



SOA 与 IT 治理

SOA 与 IT 治理

面向服务架构 (Service-Oriented Architecture, SOA) 提供了一种方法, 可以把企业的业务战略和必要项目与 IT 项目结合起来。因而, SOA 治理不但涉及技术, 同样涉及组织问题以及人员如何协同工作来实现业务目标。

IT 治理涉及获得进行变更的批准, 涉及权力, 涉及谁拥有这种决策权, 涉及在业务迅速变化的形势下这类决策需要多长时间。SOA 通过减少决策的发生, 从而大大减少了 IT 治理的需要。

治理意味着授权。它提供一个策略和最佳实践的框架, 可以用这个框架定义谁有权做出何种类型的 IT 决策。它还能指定应对这些决策负责的人员。很多分析人员已经清晰地划定了治理和管理之间的区别, 而重申这一区别是十分重要的。治理与具体的 IT 决策无关; 它会决定有能力做出这些决策的人员所充当的角色。管理则通过治理指导原则获得授权, 并做出具体的 IT 决策。

什么是 SOA 治理

“SOA 治理”包括什么? 这包括了在 IT 层面的治理观念, 还有针对 SOA 的治理框架。从 IT 的角度来看, 需要有一个能管理和跟踪整个 SOA 应用的生命周期的机制, 也就是一个能管理从发现业务需求, 到转变业务需求为 IT 实现, 并在运作中跟踪业务需求最终价值的实现的机制。然后是运用什么样的管理政策和系统, 来监控管理这些决定企业业务能否成功的因素, 来确保企业在所有的环节都在进行着这些环节所应该做的事情。建立连锁责任关系的状态。

- ❖ SOA 治理和基础设施管理
- ❖ SOA 治理 关注安全驱动 XACML 协作性

SOA 与 IT 治理区别与联系

人们经常把 IT 治理和 SOA 治理混为一谈，并且认定在不久的将来这二者就会统一起来。不过，EDS 网站上的一篇新的一位业内评论家指出，事实上，IT 治理和 SOA 治理基本没有什么共通之处。IT 治理主要侧重于管理技术，而 SOA 治理管理则是关于管理商业服务。

在企业里，所有的业务单元都是 SOA 的拥有者，但管理和政策执行则移交给治理委员会，治理委员会主要负责服务的维护和发布的情况。从理论上讲，IT 部门可能只是其中的“拥有者”之一。

- ❖ 区分 SOA 治理与 Web 服务管理
- ❖ 解读 SOA 和传统 IT 管理之间的必要联系
- ❖ 治理、质量和管理:让松散耦合运行起来

SOA 与 IT 治理的重要性

当前，IT 是企业中最为常见的组织，在大多数(如果不是全部的话)业务线(LOB)中都存在。对企业发展和成功如此重要的组织必定被企业视为重要资产之一。企业必须充分地了解这样的重要资产，不仅要从中获得最大的利益，而且要正确地加以管理，从而减少与之相关联的风险。这使得治理主体有必要规划、控制和监视 IT 治理所需的企业资产的正确维护和发展。

SOA 的概念在 IT 行业已经存在很长时间了，但是直到最近才作为使商业策略和企业要求保持一致(通过其 IT 计划)的一种方式而受到关注。使企业采用需要更加重视治理的 SOA 的原因在于服务跨各种 LOB 分布的这一特性。

- ❖ **Gartner: SOA 治理依然至关重要**
- ❖ **Dun & Bradstreet 公司高度评价 SOA 治理**

SOA 与 IT 治理实践

在实践中，SOA 治理指导可重用资产的开发，确立如何设计和开发服务，以及这些服务如何随时间增长进行更改。它将在服务提供者和服务使用者之间建立一个协议，告知使用者可以希望得到什么功能，告知提供者应该提供什么功能。SOA 治理并不设计服务，而是指导将如何设计服务。

SOA 治理以现有 IT 治理技术和实践为基础。使用 Java 2 Platform, Enterprise Edition (J2EE) 等面向对象的技术时，IT 治理的一个关键方面就是代码重用。代码重用也体现了 IT 治理的难点所在。每个人都认为可重用资产很好，但实际工作时却非常困难。

- ❖ **对服务进行合理精简**
- ❖ **数量并不是 SOA 成熟度计量标准**
- ❖ **SOA 最差实践:过多的 Web 服务=麻烦**

SOA 治理和基础设施管理



Dana Gardner 是一位在软件生产率趋势和新的 IT 商业价值机会方面知名的鉴定师。他从 1999 年任 IT 产业分析师开始，就不断磨练他的技能，完善他的见解。

Gardner 因其企业软件解决方案、战略、伙伴关系和市场方面是一位具有创造性思维的领导者而人所共知。作为一名熟练借助多媒体的传播者和布道者，他写了几十篇行业报告，致力研究 IT 及互联网创新的商业利益，以推进一般生产力，提高员工工作效率，以及降低 IT 总成本。作为 Interarbor Solutions 公司的创始人和总裁，Gardner 在为 IT 厂商、运营商及企业提供服务咨询方面，将记录提高至一新的水平：互联网社交媒体领域中惊人的新型通讯功能。

问：在 SOA 治理方面，我们的基础设施能起到什么作用？

答：区分 SOA 治理和性能管理非常重要。SOA 治理可以帮助定义服务能够规定和使用的最佳方式并规定如何为编制和使用这些服务并为复合的业务流程提供规则。但是所有基础设施统治了需要管理的 SOA 活动和需要维护的性能，所以治理是如何和基础设施相联系的呢？这是设计时间和运行时间反馈漏失的联系。当 SOA 不断上涨时这些漏失的联系尤为重要。因此需要有效的使用资源支持 SOA 活动。能量和 IT 劳动力不会越来越便宜。

现在，我想问这样一个问题：在管理基础设施方面，SOA 治理起到了什么作用？你能洞察到是什么样的 SOA 治理让基础设施的需求转移得以实现的？下面的例子就是关于 SOA 如何对你的机构产生了如此深远、churen 意料的影响。许多人并不相关(就像网络 admins 公司)必须合作并且在如何影响其它公司方面分享知识。

操作 SOA 治理和基础设施管理者之间的相互作用可能某一天会变得更加自动化。但是我会等到真要讨论它们并要求人工调整的时再考虑，到那时可能会建立新的社会网络设备，所以这些公司能够更容易的找到彼此并且互相帮助——依照严格的人性化字眼。

(作者: Dana Gardner 译者: 杨君 来源: TechTarget 中国)

SOA 治理 关注安全驱动 XACML 协作性

在面向服务架构（SOA）领域中居于领导地位的供应商都在试图证明 Extensible Access Control Markup Language（XACML）2.0 与他们的 Web 服务软件产品的协作性……

SOA 的政策要求其一是要促进 XACML，这个有两年历史的 OASIS 标准能够在供应商和他们的客户的安全和治理方案中得到充分应用，BEA 公司的中层工程首席技术官员，OASIS XACML 技术委员会的联合主席 Hal Lockhart 说道。Lockhart 和 Oracle 身份管理/标准架构技术部负责人 Rich Levinson，一起与其他的供应商合作解决 XACML 协作性的问题。上月月末他们展示了一个为即将在本月举行的 Burton Group Inc.'s Catalyst Conference 设计的 XACML 协作性示范。

分析企业为竞争的供应商提供了一个中立的平台来展示 XACML 的协作性，Burton 的副总裁兼服务主管 Gerry Gebel 解释道，主持了视频会议的预演。Burton 也将提供为 Microsoft and the Liberty Alliance 提供一个论坛以开始他们的竞争性的以用户为中心的一致管理技术的协作努力。

