrum</i>)	速率
Version Control	版本控制
Waterfall	瀑布式
WiP-limit	在制品数量限制
Work-in-Progress (WiP)	在制品