

服务工程方法论

服务工程的技术范畴很广,覆盖了服务系统全生命周期的各个环节。为了设计和实现高质量的服务系统,服务工程中的各环节需要充分的技术支持,将会用到需求工程、模型论、系统论、软件工程等不同领域的知识,这对服务设计与开发人员提出了较高的能力要求。为了帮助服务设计与开发人员对服务系统进行快速、准确、高质量的定义、描述、设计、优化和实现,有必要为他们提供一套可参考的方法论。

服务工程方法论是服务工程的重要组成部分,被看做是该领域迫切需要解决的核心问题之一^[1]。服务工程方法论通过定义一整套方法、规则与步骤,为服务工程领域的分析人员、设计人员、开发人员、服务从业人员提供服务系统构建阶段的工程方法指导。

本章重点介绍服务工程方法论的有关内容;给出服务工程方法论的概念,详细介绍服务工程方法论的构成要素,从服务模型规范、服务建模方法、服务系统构造方法、服务质量保证、支持工具、实施指南共 6 个方面展开论述,最后简要介绍几种目前得到广泛应用的典型服务工程方法。

3.1 服务工程方法论概述

3.1.1 方法论

“方法论”(Methodology)是一个在各研究领域得到广泛应用的术语。它可被定义为“某一学科领域所采用的一组特定方法的集合”;每种方法解决该领域内的特定问题;每种方法由一系列过程、活动与任务构成。除了“方法”之外,它还包含了方法中所必须遵循的基本原理、标准、规则、所做出的假设,以及支持方法实施的支撑工具等。

“方法论”是任何工程性学科均应具备的重要组成部分,它为该领域的研究与业务人员提供了一整套求解问题的通用途径。在面临实际问题时,可直接采用方法论中的特定方法与原则。从这个意义上看,方法论实际上是特定领域中问题求解的“可复用模板”,也是一种“最佳实践”(best practice)的总结。

一套完整的方法论包括三个部分:体系结构、建模、实施指南。“体系结构”提供了对特定系统的结构、关系、环境等方面进行刻画的方法;“建模”提供了对系统进行设计、分析和优化的方法;“实施指南”则提供了如何将系统实施于用户应用环境的步骤指南。

以软件开发领域为例,目前一种典型的软件工程方法论 RUP (Rational Unified Process,统一开发过程)由以下部分构成,这些方法、原则、哲理、工具共同构成了软件开发人员可直接遵循的原则与方法集合^[2]:

- (1) 多视图的软件描述;
- (2) 可视化的软件建模语言;
- (3) 迭代式软件开发过程;
- (4) 结构化与面向对象的编程泛型与高级语言;
- (5) 基于构件与框架的体系结构;
- (6) 持续的质量验证;
- (7) 严格控制的软件变更;
- (8) 计算机辅助软件工程与集成开发环境 (IDE)。

在项目管理领域,不同项目阶段应用不同的方法,也形成了较为完善的方法论。美国项目管理协会的 PMBOK^[3]将项目分为 5 大过程组(启动、计划、实施、监控、收尾)、9 个知识领域(范围管理、计划管理、成本管理、人力资源管理、质量管理、沟通管理、风险管理、采购管理、综合管理)和 44 个过程。每个过程都有“输入”、“输出”以及“技术工具”。所有的“技术工具”汇总起来可以构成针对成果性目标的项目管理方法,举例如下。

(1) 质量管理方面:全面质量管理(TQC)、PDCA 循环、因素图法、因果分析法、控制图法、直方图法等。

(2) 项目技术经济指标和评价(财务评价、国民经济评价、净现值、内部收益率、投资回收期等)。

(3) 时间管理方面:网络计划方法、关键路径法等。

(4) 费用管理方面:费用估算、净值分析法等。

(5) 风险管理方面:SWOT(优势、劣势、机会、威胁)分析、概率分析(决策树、蒙特卡洛模拟、敏感性分析、盈亏平衡分析)等。

计算机集成制造(CIM)领域是另一个包含丰富方法论的工程领域。CIM 方法论是一组方法的有机结合,用来指导和帮助 CIMS 设计人员进行 CIMS 分析、设计和实施^[4]。它包括五部分。

(1) 根据问题需要构成的参考模型体系结构:一组代表整个集成企业或 CIM 系统的各方面的多层次、多视图的模型集合,例如 CIMOSA、GIM/GRAI、Purdue 参考体系结构等。

