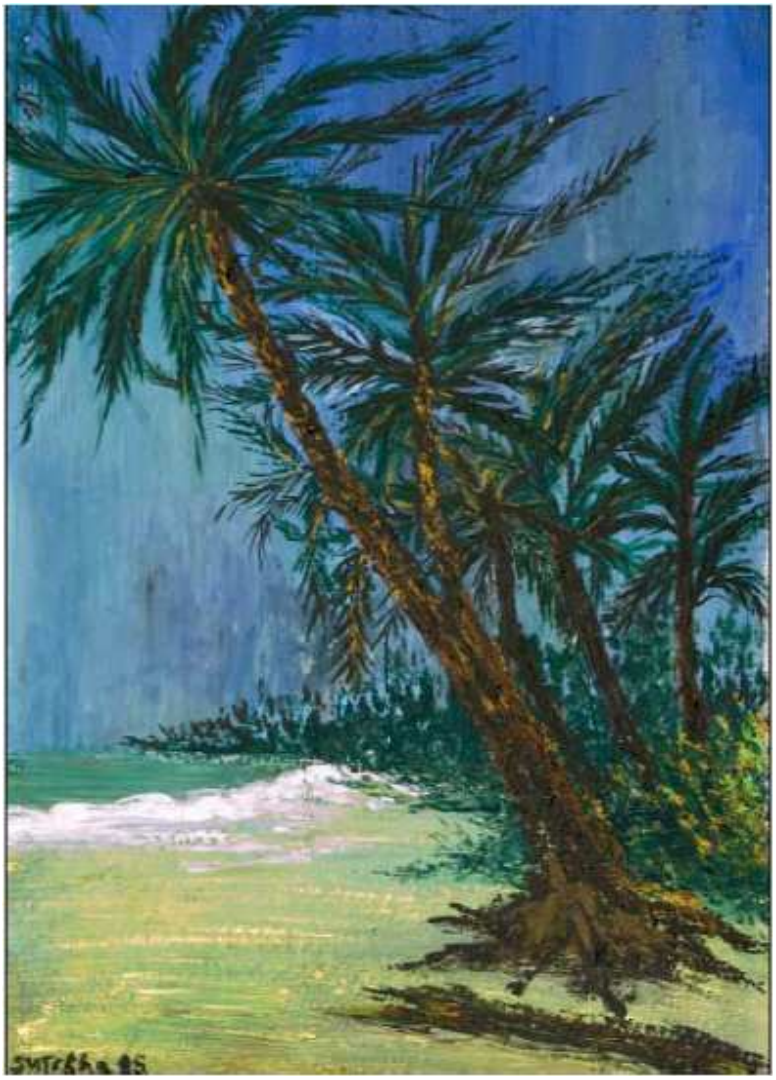
# SOA专业人员指南

# 第3部分

# *服务生命周期简介*



*作者：Surekha Durvasula*

**参与本书的SOA专业人员**

Surekha Durvasula，Kohls，企业架构师

Martin Guttmann，Intel公司Customer Solutions Group，首席架构师

Ashok Kumar，Avis/Budget，SOA Architecture经理

Jeffery Lamb，Wells Fargo，企业架构师

Tom Mitchell，Wells Fargo Private Client Services，首席技术架构师

Burc Oral，个人贡献者

Yogish Pai，BEA Systems, Inc.，AquaLogic Composer首席架构师

Tom Sedlack，SunTrust Banks, Inc.，企业架构工程师

Dr Harsh Sharma，MetLife，高级信息架构师

Sankar Ram Sundaresan，HP-IT，首席电子商务架构师

**审校人员**

Prasanna Deshmukh，WebEx Communications，架构主管

Noam Fraenkel，Mercury Interactive，首席IT技术官

Steve Jones，Capgemini Group的，Application Development Transformation部门，首席技术官

Brenda Michelson，Elemental Links, Inc.，首席顾问及分析师

Ashok Nair，关于EAI和信息技术服务的管理系统分析师，卡尔加里城

George Paolini，Georgepaolini.com，顾问

Jeff Pendelton，SOA Alliance，执行主管

Annie Shum，BEA，SOA战略副总裁

作者十分感谢为本文档做出贡献的组织和个人，他们撰写了部分文档，进行了大量编辑工作，还帮助审校了文档并提供反馈。此外，作者还要特别感谢BEA Systems, Inc.，是这个公司提供了本文档中的开发和演示所需的基础架构和平台。

# 目　录

# 1 关于本文档

## 1.1 摘要

SOA是一门相对较新的技术，所以很多致力于实现它的公司无法获得良好的实践专业指导。如果没有基于共同经历的通用语言和行业词汇，SOA技术很可能最终只会为IT基础架构带来更多自定义逻辑并提高其复杂度，而无法实现其提供企业内和企业间服务重用和流程互操作性的承诺目标。为了帮助开发共享语言和有关SOA的知识集，一群SOA专业人员创建了《SOA专业人员指南》这个文档系列。在这个文档系列中，这些SOA专家描述并评注了一些与SOA相关的最佳实践和关键知识，以帮助其他公司迎接SOA的挑战。他们对《SOA专业人员指南》的期望是，这个分为多个部分的知识集在出版之后，能够作为所有SOA相关人员的标准参考百科全书。

## 1.2 目标受众

本文档的目标受众为：

- 需要跨企业/LOB启动和管理SOA战略的业务和IT领导
- 需要促进实现SOA计划的理想和路线图以及其中包含的每个实现的架构的企业架构师
- 需要管理SOA整体业务战略中的子项目组合的程序管理员
- 需要映射依赖关系并开发满足业务期望的时间线的项目团队成员
- 为业务和IT提供新业务能力的解决方案和工具的供应商
- 需要更好地了解业务和IT计划如何利用技术实现其目标的用例的标准机构。

## 1.3 《SOA 专业人员指南》的优点

本文档可帮助读者：

- 向他人学习：很早就开始使用SOA的用户可共享其有关跨行业采用SOA的最佳实践、看法以及观点
- 比较替代产品：识别和定义SOA的关键技术组件，建立对选项比较的基线参考
- 改进合作：使用一种通用语言澄清本文档中定义的SOA组件的性质
- 加速实现：本指南定义了其服务生命周期和需求、建议的工具以及每个阶段的最佳实践。
- 了解标准的价值：本文档建议了用于SOA的各个方面的标准
- 避免潜在风险：本指南指出了供应商社区尚未指出的一些问题领域。

## 1.4 SOA 专业人员指南：部分

《SOA专业人员指南》包含三个单独的章节。

第1部分（《为何使用面向服务的架构？》）提供了一个SOA高级摘要。

第2章《SOA参考架构》提供企业级SOA实现的工作设计，包括详细的架构图、组件描述、详细要求、设计模式、有关标准的观点、法规遵从性模式、标准模板以及成员的潜在代码资产。

本文档是第3章，标题为《服务生命周期简介》，它提供了在整个服务生命周期（从项目初期到完成或重新定位服务）中提供服务管理的详细过程。该文档还包含一个附录，其中包含组织和治理最佳实践、模板、对关键SOA标准的评论以及建议的到详细信息的链接。

# 2 服务生命周期简介

## 2.1 简介

根据SOA参考架构建立架构基线后，专业人员就可以审查服务生命周期。本节简要描述服务生命周期，并标识与其生命周期的每个阶段关联的操作人员、潜在工具和工件。本文档不会介绍要想成功地使用SOA所需的全部文化、治理和组织更改；相反，本文关注为服务生命周期定义最佳实践。服务生命周期属于SOA生命周期中的执行阶段，如下图所示。

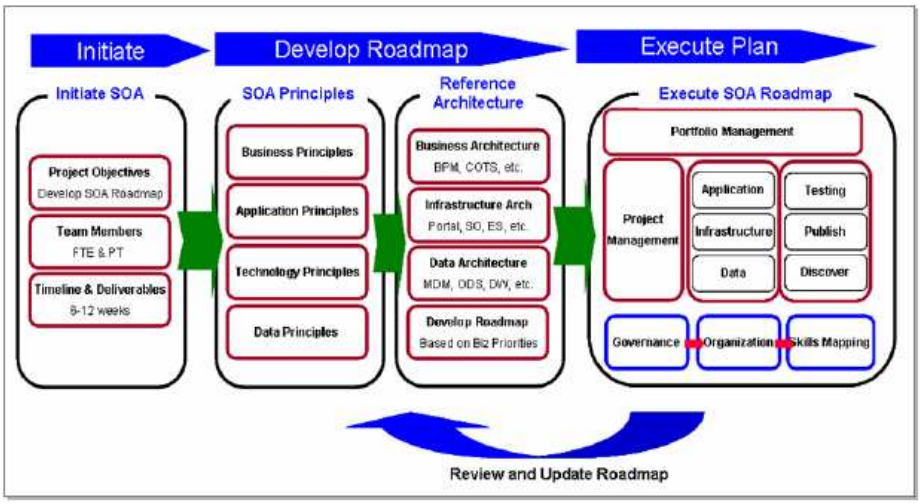



*图1：SOA生命周期*

## 2.2 定义

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

允许跨三个阶段的服务治理：要求和分析、设计和开发，以及IT操作。

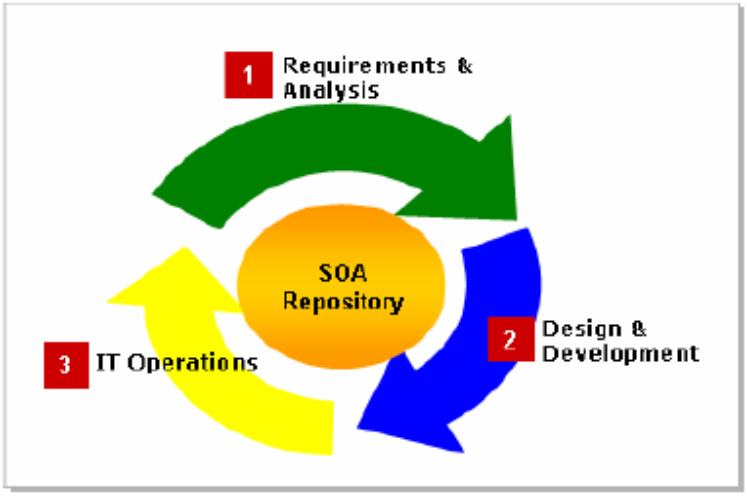



*图2：服务生命周期的三个阶段*

上图演示了服务生命周期的三个阶段，并说明了要启用服务治理需要企业服务储存库。

- **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：
  - ✧ 从级别0开始向下可视化映射业务流程
  - ✧ 定义每个业务流程
  - ✧ 标识每个流程的业务所有者
  - ✧ 标识目标和当前业务服务的差异
  - ✧ 映射输入和输出数据元素
  - ✧ 确定业务流程和业务服务的优先级
  - ✧ 捕获业务服务定义的所有方面
  - ✧ 模拟用户界面和/或业务流程。
- **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：
  - ✧ 审查需求并标识每个业务流程的备选方案
  - ✧ 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
  - ✧ 标识业务服务的重用机会
  - ✧ 开发和执行详细的项目计划
  - ✧ 跟踪和报告业务和IT管理的进展
  - ✧ 在提交每个业务服务时获得业务的签收。
- **IT操作团队：**此团队负责测试环境、分级(staging)环境和生产环境，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：

