SOA 生命周期

SOA 生命周期

正如面向服务架构（SOA）自身一样，SOA 生命周期是一个受到广泛关注的话题；而专家对生命周期的定义各持己见，他们认为需要强调的领域也不尽相同。

服务生命周期管理是 SOA 治理向 SOA 及 SOA 服务的实际构建中的一个应用。然而，治理属于业务涉众，管理是技术人员（负责“实现”的团队）的权限。服务生命周期管理必然与 SOA 治理紧密结合，因为在软件交付的每个步骤（从业务分析人员到架构师到开发人员到测试人员，再到操作）上，确认了将要构建的内容结合了企业的明确业务需求是关键的。

SOA 质量管理是服务生命周期管理的一个方面——与交付生命周期所有阶段的规程相交叉。IBM 将 SOA 质量管理定义为这样一个过程，通过确认 SOA 生命周期中服务的功能和操作此过程能够确保服务满足业务需求。

SOA 原则

通常采用 SOA 技术的人都希望只要成功实施 SOA，那些和面向服务技术平台相关的利益就能得以实现。但是，要想真正且有效的实现 SOA 转变这个长期且具有战略地位的目标必须采用和自动化逻辑设计一致的方法。

在建立一个面向服务的方案之前，先要了解是什么令一个独立的服务适合 SOA 支持其战略目标。换句话说，应该在工程的生命周期前期提出这个问题，即确保服务真正是面向服务建立起来？

- 服务定向的原则第一部分：服务定向简介
- 服务定向原则第二部分：服务合同和松耦合
- 服务定向原则的第三部分：服务抽取性和服务重用
- 服务定向原则第四部分：服务的可发现性与可组合性
- 服务定向原则第五部分：服务独立性和无状态性
- 服务定向原则第六部分：原则的相互联系和服务层

服务生命周期

SOA 生命周期可以从企业架构和商业架构等情景理解开始，通过服务分析和建模向各方面拓展。建模不仅仅和功能性事件相关，还与安全、性能、审计等非功能性的事件相关。然后就有了开发、测试、供给、监测和改变管理。即使你的想法局限于从应用程序开发的角度来看 SOA，这也提供了前所未有的考虑方式。例如，你需要考虑采用一致性方法来建立商业服务，这样的话非功能性事件也可以与基础架构服务的运行时间保持一致。

一方面，你依然拥有传统的软件生命周期，此时服务是运行软件的接口。你必须采用以代码形式编写的软件。那就是传统的软件生命周期。但是，现在你还拥有服务生命周期，在元数据级中发生作用。在你更新服务、创建服务及重新配置服务的时候，服务生命周期又发生在不同的级别上，因为除非在软件中，否则它就不涉及新的编码过程，但我们的目标是要在元数据级中完成这些变化。

- 了解 SOA 生命周期
- SOA 生命周期:我们到底在谈论什么?

SOA 规划

面向服务的架构是一种 IT 战略，它把包含在各种企业应用中的分散的功能组织为可互操作的、基于标准的服务，而这些服务可以再被迅速组合和重用以满足业务需求。

SOA 与其说是一种技术，不如说是一种的思维方式。它是一项大胆的基础架构变革议程，表达我们如何通过技术和协同工作来实现文化变迁。它的突然普及不是大规模宣传的结果，而是对 SOA 作为一种使业务和 IT 系统更密切结合的演化的认知。这种演化是震撼的，必将为企业的成功带来深远的影响。

投资回报率（ROI）

由于 SOA 价值议题多面性的特征，为不同 SOA 项目计算 ROI 差异很大，不仅是为 SOA 实施寻求一个单一的 ROI 目标，公司也要采取同样的、反复的、复杂的方法来计算 ROI，这些公司本身就从 SOA 实施中获取 ROI 例如，每次他们把服务定义成公司服务模型的一部分，他们也可以为该服务定义一个相应的 ROI 目标。他们在这项服务上要花费多少钱呢。依照降低的集成成本、增加的资产重用和更大的业务灵活性，他们能从服务实施中获得多少直接或间接的回报呢？另外，随着某些特定的服务在公司中得以重用，将这些服务组合融入这些流程如何能为业务提供额外的 ROI 呢？

在许多情况下，SOA 实施能从服务第一天作用起就提供明确的、积极的投资回报率。但是，这更像是一个 ROI 期望，正如 SOA 实施本质上是可重复的、经常被评估并且是复合的。这样，用户不仅可以量化还能获取 SOA 实施投资的回报。

- SOA 的投资回报率
- 在 SOA 中寻找投资回报率
- Web 服务和 SOA 的 ROI（一）
- Web 服务和 SOA 的 ROI（二）
- 通过 Web 服务快速获得 ROI
- 如何使用 ROI 推销 SOA 和 Web 服务项目

服务定向的原则第一部分：服务定向简介

分六部分探讨服务定向原则，这是第一篇，著名作家 Thomas Erl shares 通过他的第二本 SOA 专著的部分节选和我们一同分享他关于服务定向设计实例的独特见解。书名是“面向服务的架构：概念，技术和设计”书中额外还附有评论。

通常采用 SOA 技术的人都希望只要成功实施 SOA，那些和面向服务技术平台相关的利益就能得以实现。但是，要想真正且有效的实现 SOA 转变这个长期且具有战略地位的目标必须采用和自动化逻辑设计一致的方法。

在建立一个面向服务的方案之前，先要了解是什么令一个独立的服务适合 SOA 支持其战略目标。换句话说，应该在工程的生命周期前期提出这个问题，即确保服务真正是面向服务建立起来？

答案取决于一个设计范例，这个设计范例把面向服务的结构模型和原来其他的模型区别开来。这个范例是面向服务的，其模拟商业自动化逻辑的方法已经成为一套为人们普遍接受的原则。我们可以在一个典型的面向服务的环境中，在应用、定位并形成原始成分（服务、说明、信息）时发现这些原则。

在这一系列的文章中，我们主要探讨如何将服务定向原则应用于构成服务的自动化逻辑。

在文章的后面，我们将讨论如何越过单个服务层面，应用作为范例的服务定向并形成能够封装整个企业领域的服务层。现在让我们先从解释什么是服务定向入手。

服务定向及其分类

服务定向影响很大，它产生影响的根源是软件工程理论，例如"separation of concerns."（相关分类）。这个理论在一个概念的基础上提出的，即把大问题分解成一系

列较小的、单个的问题。这样的话，处理大问题的逻辑也能被分解成较小并与之有关的小集合。该逻辑的每一部分都能解决一个特定的问题。

我们已经在既定的范例中实施了这个理论，比如面向对象和基于构件的方法。服务定向被认为是实现相关分离的独特方法（尤其在最近）。更具体地说，关键的服务定向原则为如何实现分离提供了一个独特的方法。通过实施这个理论，为 SOA 建立了一个基本范例。如果你研究与目前 SOA 实施相关的普遍特征，你就会发现许多这些特征都和如何将这些相关分离有关系。

服务定向的普遍原则

公共 IT 机构、供应商、以及咨询公司对于服务定向的构成持不同的看法。作为 SOA 系统公司发起的现行行业分析措施的一部分，很长一段时间里我一直在进行对服务定向前景的研究和评估工作。该项目意在识别并描述一套由所有主要 SOA 平台所支持的普遍原则。因此形成了服务定向现时世界的定义。我们这一系列文章的焦点和中心都是围绕那些被 SOA 行业普遍接受并被相应的供应商所支持的原则所展开的。

最近创立的八个原则：

- 服务共享一个正式契约
- 服务是松散耦合
- 服务提取潜在逻辑
- 服务可以组合
- 服务可以重用
- 服务是独立的
- 服务是无国籍的

• 服务是可发现的

这八个原则里，其中独立性、松散耦合、抽象性、以及对一个正式契约的需求可以被看做是形成 SOA 基础的核心原则。尽管八个原则支持或被其它原则支持，但这四个原则是另外四个得以实现的直接条件。研究这些原则间相互关系的非常有趣，这些研究还为服务定向带来的独特动力提供了一个更深层次的视角。

值得注意的是 Web 服务本身就支持这些原则的子集，这就表明了为什么 Web 服务技术平台被认为是适合于构建面向服务的方案。在文章的后面，我们将要讨论这些原则间的关系，以及其中的一些原则是怎样通过使用 Web 服务在内部实现的。

下一步如何

要想全面了解服务定向是如何影响面向服务架构的，我们需要研究该应用在构成 SOA 基础部分的含义。除去这项研究，在该系列文章的第二部分我们将关注松散耦合以及四个原则中前两个有关正式契约的使用。

(作者：Thomas Erl 译者：杨君 来源：TechTarget 中国)

服务定向原则第二部分：服务合同和松耦合

在前面的文章里，我们建立了一个服务定向设计范例。目前这些范例为我们提供了下面八个普遍原则。

- 服务共享一个正式契约
- 服务是松散耦合
- 服务提取潜在逻辑
- 服务可以重用
- 服务可以组合
- 服务是可发现的
- 服务是独立的
- 服务是无国籍的

这些原则是“普遍的”，因为它们代表了行业层面的关于服务定向的观点。该服务定向由 SOA 系统公司执行的研究所决定。在文章的第二部分，我们要通过讨论清单上的前两个原则：必须使用服务合同和创建服务间叫做“松散耦合”的关系来深入探讨这些原则。

服务合同的用途

服务被定义成使用一个或多个服务描述文档。在 Web 服务领域，技术服务描述文档是典型的 WSDL 定义和 XSD 模式。该文档是又一越来越具重要性的文档形式。每个文档都可以被分成服务元数据，每一个服务元数据都提供和服务相关的信息。我们可以把服务描述

文档整体看做是建立了一个服务合同——一套必须被满足并被能为潜在的服务请求者所接受的条件，以确保成功完成通信和互动。

- 服务端点
- 每个服务操作
- 每个由操作支持的输入和输出信息
- 每个信息内容的数据表示模型
- 服务和操作的规律和特点

因此，服务合同界定了大量的解决方案环境的底层架构，甚至可能提供的语义信息，这些语义信息能够解释作为方案一部分的服务是怎样完成一个特殊任务的。总之，该信息确立了该协议的条件，服务的用户都应该遵循这个条件。

因为许多协议共享这个合同，合同的设计就尤为重要。同意合同的服务请求者便依赖于这个定义。因此，在合同最终发行前，应进行仔细的维护和校对。

在服务定向中，服务合同代表一个基础原则。因此，它能支持并使其它原则得以执行。例如服务抽象性、组合性、可发现性以及接下来要讨论的松散耦合。

怎样才算是松耦合呢？

没人能预测一个 IT 环境将如何演进。自动化方案将如何发展，是集成化还是随着时间的推移被取代，这些永远都不会被精确的规划出来，因为促使这些变化的要求对于 IT 环境来说是外在的，应用服务定向的主要目标就是能够用高效的方法回应无法预测的变化。通过在服务之间建立松耦合联系，我们就能实现其灵活性。

对于一个传统的分布式结构模型来说，SOA 建立在一个概念之上，即把方案逻辑划分到许多逻辑单位之中。这些逻辑单位能够被集中在一起令商业工作自动化。一个将 SOA 和其它架构区别开的特征就是这些逻辑单位是如何被要求联系在一起的。

这就是耦合的来由，软件程序间的耦合可被看做是代表测量依赖性的一种方法。依赖性越强，耦合就越紧。而无依赖不表示一个去耦状态。我们强烈推荐最大限度地减小 SOA 里逻辑（服务）单位的依赖性。这种特定的关系被称作“松散耦合”并且通过将服务和请求者间的依赖性限定到服务合同所表达的信息范围内，同时在设计协议时不必要只针对某个特定的服务请求者，就可以实现松耦合。

如果坚持这个原则，新建的服务就会明显的独立于其它服务。这样就使正在标准化 SOA 的机构能够积累一系列服务，作为相对独立的逻辑单位，这些服务可以按照需要被配置到新的结构中。当为了建立一个面向服务的企业而创建多种服务时，这些服务可以被分成特定的服务模型。这就使他们在自己封装的逻辑方面更为专业。

在标准化一套服务模型时，能够成功抽取更大的域。这就使机构能够通过创建代表不同域的服务层而在企业的特定领域有效的建立具有战略意义的松耦合联系。通过这样做，松耦合带来的利益会得到增大—最明显的是它促进了机构灵活性—。这就展示了作为范例的服务定向并没有局限于服务设计层面。它的原则可以在企业整个领域得以广泛应用。

下一步

在文章的第三部分，我们将继续探讨另外两个原则，其中一个是服务抽象性。我们将扩展有关松耦合的解释，并研究怎样通过让基础逻辑和被服务封装的技术独立演进把抽象性和促进组织灵活性联系在一起的。

（作者：Thomas Erl 译者：杨君 来源：TechTarget 中国）

服务定向原则的第三部分：服务抽取性和服务重用

与我们之前谈的服务合同及松耦合直接相关的服务定向的一个方面就是抽取性。只有通过抽取性我们才能控制基础服务逻辑向外部世界展现的那部分。

抽取功能和技术

提及服务接口层面的提取，原则上是鼓励人们建立类似黑盒的服务，并有意隐藏潜在用户的基本详细资料。通过规范的使用服务合同可以完成抽取。将服务的公开信息限制在服务合同指定的范围内，就可以极大程度的在私人（隐藏）信息和公开（可消费）信息间实现分离。

这里对逻辑服务代表事物的数量没有限制。一个服务可能只是执行一个简单的任务，或者在整个自动化方案被用作网关。对一个服务能使用的自动化方案的来源也没有限制。

举个例子，单个服务可以揭示多种不同基础系统的逻辑。事实上，当我们在向服务模型标准化迈进时便建立了一个和营业个体及和商业活动相关的功能环境，人们希望在充满旧方案的环境下，一个服务能够普遍揭示依赖许多不同系统的功能。

服务接口层提取是分布式平台提供的固有的品质，例如组件和基于服务的 Web 架构。Web 服务的应用是协同的，因为它提升了可提取的层次，使其远远超过了功能层面。Web 服务从基础自动化逻辑中提取专有的实施细节，这使潜在用户免于和特定的供货技术相连接。尽管我们把抽取看作是服务的一个特征，但事实上却是集中抽取基础逻辑的单独操作。服务就是这些操作的容器。任何既定服务抽取的水平很大程度上取决于每个服务操作的水平。

这就要强调服务合同的设计。服务合同上表达的越多，我们抽取的内容就越少。服务合同越一类化，服务的客户就会更不具体，过程就越缓慢。这就决定了我们选择从服务合同中要表达的(而不是提取的)重用的潜能

通过重用促进灵活性

不管是否是即时要求重用，服务定向支持所有服务中的重用。这个基本的原则迫使我们尤其注意每一个自动化逻辑的交付单位，我们称其为“服务”

最初的战略目标和重用有关，即用可重复价值将每个服务定位成一个 IT 资产。随着可重用资产的增加，要少建设而多使用我们已有的一切，完成新业务自动化要求的机会也在增加。

人们希望通过减少建立自动化逻辑的时间，改进机构对变化的反应能力。通过减少共同努力，自动化要求的完成有望更有助于提高成本合算的效率，令提高 IT 开发环境的效率成为可能。这听起来像是无理的要求，但是为了实现这些利益，许多机构都在创建高度可重用服务清单上面进行了巨额投资。