“能看见竞争者为共同的目标而工作很高兴，” Gebel 说到，解释说 Burton 相信协作是那些网络技术和标准达到成熟的关键。

在讨论 SOA 的过程中 XACML 不是第一个进入人们思想的标准，但是 Lockhart 说 SOA 及软件即服务的兴起 (SaaS) 是驱动协同示范的包括 BEA、IBM 及 Red Hat 在内的主要因素。

“存在三种趋势，这些趋势都驱动了对于细粒存取控制的需求。”他解释道。“SOA 只是其中之一，因为与从许多你已经拥有的集成块来建设你的系统相反，许多独立的服务和每一项服务都将需要执行它自己的存取控制。既然它将被其他服务以不同的方式加以运用，较之于传统环境，政策要求将肯定变得更复杂。”

其他两种趋势是与商业伙伴和客户间越过防火墙的 Web 服务交换数量的增加及 SaaS 的上升，Lockhart 说道。“我认为你在看到软件作为一项服务的同时将看到对于细粒控制存取的需求，因为你将集成那希望在你的组织中由一个或更多个第三方提供的组件，”他说道。“细粒控制存取将被考虑纳入商业关系合同中。”

细粒控制存取是不同于 XACML，器不同之处体现在 Web 服务交互的两个配套标准上，Security Assertion Markup Language (SAML) 和 WS-Security Policy，当被问到这三个标准间的关系时，Lockhart 解释到。

当被询问到 XACML 和 WS-Security Policy 的不同之处时，他回答道，“如果你将 WS-Security Policy 看作其最大的目的在于允许一个服务发布或使那些存在存取路径的需求可获得。然而在 XACML 中，目的在于去定一个给定的存取是否应该被允许。那些可能听起来很相似，但实际上在意图上的不同导致他们在设计方式上的差异。WS-Security Policy 提供较少的细节，并以一个形式来进行设计以使其能匹配。比如说，一个请求能匹配该服务所要求的它能够与之协作的。这些就是配套的目的。”

对于更高安全和隐私的需求将使 XACML 和 WS-Security Policy 之间以后拥有更加紧密的联系，Lockhart 预测到。

“存在一些因为隐私的缘故而对在他们所倡导的政策中加入更好的细节感兴趣。”他说道。“所以我们致力于总体上的 XACML 3.0，这将是你将一个 XACML 政策和 WS-Security Policy 配套以提供更多有关的细节，比如说，隐私考虑。

当被问及 XACML 和 SAML 声明间的关系时，Lockhart 回答到。“因为 XACML 的目的在于利用任何种类的信息以做出决策。SAML 是一个以技术术语或主题提供用户相关信息的非常好的来源。所以举例来说，SAML 主张告知一个特定的团体在一个特定的时间以一种特定的方式登录。还有 SAML 有一个不同的声明，使你能说出一个特定团体所拥有的特定属性。所有这种信息都可以用于 XACML 政策决策的输入。”

协作性将使用假定的金融服务股票交易应用，将特别专注于 XACML 的能力及参加的供应商的产品是如何在异制的环境中一起工作的，Oracle 的 Levinson 说道。

“我们将论证一个产品创造的政策可被另一个产品消耗，”他解释道。“协作的第一个目的就是论证 XACML 的最初的功能能力，它使来自于应用的登录授权能够抽象化，所以你能从中心点进行授权。从商业的角度来看中心点的价值，在政策决策点上，你能以通常、无缝的方式制定你所有的关键商业规则。它为企业定义企业政策提供了一个焦点。”

Levinson 说供应商的协作性可通过遵循在 OASIS 标准中制定的通常的政策规格语言和运用常用的具体应用的词汇表来达到，现在这些都是供应商所认同的。

(作者: Rich Seeley 来源: TechTarget 中国)

区分 SOA 治理与 Web 服务管理



Paul Lipton, Computer Associates 高级建筑师 Paul Lipton 是 Web 服务和应用程序管理组在 Computer Associates (CA) 机构的高级建筑师和联系与该办事处的 CTO。他是一位在开发商和企业制度有 20 多年的建筑师，并与主要的核证机关的顾客密切合作，以解决重要的商业挑战，通过建立管理，关键任务分布式的解决方案。Paul 代表了 CA 在众多的标准组织，如 W3C, OASIS 和 Java Community Process。他是一个非常受欢迎的作者和会议发言者，并分享他的知识与欣赏世界各地的听众对这些不同的主题，作为 Web 服务，SOA，管理和安全，Java，.NET，EAI, 分布式计算系统和有效处理。

问：对于 SOA 和 Web 服务的发展已经不再仅仅局限于技术或协议规范方面的更新和成熟，更多的时候，管理已经在这个领域扮演更重要的角色。只是，对于 SOA 管理和 SOA 治理是怎样去区分理解的？那 SOA 治理和 Web 服务管理又该如何理解？

答：这些日子以来，Web 服务管理正逐渐演化成一个更广泛的主题，称为 SOA 管理。SOA 管理产品很多，但是他们典型地扩展了传统的 Web 服务管理从而包括附加的功能，这些功能在一个 SOA 中——例如监视和管理基于商业消息内容的服务，是很有作用的。因为这个原因，我已经稍微修改了这个问题，以至于讨论的 SOA 的两个方面，管理和执行，都属于同一类型的实体。因此，让我们假设问题是：“SOA 治理与 SOA 管理的区别是什么？”

这个问题的简单答案是 SOA 治理基本上是专注于开发过程，然而 SOA 管理主要关注 SOA 运行时方面。此管理系统必须保证 SOA 上的业务服务保持可靠性，以及不辜负商业合作者，客户，提供商，内部用户以及管理者的期望。

尽管新的 Web 服务管理标准很重要，例如 OASIS WSDM 将逐渐促进 Web 服务以及 SOA 管理的艺术，以令人激动的新方法，最重要的管理产品今天将提供两种基本的方法。它们可能提供不同的代理服务，这些服务监控以及控制服务间的消息通信。或者它们提供代理来为 Web 服务平台提供工具来模拟类似的功能。一些管理系统可以提供这两种功能。第一种方法更趋于灵活，但是提供更多的故障点。第二种方法趋向于更高效因为它作用于 XML 过程，其已经在 Web 服务平台上得到了实现，但是需要在 Web 服务平台上安装一个代理。

什么呢?你是否称它们为 ESB(企业服务总线)或者你更喜欢另一个不同的名称，Web 服务平台是构建和部署 Web 服务的基础了。这些平台展现习惯或者打包好的好像 Web 服务的应用程序的功能。作为产品，他们大多数经常在应用程序服务器以及消息中间件上有自己的根。

我认为，为了更高效地管理它们的 SOA，企业构建师们应该懂得此管理系统必须保持可见性以及在一个高于任何单个 Web 服务平台水平的概念和架构上控制。这是因为 Web 服务平台趋向于操作，好像他们是世界的中心一样，唯一需要来开发和扩展的平台。但是在真实的世界里，一个 SOA 似乎包含许多 Web 服务平台，这些平台因为他们独特的能力和特征而被选中。当每个平台能够为自己提供一些基本的管理功能的时候，在所有的 Web 服务平台上仍然需要提供一个更高层次的业务为中心的观点。这是一个高效的 SOA 管理的一个重要的需求。

正如我前面所说的，SOA 治理主要关注于跟踪和控制开发的典型产物。他们必须以一种相当平和的方式集成自己到开发方法中，同时监督整个开发过程包括人力方面，例如人力经理批准把一些东西从一个平台(例如测试)移植到另外一个平台(例如产品)，控制过程并且甚至产品的可见性，以及更多。

