

**Faculdade de Tecnologia de São Paulo**

Autora: Débora Momono Alves Kuroiwa

**Reflexões sobre a Arquitetura Orientada a Serviço e  
o Surgimento de uma Nova Disciplina, a Engenharia  
de Software de Serviço.**

**São Paulo**

**2011**

**Faculdade de Tecnologia de São Paulo**

Autora: Débora Momono Alves Kuroiwa

**Reflexões sobre a Arquitetura Orientada a Serviço e  
o Surgimento de uma Nova Disciplina, a Engenharia  
de Software de Serviço.**

Monografia submetida como exigência parcial para obtenção  
do Grau de Tecnólogo em Processamento de Dados

Orientador: Prof.<sup>a</sup> Márcia Ito

**São Paulo**

**2011**

**Dedicatória**

À minha família.

### **Agradecimentos**

Agradeço a Deus pelas pessoas que colocou em meu caminho para que me guiassem. À minha orientadora, Prof<sup>a</sup> Marcia Ito, pela paciência. À minha família, pelo carinho e apoio.

## **RESUMO**

Com a crescente competitividade do mercado econômico, as empresas depararam-se com a necessidade de (re-) adequação de seus processos de Negócio, visando tornarem-se mais produtivas e lucrativas. Considerando TI como um dos pilares das empresas para produção de respostas às demandas do mercado, as empresas passam a demandar que TI apresente um alto grau de alinhamento Negócio-TI e flexibilidade no desenvolvimento de seus sistemas. Essas características são observadas em um paradigma de arquitetura conhecido como Arquitetura Orientada a Serviço, porém que apresenta alto grau de complexidade em sua concepção. A Engenharia de Software de Serviço surge como uma proposta de disciplina complementar à Engenharia de Software Tradicional, apresentando princípios que visam suportar o desenvolvimento de sistemas construídos com base na Arquitetura Orientada a Serviço, permitindo melhor aproveitamento das características oferecidas por esse paradigma de arquitetura e, conseqüentemente, gerando os resultados esperados pelas empresas.

## **ABSTRACT**

With the increasing competitiveness of the economic market, business were faced with the need to (re-) adequacy of its business processes in order to become more productive and profitable. Considering IT as a pillar of the companies to produce responses to market demands, companies are demanding that IT provides a high degree of IT-Business alignment and flexibility in developing their systems. These characteristics are observed in an architectural paradigm known as Service Oriented Architecture, but it has a high degree of complexity in its design. The Software Service Engineering emerges as a proposal of complementary discipline to traditional software engineering, presenting principles that aim to support the development of systems built on Service Oriented Architecture, enabling better use of the features offered by this paradigm of architecture and, thus, generating the results expected by the companies.

## LISTA DE ILUSTRAÇÕES

	Pág.
<b>Figura 2.1</b> – Resumo dos princípios SOA	19
<b>Figura 2.2</b> – Aplicações em camadas lógicas, sem implementação da Arquitetura Orientada a Serviço.	23
<b>Figura 2.3</b> – Aplicações baseadas na Arquitetura Orientada a Serviço.	24
<b>Figura 3.1</b> – Processo de Engenharia de Serviço	28
<b>Figura 3.2</b> – Construção de Serviço por Composição	30

## LISTA DE SIGLAS

<b>SOA</b>	Service-Oriented Architecture <sup>1</sup>
<b>TI</b>	Tecnologia da Informação
<b>ESB</b>	Enterprise Service Bus <sup>2</sup>
<b>BPM</b>	Business Process Management <sup>3</sup>
<b>OO</b>	Object Oriented <sup>4</sup>
<b>CBD</b>	Component-Based Development <sup>5</sup>

---

<sup>1</sup> Arquitetura Orientada a Serviço.

<sup>2</sup> Barramento Empresarial de Serviço.

<sup>3</sup> Gerenciamento de Processo de Negócio

<sup>4</sup> Orientação a Objeto

<sup>5</sup> Desenvolvimento Baseado em Componente

# SUMÁRIO

<b>1. Introdução .....</b>	<b>10</b>
1.1. Objetivo.....	11
1.2. Justificativa .....	11
1.3. Método de Pesquisa .....	12
1.4. Organização do Trabalho .....	12
<b>2. Arquitetura Orientada a Serviço .....</b>	<b>14</b>
2.1. Conceitos.....	14
2.2. Benefícios .....	17
2.3. Desafios.....	18
2.4. Aplicação .....	19
<b>3. Engenharia de Software de Serviço .....</b>	<b>22</b>
3.1. Princípios .....	22
3.2. Desafios.....	24
3.3. Engenharia de Software de Serviço x Engenharia de Software Tradicional .....	25
3.4. Processo de Engenharia de Serviço .....	26
<b>4. Conclusão .....</b>	<b>28</b>
<b>5. Referências Bibliográficas .....</b>	<b>29</b>

# 1. Introdução

Ao longo da história, o mundo atravessou diversas transformações tecnológicas e econômicas que promoveram constantes avanços nos processos produtivos, focando no aumento da produtividade e lucratividade.

Em meados do século XVIII, inicia-se a Era Industrial, que se estendeu por dois séculos (1750 – 1950), onde começaram as maiores transformações tecnológicas, com o surgimento da máquina a vapor, o telégrafo e, posteriormente, a invenção do automóvel, o rádio, a televisão, entre outros.

A Era Industrial é seguida pela Era da Informação, após a Segunda Guerra Mundial, quando surgiram tecnologias que promoveram o aumento na comunicação mundial, e onde o mercado passou a se concentrar não mais na produção agrícola, artesanal ou industrial, mas sim na produção da informação. (LUCCI, 2008).

