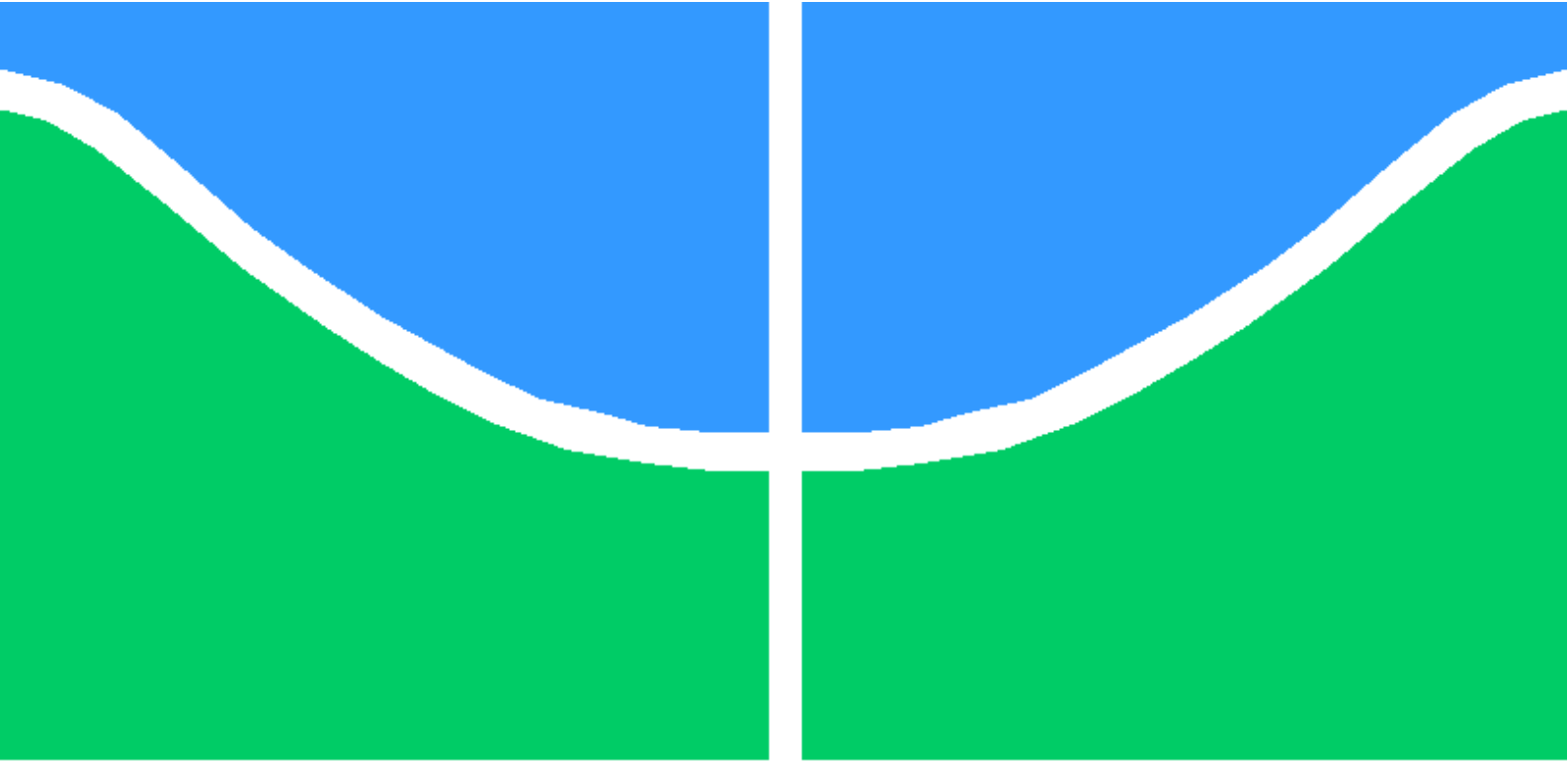

TRABALHO DE GRADUAÇÃO

A Systematic Mapping Study: Qualidade de Serviço em Computação Orientada a Serviços

Danilo Filgueira Mendonça

Brasília, Março de 2012

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASILIA
Faculdade de Tecnologia

TRABALHO DE GRADUAÇÃO

A Systematic Mapping Study: Qualidade de Serviço em Computação Orientada a Serviços

Danilo Filgueira Mendonça

Relatório submetido ao Departamento de Engenharia Elétrica
como requisito parcial para obtenção do grau de
Engenheiro de Redes de Comunicação

Banca Examinadora

Profa. Dra. Genaína Nunes Rodrigues , _____
CIC/UnB
(Orientadora)

FICHA CATALOGRÁFICA

MENDONCA, D. F.. A Systematic Mapping Study:
QOS em Computação Orientada a Serviços [Distrito Federal] 2012.
v, 52p. (ENE/FT/UnB, Engenheiro de Redes de Comunicação, 2012)
Monografia de Graduação - Universidade de Brasília. Faculdade de Tecnologia.
Departamento de Engenharia Elétrica.

1. Computação Orientada a Serviços	2. SOC
3. Arquitetura Orientada a Serviços	4. SOA
5. Qualidade de Serviços	6. QOS
I. ENE/FT/UnB	II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

MENDONCA, D. F. e (2012). A Systematic Mapping Study: QOS em Computação Orientada a Serviços Monografia de Graduação, Publicação ENE 01/2012, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 52p.

CESSÃO DE DIREITOS

NOMES DOS AUTORES: Danilo Filgueira Mendonça e .

TÍTULO: A Systematic Mapping Study: QOS em Computação Orientada a Serviços

GRAU / ANO: Engenheiro de Redes de Comunicação / 2012.

Danilo Filgueira Mendonça
SQN 303 BL F APTO 605 - ASA NORTE
CEP 70735-060 - Brasília - DF - Brasil.

Dedicatória

Danilo Filgueira Mendonça

Agradecimentos

Danilo Filgueira Mendonça

RESUMO

A Computação Orientada a Serviços surgiu com o intuito de prover maior eficiência à produção, provisão e consumo de recursos computacionais, especialmente os *softwares*, que passam a compor unidades coesas e granulares de lógica capazes de se intercomunicarem e de formarem novas soluções por meio de sua composição em novos rearranjos, aumentando o reuso, a agilidade, o retorno de investimento e o alinhamento da TI com os processos de negócio. Dada a incerteza a respeito do estado das pesquisas relacionadas ao tema, detectou-se a necessidade de uma classificação de estudos que possibilite identificar de forma esquemática os tópicos existentes e que aponte tendências, atores, tipos de pesquisa e quais subáreas receberam maior ênfase em detrimento daquelas que ainda carecem de avanços. A partir do Systematic Mapping Study, que envolve a busca por estudos em sistemas de registro e busca de publicações e em fóruns de interesse de modo a classifica-las segundo facetas escolhidas, avaliamos artigos que tratam da qualidade de serviços na Computação Orientada a Serviços. Como resultado...

ABSTRACT

English version

SUMÁRIO

LISTA DE FIGURAS

NOTAÇÕES

Siglas

RSO	*Redes Sociais Online*
OSN	*Online Social Networks*
WEB	*Sistema de Documentos de Hipertexto Interligados que podem ser executados via Internet.*

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

Services, SOA, Web Services, SOAP, REST, Service Orientation, Cloud, SaaS. Nos últimos anos esses e outros acrônimos tornaram-se frequentes na tecnologia da informação. O surgimento de um novo paradigma, impulsionado pelo amadurecimento da internet e pela proximidade crescente entre negócios e TI, criou novos caminhos e oportunidades para trabalhos de desenvolvimento e pesquisa. Nesse sentido, um grande número de estudos foram e vem sendo conduzidos com foco nos diversos aspectos da computação orientada a serviços, tais quais arquitetura, modelos, métodos, processos, ferramentas diversas, frameworks, métricas, problemas solucionados e ainda vigentes. Desta forma, a intenção daqueles interessados em iniciar suas atividades na área fica comprometida pela dificuldade em obter informações claras sobre o atual estado da arte, os desafios, os temas mais abordados e aqueles com deficit em pesquisas. Esses dados são cruciais para que esforços sejam bem direcionados e para que a ciência caminhe em cooperação e com eficiência.