- ✧ 审查要求并标识基础架构需要
- ✧ 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
- ✧ 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
- ✧ 跟踪和管理服务和资产之间的依赖关系
- ✧ 部署和管理生产环境中的业务服务
- ✧ 根据业务优先级为业务服务提供应用程序支持

每个阶段的详细信息在本节的后面部分提供。下面是一些用于为业务提供复合应用程序的高级IT流程。

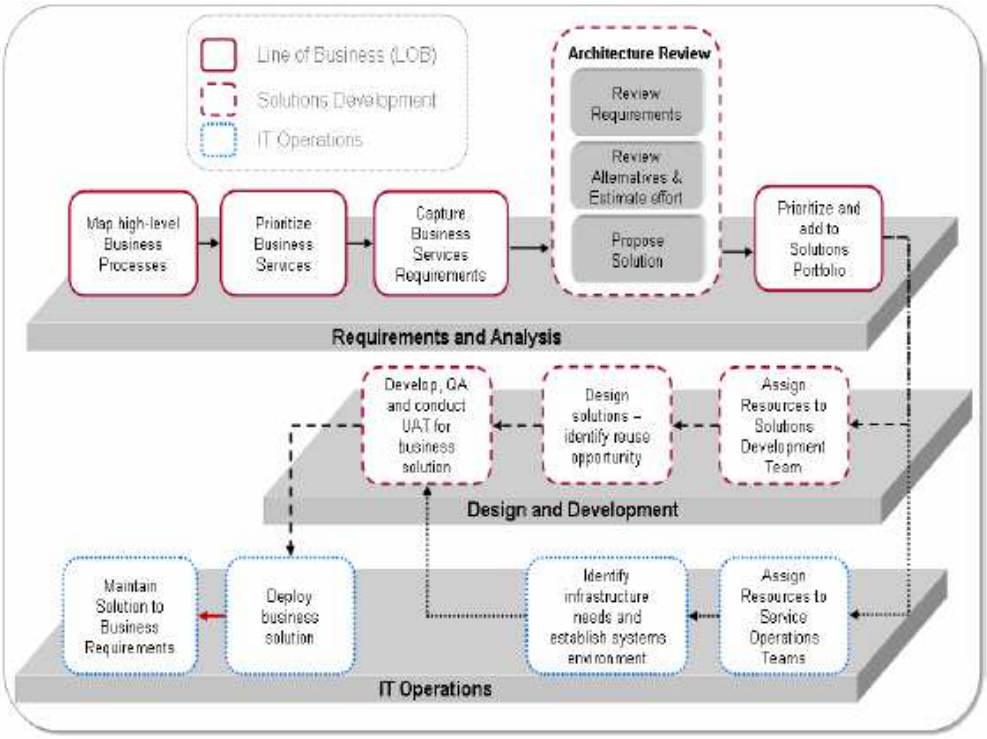



*图3：提供业务复合应用程序的IT流程*

此流程还标识三个组织中的每一个在提供业务应用程序方面的角色。

## 2.3 服务生命周期治理

治理是要实现SOA必须提供的一组流程、工具和组织结构。只有当组织在整个服务生命周期中坚持遵循标准并且遵循正确的过程时才能有效重用服务。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silo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。SOA治理功能的主要责任是：

- 发布SOA标准和最佳实践
- 定义和执行流程以便在项目级别改进服务的使用和重用

- 管理企业或LOB的所有共享服务
- 跨组织传播标准和最佳实践
- 在组织内宣传SOA的成绩

服务治理是整个服务生命周期的基础。

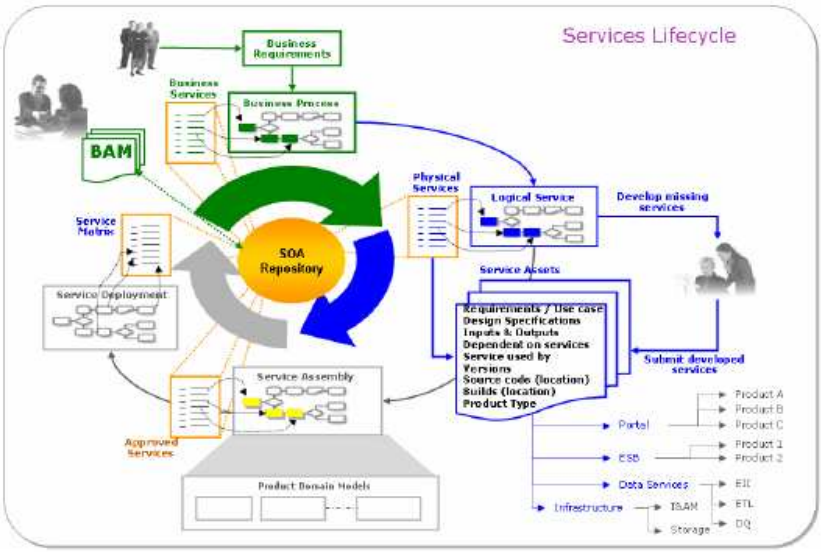



*图4：服务生命周期治理*

上图演示高级别的服务生命周期，并观察下面的阶段。

## 2.3.1 需求和分析

业务分析师需要分析业务以捕获业务需求，以业务流程的形式更可取。对于最初的SOA项目，团队通常关注非企业级或LOB级业务流程，但仅限于领导团队在批准项目资金时标识的范围。团队捕获正在交付的复合应用程序的业务逻辑。

一旦团队捕获了业务流程，业务分析师就可以标识跨企业或LOB的所有流程副本。业务分析师为团队可以重用的业务流程搜索SOA储存库。一旦完成了此阶段，业务分析师就可以将工件上传到SOA储存库，来依次触发治理流程。

该治理流程应该特定于该组织和项目。在得到所有批准（尤其是业务所有者的批准）前，团队不应认为此阶段已经完成。

## 2.3.2 设计

业务分析师应将需求和业务流程传递给架构师来设计应用程序。每个IT组织通常都具有自己的应用程序设计方法和框架。

在此阶段，架构师标识服务及其实现。然后架构师会搜索SOA储存库以进行再次使用。架构师不需要将搜索限制于生产环境中当前部署的服务；搜索应该扩展到其他团队目前正在开发的服务。

在此流程的最后，架构师可能已经标识了产品中已经部署的可以进行重用的服务、需要进行修改以创建新版本的服务、需要进行开发的服务以及需要退出的服务。

架构师将所有设计工件上传到SOA储存库，触发得到了企业架构审查委员会、项目经理和操作团队的批准的治理流程。项目经理还应该使用这些信息分配服务开发任务。

### 2.3.3 服务开发

架构师最好是从SOA储存库将设计详细信息发送给开发团队。开发团队可能在地理上分散分布，每个团队都可能有业务或产品领域的专家。

开发团队以迭代方式开发和测试复合应用程序，并将工件上传到企业服务储存库。当开发团队指示准备好部署服务时，他们就会触发治理流程。

### 2.3.4 IT 操作团队

此团队通常负责提供开发环境、QA环境、分级（staging）环境和生产环境。当服务开发组织从架构师处接收到设计详细信息时，IT操作团队就会建立开发环境。IT操作团队通常还要管理QA环境，因为此环境与生产环境是相同的。

开发团队通常为操作团队提供基础。对于由服务组成的复合应用程序，开发团队为IT操作团队提供服务装配。建议的最佳实践是根据SOA储存库中的信息装配服务。

一旦IT操作团队装配了服务，团队就会将这些服务部署到目标节点。业务分析师和架构师应该在初期阶段定义了业务、安全和管理策略。现在IT操作团队需要负责监控并提供业务指标以跟踪业务KPI以及审查IT-SLA。建议的最佳实践是让IT操作团队将产品实例（包括硬件、节点名称、产品版本和应用程序版本）映射回SOA储存库。

### 2.3.5 业务仪表板

业务可能想要查看各种不同类型的信息，包括通过监控系统、操作数据存储和BPM工具得到的数据。这些信息分为如下几个类别：

- IT-SLA
- 业务活动监控
- 策略管理
- 服务成熟度模型（监控服务生命周期的指标，以及企业服务储存库的一个可搜索属性）。

# 3 服务生命周期阶段

本节标识参与人员、使用的工具、可交付对象、生命周期阶段建议、生命周期阶段流程、最终用户工具描述以及服务生命周期的各个阶段的最佳实践。

## 3.1 需求和分析

### 3.1.1 参与人员

- 业务工作人员（通常是来自LOB的业务操作团队）
- 项目经理（业务和IT）
- 业务分析师
- 架构师（可选）

### 3.1.2 使用的工具

- 业务需求工具包括办公室、业务流程建模工具和需求捕获工具
- 业务流程建模工具包括BPMN、Visio和Pro*Activity
- 业务规则工具包括产品规则引擎和Word
- 用户界面工具包括门户模拟工具、Macromedia、Visual Studio、Eclipse，以及JSP和HTML编辑器

### 3.1.3 工件（可交付对象）

- 设计模型，如UML、BPM（业务流程模型），数据流模型
- 绑定，例如JMS、RMI、IIOP和HTTP(s)

#### 3.1.3.1 工件描述

每个LOB都定义了自己的业务流程，业务流程是在服务生命周期的这个阶段捕获的。有些LOB可能有类似的流程或者稍有差异的子流程。例如，消费贷款部和抵押贷款部可以是在大型企业中具有公共业务流程的两个单独的LOB。这两个LOB都可以通过记录并共享他们的业务流程和关键知识中受益。

LOB可能还可以运行模拟以优化业务流程，也可以使用监控和管理系统捕获和比较实际的和模拟的结果。业务活动监控器提供了比较业务结果与已制定目标的仪表板。

较低级别的业务流程定义通常可用于开发复合应用程序。在这个级别，服务要被标识出来并映射到每个业务服务（活动）。SOA储存库是所有这些服务定义和依赖关系以及外部消耗的记录系统（均为外部使用），并帮助IT操作团队部署、监控和管理服务。

## 3.1.4 服务生命周期阶段的关键考虑事项

下面是在服务生命周期需求和分析阶段业务应该关注的一些关键考虑事项。

### 3.1.4.1 业务动机

记录业务动机可帮助将业务流程映射到服务。这允许业务和IT就如何开发服务组合以及相关资金计划进行富有成效的对话。这种映射有助于为服务开发的资金筹集制作业务案例，因为这可以帮助业务了解服务如何从中获益。

### 3.1.4.2 业务生命周期的差异

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。

## 3.1.5 服务生命周期阶段的建议的流程

下面是启动项目的一些建议的模板业务。

### 3.1.5.1 项目初始化请求

业务使用此模板为项目提交请求。在这个阶段，项目的业务发起人评估项目是否可行，并联系LOB-IT或PMO以帮助完成此工作。

### 3.1.5.2 工作的架构声明

一旦业务将PIR提交到LOB-IT或PMO，IT领导团队就会让业务参与进去，评估其是否满足所有初始条件。然后IT组织可以建立一个包括项目经理、业务分析师和架构师的项目团队来处理业务以估算需要的工作量。根据业务优先级情况，团队可能专职进行估算，也可能不是这样。