Todas essas transformações mantiveram o foco no aumento da produtividade, e para se manterem competitivas, as empresas continuamente ampliam seu mercado focando não apenas no bem produzido, mas também no serviço a ele agregado, tornando, desta forma, o mercado ainda mais desafiador. Considerando isto, a informatização da empresa constitui um dos pilares fundamentais para a produção de respostas da empresa ao mercado, permitindo a inovação e sustentação em novos produtos e serviços, por meio de integrações entre sistemas de empresas parceiras, fornecedores, governo entre outros.

Tem-se então, um grande desafio: Como a área de TI conseguirá suportar continuamente a produção de respostas, com agilidade, ao mercado cada vez mais dinâmico e desafiador?

Diante desse contexto, tornou-se cada vez mais evidente a necessidade de sistemas mais flexíveis, aderentes e de fácil e rápida adaptação aos processos de negócio. Surge então um novo paradigma de arquitetura de software, baseado no conceito de Serviço, e que tem como principais características o baixo acoplamento, dinamismo e adaptabilidade, a reutilização de serviços, interoperabilidade e a independência de Tecnologia: Arquitetura Orientada a Serviço.

Porém, a utilização desse novo paradigma pelas empresas ainda é considerada “precoce” e, em virtude das características de sistemas SOA (Service Oriented Architecture – Arquitetura Orientada a Serviço) citadas acima, a utilização de princípios e técnicas da

Engenharia de Software tradicional tem-se mostrado limitadas para o seu desenvolvimento. (HEUVEL et al., 2009).

With SOA systems operating in distributed and heterogeneous execution environments, the engineers of such systems are confined by the limits of traditional software engineering [...]<sup>6</sup>. (HEUVEL et al., 2009, p.26)

Começa-se então a discutir a Engenharia de Software de Serviço como uma disciplina emergente que envolve parte da Engenharia de Software Tradicional e introduz outros novos conceitos para o desenvolvimento de sistemas baseados em SOA. (HEUVEL et al., 2009).

## 1.1. Objetivo

O objetivo desse trabalho é discutir sobre a utilização da Arquitetura Orientada a Serviço (SOA – Service Oriented Architecture) como principal paradigma emergente de arquitetura de software, focando na Engenharia de Software de Serviço como nova área do conhecimento no desenvolvimento de sistemas, relacionando sua necessidade e desafios.

## 1.2. Justificativa

A discussão sobre o tema “Engenharia de Software de Serviço” foi incentivada pela repercussão a respeito da utilização da Arquitetura Orientada à Serviço, que apresenta muitas vantagens (e desafios) em sua aplicação, porém, ainda necessita de metodologia que norteie o desenvolvimento de sistemas baseados neste paradigma de arquitetura, uma vez que a Engenharia de Software Tradicional apresenta-se muito limitada para esse tipo de arquitetura.

---

<sup>6</sup> Com sistemas SOA operando em ambiente de execução distribuído e heterogêneo, os engenheiros de tais sistemas são confinados pelos limites da engenharia de software tradicional [...] Em particular, nós introduzimos a engenharia de software de serviço com uma disciplina emergente que envolve parte das disciplinas da engenharia de software tradicional, adotando o ‘pressuposto do mundo aberto’.

Service Oriented Architecture (SOA) is rapidly emerging as a premier distributed computing paradigma for developing, integrating, and maintaining enterprise applications. Many organizations are now in their early use of SOA, and assume that they can simply apply principles and techniques from pre-existing software engineering paradigms [...]. SOA-enabled applications operate in distributed, non-deterministic, unpredictable, and highly dynamic heterogeneous execution environments; hence, SOA engineers quickly encounter the limits of such traditional software engineering paradigms, which do not provide any style-specific advice. <sup>7</sup>(HEUVEL; ZIMMERMANN; LEYMANN; PATRICIA LAGO; SCHIEFERDECKER; ZDUN; AVGERIOU, 2009, p.26)

### **1.3. Método de Pesquisa**

O trabalho apresentado foi desenvolvido por meio de levantamentos bibliográficos e artigos científicos.

### **1.4. Organização do Trabalho**

No Capítulo 1 é apresentada uma introdução contendo o contexto histórico e atual que fomentou o surgimento da Arquitetura Orientada a Serviço. Apresenta também o objetivo do trabalho, sua justificativa e o resultado que se espera atingir.

O Capítulo 2 apresenta os conceitos da Arquitetura Orientada a Serviço, vantagens e desafios deste novo paradigma de arquitetura.

---

<sup>7</sup> Arquitetura Orientada a Serviço (AOS) está rapidamente emergindo como principal paradigma de computação distribuída para o desenvolvimento, integração e manutenção de aplicações corporativas. Muitas organizações estão agora em seu uso precoce de SOA, e assumem que eles podem simplesmente aplicar princípios e técnicas dos paradigmas pré-existentes da engenharia de software [...]. Aplicações habilitadas SOA operam em ambientes de execução heterogêneos, distribuídos, não-deterministas, imprevisíveis e altamente dinâmicos, portanto, engenheiros SOA rapidamente encontram os limites de tais paradigmas da engenharia de software tradicional, que não fornecem qualquer conselho de estilo específico.

O Capítulo 3 introduz o conceito de Engenharia de Software de Serviço, uma breve comparação com a Engenharia de Software Tradicional e também sua aplicação no desenvolvimento de sistemas baseados em SOA<sup>8</sup>.

No Capítulo 4 é apresentada a Conclusão do trabalho.

---

<sup>8</sup> Software Oriented Architecture – Arquitetura Orientada a Serviço

## 2. Arquitetura Orientada a Serviço

A Arquitetura Orientada a Serviço desponta como uma aposta de arquitetura para utilização, principalmente empresarial, no desenvolvimento de softwares, pois com o crescimento do mercado e o aumento da competitividade, os processos de negócios têm-se tornado cada vez mais dinâmicos, desta forma demandando que o desenvolvimento e manutenção de sistemas de software empresariais respondam com a mesma velocidade.

Os paradigmas de arquitetura adotados até então, se apresentaram pouco flexíveis ao dinamismo exigido pelos negócios.