(2) 一组建模方法:用来建立参考体系结构中的各种模型,这些模型往往由多个视图构成。例如 IDEF0 方法用来建立企业系统模型的功能视图和物理视图; IDEFIX 用来建立信息视图; GIM/GRAI 方法用 GRAI Grid 和 GRAI Net 来建立决策视图; E-R 模型用于建立信息模型或数据模型; CIMOSA 包含了功能视图、组织视图、资源视图和信息视图等。

(3) 保证逐步实现目标的结构化进程:一般按四个阶段对 CIMS 的发展过程进行设计,分别为初始化、分析、设计以及实施。

(4) 系统开发与实现指南:一组用来指导 CIMS 系统设计、开发与实现的过程步骤、实施原则和技术规范、组织方式等。

(5) 支持工具:包括 CIMS 建模支持与管理工具、CIMS 系统开发过程的支持工具和集成化开发支撑环境等。

3.1.2 服务工程方法论概念

在服务工程中,同样需要一套覆盖服务全生命周期的方法论以支持服务企业建立其服务系统,该方法论称为“服务工程方法论”(Service Engineering Methodology)。它一般由一整套方法、规则与步骤构成,用于对服务系统进行架构与功能规划、描述/建模、构建,并对服务系统性能进行优化改进,以满足快速变化的客户需求。

以下是对服务工程方法论的几个认识^[1,5~8]。

(1) 服务工程方法论侧重于关注服务系统的构建阶段而非运行阶段,以“将满足顾客需求的服务系统部署到运行环境下”为目标。当外界需求和环境发生变化时,应提供相应的方法使服务系统通过演化或重构来适应变化。

(2) 服务工程方法论由覆盖不同阶段的方法与指导性原则构成,包括服务创新、需求工程、服务设计、服务开发、服务部署等典型阶段。

(3) 服务工程方法论应遵循统一的工程化原则,例如以服务体系结构为中心、全过程文档化跟踪与可追溯性、标准化的建模符号、基于并行工程、系统化的复用等。

(4) 服务工程方法论可以采用模型驱动的方法,通过需求模型定义(CIM模型)、系统模型设计与分析(PIM模型)、实现模型生成与改进(PSM模型)等步骤完成服务系统的设计过程。

(5) 服务工程方法论的目标是提高开发效率,降低开发成本,提高目标系统的质量,降低开发过程中的管理与协调难度。

(6) 评价服务工程方法论的准则是:是否易学易用、是否标准化、是否考虑了服务设计的各个方面、是否能够适应服务开发中的各种变化等。

根据服务系统的生命周期,服务工程方法论可以抽象出服务模型规范、服务建模方法、服务系统构建方法、服务性能评价方法、服务系统实施指南、支撑平台与工具这6个组成部分^[9],分述如下。

(1) 服务模型规范(service model specification): 形式化地描述和表示服务系统及各类服务要素(包括资源、能力、人员、行为、过程等)。

(2) 服务建模方法(service modeling method): 服务系统的需求分析、设计和实施模型的转换与映射。

(3) 服务系统构建方法(service system development method): 建立服务系统的指南与过程,包括具体的步骤以及方法指南。

(4) 服务性能评价方法(service performance evaluation): 对即将或者已经建立的模型、服务系统的评价,服务系统的性能评价(顾客满意度、质量、成本、时间等)。

(5) 支撑工具与平台(supporting tools and platform): 支持服务系统工程的IT工具及平台。

(6) 服务系统实施指南(service system implementation guidelines): 将已开发的服务系统部署到运行环境下的方法与手段。

如图3-1所示,服务工程方法论的6个要素涵盖了服务的全生命周期。通过调查分析顾客需求,采用按需应变(on demand)的思想进行服务设计,建立一套顾客、企业理解相一致的服务模型;然后利用服务系统开发方法将服务模型转化为可运行的服务系统,完成服

务的开发与部署；对于要建立或者已经建立的服务模型/服务系统,提供相应的评价方法来加以评价以支持优化；服务的建模、系统实现及评价由相应的开发平台和软件工具来支持实现。