治理产品也必须集成他们自己到开发者使用的工具中去，从而使得这些被创制的或者部署的产品在所有开发平台上都符合企业要求，以及符合工程标准。他们经常必须为多种 Web 服务平台做这些事情，以及在多种地点例如数据库，多种独立的开发项目跟踪产品的地址，改变系统配置，管理/安全仓库等等。一些治理系统也提供他们自己的仓库允许开

发产品存储其中。这个在 IT 技术所有水平阶段提供治理并且不只是在 Web 服务水平提供的能力实质上非常重要。想想所有那些 CICS 主机应用程序以及老式的在 UNIX 系统上运行 C 程序，举例来说。

底线是管理和治理能够合成到一起，就像豌豆在豆荚里面一样。越来越多地，它们将一起协作以在开发的整个生命周期以及一个 SOA 生产部署中形成一个完整的链接，以实施和共享企业以及商务策略的方式。例如，管理系统能够帮助治理产品使得开发者知道决策，以至于应用程序以及服务基于历史数据以及管理系统能够提供给他们的管理策略。同样地，治理产品能够提供更多的信息给管理系统关于编码和接口上的变化，从而使得管理系统能够实施更多更有效的事件合作以及问题的决策。

(作者: Paul Lipton 来源: TechTarget 中国)

解读 SOA 和传统 IT 管理之间的必要联系

在硬件和软件相对封闭的时期，人们可能通过把基础系统管理程序写在计算机房间的黄色便签纸上从而实现对整个系统的管理流程。但是在面向服务体系结构(SOA)时代，这种方法已经完全被废弃了，Ovum 咨询顾问分析公司的副总裁 Mary Johnston Turner 如是说。

能有效解决所有涉及维护 SOA 应用程序的 IT 管理问题的方法之一，则是 IT 基础设施库架构(ITIL)，她说。与 SOA 相比，ITIL 并不是一个全新的事务。从 20 世纪 80 年代开始英国中央计算机与电信局(UK Central Computer and Telecommunications Agency，CCTA)就致力于研究最佳实践方法及相关流程以确保 IT 提供的相关服务能达到企业所要求的水平。

“ITIL 是无所有权的，” Hewlett-Packard Corp 公司 IT 服务管理教育部经理 Ken Hamilton 说，他建议在 ITIL 的基础上去实施 SOA。“有很多介绍这方面知识的公共书籍。”

由于很多 IT 专业人员没有这些书籍，Hamilton 给出了由 UK Office of Government Commerce 和 HP's online library of ITIL materials 提供的关于 ITIL 的定义及相关信息。同时，他还拿出一份 Turner 写的 Ovum 会议报告，内容则是关于为了保证 SOA 的成功需要提高 IT 管理水平。

在定义管理架构时，Turner 说，“ITIL 是目前业界普遍采用的一系列 IT 服务管理的实际标准及最佳实践指南，包含着如何管理 IT 基础设施的流程描述；它以流程为导向、以客户为中心，通过整合 IT 服务与企业业务，提高企业的 IT 服务提供和服务支持的能力和

水平。ITIL 包括变更管理、结构管理、容量管理、服务级别管理、成绩管理、可用性管理。”

在开发这些 ITIL (IT Infrastructure Library IT 基础设施库) 指导方针之前，她说，并没有在需要复合使用多种应用程序的本质的任务上取得共识。

Hamilton 说尽管 ITIL 产生已经有二十多年，但是他也只是在近三、四年才发现世界财富 500 强公司采用 ITIL，例如通用汽车。Turner 说在 SOA 取代封闭式之前，ITIL 所包含的细节级别并不是必需的。

“在封闭式环境中，哪里是服务器、哪里是中间设备、哪里是应用程序、哪里是数据库，这些都不是难题，所有这些软硬件都紧紧连接在一个位置。”她说。“如果其中一个软硬件出现问题，你很快会发现，知道这个问题会影响到那些业务环节。业务环节与应用程序基本对应。现在，到了 SOA 世界，你会将很多软件、服务放置在不同的服务器，不同的地点，在需要什么软件或服务时在进行联系。”

在 SOA 世界，IT 经理要想了解软硬件问题出现后会发生什么或问题的最坏影响是件非常困难的事。

“现在你明白这是一个非常有意思的问题，” Turner 说。“我怎样才能了解端到端的业务流程是如何工作？还有如果该业务流程不能工作，我怎样才能发现到底是哪一个相互连接的工作件出现了问题。”

举一个例子，她说譬如你的业务流程出现了一个问题，是由于另一个 IT 专业人员修改了一个应用程序的配置，而他却没有通知你。这就是像 ITIL 这样细节化的架构，通过规定任何人不可以在没有通知其它人的情况下擅自更改配置，能够帮忙避免的问题。

Hamilton 说他已经看到 ITIL 与 SOA 结合的趋势，但是他说这样的结合还处于萌芽阶段，以致于他还找不到一个二者结合的用户。

Turner 说在她调查的 IT 组织中或许有三分之一的组织采用 ITIL 或其它架构，而那些使用 ITIL 管理 SOA 的组织比例仍然非常低。

不过尽管目前接收 ITIL 管理 SOA 的企业数量仍然很少，但是 Turner 说，如果 IT 组织希望成功执行 SOA，那么这些组织将不得不在适当的位置配置基于 ITIL 或其它质量架构 (例如 6 Sigma) 的详细程序。

“越是依赖于 SOA，对传统 IT 管理的压力也就越大，”她说。“如果你不能以通知所有人你完成了自己工作的方式对将要递送的复合服务达成协议，或者你不知道如何以此衡量端对端性能，这将是噩梦。”

由于在采用 SOA 之前的 IT 管理程序似乎不能满足需求，所以需要一种更为细节化的最佳实践方法架构。

“传统模型基于封闭式环境，而 SOA 是封闭式环境的对立面”她说。

同样，尽管传统模型关注软硬件在封闭式环境中的表现，但像 ITIL 这样的框架仍然涵盖了 IT 员工们如何工作，包括改变软硬件以及设置优先级来确定什么时间做什么工作。

“你或许还有在自己小范围内工作的封闭式专家，但他们必须做出更大的文章，”Turner 说在 SOA 环境中，“他们必须懂得在他们自己领域内的变化对其他区域产生的影响。还要考虑什么时间做什么工作的优先级。你的队列中的下一个任务并不一定是恰好要工作的，任务是否工作取决于这个任务的影响。”

Turner 说广泛采用 SOA 是推广像 ITIL 这样管理框架的驱动力，但是没有 CIO 的支持这也很难成功。如服务台人员这样的群组，开始尝试最优的业务实践，如果他们由一个 C 级主管来领导的话成功机会不大。

(作者: Rich Seeley 来源: TechTarget 中国)

治理、质量和管理:让松散耦合运行起来

随着面向服务架构(SOA)按照 IT 的一般方法很好的得以建立，公司正在专注一些为使 SOA 运行而进行的具有挑战性的关键事务。那些在 SOA 采纳路径上一路走好的企业已经充分地懂得，建立和开发服务的技术挑战远没有建立松散耦合的障碍显著，与业务相关的服务利用持续改变的业务流程。实际上，建立松散耦合的挑战只比与组织的 SOA 采纳相关联挑战稍微弱一些。

当 SOA 抽象与异质性、点到点集成及紧密耦合应用逻辑相联系的复杂性，它引入了另一种不同的复杂性：分布式、松散耦合及动态组成服务的管理。在过去的几年中，企业开发了许多处理这种新的复杂形式的方法：隔离失败和提供抽象终端差异性的机制的管理方案，提供保证变化在显著变化的环境中可传播性的机制的质量方案，及提供对面向服务的系统开发的整体视图的治理方法，改革和版本管理问题的缓解，及对于业务作为一个整体运转的核心的执行政策。