这一原则有利于各种形式的重用，其中包括应用程序互操作性，组成和建立跨领域服务或公用服务等。正如我们以前设立的服务一样，一个服务就是一个相关业务的集合。因此，由单个操作封装的逻辑必须可重复使用，以保证其作为一种可重复使用的服务。

对意义深远的重用的强调，也突显了作为一项执行方案 Web 服务的适用性。通过行业标准通信框架，使每一个服务都可以得到使用，并可以大大拓宽重用的潜力，，因为服务封装的逻辑现在已经可以为服务请求者所用，这些服务请求者由不同的基本技术建成。

这归根结底取决于服务合同

这些原则让我们再次思考要求使用的服务合同，合同的内容决定了什么要被抽取出来，什么不用抽取。通过设计内容我们能决定没被抽取部分的类属和重用性。这就很有必要把一个服务看成是一项投资。建立面向服务方案逻辑往往更昂贵更耗费时间，因为人们

需要考虑即时战略要求以外的事情。因此，鉴别服务定向打算完成的任务在证明投资合理方面非常非常重要。

下一步做什么

我们的文章已经进行一半儿了，希望能够进一步阐释这些独一无二的特征、要求、以及服务定向范例能带来的潜在利益。

在第四部分，无论一个环境里是否真正存在一个服务登记处，我们将要讨论发现服务的必要性。我们还要近距离观察一个十分重要却经常被人误解的特征即服务的组合性。

(作者：Thomas Erl 译者：杨君 来源：TechTarget 中国)

服务定向原则第四部分：服务的可发现性与可组合性

我们继续探索服务定向，我们现在关注两个原则，这两个原则似乎没有受到应有的重视。当涉及到服务设计时，人们的目光都集中在和 SOA 营销普遍相关的设计特征上，就是松耦合和重用。这些都很重要，但却不是实现 SOA 转变过程长远目标的关键，我们还需考虑得更多。在文章的第二和第三部分，我们讨论抽取和战略性的使用服务合同是怎样描述另外两个原则的，这些在一定程度上有助于实现可重用的松耦合服务。但是，如果不能被那些负责创建新客户的人所发现，即使是重用率最高的服务也不管用了。另外，如果不能形成有效的组合，即使是最松散的耦合重用的潜力也十分小。

服务的可发现性

服务可发现性这个特点能够帮助避免建立冗繁的服务或者执行冗繁逻辑的服务。因为每个服务操作都提供一个潜在的可重用自动化逻辑。和服务相关的元数据不仅需要充分描述服务的整体意图还包括单个操作提供的功能。

服务定向和可发现性相关，但又不同，在结构层面，服务可发现性指结构提供发现机制的能力比如一个服务登记簿或目录。这些扩展都成为支持 SOA 实施整体基础设施的一部分。

在服务层面，可发现性原则指的是单个服务的设计。所以不管具有可发现性的产品或延伸在它周围的实施环境是否存在，单个服务被设计得能够尽可能被发现。

原因是这里不需要服务登记簿，因为没有足够的服务目录来保证有一个服务登记簿，服务应该被设计成高度可发现资源。这样，当服务文件增加的时候，人们就可以更好的管

理服务的改良过后的统治，因为每一个服务都配备了足够的元数据用来恰当的表达其意图和能力。

服务组合性

随着服务文件增加，服务的组合在所难免，并且成为建立面向服务方案这个设计的很重要的一个方面。主要的原因是这个特殊的原则非常重要，以至于它能以这些组合中现有成员、控制器的身份参与其中。

任何服务都可组合的要求同时也强调服务操作的设计。可组合性是重用的另一种形式，因此需要用标准方式（和恰当程度的颗粒性）来设计操作以便最大限度的增大组合的机会。

普通 SOA 扩展强调组合相关性的和谐一致。这里，面向服务的业务流程可通过一个组合语言表达出来。例如 WS-BPEL，将流程本身归类为一个由父进程代表的服务组合。希望服务高度组合需求的和即时组合要求是否存在无关。

组合的可发现性

我们文中解释的每个原则都相互关联。例如，服务组合性，和其他几个原则一起应用的程度有关。

甚至可发现性也和有效组合相关。服务提取的基本规则就是一个服务能代表来自一切被支持源的逻辑范围。如果服务封装了其它，我们就有了一个组合。为了建立一个有效的组合，服务设计者需要一个方法，这个方法能找到能够用作组合部件的最适合服务。另外，一旦组合得以完成并被部署，代表这些服务的潜在用户能从意识到其存在、目的及其潜质方面获益。

发现支持所有这些情况，并使其得以实现，因此加速了服务定向的进程。

下一步如何

目前为止，我们讨论的许多原则都关注于服务合同的设计和利用。在下面的文章里，我们将要描述服务的无国界性和独立性，并挖掘其深层的内涵，宣传一个服务基本服务的特定设计特征。

（作者：Thomas Erl 译者：杨君 来源：TechTarget 中国）

服务定向原则第五部分：服务独立性和无状态性

机构继续以服务的形式建立企业自动化逻辑，这就极大的需要增加服务在运行时间操作的可靠性和效率。在用大量可重用服务组合一项服务时，这个需求就更加重要了。

可重用服务和并行操作

重用是 SOA 的核心部分，其作用非常重要，一些与企业相关的 SOA 变化的战略目标都和其有直接联系，以便成功实现自动化逻辑的重用。因此，我们要保证交付的服务不仅拥有重用逻辑，同时在用于现实世界时，还能被重用。

总之，我们要增加重用的机会。每一个被划分为“重用”的服务能为潜在的大量客户程序所用。结构是可以预测的。随着时间的流逝，我们需要同样的服务来促进不同业务流程或服务的自动化，这个服务可能成为众多服务组合的一部分。

这最终翻译成大使用量、不可预测的使用方案和一个我们非常有兴趣准备的运行条件；并行存取的运行环境。当一个服务被两个或更多的服务用户同时访问时，就会产生服务实例。这些的发生很大程度上取决于负责运行服务的供应商运行时刻平台。

但是，为了构造基础服务逻辑以便方便并行存取的条件和其它和与依赖性相关的条件，我们需要通过几个重要的步骤来为服务的设计定型。

这涉及到两个设计原则，我们将在文章中简要介绍。

- 服务是独立的

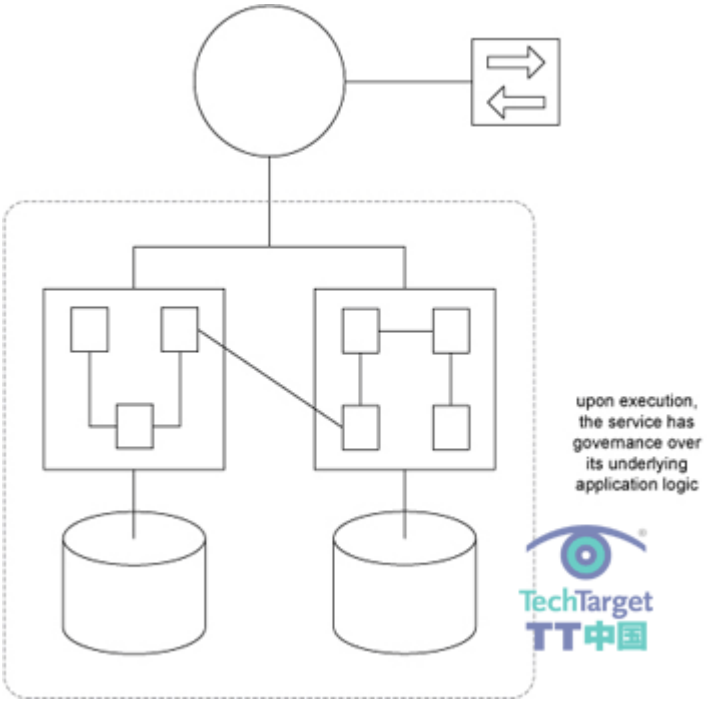
• 服务是无状态的

如果创建一个使用某个特定服务的程序，我可能不知道或不关心有多少其他的人正在使用这项服务。我会想基于服务合同和辅助 SLA 里的内容，这个服务能做些什么。因此，我会依靠服务的性能来提供一个行为、实行性、和可靠性可预测层，不管其是否会被使用。

服务独立性

当涉及到分解时，服务定向带来许多不同的态度。在建立一个企业服务清单时，尤其需要强调将清单上的每个组件定位成一个构造块。

对于服务来说，要提供可靠可预测的性能，他们需要很大程度上控制基础资源。独立性代表其量度，这个原则强调单个服务拥需要有高度的个体独立性。



通过增加服务在执行环境的控制，我们减少了在企业里共享资源所要求的依赖性。尽管我们不能经常提供一个被封装的逻辑专有所属权的服务。我们主要关注的是在执行时间里不管逻辑代表什么，服务能够实现合理的控制。

因为衡量独立性的不同方法的存在，有助于将它们区别开。下面，我们单列出独立性的两个层次。

服务层面的独立性—服务间的界限清楚明晰，但是服务也可能共享基础资源。例如，一个封装旧环境的包装服务，它控制旧系统，但同时也和老用户共享资源。

纯粹独立性——服务拥有并控制基本逻辑。这是建立支持服务逻辑时最常见实例。很显然，一个含有纯粹独立服务的服务清单更令人向往。这不仅帮助我们处理伸展性问题，如并行存取条件，还能帮我们将服务定位的更可靠以应对在杠杆作用重用自动化逻辑时“单点故障”。但是它通常要求建立新的服务逻辑并需要特别部署。这就需要花费更多的时间和精力。

服务无状态性

独立性是 IT 最容易理解的一个方面，但状态信息的构成却不太透明。由于这个原因我们需要在讨论这个原则前，花点时间来定义状态管理。

状态指一件事物的特殊情况。一辆车在运动的状态下行驶，而不是在固定不动的状态下行驶。

在商业自动化方面，一个软件程序通常被认为是有两个主要的状态。

- 主动状态

• 被动状态

第一个状态代表被调用的或者被执行的软件程序处于一个主动状态。当程序没有使用时，就处于被动或非主动状态。

在设计程序时，我们对软件程序在主动状态下的表现很感兴趣。我们如此感兴趣，事实上，我们还有应用于程序的其它状态，该程序代表主动状态的特定类型。在我们谈到状态管理时，有两种基本状况：

• 无状态的

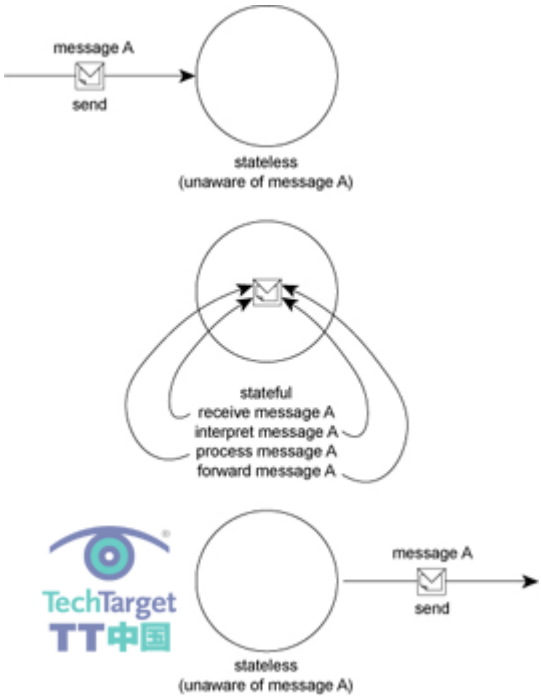
• 有状态的

这些术语常常被用来证明一个程序的主动或运行时间状态，因为这和执行指定任务的处理有关。我们可以把这个数据称为状态信息。

一个程序可能是积极的，但其不一定参与了状态信息的处理。在理想情况下该程序是无状态的。正如你所猜想的，一个积极处理或保持状态信息的程序被认为是有状态的。

处理过程要求服务在常规基础上，增加被重用、组合和并行存取的机率，同时也要优化处理逻辑的服务。当我们建立面向服务架构时，需要特别注意状态管理。因此，我们如此强调优化架构内服务信息的管理，以至于我们有了一个用于服务设计方面的原则。

该原则规定，服务应该最大限度减少他们管理的状态信息的内容和他们有状态的期限。在一个面向服务里，状态信息通常代表现有服务活动的特定数据。当服务在处理一个消息时比如，它是暂时有状态的（图 2 ）如果一个服务负责在长时间内保持状态，其被其它用户使用的能力就会受到阻碍。



至于独立性，无状态是一个服务最佳状态。该状态能促进可重用性和伸展性。对于一个服务来说，应该尽量保持小状态。其基本服务逻辑应该考虑设计成无状态的过程。另外，架构本身就应该配有在大范围服务内支持该原则应用的延期扩展。

下一步如何？

在服务设计中，无状态性和独立性并行不悖.他们每一个都支持另一个的目标，并一起支持 SOA 的目标。在文章的最后一期里，我们将关注面向服务设计原则间的主要关系，以及怎样通过使用服务抽取层来进一步更好的应用范例。

（作者：Thomas Erl 译者：杨君 来源：TechTarget 中国）

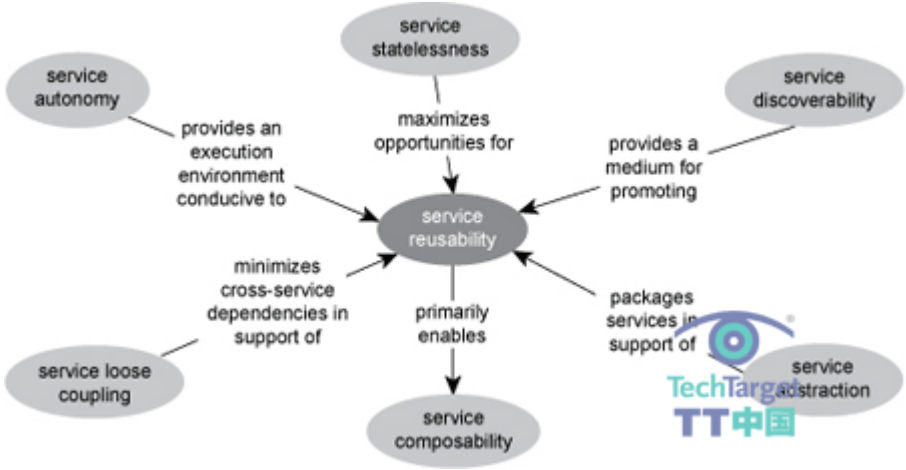
服务定向原则第六部分：原则的相互联系和服务层

既然我们已经分别介绍了面向服务的普遍原则。我们就能研究它们应用的其它方面。在文章的最后部分，我们将讨论这些原则是怎样相互联系并相互影响的。它们之间的关系非常具体并形成了面向服务范例独特的动力。我们简要的看一下这些特定的原则是怎样通过使用抽取层来让这些原则的应用超越服务层面的。

这些原则是怎样联系起来的？

要深入理解服务定向，认清应用这些普遍原则的原因和后果非常重要。通过研究原则间的相互关系，我们不仅要保证每个服务支持或被其它服务支持得以实现，同时要对什么能令服务定向和其它设计范例相区别进行评估。

拿服务重用性这个基本原则为例，在图 3 中你会看到这些椭圆符号代表的原则和箭头表示的关系。服务重用被放在了中心，因为这是我们目前感兴趣的关系。指向该原则的箭头代表其它原则产生的影响。从符号指出的箭头指明了实施可重用的自动化逻辑怎样影响到另一项原则的实现。