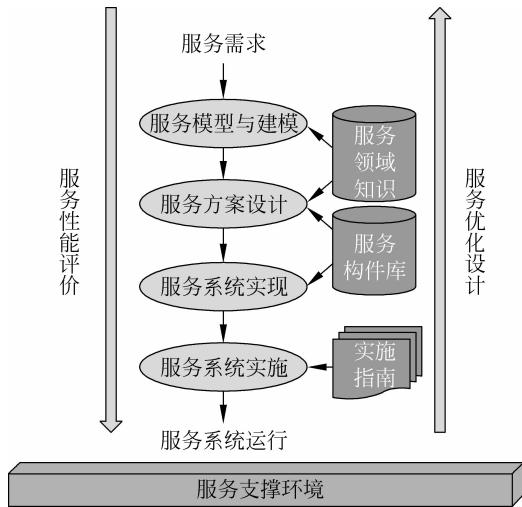


图 3-1 服务工程方法论的组成部分

3.2 服务工程方法论的构成要素

本节针对服务工程方法论的 6 个构成要素分别进行论述。

3.2.1 服务模型规范

模型(model)是对系统中的某些理论或现象的图解性或形式化描述,并通过一组变量以及作用在这组变量之上的逻辑或数量关系来描绘其已知的或推测的特性。建立模型的目的在于提供一个理想的论证框架,应用逻辑和数学工具,评估其某一侧面的性能,并在多个类似的场景下进行推理。

服务工程的一个重要原则是“模型驱动”,即通过模型来表达服务设计者的创新思想、业务需求、目标服务系统的形态(体系结构、构成要素),刻画系统客户/使用者与服务要素之间的关联关系,进而建立这些模型之间的(半)自动映射与转换,以达到从需求获取/定义到系统最终实现的转化^[10]。因此,服务模型是一套完善的服务工程方法论的首要组成部分。

建立服务模型的另一个目的是:通过一套标准的建模符号,对各类服务要素(过程、资源、角色等)进行描述,从而在参与服务设计、开发、实施过程中的各参与者之间建立一个交流与协作的平台^[11]。

最初的服务模型规范来源于服务管理领域。随着“面向服务”的思想在各领域的普及,诸如系统工程、软件工程、面向服务的计算等领域的模型开始被广泛引进来。目前可用于服务建模的模型规范非常多。但是,不同的模型规范有不同的目标与定位,其所采用的建模标准与符号体系也很不同。以下简要给出不同角度的服务模型分类^[7,10]。

(1) 按照服务模型所描述服务的不同侧面,可分为过程模型、组织模型、资源模型、信息模型、知识与规则模型、价值模型等服务视图。不同的服务视图分别从某一视角关注服务,

模型形态相对简单和清晰。通过视图之间的语义关联将它们形成整体。

(2) 按照服务模型所起到的作用,可分为用来指导服务创新与设计的模型、用来描述服务需求的模型、用来指导服务系统设计与开发的模型、用来指导服务部署与实施的模型等。也可分为战略层模型、战术层模型、操作性模型、技术层模型等。

(3) 按照服务模型的形态,可分为图形化模型与形式化模型。前者被用于帮助服务设计人员之间的沟通和理解;后者被用于软件工具支持下的服务分析与验证,例如一致性、完备性、正确性、性能等方面。

(4) 按照服务模型所采用的建模标准,可分为面向企业与商务的建模规范、专门面向服务的建模语言以及面向软件系统的建模语言。

国内外研究者从各个角度提出了适合上述不同分类的服务模型规范^[12~14],以下是几个典型代表。

(1) 分子模型(molecular model):参考化学中的分子结构提出了产品/服务设计分子结构模型。通过产品/服务的组件(原子)和产品/服务组件之间的关系(化学键)组成产品/服务(分子)。该模型体现了对产品/服务的分层结构化设计,但是缺乏对服务整体过程的直观表现。

(2) 服务蓝图(service blueprint):以流程图为基本形式,在服务流程分析基础上描述服务传递的各方面,鉴别用户与服务人员以及服务体系内部的服务接触点,将服务提供过程、员工、顾客的角色和服务的有形证据直观地展示出来,从而把服务合理分解成服务提供的步骤、任务和方法,使服务提供过程中所涉及的人都能客观地理解和处理它。

(3) 结构化服务模型(SADT model):借鉴 IDEF0 模型的基本形态,自顶向下逐步对服务功能进行分解,构造分层的 service 功能模型。每层模型均以服务活动为基本单位,每个活动使用输入、输出、控制、执行者 4 个方面加以描述。