Os sistemas tradicionais cliente servidor sempre focaram a interação com o usuário, e os dados de forma centralizada o que tornava difícil qualquer mudança estrutural pelo fato de exigir um grande esforço, até mesmo nas novas aplicações clientes. (COSTA; CARVALHO NETO, 2007, p.02).

Pela dificuldade de adaptá-las ou reorganizá-las, vemos que as aplicações acabam sendo um empecilho no caminho das mudanças ou movimentos estratégicos das corporações. (TAURION, 2009, p.06).

SOA passa, então, a ser fortemente adotada por sua principal característica de promover o alinhamento entre Negócio e TI<sup>9</sup> (BIEBERSTEIN et al., 2008).

### 2.1. Conceitos

Diversas definições para SOA são encontradas em livros acadêmicos e sites: já a definiram como uma nova tecnologia, framework, metodologia, outros como um produto, solução de negócio, e assim por diante.

Neste trabalho, a definição adotada é a que apresenta SOA como um estilo de arquitetura: SOA é um paradigma de arquitetura, baseada no conceito de Serviço, que promove o alinhamento entre TI e Negócio. Esse paradigma de arquitetura visa possibilitar

---

<sup>9</sup> Tecnologia da Informação

maior agilidade para atender às demandas empresariais, flexibilidade para responder às mudanças, trazendo redução de custos à empresa e possibilitando a reutilização e (re-) combinação dos serviços. (SANTOS, 2009).

Segundo Santos (2009), Serviço pode ser definido como um componente que atende a uma função de negócio.

Serviço é um componente que atende a uma função de negócio (business function). Ele pode receber e responder requisições ocultando os detalhes de sua implementação. (SANTOS, 2009, p.25).

Os serviços em SOA apresentam as seguintes características:

I. *Contrato de Serviço padronizado:*

O contrato especifica o que o serviço faz e qual o procedimento para utilizá-lo. (Exemplo. O “contrato” dos webservices são os WSDLs, onde estão declarados os métodos e a forma como o serviço deve ser invocado);

II. *Autonomia e Baixo Acoplamento:*

Serviços devem ser autônomos e com baixo acoplamento, ou seja, não devem ter dependências funcionais e/ou técnicas.

For services to carry out their capabilities consistently and reliably, their underlying solution logic needs to have a significant degree of control over its environment and resources.<sup>10</sup> (ERL, 2011)

Segundo Evdemon (2005), serviços devem ser projetados e implantados sem dependências de outros serviços, e se comunicam por meio de mensagens controladas (especificadas) por contrato.

O baixo acoplamento visa reduzir impactos nos serviços quando necessária a manutenção corretiva ou evolutiva de um determinado serviço (JOSUTTIS, 2007).

---

<sup>10</sup> Para que serviços realizem suas capacidades consistente e confiavelmente, sua solução lógica precisa ter um grau significativo de controle sobre seu ambiente e recursos.

### III. *Abstração:*

Serviços devem ocultar detalhes de sua implementação, praticando o conceito de “caixa-preta”, que prega que para uso e reuso de um serviço não é necessário conhecer detalhes além do que é especificado no contrato de serviço. (MACHADO, 2004)

On a fundamental level, this principle emphasizes the need to hide as much of the underlying details of a service as possible.<sup>11</sup> (ERL, 2011)

### IV. *Granularidade e Reutilização:*

A granularidade do serviço pode ser definida como grossa ou fina, dependendo do nível de detalhe do serviço. Quanto maior o nível de detalhe, mais fina é a granularidade do serviço (SANTOS, 2009).

Conforme definido anteriormente, ‘Serviço’ representa um componente que atende a uma função de negócio e, considerando esta definição, os serviços devem apresentar a granularidade de uma função de negócio, porém implementando granularidade grossa, que proporcionará assim, a reutilização desse serviço em ‘n’ processos do negócio (FURTADO et al., 2009).

### V. *Componibilidade:*

Segundo Thomas Erl (2011), serviços devem ser capazes de participar como membros efetivos na composição.

### VI. *Estado:*

Serviços apenas mantêm-se ‘stateful’<sup>12</sup> quando necessário, visando não comprometer sua disponibilidade e seu potencial de escalabilidade (ERL, 2011).

### VII. *Descoberta:*

Serviços devem ser facilmente identificados e interpretados à medida que as oportunidades de reuso se apresentam (ERL, 2011).

---

<sup>11</sup> Em um nível fundamental, este princípio enfatiza a necessidade de esconder os detalhes subjacentes do serviço o tanto quanto possível.