在该示意图里，明显可以看出有多少其他原则影响并支持服务在现实中能够实现的重用。实现重用不仅取决于设计，还和服务在现实世界的实施有关，该实施是服务定向的核心目标，也是实现 SOA 利益的关键.因此一些其他的原则显露出来支持这个目标。

我们仔细研究一下这些关系。

• 服务独立性形成了一个有利于重用的执行环境，因为服务实现了高度的独立和自治。服务的依赖性越小，可重用性的范围就越广。

• 服务无状态性支持重用，因为它能最大限度地提供服务并促进一个普通服务设计，这个普通服务设计推迟对状态管理服务范围以外的对特定活动处理。

• 服务抽取性促进重用，因为它确定了黑匣子的概念。专有数据处理详细资料被隐藏起来和潜在消费者只知道有一个由普通公共接口代表的存取点。

• 服务的可发现性促进重用，因为这可以让那些建立用户的人去寻找，发现和评估提供重用功能的服务。

• 服务松耦合建立一种内在的独立性，该独立性把一个服务从即时联系中解放出来，这就使实现重用更为简单。

• 主要是由于重用，服务组合性才得以实现。当这些被组合的服务是为了重用而建立时，通过将现有服务组合，自动化要求就能够得以实现。（从技术角度来说，可以建立一个服务，该服务的唯一目的就是被其它服务组合，但是我们普遍强调重用。）

研究原则间的关系可能很复杂。这不仅是一个原则是否被应用的问题，同样也是其被应用程度的问题。

在该系列文章的前面，我们知道，每一项原则可以落实到某一个程度。这可以使量度一个原则实际实现的程度及其可以在何种程度上影响到其它原则十分具有挑战性，。不过，根据第一手的经验，我可以告诉你，反复浏览面向服务的分析与设计过程可以使我们熟悉服务定向，并最终明智的判断这些原则在何种层次上相互联系。

抽取，松耦合和服务层

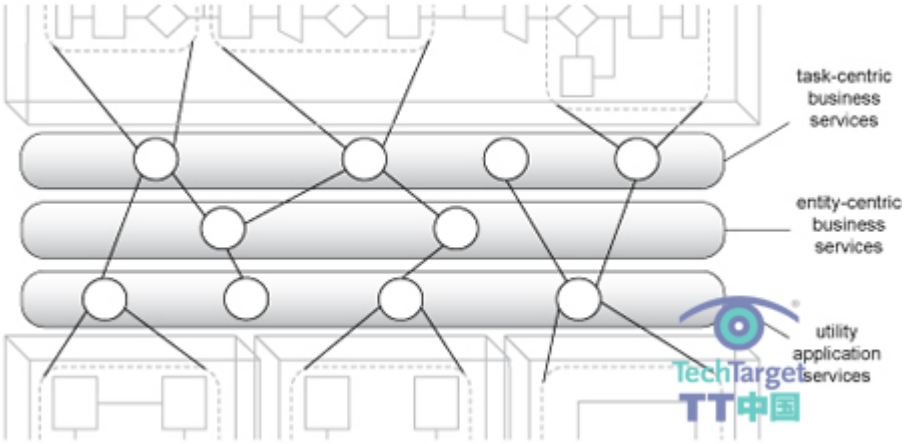
学习这些关系使得我们要考虑到一个既定服务的单个原则的应用以外的东西。但是，要在整个企业范围成功的应用服务定向，需要进一步研究超越现实服务界限的一些特定的原则。

我们已在由六部分组成的业务分析和 SOA 系列文章的第二部分介绍了服务模型。简要的说，作为一个整体，服务模式，用预先定义的设计特点为各类常见的服务建立样板。在业务分析和 SOA 的文章中，我们的重点是有业务环境的服务模式，其中包括：

- 以实体为中心的业务服务
- 任务为中心的业务服务
- 过程服务

另一个我们应该简要介绍的模型本意不是以业务为中心的。我们将其看作是一个应用或一个基础设施服务，这个基础设施服务普遍代表一个处理逻辑的实体，该逻辑用于解决跨领域问题。常见的例子包括通知，事件日志，异常处理和数据格式转换

如图 4 所示的情况，涉及三个服务层，底层是把先前所描述的服务模式称作实体应用服务。



使用服务模型使我们能在大范围内实现功能抽取的概念。基于同样模型的一组服务和其有共同的特点可以在企业里集体抽取一个域。这样就能有效的建立一个服务抽取层。

这些抽取层不用局限于先前我们提到的那些服务模型。一个机构可以定义自己的一套服务模型。每一项服务模型都可以具体到它喜欢如何分割逻辑企业域。举例来说，一个商业服务模式，可以专为一个公司业务部门所建立（如人力资源部或销售部）。基于该模型的服务就会有功能边界，这些功能边界被局限到机构的那个部门，并且共同封装一个和该部门相对应的域。所有的这些任务将服务可抽取性这个原则提升到了一个新的层面，因为我们应用的服务超越了单一服务界限，代表服务的集合。

使用域级别的抽取，让我们也可以应用远远超出服务界限的松耦合。封装域的服务层代表企业内部的功能逻辑体，这些功能逻辑体为了能让普通业务流程和任务自动化，需要相互作用。就我们所知，松耦合通过让服务相对独立的演进令 SOA 受益。一旦应用于企业领域，服务抽取层在一个机构的所有部门建立了松耦合。因此，在支持高效、跨域交互作用时，能够准许域更加独立地演进。

结论

在目前市场里，围绕“SOA”一词的使用有很多歧义。了解那些被认为是"以服务为本"的事物的真正含义是什么，比以往更加重要。对服务定向设计范例和其原则的丰富知识，可以帮助我们更好的理解。这使杠杆作用面向服务的计算机处理技术为了支持你的战略目标可以提供些什么，并能使评估“面向服务”的产品和平台的合法性所需的清晰度得以实现。

(作者：Thomas Erl 译者：杨君 来源：TechTarget 中国)

了解 SOA 生命周期

正如面向服务架构（SOA）自身一样，SOA 生命周期是一个受到广泛关注的话题；而专家对生命周期的定义各持己见，他们认为需要强调的领域也不尽相同。本文主要阐述了 SOA 生命周期的基本概念，及其与传统的应用程序开发生命周期有什么不同、对 SOA 架构师需要执行的任务有什么影响。文章汇总了专注 SOA 的分析师的观点，提供该话题的专业意见。

OnStrategies 公司首席分析师 Tony Baer 简短地总结了 SOA 生命周期需要承担的任务，他说：“我们要想方设法把事情搞定。首先，你要明确要求，收集需求条件，设计解决方案，然后对方案进行开发、测试、部署以及管理。在生命周期终止时，你需要修改方案或者更换方案。”

他补充道，SOA 生命周期与传统应用程序生命周期的不同之处在于，管理 SOA 生命周期需要维持松耦合服务的秩序。

Burton Group 公司研发主管 Anne Thomas Manes 表示同意：“实际上，SOA 生命周期相当复杂。你可以想想服务生命周期，包括最初的身份认证阶段，及在最初的位置提供服务的承诺。”

Manes 提出的 SOA 生命周期第一步有别于其他人提供的第一步。一开始，她会事先进行规划，得出创建服务及根据服务编写应用程序所需的费用预算。

“当你观察 SOA 项目的起始位置时，”她说，“你需要明确成为一项服务所需的条件。许多组织并没有仔细审查究竟应该在哪建立服务。目前，很多工作人员都没有事先进

行规划。他们通常会说：‘我们获得了这项很棒的新技术。我们要将一切都设为服务。’他们并没有真正经历决策过程——明确哪些应该成为服务以及应该在哪付出努力。决定哪些成为一项服务是生命周期中非常重要的一个方面。

Macehiter Ward-Dutton 公司研发主管 Neil Ward-Dutton 从了解 SOA 应用程序的情景开始，全面地阐述了 SOA 生命周期。

“在我们看来，SOA 生命周期可以从企业架构和商业架构等情景理解开始，通过服务分析和建模向各方面拓展。建模不仅仅和功能性事件相关，还与安全、性能、审计等非功能性的事件相关。然后就有了开发、测试、供给、监测和改变管理。即使你的想法局限于从应用程序开发的角度来看 SOA，这也提供了前所未有的考虑方式。例如，你需要考虑采用一致性方法来建立商业服务，这样的话非功能性事件也可以与基础架构服务的运行时间保持一致。”

ZapThink LLC 公司的主分析师 Jason Bloomberg 认为可以在服务中找到 SOA 生命周期和传统应用程序生命周期的主要区别。

“SOA 生命周期的主要挑战在于，实际中有两套彼此交叉的生命周期在运行。”

Bloomberg 说，“一方面，你依然拥有传统的软件生命周期，此时服务是运行软件的接口。你必须采用以代码形式编写的软件。那就是传统的软件生命周期。但是，现在你还拥有服务生命周期，在元数据级中发生作用。在你更新服务、创建服务及重新配置服务的时候，服务生命周期又发生在不同的级别上，因为除非在软件中，否则它就不涉及新的编码过程，但我们的目标是要在元数据级中完成这些变化。”

因而，面向服务需要一种新的生命周期，能将注意力集中于元数据和注册表/存储库的内容。

“SOA 架构设计部分的关键在于建立和维持服务抽象。” Bloomberg 说，“一方面，服务是软件的接口，你必须运行相关软件。另一方面，服务是商业能力的抽象表征，企业将这种能力融合在进程中。所以你还需要支持软件的抽象概念，支持其成为企业元数据，使企业变得更加灵活，这是 SOA 的主要目标之一。”

SOA 基本生命周期

Current Analysis LLC 公司应用程序基础架构主分析师 Bradley F. Shimmin 提出，利用服务建立应用程序的基本 SOA 生命周期需要八个主要步骤。有些方面与其他分析师提出的步骤不一致，但没有本质区别，因此我们从这里开始讲。

利用服务建立 SOA 的八个步骤

1. 收集数据：包括收集企业需求和使用案例
2. 设计：包括明确服务需求、设置服务策略、建立一致性任务、建立并测试模型以及构造数据集成
3. 发展：包括开发服务、根据服务编写应用程序
4. 质量保证/测试/认可
5. 部署
6. 监测/管理
7. 变化
8. 退休

谈到这个纲要时，Shimmin 说：“我把它看成是梯子上的梯级，通向生活中确实存在的某些东西。你必须从某个地方开始，然后又在某个地方结束。生命周期始于企业分析师从企业中收集信息，这些信息主要是关于应用程序和服务的需求条件。”

Shimmin 说，信息因公司而定，但是企业分析师通常需要创建一份需求文件，里面有各种使用案例。就像用户需要检查银行财务状况一样，使用案例可以很简单。企业分析师会提供生命周期需要承担的详情。这与最终的应用程序没有关系，只是涉及需求条件和具体使用案例之间的关系。

Shimmin 说，在收集阶段，企业分析师会给 IT 人员和系统架构师带来各种需求条件和使用案例。他们一起工作，明确如何将需求条件转化为最终的软件。从这个角度讲，传统应用程序生命周期和 SOA 生命周期之间存在很大区别。

“你在讨论传统的软件开发生命周期时，你拥有软件的开发生命周期，只是在最后几天扩建应用程序，通常一个应用程序拥有一个项目。” Shimmin 这么解释，“采用了 SOA，你讨论的就是两种截然不同的应用程序类型。你建立的第一种类型是一种服务，这种服务是一个完全独立的单元，没有‘银行帐户的返回值’等具体参数。因此，针对服务具有一种生命周期，另外一种生命周期将那种服务和其它服务合成到最终的应用程序中。”

Shimmin 强调，这么阐述 SOA 生命周期的前提是这些服务都会建立。而不是假设服务已经存在且能重复利用，这一点将在后面说明。

生命周期基本步骤既可应用在服务中，也能应用于根据这些服务编写的应用程序，但是通常情况下这两种生命周期过程截然不同，包括在 SOA 架构保护伞下的任务也各不相同。最初，在开发需求条件的数据收集阶段，哪些将成为服务、而哪些将成为合成的应用程序，两者间的区别并不重要。

但是在设计之前，架构师就要开始明确哪些使用案例要成为服务，如检查银行财务状况；而哪些商业进程又要包含在 SOA 应用程序中。

服务设计

在设计阶段，企业分析师需要和 IT 人员、系统架构师一起坐下来，根据已确认的服务挑选出需求条件在服务策略方面需要承担什么任务，需要布置怎样的服务策略，一致性方面需要遵守什么条件，如服务等级协议(SLA)和公共密钥基础结构(PKI)。

“一些 PKI 和 SLA 可以在收集阶段提前完成。” Shimmin 说，“但是从我所见到的来讲，大多数都是根据软件即将形成的样子才变得更加明确，然后才完成 PKI 和 SLA。”

前两步会产生一些人工的痕迹，如使用案例、设计阶段的模型、以及 PKI 和 SLA。然后开发人员就要根据提供的流程图设计服务。

开发和质量保证/测试阶段与传统的软件开发生命周期非常相似。

“开发阶段建立服务。” Shimmin 解释道，“在第 4 阶段加入质量保证，可以确保已经形成的架构满足 SLA。在这一步中，你也可以在企业测试认可度，从而保证企业用户的需求得到满足。”

Shimmin 指出，经过了质量保证和认可阶段，项目进入开发阶段，有时候也成为“发布”阶段，这是因为在这一步中服务将发布到存储库和注册表中。人工痕迹也都进入存储库中。

“这时，你就进入了下一步，我称之为嵌套软件生命周期。”他说，“因为在这一步你所做的是集成应用程序，所以你可能把已经创建的不同服务综合在一起，然后集成到最终的程序中。可能形成 BPEL 或 Java WAR [Web ARchive]文件，在服务器中运行。要编写这样的应用程序，你得先经历之前的步骤。你需要设计、合成，而不是开发，这就是不同之处。然后你还得经历质量保证/测试/认可等阶段，最后就能部署应用程序了。”

步骤 6 采用了监测、管理及 SOA 运行时间控制工具。为保证开发软件与策略一致，之前的步骤还采用了设计时间控制工具。监测和管理能保证满足 SLA 和 PKI，并符合一致性

需求。如果遇到什么问题，可能还需要回到服务设计阶段，修改某项服务使其符合一致性需求。

Forrester Research 公司副总裁 Randy Heffner 认为 SOA 控制和 SOA 生命周期彼此交叉。

“生命周期就是传送你的服务，你需要在生命周期中有很好的控制权。”Heffner 说，“控制是对你所做东西的覆盖。”

Shimmin 解释，这样生命周期可能得跳回到之前的步骤。例如，在变化阶段，生命周期可能得回到收集或者设计阶段，重新开始改变服务。

“如果你需要在服务中添加选项，” Shimmin 说，“跳过所有的步骤回到第一步——收集，从头开始。若有问题，就跳到步骤 3，也就是开发阶段。如果逻辑上有问题，就跳到步骤 2 设计阶段。”

项目回到步骤 1、2 还是步骤 3，都需要再次经过质量保证和部署阶段。这样，SOA 生命周期就会无止境地循环。

退休是 Shimmin 生命周期概要的最后阶段。采用传统的应用程序管理方法，就可能简单地使程序离线；但是在 SOA 环境中就没这么容易了，因为最初的开发人员和架构师并不了解这项服务在企业中的使用方式。因此在取消一项服务之前，必须检查其与其它服务的依存性。如果一项服务由新的版本取代，必须进行检查并确保依靠就版本工作的 SOA 应用程序仍然有效。