(4) 统一建模语言(UML model):采用面向对象的思路,从用例视图、逻辑视图、物理视图、进程视图、开发视图 5 个侧面对服务系统进行描述。

(5) Web 服务模型(WSDL + BPEL):这是服务计算领域中用以描述软件化服务及其形成的业务流程的一种模型形态,采用 XML 作为基本形态,在基础软件的支持下可以直接执行。

(6) 业务过程建模标注(BPMN):通过一套可直观理解的模型符号描述服务业务流程,并可映射为可执行的 BPEL。

(7) 面向服务的本体 Web 语言(OWL-S):这是一个以 OWL 语言描述的服务本体,从 process、grounding、profile 三个视角分别描述 Web 服务,其目标在于支持 Web 服务的自动发现、调用、组合、管理等。

(8) Web 服务建模本体(WSMO):与 OWL-S 有着相似的目的,它提供了一个概念框架和一种形式化描述 Web 服务的语言,用户通过目标(goal)描述需求,通过基于语义的推理来确定满足需求的服务。

上述主要的服务模型描述语言将在本书第 4 章中详细介绍。

不管何种模型描述方式,每个服务模型都由一组服务建模要素及其之间的逻辑关系构成。而且在现实中,单一视图、单一作用、单一形态的服务模型是不够的,通常需要将上述不同分类的小粒度模型组织在一起,形成二维乃至多维的模型体系,以全面描述服务系统^[7]。从服务生命周期的角度,不同阶段的服务模型所侧重的内容也有所差别,因此形成了多层次的服务模型体系,相邻层次之间通过模型驱动或转换的方式建立联系。

在服务建模过程中,建立良好的服务模型的主要挑战之一是变量太多。由于服务的对象是“人”,客户服务的要求、服务团队的人员等都会变;不同的人员在服务生命周期里担任不同的角色,所需的交互非常频繁。因此,针对不同服务领域选择一组恰当的变量并在模型中充分刻画出来,是一件非常困难的事情。

服务建模的另一个挑战是对服务中频繁的、复杂的交互过程(尤其是各参与方的行为)进行精确的描述。服务本身可看作是供需双方之间的协同生产过程,这是服务模型区别于传统的企业模型或业务模型的显著特征。除了交互过程外,服务价值、服务质量、服务性能等非功能特征也是服务模型区别于其他模型的重要特征。

另外,服务中蕴含的丰富语义特征(如行为的约束条件、时序、依赖关系等)也需要在模型中加以严格的刻画。通过图形化方法可以直观描述服务业务,但无法做到对非可视化语义的描述,为此需要引入语义扩展机制,通过对象约束逻辑(OCL)、动态描述逻辑(DDL)、过程逻辑(PL)、本体描述语言(OWL-S)等语义工具对传统的服务模型进行扩展,增强服务模型表达能力。

3.2.2 服务建模方法

服务建模方法是指使用恰当的服务建模语言,按照特定的方法和步骤,构造出目标服务和系统模型的过程。服务建模是一个将现实业务和需求转变成服务模型并最终指导服务系统实施生成的过程。通常不同的服务模型规范具有自身独特的建模方法。

服务建模方法通常包含若干个主要阶段,即服务需求定义、服务行为设计、服务执行机制建模、服务技术实现建模等。每个阶段又可划分为若干个更细小的步骤,分别建立服务模型规范中所包含的功能、价值、质量等内容。

以服务蓝图建模方法为例,共分为9个步骤,其中前6个步骤属于建立图形化模型的过程,后3个步骤则是在此基础上所进行的分析活动^[13],如图3-2所示。

Step1: 识别需要制定蓝图的服务过程。

Step2: 识别顾客(细分顾客)对服务的经历。

Step3: 从顾客角度描绘服务过程。

Step4: 描绘前台与后台服务雇员的行为。

Step5: 把顾客行为、服务人员行为与支持功能相连。

Step6: 在每个顾客行为步骤加上有形展示。

Step7: 在模型上标记可能出现服务失败的地方并设计服务补救措施。

Step8: 为每个过程设立时间标准。

Step9: 分析收益率。



图 3-2 服务蓝图建模过程

虽然大多数服务模型为可视化的图形形式,但服务建模并不等同于“绘图”,绘制模型只是其中的一个子阶段而已。除此之外,服务建模方法还包括模型细化、模型转换、模型语义增强、模型验证等内容。