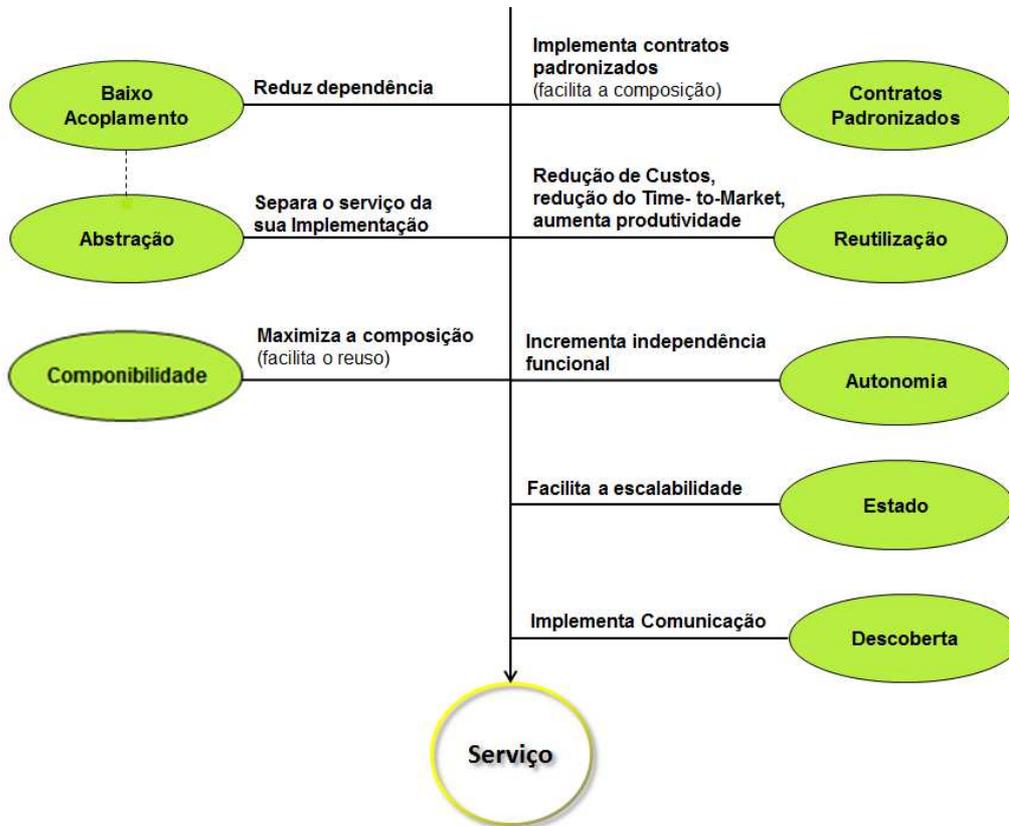
<sup>12</sup> Stateful – Capacidade de manter ‘estado’, ou seja, de armazenar informações na sessão entre chamadas.

### VIII. *Interoperabilidade:*

A interoperabilidade é considerada por Thomas Erl (2011) como fundamental para cada um dos princípios listados, à medida que cada um dos princípios contribui de alguma maneira para a interoperabilidade.

Este princípio trata da necessidade de integrar diferentes sistemas por meio dos serviços, independente dos ambientes em que tais sistemas operam ou linguagens de programação utilizadas na sua construção.

A seguir, ilustração que resume os princípios de SOA:



**Figura 2.1 – Resumo dos princípios SOA**

Fonte: Adaptado de <http://www.slideshare.net/Ridlo/soa-fundamentos>

## 2.2. Benefícios

Como dito anteriormente, a adoção desse paradigma de arquitetura visa proporcionar maior alinhamento entre Negócio e TI trazendo benefícios consideráveis às empresas.

Sistemas baseados na Arquitetura Orientada a Serviço possuem procedimentos mais ágeis de evolução e manutenção, pois implementam Serviços e suas características que proporcionam flexibilidade, como a componibilidade e a reutilização, possibilitando que TI responda às mudanças do negócio na velocidade desejada. Além disso, proporcionam:

*I. Integração ente sistemas:*

Devida às características de Interoperabilidade, Baixo Acoplamento e Contratos padronizados, sistemas construídos em diferentes linguagens e que executam em diferentes ambientes de execução conseguem se comunicar, ou seja, tais características promovem a integração entre sistemas, desta forma tornam ágeis os processos de negócios entre os diversos departamentos da empresa, entre fornecedores, parceiros e governo.

*II. Redução de custos:*

Pelas características de Reutilização, Autonomia, Baixo Acoplamento, Abstração e Estado, a Arquitetura Orientada a Serviço pode proporcionar a redução de custos da empresa com a manutenção de sistemas e até mesmo em novos desenvolvimentos.

Por exemplo, com serviços implementando as características de ‘baixo acoplamento’, ‘autonomia’ e ‘abstração’, a manutenção de determinado serviço não deverá gerar impactos aos consumidores de tais serviços, reduzindo desta forma os custos para manutenções “em cadeia”. No caso da característica ‘estado’, traz redução nos recursos de hardware, à medida que reduz o consumo de memória.

Além disso, segundo Gouveia e Gouveia (2011), gera redução de custos à medida que otimiza a comunicação entre técnicos e utilizadores de negócio, o time-to-market e o retorno sobre o investimento dos projetos.

## **2.3.Desafios**

A implantação da SOA, porém, tem encontrado em suas próprias características seus maiores desafios, o desenvolvimento de Serviços se torna muito complexo.

Os seguintes desafios são listados por Gouveia e Gouveia (2011):

### *I. Segurança*

Devida à característica da Interoperabilidade, os sistemas ficam mais vulneráveis a ameaças externas.

### *II. Rastreabilidade*

Pela capacidade de composição dos serviços, aumenta a necessidade de monitoramento e rastreabilidade.

### *III. Disponibilidade da Informação em Tempo Real*

Devido ao baixo acoplamento, SOA melhor adequa-se a sistemas que não demandam respostas em tempo real.

### *IV. Competências e experiência na Implementação SOA*

Processos e padrões de implementação SOA ainda encontram-se em nível baixo de maturidade e experiência.

### *V. Custo – Alto investimento inicial*

A implantação da SOA nas empresas gera um alto investimento inicial, considerando o lento processo de reengenharia das arquiteturas existentes, que requer recursos humanos e financeiros.