O ambiente proposto pelas arquiteturas orientadas a serviço está sujeito a condições particulares diferentes daquelas já estudadas e conhecidas em outros paradigmas, havendo variáveis que elevam a complexidade da análise de parâmetros de qualidade, tanto na fase de planejamento quanto em fase de execução por meio do monitoramento e da gerência dos serviços, sendo esse um obstáculo sólido à adoção de arquiteturas como o SOA. Nesse sentido, o presente estudo visa mapear publicações cujo foco está na qualidade de serviços em SOC, contemplando cenários com ou sem o uso de SOA, proporcionando uma redução da incerteza quanto ao atual estado de desenvolvimento da ciência contribuinte ao tema abordado e quanto aos desafios e avanços já conquistados.

Um Mapping Study visa classificar de forma sistemática e ampla um conjunto de estudos. Dada a grande quantidade de publicações no escopo da orientação a serviços, sua metodologia ágil e que permite a análise de um maior número de estudos [2] justifica sua escolha em detrimento de outras metodologias, sendo a mais conhecida o Systematic Literature Review. Essa última exige uma análise minuciosa e detalhada de cada publicação, o que requer um esforço considerável e inviabiliza a inclusão de um grande número de publicações num quadro de poucos pesquisadores. Assim, dados os fatos citados e o interesse em se obter uma classificação ampla e significativa da ciência relacionada à orientação a serviços, de cará-

ter inicial e que irá servir de subsídio a outros estudos, este trabalho de conclusão de curso em Engenharia de Redes de Comunicação realiza um Systematic Mapping Study com foco em orientação a serviços. Segundo [3], devido ao crescente acordo na implementação e gerência de aspectos funcionais de serviços, tal qual a adoção de WSDL para a descrição, SOAP para troca de mensagens, ou WS-BPEL para a composição, os interesses de pesquisadores estão se voltando aos aspectos não funcionais de aplicações orientadas a serviços. Visando essa constatação, nosso mapeamento irá focar a questão de qualidade, ou aspectos não funcionais, sobretudo a qualidade de serviços, termo aqui empregado de forma literal e posterior ao termo QoS, uma vez que os principais agentes do paradigma em questão são, coincidentemente, denominados serviços.

Sobre a motivação pessoal desse trabalho é preciso destacar meu interesse na Engenharia de Software, principalmente em seu aspecto distribuído, uma vez que possibilita novos patamares de interoperabilidade, aumentando a gama de soluções da tecnologia da informação e tomando proveito do avanço nas tecnologias de comunicação de dados, o que envolve questões próprias da Engenharia de Redes. Além disso, acredito na necessidade de uma melhor e mais eficiente organização e uso da tecnologia da informação, sendo essas premissas presentes no paradigma em questão. Assim, vejo justificada a iniciativa deste trabalho, que almeja gerar contribuições para ambas as áreas.

1.2 OBJETIVOS

1.2.1 Objetivos Gerais

O objetivo do presente mapeamento de estudos é esclarecer o paradigma da orientação a serviços por meio de uma classificação ampla e sistemática, obtendo informações sobre frequências de publicações, áreas e tópicos de pesquisa, enfoques, tipos de contribuições dadas, os agentes e fóruns envolvidos, além de modelos arquiteturais e contextos relacionados. Deseja-se obter resultados gráficos que ilustrem de maneira intuitiva e lógica, através dos dados coletados, as classificações propostas, contribuindo assim para a obtenção de respostas às questões de pesquisa que guiam e motivam este mapeamento de estudos.

1.2.2 Objetivos Específicos

Objetiva-se obter melhores definições acerca da história recente e das atuais tendências para a arquitetura orientada a serviços, ou SOA, uma vez que há indícios preliminares de que o termo, após grande ênfase

tanto por parte de pesquisadores quanto por parte de vendedores, recebeu menor atenção nos últimos anos, sobretudo após o ano de 2008. Em contrapartida, deseja-se igualmente mapear a evolução das pesquisas relacionadas ao Cloud Computing, termo também abrangido pela computação orientada a serviços e que, por sua vez, vem sendo frequentemente citado em jornais, artigos e fóruns de discussões da área.

Em relação aos aspectos de qualidade, tem-se por objetivo compreender que tipo de abordagens vem sendo tomadas para a garantia de QoS em SOC, além de identificar quais atributos tem maior importância para as pesquisas, quais representam os maiores desafios para a concretização da adoção deste paradigma e quais são pouco abordados.

Por fim, é objetivo que este estudo sirva de embasamento para posteriores trabalhos de pesquisa, dado que a metodologia utilizada tem um caráter exploratório e anterior a estudos mais focalizados.

1.3 TRABALHOS RELACIONADOS

1.4 ORGANIZAÇÃO DO TRABALHO

2 EMBASAMENTO TEÓRICO

2.1 SERVICE ORIENTED COMPUTING

A tecnologia da informação já se consolidou como ferramenta essencial para as instituições, afetando rotinas, ações, estruturas e modelos de negócios. Os alto custos associados ao consumo e provimento de recursos de TI e a necessidade de uma maior agilidade para responder às mudanças casam-se às propostas da computação orientada a serviços, ou SOC, cujo paradigma de design é composto por princípios que, quando absorvidos, criam condições para o cumprimento dos objetivos e benefícios almejados pela orientação a serviços, entre eles o aumento do retorno de investimento, da agilidade, da reusabilidade e da interoperabilidade, assim como um maior alinhamento entre TI e modelo de negócios.

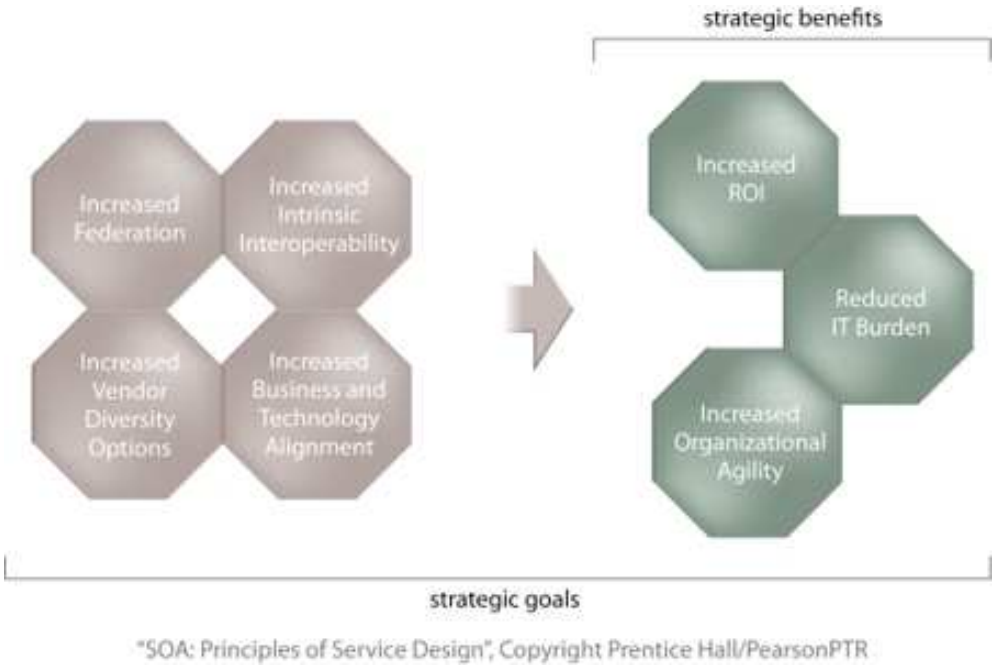

Figura 2.1: Princípios relacionados ao SOA e seus benefícios

O paradigma de design da orientação a serviços norteia a configuração do suprimento e consumo dos recursos de TI, em especial as soluções lógicas ou softwares. Apesar do sucesso de outros paradigmas na computação, por exemplo a orientação a objetos, nota-se que problemas distintos, ou seja, da eficiência, agilidade e retorno de investimento na composição de aplicações, ainda deveriam ser atacados. Assim, as aplicações silos com pouca interoperabilidade e sem reuso passaram a ser vistas como um modelo ineficiente, visto que aumentam o custo de desenvolvimento de novas aplicações a longo prazo e reduzem

drasticamente a agilidade ou tempo de resposta às novas demandas, sobretudo quando os modelos de negócio exigem maior flexibilidade. Dessa forma, os benefícios idealizados pelo SOC fizeram desse paradigma um foco de estudos em diversos centros acadêmicos, sobretudo à partir do surgimento da arquitetura orientada a serviços, ou SOA. Também é notável a adesão de grandes vendedores da indústria de TI a essa tecnologia, fomentando tanto pesquisa quanto sua adoção em instituições públicas e privadas.