(1) 服务模型细化:按照自顶向下的原则,对服务模型中的某个局部进行分解,使用一组粒度更小的模型要素及其之间的关系描述细节,得到下层的模型。上下层模型之间要保持等价性原则。采用 SADT 方法建立服务模型时,逐层向下的模型细化与分解是核心步骤。

(2) 服务模型语义增强:服务中蕴含的诸如约束、规则、知识等业务语义信息也需要在服务模型中加以刻画,但它们很难以图形化的方式加以描述,需通过引入模型属性值对、OCL、DL 等形式化或半形式化语言对服务模型进行语义信息的增强,提高服务模型的表达能力;还可进一步支持基于模型的服务分析、仿真、测试等活动。

(3) 服务模型转换/映射与互操作:由于不同的组织、建模人员、多层建模框架中的不同层次可能采用不同的服务建模语言,使得多个模型间出现差异,为了使这些异构模型之间能够建立一致化的语法和语义上的映射关系以实现相互理解,需要通过诸如本体的支持,为双方或上下层的模型进行本体标注,进而达到模型转化与互操作的目的。

(4) 服务模型分析与验证:通过内嵌于服务模型的一组定量参数或逻辑参数,对服务模型进行多视图之间的一致性分析、正确性分析;通过仿真,可对服务的时间特性、成本特性、可靠性与可用性、收入与盈利特性、顾客满意度特性等内容进行量化分析,从而可在服务被投入实际运营之前发现存在的问题并消除之。

(5) 服务价值的声明、描述、转换、验证:服务价值是服务系统产出的最终目标,因此需要在建模阶段充分描述价值,并对服务模型所能实现价值的能力进行评价,以确保基于服务模型所开发出来的服务系统具有较高的服务价值提供能力。

(6) 服务质量的声明、描述、转换、验证:服务质量是确保服务价值实现的重要方面,而质量的高低在服务建模阶段就可以初步确定。为此需要在服务功能建模的基础上强化局部质量(服务要素的质量)和全局质量(服务的整体质量)的设计和验证。

3.2.3 服务系统构建方法

在建立服务模型之后,下一步需要根据服务模型构建相应的服务系统,以支持服务业务运行与价值创造。

在目前的 Web 服务与 SOA 环境下,大部分现实中的服务过程与服务行为均可被转化为 Web 服务的形式向外发布,服务的需求者可查询并绑定这些服务,双方即可进行具体的服务交互过程。SOA 自身具备的良好特征使得参与服务的各类资源能够有效和灵活地组织在一起。通过服务的互操作,将跨组织的、分布在不同技术平台和协议上的人、过程、信息等要素无缝地连接在一起,形成松散耦合的服务系统与服务网络。

在实践中,借助于 Web 服务、服务构件体系结构(SCA)和企业服务总线(ESB)等 SOA 技术和“面向服务的建模与体系结构”(SOMA)等方法的支持,通过服务构件的选取与组合可以得到最终的协同服务系统。

典型的服务系统构建方法有以下三类。

(1) 自顶向下(top-down)的构建方法:根据服务模型所描述的业务,从中分析与识别

出一组松散耦合的服务,设计服务接口与内部机制;然后,通过某种方式实现之;进而,按模型中刻画的逻辑关系将这些服务组合到一起,形成完整的服务系统。

(2) 自底向上(bottom-up)的构建方法:该方法与前一种方法的不同之处在于,构建服务系统的基本服务单元不是完全从无到有被设计和开发出来的,而是多数已经存在了。因此在建立了服务模型之后,通过功能匹配、QoS 匹配、语义匹配等方式,在已有服务构件库中选择恰当的服务构件并按模型描述将它们组织在一起,即可得到服务系统。

(3) 基于领域工程的服务构建方法:该类方法是对上述两种方法的复合,它将系统构建过程分为两阶段(领域工程和应用工程)。领域工程阶段采用自顶向下的方法构建一个面向领域中群体顾客需求的服务系统;而应用工程阶段则采用自底向上的方式,通过对领域服务系统的配置、修改、重新组织,以较小的代价满足特定顾客的个性化需求,得到最终服务系统。