## **2.4. Aplicação**

Aplicações desenvolvidas em SOA são baseadas em princípios como local, protocolo e independência de plataforma tecnológica. Também são baseadas em padrões como contrato de serviço, composição de serviço, ESB<sup>13</sup> e registro de Serviço (HEUVEL et al., 2009):

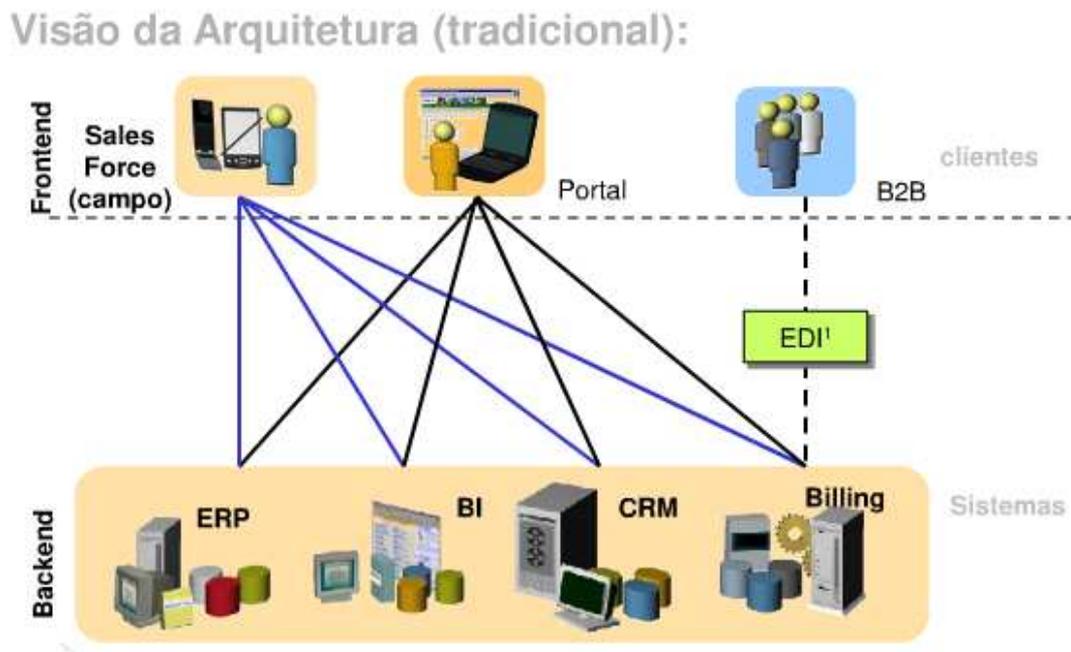
### *I. Protocolo: Utilização de protocolos de comunicação.*

---

<sup>13</sup> ESB – Enterprise Service Bus: Barramento de Serviço Empresarial

- II. *Enterprise Service Bus (ESB)*: Camada de integração que permite a troca de mensagens entre os consumidores e os provedores de serviços (HEUVEL et al., 2009).
- III. *Registro de Serviço*: Diretório de provedores de serviços disponíveis (HEUVEL et al., 2009).

A seguir, ilustrações de aplicações que não se baseiam em SOA (Figura 2) em comparação a aplicações baseadas em SOA (Figura 3):

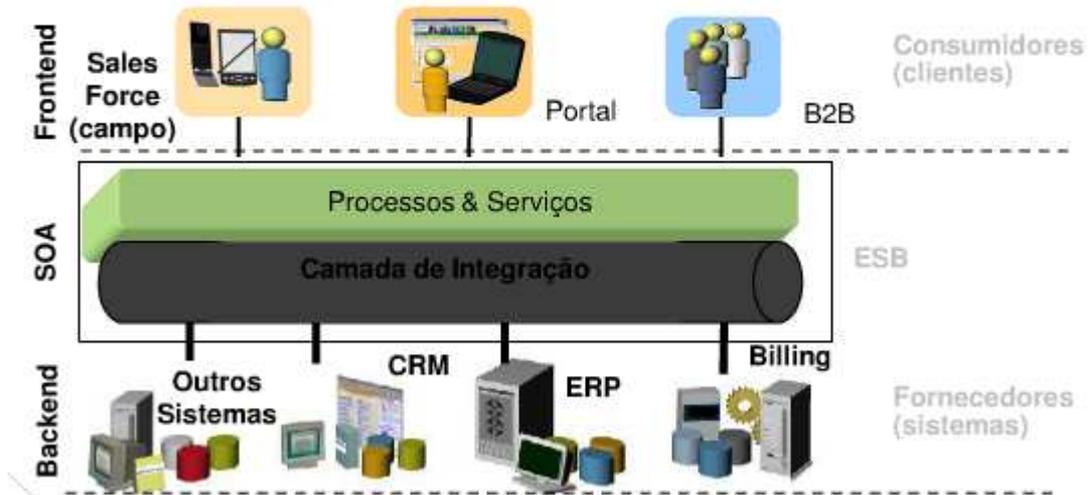


**Figura 2.2 – Aplicações em camadas lógicas, sem implementação da Arquitetura Orientada a Serviço.**

Fonte: <http://www.slideshare.net/Ridlo/soa-fundamentos>

Cada aplicação acima, independente da área de negócio, implementa o seu acesso e integração aos sistemas da empresa e bases de dados. Com isso, para uma mesma necessidade, existem diversas aplicações desenvolvidas, o que gera redundância, aumenta o risco, a complexidade e o custo na manutenção das aplicações.

## Visão da Arquitetura com SOA:



**Figura 2.3 – Aplicações baseadas na Arquitetura Orientada a Serviço.**

Fonte: <http://www.slideshare.net/Ridlo/soa-fundamentos>

Nessa visão acima, as aplicações se comunicam com os diversos sistemas da empresa por meio do ESB (Enterprise Service Bus) como camada de integração, também responsável por coordenar os processos e serviços.

A gestão dos processos de Negócio (Business Process Management) deve ser realizada entre Negócio e TI, a fim de realizar a identificação e definição dos serviços bem como a composição dos fluxos de negócio, promovendo a implementação de SOA de forma adequada às reais necessidades da empresa.

SOA, então é uma evolução dos paradigmas de arquitetura existentes até o momento.

SOA combina elementos de várias disciplinas, tais como modelagem e gestão de processos de Negócio (BPM), Arquitetura de Software, CBD<sup>14</sup>, OO<sup>15</sup>, EAI<sup>16</sup>, computação distribuída e gestão de sistemas. (HEUVEL et al., 2009)

<sup>14</sup> Component-Based Development – Desenvolvimento Baseado em Componente

<sup>15</sup> Object Oriented – Orientação a Objeto – Paradigma de desenvolvimento baseado no conceito de Objeto.

