

图 7-1: SOA 实施建设方法

### 7.1.3 实施建设路线

实施 SOA 架构是一项 IT 战略目标，是一个持续改进的过程，必须在充分保护现有应用系统信息资源的基础上分阶段实施，迭代式推进，最终达到各级无线电管理机构信息技术架构模型的规范与统一，实现全组织范围内信息资源（服务）的共享和重用。

无线电管理一体化平台实施 SOA 架构的路线图：

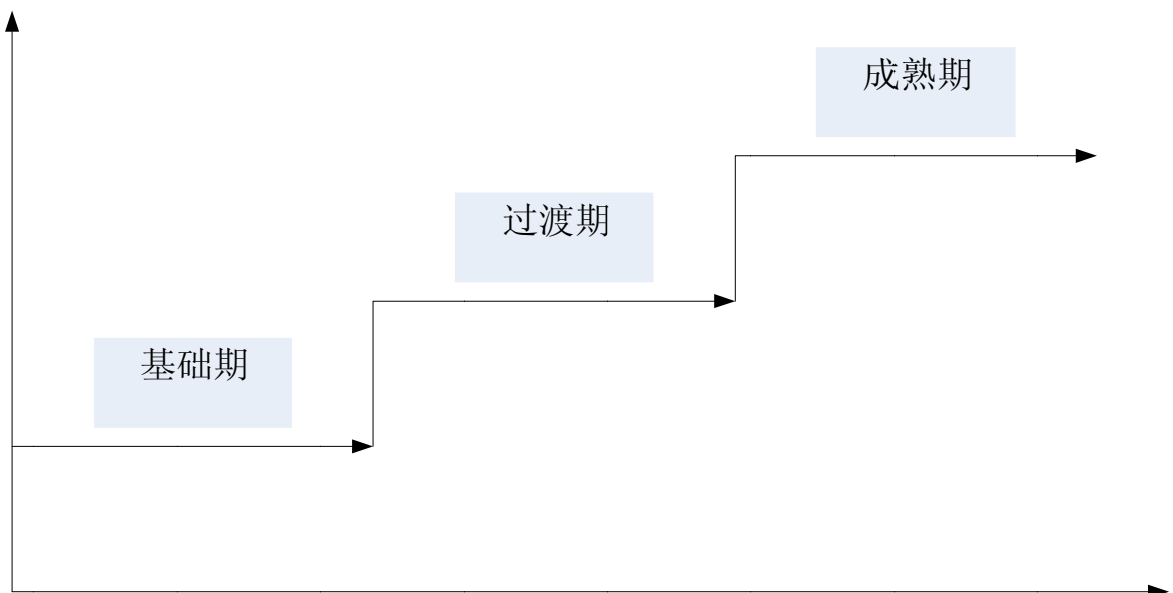


图 7-2: SOA 架构实施路线图

**a) 基础期**

无线电管理一体化平台目前主要还是以“烟囱式”的应用系统为主，数据分散在各个系统之中，系统与系统之间主要是点对点的连接和集成，数据孤岛、信息孤岛、业务孤岛现象严重，也缺乏总体技术架构和规范。因此，基础期重点放在构建一体化平台的基础架构，确定数据交换、服务总线、流程平台、以及门户等不同技术所适用的业务范围，开发、运行、管理环境的构建，以及基础性服务的发布并建立服务管理标准，其中服务基础架构的构建与设计参见附录 A；同时利用已有的应用系统资源，封装已有的数据和业务逻辑，积累可重用的服务资产和项目建设经验，统一应用集成的技术路线和 ESB；同时开展新建应用系统的试点项目工作。基础期工作任务包括但不限于：

- 1) 推广和应用信息资源规划成果，逐步统一数据资源环境；
- 2) 建设数据中心，开发和管理共享和交换的数据服务；
- 3) 按照 SOA 服务体系规范，开发各种应用集成的服务功能；
- 4) 统一对服务进行描述定义、注册和查找；
- 5) 通过 ESB 对服务进行统一管理，统一对服务的调用方式、数据格式；
- 6) 选定 ESB，实现基于 ESB 的应用系统建设；
- 7) 实现简单的服务组合与编排。

**b) 过渡期**

完成基础期工作后，在 SOA 的基本特性的基础上，加入一些高级特性。同时启动核心应用系统的改造，重新开发基于 SOA 架构的应用系统，逐步升级或替换掉原有的应用系统，实现从多种信息技术架构模型并存，逐步过渡到以 SOA 架构模型为主。过渡期工作任务包括但不限于：

- 1) 按照统一的 SOA 架构规范，改造或重新开发核心应用系统；
- 2) 扩展服务描述定义，增加服务质量（Qos）的属性和要求；
- 3) 按照 SOA 架构规范，对服务的安全性、消息传递的可靠性、事务的完整性进行

完善和管理；

- 4) 对服务的全生命周期进行管理，提供过程管理的流程；
- 5) 持续的业务流程梳理，使用集成开发工具定义流程，使用流程管理引擎运行流程，并监控流程的执行；
- 6) 持续的数据分析，提供数据深加工的能力，提供数据挖掘，辅助决策分析的服务。

#### c) 成熟期

在过渡期工作的基础上，通过持续不断的 SOA 治理，提高 SOA 能力成熟度，使得 SOA 架构能够快速适应业务的变化，通过对服务的组装和编排，能够快速搭建新的应用系统。成熟期工作任务包括但不限于：

- 1) 提供更完善的系统运行监控、分析管理工具，帮助各级人员更好的使用和管理无线电管理一体化平台系统；
- 2) 建立 SOA 能力成熟度评价模型，并执行评价和持续改进；
- 3) 建立与业务部门的回馈与交互机制；
- 4) 对服务元数据进行统一管理；
- 5) 服务和流程仿真，通过模拟优化服务与流程。

#### 7.1.4 实施建设步骤

基于无线电管理一体化平台的业务应用系统的建设主要包括新建开发和升级改造两种类型，实施建设步骤参见《无线电管理一体化平台实施开发规范》。

#### 7.2 服务生命周期

在基于无线电管理一体化平台的业务应用系统中，其 SOA 架构设计需参考由 OASIS 组织制定和发布的 SOA 参考模型——Reference Model for Service Oriented Architecture 1.0。