## 3.1.6 最佳实践和需求

只要项目资金到位，并且组织好核心团队，此阶段就会开始。下面是服务生命周期此阶段的高级别最佳实践和要求。

### 3.1.6.1 组合管理

业务和IT共同利用组合管理工具管理SOA项目的组合。这些工具允许业务用户提交自己的项目请求。工具将请求映射到IT治理模型，并帮助IT管理应用程序、集成、数据中心和网络。根据IT组织的特定工具和目标，工具还可用于资源计划、技能映射和监控项目资金。

组合管理软件可通过将业务战略转换为高级别业务优先级计划帮助IT定义与业务紧密对应的路线图。然后IT组织就可以执行此计划，同时进行监控以确保不存在任何偏差。此类工具支持的功能包括：

- 将所有项目映射到原始业务案例以帮助确保与业务目标对应。
- 留意单个储存库（不同于SOA储存库）中的所有这些资产，并能够突出重复的业务逻辑、依赖关系和资源约束
- 映射所有依赖关系，并能够在更改违反了业务规则时发送警报
- 管理所有更改请求和错误跟踪，包括对其他项目的影响
- 将任务和项目导出到第三方工具，例如企业服务储存库、项目管理和开发工具。

通常只有大型IT组织才要求组合管理工具，因为许可和管理开销成本与这样的工具有关。此工具的单个实例可用于整个企业或特定LOB，除非法规要求IT组织部署此类工具的多个实例。利用此类工具的最大挑战在于鼓励采用；IT领导团队需要对此类工具的使用进行强制执行。

所有IT组织的首要任务就是为管理项目请求记录并传达流程和生命周期。此任务通常是程序管理办公室（PMO）或应用程序开发团队的责任。

启动此项目通常要完成两个步骤。业务管理者向IT提交一个请求以启动一个项目，然后IT与技术团队协作估算项目所需的工作量。

根据估计，IT董事会可能批准或拒绝该项目。如果批准，则会在项目经理的领导下组织一个团队，以便为业务提供应用程序。最佳实践是使用此工具映射整个应用程序生命周期管理。

### 3.1.6.2 需求捕获

这些工具可帮助捕获所有业务需求，确定其优先顺序，并跟踪相关的开发工作。下面是这些工具应该提供的一些功能：

- 为给定项目创建业务需求的储存库
- 为所有项目提供单个储存库
- 提供根据成本、资源和收益确定需求的优先级的功能
- 标识和跟踪跨项目的依赖关系
- 实施假设分析方案并提供影响分析报告
- 将所有者分配给需求和任务并跟踪状态。

业务分析师捕获这些信息时使用的是需求捕获工具还是类似Microsoft Office的产品并不重要。重要的是需求是使用业务术语捕获的，而不使用技术词汇，SQL语句，或者打包的应用程序参考文献。 此外，项目团队可根据业务功能对需求加以分类，并在不同的阶段批准这些需求。

### 3.1.6.3 用户体验模拟

有了基于文本的应用程序规范，业务人员在应用程序完成之前不需要查看和影响它。在这个阶段进行更改会导致成本超出预期，面市时间延迟，并可能导致业务和IT之间的劣质通信。原型（静态屏幕图像或低保真度的编码线框）很难解决这样的问题，因为业务人员和最终用户必须“填平鸿沟”才能提供应用程序的全部功能。原型通常要挤走本就十分珍贵的资源，从而导致成本增加和面市时间延迟。

新工具可模拟所有业务应用程序的用户界面。这些工具提供了一个简单的拖放范例，可在开发前装

配丰富的、高保真度的应用程序模拟，包括业务逻辑和数据交互。这种模拟提供了最终产品的真实“试验驾驶”，使业务人员和最终用户很容易理解。输出用于确定构建什么，排除混淆，削减成本和提高用户采用率的可视化蓝图。最好的是，构建模拟时不需要任何IT资源，而是可以由业务分析师、用户体验设计人员、项目经理和架构师快速完成。

此类模拟工具可提供以下功能：

- 高保真度或低保真度的模拟屏幕的拖放组合
- 将数据和业务逻辑链接到模拟中的能力
- 基于团队的协作定义环境
- 在模拟环境中对需求的捕获和记录
- 对用户案例方案和工作流的内部支持
- 可重用的定义资产
- 将模拟封装到可通过电子邮件发送的文档中的能力。

人们只会使用易于使用的业务应用程序——至少应该比用户以前使用的应用程序更易使用。但团队一般不会让最终用户测试应用程序，直到已经准备好进行部署。如果用户需要变更，就会导致代价昂贵的后果。一种新的工具通过在开发前让可用性专家快速创建并与用户迭代地快速测试高保真度模拟解决这个问题。

对于想要将可用性测试流程移动到软件开发生命周期（SDLC）流程前面的开发团队，目前的最佳实践就是模拟。通过在定义应用程序的流程早期与业务分析师和开发人员紧密协作，团队可快速廉价地修复许多与低可用性相关的常见问题。集中的用户体验团队可提供以下优点：

- 有关可用性和用户测试的标准化最佳实践
- 设计和开发团队中其他人员的教育背景、培训情况和指导关系
- 创建和维护反映公司的最佳实践的可重用模拟（定义）资产
- SDLC中的早期反馈和指导
- 流程指南，样式指南。

#### 3.1.6.4 业务流程建模

BPM工具用于在要求阶段捕获业务流程。该工具应该提供下面的功能：

- **业务流程建模：**业务用户的功能是方便地建模流程、定义业务规则和关键性能指标（KPI），以及模拟、测试和开发端到端过程流。
- **业务活动监控：**实时和历史分析以及报告功能。实时流程监控、升级和管理可帮助快速标识业务流程中的任何问题并帮助快速解决问题。
- **业务流程执行：**业务自动组件的执行。这包括编排所有资源（人员、组织、应用程序和系统）以确保完美的执行和异常管理。

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。业务流程建模通常从级别0开始，此后会根据需要对流程进一步定义以用于较低的级别。可以对业务流程建模，以便达到以下目标之一：

- 在致力于执行新流程的资源前，通过多个方案模拟新流程。例如，业务线要关注业务优化，因为它正进入新的市场，推出新产品，或开始市场营销战役。

- 使IT与业务更紧密地对应。在此情况下，业务流程建模工具帮助捕获并共享业务流程（可能甚至是屏幕流）以帮助跨多个团队和地区构建讨论。

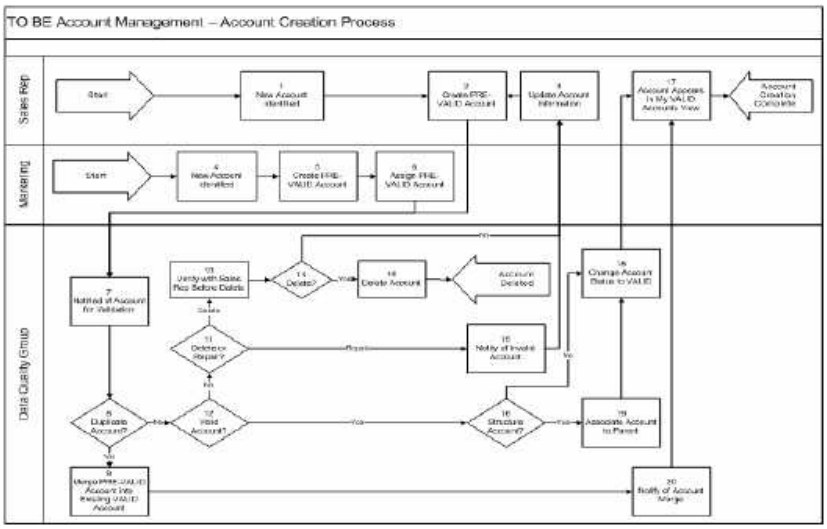



*图5：业务流程建模*

上图演示了需求阶段建模的一个典型的业务流程。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企业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。在此流程中，BPM工具可帮助他们：

- 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
- 模拟端到端业务流程以标识瓶颈并改进整体流程
- 全局审查业务流程并邀请其他LOB参与这些讨论
- 捕获本地和地区需求
- 捕获手动和自动化流程。

### 3.1.6.5 SOA 储存库

这是一个允许跨所有产品自动化治理流程的储存库。它可以存储定义业务流程、需求和模拟参数的所有相关产品元数据。此外，企业架构师还可以向此储存库上传包含模式和架构框架的企业标准文档。SOA储存库还应允许IT和业务修改业务流程，以使二者的内部治理匹配。

## 3.2 复合应用程序设计

### 3.2.1 参与人员

- 项目经理（IT）
- 业务分析师
- 企业架构师
- 项目架构师
- 设计人员
- 技术领导或开发人员领导

### 3.2.2 使用的工具

- 设计：Rational、Together Architecture、Eclipse及其他

### 3.2.3 工件（可交付对象）

- 设计模型：UML、SCA服务装配模型及其他
- 绑定：JMS、RMI、IIOP、HTTP(s)及其他

#### 3.2.3.1 工件描述

服务生命周期的这个阶段生成表示系统流、数据流、企业数据模型（以实体关系图（ERD）的形式表示）、应用程序设计（以UML表示）、活动图和序列图的模型。在这个阶段，团队还生成高级别部署模型，标识服务器、操作系统、中间件、数据库、防火墙和负载平衡器。

应用程序设计人员可能是架构师、技术领导或开发人员领导，他们可能会决定使用市场中的一些工具。此工具通常基于RUP或各种RUP，可能包含活动图、用例、类图、ERD和部署模型。架构师可以与团队共享这些工件以进行审查和批准，并可为开发团队提供指导。

## 3.2.4 服务生命周期阶段的关键考虑事项

### 3.2.4.1 企业架构框架

IT组织应该将定义架构标准、开发流程、设计模式和工具的架构框架标准化。大多数IT组织已经采用标准应用程序生命周期管理流程中的一个，并对其进行修改以满足自己的需要。此外还存在许多其他的著名架构框架，例如Zachman、联邦企业架构（FEA）和开放组织架构框架（TOGAF）。IT必须采用一种企业级架构框架才能成功实现SOA。

### 3.2.4.2 服务分类框架

服务分类框架可帮助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：

- SOA参考架构，包括共享数据服务、业务流程和门户服务
- 服务组合，包括询价到现金服务和抵押批准服务
- LOB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。

### 3.2.4.3 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，对齐服务与现有EJB可能“不是”最优的解决方案。依赖多个请求和响应对的细粒度架构（“饶舌”协议）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（而且必需保持）新协议。架构师必须仔细地进行计划，以避免在以Web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客户的粒度级别进行设计。

### 3.2.4.4 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。

## 3.2.5 服务生命周期阶段的建议的流程

此阶段在项目经理或业务分析师将要求移交给技术团队后立即开始。然后该团队开始复合应用程序的设计，通常包括以下步骤：

1. 架构师从SCM/SOA储存库下载需求，审查这些需求，并在必要时将其提交给业务分析师以获得进一步说明
2. 架构师选择工具以对应用程序进行建模和设计
3. 架构师标识服务，还可能搜索SOA储存库以标识潜在的重用
4. 架构师定义服务、实现、绑定和依赖关系
5. 一旦设计完成，架构师将所有设计工件上传到SOA储存库以获得批准
6. 架构师审查委员会（由企业架构、项目架构师、业务分析师和项目经理组成）审查设计，以确保架构能够满足业务要求，并且使其在整个企业内保持一致。