O termo SOC representa um aglomerado de conceitos relacionados à orientação a serviços [4]. Além de um paradigma de design, engloba princípios, padrões, governança e arquitetura, notavelmente o SOA. Nesse trabalho, o termo SOC poderá ser usado para se referir de forma genérica à orientação a objetos como um todo.

2.1.1 Service Orientation History

Historicamente, o paradigma de orientação a serviços tem como base diversos outros paradigmas e arquitetura. Por meio da adaptação e amadurecimento de conceitos já experimentados em TI e dada a conjectura de que alguns objetivos poderiam ser melhor conquistados, formou-se o que hoje define a orientação a serviços em termos de princípios de design, arquitetura e tecnologias capazes de implementa-los.

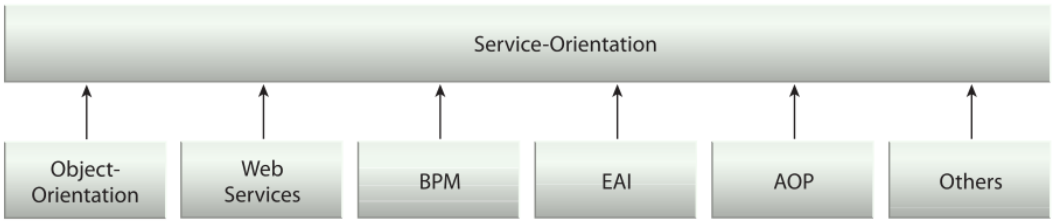

Figura 2.2: Origens da Orientação a Serviço

2.1.1.1 Component Based Architecture

Entre as mais importantes influências está a arquitetura baseada em componentes. Com ela, a orientação a serviços compartilha visões, uma vez que ambas se sustentam sobre os conceitos de unidades lógicas auto-contidas, auto-descritas, modulares, encapsuladas, que fazem uso de interfaces, contratos e especificações com possibilidade de composição com outras unidades.

No entanto, serviços diferenciam-se fundamentalmente de componentes. Enquanto componentes variam entre modelos *white box, gray box ou black box*, de acordo com o nível de customização realizável, os serviços são sempre hermeticamente encapsulados e distribuídos em forma de *black boxes*. Nos primeiros,

foca-se na especificação técnica do conjunto de funcionalidades que desempenham, permitindo o uso das mesmas nos códigos que as invocam uma vez que os componentes tenham sido devidamente importados ou ligados à aplicação. Sua especificação pode incluir também uma definição abstrata de sua estrutura interna [5]. Para os últimos, contratos são estabelecidos em conjunto às descrições das funcionalidades expostas pelas interfaces do serviço. Segundo [6], a interação dos serviços ocorre de forma desacoplada por meio do uso de parâmetros pré-estabelecidos em sua descrição.

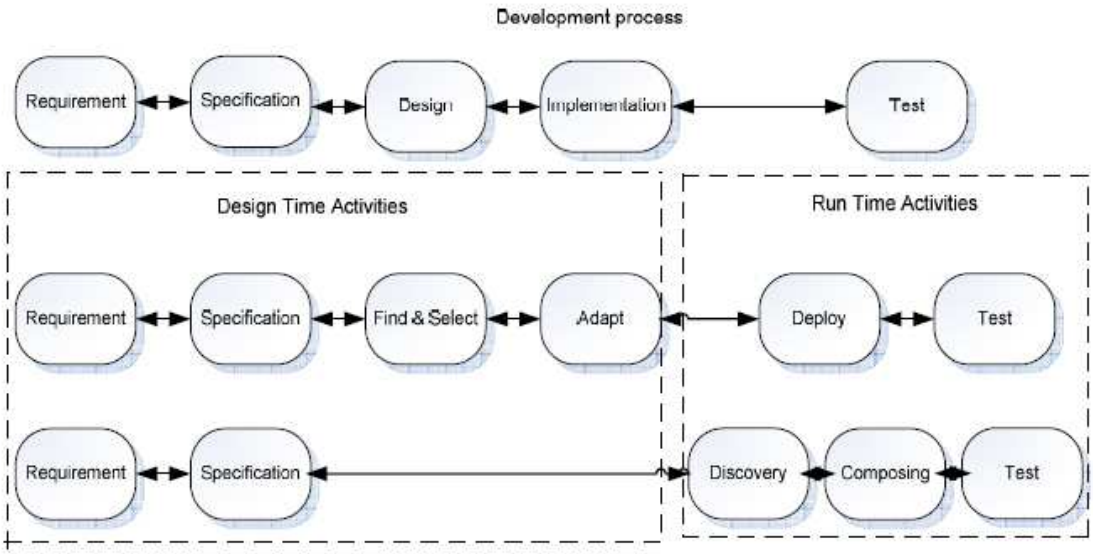

Figura 2.3: Comparação entre atividades típicas de componentes e serviços

A composição dos serviços é feita de forma dinâmica em tempo de execução, enquanto a composição de componentes é feita na fase de design ou em tempo de execução. Isso relaciona-se ao tipo de acoplamento entre provedor e consumidor, sendo essa a principal diferença entre ambos paradigmas. Nos componentes é frequente um maior acoplamento, de forma que um grau mais elevado de cumplicidade entre provedor e consumidor é exigida de acordo com o tipo de interação. Outro fator está na dependência quanto ao modelo de componentes usado, que deve ser compatível. Em contraste, nos serviços o baixo acoplamento entre provedor e consumidor é um princípio a ser seguido e um de maior importância no paradigma de orientação a serviços, havendo transparência no processo de troca de mensagens. Dessa forma, o provedor irá desfrutar de liberdade e flexibilidade para a escolha da tecnologia para implementação, assim como o contratante irá somente se ater, funcionalmente, à interface. Essa separação propicia a criação de serviços abstratos e reutilizáveis, o que segue princípios de design da orientação a serviços. Para componentes, é possível uma optimização da performance em detrimento da flexibilidade por meio da

composição na fase de design, mas dificilmente se concebe um cenário de integração de um componente sem uma documentação ou até mesmo a comunicação com a equipe responsável pelo seu provimento.

Mesmo com modelos que definem a distribuição de componentes, i.e. J2EE e CORBA, os resultados obtidos são de naturezas distintas. O processo de integração remota previsto pela distribuição de componentes mantém o aspecto citado para a sua especificação, uma vez que irá, de forma refinada, distribuir modelos de objetos, seus estados e suas propriedades, enquanto os serviços estão num patamar menos elaborado de integração, fornecendo funcionalidades coesas em alto nível no padrão requisição e resposta. De fato, há casos em que um serviço irá fazer uso de componentes em sua implementação, que por conceito independe de sua descrição, desde que mantenha a funcionalidade por ela exposta. Portanto, componentes e serviços se situam em domínios diferentes, assim como as necessidades atendidas e os problemas por eles enfrentados [7]. Em suma, os maiores ganhos dos serviços em relação aos componentes estão na maior interoperabilidade, na maior gama de possibilidades para composições, uma vez que permite integração com com serviços de terceiros e por fim no menor acoplamento entre provedor e consumidor.

2.1.1.2 Object Orientation

A orientação a objetos também teve grande influência na orientação a serviços. Muitos conceitos por ela reforçados tem precedência nesse paradigma, entre eles o alinhamento do modelo de negócio e da TI, visto que as classes são moldadas à partir de conceitos e objetos reais, muitas vezes representações do próprio negócio. O aumento da robustez, uma vez que a orientação a objetos mantém um forte processo de design da solução com uso de diagramas UML e possui um complexo modelo para exceções e rotinas de teste. O aumento da extensibilidade, haja vista a característica modular das classes e as diversas possibilidades de herança, polimorfismo e associações capazes de estender o escopo da solução. Da flexibilidade, dado o uso de encapsulamento e abstração, sendo possível a adaptação do funcionamento a novas realidades. Por fim, da reusabilidade e da produtividade, dada a existência de classes abstratas e de códigos genéricos e reutilizáveis [4].