根据现状，无线电业务应用系统将采用“中间相遇”的策略实施面向服务，在实际业务场景中应用时，主要有以下三个步骤：

##### a) 大处着眼，小处着手：从一个核心流程做起

首先，利用业务流程分析方法识别核心业务流程。识别是否这些业务流程是多业务驱动



还是特定单业务驱动；多业务驱动的业务流程是通用流程，可以被共享，使得业务流程合并和标准化成为可能。

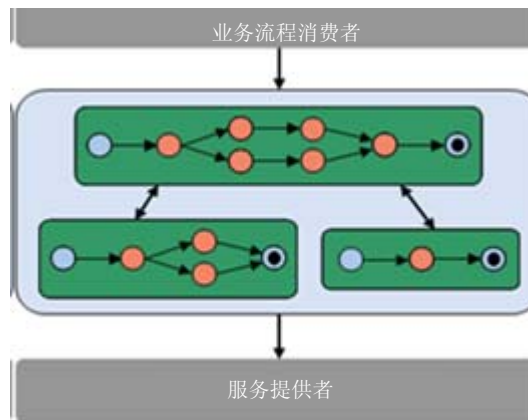


图 7-3：服务生命周期步骤一

**b) 自上而下：确定支持这个核心流程所需的各项服务**

采用自上而下的方式，确定支持这个核心流程所需的各项服务。业务流程分析及分解能够识别原子业务操作；有大部分的粒状业务操作都是通过业务流程分析确定的，并且在业务层面不能被进一步分解，称之为原子业务服务。

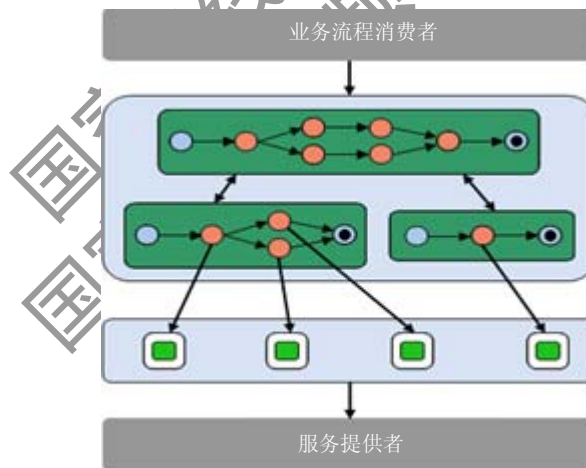


图 7-4：服务生命周期步骤二

**c) 由下而上：确定现有系统中可表示为服务，从而支持该核心流程的功能**

确定现有系统中可表示为服务，从而支持该核心流程的功能。将业务服务映射到潜在的服务提供者；在既有系统中识别能够对外暴露以支持业务服务的功能；根据业务服务的复杂度，其执行会有不同：

- 1) 通过服务提供者的单个功能就可以充分执行的简单服务
- 2) 跨越许多服务提供者，涉及许多功能的复合服务

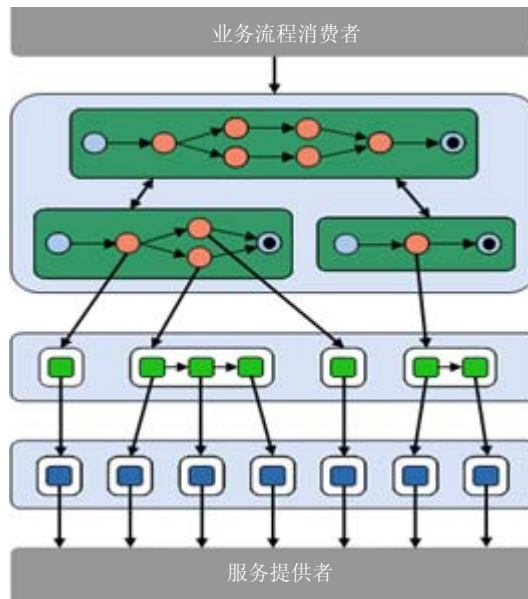


图 7-5：服务生命周期步骤三

从以上分析过程中的服务定义、装配、部署、管理等，即可称之为“服务生命周期”，则是关键所在。服务生命周期从项目初期（定义）开始，到完成（退出或者重新确定目标）时结束。服务生命周期跨三个阶段：需求和分析（Requirements & Analysis）、设计和开发（Design & Development），以及 IT 操作（IT Operations）。

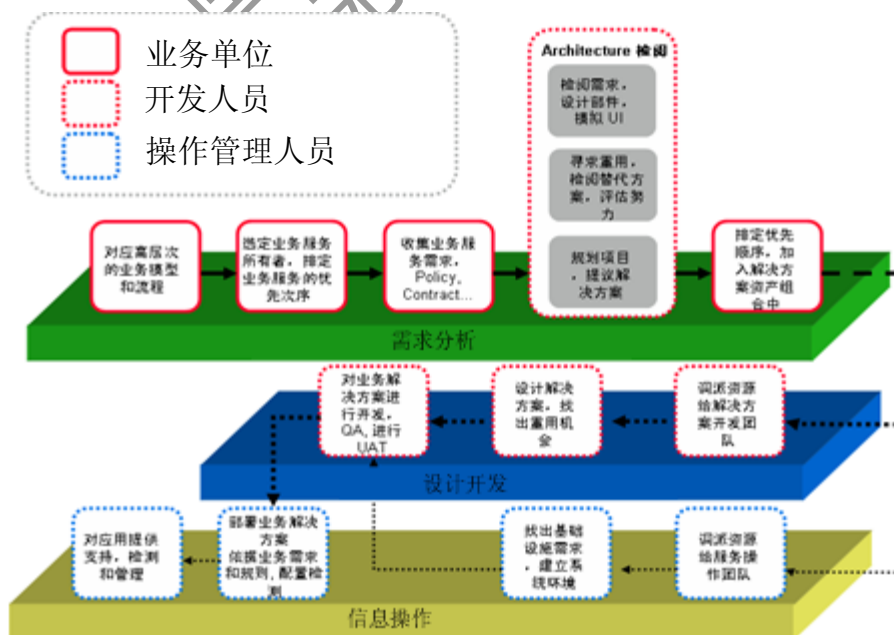


图 7-6: 服务生命周期概览

### 7.2.1 需求和分析

业务开始要标识业务需求并确定业务需求的优先级。根据标识的优先级，非技术人员可以与业务分析师紧密协作，以记录业务流程、规则和需求。高级需求包括：

- a) 从级别 0 开始向下可视化映射业务流程
- b) 定义每个业务流程
- c) 标识每个流程的业务所有者
- d) 标识目标和当前业务服务的差异
- e) 映射输入和输出数据元素
- f) 确定业务流程和业务服务的优先级
- g) 捕获业务服务定义的所有方面
- h) 模拟用户界面和/或业务流程

#### 7.2.1.1 关键考虑事项

在此阶段业务应该关注的关键考虑事项，包括业务动机和业务生命周期的差异：

- a) 记录业务动机可帮助将业务流程映射到服务。
- b) 即使服务生命周期是迭代的，它还是与应用程序生命周期相似。不过，服务生命周期的一个最佳实践是标识可能提供要求的功能的现有服务。设计人员首先审查现有服务，查看其是否适用；这可增加现有服务的重用并节约时间。