SOA储存库应该提供根据企业架构师定义的模式为每个服务类别生成服务模板的功能。

## 3.2.6 最佳实践和要求

一旦复合应用程序的需求得到批准，此阶段就会开始。本节描述服务生命周期的这个阶段的高级别最佳实践和需求。

### 3.2.6.1 服务编排（建模）

在服务生命周期的需求和分析阶段，团队应该以业务流程形式捕获所有要求，包括业务规则和策略。这允许业务流程和服务级别的重用。大多数BPM工具都提供了两种分离工具：一个由业务分析师用来捕获需求的BPM工具，以及一个用于架构师的服务编排建模工具。服务编排建模工具通常包括这个BPM工具，并允许架构师为每个业务服务或活动开发服务编排或流程模型。这些工具中的大多数都生成代码或创建元数据以执行逻辑。

服务编排建模的最佳实践包括：

- 利用BPM工具进行服务编排建模
- 采用用于建模（如BPMN）和编排（BPEL）的基于开放标准的工具
- 让业务分析师关注业务流程建模，让架构师关注服务编排建模
- 让架构师开发用于复合应用程序的服务编排模型，即便业务分析师尚未使用BPM工具定义业务要求
- 将所有服务编排模型上传到SOA储存库。

### 3.2.6.2 服务组合

大多数领先的软件供应商都遵循一个名为服务组合架构（SCA）的新标准。SCA标准定义服务、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方面。此外，开源项目正在开发用于SCA的Java和C++运行时，例如Apache的Tuscany项目。

SCA标准进行了如下定义：

- 如何通过字节码增强、依赖注入和面向方面编程的机制（AOP）自动生成基于元数据的代码
- 扩展注释的机制（控件框架）
- 用于Java“简单传统Java对象（POJO）”、业务流程执行语言（BPEL）、Web服务，以及.NET或J2EE组件的注释
- 服务的配置参数和部署选项
- 测量和监控服务的机制列表。

有关这个建议的标准的附加信息可在http://www.osoa.org中找到

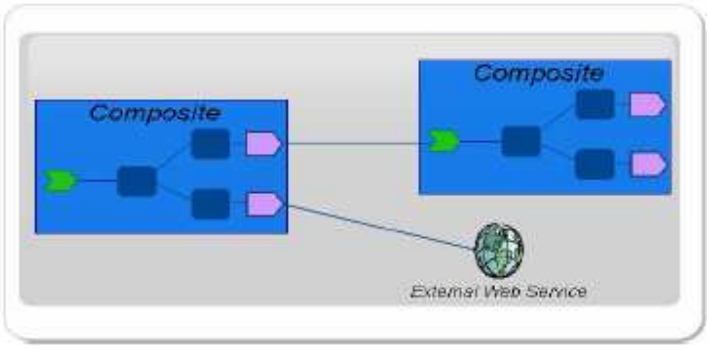



*图6：服务组合架构*

上图演示了高级别服务装配模型，其中应用程序由多个服务组合组成。服务组合由组件服务组成。

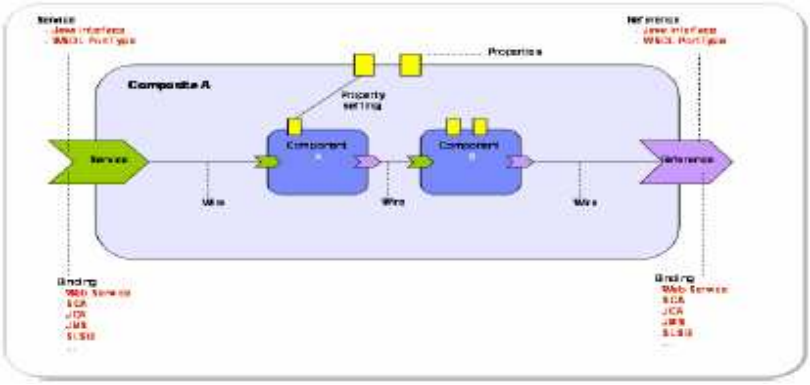



*图7：服务组合架构*

架构师可以使用多种工具组合服务。架构师需要下面的功能：

- 将每个服务映射到业务活动。这称为服务编排，是服务流程的最低的级别。
- 基于SCA组合服务。架构师根据SCA定义服务、实现、属性、接口和绑定。然后开发团队就可以利用此服务模型开发和修改服务。
- 标识哪些业务规则应该具体化并嵌入到代码中以获得更好的性能。这仅仅适合相对静态的规则；规则必须经常更改以便满足业务的需要。规则引擎应该根据规则参数精确地解释规则。
- 定义与每个服务关联的业务、管理和安全策略
- 标识和定义启用业务组审查其KPI的测量指标。

### 3.2.6.3 SOA 储存库

SOA储存库是所有企业元数据、服务定义和依赖关系的记录系统。架构师/设计人员使用的工具生成的所有元数据都要上传到SOA储存库。它提供消费者要求的信息以及IT操作团队管理服务所需的附加详细信息。储存库不仅定义从支持的角度来看监控服务时需要捕获的指标，而且提供供应商提供的技术服务。这些技术服务可为业务用户提供触发如服务警报等事件的仪表板。

团队还可以使用SOA储存库进行服务治理，以便支持将设计的工件提交给合适的审查人员进行批准的流程。要得到所有批准后项目经理才会将任务分配给开发人员和IT操作团队。

目前不存在用于储存库的标准，但多个供应商都具有能够管理SOA元数据的储存库。存在多个可用于完整企业元数据管理的“集中”储存库，但有些供应商还是只关注存储元数据的子集的SOA储存库，例如SCA模型、BPEL、XPDL、JPD和WSRP。

下面是SOA储存库的一些最佳实践：

- 搜索SOA储存库的潜在重用，包括业务流程、策略和服务
- 在标识服务进行重用前先审查SLA和服务详细信息
- 将所有架构框架文档、企业和应用程序模型以及标准上传到SOA储存库
- 定义SOA储存库工作流，以便跨企业实施设计时和运行时治理
- 利用服务注册表（UDDI）将SOA储存库与企业中的其他储存库集成起来。

### 3.2.6.4 服务注册表

服务注册表基于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。服务注册表现在已经扩展为可以充当储存库，虽然它们自身并不存储资产而更被运行时元数据所关注，后者是SOA储存库中元数据的子集。

服务注册表的最佳实践是搜索其中的服务的潜在重用，并在标识进行重用的服务前先审查SLA和服务详细信息。

### 3.2.6.5 信息建模

存在许多与信息建模相关的建模类型

- 参考数据建模
- 主要数据管理（如CDI和PIM）
- 共享数据服务
- 在合适的情况下使用域标准（用于基于标准的服务负载的HR-XML、XBRL、ACORD、MDDL和RIXML）。

IT组织很少建模参考数据，常见的数据有国家或地区名称、州名和地区代码。不过，参考数据的部分标准化可能要求跨企业提供一致的用户体验。

下面是一些用于信息建模的最佳实践

- 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
- 在捕获业务要求的同时标识数据流

- 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
- 为每个自定义应用程序开发ERD
- 在任何时候修改打包的或自定义应用程序时更新ERD。

有关主要数据管理和共享数据服务的信息，请参见《SOA专业人员指南》的第2部分《*SOA*参考架构》。

#### 3.2.6.6 应用程序建模

大多数团队都使用基于UML的建模工具，以及允许架构师建模序列图、活动图、数据流、系统流和网络图的工具。

打包的应用程序供应商提供了设计和建模其产品的功能，并且可能建议IT组织采用与这些应用程序关联的记录的最佳实践。

## 3.3 服务开发

### 3.3.1 参与人员

- 项目经理
- 架构师
- 开发团队
- 版本管理团队
- IT操作团队

### 3.3.2 使用的工具

- IDE（例如Visual Studio、Eclipse、JBuilder、JDeveloper和BEA Workshop）
- 打包的应用程序开发工具
- 回归和性能测试工具
- 构建工具（例如Maven、Cruise Control、ANT和shell脚本）
- 源控制管理系统

### 3.3.3 工件（可交付对象）

- 源代码

- 用于配置和服务执行的产品特定的元数据
- Java文档
- 版本信息

#### 3.3.3.1 工件描述

此阶段生成的工件包括服务定义、配置、合同和应用程序配置，以及服务、应用程序、部门和企业级别的管理、安全和业务策略。应用程序配置还包括实现配置。

## 3.3.4 服务生命周期阶段的关键考虑事项

服务生命周期的此阶段的任务包括：

- 定义有关在何时外包服务开发的条件
- 在IT使用新技术前执行技术审查流程
- 在服务开发前评估整个服务生命周期所需的技术
- 将每个服务映射回业务需求
- 评估现有服务以消除重复开发。

## 3.3.5 服务生命周期阶段的建议的流程

开发人员可在项目经理或架构师提供了设计文档后立即开始开发服务。这些文档可能是基于企业架构标准的办公文档或模型。

开发人员也可以在业务或IT操作团队打开变更请求（CR）后开始开发或修改服务。在这种情况下可能不需要架构师。应用程序支持技术领导或开发人员将审查变更请求并进行变更。根据更改的级别和IT PMO流程，当开发人员将元数据变更上传到SOA储存库后就会触发治理。

服务创建通常遵循下面的流程：

1. 应用程序设计或变更请求得到批准可以进行开发。
2. 对于新的开发，设计人员可能已经创建了服务模板并将其分配给了开发人员。开发人员下载服务模板并完成模板。
3. 对于不存在模板的新开发，开发人员可以使用开发工具创建服务，或者在SOA储存库中搜索类似服务并将其作为模板复制。
4. 对于变更请求，开发人员应下载代码并进行修改。
5. 在所有这些情况下，开发人员还下载需求与设计工件进行审查，并在架构师、项目经理或业务用户进行解释后进行修改。任何时候开发人员修改需求或设计工件时都会触发服务治理流程。
6. 如果架构师根据SCA连接了服务，开发人员会利用基于SCA的工具双击组件以确认创建了该服务。
7. 如果创建的服务将用于其他团队开发的服务，则开发人员应该定义服务并开发和配置服务模拟。

服务生产商通过定义对示例请求的响应以及发布响应以供使用达到这个目的。这就允许使用服务以便在实际开发服务前彻底检验合同。

8. 如果创建的服务要使用其他团队开发的服务，则开发人员可以利用其他团队生成的服务模拟。
9. 开发人员通过多种交互开发和测试服务。回归测试通常是端到端的，因此开发人员可以使用模拟的测试环境进行单位级别的测试，也可以在不同的测试阶段之间移动服务。
10. 开发人员周期性地同步元数据与SOA储存库。SOA储存库在接受任何元数据前都需要进行验证，需要时还会触发服务治理流程。
11. 开发人员生成与SOA储存库同步的文档，例如Java文档和版本信息。

这描述了传统的服务创建和维护流程。大多数软件供应商现在还为业务用户提供使用门户创建复合应用程序的功能。业务用户可以通过松耦合的多种用户交互和交互流程以及根据服务定义动态创建格式的方式来创建复合应用程序。业务用户可直接在生成过程中执行这里列出的大多数任务。使用工具可开发、测试和部署服务，还可以将驱动这些复合应用程序的元数据与SOA储存库同步。