SOA 服务组合管理

Forrester 公司的 Heffner 比较关注 SOA 服务，其中重复利用也是一个因素，他认为 SOA 服务生命周期的最高级别由三个关键部分：

1. 服务消费
2. 服务创建
3. 服务组合管理

“人们都认为服务消费和服务创建是两个子生命周期。” Heffner 说：“你可能在开始使用时满怀希望，但是你还得创建服务。而且应该在服务组合管理的情景下完成，我们尝试着管理长期的组合，以确保我们拥有正确的服务，而服务也能得到合时、合地、合理的应用。我们在创建这些东西的时候，就把它作为内部组合的一部分。”

似乎应该在服务消费前创建服务，但 Heffner 比较关注在 SOA 已经准备好的地方重复利用服务。在那种情况下，SOA 应用程序开发的错误消费可能重复利用现有的服务。然而，他也承认应用程序需要的服务还未完成时，由这种情况来决定。这时你就应该开始第二步，创建服务。

“首先关注消费，这样你采用 SOA 就有正确的想法了，因为那是错误的方法。” Heffner 说，“这使你第一次有重复利用的想法。”

按层次来讲，Heffner 认为服务组合管理应该体现在服务消费和服务创建中。

“如果你正在实施的服务组合管理正好与其它两项有关，你就需要改变用以实现重复利用的模型，而且要用业内还不怎么讨论的方式来实现这种变化。”Heffner 说，“专业人员在讨论重复利用时主要关注‘发现’，但是如果你能仔细考虑，这就是类似于哥伦布式的大事件。”

这种发现的方法相当于一种说法：“我们要开始一个新的项目，让我来发现一个新的服务世界吧。” Heffner 说，这将是开发 SOA 的专用方法。

他说：“如果你仅仅拥有服务消费和服务创建，那就是你即将拥有的。”

采用服务组合管理，你能获得由企业进程域组织的企业主要服务结构组。

“然后在你开始设计一个新的项目时，你就不会说我们看看能找到什么。”Heffner 进一步解释，“你会说这个项目属于这个领域。我们拥有组合管理。这种组合如何体现在项目中？从而在项目中设计重复利用。我们已拥有一套组合，我们正在实施这些主要的商业单元。我们拥有了解、理解这个领域的人员。我们可以采用正确的方法合法地利用服务。因此就不再是发现式的模型。脱离组合，这个项目中我们还要消费什么？”

只有当组合中没有需要的服务时，生命周期才开始创建服务。

SOA 服务消费者生命周期

在 Manes 看来，使 SOA 生命周期异常复杂的其中一项原因在于，生命周期需要同时处理服务创建和服务消费这两方面内容。生命周期在消费方面有别于创建，但仍属于 SOA 架构师的工作。

服务创建生命周期的基本消费在于，建立的服务至少拥有一名消费者，但是其他消费者可能会把现有的服务融入到新的应用程序中。

Manes 认为 SOA 服务生命周期在消费方面需要三步：

1. 检索存储库中的元数据
2. 为新应用程序提供服务

3.测试性能并规划容量

消费者要做的第一步就是检索存储库中的元数据。

她说：“根据契约规定，元数据可以在将要采用该项服务的应用程序之间正确进行连接。”

但是服务消费在部署之前，需要得到正确供给。消费者需要在策略管理体系中注册，此时才允许享受这项服务。使用服务还需要身份验证机制。

第三步是测试性能和规划容量，确保服务能支持应用程序的需求条件。这可能包括协调服务的 SLA。

Manes 说，服务提供商需要了解参与服务、准备服务的负担。

缩短 SOA 生命周期

上述内容大都和 SOA 生命周期的复杂性相关，Interarbor Solutions LLC 公司首席分析师 Dana Gardner 认为，在时间框架方面应尽可能缩短，以维持 SOA 的动态质量。他说，这是 SOA 生命周期和传统应用程序生命周期的主要区别。

“你的服务是动态的，所以你想要根据需求相对频繁地更新或者替代专项服务。”Gardner 这么解释，“更抽象地讲，你要改变那些服务支持的商业进程，因为你想相对较快地进行调整和修正，以保证效率。你想给更多的人——那些与商业进程有关的人——一个说法，告诉他们如何适时修改、优化进程。”

为什么 SOA 架构师的工作与众不同

分析师说，SOA 生命周期改变了 SOA 架构师的作用，使他们有别于传统的企业架构师。

“几乎可以说是思想上的差异。”Gardner 说，“SOA 架构师应该不断寻求更高的生产力。他的作用应该是一个城镇的规划者，而不是高楼的建设者。他需要考虑所有影响企业能力的动态因素，而一个企业的能力往往包括完成各项工作、进军新市场、生产新产品等。因此，从 SOA 角度讲架构师总是在寻求生产力的持续提高。”

实际上，可以认为架构师是实施 SOA 的关键人物。

“SOA 生命周期以很有趣的方式影响了架构师。” Ward-Dutton 说，“首先，很多架构师在以前仅仅是‘办公室角落的设计师’，实际上，这样的人也很少。他们如果想要摆平 SOA，就得拓宽视野，担起责任，协调服务设计人员和开发人员的要求与期待，协调基础结构管理人员和系统管理人员的需求。其次，服务提供商和消费者在 SOA 上很容易出现空白，架构师需要填补这种空白。这些挑战非常严峻，因为它们并非关于技术技巧，或者不仅仅是关于技术技巧。迎接这些挑战要求架构师成为真正的外交官，具备协商能力和影响力。”

（作者：Rich Seeley 译者：周姝嫣 来源：TechTarget 中国）

SOA 生命周期:我们到底在谈论什么?

在 IT 和软件开发中，SOA 生命周期的定义与生命周期这一术语的传统用法有所不同，Infravio 公司技术标准副总裁 Miko Matsumura 如是说。在讨论 SOA 时，生命周期这一术语的不同涵义可能会引起混淆，尤其是在他感兴趣的领域—策略和治理中。虽然，这似乎仅仅是关于术语如何定义的争论，但是，根据他的反驳，在高层大范围的 SOA 生命周期中，对策略生命周期的关注能够为业务提供更具优势的竞争力。

为了澄清原因，Matsumura 写了一份 50 页的关于 SOA 生命周期治理的白皮书。在分析讨论 SOA 生命周期时，他将其分为三个部分:

- 设计时，当把服务集合起来形成商业应用程序时
- 运行时，当 SOA 实现已经能够工作、并且商业活动已经开始时
- 变更时，当为了利用 SOA 的敏捷性承诺，商业需求改变导致发生不可避免的变化时

然而，与开发人员熟悉的、传统的软件开发生命周期(SDLC)相比，这个 SOA 生命周期很少或者几乎没有类同之处，Matsumura 辩解说。

“有点讽刺意味的是:软件开发生命周期(SDLC)与 SOA 生命周期居然没有一点重合之处，”他说。“传统的软件开发生命周期(SDLC)是开始于某种设计或者蓝图。从本质上说，结束于服务或者软件的配置。如果是 SOA 项目中的服务，那么，在某种程度上，公开发布事件是生命周期结束的标志。我知道，某些开发人员了解软件开发生命周期(SDLC)包括更多的含义。但是，我认为，一些传统的开发人员可能会认为配置就是生命周期的结束点。完成配置时，他们会欢呼:‘好了，我们完成了，欢呼吧!’”

在 Matusmura 看来，软件开发生命周期(SDLC)的结束点正是 SOA 生命周期的开始点。他认为 SOA 设计时是 SOA 生命周期的第一阶段，而且，他对设计时的定义包括——把当前发布的 Web 服务组合起来，成为商业应用程序。

“我知道，甚至有一个更早的阶段——人们构建架构、制定计划、画出宏伟的蓝图，以及讨论互操作性、标准、安全和其他事项，”他说。“因此，还存在着一个架构设计阶段。”

但是，在他看来，SOA 生命周期是从“SOA 设计时”开始的，而这个架构设计阶段是开始之前的初始阶段。

“SOA 设计时实际上包括:找出等级较低的服务，并把它们组合成为商业级服务，”他说。“因此，这并不同于一般意义上的设计。”

SOA 设计时引导 SOA 运行时和变更时，SOA 变更时结束就完成了生命周期。生命周期是需要从外部进行观察，Matsumura 说，SOA 生命周期包含着两个方面。第一个方面更为明显，就是 Web 服务的生命周期包括设计时、运行时和变更时。但是，当他谈论到 SOA 治理时，他辩解说，策略层有其自己的生命周期，但在 SOA 实现中，这一内部的生命周期却常常被忽略。

“这一层有其自身的生命周期，有其独有的设计时、运行时和变更时，但是，这一层却是经常被人们忽略的，”他说。

策略也需要被设计、运行和变更，而且它是处于策略生命周期中的，Matsumura 坚信对 SOA 敏捷性的承诺一定能够实现。维持策略生命周期以及快速变更的能力能够提供有竞争力的优势，他说。

“从相反的角度来思考的话，在策略方面，你基本上可以作为公司参与竞争，”他承认。

注意到，在 SOA 应用程序中的商业活动会受到策略治理的限制，如它们如何与商业合作伙伴交流。Matsumura 说，因为策略治理能够确保商业活动按照一定的规范进行，所以，这并不是一件坏事，正如他举的例子“火车不仅是按时运行，而且也要按照轨道运行。”

在他看来，SOA 的敏捷性需要——Web 服务活动的操作必须要有策略的限制。

“Web 服务活动有可能会是敏捷的、灵活的，但是没有策略的限制，它也有可能会非常糟糕、完全失控，”他讽刺地说。

我们拿铁路来类比，如果商业大环境发生变化的话，通过维护一个策略生命周期，一个组织就能够快速切换到另外一个轨道并且采用不同的路线。

“如果你的约束模型比其他公司的适用性更强，灵活性更高，那么，你就可能具备了一个可以保持的、有竞争力的优势，”他说。“随着情况的改变，策略需要改变，而且活动流也需要改变。但是，所有这些变化都是为了保持你的目标的一致性。”

（作者：Rich Seeley 来源：TechTarget 中国）

站在 SOA 喧嚣之后

到目前为止，那些能够证明使其产品进入面向服务架构领域是正确的这些厂商们，他们的产品大多数已经成为符合 SOA 的行业术语了。当初的喧嚣——可以用“非理性繁荣”来形容——已经过去了，现在，开发人员和架构师们都真正地实现 SOA。喧嚣之后，SOA 将如何发展?为此，我们访问了一些分析师和思想领袖，看看他们是怎么看待 SOA 的发展趋势的。

驾驭技术成熟度模型:Gartner 公司的见解

Roy Schulte 是 Gartner Research 的副总裁及卓越分析师

目前，虽然厂商仍然对 SOA 的极度偏爱，但是，众多开发人员和管理者对 SOA 的幻想都已经破灭了。实际发生的变化是各大公司现在将 SOA 应用到生产中——此时，他们认识到 SOA 并不是万能之计，而且它的最初成本也相对较高。SOA 应用程序是由许多“移动部分”组成的，因此，它们要比单一应用程序更为复杂。因此，公司如果采用 SOA 的话，它就需要面对治理、测试、配置、版本控制、原数据管理、服务级监视、安全、互操作性以及其他方面的重要挑战。当然，SOA 并不是引发这些问题的导火索，而是其解决方案的一部分。在所有的分布式应用程序中，这些问题是与生俱来的。SOA 最大的优势就在于——变更时间更短。在某个特定的业务领域中，实现第一个 SOA 应用程序，以及采用单一的非 SOA 设计风格方式实现统一应用程序，前者比后者所需要花费的时间相当或者稍长一些。但是，随后的 SOA 应用程序，以及对第一个 SOA 应用程序所做的变更，耗时更少并且成本更小，因为，它们能够重复利用 SOA 的基础架构和预先已构建好的服务。

Gartner 公司在 2006 年中期所做的一项对 158 家企业的调查问卷，问题是:“在下述这几个方面，您所提到的项目，给您的企业带来何种程度的积极或是消极影响?”

调查结果包括:

• SOA 没有为企业带来显著的成本节省。32%受调查的公司表明，他们在成本方面受到了一定程度上的或者显著的不利影响，同时，47%的公司表明，使用 SOA 并没有为公司节省任何成本。

• 但是，SOA 在商业灵活性方面具有积极的影响。超过 50%的受调查公司表明，SOA 对其商业灵活性有积极的影响，其中 17%的公司表明该有利影响非常显著。

• 在 2005 年中期到 2006 年中期，各大公司在 SOA、Web 服务以及 Web2.0 上的投资增加了 58%。

• Gartner 公司预测，在 2007 年，在新生的、以及实现关键任务的应用程序和商业处理中，SOA 的应用范围将超过 50%，到 2010 年将超过 80%。

SOA 现实带来 SOA 问题

James Governor, Redmonk 公司资深分析师

我们不断地听说，有相当多的公司都做出极其失误的决策——比如，为了使服务可用，重写整个应用程序框架。换句话说，把 SOA 作为研究型项目，这本身就是极端错误的。

SOA 需要成为业务驱动，应该被引导，而不仅仅是采用 SOA 本身。事实上，任何 SOA 新方案的检验标准是，它应该能够很好地适应 IT 与业务之间的结构化通讯。类似服务管理的 ITIL(IT 基础架构库)，SOA 应该能够具备在 IT 和业务之间的更强的灵活性。

另外一个问题是，人们把注意力集中于“面向服务”，而不是“架构”。然而，使 SOA 能够实现价值的，正是其架构和规则。如果没有打下坚实的架构和治理基础，，实现 SOA 基本上就是在浪费时间。

虽然，SOA 有可能被其他的术语所取代，但是，这并不能减小 SOA 发展趋势的影响力。从商业上开始坚定地使用 SOA 的那一天起，我们就一直在实现 SOA 的道路上迈着缓慢的脚步。有些时候，它看起来像是走两步退三步，而且前进犹豫不决，但是进，以后不会再这样了，脚步不再是缓慢了。事实上，当前的商业以往任何时候都更加标准化，这使得 SOA 能够进行最新的迭代，并且应该能够使 SOA 有更深远的发展。

所有这些内容——大 SOA 厂商也正被小 SOA 厂商追随——也就是不需要由少数厂商定义的一些任意的“Web 服务栈”——采用 WS-I 堆栈的模式。因此，举例说明，UDDI 术语的使用越来越少，但是 WSDL 仍然在开发中广泛应用。

工程师着重强调 SOA 优势

Eric Newcomer, Iona Technologies 公司首席技术官

在 SOA 如此有潜力的发展趋势背后，其普遍过热期看似即将结束，因为，用户们开始思考如果他们采用 SOA 的话，那么实际的目的到底是什么。我认为用户开始从他们的研究以及最初的 POC 中清醒，他们开始注重实际项目。举例说明，有一个用户近期投入到一个关于重用的长期研究中，包括他的 ROI 投资回报率蓝图、以及候选服务的优先级列表。现在，他们已经为实施一个实际项目做好准备。我们所了解的其他用户，尤其是电信行业的

用户，他们能够更清楚的了解 SOA 为公司带来的好处——使新产品能够更快地发布到市场中去，并且使指令事务处理的过程效率更高。

与以往相比，主要的不同在于:我们看到更多的用户开始转向实际项目。我相信大多数用户都差不多花费了去年一整年的时间，来进行研发或者是非技术工作——如思考正确方法、如何使用正确的技能的问题、如何组织他们的 SOA 方案等等。

我们总是说 SOA 是一种方法，而不是一项技术，但是，这不意味着用户必须在与厂商的讨论中退让一步，而不能自己做出关于希望如何推进 SOA 的决策。当这种讨论更多的是技术讨论时，厂商能够承担责任——指出 SOA 的特性或者是功能是如何满足用户需求的，或者用户与厂商共同讨论。

但是，当新的事物采用的确实是另外一种不同方式时，它更多的是代表一种文化的改变，而不仅仅是学习如何利用新特性的优势。

该产业成熟性的影响是:用户拥有足够多的特性和功能，他们具有足够多的软件，并且他们现在用有足够多的应用程序(大部分)，但是，他们所需要的是——改进他们现在已经拥有的资源，并且使其应用程序能够更好地协作。用户们需要不断地使用更少的资源做更多的工作，而且许多用户无法缓解底层的 IT bug 所带来的影响，无法使得投资回报率(ROI)正常回归。SOA 能够帮助用户解决这些温，但是，SOA 也需要公司来重新思考当初他们是如何做应用程序开发的。随着需求的不断增加，用户们开始把 IT 看作是可重用资产的潜在集合，因此，取代依靠个人力量构建应用程序，而是采用部门之间，有时是公司之间或者是在网络上来构建应用程序。

SOA 跨越巨大分歧

Ron Schmelzer,ZapThink LLC 公司高级分析师.