Uma notável diferença entre os dois paradigmas está no escopo onde atuam. A orientação a objetos visa estruturar soluções localizadas e podem vir a ser aplicada diversas vezes, podendo englobar todo o cenário da instituição através de diferentes ciclos. A orientação a serviços desde sua aplicação inicial irá atuar em porções mais extensas, podendo ou não chegar a todo o contexto de soluções de tecnologia da organização. Isso se deve a alguns princípios do SOC, que evita a formação de aplicações silos e fomenta a formação de um inventário de serviços granulares e interoperáveis de modo a viabilizar a composição dos mesmos com

base na reusabilidade. Assim, é possível afirmar que a aplicação da orientação a objetos geralmente leva a soluções isoladas ou à criação de um nível limitado de reusabilidade e abrangência, enquanto a aplicação de orientação a serviços tende a transformar o cenário como um todo englobando unidades lógicas que, se obtiverem êxito em alcançar os objetivos e princípios do design da orientação a serviços e se vencerem as implicações técnicas consequentes, irão disponibilizar um conjunto de soluções flexíveis e que respondem a novas demandas de negócio com maior velocidade e menor custo.

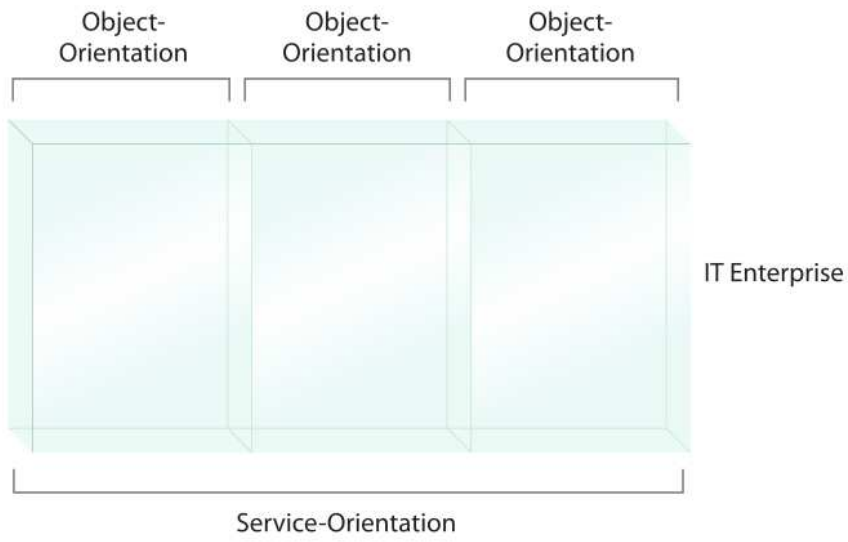

Figura 2.4: Comparação entre escopos da Orientação a Objetos a da Orientação a Serviços

Outro importante modelo arquitetural influente no SOC é a o EAI ou Enterprise Application Integration. Esse conjunto de princípios arquiteturais foca na integração de soluções heterogêneas de uma empresa. Seu principal componente é denominado *middleware*, ou aplicação intermediária, e tem a função de transportar mensagens entre as aplicações de maneira transparente e irrestrita. O motivo pelo qual essa arquitetura influenciou o surgimento da orientação a serviços está em seu caráter distribuído e integrador, possibilitando que unidades funcionais de lógica troquem mensagem e cooperem formando novas aplicações e dando vazão às demandas de negócios com maior agilidade. No entanto, é visível que mesmo havendo uma integração, as aplicações envolvidas ainda se situam num modelo de aplicações silos e com pouco potencial de reuso e de serem compostas de maneiras diversas e flexíveis. Assim, visionando as possibilidades trazidas por um inventário de soluções coesas, abstratas e de fácil composição, o SOC inovou com os princípios de design em que se baseia para criação e mantenimento de serviços, forma como passaram a ser denominadas as unidades lógicas disponíveis e integradas pela camada intermediária.

2.1.1.3 Web Services

Seria imprescindível, na tarefa de descrever as origens da computação orientada a serviços, deixar de citar a importante contribuição dos Web Services. Mais que uma influência, os Web Services estão na raiz do funcionamento da arquitetura SOA, mesmo sendo essa agnóstica à tecnologia usada na implementação de seus serviços.

Web Services surgiram anteriormente ao SOC, oferecendo uma integração transparente entre aplicações distribuídas por meio de um sistema de troca de mensagens que faz uso de protocolos padronizados e especificações. Em sua primeira geração, tentativas foram feitas com o uso direto de Remote Procedure Calls, ou RPC. Apesar desse modelo já ser conhecido e usado, seu uso em Web Services não obteve apoio devido ao acoplamento resultante de configurações específicas para cada linguagem e logo deixou de ser praticado dessa forma. Também foram desenvolvidas as especificações centrais, tais quais:

- WSDL, liguagem responsável pela descrição pública das funcionalidades de um serviço em formato XML
- UDDI, que consiste num registro de referência para a publicação de serviços e que opera ele próprio por meio de Web Services via SOAP, com estrutura baseada em XML e que pode ser utilizado publicamente através da internet ou internamente a uma organização.
- O SOAP, protocolo de troca de mensagens em formato XML e que possibilita o uso protocolos de transporte conhecidos da internet, i.e. HTTP, SMTP, ou outros independentes, passou a ser usado em Web Services mais simples e logo em Web Services descritos com WSDL e associados a repositórios UDDI, chamados de *Big Web Services* [8]. Além de promover a integração de mensagens, o SOAP também possibilita a interoperabilidade de chamadas RPC por meio do encapsulamento dessas chamadas.
- WS-I, que tem a função de definir referências e guiar a adoção de especificações de Web Services com intuito de manter a interoperabilidade por meio de perfis denominados WS-I Profiles. Os *profiles* são diretrizes do tipo melhores práticas para um grupo selecionado de especificações, em versões estabelecidas, com objetivo de assistir à comunidade na criação e implantação de Web Services interoperáveis [**?**].

A segunda geração de Web Services trouxe avanços em segurança, transações entre serviços e garantias na troca de mensagens, entre outros [4]. Essas tecnologias e especificações relacionadas seguem a

nomeclatura WS-* e tem a característica de serem passíveis de composição entre si e fornecerem um rico conjunto de ferramentas para o ambiente de Web Services. Estão inclusas as seguintes especificações:

- WS-Security provê melhorias ao protocolo SOAP visando obter integridade e confidencialidade em sua troca de mensagens. Segundo a especificação 1.1, permite com que diversos modelos de segurança e de criptografia sejam utilizados. Também define um mecanismo para a associação de *tokens* de segurança ao conteúdo das mensagens, sem restringir, no entanto, os tipos de *token* a serem utilizados, entre outros mecanismo para a segurança de Web Services. Outras especificações de segurança podem e, em alguns casos, devem ser utilizadas em complemento ao WS-Security, ou mesmo como alternativa, o que ocorre com quando a segurança é obtida através da camada de transporte por meio de HTTPS. Entre as especificações complementares estão o WS-SecureConversation, o WS-Trust e o WS-Authorization.

- WS-ReliableMessaging age em termos da confiança da troca de mensagens. Dessa forma, irá garantir que o transporte da mensagem ocorra de maneira confiável mesmo em situações de falha por parte dos envolvidos, e com maior frequência, falhas de rede. Em seu funcionamento, o WS-ReliableMessaging irá informar, por meio de exceções, a ocorrência de mensagens não entregues ao destinatário, possibilitando o seu reenvio.

- WS-Policy fornece um mecanismo com estrutura em XML para a publicação de políticas de QoS para Web Services em termos de requerimentos, capacidades, ou ambos. Trata-se de uma especificação que trabalha em conjunto a outras especificações e mecanismos para prover negociação de atributos de qualidade na escolha de Web Services.