<sup>16</sup> Enterprise Application Integration – Integração de Aplicação Empresariais

## 3. Engenharia de Software de Serviço

Diante de tantas vantagens proporcionadas pela Arquitetura Orientada a Serviço, surge a necessidade de revisitar os métodos e ferramentas utilizadas para o desenvolvimento de sistemas baseados em serviços, uma vez que a Engenharia de Software Tradicional não consegue atender às peculiaridades de tais sistemas: passa-se então a discutir à respeito da Engenharia de Software de Serviços.

A Engenharia de Software de Serviço tem seus primeiros relatos em meados de 2009 em um seminário, e é definida a seguir:

Software service engineering is the science and application of concepts, models, methods, and tools to design, develop (source), deploy, test, provision, and maintain business-aligned and SOA-based software systems in a disciplined, reproducible, and repeatable manner.<sup>17</sup>

(HEUVEL; ZIMMERMANN; LEYMAN; PATRICIA LAGO; SCHIEFERDECKER; ZDUN; AVGERIOU, 2009, p.32)

### 3.1. Princípios

A Engenharia de Software de Serviço procura aplicar o “Pressuposto do Mundo Aberto”<sup>18</sup>, e inicialmente apresenta sete princípios, listados a seguir:

#### I. *Federação Técnica:*

As vantagens proporcionadas pela Arquitetura Orientada a Serviço trazem consigo um aumento na complexidade técnica no desenvolvimento de sistemas. Com isso, o princípio da “Federação Técnica” prega que a Engenharia de Software de Serviço deve fornecer aos

---

<sup>17</sup> Engenharia de Software de Serviço é a ciência e aplicação de conceitos, modelos, métodos e ferramentas para projetar, desenvolver (fonte), implantar, testar, fornecer e manter sistemas de software alinhados ao negócio e baseados em SOA de forma disciplinada, reproduzível e repetíveis.

<sup>18</sup> Pressuposto do Mundo Aberto – Tudo é verdadeiro (possível), a menos que seja provado como falso. No caso da Engenharia de Software de Serviço, o Pressuposto do Mundo Aberto implica que essa nova disciplina não está ‘fechada’, desta forma, não é possível afirmar como não verdadeiro qualquer outro princípio que venha a ser levantado.

arquitetos e engenheiros de serviço, conceitos, linguagens, métodos e ferramentas que os ajudem a gerenciar o aumento de tal complexidade técnica. (HEUVEL et al., 2009)

## II. *Dinamismo (Virtualização):*

O Dinamismo é um dos princípios fundamentais da SSE<sup>19</sup> com relação aos serviços que são agregados na composição de serviço em tempo de execução, bem como o contexto altamente volátil em que os serviços operam. Desta maneira, o princípio ‘dinamismo’ implica que os métodos, técnicas e ferramentas devem lidar com propriedades emergentes e comportamento complexo. (HEUVEL et al., 2009)

## III. *Federação Organizacional:*

A Engenharia de Software de Serviço deve considerar que o desenvolvimento e manutenção de sistemas são, muitas vezes, realizados em unidades organizacionais distribuídas, envolvendo diversas linhas de negócio, outras empresas e instituições governamentais. Considerando isso, o princípio de Federação Organizacional demanda mecanismos sólidos de Governança Distribuída, a fim de gerenciar por exemplo, dificuldade de controle de versionamento de serviços compartilhados por vários consumidores em diferentes domínios organizacionais. (HEUVEL et al., 2009)

## IV. *Limites Explícitos (Contratos):*

Serviços devem possuir limites claros por meio de seu contrato. Quanto mais ricas são as interfaces com cenários e comportamentos, mais robusta e estável será a composição de serviços. (HEUVEL et al., 2009)

## V. *Heterogeneidade:*

As técnicas, métodos e ferramentas da Engenharia de Software de Serviço devem abranger o princípio da heterogeneidade dos serviços, uma vez que deve considerar a infraestrutura e o contexto em que sistemas baseados em SOA operam. A heterogeneidade deve ser considerada em todas as fases do ciclo de vida do desenvolvimento dos serviços, pois pode levantar restrições sobre como os serviços devem ser concebidos e construídos, uma vez que se comunicarão com serviços desenvolvidos em diferentes paradigmas e que operam em diferentes plataformas. (HEUVEL et al., 2009)

---

<sup>19</sup> Software Service Engineering – Engenharia de Software de Serviço

*VI. Alinhamento entre Negócio-TI:*

Considerando as características de componibilidade e reutilização apresentada pelos sistemas baseados em SOA, que visam permitir o alinhamento Negócio-TI, a Engenharia de Software de Serviço deve estar “unificada” com o Business Process Management<sup>20</sup> a fim de garantir não só a Qualidade do Serviço, em nível de sistema, como também a conformidade às especificações do Negócio. (HEUVEL et al., 2009)

*VII. Abordagem Holística:*

A Engenharia de Software de Serviço possui uma característica de natureza holística<sup>21</sup>, sendo assim, demanda uma abordagem interdisciplinar para a análise e racionalização dos Processos de Negócio, concepção de serviços de suporte, implantação, monitoramento e evolução. Isto significa que métodos, modelos e conceitos devem ser integrados e que as ferramentas SSE devem ser interoperáveis. (HEUVEL et al., 2009)

### **3.2.Desafios**

Os principais desafios listados por Heuvel et al. (2011) para a evolução da Engenharia de Software de Serviço são:

- I. Alinhamento Negócio-TI, adaptabilidade;*
- II. Novos modelos e abstrações para representar e lidar com a dinâmica SOA;*
- III. Como lidar com a heterogeneidade;*
- IV. O mapeamento de requisitos;*
- V. Componibilidade;*
- VI. Testes.*

---

<sup>20</sup> Business Process Management – Gerenciamento de Processos de Negócio.