3.2.4 服务性能评价方法

服务质量是保证顾客满意度的关键因素,因此在设计与开发服务系统中,应充分关注服务质量如何被保证、判断服务质量是否维持较高水平、是否出现了服务质量等问题,然后再对服务系统进行构建与调整。

如图 3-3 所示,在根据顾客功能需求建立服务模型的过程中,还需要充分获取顾客对服务质量的期望,将质量作为服务设计的约束条件。在依据服务模型开发相应的服务系统并投入运行期间,还要根据预先建立的服务质量度量体系来监控服务系统的执行,分析产生质量问题的原因,进而采取优化措施对服务模型和服务系统进行改进。

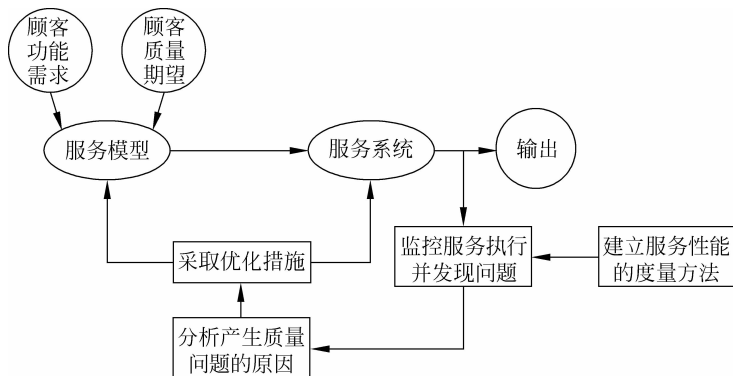


图 3-3 服务质量保证的示意图

在服务建模过程中,将质量作为服务设计的约束,在服务模型中表现质量属性,并且这些质量约束要反映在服务模型的建立、分解、转换、映射中,这是服务设计方法的组成部分;对服务质量的改进和优化设计,则是服务优化方法的组成部分。二者结合起来,就构成了服务质量保证方法的主体内容。

3.2.5 支撑平台与工具

服务工程方法论对服务系统进行设计、构建、部署一般都需要相应的软件工具来支持实现。这些软件工具应包括:服务模型的代表、建模工具、模型分析工具(仿真、价值/性能/质

量等指标的分析)、模型库/知识库/方法库、服务系统构建工具、服务构件库、服务系统部署、实现支持工具以及服务评价工具等。这些工具构成服务系统开发平台来支持服务系统从构建到实际运行的映射。

3.2.6 服务系统实施指南

在服务系统被完全开发出来之后,还需要将其从设计与开发环境迁移到客户指定或所需的实际运行环境下;通过对相关软件、硬件、数据、网络、人员等相关要素的配置,使服务系统达到可运行状态。服务工程方法论中包含了一系列指导服务项目组织与实施的指南。

服务系统的实施过程需要考虑较多因素(图 3-4),实施的内容主要包括:

- (1) 建立硬、软件环境,实现物理系统;
- (2) 装载数据,系统试运行,并做局部调整;
- (3) 用户技术培训和操作培训;
- (4) 进行系统切换和交接;
- (5) 制订系统管理和操作制度,正确运行系统;
- (6) 维护系统,实现设计目标,发挥最大效益。

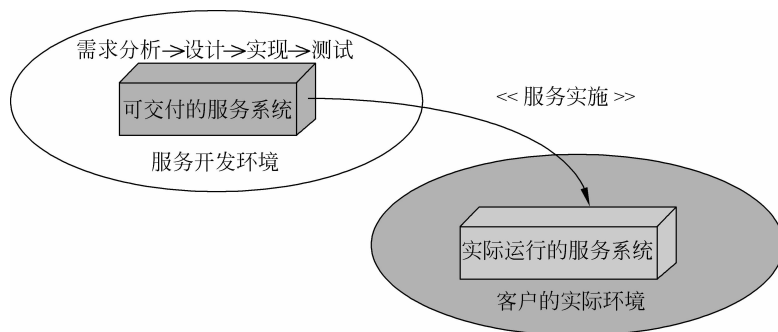


图 3-4 服务系统实施的示意图

因为服务系统本身的复杂性以及服务实施过程的复杂性,服务工程方法论应提供一套可参考的实施指南。实施中应包括实施过程中各阶段工作的结构化进程、项目管理、服务系统实施队伍组织、质量保证工作等。