## 3.3.6 最佳实践和需求

这些检查列表包括SOA团队应该使用开发和测试工具中的哪些内容，以及如何利用SOA储存库。

### 3.3.6.1 开发工具

- 基于标准
- 基于标准导入/导出元数据的能力
- 向后兼容性
- 来自供应商主要变更的提前通知
- 在任何合适的情况下根据模型生成代码的能力
- 集成的开发环境
- 方便地为配置提供工具以及运行配置的功能
- 调试功能
- 开发期间提示
- 基于注释和代码的文档生成（例如Javadoc）
- 与开发和测试架构的集成，例如：Junit、Cactus和SOA框架组件
- 通过双击服务图标方便地创建、修改或审查服务
- 现有服务的独立于实现的服务组合模型。

### 3.3.6.2 测试工具

- 定义测试计划的能力
- 开发测试脚本、将其映射回测试计划以及为自动回归测试开发测试脚本的能力
- 为单位级别测试存储功能的功能

- 模拟服务的功能
- 回归和性能测试功能。

#### 3.3.6.3 SOA 储存库

- 从SOA储存库签入和签出元数据（SCM是所有源代码的记录系统）
- 在每个开发里程碑的结尾将所有元数据上传到SOA储存库
- 在各种开发阶段中利用SOA储存库获得批准
- 利用SOA储存库帮助进行重用、架构标准审查和设计模式采用。

## 3.4 IT 操作团队

### 3.4.1 参与人员

- 项目经理
- 架构师
- 版本管理团队
- 构建团队
- IT操作团队

### 3.4.2 使用的工具

使用的工具包括产品部署工具（如Maven、Cruise Control、Ant和shell脚本）和基于IDE的工具（例如Eclipse、Visual Studio以及打包的应用程序部署工具）。

### 3.4.3 工件（可交付对象）

主要的可交付对象有产品域模型、打包的应用程序的配置、构建和服务依赖关系。

#### 3.4.3.1 工件描述

此流程的主要输出是团队可用来装配每种服务的构建（如果需要），装配产品的所有元数据以及映射运行服务的物理终点的URL的产品域模型。各团队从逻辑模型（SCA）开发物理部署模型。一旦团队装配了部署服务，就需要创建网络拓扑、服务器配置、服务配置，以及服务器和服务器资产管理。此外，团队还需要创建运行时工件，例如用于今后的下行关联的监控和事件日志。

## 3.4.4 服务生命周期阶段的关键考虑事项

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。

### 3.4.4.1 服务部署

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。

### 3.4.4.2 服务管理和监控

服务管理和监控可细分为两个领域：

- **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。
- **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，以便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SOA储存库。

## 3.4.5 服务生命周期阶段的建议的流程

此阶段在开发团队准备好处理性能集成测试、用户接受程度测试或性能测试，或者准备好将应用程序部署到生产环境时开始。IT操作应该遵循以下步骤：

1. 开发团队完成服务的开发和修改，并且使用模拟功能执行单位级别的测试和集成测试。
2. 开发团队标识服务，并将这些服务装配到准备好进行测试或到版本管理团队的部署的项目中。
3. 版本管理团队使用产品工具装配用于部署的版本。这会提示用户输入所有物理终点和应用程序参数。在测试环境中，用户应该能够输入已经在QA环境中运行的测试服务的URL，或者指向模拟服务。服务准备好部署或生产后，用户就可以输入服务的URL。
4. 如果服务正在退出，团队会创建一个新的项目退出服务，然后以其为基础重新设计所有服务。有些时候，并非所有依赖服务都在SOA储存库中进行定义。下面是一个降低风险的流程。
    a. 创建一个项目以重新设计和部署根据将要退出的服务的所有服务。
    b. 使要退出的服务一直打开并运行，并监控以确认该服务未为其他服务使用。
    c. 部署项目并创建新的项目以便退出旧服务。
5. IT操作团队应使用部署工具装配部署的服务项目。
6. 一旦IT操作团队将服务部署到生产环境中，团队就可以利用各种工具监控服务、应用程序、中间件、操作系统、硬件和网络。

7. IT操作负责使服务一直打开并运行。IT操作团队将特殊应用问题导致的支持问题提交给应用程序团队。

## 3.4.6 最佳实践和需求

当开发团队准备好将应用程序部署到QA、分级（staging）或生产环境时，他们可从了解服务生命周期此阶段的这些高级别最佳实践和需求中获益。

### 3.4.6.1 版本管理工具

- 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
- 利用开源工具（如ANT、Maven或Cruise Control）进行定期构建
- 利用SOA储存库进行服务治理
- 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
- 将版本管理团队集中在LOB或企业级别
- 让版本管理团队从项目一开始即参与其中。

### 3.4.6.2 部署工具

- 利用现有部署工具和过程
- 搜索自动化应用程序服务配置和供应部署工具（如果IT组织尚未具有此功能）；企业管理系统提供商可以提供这样的工具
- 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
- 与程序管理办公室协调部署
- 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。

### 3.4.6.3 企业管理系统

企业管理系统应该提供下面的功能：

- 应用程序管理
- 网络元素、操作系统、软件和修补程序的资产管理
- 业务服务管理
- 配置管理（服务和服务器）
- 监控
- 复合事件处理
- 关联引擎
- 自动操作
- 诊断和根源分析

- 统一控制台
- 业务仪表板
- 合同管理。

有关附加详细信息，请参见《SOA参考架构》的“业务服务管理”部分。

#### 3.4.6.4 SOA 储存库

组织应该利用其SOA储存库的功能完成下面的操作：

- 管理网络拓扑
- 管理服务部署网络
- 为容量计划和网络拓扑开发审查服务依赖关系和SLA，以及CMDB
- 在每次部署结束后将所有元数据上传到SOA储存库
- 在部署和管理阶段获得批准。

## 3.5 业务仪表板

### 3.5.1 参与人员

- 执行人员
- 业务操作团队
- CIO人员
- LOB-IT执行人员
- 业务分析师
- 项目经理
- 架构师
- IT操作团队

### 3.5.2 使用的工具

- 数据集市
- 操作数据存储
- 业务智能工具
- 基于门户的仪表板

## 3.5.3 工件（可交付对象）

- 星形模式与雪花模式
- 仪表板使用模型
  - ✧ 记分卡指标定义
  - ✧ 向上或向下分析规则和业务算法
  - ✧ 用于审计和同步的数据血统和信息原点规则

### 3.5.3.1 工件描述

业务仪表板是共享的表示服务。工件捕获向上和向下分析业务规则所需的元数据。

CWM是用于捕获数据仓库和数据市场的模型的元数据建模表示法和框架。XMI是根据XML标准构建的建模交换表示法。同时，CWM和XMI都允许使用XML跨多个供应商工具交换数据仓库和数据市场模式中捕获的元数据，以便用于其他建模工具和基于EII的服务。

该元数据储存库包括用于向上分析的业务规则以及用于派生指标的业务算法，以便用于大多数BI工具的语义转换层。这些指标形成了业务记分卡的基础，此记分卡报告关键业务驱动因素或价值链的各个方面。业务仪表板将一个或多个记分卡组织在一起以便显示对业务驱动因素或价值链业务流程的影响。例如，记分卡可以报告供应商关系的改进或者支持人员生产力的提高。单个仪表板可以并入这两者以便监控客户满意度的业务驱动因素。此外，可能还存在一个有关重复销售的相关记分卡，其中会记录客户满意度的增长。

大多数业务仪表板还提供向下分析的功能。BI工具利用血统和信息来源路径提供更多有关记分卡的某个方面的详细信息。

## 3.5.4 服务生命周期阶段的关键考虑事项

构建业务仪表板要求标识放入其中的标准元数据和规则。基于仪表板的视图和记分卡会绑定到LOB和用户角色，并且使用与用户角色链接的授权规则确保授权用户可以访问正确级别的信息。安全方面、信息粒度和治理规则由SOX定义，其他法规机构也可能影响仪表板设计。

仪表板设计人员需要进行计划以确保派生记分卡和指标虚拟化以及血统信息的精确度以提供精确的向下分析功能。每个用户群可能都具有不同的向下分析需求，这些需求必须分解为可访问性和授权级别规则。设计人员还必须确保粒度行级别数据始终能够备份向上分析数据和派生的信息。这支持精确度和审计度。

## 3.5.5 服务生命周期阶段的建议的流程

通常情况下，组织在一个或多个操作和分析服务已经可用并且已经与LOB用户建立了信任时开发业务仪表板。仪表板是一个用于归总所有记分卡并与影响实时报告的指标的价值链业务流程交互的复合服务。有些向下分析可能会指向操作数据或操作数据存储。更数复杂的仪表板都会订阅实时业务事件以提供每分钟更新的信息。业务仪表板可能还链接警报和异常报告服务，例如BAM服务。

## 3.5.6 最佳实践和要求

### 3.5.6.1 业务智能

业务智能解决方案可能包括如下内容：

- 用于决策支持系统（DSS）的分析工具
- 警报引擎
- 业务事件分析
- 执行仪表板和业务记分卡。

### 3.5.6.2 门户

像JSR 168 portlet和支持WSRP的联合门户这样的技术可帮助嵌入或使用基于DSS的分析视图和记分卡。这些技术支持共享表示服务。

### 3.5.6.3 IT 服务的服务质量监控指标和服务 SLA 需要的相互关系

为了执行服务质量（QoS）的端到端监控和报告，组织需要将响应时间和可用性与服务周转时间的SLA相关的指标。换言之，服务合同规范必须考虑响应时间的服务运行时执行指标、可用性和性能指标。

大多数组织最终都要以手动方式将业务流程和业务功能服务质量指标与SLA期望和服务合同关联。应用程序服务器实现和ESB将服务质量指标报告到数据库中，以便映射回服务注册表或储存库中的SLA。如果SLA和运行时服务性能指标用XML表示，则此映射可在数据库级别或者使用XQuery完成。

要自动化此过程，组织需要使用通用语义语言和元数据指定SLA、服务质量和运行时性能指标。此外，SLA和运行时性能指标的捕获模式可能需要标准化，以便提供自动化的相互关系和映射。有了这个级别的自动化映射，业务仪表板可以显示服务监控指标和SLA的一个复合记分卡。这就允许业务评估服务运行时影响服务周转时间的方式。

# 4 附录

## 4.1 IT 涉众

对于公司内的各个人员，由于角色不同，对SOA的定义也存在差异。这样，详细说明SOA中可能涉及的IT涉众的角色就非常重要。

### 4.1.1 IT“董事会”

和所有大型公司一样，IT也需要一个“董事会”。此功能通常由关键执行人员或其代表执行。董事会的目标是设定方向，批准计划和项目以及解决冲突。有些组织将这些董事会称为信息服务筹划指导委员会(ISSC)、信息技术审查会员会（ITRB）或信息技术领导团体。

### 4.1.2 首席信息官

CIO负责IT的所有方面。有些组织设置了多个向总CIO报告的分部CIO。在有些组织内，分部CIO具有相当的自治权，但目前的趋势是将企业架构和企业共享服务团队整合至总CIO的管辖之下，以便快速采用SOA并允许在整个企业内采用一致的方法。

### 4.1.3 程序管理办公室（PMO）

PMO负责跨企业或LOB编排项目。它充当所有跨功能活动的联系人，并确保每个项目团队都遵循企业架构师定义的标准流程。

### 4.1.4 业务发起人

业务发起人在业务范围内支持应用程序，并最终负责确保成功采用项目。通常情况下，业务发起人应该是公司的高级管理人员（副总裁或更高级的人员），通常负责确保业务转化的资金及其完成。