SOA 现在成为主流，并且在处理由 IT 无法响应多样性和变化所带来的问题时，SOA 表现得越来越有优势，因此，从这些现象看来，SOA 的采用使我们最终“跨越了分歧”。从下面这些市场上已有的关键迹象，我们知道 SOA 已经达到了主流的接受程度，具体如下:

• SOA 厂商产品的巩固以及平台产品的扩充——尽管前五年，SOA 是作为主要的市场推动力出现的，但它主要是被那些新出现的使用单点解决方案的厂商和新产品所采用，但是，最近的 12-18 月里，这些厂商的巩固和收购，以及 SOA 领域大型平台产品的重要扩展，使得 SOA 的采用更为成熟。那些曾经填补大厂商产品线空白的小厂商，他们现在大都是这些大厂商产品线的一部分。随着现有市场越来越成熟稳固，对于新加入的公司而言，它就像一个更加新鲜充满机遇的市场。

• SOA 交易在数量和规模上的剧增——过去的 12-18 个月，SOA 交易的规模和范围已经增加了 1500 万个相对短期的项目。除此之外，公共部门(联邦、国家以及地方政府)已经对 SOA 做出重要的长期许诺——SOA 将做为首选架构，因此，无论对于私企或者是公共部门而言，SOA 都不再是心血来潮了。

• 用户成熟度以及 SOA 实现程度的增加——公司以往仅仅是简单的使用 Web 服务或者是实现浅层服务的集合，而现在，公司注重的是更加复杂的服务以及 SOA 在安全、管理、可靠性、质量、治理、过程驱动组成以及意义整合方面的特性。如此一来，我们可以很清楚地看到，SOA 不再是孤立的、概念验证或者是小型实验概念，SOA 已经成为公司的关键任务和长期工作的核心。

• 当公司从服务产业转向消费，SOA 就达到了突破点——通过 Ajax、RIA、企业版 mashup 以及企业版 Web2.0 平台进行开发的消费方，他们的迅速出现是这一趋势的信号:公司已经构建了足够多的关键服务，所要强调的是组成应用程序开发要比服务开放重要的多。就 SOA 而言，这表明我们已经达到了某个关键量或者是突破快速发展的爆发点。

• 对企业架构资源要求的显著增长——现在，面对 SOA 成功的关键问题不再是它的商业意义 、工具可用性或者是标准成熟度，而是是否有足够的资源和人员来帮助加快 SOA 解决方案的实施。这一需求表明，对具备 SOA 技能的人员的要求已经超过了目前的市场，因此，市场将迅速转移以满足这一需求。

SOA 不是企业架构的终点，它是随着时代的发展而出现的产物。然而，我们也已经证实了 SOA 的重要性、价值以及复杂性，公司将长期使用他们的 SOA 以使得价值回归，而我们期盼下一个新的架构革命的到来，它可能是基于 SOA 或者是构建在 SOA 之上的。

（作者：Rich Seeley 来源：TechTarget 中国）

SOA 反常规做法：谨防 ABOS

您一定知道 SOA 是什么（这只是我根据您目前在访问该网站所做的假设），但您知道什么是 ABOS 吗？简而言之，ABOS 是指一堆服务（A Bunch Of Services）——一堆互相重叠而互不兼容的服务，拿来就实施，缺少更加开阔的想法，而仅为满足某一个项目的需要。换句话说，当您将 Web 服务技术应用于筒仓项目时，ABOS 就产生了。

那么，是什么导致 SOA 变成 ABOS 的呢？当您暗中破坏贵公司的 SOA 举措时，请记住下面的 5 个 SOA 反常规做法（或称之为 ABOS“最差劲做法”，如果您愿意的话）：

1. 只考虑自己，而不考虑别人：导致 ABOS 服务产生的最好方法之一，就是闭上眼睛不管他人，然后就开始编码。不关心其他的项目，不考虑如果多花一点时间再支持一个参数或是一项操作，就可以让别人使用您的服务的现实。毕竟，您要赶在最后期限前完成工作，而如果停止手头的工作去帮助别人则意味着给自己带来更多的麻烦。那么那些建筑师呢？他们也因为业务架构和路线问题而一直在打扰您。那么，如果您不理会他们到足够久，他们自然会知趣的离开。

2. 只是“搞定了！”（台词盗用，在这里向《王牌特派员》致歉）：这些声音都是设计审查、代码审查、测试计划审查，等等等等发出的噪声。而且众所周知，WSDL 是完全的自文档化，任何人都无法读取这种“谨小慎微”的文档，所以您当然也不想费工夫去试着编写文档——那将是多么浪费时间的一件事情。

3. 向后兼容性是留给那些胆小的：如果别人想用您的 Web 服务，他们就要自己承担风险。如果您改变操作需要重新部署时，那么问题不在于您，毕竟最初不是您请他们来使

用您的服务的。如果您需要改变它，那就完全是您自己的事情，与外人无关（参照最差劲做法 1）。

4. 不向任何人透露您的服务，因为他们实际上可能会试图使用您的服务：这个最差劲的做法是最差劲做法 3 的一个必然结果——如果您对您的服务秘而不宣，那么就会避免许多将会出现的麻烦。服务共享、“即部署即能从任何地方调用”、以及其他的事情……那些家伙从来不用担心要讨好您的老板——这全是为了使您的项目看起来好看。如果其他项目看起来有点差，这是因为他们不得不自己编写服务，而如果他们了解您的服务的话，这本可以避免，那么一切就完美了——所以，我们在优先次序上调高了一或两个等级！

5. 最后一点，但并不是最不重要的一点，使用及忘记：，有一个项目确实成功找到、掌握并/或使用了您的服务中的一个（经过与最差劲做法 4、2、1 的斗争，他们最终实现了这一点），当然我们不希望这是事实，对吧？但终究不久他们就会发现您部署了新的代码拾取（再次参照最差劲做法 3）。因为一堆程序停止运行，调页程序随之停止，在周日凌晨 2 点听到这一消息将是多么兴奋。您现在又多了一个交战胜利的故事，又可以在下周与人聊天中娓娓道来（但鉴于您公司的运行方式，将会出现大量的竞争）。

从较为严肃的角度，希望你能了解我粗劣的幽默（Steve Martin 不用担心我会与他竞争，这是肯定的！）。那些重视 SOA 的公司应该通过设计时存储库/注册表，考虑和规划架构的指导及管理，以及服务的跨版本兼容性和传递/溯源性。仅仅技术堆栈正确还不足够——围绕公司的构建实现企业级 SOA 的问题越大（在许多方面就越难），实现企业级 SOA 的目的在于实现 IT 内的跨项目联盟，以及购买 IT 的公司与使 IT 工作的公司之间的业务/IT 联盟。

（作者：Brent Carlson 译者：Eric 来源：TechTarget 中国）

体验 SOA 的创新价值(一)

在软件行业中，面向服务架构(SOA)和开源软件两项重要变革的出现和融合已经在根本上改变了厂商和客户运营以及消费的方式，更多的集中体现在了先进技术的使用上。客户传统的费用支出:大量的软件使用费用，高昂的年度维护费用，将不会，也不可能再继续下去。

大多数的 SOA 在从最初执行开始到以后的发展规模壮大这个过程中所表现出来的特性在很大程度上和传统的许可模型并不相符。而且，开源软件的出现也使得厂商可以在投入很少财力的情况下开始一项新的工程计划，从而逾越了财政方面的障碍。这也是 SOA 基本优势的具体表现，敏捷灵活，不被任何厂商独自占有和控制。

当这一切逐步的成为了关注的焦点，软件厂商也开始在开源的旗帜下将更多的筹码下注在加重其分量和专长。同时他们也在努力的尝试着 SOA 方面的工作，商讨更多的有效策略，这将是未来赢利的关键因素。从各个企业所推出的 SOA 基础架构可以得出他们赢利的方式和发展侧重点，并能够依此对将来的 IT 模式有个大致的轮廓。

Winston Damarillo 的出现打乱了整个 SOA 软件市场。

他的公司，LogicBlaze Inc.，以及相当一部分厂商都开始抱有这样的观点:理论上各个企业在投入很少甚至不需要投入现金的情况下即可开始 SOA 的实施和发展。这和以往需要提供高昂的许可费用以及被厂商独立控制是完全不同的。当然，企业也可以申请让开源提供商完全掌控这些内容，方便给予更多支持，但这并不是必须的。

LogicBlaze FUSE 在这周最初几天所展示的一组开源的 SOA 平台集合是由 Apache Software 所提供的一系列技术，LogicBlaze 正在向我们宣告“可能，一种很简单的 SOA

运行的解决方案将会打乱整个市场”。Damarillo 这样说过，他是 LogicBlaze 的董事以及知名开源公司—Gluecode 软件公司的创办者。

“在 SOA，ESB 和 Web services 的市场中是无法找到最好的解决方法用以配合这块巨石，” Damarillo 这样说。“好就好在已经有越来越多的客户理解到他们正在使用的是一个什么样的东西，也逐渐的意识到这样的一些软件将会是作为他们推出解决方案的最好催化剂。”同时，在开源这种透明技术的支持下，客户将对未来的发展方向有着更好的控制。在客户需求和厂商规范两者间以前者为主的承诺将会是 SOA 的性质所在。

事实上，在 SOA 和开源之间是有着一种密不可分的联系，这也是 Optaros 公司 SOA 专家以及企化架构组负责人 Adam Michelson 的意思。(Optaros 公司是 Boston 的主要从事于开源软件咨询的综合公司。)依照 Michelson 的话，开源能够为 SOA 消除一些不需要的麻烦和障碍，诸如许可费用，厂商的干预和控制，从而在基层结构中提供基于更高标准的方案措施并得到优化。

“开源已经在一个最基础的阶段停留过很久了，” Michelson 这样说，“SOA 是关注在中间件和基层结构的，而要将所需要的集中的 SOA 那就不得不需要开源了。”

这样的话，一个企业便可以得到一个完善的基础软件集合并可以得到像 LogicBlaze FUSE 那样的支持，这些都会在很大程度上降低了实施 SOA 的成本和难度，Damarillo 说，“从我的立场，这会是人们使用 SOA 的最好途径，因为人们所需要的 ROI 已经包含其中了。”

他还补充到，随着越来越多的 SOA 出现，我们可以清楚的认识到开源在其中提供着越来越多优势体现。“在 SOA 的发展过程中，正是开源让更多的软件技术结合在了 SOA 之中”。

从细微开始入手

Michael Goulde 这样说，开源的平台将会要求客户尽早的去开始执行，他是 Forrester Research 公司关于应用软件基础和发展的分析师。“这样客户可以在更自主也不需要有多余开销的情况下进行评断、测试以及使用，”他说到，“SOA 将是个很好的机会让开源展示它的巨大价值，从而更加迅速，更加经济的建立起新的开源软件体系来”。

Goulde 继续说道:“开源也是中间件的一个选择。FUSE 描绘了一个优于传统中间件的 SOA 完美平台。而我真的很乐意见到更多的这样的例子。就好象 WSO2 的出现一样。

JBoss 也同样提供了开源技术，JEMS(，JBoss Enterprise Middleware Suite)。Damarillo 说道这两家公司都有着相似的模式。“我们都将本是最初的许可费用改成可重复的部署。”

随着开源的关注程度不断增加，越来越多的厂商也开始了他们的开源之路，比如 Eclipse 的开源 IDE 得到了众多的厂商响应。SUN 也在年底的公告上对外宣布了它的中间件产品以及新的软件工具也会免费的提供使用。IBM 也推出了它对开源的支持以及专家级的详细计划，它也开始和一些开源的厂商有亲密的接触。

开源正处在一个众多不确定因素的环境里

大型软件的客户群已经开始介入到了这场争端。在上个月，Hewlett-Packard 公司宣布 DreamWorks 利用 JBoss’的 JEMS 技术实现了 HP 基于 Linux 操作系统的 SOA。“这样的成就能够将商业运作变的更为简单，并且将会对原有的系统结构带来巨大的改变，”来自 HP 的综合服务顾问部门的主管以及全球企业应用服务管理的负责人 Terri Schoenrock 这样说道，“SOA 将会对整个 Linux 系统结构带来天翻地覆的改变并且会着重于节约开支和加速执行这两方面。”

尽管围绕在开源周围的一些不确定因素并没有完全的消失，但是就象 IBM 的 Scott Cosby 所说的一样，IBM WebSphere 已经开始将这些转变集合在一起。“我不能肯定所有

的人都愿意接受开源已经在企业的发展中显露出来。但是我认为在今天而言人们使用开源已经是一个很大的比例。”

同时仍旧会存在一些问题，那就是开源和 SOA 真的能够解决我们所提到的这些受到关注的问题么?“我把开源看作是 SOA 的关键所在，但是这两者之间的联系真的是很难测量。” Cosby 说道，“比如说从特定一款应用软件来说，如果说 CE 能够提供对现状最好的解决，那它又是否同时具有能够解决此类问题的商业产品所具备的功能?”

“如果真的想要构建什么，而且要求需要能够得到全部的兼容并且保证相互的共存，那我的推荐则是‘开源’，这正是使用这个的时候了。” Michelson 说到。“可能，需要考虑的是如果实在是在服务质量，运行时间以及性能方面出现了问题，那再去考虑那些商业化的产品和技术。”

说到底这可能也就是一个关于是否支持的问题。“对于当前的企业，他们需要的是一个能够让他们放手一搏的理由。” Jason Bloomberg，ZapThink 的资深分析师这样说道，“对于开源，有这么一些机会可以为那些开源的厂商所利用，提供更有价值的开源产品，就像 LogicBlaze 和 JBoss 那样。通过之后的支持而不是许可费用获取利润。并且保证会不断的累积和综合以提供更高的价值。”

Bloomberg 说到，也许，对于众多厂商，“这样下去是否能够赚到更多的钱还需要继续更多的观察。”

对于使用者，他们可能会想购买到某种级别的这样的支持，Michelson 说。但是，他同时也给他的客户提出了这样的建议，开源并不就是所有问题的解决之路，“开源不是神话，你不能把它考虑成这将是你的唯一需要花费的地方，它只是让你的一切实施变的更加敏捷!”