#### 7.2.1.2 最佳实践和要求

业务流程建模工具可在需求阶段帮助捕获业务流程。此工具通常由业务分析师基于 BPMN 和 UML 等标准使用工具中提供的通用行业表示法建模业务流程。

业务分析师与业务进行交互，以捕获事件、手动和自动活动以及业务规则。架构师可以作为观察员参与这些会议，以便更好地了解促进业务流程的基本原理。业务分析师与架构师进行交互，以便在分析阶段定义和改进下一个级别的详细信息。一旦业务分析师和架构师对最低级别的业务流程进行了建模，使得业务能够在不涉及技术细节的情况下进行定义，他们就可以合作审查服务储存库中的业务流程，以确定服务是已经存在还是需要进行构建。

在业务流程设计过程中，BPMN 工具可提供如下作用：

- a) 以业务流程的形式捕获所有的业务需求，例如活动、规则和策略
- b) 模拟端到端业务流程以标识瓶颈并改进整体流程
- c) 全局审查业务流程
- d) 捕获本地和地区需求
- e) 捕获手动和自动化流程

## 7.2.2 设计和开发

在设计阶段，业务分析师与架构师紧密协作，转换业务需求。架构师负责高级估计、设计并移交给开发团队。开发团队负责开发、装配、测试复合应用程序，并将其移交给 IT 操作团队。下面是一些高级设计需求：

- a) 审查需求并标识每个业务流程的备选方案
- b) 设计和估计每个服务的组件，例如门户、集成、基础架构、数据、策略和业务（逻辑）服务
- c) 标识业务服务的重用机会
- d) 开发和执行详细的项目计划
- e) 跟踪和报告业务和 IT 管理的进展
- f) 在提交每个业务服务时获得业务的签收

### 7.2.2.1 关键考虑事项

#### a) 企业级架构

IT 组织应将定义架构标准、开发流程、设计模式和工具的架构框架标准化。

#### b) 服务分类框架

服务分类框架可为服务提供设计开发基础，为获得灵活性架构所必需。服务可以分类为多个类别，如：

- 1) 面向服务参考架构，包括共享数据服务、业务流程和门户服务；
- 2) 服务组合，包括询价到现金服务和抵押批准服务；
- 3) 业务部门服务，包括销售和支持服务。

标识了服务后，团队可根据定义的标准对其进行分类。分类可帮助业务经理、项目经理和开发团队标识即将开发的服务以及开发目的。这样项目经理可以更容易地将开发任务分发给合适的团队。

#### **c) 服务粒度**

服务粒度指抽象级别或者服务包含的功能。团队还可以将粒度的概念应用于服务本身或服务方法。

确定服务粒度时，架构师需要考虑性能需求。细粒度服务（例如 Enterprise Java Bean, EJB）通常更容易了解和实现，因为许多情况下大多数工作都已经完成。不过，考虑到性能，对其服务与现有 EJB 可能“不是”最优的解决方案。依赖多个请求和响应对应的细粒度架构提供的性能可能较低。为了降低网络延迟、系统 I/O 和线程/进程等待状态的影响，最好创建一个在内部组合多个业务域服务并且使用的消息较少的粗粒度服务。

服务粒度必须考虑到旧系统接口并未注意到（而且必须保持）新协议。架构师必须仔细地进行计划，以避免在以服务的形式添加新数据通道时更改旧系统。

最后，在确定粒度时，架构师必须考虑未来可能进行的更改对基础实现的影响。通常情况下，使用粗粒度表面或模式来隐藏其中的细粒度服务对业务十分有用。这样做的目的是将服务与对基础实现的更改隔离开来，方法是考虑到未来的扩展不影响服务的粒度级别进行设计。

#### **d) 重用策略**

组件的重用始于设计层。架构师应该设计组件，以便使用组件的客户只需执行或继承完成给定任务所需的方法。共享组件作为执行任务同时隐藏其复杂度的独立元素写入。架构师应该考虑使用表面模式，因为它倾向于封装通用接口中的相关任务或类的每一个成员以使客户端代码可以交换使用这些任务。因为任务通常编码为一个或少数几个隔离的方法，封装强调执行单个任务的方法重用。重用单个任务比重用包含代码和数据的对象容易，后者可能需要执行多个任务。

### **7.2.2.2 最佳实践和要求**

#### **a) SCA 规范**

服务设计与开发应遵循 SCA 规范。SCA 规范定义了服务描述、服务依赖关系、服务实现、服务组合以及复合应用程序开发的部署和运行时等。

#### b) 服务注册

服务注册基于 UDDI，是所有部署服务的记录系统。它包含的元数据包括服务定义、服务依赖关系和服务接口。此外，可以借助于 UDDI 服务注册库发现服务的潜在重用，并在标识进行重用的服务前先审查 SLA 和服务详细信息。

#### c) 信息建模

下面是一些用于信息建模的最佳实践

- 1) 在系统、业务所有者和记录系统之间标识所有系统、数据流和属性，然后强制每个项目团队必须更新此模型
- 2) 在捕获业务要求的同时标识数据流
- 3) 标识共享数据服务和主要数据实体，同时审查应用程序设计
- 4) 为每个自定义应用程序开发 ERD（实体关系图）

在任何时候修改打包的或自定义应用程序时更新 ERD。

### 7.2.3 IT 操作

IT 操作团队负责开发、测试和生产，其中生产环境具有最高的优先级。IT 操作团队负责调整网络和数据中心的大小。此外，IT 操作团队还负责部署和监控 IT 支持的所有应用程序，并为其提供第 1 层支持。下面是一些高级需求：

- a) 审查要求并标识基础架构需要
- b) 建立系统环境，包括开发、系统集成测试、性能测试、用户验收和生产环境
- c) 帮助解决方案开发团队进行系统和应用程序配置、定期构建和容量计划
- d) 跟踪和管理服务和资产之间的依赖关系
- e) 部署和管理生产环境中的业务服务
- f) 根据业务优先级为业务服务提供应用程序支持

#### 7.2.3.1 关键考虑事项

服务生命周期的这个阶段存在两个关键的考虑事项：服务部署以及服务管理和监控。

复合应用程序包括可通过各种技术实现的服务或服务组合。版本管理要求服务级别的计划构建，还要求提供为目标节点装配服务的功能。此外，IT 操作团队需要在装配和供应服务的的同时来组合修改服务和实例配置参数的计划和流程。

IT 操作团队可能还需要一个服务装配管理服务来帮助管理各种构建、装配目标节点的服务以及管理资产。市场上存在一些可以提供此功能的产品，但不幸的是并不存在任何有关服务打包或服务供应的标准。这个领域可能需要 SCA 标准解决。

服务管理和监控可细分为两个领域：

- a) 服务器管理：配置部署的服务和服务器实例，最好是使用通用的管理控制台。如果可能，最好提供跨所有产品和所有供应商的一致性。
- b) 服务器监控：应用监控策略以捕获每个服务的要求的指标。此指标应该基于在服务生命周期的需求和分析阶段定义的 KPI 和管理策略。团队还需要收集和关联监控事件，以确保遵循业务服务级别协议（SLA），并将聚合的管理摘要信息发布到服务注册表和/或面向服务储存库。