### 4.1.5 项目团队

项目团队的任务是提供业务功能。该团队开发用于提供功能的策略，并调查可用的选项。通常情况下每个项目团队都有一个项目经理。项目管理责任通常由LOB和IT各推举一个人共同承担。

### 4.1.6 架构师

架构师可分为多个类别。

#### 4.1.6.1 企业架构师

企业架构师定义标准、流程和设计模式，并标识新的技术。

#### 4.1.6.2 项目架构师

项目架构师设计业务解决方案或应用程序。

#### 4.1.6.3 信息/数据架构师

信息或数据架构师确保在企业间处理信息和数据的一致方法和模型。

### 4.1.7 业务分析师

业务分析师捕获业务需求、策略和规则。

### 4.1.8 架构筹划指导委员会

几乎每个项目或程序都具有一个筹划指导委员会。尤其是SOA还需要负责企业架构的筹划指导委员会。此委员会的成员包括IT领导团队和来自业务操作团队的关键涉众。

### 4.1.9 IT 操作团队

IT操作团队通常负责数据中心、安全、网络和第1层支持，使它们保持正常运行状态。此团队通常不负责特殊应用的问题。

### 4.1.10 业务操作团队

业务操作团队定义和记录操作流程。该团队中的每个成员都负责业务的一个领域，例如，销售操作团队确保对所有销售请求进行审查，并在条件合适时接受请求。

### 4.1.11 首席技术官（CTO）

有些组织中，CTO负责IT中的企业架构。CTO也可充当CIO。

### 4.1.12 企业共享服务团队

企业共享服务团队为企业开发共享服务。有时候，这是专门的共享服务团队，有时候又是架构团队的一部分，或者，有些时候每个项目团队都开发自己的共享服务。

### 4.1.13 首席流程官（CPO）

CPO通常与CIO对应，他们可能向COO或公司总裁报告，负责定义跨企业业务流程。

### 4.1.14 首席安全官（CSO）

CSO负责企业安全策略和实现。CSO通常向CIO报告。

### 4.1.15 项目经理

项目经理负责在一定的预算范围内按时交付项目。项目经理可能来自LOB，也可能来自IT。标准最佳实践是让一个人负责项目交付，不过，更好的情况是公司让一个业务项目经理和一个IT项目经理共同负责交付项目。

### 4.1.16 应用程序支持团队

组织可以选择让专门的集中的团队、LOB的应用程序支持团队、有开发人员承担应用程序支持角色的永久性项目团队支持应用程序，也可以选择将应用程序支持外包。

## 4.2 SOA 治理和组织

SOA治理和组织最佳实践非常复杂。本节仅关注观察到的一些最佳实践。

### 4.2.1 SOA 开发组织

#### 4.2.1.1 简介

最佳实践建议通过定义目标状态、开发路线图以及标识需要构建的组件和构建方式开发组织模型。治理和组织最终将成为本练习的自然结果。

替代的方法是根据任务构建组织模型而不是需要构建的组件。本节简要描述一些允许较快采用SOA的组织模型。

#### 4.2.1.2 传统开发方法

大多数IT项目都属于如下类别：

- **电子商务解决方案：**用于内部和外部用户的门户应用程序
- **打包的应用程序：**最佳的点解决方案
- **集成：**跨企业或LOB集成应用程序、门户和数据
- **基础架构：**数据中心、网络、服务器和软件平台。

业务发起人和IT领导团队会定期确定项目的优先顺序和监控进度。

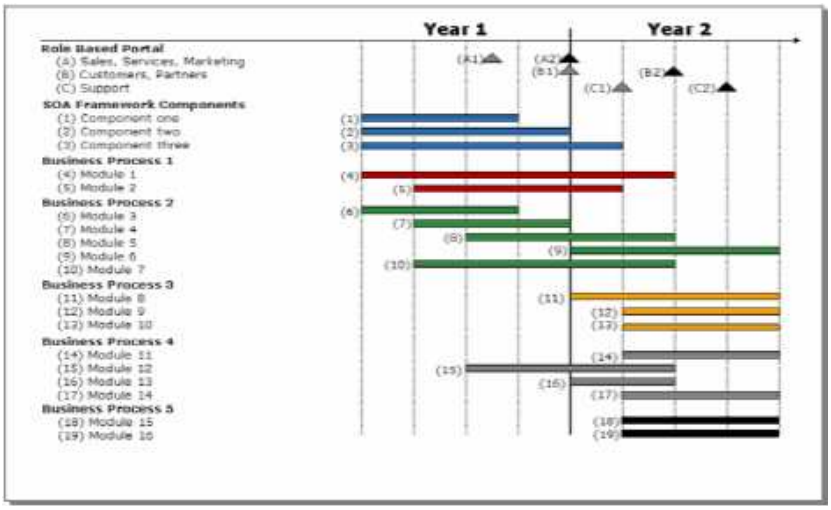



*图8：典型的IT路线图*

传统的开发生命周期可一致地应用于所有计划，如下图所示，而资源分配随计划类别的不同而不同。下图演示了一个典型的开发生命周期。

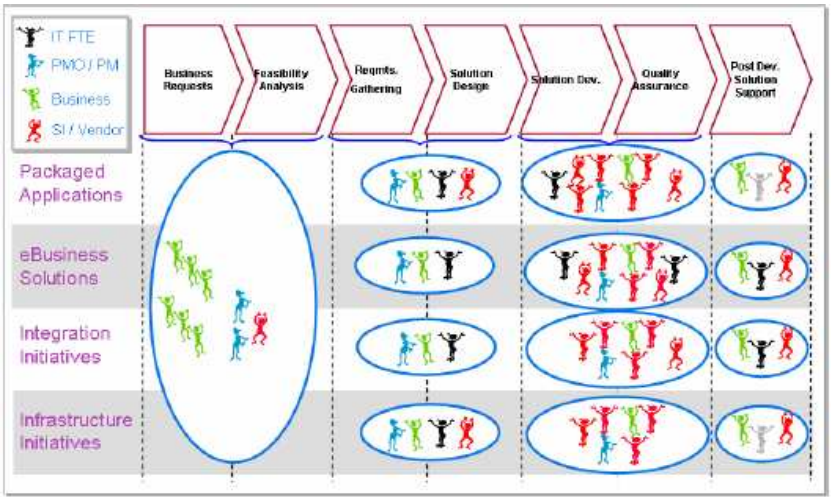



*图9：典型的IT开发生命周期*

通常情况下，由业务操作团队、PMO、项目经理以及业务分析师和架构师中的IT人员组成的团队审查每个计划的高级别业务需求。根据他们的分析，IT工作人员为联合领导团队提供建议的方法并评估资金批准。虽然各种类型的计划的生命周期都相同，但方法却各不相同。

- **打包的应用程序：**典型的方法是将开发外包给系统集成商或产品供应商，然后远程支持这些打包的应用程序，已达到降低成本的目的。这被认为是最佳实践的原因在于，打包的应用程序是专有的，在公司内部将这些技能集作为打包的应用程序开发并不能为公司带来策略价值。企业

可能会针对一个打包的应用程序的供应商进行标准化并构建内部团队，但对于大型的公司，这并不是实用的解决方案。这样公司就会向供应商的业务问题过度公开。并且，大多数LOB希望使用特定于自身的需要的打包的应用程序。

- **电子商务解决方案：**大多数企业将电子商务解决方案视为战略投资，因为这是客户、合作伙伴、提供商和员工之间进行交互和协作的增长最为迅速的渠道。首选方法是让公司内最佳的架构师和开发人员创建和支持这些解决方案，并在组织内构建这些技能集。开发内部专业技能很有意义，对于关注消费者的组织来说更是如此，因为这可以帮助快速更改面向客户的解决方案以帮助组织保持其竞争优势。对于一些较低层次的业务功能，可以将其支持外包或者通过远程方法提供。

- **集成：**电子商务解决方案与打包的应用程序的集成以及以后与BPM平台的集成非常复杂。其执行取决于各个团队之间交互的质量，因此外包不是最合适的方法。相反，行业的最佳实践是将所有自定义嵌入到中间件中。LOB应该坚持提供业务流程和规则并解决与业务定义相关的开放问题。企业架构团队应该负责进行集成决策。

- **基础架构：**基础架构的最佳方法随组织的大小而变化。组织应该自行完成基础架构，除非其收入超过10亿美元。较大企业可以将基础架构外包给第三方，这样就可以更容易地评估基础架构的真实成本。在内部完成基础架构会使IT操作人员的负担过重，导致延长周转时间并降低效率。

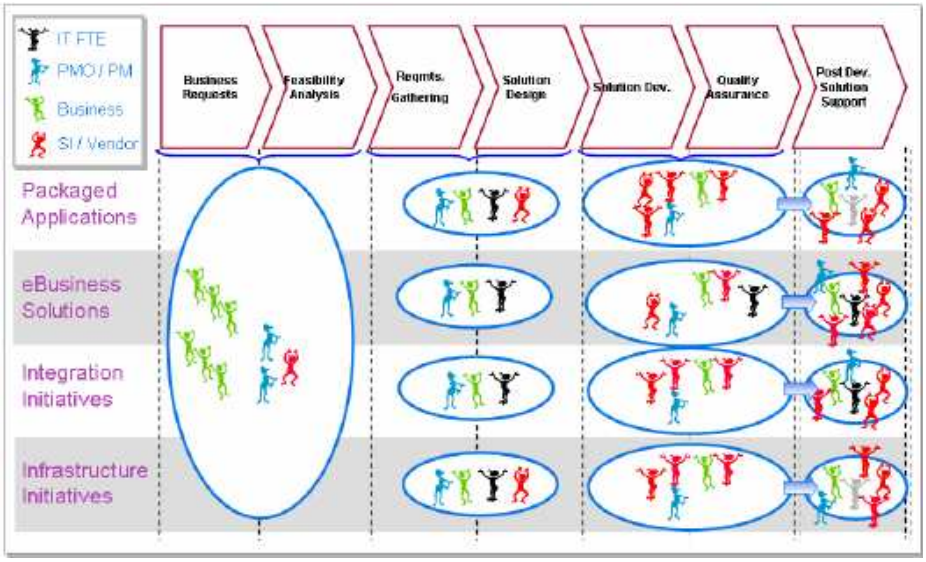



*图10：当前生命周期方法的结果*

传统的生命周期方法的主要缺点之一是，当IT提供新的功能时，资源会转而支持这些功能，导致功能的成本增加。IT可能需要引入附加资源来开发新服务或者对现有服务提供支持。IT组织无法中止这种发展倾向，但可以通过外包然后以远程方式支持功能降低新开发的成本。不过，这种方法对IT组织的作用也仅限于此。企业需要采用不同的方法为其开发组织配备人员。

### 4.2.1.3 建议的方法

建议的方法是根据技术功能创建团队，如下所示：

- **组合团队：**根据业务流程和需求将服务联系在一起。服务之间的连接不仅能够标识每个服务要求的功能，还能够标识部署、配置和管理模型。此团队应由架构师和技术领导组成。