ZapThink 已经对 SOA 治理、质量和管理的那些备受议论的领域撰写过许多文章了，但直到最近，我们才认识到那些不同的市场区隔以他们自己的权利成为了同一个问题的不同解决方案：使松散耦合的幻想成为现实。在组合上，SOA 治理、质量及管理形成一个完美的三连击，其能使所在一个持续变化的 IT 和业务环境中所预见到的松散耦合困难变为现实。

运行质量和管理之间的联系

这个三连击的一部分就是运行 SOA 质量和 SOA 管理的联系。维持高质量、面向服务的系统的挑战在 Quality SOA ZapFlash 已经有详细的介绍。总之，如果一项服务在隔离的

环境下工作，那么如果他是如何组成的方面或元数据作为一个整体对系统性能影响的一些变化。大体上，单元测试单个服务实现完全不足以决定该服务实际上在元数据控制的服务中组合环境中是否完成任务。

有效保证 SOA 质量的唯一方法就是持续地这样做，衡量质量不仅以一个离散的、原子服务在生产中持续进行测量，还要测量所有相关的元数据、组合逻辑、政策及潜在计划。大多数企业对与在生产重，持续测试他们系统的思想不熟悉或是说不适应。尽管如此，就像我们在大量的 ZapFlashes 如抓住 SOA 变革和版本管理 和 SOA 质量和治理：满足敏捷的元需求所阐述的，架构就是业务，而且因为业务持续在变，一个质量保证 (QA) 模型要求环境的复制以保证质量，而这将是异常昂贵的，不可能进行管理，而且是无效率的。

使运行质量在生产中进行实际上的工作需要一个机制，以隔离来自于回归和不可预测影响的失败。通过实现作为服务合约和政策的测试模型来达到这个要求，同时在一个“测试模型”中最小化服务的单边效应以避免数据和行为的任何不必要的委托。政策驱动测试的想法和质量的运行时间执行覆盖了那些运行时间服务管理。许多 SOA 管理解决方案提供政策执行、异常管理、错误恢复、及根源分析。运行时间质量工具的性能配对通过最小化那些变革的影响的解决方案便利了翻译的过程是恰到好处的。

进一步来说，这里有一个“管理——质量反馈路径”存在于工具方法之间用于管理，当系统接近一个不理想的状态其将提供可视化，而且其提供允许将系统作为一个整体的增量测试和质量管理的机制。这个反馈通过确保任何服务相关的变化不会打破任何东西——松散耦合的一个基本需求，帮助保证松散耦合。

SOA 管理和治理之间的联系

同样地，在 SOA 治理和管理之间也存在一个联系，其便利了松散耦合。SOA 治理有三个截然不同的，但是也是相关的部分：设计时间治理，其提供创造、开发及消耗服务的规则和政策；运行时间治理，其管理在作为一个整体的架构的生产和性能中服务的行为；及改

变时间治理，其详细说明企业如何在对现存的业务和政策所带来的扰动最小的情况下在整个系统实现变革。

SOA 管理产品专注于政策执行、基于规则的路由及决策制定及异常处理的方面。如上所述，由治理工具管理的政策能通过活跃的 SOA 管理工具在运行时间执行。当 SOA 治理工具如服务于系统记录的注册机和知识库去管理与服务相关的元数据时，SOA 管理方法提供了确保服务消耗于生产中，并与在那些系统中建立的政策相符的方法。SOA 管理工具还能发现并防止无赖服务的发生，并检查服务互动性以使 SOA 治理方法的值得以体现。加之，运行 SOA 管理通过执行在分布式环境中行政的和赞成的变革帮助了变化时间治理问题，同时最小化了质量和性能问题。

进一步，有效的 SOA 治理要求有效的管理提供在运行系统中所要求的可视化，以反馈治理流程。这个“管理——治理反馈回路”是确保在不过分限制架构的敏捷性的情况下不仅企业能治理他们整个的 SOA 项目，而且提供有效的控制和管理的一部分。这个反馈路径通过确保业务需求的变化对整个系统的行为不产生不利影响及因其已经存在于该点从实现上进行退耦业务逻辑协助保证了松散耦合。

SOA 治理和质量的联系

上述讨论所暗含的想法就是改变时间管理和治理。为了防止 SOA Butterfly Effect，企业不得不有效的管理和治理变革，所以变化将不会对服务的复杂环境产生混乱的影响。这意味着不仅要通过在运行时间上的管理方法抓住和防止失败，还要通过运行时间质量和测试方法分段实验和测试那些变化。

设计时间治理也要求企业在不受控制的服务渗透到 IT 环境之前执行服务开发实践。治理和质量方法之间的有效集成通过提供对与已建立的政策和方法之间的分离的可视化使该执行成为可能，以限制那些不相适应的服务的扩散速度。

在一个持续变化的系统中维持持续的质量的挑战是维护有效治理的方面，比如说 SOA 治理和质量工具和方法的联合以使该为题更便于管理。SOA 质量工具通过指示政策和变化如何影响整个系统提供对于治理系统的反馈，同样地，治理系统和方法通过提供在设计和运行时间上影响服务的持续变化约束喂养质量生命周期。这个“质量——治理反馈路径”通过确保任何设计时间变化对运行系统没有任何影响和在他们的服务客户和供应者中服务客户不需要艰辛地编程治理规则促进了松散耦合的实现。

SOA GQM 套装的出现

企业投入了太多注意力到单个的服务开发、运行时间执行环境及作为他们的 SOA 基础设施的内部服务通讯。实际上，处理服务基础设施的运行时间方面是做 SOA 工作的一个相当琐碎的部分。实际上，大多数企业的基础设施的种类是为实现他们的面向对象的项目，只能较容易地在旧的集成中间件之上创造紧密耦合的网络服务 APIs，因为他们能便利松散耦合、组成服务的建立。既然企业在寻找后者而不是前者作为 SOA 存在的理由，企业投注大量的精力在 SOA 的治理——质量——管理方面是势在必行的，而不是如何开发一个服务结构和运行一个服务实现。这并不是说 SOA 基础设施，如 Enterprise Service Bus (ESB) 的努力是不必要的，但是不仅仅是这样的！企业想实现他们的 SOA 对于在一个持续变化的环境中保证松散耦合和组成是不够的。

所以，企业应该专注于 SOA 的种类——作为 SOA 目标核心的具体实现技术：SOA 治理质量管理 (GQM) 套装。该套装作为单个供应商产品解决方案也许没有实现的必要，即使它看上去市场也许会以这种方式趋同。与它是否作为单个供应商解决方案还是最佳应用的集合而实现无关，SOA GQM 套装是便利于上述在这个 ZapFlash 中所提到的三个反馈路径工具和性能的集合，说明如下：

ZapThink 已经开始在成功的 SOA 项目中发现这个完全的反馈路径变为现实。现在，企业应该通过确保所有反馈路径的实现以专注于实现 GQM 套装的每一个部分。不管是通过自

已开发的解决方案，还是最佳的供应商的解决方案，或是致力于提供产品的整个套装的打包的产品，企业需要适当地表述他们的 GQM 需求，如果他们希望实现 SOA 的利益。