### 7.2.3.2 最佳实践和要求

当开发团队准备好将应用程序部署到开发、测试或生产环境时，他们可从了解服务生命周期此阶段的高级最佳实践和需求中获益：

#### a) 版本管理工具

- 1) 利用现有应用程序生命周期管理工具或源控制管理系统支持版本管理；
- 2) 利用面向服务储存库进行服务治理；
- 3) 审查可帮助在单个服务级别构建和部署应用程序的新产品的市场；
- 4) 让版本管理团队从项目一开始即参与其中。

#### b) 部署工具

- 1) 利用现有部署工具和过程；
- 2) 利用部署工具跟踪对运行时环境的所有更改以满足遵从性要求；
- 3) 与程序管理办公室协调部署；
- 4) 使用部署工具将服务从一个网络拓扑重新装配到另一个网络拓扑。

## 7.2.4 服务治理

服务是在应用程序之间共享的，因此组织必须谨慎地设计、开发和部署服务才能确保不影响服务的现有消费者。存在优先级冲突的各种组织机构之间共享服务。有效的治理可帮助确保实现最大程度的复用性并使中断降到最少。面向服务治理功能的主要责任是：

- a) 发布面向服务标准和最佳实践
- b) 定义和执行流程以便在项目级别改进服务的使用和重用
- c) 管理 LOB 的所有共享服务
- d) 跨组织传播标准和最佳实践
- e) 在组织内宣传面向服务的成效

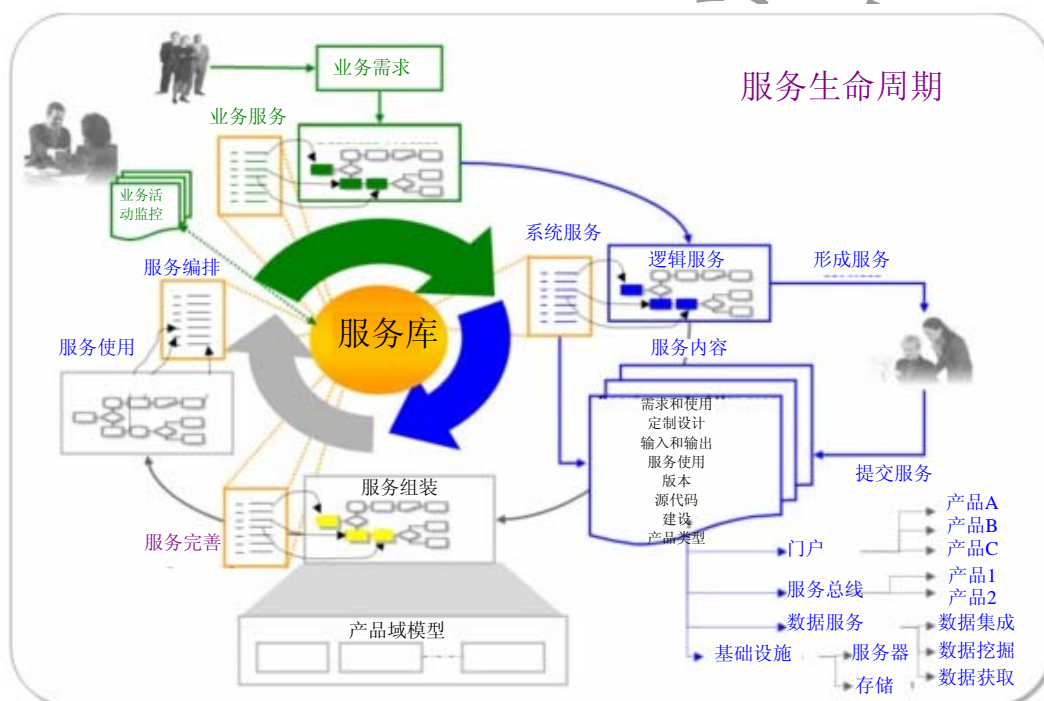


图 7-8： 服务生命周期治理



## 8. 实施的组织结构

### 8.1 组织结构

有效的组织结构是项目成功的有力保证，面向服务的应用系统建设项目的参考组织结构如下：

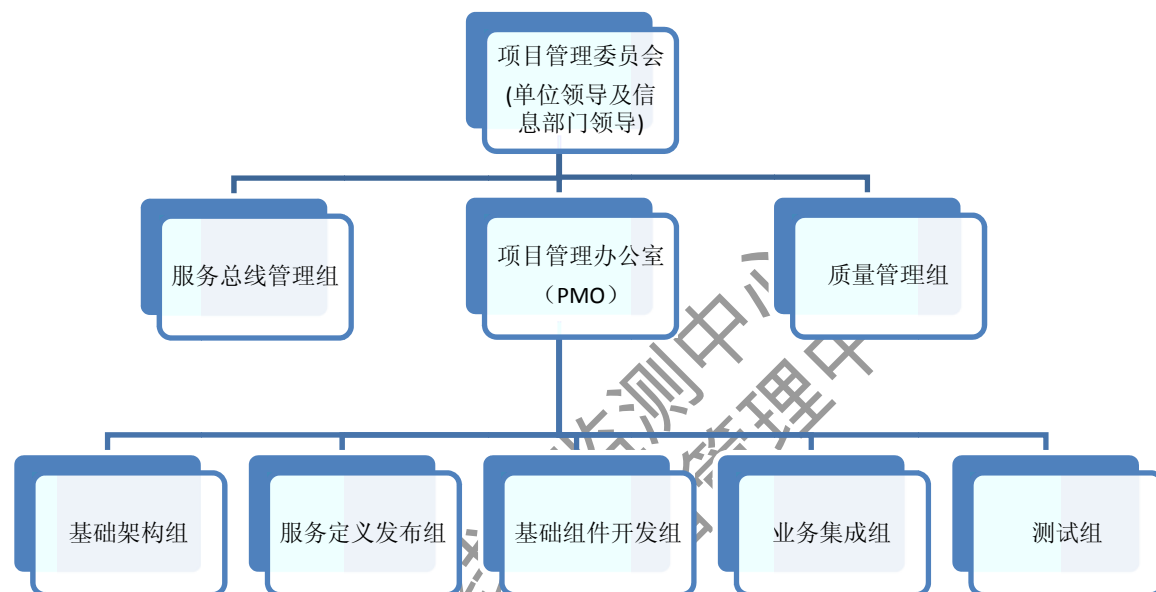


图 8-1: SOA 项目实施组织结构

#### 8.1.1 项目管理委员会

项目管理委员会负责监督并指导项目的实施进程，定期审核项目经理就项目进展执行情况的书面报告，对项目中的重大问题做出决策，协调解决重大问题和突发事件，决定对项目经理的任免。