Atualmente, as especificações WS-* continuam a evoluir e a agregar novos mecanismos, modelos e métricas. Entretanto, críticas quanto à complexidade e enrijecimento da arquitetura como consequência ao uso destes padrões levaram ao aumento de demanda por outros modelos. O REST, ou Representational State Transfer, trata da transferência de representações de recursos usando um conjunto restrito de ações, denominadas verbos, usualmente por meio do protocolo HTTP e permite a criação de RESTfull Web Services. Nesses últimos há uma liberdade de diversas decisões arquiteturais, ou *fredom-from-choice*, mantendo-se a premissa de requisições *stateless*, um vez que toda informação necessária ao processamento da requisição está nela contido. Em REST, cada recurso irá possuir uma única identificação URI, por meio da qual operações, i.e. GET, POST, PUT e DELETE, serão realizadas aos recursos. Inclui os problemas encontrados com o uso dessa arquitetura a ausência de suporte a vários atributos QoS. A segurança, por

exemplo, fica a cargo do protocolo de transporte, que na prática é limitado ao HTTPS. Não há garantias para as mensagens, salvo em implementações próprias, e a composição é feita por meio de *mashups*, termo proveniente da Web 2.0 e que define uma combinação de funcionalidades entre fontes diversas e heterogêneas de maneira livre e sem especificações. Portanto, caberá ao usuário dos serviços elaborar seu modelo de composição ou simplesmente não utilizar nenhum modelo, algo mais propício ao ambiente de aplicações de sítios da internet em pequeno e médio porte que ao ambiente controlado encontrado em organizações.

Atualmente o debate entre ambas tecnologias para implementação de Web Services aponta para a contextualização de seu uso: cenários mais simples e ad-hoc tendem a obter melhor eficiência com RESTFull Web Services. Em casos de maior complexidade que possam incluir transações, garantias de segurança e de outros aspectos de QoS que encontram correspondência nas especificações WS-* e que requerem o mantenimento da interoperabilidade, sobretudo em ambientes corporativos, o uso de Big Services mostra-se mais viável, especialmente a longo prazo, quando o custo inicial mais elevado de sua adoção é superado pelos custos associados a soluções customizadas e proprietárias na arquitetura REST. [8].

2.1.1.4 Aspect Oriented Programming

O AOP define como principal objetivo a separação de tópicos, ou *separation of concerns*, de forma a identificar interesses comuns entre aplicações que poderão ser modularizados e reutilizados. Em geral a modularização é feita em tópicos que se espalham por diversas camadas de abstração, ou *crosscutting concern*.

A principal correlação da AOP com a orientação a serviços está no princípio da reutilização de unidades lógicas que são abstratas e agnósticas à lógica de negócio e ao restante da aplicação, proposta que se encontra definida em princípios de design dos serviços.

2.1.1.5 Business Process Model

Por BPM entende-se a disciplina composta por técnicas e métodos voltados ao design, gerenciamento, aperfeiçoamento e controle de processos de negócio, ou Business Process, com intuito de atingir os objetivos de uma organização [9]. Um BP é caracterizado pelo conjunto de práticas e atividades encarregadas de produzirem um produto ou serviço. É comum representa-los em gráficos com sequências de atividades, que se iniciam no objetivo e finalizam em seu cumprimento. O grande foco do BPM está na otimização

dos resultados obtidos pelos BPs e no alinhamento da TI com os objetivos estratégicos da organização. É, em essência, uma extensão ou amadurecimento de práticas já conhecidas pela administração desde que surgiram os conceitos relacionados ao processo produtivo, mas o grande salto qualitativo se deu com o uso da TI para a automação da gestão de BPs, sendo a primeira experiência conhecida desenvolvida para General Electric em 1954 [10].

Uma das atividades centrais do BPM é o modelamento, ou Business Process Modeling, que realiza o descobrimento e modelagem de BPs numa organização, além de melhorias a processos já existentes. A representação é feita pelo padrão BPMN com o uso de diagramas gráficos que se assemelham aos diagramas de atividade da linguagem UML [11].

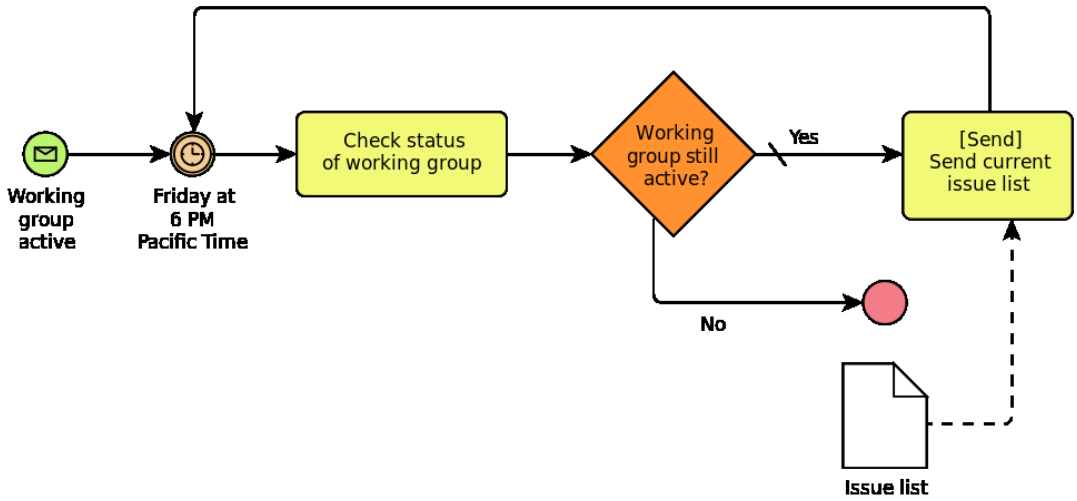

Figura 2.5: Modelagem de um BP por meio de um diagrama BPMN

Outra importante atividade está no monitoramento de processos, ou BAM, em geral mais completa e detalhada que o monitoramento obtido com ferramentas do Business Process Management Suit, o BPMS. Assim, o rastreamento de informações acerca do serviço oferece estatísticas sobre sua performance e estado.

Um dos princípios da orientação a serviços está no alinhamento entre modelo de negócios e tecnologia da informação. Como propósito, o BPM possibilita que processos desse modelo estejam mais visíveis, adaptáveis e extensíveis, o que aumenta o tempo de resposta e oferece ferramentas para o mapeamento entre requisitos de negócios e recursos da TI, o que provê alinhamento entre os dois mundos e um maior valor agregado às soluções. O BPM ocupa o papel de organizar uma abstração de alto nível dos processos de negócio, enquanto o SOA irá se adequar no suporte a esses processos com características de agilidade, flexibilidade e eficiência que serão passadas ao negócio e consequentemente à organização. Dessa forma,

a orientação a serviços por meio do SOA e de seus princípios de design surgiu num contexto que já se mostrava apto a recebe-la e não como uma revolução daquilo que já existia.

2.1.2 Service Oriented Design

O principal resultado da aplicação do design estabelecido pelo SOC está na criação de unidades lógicas denominadas serviços. Assim, ao invés de aplicações isoladas que trazem consigo todas as funcionalidades, são criadas unidades coesas, granulares, abstratas e de baixo acoplamento e alta interoperabilidade, de operação preferencialmente *stateless* e cujo acesso é feito por uma interface que encapsula sua lógica e seu funcionamento bem descrito e garantido por um contrato. Com isso, é possível realizar combinações ágeis que irão servir uma ou, idealmente, várias aplicações por meio do reuso e da composição orquestrada de serviços.

Os princípios adotados pelo service oriented design encontram-se bem definidos e

2.1.3 Service Oriented Architecture

Com base nos princípios de SOC, criou-se uma arquitetura responsável por guiar o processo de adoção desse paradigma. O SOA, ou *Service Oriented Architecture*, define uma série de processos, papéis, regras, modelos e camadas, entre outros componentes da arquitetura, propondo transformações importantes no modus operandi no setor de tecnologia da informação. Dada sua abrangência e complexidade de fundamentos, o SOA é protagonista no avanço do SOC, tendo seu conceito se difundido de forma mais ampla que o primeiro, haja vista que recebeu grande atenção do meio acadêmico, como ilustra nosso mapeamento de estudos, além de ter sido um produto de destaque para grandes vendedores da indústria de TI. Entretanto, os mesmos agentes responsáveis pela comercialização de SOA, também contribuíram para a vulgarização do termo, ao ponto do mesmo ter se tornado ambíguo e com significados diversos. Um esforço foi feito visando estabelecer um significado único para a arquitetura, conhecido por *SOA Manifesto*. Entre as prioridades do SOA apontada pelo manifesto estão: o valor de negócio sobre a estratégia técnica adotada, os objetivos estratégicos sobre os benefícios específicos de projeto, a interoperabilidade intrínseca sobre as integrações customizadas, os serviços compartilhados sobre implementações com propósito específico, a flexibilidade sobre optimização e o refinamento evolutivo sobre a perfeição inicial [12].