ZapThink 所持的观点

对于松散耦合、组合、面向业务的服务以在一个异质和持续变化的环境中实现一个平缓的成本变化真正感兴趣的企业将获得很好的服务以专注于他们的项目的具体技术部分而不是单个的 ESB 以作为他们的 SOA 成功的关键，而不是在 GQM 方面。说明 GQM 反馈路径的所有部分将对有利于将松散耦合的幻想变为现实。加之，使 GQM 空间更有吸引力的是它代表了新的技术投资，而大多数企业还没有做过。

当今天的许多 ESB 投资是对已有的已经处于适当位置的基础设施和中间件老调重弹，但同样的话不能用来形容 GQM 工具。在之前的 IT 简化的架构和技术演进中投资 GQM 在一个紧密耦合、私有系统的环境下是没有必要的，而且缺乏组合系统。在这样的环境下，测试是一个需要完成 QA 活动，只存在与开发和配置之间，只有当系统配置时才需要管理已完成的一些事物，总之，接着只用一个监控能力，及管理一个事后想法。尽管如此，SOA 代表的松散耦合和组成的吸引价值在于给予企业不仅仅是使自身架构优化的机会，还有在他们的系统中做出正确的投资以允许他们可能应该在此时拥有的治理、质量及管理种类。

(作者: Ronald Schmelzer 来源: TechTarget 中国)

Gartner: SOA 治理依然至关重要



Paolo Malinverno, Gartner 研究副总暂时停止了在 Gartner 应用架构上的工作，出席 2007 发展与集成高层会议，讨论了 SOA 治理的基础……

Paolo Malinverno 在本次 TechTarget 的访谈中大范围的论述了应用集成方面的问题，认为治理是成功的 SOA 的一个永久的支柱。

你是如何定义 SOA 治理的？

SOA 治理是关于纪律的，并确保那些非常重要的决策能够让合适的人获得，并保证有些人有适当的权利来做出那些决策。这就是 SOA 治理问题的一半了。第二部分就是不管是在何时做出的决策，SOA 治理需要保证那些决策能被很好的执行。这不仅仅是设置一个速度限制，而是要强制其执行，并最后给予人们奖励或是惩罚。这就是 SOA 治理的真正含义。

你能将治理分割为一个一个的子类或是方面吗？

一开始是存在阶段。就像我刚刚提到的那些。就存在一个阶段用于定义政策的内容。实际上是通过叙述如下的话开始的，“好的，这个决策集合，我想让它们包括在内。”你获得你的适当的支持和适当的权利以做出那些决策。然后你将这些决策分配到最适合执行它们的那组人中。这就是你所要做的第二件事。还有，你将各个人物组织起来以确保所有事情都能被接受且命令都是适当的，每个人都有各自应该有的权利。

在这之后就是管理部分，按照所做的决策执行。最后一个部分就是处理接受度——人们是否接受那些为他们而作的决策。如果说你决定不让任何人在注册库/知识库中引入新的服务，同时某些人在没有必要的清除的情况下连续工作好几周以实现这个功能。那么你将决定如何来处理它。你不得不决定处理这项服务的方式。你要将它和其他服务一起放入知识库中吗？你要将它扔掉吗？你将如何处理这个人？你将为他鼓掌或是告诉他接下来的两个月可以休假了？实际上，这里存在风险，而你不得不采取好的行为。

当他们执行 SOA 治理时用户最常犯的错误是什么？

用户所犯的最大的错误就是为其设置一个速度限制，接着就预期每一个人不会超越它。这就是最大的错误，因为人们会做他们想做的事情。这就是一直在发生的事实。我是一个意大利人，而打破速度限制是我们的本性。

所以你建议如何来调整这个问题？执行，接着不期望人们打破速度限制，这将如何改变？

存在许多技术可以用于部分地解决这个问题。存在很多技术可以用来执行合适的流程，比如说，设计通过一个 Center of Excellence 的服务。关键是要遵循我在上面所说的流程，决定管理哪些决策，谁在管理这些决策，测量和服从，接着是接受度。在纸面上来说很容易，但实际上是非常苦难的。

在过去的一年中治理是按照一个特殊的方式在改变吗？

没有，问题趋于相同。编程者对于他们的整个生活都趋于无规律，我们现在还不能找到其中的规律。

如果你现在可以投资此时能够有助于 SOA 治理的事物，那么该事物将会是什么？

我认为如果我有答案，那么我现在就不用工作了。恐怕我对此没有答案。这是因为治理是相当难处理的一件事。你可能因为囊括太多而使一个项目失败，你也可能因为内容太少而使其失败。所以，你只需要足够多。接着，所有这些流程决策和你执行那些决策的方式都应企业而异。所以，对于具体的创新只有很小的空间。IBM Global Services 已经做了很多很多。如果我有一个像这样的创新，我将在今天卖掉它，然后回到 Bahamas。

最近你还遇到过其他的基层组织问题吗？

SOA 治理都趋向于拥有不少人的集团。但是，实际上像论证 SOA 之类的组织问题起了很大的作用，而且实际上能有效地传递给企业，打通所有围绕 SOA 的怀疑论调，能揭穿 SOA 的面纱，这些都是非常重要的事情。正如我所说，SOA 治理趋向于专注于一整套具体的事情。对于我来说还有很多的事需要做。

你还有其他想对 SOA 治理进行补充的事吗？

作为一个总结论述，你不能俯瞰整个 SOA 治理，因为它总是在你周围神出鬼没。已经有很多人用尽他们的智慧对其进行思考—判定他们是否能跟上它的速度。不限定 SOA 的范围将可能给自己带来致命的威胁，同时这也限制了 SOA 的有效性和价值。这也是我们为什么耗费如此多的时间解决这些麻烦以防止这个发生的原因。

(作者: Brein Nally 来源: TechTarget 中国)

Dun & Bradstreet 公司高度评价 SOA 治理

对于一个企业级的面向服务架构项目而言，存放设计时元数据的注册储存库的重要程度有多大呢？Dun & Bradstreet (邓白氏国际信息咨询有限公司) 技术副总裁 Doug Smith 对此只回答了一个词：极其重要。

在过去的几十年里，Smith 曾经担任过 D&B 公司的首席执行官和首席信息官。为了适应 21 世纪的需要，他为这家已有 165 年历史的公司重新设计了 IT 基础结构，其中包括：寻找一种解决方案以实现快速设计和开发公司的商业信息产品——这是 D&B 公司的核心。

他认为：对于重新构建 IT 以及重新构造设计时元数据的注册储存库而言，SOA 就是解决方案。在这个案例中，Smith 使用了 LogicLibrary 公司的 Logidex 工具，它是搭建企业范围内的 SOA 环境的关键。

从一开始，Smith 就坚信：SOA 可以使公司在遗留系统的基础上向前发展，并且能够保留原系统中的有效数据以及专业技术——这些是 D&B 公司的核心产品，D&B 将其产品交付给商业用户，为他们的购买以及合作行为的决策而服务。毕竟，D&B 的全球商业数据库中包括超过 1 亿条的商业记录——在新的应用程序中，这些数据也需要单独保存并保持更新。

“我们与众不同的地方是：我们在评估数据、确保数据及时性以及确保数据高精度度方面的能力。”他解释说。

在这个艰巨的重建项目的初始阶段，他说，“我明确表示：只有将系统转移到 SOA 平台上才能够经得起未来的考验，而且，目前我们还在使用那个将要被淘汰的平台，以便平

滑过渡——将用户转移到新的替代产品中。同时，我们也能够继续保持我们在该领域中的投资——你可能并不愿意立即淘汰那些包含专业知识或者逻辑的系统，但是我希望以更有成本效益的方式来进行系统移植。也许，你已经开始考虑如何使你的业务流程商品化。”