项目管理委员会由各级无线电管理机构高层领导与信息部门领导共同组成。

#### 8.1.2 服务总线管理组

向项目管理委员会负责，在项目实施过程中进行服务标准和原则的控制，在项目实施完毕后，由该组织管理和批准新的服务发布和渠道系统的接入。同时负责制定实施 SOA 项目的总体规划，从全局业务的角度而非项目级参与项目管理。服务总线管理组由各级无线电管理机构架构师和实施公司（如由外部供应商辅助实施）架构师共同组成。

### 8.1.3 项目管理组

负责向项目管理委员会定期报告项目进展情况，就项目中存在的问题提出解决建议，对项目进行有计划地组织管理，并检查项目进展情况。项目管理组由各级无线电管理机构项目负责人和实施公司（如由外部供应商辅助实施）项目经理和技术负责人共同组成。

### 8.1.4 基础架构组

负责基础架构的设计和流程建模设计。设计整体基础架构，完成项目范围内的规划，需考虑在建项目与整体 IT 架构的一致性。

### 8.1.5 质量管理组

直接隶属项目管理委员会，按制定的标准及控制手段执行进度管理，风险管理，全面的执行各项局方及业内规定的质量标准和工作流程。

### 8.1.6 服务定义发布组

负责在总线上发布服务和设定服务标准。根据基础架构规划中的服务架构，对服务进行归类，根据服务定义模板，完成服务的识别、设定和在服务总线上的发布和配置。

### 8.1.7 基础组件开发组

负责基础的，公共的组件的统一开发；开发从日志，安全到各种便利工具的公共组件，完成在服务总线之上的各种组件的扩展工作，如扩展函数，扩展报文转换方法，扩展监控处理模块，进行监控平台的集成等。

### 8.1.8 业务集成组

负责业务系统集成中的各种具体模块开发，各种参数，通讯模式配置，以及所需的服务总线周边资源的配置和部署。

### 8.1.9 测试组

负责系统的联合测试工作，在项目质量方针指导下，进行测试管理，制定设计系统测试计划、测试方案、测试案例、各项测试、形成测试报告并对测试结果进行跟踪，包括不同阶段的测试工作。

## 8.2 组织结构

### a) 项目经理

<b>角色描述:</b>	项目经理与架构师协作来协调高级项目的交付。就小项目而言，架构师可以担任项目经理的角色。此资源创建一个详细的项目交付时间安排并跟踪交付。项目经理在进行业务级决策时还可以协调更高级
--------------	--

	业务资源的参与，例如项目的相关人员。变更控制过程由项目经理进行组织和推进。
<b>职责:</b>	<ul style="list-style-type: none"> <li>a) 开发、维护和跟踪项目交付的时间安排</li> <li>b) 评审 <ul style="list-style-type: none"> <li>1) 项目架构</li> <li>2) 测试规划</li> <li>3) 测试结果</li> </ul> </li> <li>c) 协调 <ul style="list-style-type: none"> <li>1) 变更控制工作</li> <li>2) 技术获得（硬件和软件）</li> <li>3) 部署规划</li> <li>4) 向合适的开发人员指派缺陷</li> </ul> </li> <li>d) 跟踪 <ul style="list-style-type: none"> <li>1) 环境设定</li> <li>2) 设计</li> <li>3) 实现</li> <li>4) 测试</li> <li>5) 缺陷</li> <li>6) 计划成功标准</li> </ul> </li> <li>e) 报告计划进展</li> </ul>
<b>交付成果:</b>	<ul style="list-style-type: none"> <li>a) 项目议程</li> <li>b) 状态报告</li> <li>c) 通过变更控制板变更文档需求</li> </ul>

**b) 架构师**

<b>角色描述:</b>	架构师负责计划的技术交付。此资源将协调开发需求文档的用例以及应用程序架构。架构师有责任全程监督和协调该计划直到成功交付。
--------------	--

<b>职责:</b>	<ul style="list-style-type: none"> <li>a) 用例开发/协调</li> <li>b) 开发和维护项目架构</li> <li>c) 开发和维护项目操作手册</li> <li>d) 评审 <ul style="list-style-type: none"> <li>1) 测试计划</li> <li>2) 测试结果</li> <li>3) 设计</li> </ul> </li> <li>e) 协调 <ul style="list-style-type: none"> <li>1) 实现</li> <li>2) 代码评审</li> <li>3) 测试执行</li> <li>4) 部署规划</li> <li>5) 项目部署</li> <li>6) 变更控制</li> </ul> </li> <li>f) 帮助 <ul style="list-style-type: none"> <li>1) 环境设定</li> <li>2) 实现</li> <li>3) 环境调优</li> </ul> </li> </ul>
<b>交付成果:</b>	<ul style="list-style-type: none"> <li>a) 用例文档</li> <li>b) 架构细则文档</li> <li>c) 部署计划</li> <li>d) 操作手册</li> <li>e) 源代码（如果时间允许）</li> </ul>

**c) 业务分析师**

<b>角色描述:</b>	<p>有业务分析师在场时，他们将和架构师、项目经理协作开发需求文档中的用例。业务分析师还将参与变更控制工作。</p>
<b>职责:</b>	<ul style="list-style-type: none"> <li>a) 开发需求文档中的用例</li> </ul>

	<ul style="list-style-type: none"> <li>b) 参与变更控制工作</li> <li>c) 协助用户验收测试</li> </ul>
<b>可交付项:</b>	用例文档

**d) 开发人员**

<b>角色描述:</b>	开发人员主要负责项目的设计和实现
<b>职责:</b>	<ul style="list-style-type: none"> <li>a) 负责指派的项目子系统设计</li> <li>b) 开发 <ul style="list-style-type: none"> <li>1) 指派的项目子系统的源代码</li> <li>2) 使代码遵从编码指南</li> <li>3) 单元测试代码</li> <li>4) 通过源代码存储库交付代码</li> </ul> </li> <li>c) 环境设定 <ul style="list-style-type: none"> <li>1) 管理单机开发环境</li> <li>2) 协助产品、测试、升级和开发环境的设定</li> </ul> </li> <li>d) 维护 <ul style="list-style-type: none"> <li>1) 处理指派给自身项目子系统的缺陷</li> <li>2) 通过源代码存储库来维护代码版本</li> </ul> </li> <li>e) 参与代码评审</li> </ul>
<b>交付成果:</b>	<ul style="list-style-type: none"> <li>a) 设计文档</li> <li>b) 源代码</li> </ul>