2.1.4 Qualidade de Serviços

Por qualidade de serviços ou QoS de um sistema entende-se seus atributos ou requisitos não funcionais. Ou seja, são aspectos qualitativos de estado e não definições de funcionamento, sendo possível citar a performance, a segurança, a disponibilidade, a mantenabilidade, a escalabilidade, etc. Com o amadurecimento e evolução da engenharia de software, tais atributos tornaram-se cruciais para a gerência de riscos e também de custos das aplicações, participando ativamente na definição da arquitetura do sistema e finalmente com peso maior que os próprios requisitos funcionais ao passo que o sistemas aumentam sua ligação a processos de negócio.

Diversos autores se dedicam a definir modelos e métricas para atributos de QoS. Em específico, os sistemas em concordância com o paradigma da orientação a serviços, existe um fator dificultante devido à configuração distribuída dos serviços, que se constituem de tecnologias diversas em sua implementação, o que resulta num ambiente heterogêneo de integração. Com isso, alguns modelos de QoS propostos não se adaptam bem à nova realidade e outros foram definidos focando o ambiente SOC. Alguns trabalhos apresentam uma categorização dos atributos de qualidade para uma melhor contextualização e abordagem dos mesmos. As categorias tem diferentes níveis de granularidade e representam perspectivas diversas, conforme listado em [1]. Esse mesmo artigo propõe um novo modelo que leva em conta a característica segmentada das partes envolvidas e interessadas no provimento e consumo de serviços. Segundo esse modelo, três grandes atores formam as categorias dos atributos de qualidade. São eles os atributos de desenvolvedores, de consumidores e de provedores.

2.1.5 Atributos de QoS

2.1.5.1 Performance

A performance é um atributo relacionado ao tempo. Encontra-se definida no tempo de resposta a uma requisição, na quantidade de requisições por unidade de tempo e na capacidade de atende-las num período determinado [13]. Nesse sentido, o uso de serviços distribuídos acarreta prejuízos à performance inerentes à rede utilizada, além de processamento extra devido ao tratamento e encaminhamento das mensagens entre serviços, especialmente quando trata-se de formatos baseados em texto, como o XML, que possuem tamanho maior que sua representação binária. Também irá prejudicar a performance o processo de descobrimento e alocação de serviços nas situações em que esse processo não é pré-estabelecido.

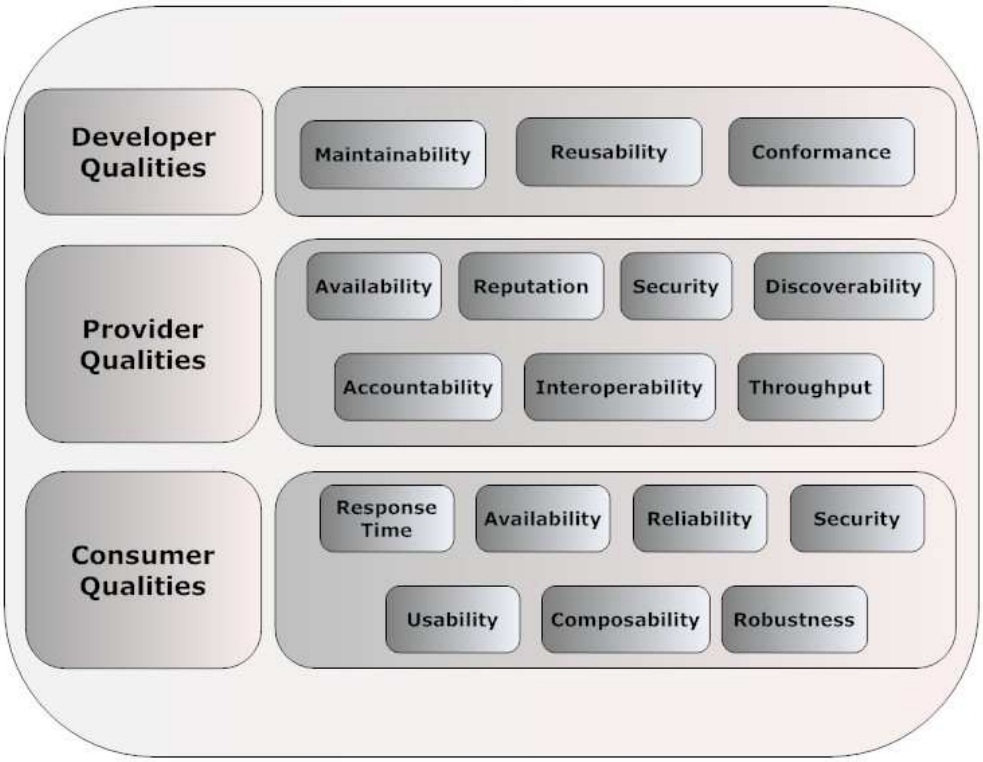

Figura 2.6: Principais atributos de qualidade de serviços na visão de partes interessadas ou *Stakeholders*. Adaptação de [1]

2.1.5.2 Interoperabilidade

A interoperabilidade é definida pela capacidade de componentes de um sistema compartilharem informação e operarem de forma acordada[13]. Assim, mede-se a comunicação e o processamento de informações de maneira independente à fonte dos dados. Visto que o ambiente SOC pressupõe a criação de serviços e, especificamente em SOA, serviços implementados com diversas tecnologias sem prejuízo à intercomunicação, verifica-se que um dos principais benefícios trazidos por essa arquitetura é para a interoperabilidade. Ademais, protocolos já existentes, tal qual o SOAP, realizam a função de troca de informação de maneira interoperável, uma vez que o fazem de forma transparente às aplicações envolvidas. Por fim, o uso de especificações representa um desafio entre avanços para a interoperabilidade, dado que promovem a padronização, mas o uso de especificações incompatíveis ou a sua adesão parcial causará conflitos, comprometendo tal atributo.

2.1.5.3 Segurança

A segurança, como um atributo de qualidade de sistemas de informação, possui alguns princípios.

– A autenticidade trata da capacidade do sistema em identificar a autoria da informação enviada, ou da impossibilidade de que sua autoria seja negada, também conhecido por princípio da não repudiação.

– A integridade irá resguardar a informação tal qual foi produzida, sem alterações, ou garantir que qualquer modificação feita seja detectável.

– A confidencialidade garante que somente atores ou entidades autorizadas tenham acesso às informações, mantendo-as em sigilo para os demais.

É imediata a percepção de que a troca de informações entre as unidades lógicas do sistema pode acarretar sérios riscos à segurança, estando os esforços disponíveis para garanti-la limitados pela necessidade de se manter a interoperabilidade e a performance no ambiente SOC, já que requerem a adoção adicional de especificações, protocolos e consequentemente uma maior redundância de informação e processamento.

2.1.5.4 Disponibilidade ou *Availability*

A disponibilidade define o tempo em que o sistema estará operacional enquanto requisitado, portanto respondendo com sucesso a requisições. Tratando-se de serviços, cada um irá possuir uma disponibilidade que irá influenciar diretamente ou indiretamente a disponibilidade dos sistemas que dependem desses serviços, portanto o ambiente SOC possui o desafio de manter-se operacional diante de suas unidades com particularidades próprias. Como possíveis estratégias para mitigar os problemas causados por serviços inoperantes estão a replicação e o balanceamento de carga, além da troca do provedor do serviço. Nesse sentido, o monitoramento, a gerência e a seleção de serviços são processos importantes em SOC e sua automação um dos maiores desafios.

2.1.5.5 Confiança

A confiança é um atributo relacionado à disponibilidade e trata da capacidade do sistema em operar sem erros. Dessa forma, irá mensurar a frequência de erros ocorridos enquanto o sistema estiver disponível num período de tempo. Mais uma vez, no ambiente distribuído essa capacidade irá depender da confiança dos serviços em composição, mas também do transporte das informações. Portanto, um sistema SOC confiável deverá prezar pela confiança dos serviços e também do mecanismo para troca de mensagens.