<sup>21</sup> Holística – relativo ao holismo. Que busca tudo abranger.

### 3.3. Engenharia de Software de Serviço x Engenharia de Software Tradicional

A Engenharia de Software é definida por Fritz Bauer como: O estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais.

Segundo Pressman (1995), a Engenharia de Software abrange:

- Métodos: proporcionam os detalhes de “como fazer”. Tarefas que incluem: planejamento, estimativa, análise de requisitos, projeto da estrutura, arquitetura, algoritmo, codificação, teste e manutenção;
- Ferramentas: As ferramentas proporcionam apoio automatizado ou semi-automatizado aos métodos.
- Procedimentos: Constituem o elo de ligação que mantem juntos os métodos e as ferramentas [...] Os procedimentos definem a sequência em que os métodos serão aplicados

Os princípios da Engenharia de Software a serem destacados, citados por Rezende (2005), são:

*I. Formalidade:*

Evitar a dependência de pessoas ou processos em específico.

*II. Abstração:*

Identificar aspectos importantes de determinado fenômeno.

*III. Decomposição:*

Subdividir problemas complexos.

*IV. Generalização:*

Disseminar soluções semelhantes e reutilizar resultados.

*V. Flexibilização:*

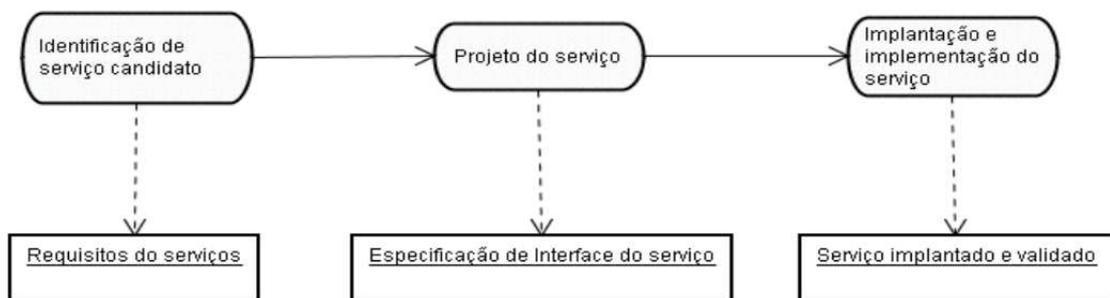
Facilitar eventuais mudanças modulares.

A Engenharia de Software de Serviço busca a definição de métodos, ferramentas e procedimentos que se baseiem não apenas nos princípios da Engenharia de Software Tradicional, mas também nos princípios identificados que visam proporcionar a melhor utilização da Arquitetura Orientada a Serviço.

Percebe-se então que não existem princípios que se contradizem, e sim uma complementação da Engenharia de Software Tradicional com os princípios apresentados pela proposta de Engenharia de Software de Serviço.

### 3.4. Processo de Engenharia de Serviço

A figura abaixo ilustra o processo de engenharia de serviço:



**Figura 3.1 – Processo de Engenharia de Serviço.**

Fonte: <http://www.i2p.com.br/ricardo/eng-orientada-a-servicos.pdf>

#### *I. Identificação do Serviço Candidato:*

Serviços podem ser classificados em “Serviços de utilidade”, “Serviços de Negócio” ou “Serviços de coordenação ou de processo”.

Serviços de utilidade implementam funcionalidades gerais.

Serviços de negócio implementam uma função específica do negócio.

Serviços de coordenação ou de processo apoiam os processos de negócio.

Nessa fase devem ser aplicadas as características de Granularidade e Autonomia de Serviço, bem como podem ser observados os princípios de Alinhamento Negócio-TI e

Heterogeneidade da Engenharia de Software de Serviço, uma vez que tais princípios devem ser observados desde o início do processo pela possibilidade de gerarem restrições ao serviço a ser construído.

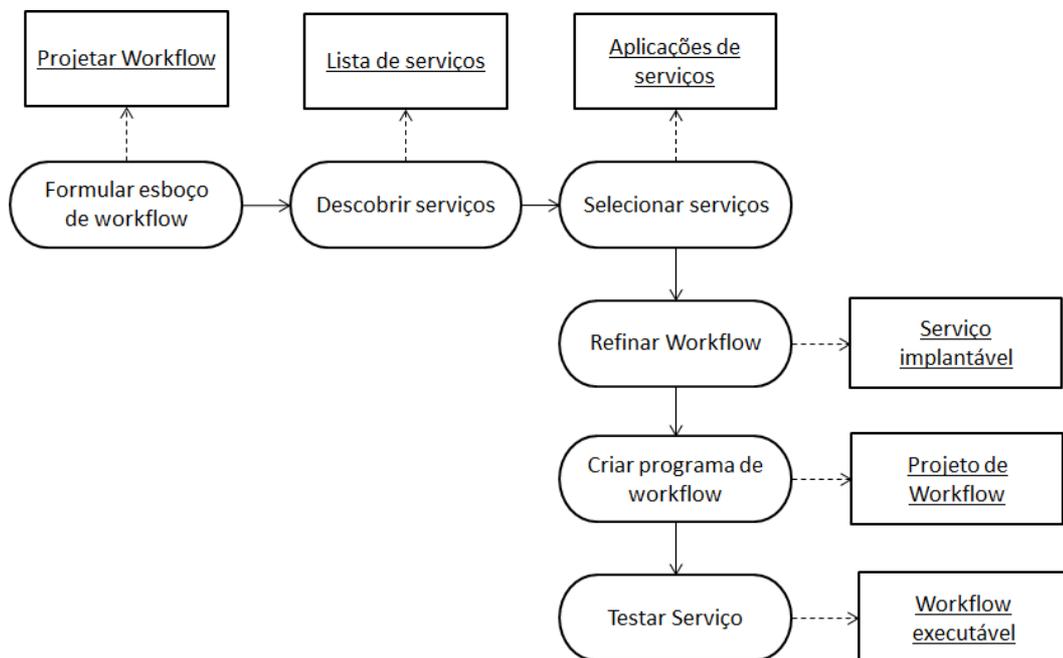
### II. Projeto de Serviço:

Durante o projeto de serviço é realizada a especificação de interface do serviço, aplicando então o princípio “Limites Explícitos” da Engenharia de Software de Serviço, para proporcionar características de Serviço como “Contrato de Serviço Padronizado” e Abstração.