正如 Smith 所观察的，大约有 20 到 30 种 D&B 的产品使用了业务流程。他的想法是通过可重复的服务来商品化那些流程，这些服务能够“根据安全质量不断地形成可预测性的等级”。

“SOA 允许你在多种产品中重复嵌入 IT 流程，这些产品是出售给用户的，”他解释说。

通过在多个服务中调用相同的、经过检验可靠的流程，开发小组能够更迅速地装配新的应用程序，也能够捕获用于替代遗留系统的现有逻辑。但是要做到这一点，架构师和开发人员必须具备一定的知识背景，他们需要知道什么服务是可用的，以及哪些服务能够提供治理——Smith 所提及的一个关键词——用于描述服务能够或者无法被使用。

“这就是选择 LogicLibrary 公司的工具来搭建 SOA 环境的过程，因为 Logidex 工具不仅能够确保开发小组有权利访问企业架构的原始工具，也允许我们使用产品来执行治理。我们意识到：使用 SOA 实现来推动公司向前发展不仅仅是技术问题，实际上，它是关于如何选择工具来简化治理流程的问题。从发现原始工具开始到最终确定原始工具，这中间的每一个步骤都需要对影响因素——与改变原始工具或者组件相关——完全了解”。

在 2005 年，使用 SOA 实现的工作已经开始了，但只是一个典型的试验项目；今年，该工作进一步发展，已经涉及到了主要的业务。在此过程中，D&B 公司的 Smith、架构师以及开发人员发现了设计时的注册储存库所具备的意想不到的优势。它不仅提供了是原始工具的详细记录，同时也显示了变更和消除这些原始工具之后的结果。

“应用程序合理化组件依赖于原始工具，如果我们在环境或者应用程序中不使用某一特殊的原始工具，那么该组件知道接下来将会发生什么事情，”他解释说。“有哪些其它的原始工具也在使用这个组件呢？”

与测试类似，治理有时看起来也是开发过程中不可避免的“灾难”，但是，Smith的经验是：如果你所使用的工具能够显示变更原始工具的原因和结果的话，那么，它实际上是促进了项目之间的协同工作。

“通过做成本影响分析来支持重新构建，我们发现：如果要得到强有力的因果关联关系，那么我们就需要增强的、远远优于以前平台所具有的通信和协作能力，”他说。

(作者: Rich Seeley 来源: TechTarget 中国)

对服务进行合理精简

SOA 是企业架构的一个方面，本身并非一种技术，而更是一种规则。SOA 最佳实践的精髓就是理解如何用正确的服务处理正确的问题。毕竟，IT 是用来处理业务问题的，而不是实现技术方案然后用它们找出一些业务问题。

我们大家都知道金发姑娘与三只小熊的故事，这个故事讲述了想得到她的椅子、粥和床的三只熊的故事。就像能满足所有人的事物或不能让金发姑娘满意的食宿一样，很多早期面向服务架构的支持者都认识到他们必须测算服务的大小才能解决他们的业务问题，这样他们就能用复用最大化减少不必要的开销。不仅如此，这些早期的采用者还意识到他们必须处理服务基础设施的重要方面以及重新设计架构的最佳实践，哪怕是部署一小部分服务。毕竟，成功的 SOA 实践需要的不仅是正确的服务，还需要用正确的方式实现它们。

ZapThink 反复强调的就是 SOA 是企业架构的一个方面，本身并非一种技术，而更是一种规则。SOA 最佳实践的精髓就是理解如何用正确的服务处理正确的问题。毕竟，IT 是用来处理业务问题的，而不是实现技术方案然后用它们找出一些业务问题。

识别合理尺寸的问题

正确识别服务的合理尺寸一个关键是识别 SOA 项目的适当范围，换句话说，就是解决正确的业务问题。在处理 SOA 项目时关注公司想要的所有潜在服务是非常危险的。这种方法实际上就是想把整锅水都烧开，其实是毫无意义的。毕竟，SOA 的目的是处理公司面对的不可预测的经常性的变化。所以，我们没有必要试图通过识别公司需要的所有服务来处理一个非常庞大的业务问题，因为这些服务还是会继续随着时间变化的。

另一方面，ZapThink 曾经鼓励公司应该首先处理最小的问题才能成功应用 SOA。现在，这很容易被误解成鼓励公司只开发粗粒度的服务，甚至是解决每人实际关心的一些小问题。SOA 领域一些著名的专家对这个问题也存在争论，但他们对此问题的解释是适用于市场的。事实上，我们赞同其中的很多观点，尤其是 James Governor 的。相反，公司想要的最后的東西是细粒度的服务集合。它们满足的业务需求很小以至于每个单独的服务几乎是没用的。公司需要合适粒度的服务来处理业务问题，但是业务问题又应该是多大呢？这个问题不同于服务的粒度或者实现服务所需的基础设施、架构和计划。

理解构建一个完全 SOA 的细粒度服务(或者相同风格的完全粗粒度服务)的错误很重要。我们喜欢使用 LEGO 拼装玩具作为 SOA 的隐喻来说明这个问题的答案。单从细粒度服务来实现 SOA 就好比只用最小的 LEGO 构建某种东西一样。当然，每一小块都有合适的接口使你能把它连接到其它的 1x1 块上。那就像拥有一箱马形状和车形状的 LEGO。当然，它们有突起，而且与你想要的很相似，但它们不是可重用的，而且完全无助于构建其它任何马和车之外的东西。这个故事的道理就在于，你需要对服务合理分类，从细粒度的一直到粗粒度的，这样才能说明你特有的业务问题。

我们的首个服务需要多少基础设施

假如你正在启动一个 SOA 项目，你或许只看到了有限范围的服务功能。但这并不表示你需要限制基础设施。因为一个组织为他们的第一个项目可能实现一小部分服务并不意味着他们就能提供产品中成百的服务。事实上，拥有业务依赖的粗粒度服务是很重要，至少跟有成百个细粒度的，且每一个都能解决整个业务问题一部分的服务一样重要。所以，公司需要从一开始就考虑在整个生命周期内如何开发、部署、测试以及维护你的服务。

一个例子就是 ZapThink 对注册、存储、元数据管理和 SOA 控制的观察。在早期的 SOA 和 Web services，根本没有人对基础设施构件感兴趣。这是因为注册和控制直到有足够多的服务在组织中被构建和部署时才会需要。然而，成功部署了 SOA 的公司发现实际上在服务部署过程中更早需要这些解决方案，因为尽管只有一部分服务被部署(甚至没有部署)，

但更早地搭建架构上的最佳实践很重要，这样的话，控制和元数据管理就能尽可能早地实现。对安全和身份管理同样如此。事实上，对于任何 SOA 的业务问题，SOA 的实现者都要意识到他们需要从用面向服务来解决他们问题的那一天起就考虑整体的 SOA 实现路线图。

从业务需求开始

没有公司采用 SOA 是不以处理重大业务问题为目标的。毕竟，如果我们所做的仅是为了用服务处理小问题，那么我们会跌入我们构建。所有的软件项目都必须很自然的从业务需求开始，而对于 SOA，主要的业务需求是“元需求”，它的目的是实现一个响应不可预见的变化的架构，一个本质上很灵活的架构。无论如何，架构师首先要意识到对 IT 的特殊业务需求，例如，一些新的业务功能，新的产品，新的业务关系或者新的预算限制。无论这些业务需要什么，面向服务的架构都必须铭记这样的需求永远没有尽头。