为了保证服务系统的实施过程能有条不紊地进行、工作内容方面不出现纰漏、各阶段的结果尽可能地得到优化,整个系统实施进程的内容、组织、步骤和应有的文档等都应实现结构化和标准化。

一般地,服务系统实施可分解为服务业务分析与转型阶段(服务目标定义、服务环境分析、诊断、会议组织、方案制定)、服务系统规划阶段(项目范围、项目组织、项目计划、实施形式、项目启动)、服务系统研发阶段(流程优化、系统配置与安装、系统测试与试运行等)、服务系统交付与具体实施阶段、后续改进阶段等。借鉴传统 ERP 系统的实施过程,服务系统实施的典型阶段划分如图 3-5 所示。

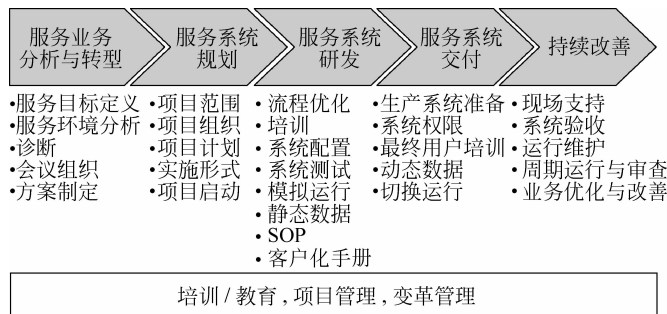


图 3-5 服务系统实施的典型阶段划分

3.3 典型的服务工程方法论

随着人们对服务领域研究的不断深入,出现了越来越多的服务模型规范、服务建模方法、服务系统构建方法、服务实施指南、服务质量评价方法,以及相应的支持工具。但是,每种方法只是解决了单一侧面的问题,无法完全满足企业应对顾客日趋个性化、复杂化和多变的服务需求的服务系统开发要求。将服务建模、开发和评价等方法集成在一起,进而研究一整套支持服务全生命周期开发的服务工程方法论已经成为必然的发展趋势。

经过多年积累,学术界与工业界已经形成了较多可借鉴的服务工程方法论。按照它们所面向的服务系统类型和所采用的技术路线的不同,可以将它们分为 4 个大类。

3.3.1 基于传统软件工程方法的服务方法论

该类方法论主要面向以软件为主要形态的服务系统,属于服务计算领域的方法论。它借助于软件工程领域中“面向对象的分析与设计方法”(OOAD)和“基于构件的软件开发方法”(CBSD)的相关技术来建立服务系统模型、识别和设计服务,进而将这些服务组合起来形成服务系统。代表性方法包括以下几种^[15~17]。

(1) IBM 提出的“面向服务的建模与体系结构”(Service Oriented Modeling and Architecture, SOMA): 将服务系统分为多层,逐渐将传统的对象封装为构件、服务、服务组合、服务流程,进而被用户使用。整个开发过程分为三个大的阶段: 服务识别、服务规约设计、服务实现。在服务识别阶段,通过领域分解、目标服务建模等手段来建立服务模型,进而识别出一组可独立存在的服务。在服务规约设计阶段,针对每一个服务,从构件规约、服务接口规约、信息、事件/消息等方面展开设计。在服务实现阶段,选择特定技术将服务实现出来并按分层结构组织为系统。针对顾客和提供者两种角色,该方法论分别为其提供相应的方法支持。

(2) Sybase 提出的“面向服务的应用系统开发方法”(Service Oriented Development of Application, SODA): 包括服务建模、服务发现、服务设计、业务过程编排、封装与部署、测试等具体方法,并通过 Sybase WorkSpace 开发环境加以支持。它的主要优势在于: 可对服务模型进行各种分析与验证、支持灵活的复用、支持逆向工程、变化时的影响分析、代码自动生成等。

(3) 面向服务的统一过程(Service-Oriented Unified Process, SOUP)。它是一种使用统一过程(RUP)和敏捷过程(XP)中的最佳部分来构建和管理 SOA 项目的软件方法; 使用 RUP 来对新服务系统进行初始构建,使用 XP 对已存服务系统进行分析、装配、复用和优化。该过程如图 3-6^[17]所示,具体内容参见参考文献。