### III. Implementação e Implantação do Serviço:

Nessa etapa, o serviço projetado anteriormente é construído e implantado.

A partir desse momento, o serviço estará disponível para consumo e/ou reutilização para construção de fluxos de negócio ou de outros serviços, por meio da Composição de serviços, ilustrada na figura a seguir:



**Figura 3.2 – Construção de Serviço por Composição.**

Fonte: Adaptado de <http://www.i2p.com.br/ricardo/eng-orientada-a-servicos.pdf>

## 4. Conclusão

A Arquitetura Orientada a Serviço, no que tange ao seu aspecto técnico, já está amplamente difundida, no entanto, considerando as fases que antecedem à implementação de sua unidade básica, o serviço, observa-se grandes desafios, pois suas características produzem alto grau de complexidade, tanto na identificação de sua granularidade quanto no próprio alinhamento com o Negócio, a fim de proporcionar flexibilidade, reutilização, componibilidade e promover produtividade e redução de custos às empresas nos investimentos em TI.

Percebe-se então, a carência de métodos, ferramentas e procedimentos específicos para esse paradigma de desenvolvimento, uma vez que a Engenharia de Software Tradicional não supre determinadas exigências da Arquitetura Orientada a Serviço.

A Engenharia de Software de Serviço, então, é baseada em princípios que enfatizam a necessidade de se adotar métodos e ferramentas específicas para o desenvolvimento de sistemas baseados em serviço, e que complementam os princípios apresentados pela Engenharia de Software Tradicional, a fim de construí-los abrangendo todas as características inerentes à Arquitetura Orientada à Serviço, e contribuir, desta forma, para os resultados das empresas que a adotam.

[...] software service engineering as an emerging discipline that entails a departure from traditional software engineering disciplines [...] (HEUVELet al., 2009, p.26)

## 5. Referências Bibliográficas

*LUCCI, Elian Alabi. A Era Pós-Industrial, a sociedade do Conhecimento e a Educação para o Pensar.* Disponível em: <<http://www.mendeley.com/research/era-psindustrial-sociedade-conhecimento-ea-educao-para-o-pensar/>> Acesso em: 08 nov. 2011.

*KRAFZIG, D., BANKE, K., SLAMA, D. Enterprise SOA: Service-Oriented Architecture Best Practices.* 2004.

*HEUVEL, Willem-jan Van Den et al. Software Service Engineering: Tenets and Challenges.* Disponível em: <<http://www.cs.rug.nl/~paris/papers/PESOS09.pdf>>. Acesso em: 09 nov. 2011.

*BIEBERSTEIN, Norbert et al. Executing SOA: A Practical Guide for the Service-Oriented Architect.* Pearson Education, 2008.

*TAURION, Cezar. SOA Arquitetura Orientada a Serviços.* Disponível em: <<http://www.smashwords.com/books/view/4104>>. Acesso em: 20 nov. 2011.

*COSTA, Ivanir; CARVALHO NETO, Antonio Rodrigues. Tendências Sobre a Arquitetura Orientada a Serviços - SOA.* Disponível em: <[http://www.abepro.org.br/biblioteca/enegep2007\\_tr670485\\_9968.pdf](http://www.abepro.org.br/biblioteca/enegep2007_tr670485_9968.pdf)>. Acesso em: 20 nov. 2011.

*SANTOS JUNIOR, Alfredo Luiz Dos. Integração de Sistemas com Java.* Rio de Janeiro: Brasport, 2007.

*EVDEMON, John. Princípios do design de serviço: padrões e antipadrões de serviço.* Disponível em: <<http://msdn.microsoft.com/pt-br/library/ms954638.aspx>>. Acesso em: 26 nov. 2011.

*ERL, Thomas. Service-Oriented Design Principles.* Disponível em: <<http://www.soapprinciples.com/>>. Acesso em: 27 nov. 2011.

*JOSUTTIS, N. M. SOA in practice: The Art of Distributed System Design.* O'Reilly, 2007.

*FURTADO, Camille et al.* **Arquitetura Orientada a Serviço - Conceituação.** Disponível em: <<http://np2tec.uniriotec.br:9093/np2tec/publicacoes>>. Acesso em: 27 nov. 2011.

*MACHADO, João Coutinho.* **Um estudo sobre o desenvolvimento orientado a serviços.** Disponível em: <[http://www2.dbd.puc-rio.br/pergamum/tesesabertas/0210486\\_04\\_cap\\_02.pdf](http://www2.dbd.puc-rio.br/pergamum/tesesabertas/0210486_04_cap_02.pdf)>. Acesso em: 28 nov. 2011.

*GOUVEIA, Miguel; GOUVEIA, Vitorino.* **Service Oriented Architecture (SOA), Desafios para o Processo de Desenvolvimento de Software.** Disponível em: <<http://isg.inesc-id.pt/>>. Acesso em: 28 nov. 2011.

*PRESSMAN, Roger S.* **Engenharia de Software.** São Paulo: Makron Books, 1995.

*REZENDE, Denis Alcides.* **Engenharia de Software e Sistemas de Informação.** Rio de Janeiro: Brasport, 2005.

*LIMA, Ricardo Roberto de.* **Engenharia de Software Orientada a serviços.** Disponível em: <<http://www.i2p.com.br/ricardo/eng-orientada-a-servicos.pdf>>. Acesso em: 04 dez. 2011.