- **UI团队：**开发由连接框架、用户交互流、导航和类型验证组成的业务交互层。此层的开发可以外包，只要拥有良好过程来捕获和记录需求。

- **服务团队：**设计和开发业务逻辑。建议是配备在公司内担任重要职务的人员，尤其是架构师和主要开发人员。开发本身可以外包，只要是基于架构方针并由内部人员管理和审查。
- **数据团队：**为其他团队开发共享的或特定的数据服务。信息和数据架构师是该团队的主要成员，也应该是公司员工。他们定义企业的数据模型、数据质量和公共对象（如客户、订单和产品）。开发本身可以在任何位置完成，但必须基于架构方针并由内部人员管理和审查。其他团队可从此团队接收服务，但他们不需要了解企业数据模型。

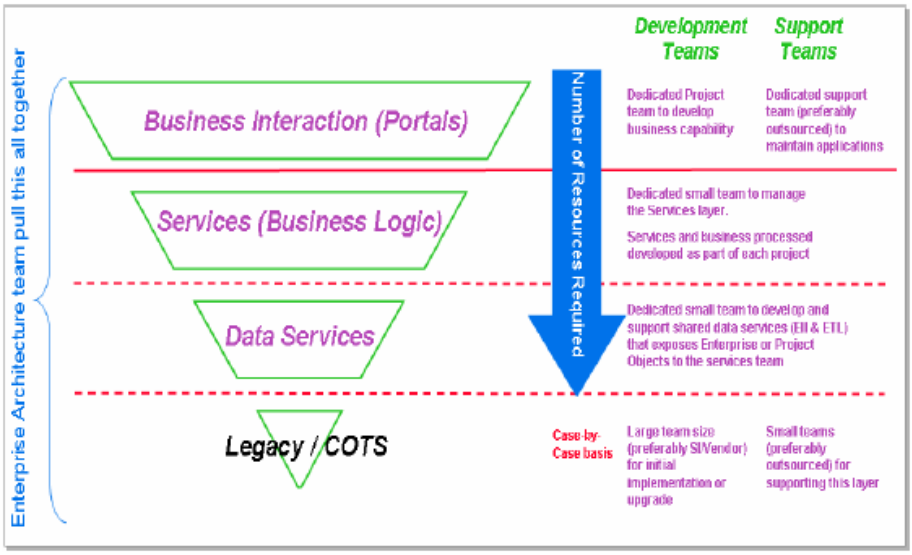



*图11：基于功能的组织*

#### 4.2.1.4 摘要

这个建议的方法根据其功能组织开发团队。这可以降低成本，因为不需要为每个项目团队配备具有类似技能集的人员。

## 4.2.2 企业架构：角色和责任

### 4.2.2.1 任务声明

企业架构的目的是开发IT策略并通过理论指导、流程、结构和价值交付提供与不断变化的业务优先级对应的技术理想。

### 4.2.2.2 企业架构责任

下面是企业架构师的一些高级别责任：

#### *4.2.2.2.1 开发IT策略和技术理想*

- 审查IT处理企业业务流程、信息、技术和应用程序的方式的目标（未来）状态，以便达到允许业务执行公司任务的阶段性业务目标
- 审查当前状态以标识差距，并开发一个可操作的实现目标状态的路线图，包括业务流程、组织、

信息、技术和应用程序

- 评估替代方式，并根据与每个替代方式关联的成本、收益和依赖关系的整体视图提供建议。

#### *4.2.2.2.2 开发架构标准*

- 开发和发布包括定义业务应用程序的服务生命周期方法，以便设计、开发、部署、支持和升级基础技术和应用程序
- 开发每个团队的角色和责任的详细检查列表，包括每个生命周期阶段的可交付对象和退出条件
- 定义（生命周期）流程（例如快速应用程序开发方法）和公共任务（如项目管理、重用、指标和测试）。

#### *4.2.2.2.3 定义技术生命周期*

- 标识支持可操作路线图的新兴技术
- 定义技术标准以及各自的使用范围
- 获取有关新兴软件的信息并管理与战略合作伙伴的关系。

#### *4.2.2.2.4 支持技能计划*

企业架构团队标识了要采用的新技术后，就必须标识要达到目标状态所需要的新技能集。所有其他团队都应该将这些作为其技能计划的一部分使用。

#### *4.2.2.2.5 信息和数据架构*

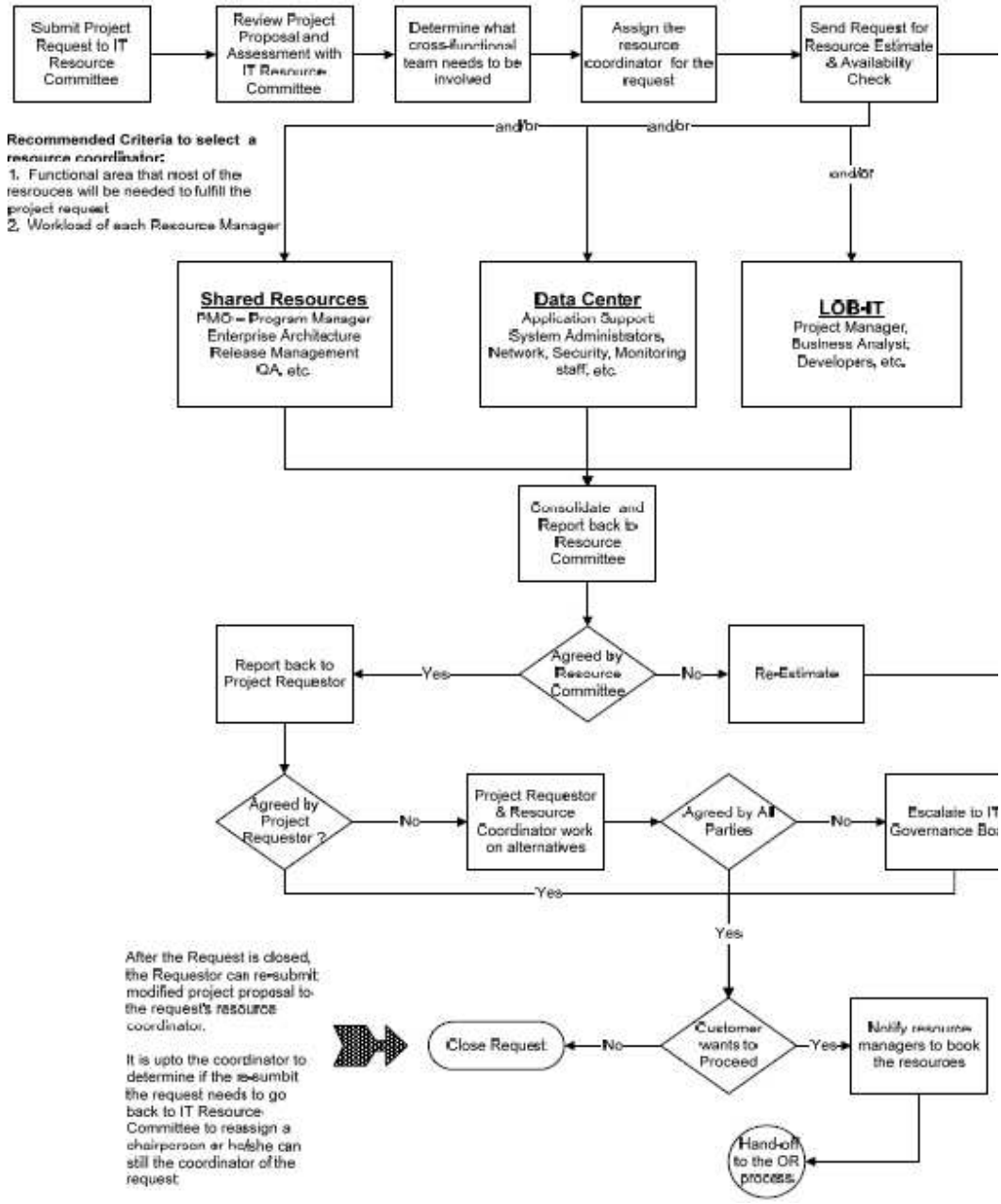
企业架构团队合作伙伴与业务紧密协作来开发一个企业数据模型，尤其是对于公共对象（例如客户、产品和订单）的模型。团队还必须为数据仓库、数据集市、共享数据服务和路线图开发模型。

#### *4.2.2.2.6 集成架构*

企业架构团队的角色类似于城市规划。他们可能不会定义每一个建筑，但会定义连接与基础架构，以及在这种连接和基础架构下的主要系统、模型、组件以及IT基础架构的主要组件之间的关系。

## 4.2.3 IT 企业资源管理流程

下面是高级别IT企业资源管理流程流的一个示例。




## 4.2.4 项目初始化请求格式

| **请求——名称：** | **请求数据：** |
|---|---|
| **项目——名称：** | **附加涉众/优点：** |
| 业务发起人名称： | 请求编号（LOB-IT使用）： |
| 业务发起人组织： | 请求状态（LOB-IT使用）： |

| 审查和批准（LOB-IT使用） | |
|---|---|
| **业务程序办公室：** | **审查数据：** |
| **LOB IT工作人员：** | **审查数据：** |
| **LOB IT审查委员会** | **审查数据：** |