**e) 测试主管**

<b>角色描述:</b>	测试主管负责开发项目的总体测试策略和规划。
<b>职责:</b>	<ul style="list-style-type: none"> <li>a) 规划 <ul style="list-style-type: none"> <li>1) 开发功能测试计划</li> <li>2) 开发用户验收测试计划</li> <li>3) 开发系统测试计划</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>4) 开发回归测试计划</li> <li>b) 协调 <ul style="list-style-type: none"> <li>1) 测试用例开发</li> <li>2) 测试执行（功能、UAT、系统、回归和部署）</li> </ul> </li> <li>c) 跟踪 <ul style="list-style-type: none"> <li>1) 缺陷</li> <li>2) 成功标准</li> <li>3) 部署</li> </ul> </li> <li>d) 报告测试进展和状态</li> </ul>
<b>交付成果:</b>	<ul style="list-style-type: none"> <li>a) 测试策略</li> <li>b) 功能测试计划</li> <li>c) 用户验收测试计划</li> <li>d) 系统测试计划</li> <li>e) 回归测试计划</li> </ul>

**f) 测试开发人员**

<b>角色描述:</b>	<p>测试开发人员负责测试安全的实现和执行。测试开发人员实现和执行测试计划。这些资源还报告测试过程中遇到的缺陷问题，并在进行修复报告时重新测试来验证。</p>
<b>职责:</b>	<ul style="list-style-type: none"> <li>a) 审查测试计划</li> <li>b) 开发测试脚本</li> <li>c) 执行测试用例</li> <li>d) 跟踪缺陷修复</li> <li>e) 验证修复的缺陷</li> <li>f) 报告测试执行结果</li> <li>g) 基于失败的测试用例公开缺陷</li> </ul>
<b>交付成果:</b>	<ul style="list-style-type: none"> <li>a) 测试脚本</li> <li>b) 测试结果</li> </ul>

### g) 数据库管理员

<b>角色描述:</b>	数据库管理员负责项目所要求的所有与技术数据有关的活动
<b>职责:</b>	<ul style="list-style-type: none"><li>a) 数据库设定<ul style="list-style-type: none"><li>1) 项目的数据库设计</li><li>2) 创建/更新数据库模式</li><li>3) 创建初步索引</li><li>4) 分配用户和应用程序访问权限</li></ul></li><li>b) 支持<ul style="list-style-type: none"><li>1) 分析访问模式以便于最大程度优化数据库</li><li>2) 推荐对数据访问方式进行应用程序变更以提高效率</li></ul></li><li>c) 在系统和压力测试之后调优数据库以实现最优性能。</li></ul>
<b>交付成果:</b>	<ul style="list-style-type: none"><li>a) 数据库设定</li><li>b) 数据库模式</li><li>c) 数据库维护和调优指南</li></ul>

### h) 系统管理员

<b>角色描述:</b>	系统管理员是硬件和软件之间的纽带。他们可以建立并提供系统部署过程中需要安装和使用的硬件和软件。
<b>职责:</b>	<ul style="list-style-type: none"><li>a) 环境设定<ul style="list-style-type: none"><li>1) 软件安装</li><li>2) 软件部署</li><li>3) 软件配置</li></ul></li><li>b) 安全<ul style="list-style-type: none"><li>访问控制和用户访问权限</li></ul></li><li>c) 环境调优的测试支持</li></ul>
<b>交付成果:</b>	<ul style="list-style-type: none"><li>a) 操作开发、测试、分割和生产环境（如果适用）</li></ul>

## 9. 实施标准及规范

面向服务的系统建设过程中，需至少遵循/制定如下标准及规范：

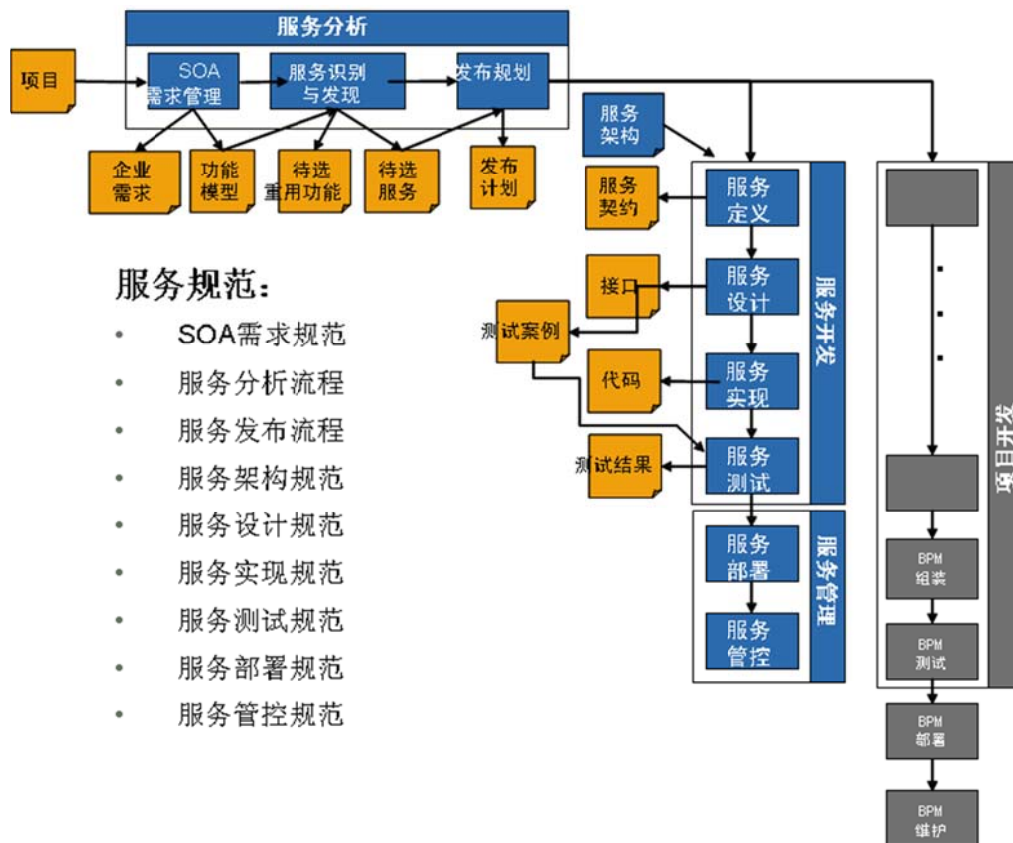


图 9-1 面向服务项目实施标准及规范

详细实施标准及规范参见：《无线电管理一体化平台实施开发规范》。



## 附录 A

### 服务基础架构构建

基础架构的构建是一体化平台建设中非常重要的一个环节,直接影响一体化平台的实施效果。虽然通用中间件产品本身已经提供了通讯接入,内容处理,路由控制等大量的服务总线功能,但是这些功能过于基础和细粒度,因此需要经过架构上的组合,才能形成更适合于无线电一体化平台的基础架构。

**处理架构设计**将完成服务总线平台、业务流程平台之间架构上的划分,确定服务消费者和服务提供者之间的调用关系。

**服务架构设计**将对一体化平台的服务层次划分、用途、使用原则进行规划。

**安全设计**将解决从总线服务访问、数据传输到安全审计的原则和功能实现。

**监控与管理设计**将建立面向业务的监控管理方案。

并通过开发和测试将完成基础架构中所需的各种组件封装、开发,以及架构本身的功能性及扩展性测试。

#### A.1 处理架构设计

处理架构参考架构:

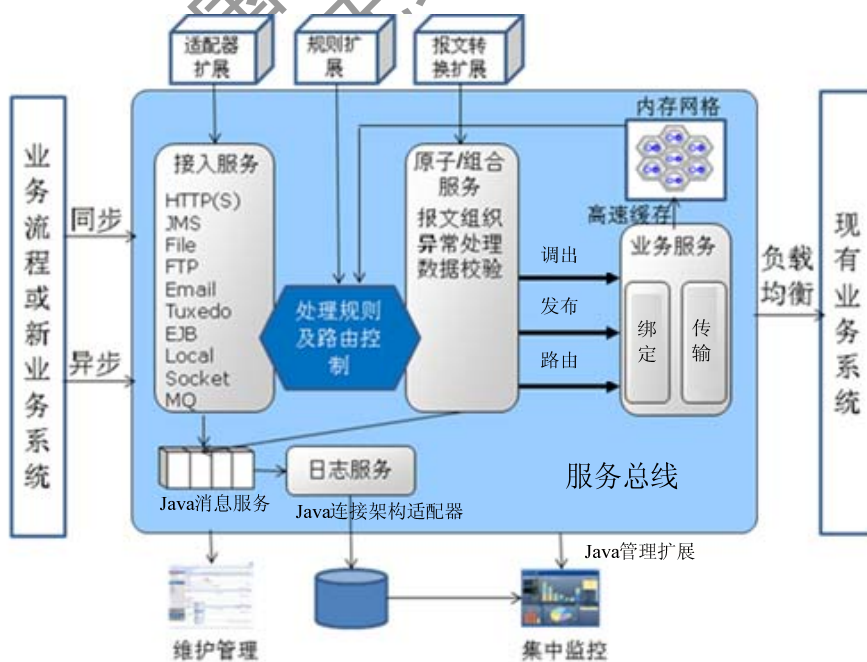


图 A-1：处理架构参考架构图

在总线处理架构中需要解决总线运行结构，公共功能以及监控管理方面的架构问题，并确保整个总线架构的可扩展性。

将对业务流程、或新的业务系统接入的请求进行标准化处理，以便服务总线可以按照统一的模式进行后续的相应处理。

处理规则以及路由控制，通过动态路由的方式实现前端请求和总线服务的匹配，并根据无线电管理的业务特点建立起灵活可扩展的架构。

原子服务/组合服务是利用服务总线的代理服务功能构建的众多独立的维护单元，以满足无线电管理众多服务不断发展完善的需要。

统一的日志处理模块统一汇集各个处理环节中产生的日志信息，以异步的方式写入数据库或相应的系统，减少日志处理对交易时间的影响。

内存网格可以为总线平台提供高速的业务数据和服务元数据缓存，提高总线平台在访问公共数据和重复数据时的处理效率。

## A.2 服务架构设计

SOA 服务参考架构如下：

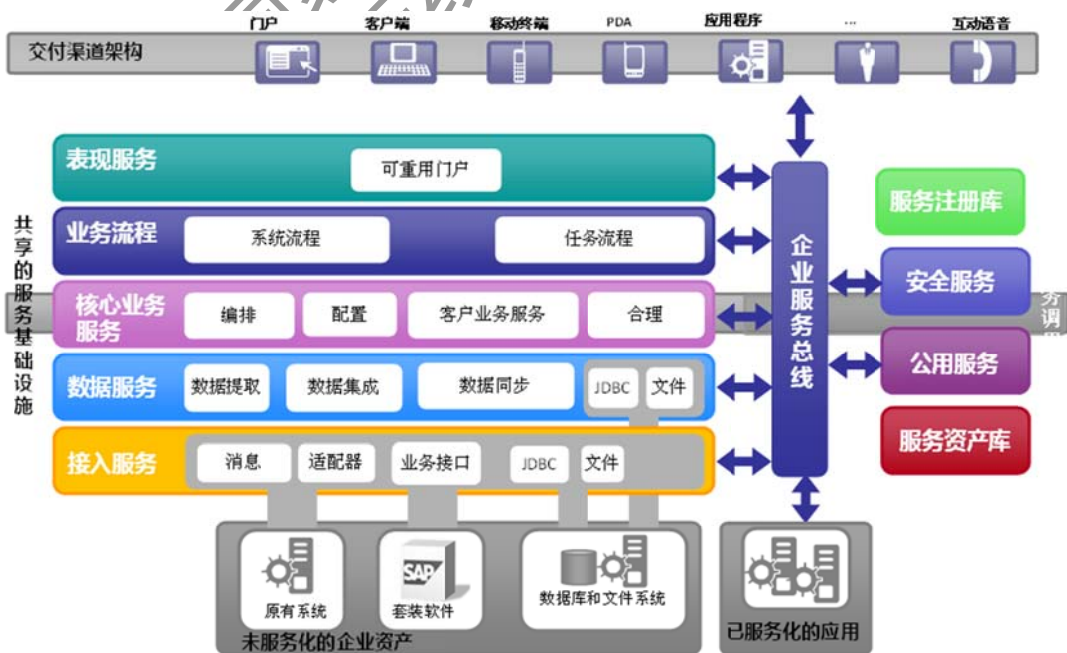


图 A-2：服务架构设计参考

SOA 基础设施提供了一个管理、监测并协调所有服务请求的环境，它既是开发环境，也是运行环境。SOA 基础设施（SOA infrastructure）的最前端为企业服务总线，任何服务调用者提出的请求都将通过服务总线。服务总线的作用类似于一个服务中介，中介的责任是协调和调度业务服务，它的功能包括路由、转换、聚合、递增等。作为调度过程的一部分，它也提供了包括审核、日志、监测和安全等公用服务。

运行在 SOA 基础设施之上的服务被划分为几种基本服务类型：连接服务、数据服务、业务服务、业务流程服务和表示服务。这些基本服务进行组合可以形成组合应用。

### A.3 安全设计

安全基础设施是与 SOA 架构紧密关联的一个重要的 IT 基础设施。安全基础设施为 SOA 环境提供一个集中的安全控制框架，为 SOA 架构内部所有的组件（同样也为所有应用系统）提供统一的安全服务，以达到统一用户身份管理、认证和授权。