毕竟，SOA 项目不是典型的 IT 项目。典型的 IT 项目以确定的业务需求开始，架构师或项目团队里的其他人可以和业务干系人(提出需求的人)坐在一起提炼出一套需求用例。但是，SOA 项目的用例完全不同。它不会说明特定的“我需要系统这样做”一类的需求，而是描述很多用户想如何利用。于是，面向服务的架构必须允许需求变更而不是假定有一套固定的需求。解决方案必须不只满足业务的短期需求，而必须能够满足未来的需求。

ZapThink 的做法

因此，我们说 SOA 项目应该以最小的业务问题为目标。这并不意味着最小的应用或最小的解决方案。就好比业务过程以组成服务的方式组合，也能包含更小的业务过程。业务问题也是组合的。公司可能会有如何更好服务客户的业务问题，但这个问题实际上由很多更小的问题组成，例如更好的确定他们客户的身份，改进客户支持过程或者减少支持客户的时间和费用。每个这种问题又是由更小的问题组成。

因此，面向服务的架构必须能为组织如何解决迭代式问题维护一个实用的思维模型，而不是维护一个单独的附带的组织架构。通过这种方式，架构师能用不大也不小的合适粒度的服务为业务问题构建 SOA 解决方案。

尽管我们的产业专家不会同意，但实际上我们正在谈论的东西是很能给人鼓励的。很长时间里我们没有谈论 SOA 为何物了，而只是谈论如何使用它。所以，现在的谈论正好帮助大家意识到 SOA 成功的关键并不在于选择合适的中间件，而在于选择合适的服务，并且用合适的基础设施使这些服务对组织产生价值。

(作者: Ronald Schmelzer 来源: TechTarget 中国)

数量并不是 SOA 成熟度计量标准

随着公司日益追求面向服务架构(SOA)的利益，组织想要通过一个客观的标度计量他们的努力变成了有意义的事情。SOA 是代表着 IT 不断满足业务持续的变化需求同时 IT 组织的管理和治理方式也在不断地变换的方法。SOA 使得公司可以抓住一些切实的东西并能够凝结出实物来计量他们的前进。无论是软件厂商，分析人士，咨询公司还是市场商人，他们都堆积在用他们的方式衡量 SOA 成熟度上。

就像 ZapThink 公司之前提到的，很多已经存在的成熟 SOA 模型都是自给服务的、受软件厂商驱动的结果，他们的目的在于将公司购买的产品比他们构架解决方案结合在一起的时候能够更加成熟一些。不过，我们也注意到越来越多的公司在衡量他们的 SOA 应用的成熟度时开始将多种多样的，试图理解维度效益产生努力结合起来。然而，在过去的十八个月中，我们已经看到了相当多的失望，公司们都在单独的使用着他们创造的 SOA 程序并对其进行管理，并将其作为衡量成熟度的一种表现。这就意味着，拥有数以千计的服务将远远重要于一个成熟服务。毫无疑问，这绝对是一个错误的结论。

关注一个服务 Vs. 上千个服务

对于服务创建、管理或者消费的数量并不能用来衡量成熟的很简单的一个解释就是，创造服务是一个价值很低的活动。实际上，现在市场上有大量的工具可以通过已有的软件逻辑、数据库、中间件设备甚至是硬件网络设备等，创造出数以千计的服务界面来。为了更进一步强调这一观点，我们甚至可以得出一个极端的观点，展示服务是没有价值的。服务价值是在一些事物对其进行了消费的时候产生的。换句话说，在一千个服务当中，如果几乎没有甚至没有服务有人对其进行消费的话，那么它的价值根本赶不上一个拥有成千

上万的用户的服务。这种 SOA 价值的观念并不是有多少服务，而是聚焦于服务的消费有多少。

不过，这并不意味着在多个流程领域中不得不再使用的一些服务是有价值的。在 ZapFlash 的文章《所有的服务都应该能够再使用吗?》中，我们讨论了使用服务的很多原因。其中，再使用也位列其中。但是，至少要有一个消费者要去给服务价值。因此，在创建服务是价值很低的以及相关的活动也是低价值的环境下，大家最关心的问题就成了如何使用服务更加简单、可靠以及可以更灵活的使用。因此，即使一个公司有一个他们极为关注的服务，而且他还是可靠的，可使用的以及面对不断变化的情况时是灵活的话，那么这个服务以及导向这个服务的架构都具有很高的成熟度。不过，如果一个服务是缺乏质量的，缺少治理的，而且由于是单独的缘故而缺少控制，那么公司只能完全的错失 SOA 带来的利益了。公司应该成熟化那些对它们而言最重要的服务，而不是关注他们有多少服务。我们来假设数量代表着成熟的程度使得公司建立了潜在的隐患。

度量结构的成熟度

现在有一个迹象，那就是公司并没有恰当考虑有关成熟度的概念。他们在架构过程中，都太晚考虑有关质量、治理、生命周期以及可靠性的有关事宜。创建服务只是在 SOA 开发生命周期当中的一步，但是甚至不是第一步。实际上，公司采取开发和执行服务的步骤应该是他们采用成熟的架构过程的最后几步之一。特别是质量因素，应该在任何服务被建立以前就要慎重考虑。如果一个公司没有考虑过这些因素的话：服务是如何被测试的，顾客在使用服务的过程中是如何成功可靠的运行或者是如何失败的，以及他们在服务版本中是如何重申的，那么，所以签约了这些基础服务的客户将在架构他们自身服务时面临同样的风险。

同样的，治理也是要在 SOA 设计阶段的早期需要被考虑的因素之一。公司如何确保服务被按照企业和政府的要求恰当的使用而不是走向毁灭是需要考虑的问题。就像我们已经讨论了多次的说法，没有治理的 SOA 项目将导致没有人准确的知道服务在企业中是如何被

使用或消费的混乱的系统。商业上是极为痛恨混乱的，他们将很快导致一个混乱的 SOA 结果。在这种情况下，ZapThinks 也给那些咨询客户一个建议：建议他们将 registry/repository 系统和元数据管理系统当作最初的 SOA 活动进行执行。通过搜集和管理企业的元数据，组织可以消除简单系统随着时间的流逝变得越来越复杂而又笨拙的问题。

另一个需要被考虑的组成成熟度的因素就是安全性和可靠性。就是因为公司可能将服务暴露给客户或者合作伙伴并不意味着他的安全性和可靠性比暴露给更多的人的安全性和可靠性低。实际上，如果服务是独立运行于公司与顾客接触的话，那么任何危及到服务提供可靠性和安全访问能力安全的，都会危及到整个系统的价值。虽然这看上去是十分有道理的，但是我们不能为你准确的指出我们实际发现多少次客户在决定先使用在做安全的环境中暴露了他们的服务。正像质量和治理一样，成熟的 SOA 组织在他们使用第一个服务之前就要考虑可靠性和安全性。

所以，一个计量 SOA 效果成熟度的方法就是通过计量服务中质量、治理、安全性以及生命周期的方法来度量，无论是多少个服务都是如此，即使只有一个也不例外。同样的，数量和组织效果的成熟度毫不相干。但是，这些 SOA 技术的方面也仅仅是当今组织整个 SOA 成熟度的一小部分。用面向服务的方法进行治理、质量管理、可靠性管理以及变化管理改变了 IT 组织运行中的运行方式。公司如何管理组织采纳的 SOA 衡量着他们采用架构最好实践的成熟度。

衡量组织的成熟度

随着公司的注意力从仅关注生产服务向消费服务进行转变，他们开始注意到他们已经存在的组织框架和管理框架已经不再有效。特别是，IT 组织也许已经排列入仓指向了那些以紧密连接的，非组件化的，毫无疑问不可能是面向服务的世界的形式存在的具体的系统或流程。至今为止。在持续变化的并且异构的环境中进行再使用也是对不同的组织结构进行质量要求、服务生命周期要求和治理要求的一种不同的方法。