2.1.5.6 Manutibilidade

Por manutibilidade entende-se o nível de esforço e custo necessário para realizar correções, modificações e evoluções num sistema. É um atributo diretamente ligado ao desenvolvimento das unidades lógicas e impacta indiretamente na disponibilidade dos serviços existentes e futuros, visto que o tempo de se efetuarem correções ou novas implementações será afetado [14]. É possível ainda encontrar referências a dois subtipos desse atributo, sendo esses a modificabilidade e a testabilidade. O primeiro segue a linha geral que define a manutibilidade, e o segundo define em específico a capacidade de se operar testes nas soluções, seja para avaliação de desempenhos diversos ou pela capacidade em encontrar erros por meio do *debbuging*.

2.1.5.7 Escalabilidade

A escalabilidade diz respeito à capacidade do sistema de suportar escalas crescentes de uso ou modificações sem tornar-se indisponível ou degradar outros aspectos de qualidade. Assim, um sistema será escalável se possibilitar o agregamento de mais usuários, via a escalabilidade de uso, ou de novas funcionalidades via a escalabilidade das funções do sistema.

A orientação a serviços dispõe da composição de serviços com possibilidade de adesão dinâmica e do rearranjamento daqueles já existentes, privilegiando o modelo *stateless* que evita problemas relacionados ao gerenciamento da sessão e à propagação de contexto [13], com possível substituição ou melhoria de serviços por meio do escalonamento vertical e pela replicação de serviços por meio do escalonamento horizontal, fatores que aumentam a vazão e diminuem o tempo de resposta [**?**]. Assim, nota-se que os princípios de design do SOC, sobretudo quando expostos a uma arquitetura orientada a serviços capaz de estruturar processos e recursos irão proporcionar um aumento da escalabilidade.

2.2 SYSTEMATIC MAPPING STUDY

O *Systematic Mapping Study* propõem uma abordagem que preza pela amplitude em detrimento da profundidade de análise. Com isso, é possível incluir um número mais significativo de publicações e um escopo de estudos mais amplo que aquele obtido pelo *Systematic Literature Review*, tendo esse último uma abordagem que difere-se pela maior precisão e riqueza de detalhes em sua classificação, podendo ser aplicado num possível trabalho futuro, cujo enfoque seriam subáreas de interesse identificadas pelo

primeiro processo de mapeamento aqui apresentado.

Um dos maiores revezes dos SRs está no esforço considerável por ele requerido [2]. Dada a vastidão de tópicos abordados pela computação orientada a objetos, é factível supor que um trabalho de SR se tornaria inviável devido ao elevado número de estudos publicados e um número reduzido de pessoal responsável por suas análises.

Um Mapping Study é designado para prover uma ampla visão de determinada área de pesquisa, para verificar se indícios de pesquisa existem em determinados tópicos e prover indicações quantitativas dessas evidências. [15]. O MS segue um processo sistêmico que visa garantir um nível de qualidade adequado aos resultados obtidos. Algumas passos são diretamente herdados dos SRs, enquanto outros são adaptados para se garantir a viabilidade de seus objetivos. Cada passo do processo de mapeamento irá gerar um resultado, sendo o resultado final o mapeamento propriamente dito [2].

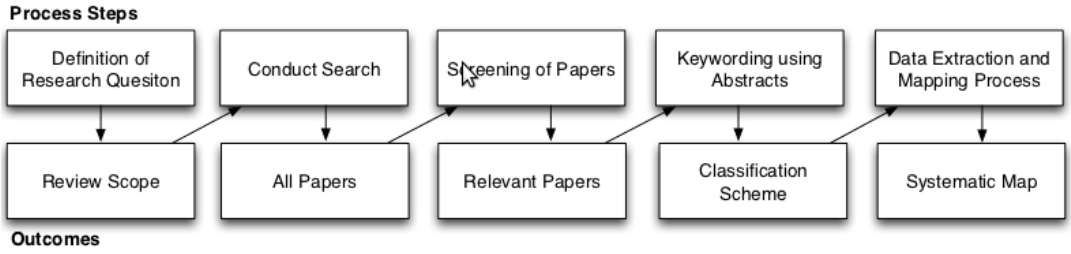

Figura 2.7: Processo para o Systematic Mapping Study. Adaptação de [2]

2.2.1 Escolha das Perguntas de Pesquisa

As *research questions*, ou RQs, fundamentam a motivação para um mapeamento de estudos. São elas que irão guiar todo o processo consequente. Em sua elaboração é preciso ter em mente as características de abertura do escopo e de profundidade de um MS, de modo que as perguntas devem ter como objetivo respostas passíveis de serem encontradas por meio do tipo de análise efetuada num MS. Assim, questões relacionadas à frequência, à identificação dos centros de pesquisa, à identificação dos tópicos existentes ou que vem sendo pesquisados ou ao tipo de pesquisa são mais comuns. Ou seja, o foco das questões é menos estreito e envolve um número maior de publicações.

2.2.2 Pesquisa primária

À partir das RQs é possível elaborar termos que serão aplicados em sistemas de busca de publicações. Para tanto, a frase de busca deverá estar estruturada visando englobar todos os tópicos de interesse numa área, restringindo-os se necessária a determinados aspectos. O uso de operadores lógicos é indispensável para que se obtenha resultados mais precisos e que não deixem escapar publicações devido à diferença usada no emprego de palavras chave, havendo pequenas variações nos padrões de uso para cada sistema de busca. Também é interessante realizar testes que observem de maneira preliminar os resultados obtidos nas buscas para se preciso realizar modificações na *string* usada.

Uma abordagem metódica para a construção da *string* de busca consiste na definição de termos categorizados em população, intervenção, comparação e resultados, visto que a medicina é uma das áreas onde as Systematic Reviews são mais usadas. No entanto esse modelo pode também servir para definir os termos de busca em qualquer outra área por meio da abstração desses conceitos.

A população irá definir o objeto de interesse, ou seja, a área em que se deseja efetuar o mapeamento. Geralmente é necessária a inclusão de um ou mais termos que a representem na *string* de busca. A intervenção representa metodologias, procedimentos, arquiteturas, modelos, ferramentas, etc, que atuam numa determinada população. Em geral são criados termos de busca que associam os termos de população aos termos de intervenção por meio de operadores lógicos *E*, sendo sua variação ou composição com outras intervenções feitas com o operador lógico *OU*. A comparação define intervenções com as quais as primeiras serão comparadas ou pela comparação com a ausência de uma intervenção, ou seja, pelo não uso ou aplicação de uma intervenção ao seu uso ou aplicação. Por fim, os resultados são estadosdistribuem as submissões de interesse para a população após serem submetidos a uma ou mais intervenções. Por exemplo, pode-se mesurar se certos níveis qualitativos ou quantitativos foram atingidos para determinados aspectos [15].

2.2.3 Inclusão e Exclusão de Publicações

Como resultado da etapa anterior, espera-se obter um número elevado de publicações, estando uma parcela das mesmas fora do escopo do mapeamento. Deve-se então realizar a análise rápida dos resumos ou *abstracts* e dos demais metadados para embasar o processo de exclusão das publicações de acordo com critérios de inclusão e exclusão do MS.

A definição dos critérios de inclusão e exclusão deve prezar pela praticidade de sua análise. É possível

definir restrições quanto ao tipo de publicação, origem, quantidade mínima de páginas e principalmente verificando o comprometimento da pesquisa com o foco e escopo do mapeamento. Essa correspondência semântica pode ser checada entre as palavras chaves da publicação e no próprio resumo. Outra observação importante é para o fórum em que a publicação está disponível, uma vez que os sistemas de busca realizam distribuições por áreas e tópicos de pesquisa.

2.2.4 Definição de Categorias e Extração de Dados

Tendo definido as questões de pesquisa, é necessário identificar um esquema que será usado na classificação das publicações. Existem inúmeras possibilidades para a definição de facetas, ou *facets*. Dado o interesse comum em mapeamentos de estudo em obter dados sobre o estado das pesquisas, uma categorização que identifique o tipo ou tipos de pesquisa realizada em cada artigo, tal qual sugerido em [16]. Outra categorização possível e complementar está no tipo de contribuição dada, ou seja, identifica o foco da pesquisa em termos de modelo, processo, método, ferramenta ou métricas.