(作者：Colleen Frye 来源：TechTarget 中国)

体验 SOA 的创新价值(二)

这是关于面向服务架构(SOA)和开源技术如何改变当前软件行业的系列文章中的第二篇。关于 SOA 和开源如何在软件行业中被采用并带来转变这一主题在本系列的第一篇文章中提到了。

面向服务架构(SOA)和开源软件技术所掀起的新的浪潮已经给厂商带来了深刻思考:应该如何对当前的软件体系进行改变?

“这将会是在接下来的五年时间内最需要思考的焦点问题，”来自企业应用软件咨询部门的 Joshua Greenbaum 这样说道。但是同时“业界并没有对应该如何解决这个问题有一致的意见”，他补充到了这关键的一点。

传统的使用许可模式使得企业开始使用某一软件技术的时候必须面对一个非常大的金额投入，而这一点已经与当前在 SOA 道路上所越来越多被广泛采用的模式相违背。一旦当前的新的架构模式开始占据主导地位，企业将会寻找更为简小的形式以及更多服务支持的应用软件从而完全摈弃以前不得不接受的模式。

“可以预料到的一点即是当企业开始实施 SOA 的时候他们将会把目光放在如何更方便更全面的实现升级。”这是 IDC 调查主管 Sandy Rogers 的观点。

转变，但并不是完全的免费

也许，所有的 CIO 们并不是很乐意的接受关于当前软件使用许可的转变，而这一转变正是给 IT 行业带来了显著的开销节余。考虑到基于源码所带来的收入，厂商已经开始重新拟订了产品的定价模式并有可能重新理解现在这个新的世界。

那当下的思考是“开始这么一种令人向往的新模式，但是其背后的厂商确实又令人生厌，而面前又是基于价值和基于效用的两个选择。” Greenbaum 这样说。从收入的立场出发，他希望在接下来的五年时间内企业能最大程度的实现在 SOA 基础上的“寻常如以往的商业活动”。“对于绝大多数的厂商而言，他们的客户会很乐意填写一张相对大面额的支票以获得他们所需要的服务基础架构。所以这将会是一笔很大的收入。”他这样说到，“但是，再接下去，基于这样的服务架构模式，客户将不必再需要花费更多象以往的费用，比如$14 million 的 CRM 套件。厂商也许可以从这样的例子看出这样的交易应该会有怎样的改变。”

来自 IDC 的 Rogers 这样说:“有一个稳定的区域则是数据包的提供者，但是我们还只是在基础的实施上，能够达到这个高度还是需要一定的时间。”

一旦 SOA 开始得到广泛的开展，“毫无疑问，所有的付费软件必将会面临一个戏剧性的转变，” Greenbaum 说到，“不久的将来你也会乐意去购买这样的应用软件，去购买这些服务并把这些服务集合到你的软件模型中去。这样一来每一位使用者和每一台机器都会因此而做出改变。”

新的许可模式

SAP NetWeaver 的市场顾问资深主管 Ori Inbar 说道，公司并没有准备好将针对现有的价位模式公开出来，至少在这一年内是这样的，对以前的客户，SAP 将继续为其提供一些需要使用许可的软件技术进行相关的升级和使用，而这正是他们对其商业运作中称为“engines”的关键所在。但是，他也希望在这个不可避免的转变中尽快的取得平衡。“在今后的软件使用中通过 Web services 对许可的范围进行转变。”

Inbar 补充道“当软件以服务的姿态出现，开放已经成为许可体系中非常重要的部分。”SAP 的 CRM 需求建议在二月宣布。“这是我们第一次在这方面的尝试。”

其他的厂商也开始在这个新的方向领域里努力尝试。举个例子来说，Flashine 公司，一个 SOA 的管理以及产品支持公司，提供了他们基于唯一客户的价位模式。

“我们已经尽全力的去寻找这么一个有效的方法去使得价格和价值之间的差异不再比想象中的那么严重，” Charles Stack 这样说。他是 Flashine 美国公司的创始人和 CEO。“这是软件行业中正在努力的一个尝试。也许’shelfware’正充分的描述了那些客户所面对的困难。”

对于一个产品，比如 Flashline 的产品最大的价值所在则是大量的用户通过企业的实施而可以展开运用，Stack 这样给我们说道，“但是，让客户去购买这些当前看来可能远不需要的东西必定会是个不小的挑战。” 基于唯一客户的价位模式让我们看到“让所有的客户明白到这样的产品价值所在才能让他们深刻理解到如何从一个节约开销的角度出发，最小程度的控制成本的来展开运作，并且避免那些因为使用许可所带来的额外花费。”

另外一家新兴的公司，StrikeIron 公司为所有出售者和消费者提供了一种类似于开源的 Web services 体系市场，让他们把自己的一部分内容捐献出来共享使用。在这个市场上提供一个月的或者是一年的期限，当然也可以购买只是使用一次的。这样的服务就象是“一个便携的包，” StrikeIron 公司的创始人以及现在主管和 CEO Bob Brauer 说到，“如果你使用了这样的服务并开始去做，一个月已经能够处理很多的业务了。而这样的模式将会让你避免以往的那种许可费用的开销。大多数的客户都还是选择一次性的使用，他们可以这样去试一试。当他们觉得这是真正需要的他们就会把自己的内容也捐献出来。这样做也能够很好的减少当一个企业开始走上不同以往的创新之路时所需要面临的风险。”

投入最少，回报最多

但是，不得不问的一点则是这些不同的定价模式到最后真的能减少软件使用所带来的开销?开源 SOA 的解决方案真的能够节约成本或者让不同企业通过捐献内容以获得厂商的支持从而停止开销?

而且，最关键的一点，在 SOA 的世界里对于厂商而言是否又真的能够获得收益?从 IDC 的调查可以看出，全球范围内的 SOA 架构下软件使用的开销大概只占了所有软件使用开销的 0.6%，而这个比例到 2009 年为止可能会上升到 3.4%。

可能对于当前而言这只是一个很小的市场，来自 StrikeIron 的 Braue 这样说道，他“希望这个市场能够尽可能大起来，如果我不这样认为那我就不会在周末也忙于工作了。”

Greenbaum 说道现在所有的厂商都在想办法让更多的软件为更多的用户所使用，而且会有更全面的功能为每个使用者所利用。“所以这也会在同时扩大了每个用户的使用能力，也在很大程度上降低了实施的难度。”

最后，他建议客户能够把关注的重点放在如何更高效的提高使用能力以及产品的可用性而不是单纯的为了节约成本开销。“软件行业的最终发展趋势可能是你不再花费那么大的数量，并且获得更多的价值所在。SOA 也并不是完全就可以让你不出一分一毫，你仍旧需要不断的投入，只是这种投入将会是非常值得的。”

(作者：Colleen Frye 来源：TechTarget 中国)

体验 SOA 的创新价值(三)

这是关于面向服务架构(SOA)和开源技术如何改变当前软件行业的系列文章中的第三篇同时也是最后一篇。 该系列的第一部分主要是 SOA 和开源如何在软件行业中被采用并带来转变。该系列的第二部分是关于利用不同定价和打包模型来测验提供商以更好的适用于 SOA 领域。

计算一下

IBM 公司为最近推出的 SOA 管理领域提供了 500 名顾问，同时培养了近 90，000 名 IBM 业务咨询服务人员来帮助客户解决 SOA 问题。

据 Hewlett-Packard(HP,惠普)公司主管兼负责惠普公司综合咨询服务的公司企业应用系统服务项目办公室全球管理负责人 Terri Schoenrock 说，惠普公司在过去的几年中新增了 2000 名咨询服务人员。

在 2005 年，SOA 咨询与系统整合仅占全球咨询与系统整合市场的 3%，不过，根据市场研究机构国际数据公司(International Data Corp.,IDC)的观点，预计在 2010 年 SOA 咨询与系统整合的市场份额将提高到 20%左右。

同时，据国际数据公司(IDC)世界服务，SOA 与新兴市场负责人说，每一件与 SOA 相关软件产品的销售，将为围绕这些产品的服务工作带来 6 到 8 美元的收益。

“你不能仅仅以购买来的一件产品作为基础构架 SOA 体系。面向服务的体系结构(SOA)是一种不同的架构方法，” Mayo 说。“由于目前大多数企业将注意力放在资源上，没有

重视服务。如果一个企业真的希望建立贯穿企业内部的 SOA，我不能想象离开了服务提供商，他们能做什么。”

所以，这就是为什么像 IBM, HP 等服务提供商在快速扩展其 SOA 市场的同时不断扩展其服务。

事实上，Mayo 认为未来 SOA 行业比其它类型行业需要更多的服务。原因是什么?人们追求现代化，她说，而标准化还没有十分“标准”。

“SOA 领域背后市场上的两大软件提供商，Oracle 和 SAP，他们分别有各自不同类型的标准，所以他们的组件也不能兼容。有许多公司同时拥有 Oracle 和 SAP 产品，所以，他们将需要 SIs [systems integrators]系统集成商来集成两家公司的产品组件，” Mayo 说。“IBM 公司自身开发了大量连接组合式应用 (Composite Applications)的应用程序组件。Accenture 公司也开发了许多基于 SAP, Oracle 和其它软件产品的组合式应用 (Composite Applications)。不幸的是，这些产品不能兼容;除非你自己有能力或者一些公司利用工具使你能够让这些软件兼容，否则你还是需要‘帮助’”。

最初，SOA 不能使人们工作得更加轻松，Mayo 说。“我们以标准化为目标不断努力，但是想要实现这一目标，迎来即插即用的时代还需要很多年的时间。同时，我们已经有很多种软件，而每天我们又生产出更多的软件产品。”

开不清的未来

惠普公司的 Schoenrock 不认为 SOA 行业比其它类型行业需要更多的服务工作，不过她也认为许许多多不同 SOA 产品提供商的定义和方法使机构更难理解 SOA。

“SOA 使业务和 IT 以一种前所未见的方式发展，所以当哪里有客户寻求帮助，哪里就围绕服务设计展开了‘IT 与业务融合(IT/business alignment)’”，她说。“第二点

是，SOA 业内每一家提供商都以自己的方式为 SOA 下定义。当我们惠普公司来定义 SOA 时，我们要结合它的业务服务、管理与安全、可行的方法与技术、设计与结构、应用与管理等。这些工作本身并不是什么新鲜的事务，他们都以其之前的方式工作，但是，由于提供商不同的定义使工作变得更难以理解。”

Schoenrock 说，HP 本周新推的垂直行业服务优先架构，就是建立一个业务案例，来帮助客户解决原有应用软件与 SOA 融合问题。金融服务业、制造业、分发、Web 服务提供商和公共部门架构都是既包括构架 SOA 的方法，又保构惠普管理平台与其它合作伙伴产品的整合。

“惠普公司发现 SOA 的咨询领域是一个巨大商机，ZapThink 咨询公司高级分析师 Jason Bloomberg 说。“他们将企业重点转向 SOA 领域。与此同时，开始有意识地推出中间件业务领域，一个使惠普公司与所有其它软件供应商成为合作伙伴的领域。 HP 有着丰富的 Microsoft 和 Java 的开发经验”

同时，IBM 也努力发展 SOA 市场，Bloomberg 说。但是，与惠普公司不同，IBM 已经牢牢占据了中间件业务领域。据负责 IBM 业务咨询服务的 SOA 与 Web 服务副总裁 Michael Liebow 说，“大部分宣称在年度中取得的更好部分都在服务和软件两方面进行了适当的整合”

与惠普一样，IBM 也从 SOA 行业获得盈利。“我们在 SOA 服务业务方面看到巨大的增大潜力，” Liebow 说。

软件提供商也都对 SOA 服务有浓厚的兴趣，不过“他们的主要精力都投入到软件产品与技术上” Mayo 说。Bloomberg 补充到，“一些提供商在专业服务领域得到很大收益，但是这是一项挑战，因为服务这一模式与产品模式完全不同。他们在销售、市场和提供价值方面存在巨大差异。以 SOA 软件为例。很长一段时间，他们都不想讨论一个事实，那就

是他们的主要收益来自与提供专业服务。所以，对于提供商来说，递送 SOA 最好的办法就是将服务于产品相结合。”

并行发展的伙伴

服务提供商不仅将 SOA 看作为一个好的收益机会，同时，他们还希望利用目前被广泛接受的开放源码软件(Open Source Software 简称 OSS)创造更多收益。

“由于开放源码大多与基础架构紧密结合，所以企业需要服务提供商保证基础架构支持开放源码，” Mayo 说。“开放源码已经成为提供商提供服务的一部分，服务提供商不仅要支持开放源码，更要使其最优化。”

开放源码将“引领服务”，惠普公司 Schoenrock 说。惠普公司的服务业包括 Linux 参考体系结构(Linux Reference Architecture)。惠普公司除加大与其它商业中间件合作伙伴的合作外，还围绕 JBoss Inc 公司开放源码中间件加强对 SOA 配置的支援。

SOA 与 OSS 是齐头并进的两个领域，其中有一些因素相互重叠集中，Mayo 说。“JBoss(Red Hat Inc.公司本周收购)将 SOA 需求作为服务对象。Optaros(一家位于波士顿得开放源码软件咨询整合公司)以开放源码与 SOA 的结合为经营方向。他们的经营模型是以纯粹的服务为基础，提供有偿服务、咨询、整合以及配置。他们经营得非常成功。大公司渐渐将目光转向这一领域。”

显而易见，变化正在一点一点地发生。不仅表现为提供商们正在抢夺 SOA 和 OSS 服务市场，同时也表现为提供商们正在努力预测，在软件许可模型下这些变化将如何影响未来的收益流。

“整个信息技术业聚焦于最初的 SOA 激活浪潮，” Liebow 说。“一旦企业已经建立起 SOA 架构，那么接下来该做什么?这个问题将影响到软件如何销售如何购买，同时也会

影响到服务需求。业务将发生根本转变，不过很显然，IBM 已经为这次剧变做好了准备。”

Liebow 说，随着软件提供商和服务提供商的变迁，指出行业将如何变化是信息技术业中大多数人相了解的问题。“每个人都看到市场的分裂性，同时，大家都在运用各种手段谋取利益，看怎样才能获得战略上的优势。”

（作者：Colleen Frye 来源：TechTarget 中国）

SOA 的投资回报率

在今天紧张 IT 预算的后 dot.com 世界，增加的规例，更加激烈的全球竞争，快速发生的变化，公司（和政府）要求从他们的技术投资中获得量化的收益。如果一项投资不能为业务增值，没有长官会签署批准这项投资。当人们了解一个既定的技术，以及该项技术如何随着时间的推移提供价值，为 IT 开支计算投资回报率将总是一个直接的过程。不过计算涉及新技术以及例如面向服务架构（SOA）IT 新方法的项时，这通常是一门艺术而非一门学科。

令计算投资回报率更具挑战性的是，架构本身并不提供公司能够通过特定的回益可以识别的具体特征。毕竟，在获得收益之前，架构是公司必须做好的一项投资。并在 SOA 整个使用寿命中能够继续作为投资盈利。那么，经理人是如何在这些项目开始前计算他们 SOA 的投资报酬率的？能够为实施者带来定量投资回报率的 SOA 能带来多少有形的收益呢？另一方面，公司怎样计算那些有形的收益能够带给机构的期望中的投资回报率？只有全方位的理解 SOA 的价值议题，公司才有可能开始计算 SOA 的投资回报率。尽管是这样，在项目完成之前，也很难理解 SOA 的真正投资回报率。因为 SOA 针对的是基本上不可预测的业务变化问题。