在目标架构里，服务总线作为所有服务调用的入口，必须进行服务调用的安全控制。安全控制的机制至少应包括认证、授权、消息层安全及传输层安全等

安全控制是通过安全服务实现的。安全服务可以被看作是公用服务的一种，可以被所有的服务，甚至应用系统使用。

安全服务由安全基础设施提供。目标架构中使用的安全服务如下：

**认证服务：**调用者访问 SOA 基础设施服务平台上的服务时，应该将能够表明自身身份的用户凭证随业务调用报文一起发送到平台，平台调用“认证服务”进行用户的身份验证，防止用户身份欺诈。

**授权服务：**用户身份经过验证之后，SOA 基础设施服务平台应调用“授权服务”。授权服务查找该用户所属的角色，获得权限规则。如果该用户拥有这个服务的访问权限，则允许它调用服务，否则拒绝访问。

**传输层安全服务：**由于传输层安全机制会对所有传输的数据流进行加密，增加了服务器的工作量，因此如果没有合适的硬件加速方案，传输层安全机制应该被有针对性地使用。通常，当网络自身无法提供安全保证，并且是跨地域的消息传输，就应该采用传输层安全

(Transport Level Security) 机制。

**消息层安全服务:**当传输的消息包含敏感信息,而网络层面无法提供安全保证的情况下,就有需要进行消息级别的安全机制设置。在消息级别加密机制中,消息的发起端会对消息的全部或部分进行加密;相应地,消息接收端对加密后消息进行解密。在消息的传输过程中,其它组件或中介都因为缺乏密钥而无法对消息进行解密,这将保证消息端到端的安全传输。与传输层加密相比,它比较有效率。同时,因为加密是端对端的,服务总线这样的信息处理中间节点是看不到加密后的内容的。

#### A.4 监控与管理设计

随着共享服务增加,基础设施服务的完善,有必要建立服务体系度量,使服务总线系统形成完整的闭环。服务体系度量的结果,将由运维管理部门回馈给架构人员和业务流程分析人员,用于服务总线平台的持续改进。

该管理和监控,包括但不限于项目或者具体应用中。通过对该管理和监控方法的利用,能够使 IT 对业务的支持变得更加自然。更进一步,随着 IT 对与业务需求的相应的更加熟悉,那么 IT 针对业务的理解而提供的服务将更加能够体现业务的利益。这种结果,只有 IT 通过测量和收集一些性能数据,并通过这些数据来提高和改善整体的性能,才能够得以实现。

SOA 管理和监控提供了一整套的方法,用于衡量 SOA 的执行,并用于对 SOA 的提高。

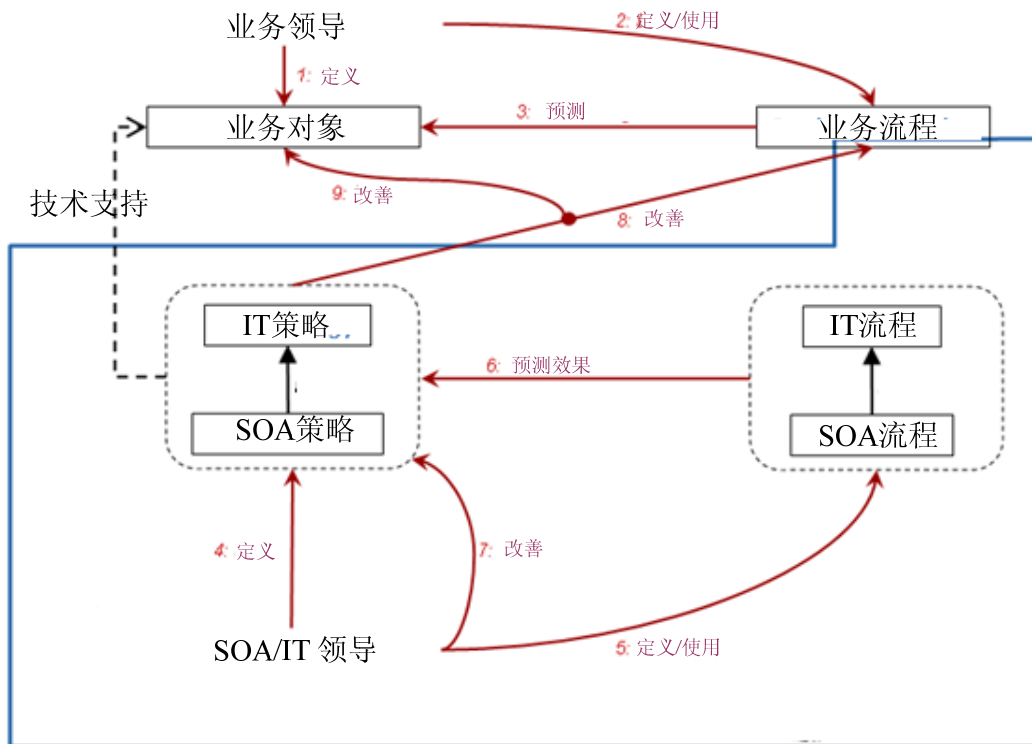


图 A-3: SOA 管理和监控方法

业务人员通常会定义一些主要的业务目标。在通常情况下，他们也会定义或者使用一些业务方法来衡量针对这个业务目标的业务实现的效率。通俗的说就是有一套方法能够告诉业务人员，业务的目标是否已经实现。SOA 是一个 IT 的策略，它的负责人为了支持业务的目标而定义了 SOA 的策略。SOA 的管理和监控作为这个策略的一部分，定义了一个主要的 SOA 的测评方法，来测量 SOA 策略的执行情况。如果该策略被正确的执行，而且 IT 以及达到或者超过了它的目标，则其对是否达到业务目标的衡量也将得到提高。同时也也能提高业务目标实现的能力。

具体的，服务总线的管理与监控系统将包含：

- a) **服务工程规程有效性度量体系：**Service 计分卡，服务重用评价，一致性度量，交付效率等。
- b) **基础设施运行度量体系：**SLA，审计信息，服务使用率，异常及失败，服务消费能力。
- c) **业务流程度量体系：**业务活动监控，SLA 目标，异常及失败统计监控。

- 服务水平
  - 服务平均响应时间
  - 服务故障率
  - 服务排队长度
- 处理能力
  - 系统总TPS
  - 单位资源处理能力
  - 按渠道、服务号的服务量占比
- 趋势
  - 交易量时间分布
  - 交易量变化趋势
  - 响应时间趋势
- 特殊状态监控
  - 交易冲正
  - 交易失败
  - 交易拒绝
- 特殊业务监控
  - 密码重试
  - 账户锁定

图 A-4：业务流程度量体系指标

服务监控是服务应用过程的总结和报告管理等，它的成功应用可以为服务开发、服务路由以及服务使用提供如何进行优化的依据，从而进一步提高服务质量。

服务监控提供以下功能：

- a) 管理和维护所有服务级别；
- b) 检测和管理异常情况；
- c) 审查和监控业务事务，并提供审查动态事务的功能；
- d) 管理服务生命周期并在部署前进行验证；
- e) 跨不同的系统提供非插入式服务发现功能；
- f) 实时性监控服务的应用状态；
- g) 自动化异常警告；
- h) 根据监控数据为服务安全提供策略执行依据；
- i) 提供策略管理，并预定义主流策略；
- j) 支持集中定义安全性和身份管理策略，再在全局进行实施；
- k) 通过策略更新来自动解决问题。