随着公司不断成熟化他们的服务和架构，他们将必然的成熟化他们的组织。这意味着他们也许会成立杰出的 SOA 中心，EA 团队，以及集中的或分散的解决方案体验中心。他们的目的是贯穿组织，能够帮助他们指导、技术化 SOA 最好的方法，管理增长的服务使用（甚至包括了暴露的服务数量）以及用 SOA 的方法维持企业效率和 IT 治理。

同样的，即使公司只有一个他们关注的服务，增加这个服务质量和可靠性的使用和反复将会必要的要求组织进行变革。变革将会要求自上而下的管理，自下而上的面向服务的方式，以及循环的开发方式和服务生命周期管理，持续建立和改进业务使用的服务案例的能力。就像在这里服务的数量一点都不重要一样，上述的这些要求对于 SOA 项目来说也都是些小虾米。即使组织将注意力放在为特别的一部分进程或者是个人部门开展的服务中的很小的一部分上，他们也能将他们的组织和结构的方法成熟化，从而获得他们努力的最大的价值，即使在他们并没有传播到公司的其它部门的情况下。也没有仅仅为了达到称为最高级的成熟度就进行“企业范围内的 SOA”的必要。将精力集中在问题上，并且通过得到最有价值的面向服务的解决方案的方法使得企业尽可能的成熟是必要的。

ZapThink 的看法

成熟度是和自身相比较，而不是和其他公司比较的。也就是说，没有什么外部的度量方法可以被公司采纳来考虑自身的 SOA 活动是不是“最成熟的”。正确的方法是，公司需要理解服务的数量或者全企业应用 SOA 的努力和他们使用的服务给企业带来的重要性和价值几乎不相关。将精力放在多维的方法上用来完备对结构和组织以及技术上的考虑将帮助公司完善他们的 SOA，因此将给他们业务的带来最多的收益——这也是唯一的与成熟化有关的方法。

当我们满怀希望的看着公司将更深层的在 SOA 项目中考虑有关结构和组织成熟度的有关事宜时，我们知道简单的诱惑性。如果不恰当的，成熟度的模型将强迫组织寻找他们希望 SOA 那样的快速的解决方案。不幸的是，SOA 对企业而言是一个很少见的可以和混乱的

结构挑战斗争的快速解决方案。就像 ZapThink 建议的，一个可以反复使用的，持续的面向服务的方法将会考虑可以被 SOA 恰当解决的业务问题。ZapThink 也建议一个 SOA 成熟度的逐步式的方法，它不会考虑服务数量的问题，而会考虑企业选择使用和消费的服务的质量。

(作者: Ronald Schmelzer 来源: TechTarget 中国)

SOA 最差实践:过多的 Web 服务=麻烦



Sonic 软件和 Progress 软件公司技术总监 Dan Foody, 在接受 TechTarget 采访中他所认为 SOA 最差实践基础上正见证着 SOA 的一系列失败。作为技术总监, 他并没有从技术层面上来看待这个问题。在他看来, 问题在于项目的管理, 缺乏对 SOA 方案的理解和企业没有能够作出适合 SOA 理念应有的改变。

在 SOA 方面, 你们看到了什么?

如今, 人人都在谈论 SOA, 企业里人人都知道 SOA 并想参与其中。但是, 其自身正遭遇着挑战, 因为并不是每个人都能正确理解在面向服务架构里应该做些什么。

这种对于 SOA 的无知会带来什么挑战呢?

我们发现当人们宣称他们在做面向服务架构时, 他们所做的是并不是 SOA 所预期的, 甚至有些吓人。

有例为证吗?

给你们举一个我最喜欢的例子。有一家公司在做一个大会议主题演讲中提到他们在面向服务架构方面有多么的成功: 他们已经具有 300 项服务, 并且在一年之内能达到 1000 项。听起来很不错不是吗? 唯一的问题在于这种类型在发展中却很糟糕。

为什么 SOA 理念会脱轨呢？

我们做面向服务架构，目的是避免复杂性，而不是让多达 30 种 web 服务去做同一件事情。你的目的是要去再利用。如果有上千个 Web 服务，实现再利用的可能性事实上为零。因此，他们也许拥有许多 Web 服务，但却无法实现面向服务架构。所以，在业务上也表现不出任何好处。这就是一个 SOA 脱轨的例子，人们错误的认为“越多越好”。

这是因为人们在定义面向服务架构时的问题吗？

这个问题有两个方面引起。一方面，一些人根本就不理解 SOA 的含义。建立 Web 服务，并不代表你建立了面向服务架构，这与 Web 服务的数量无关。SOA 关系到你对已有的 Web 服务利用途径的数量。这才是在各方面定义面向服务架构的一个标准。许多人都在这一点上误解了。真正有价值的是对于 Web 服务的再利用而不是 Web 服务本身。跨组织的共享资源才是面向服务架构的价值所在。

还有一部分人，即使他们理解了这一点，也不一定知道如何去建立成功的面向服务架构。有的人认为建立所有权标准可以创建有效的面向服务架构。于是我们看到一些组织将 Web 服务与 SOAP 作为基础，将之拆散，让其不能与外界任何程序相互作用，然后将自己的特性加入准则当中。然后，他们需要将所有的东西编码然后联结，在此过程中将丢失掉许多协调性的益处而致使在开始的时候建立面向服务架构变得很困难。虽然在这些例子中，尝试 SOA 的人们理解了他们将要达成的目标，但他们并不知道途径为何。

标准能够帮助我们看清道路吗？

多数人会说:SOA 并不存在技术问题，最大的挑战在于结构性问题。所以对于技术架构的重视只能解决 20%的问题，但 80%关于人员的问题仍有待解决。

我认为，犯了最差实践这些错误的人们不会在一开始就按照规则来办事。他们相信自己知道自己在做什么，这也是问题之一。那个在大会上发言的先生，讲述建立上千 Web 服务，他就相信自己比任何人都了解如何做 SOA，并且不会听取他人的建议。

因此，我并不认为标准结构能够解决这个问题。一个企业能从 SOA 得到的最重要的益处并不是用纯技术的形式能够呈现的，而是使信息技术将对商务有意义的资源组织起来。将信息技术与商务结合起来并使其随着时代的发展而持续进步。因此，当一个商人谈到盘点时，IT 人士就能知道这意味着什么，他们会将相关的资源和技术组织到一起运作。这才是 SOA 的真正益处所在，这是你在任何一张架构图表上找不到的。

当人们有需要的时候，也会有技术帮助加入其中。他们想要，就能找得到。而恰恰是那些自以为是并没有好好利用技术帮助的人在技术层面上出现问题。

那么，这是不是说无知和缺乏理解是非常危险的？

对，自信和缺乏知识是非常坏的结合。

那么，在企业认识到他们应该采取别的途径之前，我们是不是会看到许多 SOA 失败的最差实践案例呢？

我们必然会看到许多项目的失败例子。一切引起企业变革的因素都会有附加效果。不是所有企业都能够完成变革的，也不是所有企业都知道如何进行变革的。所以，我认为在进行此类变革的同时，我们将不可避免的看到许多附带作用的产生和项目的失败。关键在于企业要认识到从失败中吸取教训而不要半途而废。

因此，我们要关注所有关于 SOA 失败案例的报道，避免同样的错误。如果一个人要引领 SOA 的启动，他不能在遇到失败时觉得羞耻，而是应该接受这些失败并将之当成又一个他能从中学到经验的案例。

(作者: Rich Seeley 来源: TechTarget 中国)