降低集成化费用

SOA 提供四类收益：降低集成化费用，增加资产重用，增强商业灵活性，以及降低商业风险。这四种主要的收益为机构各个层面和部分提供回报，该回报取决于公司将 SOA 应用于何种业务问题上。首先，实施松耦合集成化这个方法可以降低复杂性因此降低集成和管理分散计算环境的成本。在转向 Web 服务这些基于标准的接口时，一定程度上降低了集成化成本，真正的胜利要数通过 SOA 在带有紧耦合、松耦合服务的颗粒性的好的层面替代

多种功能呼叫。比起基于 API 的集成化，这些紧耦合、松耦合服务能够更灵活的解决更大范围的交互作用。

为了用 SOA 来降低集成化费用，计算投资回报率非常简单。公司可以将等同于传统集成化中间件方法和他们在基于 Web 服务的 SOA 中的投资相比较。然后和即时特许、配置成本的降低、以及更长时间的维护和变化的成本。通过进一步深入了解集成化的真实成本，ZapFlash,公司能够简单的从紧耦合形式的集成形式移到松耦合形式来实现即时、有效的投资回报率、

增加资产重用

降低集成化费用的同时也能够调整一个最初的 SOA 工程，仅仅把 SOA 看做是降低费用的一个方式目光非常短浅。一方面，即便在一个同种性的环境里，SOA 也是可行的，第二，当公司要求对现有基础设施进行新的使用时，以集成化为中心的思维模式无法为获取投资回报率提供方法。相反，重用现有的服务为要实施 SOA 的公司提供了额外的投资回报率。众所周知，公司很少花大量的时间和预算来创建新的应用程序。这些应用能够解决不断变化的业务需求。人们不够重视新程序开发的部分原因是因为每个新程序都是在和其它以前的程序相隔离的状态下建立起来的，这样就导致了一个新的 IT 难题。该难题最终要和其它程序合成一体，这就更加剧了我们早些时候在这个 Flash 中要解答的集成化问题。

SOA 带来的最大收益就是，用户能够在现有的服务中建立新的业务流程和复杂的应用。换句话说，服务重用成为祷文而不是应用集成。一旦他们建立了新的服务，他们就会为新的复合应用重用这些服务。公司可以从复合应用开发投资中获得巨额回报。结果是，当公司建立并重用日益增长的服务时，能够杠杆作用 SOA 的复合应用程序开发经济学随着时间的推移得到不断改进和提高。将原来花在集成化 70%的资金花在新 app 的开发上也是有可能的。

增加商业灵活性

降低成本增加重用的过程为 SOA 提供了明显的投资回报率，增加业务灵活性是 SOA 最具希望的收益也是最难量化的收益。简化集成化和提高重用带来的是技术密集型收益。业务用户同样要求从 IT 中获得更大的灵活性。而不是简单的只是为了在几个月之内翻越 IT 开发实施循环这座墙。业务用户想要直接控制他们的操作，这样在市场力量发生变化时，他们能很快的相应做出变化。

要求计算商业灵活性收益的投资回报率样本要以业务用户控制业务流程定义和管理的能力为中心。通过面向服务过程，公司能够将整个业务流程流托付给机构的不同部分。每个部分都可以直接控制和管理业务的真实运作。从 SOA 投资获得的商业回报在业务效率和在供应商和业务伙伴操作的内部镶嵌业务流程的能力是极大的进步。增强业务灵活性意味着能够扑捉公司以前不能达到的收入流并为公司足以能够为他们的供应商和合作伙伴提供价值。通过将 SOA 传递到业务用户，该技术可以使业务整体而不是简单的 IT 部门获得投资回报率。

计算 SOA 的商业灵活性投资回报率可能非常困难，因为商业人士应用到 IT 资源的新方法本质上是不可测的。毕竟，灵活性的重点是能够应对意外的变化。因此，将商业灵活性投资回报率的计算控制在一定范围内比如，产品信息经常改变，业务合作伙伴和业务流程经常被改动的状态，以及其它供应链接方案。

减少业务风险和曝光

管理一致性在本质上是商业灵活性问题，因为这样的规章本来就是随机的，并能随着时间而变化。现在，像 Sarbanes Oxley, PATRIOT Act, 和 Basel II mandate 这样的公司内部规章促进了 IT 的实施。对待不一致性的惩罚可能严重影响一个公司的财务状况以及其行政人员的特权。许多公司需要做出明智的计划性决定来看它们的业务操作，并且控制他们的风险，更不用说回应这些规章要求的与日俱增的可见性了。

在面对不断变化的规则时，增加商业的能见度是 SOA 能够带来的业务灵活性收益的一个实例。尤其是 SOA 可以为那些寻求更多更多操作可见性的公司提供降低风险的能力。和增加的业务灵活性相比，管理、一致性以及一般性的降低风险是一个不同的可量化的收益。

当业务灵活性增加了业务机会的同时，一致性和管理可以减少债务。二者都很重要，但是他们面对的是公司不同部分的心理。为了计算基于 SOA 上一致性项目的投资回报率而量化降低的风险是一项非常复杂的提议。一致性到底值多少？答案取决于非一致性将要花费公司多少。降低风险的投资回报率很像保险或担保投资回报率——其价值取决于预防未知的开销。

为了控制业务流程、建立共同的安全、保密和实施原则以及提供可调查信息追踪而执行 SOA 都是 SOA 降低公司目前面临的风险的方式的实例。事实上，许多大公司的中心技术发现 SOA 带来的主要收益源于管理一致性并和降低风险有关。SOA 提供的降低风险是有形的，很难量化一个 SOA 实施的真正投资回报率，在该实施中，风险降低是主要收益。在今天晚些时候，公司将在执行 SOA 将风险降低到任意可以接受的程度的过程中找到价值，并把实施解决的避免损失方法作为该实施 SOA 的基础。

ZapThink 态度

由于 SOAJ 价值议题多面性的特征，为不同 SOA 项目计算 ROI 差异很大，不仅是为 SOA 实施寻求一个单一的 ROI 目标，公司也要采取同样的、反复的、复杂的方法来计算 ROI，这些公司本身就从 SOA 实施中获取 ROI 例如，每次他们把服务定义成公司服务模型的一部分，他们也可以为该服务定义一个相应的 ROI 目标。他们在这项服务上要花费多少钱呢。依照降低的集成成本、增加的资产重用和更大的业务灵活性，他们能从服务实施中获得多少直接或间接的回报呢？另外，随着某些特定的服务在公司中得以重用，将这些服务组合融入这些流程如何能为业务提供额外的 ROI 呢？

在许多情况下，SOA 实施能从服务第一天作用起就提供明确的、积极的投资回报率。但是，这更像是一个 ROI 期望，正如 SOA 实施本质上是可重复的、经常被评估并且是复合的。这样，用户不仅可以量化还能获取 SOA 实施投资的回报。

（作者：Ronald Schmelzer 译者：杨君 来源：TechTarget 中国）

在 SOA 中寻找投资回报率

现在，为新项目做预算时技术要获得收益，IT 人员要守住空泛的承诺是远远不够的。

目前，许多首席财务官和其他财政预算案人员在他们批准一个项目之前都要看看投资项目的回报率（ ROI ）。对于面向服务架构、Web 服务以及其它技术也是同样如此。这又引发了另一个问题。任何研究 SOA、Web 服务技术的人都可以证明，要想事先预知一个项目的投资回报率有多少困难。

我希望能够为大家提供帮助。在这个和下一个专栏里，我们将看一看 Web 服务，SOA 和 ROI。在该专栏中，我要谈一谈，在何处能够找到 ROI，并为大家介绍几个收益颇丰的项目。

在下一个专栏里，我们要介绍怎样使用 ROI 出售你的项目。

同意的声音

最好向那些在此之前进行过多次估算并且是专家的人寻求帮助，询问在 Web 服务和 SOA 项目的投资回报率。Maja Tibbling 是 Con-Way 运输服务公司的企业结构设计师。该公司是一家在和美国、加拿大、墨西哥运营的拥有 26 亿美元的运输和逻辑公司。Tibbling 最近加入了一个 Gartner 应用集成和 Web 服务主题峰会的座谈小组。该峰会的主题是“SOA 与 Web 服务的成本和收益”她向大家展示了一个实例研究，该研究侧重 Con-Way 公司对于 SOA 和使用 Tibco Software 公司工具的事件驱动架构。

Web 服务或 SOA 主要由两种方式传送 ROI—IT ROI 和一个质量或业务价值。在一个 IT ROI 中，IT 部门节省资金，例如通过减少发布新项目或现有项目的工作量。对于一个定性

的 ROI 来说，依靠发迅速布新产品和服务获得的回报，不是获得了额外的收入就是提高了生产力。因此，较少的员工需要执行该项工作。

Tibbling 警告说，当你规划项目时，要考虑到一开始可能没有 IT ROI。她说，事实上，最后你花在 IT 成本上的钱可能比现在的多，一开始就获得 ROI 基本没可能。这是因为开发商可能正在使用新的技术、需要购买的新工具。这将是一个上艰难的而又缓慢的学习过程。

她说“你不能单靠一个项目就量化 ROI”为了证明花费的成本物有所值，一个 SOA 必须系统化并能用于许多不同的项目。

但是，一旦 SOA 或者 Web 服务应用于多种项目。IT 储蓄就更多了、因为每个服务或你建立的模块都能在新项目中得以重用，极大的降低了后面项目的开发时间和成本，这样就会事半功倍，随着时间的推移，越来越多的模块可以被使用，开发新项目的时间也越来越短。因为更多的模块可以重用。这时就会产生 IT ROI—更少的人发布新项目，因为所有可重用的模块都可以被使用，

Tibbling 依照个人经验来看，她说，因为 Con-Way 建立了如此多的模块——她称其为“人工制品”——节省的钱就更多了。她引用了一个用户担保有关的工程。她将在最近几周发布该工程因为她能使用现有的人工制品。如果没有这些可重用模块，Con-Way 发布新项目可能要花六个月而不是几周才能完成。

介于此，SOA 随着时间增加的 ROI 有时将以指数增长。

她说“作为项目的一部分，我需要获取客户信息。”“我了解到一个服务可能要花我一天的时间”所以第一次意识到这点时，ROI 可能并不多，第二次我需要获取用户信息。我已经知道了这点并且我所要做的一切就是呼叫一个服务。因此我为自己节省了 8 个小时。

商业价值 ROI

许多公司能够从商业价值中获得比从 IT 节省资金中更高的投资回报率。业务一个项目的业务影响是惊人的，比其他大的 IT 生产力有更高的金融回报。比如，比平时提前六个月发布一个能够获得收益的服务可能意味着六个月额外的收入。在整个企业提高业务效率比在单一的部门依靠 IT 方面的节省节约更多的资金。

Tibbling 引用了一个 Web 服务项目的经典例子。该 Web 服务项目从业务角度获取了巨额投资回报。Con-Way 公司在美国和加拿大之间运送大量的货物，公司的卡车经常要跨越国境。由于复杂且经常变化的规章，卡车途中还有许多文书工作。每辆卡车要花两到三个小时等待通过海关。

Con-Way 编写了一个服务，该服务能够用电子的方法自动的为已经填好的表格指定路线。这样，卡车就做好了准备工作，这样，它们用不到一分钟就可以穿越国境。节省的时间是惊人的——一天中更多的卡车能够穿越国境。因为司机不用再和政府机构打交道了。

Tibbling 的底线是;在估算时，ROI 更像是商业收益。要做到这一点，就要确保你的 IT 员工和你的业务员工是站在同一战线上的，并保证二者经常进行沟通。在大多数情况下，商业价值比寻找 T ROI 更重要。慢慢你就会意识到你的 IT ROI 会得到增加，

这一切都非常好，但是如果你想在发布新项目之前将 CFO 出售呢并且他或她正在寻找潜在的 ROI?你将会在下一个专栏中就这个问题得到解答。

(作者：Preston Gralla 译者：杨君 来源：TechTarget 中国)

Web 服务和 SOA 的 ROI（一）

任何技术的项目都能得到很好的结果的日子早已远去了。现在，技术不得不考虑它的成本，任何项目都要知道从哪里它能够得到回报（投资的回报(ROI)）——甚至要考虑能够多少回报。

Web 服务和 SOA 的项目同样要考虑 ROI。但很难知道你从那里可以得到回报、你能得到多少回报以及怎样去衡量回报。在文章的这个部分里，我将分析一下你可以在哪里找到 ROI，而在第二部分里，我将分析你能得到哪些类型的回报以及当你配置好项目之后怎样去衡量这些回报。

哪些公司能够期望更大的 ROI

要分析从哪里能够找到 ROI，第一个要考虑的因素是你的公司的业务类型。有些行业的业务比其他行业的业务可能得到更大的 ROI。

Liebow，IBM 全球服务 Web 服务和 SOA 部的负责人说，银行业、保险业、长途通讯、零售业和政府部门都是可能从 Web 服务和 SOA 得到较多 ROI 的行业。他解释说，原因是这些行业需要面对复杂的、分布式的环境，需要水平的综合的能力，而这些都是 Web 服务和 SOA 所擅长的。所以类似的企业和行业同样有可能得到较多的 ROI。

Liebow 也提到可能节省资产重用的费用和业务劳力的费用，但不能节省 IT 劳力的费用。虽然 Web 服务和 SOA 的实现完全可能带来更多的利润，但他认为不应该提早地去期望这些利润，因为这是很不确定的。

四种类型的可以期待的 ROI

ZapThink 的高级分析员 Ron Schmelzer 说，很多公司都要在项目实现之前找到 ROI。他说他的公司找出了四种可以从 Web 服务得到的 ROI。

最简单和最直接的是短期的、战术上的 ROI，这是从聚合业务的成本的降低中直接得到的。他说：“这将在你实现之后就马上节省资金，你可以立即节约聚合的成本。”你将可以不用到中间件，减少转换的时间，还有其他的一些好处。

他说：“这种 ROI 很容易计算的，因为你可以识别那些被代替的业务过程和系统。”

他说，第二种 ROI 是应用程序的重用。这种 ROI 在过一段时间之后（中期）才会显露出来。Schmelzer 说这比聚合成本的节约更难得到。为了得到这种 ROI，一个企业必须识别出应用程序之间的共性，然后为它们建立服务。通过这样，设计、编码和开发的时间将会减少，因为那些服务只要被设计一次而不是多次。

第三个能够获得的 ROI 是因为业务的灵活，这种 ROI 将会在中期或长期的时候得到。这种 ROI 更难衡量——它能令企业和系统能够做出更自然的和更快速的业务决定。你将可以更快地响应市场需求，以及直接和业务伙伴进行交流。这种好处是一种额外的收入，正如 IBM 的 Liebow 所说，是不可估量的。Schmelzer 也认为这是很难估量的，此外他还指出一种技术并不足以产生一个 ROI，因为还需要做出决策的人用这种技术来作出增加收入的决策。