**业务问题定义和合理性**

**业务问题：** *描述此项目或工具增强请求可能可以解决的当前业务问题。请包含它可以支持的业务流程的描述。*

**业务合理性：** *提供定性和定量信息以支持投资。包括受影响的组织；用户数目、生产力提高、收入增长、成本节省，等等。*

**业务目标：** *列出想要达到的目标；包括合适的位置及其与公司或组织目标的关系*

**背景信息：** *提供任何有关此请求的附加信息，包括以前为解决问题所付出的努力；涉及的人员；考虑的新的解决方案或软件；潜在流程或组织变更影响，等等。*

**LOB IT信息：** *提供此请求与IT有关的任何相关信息。*

# 4.2.5 请求架构工作

业务分析师会在从LOB捕获高级别要求后向架构师提交对架构工作的请求（或估算请求）。此请求应包含工作声明以及来自项目启动文档的信息。

## 工作声明

要求的工作的简短描述。

## 项目请求和背景

业务要提供详细描述请求的信息以及可帮助架构师估算工作量的附加背景信息。

## 工作架构声明

工作的架构声明包含对提议的阶段、时间线、里程碑和成本的估计。

## 项目请求和背景

**项目范围、收益和成本**

**范围：** *定义此项目或者此项目的各个阶段的范围。包括：*

*a. 任何业务流程变更*

*b. 支持或自动化上述变更流程的系统功能。*

**阶段1**

□

□

□

**超出范围——边界：** *标识**超出**此项目或此项目的各个阶段的范围的工作。包括：*

*a. 任何业务流程变更*

*b. 系统功能。*

**超出阶段1**

- ☐
- ☐
- ☐

## ROI（投资回报）——收益和成本

**收益：**

***直接收益***——*列出完成此项目时为BEA提供的以及成为项目ROI的可测量直接收益。主要收益通过以下形式表示：操作成本降低、收入获得、生产力提高以及客户满意度。*

- ☐
- ☐
- ☐

***间接收益***——*标识此项目将提供的其他间接或无形收益以及优势。*

- ☐
- ☐
- ☐

**成本**

***资源***——*估算要成功地实现此项目以及正在进行的系统操作和支持所需要的资源，如下表所示。*

**已有预算**

- 永久和合同总人数：
- 软件和硬件：
- IT基础架构——网络和电信：
- 总预算：

**未预算**

- 永久和合同总人数：
- 软件和硬件：
- IT基础架构——网络和电信：
- 估计的总预算：

## 风险管理

**对其他项目的依赖性：**

*列出要开始和完成此项目对其他项目的依赖关系。*

- ☐
- ☐

**重要成功因素：**

*标识从此项目开始就需要考虑的所有重要成功因素。*

- ☐
- ☐
- ☐

**接受条件**

*列出项目的接受条件*

**风险：**

*标识可能对此项目产生负面影响的所有潜在风险。*

- ☐

□

**角色和责任**

**定义业务涉众**

*列出所有业务涉众及其在项目中的角色*

□

□

**定义IT涉众**

*列出所有IT涉众及其在项目中的角色。*

- IT发起人
- 项目经理
- 架构师
- PMO
- 系统管理员/DBA
- 应用程序支持团队

## 架构方法（映射到理想）

架构描述，映射回参考架构的模型

## 高级别项目计划和调度

为使其不断运行和支持应用程序所需要的后续部署工作（业务逻辑更改）

# 4.3 简化的公共词汇

下面是公共词汇的一个简短列表，可供业务团队、业务分析师、架构师、项目经理、开发人员、QA和操作人员使用。

| 术语 | 定义 |
|---|---|
| 业务流程建模 | 建模、捕获和模拟业务流程要求 |
| 业务事件 | 调用业务流程的外部变更 |
| 业务概念 | 企业对象，例如客户或产品 |
| 业务应用程序 | 作为打包的应用程序、自定义应用程序和业务流程实现的业务应用程序逻辑 |
| 业务服务 | 业务应用程序中的每一项离散的业务活动 |
| 逻辑服务模型 | 物理服务到业务服务的一对一映射，用于促进业务服务级别而不是物理服务级别的重用 |
| 物理服务 | 按功能分类的离散型粗粒度服务 |

## 4.4 相关 SOA 标准

## 4.5 服务组件架构和服务日期对象

- 业务逻辑（嵌入在组成服务的组件中）与需要访问服务组件所需的传输和协议绑定的隔离。
- 服务组件的核心业务逻辑以及组件要与外部服务交互或者参考外部服务要使用的协议的完全隔离。
- 使用依赖注入技术的运行时服务装配
- 服务装配的配置和部署元数据的定义
- 服务实现层组件的注释驱动并自动生成的实现的容器方针
- 使用元数据入口点的客户端与服务的运行时绑定，以及使用元数据外部服务参考的组件与外部服务的绑定
- 定义服务访问和服务依赖关系的元数据配置策略，由入口点和外部服务参考定义。
- 使用SDO作为参数和返回值的文档样式信息交换
- 定义安全、事务和可靠消息策略等服务策略（用于许多互操作性功能的WS策略）的元数据
- 一致服务实现：提供的业务服务是固定的，但提供的服务可使用SCA外部参考和属性进行更改。
- 使用不同的属性值并且使用到两个分离的目标服务的不同绑定（只要服务接口相同）来对来自相同基实现的两个不同的组件的装配
- 通过将实现连接到目标服务的参考配置
- 通过设置不同属性值完成的属性配置
- 通过Web服务、JMS消息传递、EJB、数据库存储过程和EIS绑定，多种方式的运行时访问绑定来访问相同实现代码基底
- 使用相同服务接口支持不同类型实现的SCA注释，例如<implementation. java>和<implmenation.ejb>
- 有关容器提供商如何解释注释以提供对并非SCA内置的实现的实现支持的SCA定义
- 要实现BPEL，以便BPEL流程与服务和服务活动协调，同时SCA提供服务与其实现的连接的SCA
- 通过支持<implementation.bpel>注释实现的SCA容器的WS-BPEL实现
- SCA定义以及对BPEL样式的长期运行的同步事务的支持，允许非阻塞调用服务、回调、以及补偿性事务支持和可靠性功能
- 对提供业务功能的对话服务和服务序列的支持
- 在运行时使用外部服务参考绑定连接地理上分散的服务和合作伙伴服务消费的模型定义
- 用于装配所有无需更改核心实现基础即可独立配置、部署、管理和监控的复杂服务的层集定义。

### 4.5.1 SCA 扩展

- 本地接口定义的规范的接口级别扩展（Java和WSDL接口是SCA提供的基础级别的接口）
- 用于使用SMTP或会话初始化协议（SIP）访问服务组件提供绑定的绑定级别扩展

## 4.6 Java 业务集成

- 根据行业标准服务提供者接口（SPI）固有的容器和插件的概念定义集成环境
- 定义用于集成服务引擎和协议绑定组件的SPI
- 定义如何一次构建服务引擎然后使用供应商绑定组件将其绑定到多个通信协议
- 将服务引擎（业务功能如流程引擎和纵向行业转换数据包如AS/2或eb-XML或HIPPA转换）和通信协议（如SMTP、WS-I和JMS）隔离开来
- 默认情况下构建绑定组件，以便从服务引擎中部署的服务读取WSDL，或者帮助在服务消费者和服务提供商之间交换“标准化消息”
- 让服务使用不透明的“标准化消息服务”在服务引擎和JBI容器中部署的绑定组件之间传输基于文件的信息或工作负载
- 让协议特定的绑定组件与“标准化消息路由器”交互，以便与具有“正确的服务提供商”的服务通信，然后传输消息负载
- 允许专门研究创建纵向行业业务流程的供应商使用现有服务组件构建服务，然后只要其接口一直使用服务引擎SPI就使其处于相同的JBI容器中

## 4.7 传统数据移动技术

### 4.7.1 电子数据交换（EDI）

电子数据交换（EDI）是指业务数据在计算机间的交换。在EDI中，信息是根据双方一致的格式进行组织的，因此允许“提交”不需要在两端进行人员干预或密钥更新的计算机事务。在大部分情况下，EDI事务中包含的所有信息都与传统的打印文档相同。

组织已经采用了EDI以便增强有效性和提高利润。EDI的优点包括：

- 周转时间降低
- 库存管理更好
- 生产力提高
- 成本降低

- 精确度得到改进
- 业务关系得到改进
- 客户服务得到加强
- 销售提高
- 纸张使用和存储达到最小
- 现金流增加。

EDI标准由公认标准委员会（ASC）X12开发和维护。这些标准旨在跨行业和公司边界使用。标准的更改和更新通过讨论决定，可反映整个标准用户群的需要，而不是单个组织或业务部门的需要。目前已经有超过300,000个组织使用300+EDI事务集来管理业务。

*来源：http://www.x12.org*

## 4.7.2 提取、转换和加载（ETL）

ETL完成提取、转换和加载。“提取”是指从多种源移动数据。“转换”是指公司清除并重新设置为目标要求的格式。然后就会将其加载到其他数据库、数据集市或数据仓库以进行分析，或者记载到其他操作系统以支持业务流程。

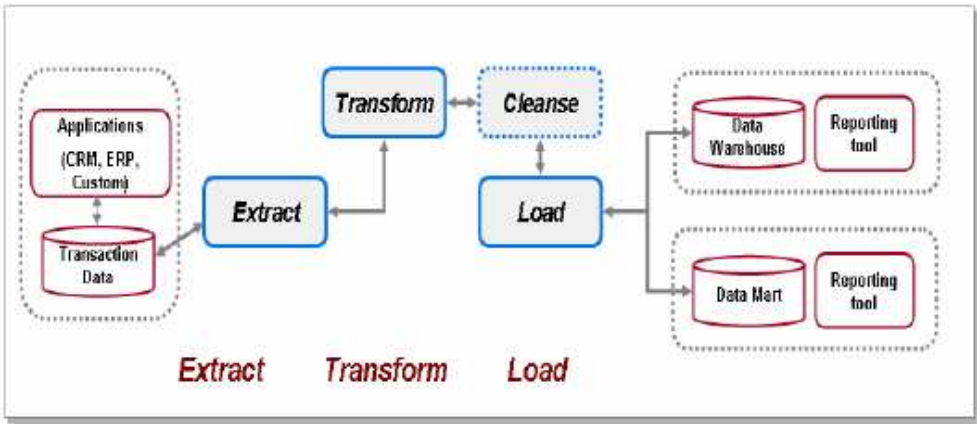



*图12：提取、转换和加载（ETL）*

虽然大多数ETL脚本都是批处理的，但是大多数ETL供应商现在都提供了作为Web服务调用这些脚本的功能。

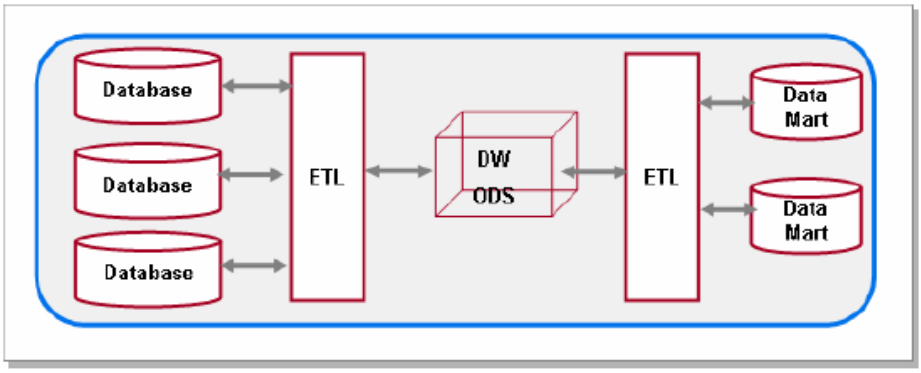


*图13：数据仓库和数据集市的DTL的传统用法*

上图演示了ETL系统的典型用法。数据要从多个源提取出来以便创建操作数据存储（ODS）或数据仓库。从下行方向看，数据集市可能利用ODS或数据仓库。这是建议的方法，不过有些时候组织可以在不首先创建ODS或数据仓库的情况下构建数据集市来节约时间和资金。

传统方法是利用EAI基础架构填充ODS或数据仓库，但越来越多的公司已经开始使用EDA填充ODS/数据仓库。他们使用ESB和EII工具而不是传统的EAI和ETL工具。这是ETL和EII工具之间的驱动聚合，以便使用EII工具提供数据操作和移动功能，以及使用ETL工具提供实时事务功能。

## 4.8 推荐的读物

下面是一些与SOA最佳实践相关的附加信息源的链接。

### 4.8.1 架构框架

联邦企业架构（FEA）

http://www.whitehouse.gov/omb/egov/a-1-fea.html

开放组织架构框架（TOGAF）

http://www.opengroup.org/togaf/

Zackman Institute for Framework Advancement

http://www.zifa.com/

### 4.8.2 关联标准

服务组件架构（SCA）

http://www.osoa.org

Web服务互操作性组织

http://www.ws-i.org

结构信息标准化促进组织（OASIS）

http://www.oasis-open.org

对象管理组（OMG）

http://www.omg.org

### 4.8.3 行业论坛

SOA协会

http://www.soainstitute.org/index.php

Financial Services Technical Conference（金融服务技术会议）

http://www.fstc.org

用于SOA的面向一致性的架构（COA）

http://www.s-ox.com/Feature/detail.cfm?articleID=1202

### 4.8.4 关注 SOA 的分析师网站

ZapThink

http://www.zapthink.com/

CBDI论坛

http://www.cbdiforum.com/

### 4.8.5 模板

捕获业务方案（用例）

http://www.ws-i.org/Requirements/SubmittingScenarios-1.0-2004-09-27.html