第四种 ROI 是一种即使不是不可能也是很难具体衡量的 ROI，它涉及一个对很多公司都至关重要的问题，一些公司专门部署了一个 SOA 来解决这个问题。

Schmelzer 说，这个问题是遵守和适应政府的法规，例如 Health Insurance Portability and Accountability Act (HIPPA) and Sarbanes-Oxley Act。Sarbanes-Oxley 使企业必须遵循保存档案的法规，而这些档案记录了财政和会计信息。HIPPA 限定了健康记录应该被怎样处理。如果企业不遵守这些法案，都可能被罚款，因此这里的 ROI

是避免了一些风险。一个公司从这里的不到额外的收入或者是普通的收入，但至少会降低被罚款的风险。

综上所述，你可以了解你可以得到什么类型的 ROI，那么，在你实现配置（Web 服务和 SOA）之后怎么来估计你能得到的 ROI 呢？这将是本文的第二部分所讨论的问题。

（作者：Preston Gralla 来源：TechTarget 中国）

Web 服务和 SOA 的 ROI（二）

现在如果你想启动一个 Web 服务或 SOA 项目，你必须估计它的成本和收入。那些技术项目肯定会有收益的日子已经远去了。现在，公司总是要知道技术的支出和收入。这意味着，在很多情况下，你必须衡量一个项目的 ROI（return on investment，投资的回报）。

在本文的第二部分里，我将分析你的技术投入所能得到的回报的类型，以及怎样去衡量这些收益。

是否能够很多的收益

在具体实施一个项目之前，并没有确切的方法可以知道你能得到的 ROI。这就常常让你处在一个哭笑不得的状况：如果你不能证明能有多少收益，你可能得不到项目的资金，但你在项目实施数月甚至数年后才能衡量项目的收益。那么，什么类型的 ROI 是你能够期待的呢？

可能最好的方法是借鉴一些已经从 Web 服务和 SOA 中得到 ROI 的公司的经验。

IBM 全球服务 Web 服务和 SOA 部的负责人 Michael Liebow 说，很多他的客户在很短时间内看到了巨大的 ROI。

他说，关键是小心地选择项目。聚焦在一个特定的业务过程上，然后解决这个问题是最好的方法。他说，如果你这样做，你能得到很显著的 ROI。

虽然 Liebow 没有说他的客户是哪些公司，但他提供了两个他们的 Web 服务和 SOA 项目的细节：

一个很大的信用卡公司启动了一个 Web 服务/SOA 项目，这个项目使支票周转周期从多于三十天降低到少于三十天。他说这个项目的回报在三个月内有数亿美元。

一个长途通讯公司觉得不能依靠一个分散的支付系统，于是自己启动了一个 SOA 项目来将分布的系统来组合成一个单一的、统一的系统。他说，结果是第一年就节省了 5 亿美元。

他说，甚至 IBM 也从 Web 服务/SOA 项目中获益，而他们谨慎地追踪 ROI，并发现了很大的收益。IBM 花了五万美元来给一个配置引擎增加了一个易用的 Web 服务的前端工具，这个引擎是用来提供 IBM 和它的商业伙伴之间的事务的连接的。这个引擎在事务执行时检测订单，并验证事务。在头三个月里，这个项目使事务的错误率从 3%降到 1%，由于这个项目很容易使用，订单处理从 5%上升到 90%。结果是每年节约了数百万美元。

Liebow 承认，这些数字都是 ROI 的最好的情况。而企业必须小心选择他们的项目。他强调选择一个特定的业务流程然后改进它将得到最快的、最清晰的和最大的 ROI。他说，在这种情况下，企业用一个 Web 服务/ROI 项目一般能够在一年或一年内节约 20%到 30%的成本。

他说：“你可以在数月内实现一个项目，然后在一年或一年内取得业务收益，因此这真是一个自动增值的模型。”

开始小，结果大

Liebow 说因为这是一个自动增值的模型，因此可以在实现之前衡量收益。但在现实世界中，常常是说比做容易。ZapThink 的高级分析员 Ron Schmelzer，建议企业从一个小项目开始，在小项目上衡量 ROI，然后扩展它。

他说：“从越小的　迹　饺菀缀饬縍 OI。我们的建议是你从最小的最容易的服务开始，然后衡量它们的 ROI。例如，可以是从一个数据库中取一些信息那样的简单的细小的服务，因为即使是很简单的服务也有相关的 ROI。”

确认那些服务是简单且可衡量的。在启动项目之后，监控、测量结果并计算 ROI。他同时建议在实施之前先估计一下 ROI，然后和实际的结果进行比较。 这将帮助你学习怎样在实施之前预测 ROI，这是很有必要的。他建议在最初的项目中，在很多小项目上做同样的预测。如果在小项目上没有 ROI，那没什么损失。而如果在小项目上有 ROI，那对大项目来说，可能得到更大的收益。

面对 ROI 的阻碍

IBM 的 Liebow 是一个衡量 ROI 的支持者，但他也指出，如果企业只关注实施之前的 ROI，它们可能会得不到可能得到的很大的收益。实施一个 ROI 项目，随着需求链的发展，将是很费时的。

那么，当你处在这样的情况下：你知道 Web 服务和 SOA 的收益将在很长一段时间后才得到，而你不想跟公司的官僚机构磨太多时间时，你应该怎么办呢？

他说：“任何预算都会有一些资金用于创新，即使百分八十到八十五的预算都是被锁定的，在很多地方还是会有百分十的预算是可用来改革创新的。而你可以用这些资金来进行创新，然后 ROI 就会接踵而至。”

(作者：Preston Gralla 来源：TechTarget 中国)

通过 Web 服务快速获得 ROI

如果你想通过 Web 服务获得快速的投资回报率（ROI），咨询师们声称，最好开始设计小型的点对点的集成项目，并且运用对于内部 IT 人员已经很熟悉的技术和方法。

根据工业界观察者分析，通过 Web 服务活动投资回报率（ROI）仍然是一个新的概念，并且目前正在许多客户商店中进行调查。事实上，正在被创制的 Web 服务中的绝大部分都包含具体的集成项目，这些集成项目承担着将系统相接到一起的任务。

理所当然地是，这个过程与目前被 Web 服务应有吹捧为性价比很高的面向服务架构(SOA)非常不一样。通过 SOA，一个公司能够将整个合作阶段的 Web 服务接口标准化，然后运用简单对象访问协议(SOAP)，Web 服务描述语言和其他规范定义文档作为大多数 IT 架构的基础。但是很少有公司已经实现了完成的 SOA 架构，因此在这点上 ROI 的数据实际上是不存在的，ZapThink LLC 的高级分析师 Jason Bloomberg 在 Mass 的 Waltham 说到。

SOA 带给 ROI 的难题就好像经典的先有小鸡还是先有鸡蛋的问题。不管其是否是一个 Web 服务，只要整个公司运用相同的方法和技术，就会给标准化带来最大的好处。但是这些企业级的项目同样也为实施 ROI 耗费了许多时间，因为它们中间的许多部分是如此巨大和一体化;许多客户正是为此原因在犹豫是否采用它。

因为这些原因，“通过它来衡量回报是一个相对新颖的概念。”在 Stamford, Conn.-based Gartner Inc 研究保险业的分析家 Kim Harris 说到。这些公司中超过 70%的公司已经开始运用 XML 作为通过 Web 进行信息交换的工具——大多数是用来帮助标准化被客户和其他人充塞的窗体的过量——并且也正期待着与 Web 服务的合作。“许多好处都是理论上的。”Harris 谈到，但是包含一些类似客户服务，增长的连通性过程以及增加的 IT 生产

力和生产效率等除外。“我从我的数据库得道信息的速度越快，我能够做的工作就越多。”她解释到。

大多数关于采用 Web 服务处理数据的 ROI 的信息都是因为节省了开支，而不是因为产生了新的收入，Harris 和 Bloomberg 解释到。客户节省了开支是因为他们不需要购买第三方的中间件，这些第三方的中间件以前是需要和应用程序连接在一起的。

另外一些节省开支的原因是因为能够使用已经存在的内部的专业技术人员，包括 Java 和 Visual Basic 开发者，来生成基于集成接口的 Web 服务。

尤其是集成项目能够很快地产生回报，因为 Web 服务相比其它那些集成技术而言不是那么负责。绝大多数开发者能够快速地接受这个概念——包装组件，显示它们，然后使用 SOAP 命令去连接它们。在这些图例中它不需要一个大的骤变来成功使用 Web 服务。

向外部合作者或者客户通过 Web 服务发布信息是潜在使用快速 ROI 的另外一个领域。Amazon 和 eBay 以及其它一些应用提供了基于 Web 服务标准的应用程序编程接口，这些接口甚至允许非常小的商务能够连接到 Amazon 或者 eBay 平台上去卖它们的产品。

当然，实际的 ROI 时间段取决于具体的 Web 服务的实现方式，甚至在这些快速发展的工程中也不例外。即使是“基本”的集成也能够变得非常复杂。

两类基本的集成项目

根据专家介绍，存在两类基本的集成项目。通过数据传输，一个 Web 服务得到一个用户的帐户号码，比如说，这个号码能够被一些不同的应用程序所利用。

第二类更复杂一些，一个 Web 服务包含其它一些服务，并且结果随着掌握的数据动态变化。让我们考虑一个用户在一个 Web 站点上发出了一个命令。在这个命令执行完成之

前，系统向数据库(实际上是一个 Web 服务)发出一个请求。这个请求通过验证信用卡用户的数据库登录，从而确保信用卡不会被盗或者混淆。

当第一种简单集成带来一个快很多的回报时期，第二种类型的 ROI 需要花费更多的时间因为它包含更多动态的部分。

另外一个关于计算 Web 服务 ROI 的问题是“事实上我们没有非常好的花费预算。”Gartner 的 Harris 说到，“不同的销售商会有不同的需求，并且我到目前为止还没有发现真正满足这种情况的方式。”如果没有一个好的对于需要的费用预算的理解，ROI 是不可能精确地确定下来的，尤其是对于那些更大更复杂类型的 Web 服务项目来说。

ZapThink 的 Bloomberg 谈到，将来，Web 服务中心的财政状况将成为怎样管理 SOA 的一部分。为了成功地实施 SOA，IT 人员将不得不学习怎样管理和跨越传统的边界，这些边界是商业领域和 IT 组织根据操作系统或者硬件平台强加的。SOA 中一个主要的变化就是穿过了这些边界并且以提供给客户或者最终用户的服务的角度考虑问题。

“重新将 IT 组织成真正的面向对象确实是 SOA 中很难的一部分”，Bloomberg 说到。财政问题和 ROI 问题都是 SOA 管理中需要考虑的问题，并且“解决他们没有快速的办法”，他解释到说，“你需要建立企业级的策略，并且基本上每个公司的每个商务行为都得进行分别处理。”

与此同时，一个为任何人设计的建立在意味着外在消费的 Web 服务之上的 ROI 策略旨在和合作者来往愉快。“通过建立良好关系赢得增长，”有着技术咨询顾问资历的西雅图分析家 Richard Stevens 说到。这包含了“在你的客户中建立信任关系”，他谈到，“如果没有这些信任关系，即使是最有技术性的创新计划在执行的时候也会遭遇失败。”

(作者：Johanna Ambrosio 来源：TechTarget 中国)

如何使用 ROI 推销 SOA 和 Web 服务项目

因为技术回报具有长期性这一事实而向 CFO 要求为技术项目开一张空白支票的日子早已远去了。现在，技术必须要实现赢利，无论该技术是应用于 Web 服务和面向服务架构（SOA）项目，还是应用于其他的项目。

投资的回报（ROI）计算，通用于 IT 运营界，现已逐步应用于 SOA 领域。在上一个专栏中，我列出了能够期望实现 ROI 的领域，以及一些成效显著的项目案例。在这个专栏中，我将就如何使用 ROI 推销项目提出建议。

从何处开始

如果想知道如何使用 ROI 推销项目，最好的办法是去求助专业人士——那些实际上已经身处前线而不得不一再进行项目销售的人。因此，为了获得建议，我与 Con-Way 运输服务公司的企业建筑师 Maja Tibbling 进行了讨论。这是一家资产达 26 亿美元的运输物流公司，经营范围包括美国，加拿大和墨西哥。Tibbling 一直是使用 SOA 和 Web 服务的先驱者，而且她在衡量 ROI 方面经验丰富。

Tibbling 说，从 Web 服务或 SOA 中可以期望获得两种类型的 ROI——IT ROI 和商业价值 ROI。IT ROI 是一个我们熟悉的概念。它是指由于组件复用及降低开发费用的能力创造了更大的生产力，IT 部门从而能够得到 ROI。定性 ROI 的获得，来自之前不可能实现的服务及产品的开发能力，以及将这些服务和产品更快的投入市场的能力。通常情况下，定性 ROI 总是远远大于 IT ROI，她说。

Tibbling 说，最初，你很可能不会得到 IT ROI。事实上，你甚至可能会发现有额外费用发生，这是因为一项新技术的使用需要新工具、新的培训及新的工作人员。但是随着时间的推移，IT ROI 会相当大，但这通常要耗费几年的时间，因为需要建立可重用组件及模块的资料库。

正因为如此，在最初几年递交项目 ROI 评价时，要注意任何有增长潜力的 IT 方面的 ROI。事实上，她说，最好要老老实实的承认初期 IT 成本可能会增加，虽然商业方面的 ROI 会弥补这些增加的成本。

“定性 ROI 的好处体现在寻找 ROI 的必要性上，”她说。“如果你尝试在 IT 资金积累中对 ROI 进行成本验证，那么，你只不过是在编数据罢了，因为 SOA 初期会比较昂贵”。

这意味着，她说，要比通常情况下对项目的业务方面有更为深入的研究——这可能意味着考量 IT 工作人员能力的一种新的方式。

“IT 人员必须要真正了解业务，这一点非常重要，”她说。“他们不能只精通技术；他们还必须要了解业务”。

长远的考虑

Tibbling 补充说，当介绍项目时，“你还需要明确表示，将不会有任何意外发生。你必须要解释，项目初期可能会有额外开支，但从长远来看它会使你得益”。

要做到这一点，她说，应避免提出一个短期的商业案例。而是要提供一个长期的商业案例和长期的 ROI——至少两年，最多五年。在这个项目的第一年，ROI 可能会出现负值，因为只有可复用的模块出现时，ROI 才会增加。所以，你提出的项目计划周期越长越好。这是 IT 人员需要更多的了解业务的又一个原因，他们还需要了解业务的进展方向—

—如果他们要提出一项长期的业务计划和长期的 ROI，他们就必须要了解业务计划在未来的两到五年内将是个什么样子。

她还建议要寻求降低开发成本的方法，尤其是“注意设计时不要太过分求精。不要认为设计时需要将整个项目分析透彻。”

为了实现这一目标，她建议要利用一切能利用的，而不是试图一切靠自己，从零开始。你应该利用现有的业务逻辑和遗留系统，并将其合并为服务。这样做将减少初始成本，有利于项目的销售。另外，这将确保项目不会变得太复杂而无法实施。

最重要的几点

提供 ROI 时最重要的是什么？一定要老老实实承认，成本最初可能会增加而不是减少。要提供一个长期的业务计划和长期的 ROI，因为在项目的后几年资金积累才会增加，而不是最初几年。确保工作人员在业务和技术方面都非常精通，从而可以确保对项目中的商业 ROI 和 IT ROI 都非常重视。

做到所有这些，Tibbling 说，你就会顺利获得项目的签订。

（作者：Preston Gralla 译者：Eric 来源：TechTarget 中国）