Para a definição de facetas relacionadas ao contexto das pesquisas, é possível que alguns tipos sejam determinados baseando-se nas RQs elaboradas para o mapeamento. Em [2] é proposto o procedimento de busca de palavras chaves nos resumos das publicações. Isso possibilita o agrupamento dessas palavras em categorias que podem ser inclusas entre aquelas relacionadas ao contexto. Considerando um número elevado de publicações, a busca, agrupamento e definição de categorias contextuais deve ser feito de modo iterativo, inclusive para reavaliar a inclusão dos termos já escolhidos de acordo com a evolução do cenário, conforme está ilustrado em 2.8. Esse procedimento cíclico também se aplica às demais categorias definidas para o mapeamento.

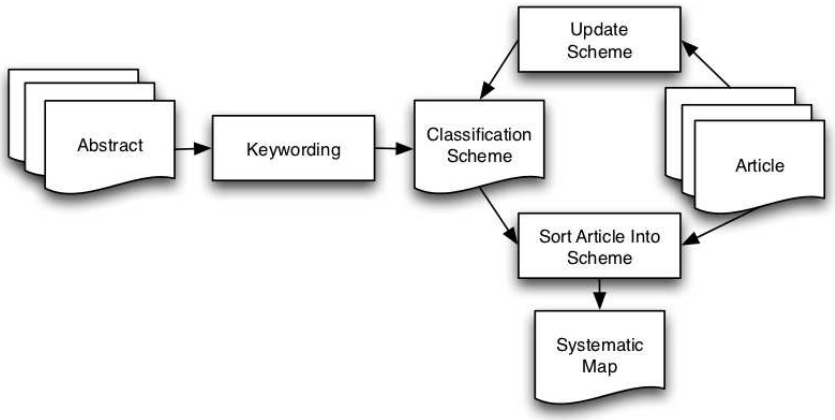

Figura 2.8: Busca iterativa de palavras chaves para categorização. Adaptação de [2]

Tal abordagem pode ser vista como dinâmica, uma vez que em paralelo à obtenção de palavras chaves, o próprio conteúdo da pesquisa estará sendo analisado por meio de seu resumo e demais metadados. Assim, a extração dos dados que posicionam cada publicação em determinadas categorias ocorre junto ao aperfeiçoamento na definição das mesmas.

Durante a abordagem de cada publicação, é importante tomar notas explicativas justificando a classificação realizada em cada categoria. Isso auxilia tanto o responsável pela classificação quanto aqueles que posteriormente também passarem pela classificação da mesma publicação ou que servirem de revisores, reduzindo possíveis desvios. Para tanto, é indispensável o uso de ferramentas que organizem as informações colhidas e anotadas, de preferência com acesso gráfico e de boa usabilidade.

2.2.5 Mapeamento e Apresentação de Dados

Após obter os dados, o mapeamento prossegue com a apresentação dos mesmos por meio de gráficos. É preciso escolher um tipo de representação clara e que possa ilustrar os dados nas dimensões necessárias. Caso seja objetivo cruzar informações em três dimensões, o uso de *bubble plots* ou gráficos em bolhas permite diferenciar quantidades entre cada cruzamento das categorias definidas nos eixos x e y, ou seja, cada intersecção poderá ser dimensionada pelo tamanho da bolha que a representa. Dessa forma, esse tipo de gráfico serve bem ao propósito quantitativo do mapeamento de estudos que em geral relaciona frequência com duas ou mais categorias de classificação.

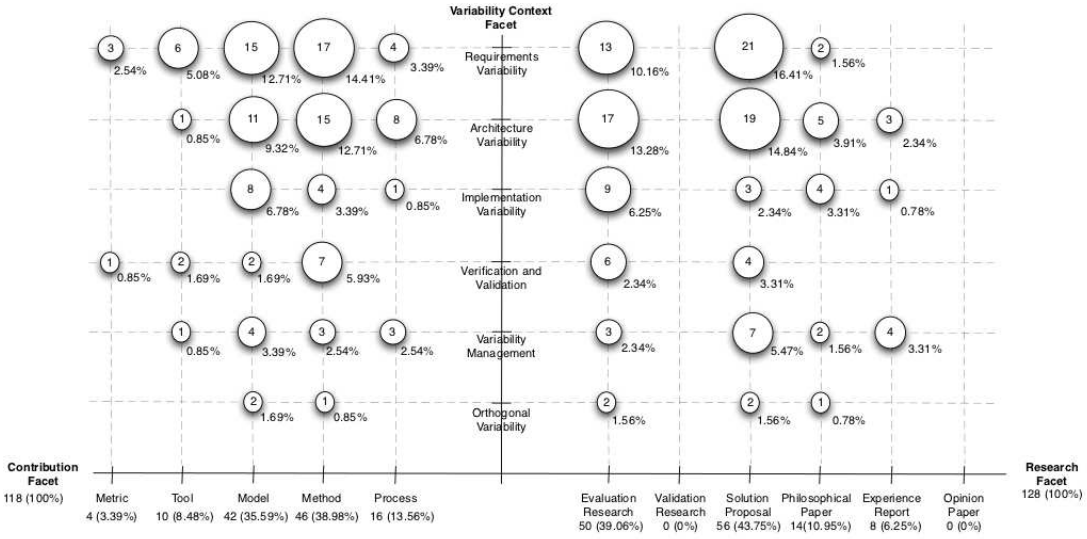

Figura 2.9: Exemplo de gráfico de *bubble plot*. Adaptação de [2]

3 ABORDAGEM

REFERÊNCIAS

[1] BALFAGIH, Z.; HASSAN, M. Quality model for web services from multi-stakeholders' perspective. In: *Information Management and Engineering, 2009. ICIME '09. International Conference on*. [S.l.: s.n.], 2009. p. 287 –291.

[2] PETERSEN, K. et al. Systematic mapping studies in software engineering. *12th International Conference on Evaluation and Assessment in Software Engineering*, v. 17, n. 1, p. 1–10, 2007.

[3] PAPAZOGLOU, M. P.; HEUVEL, W.-J. Service oriented architectures: approaches, technologies and research issues. *The VLDB Journal*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, v. 16, p. 389–415, July 2007. ISSN 1066-8888. Disponível em: <http://dx.doi.org/10.1007/s00778-007-0044-3>.

[4] ERL, T. *SOA Principles of Service Design (The Prentice Hall Service-Oriented Computing Series from Thomas Erl)*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2007. ISBN 0132344823.

[5] BREIVOLD, H. P.; LARSSON, M. Component-based and service-oriented software engineering: Key concepts and principles. In: *Proceedings of the 33rd EUROMICRO Conference on Software Engineering and Advanced Applications*. Washington, DC, USA: IEEE Computer Society, 2007. p. 13–20. ISBN 0-7695-2977-1. Disponível em: <http://dl.acm.org/citation.cfm?id=1302497.1302990>.

[6] CRNKOVIC, I.; CHAUDRON, M.; LARSSON, S. Component-based development process and component lifecycle. In: *Proceedings of the International Conference on Software Engineering Advances*. Washington, DC, USA: IEEE Computer Society, 2006. p. 44–. ISBN 0-7695-2703-5. Disponível em: <http://dl.acm.org/citation.cfm?id=1193212.1193814>.

[7] PATIL, S. *Integration Approaches: Web Services vs Distributed Component Models PART II*. 2003. [Online; accessed 20-February-2012]. Disponível em: <http://soa.sys-con.com/node/39754>.

[8] PAUTASSO, C.; ZIMMERMANN, O.; LEYMANN, F. Restful web services vs. "big"' web services: making the right architectural decision. In: *Proceedings of the 17th international conference on World Wide Web*. New York, NY, USA: ACM, 2008. (WWW '08), p. 805–814. ISBN 978-1-60558-085-2. Disponível em: <http://doi.acm.org/10.1145/1367497.1367606>.

[9] JESTON, J.; NELIS, J. *Business Process Management: Practical Guidelines to Successful Implementations*. Butterworth-Heinemann, 2008. ISBN 9780750686563. Disponível em: <http://books.google.com.br/books?id=QI9aaKRIPIsC>.

[10] POSSEL, J. *Accenture's History: The beginnings in the early 1950?s*. January 2011. [Online; accessed 29-February-2012]. Disponível em: <http://www.accenture-blogpodium.nl/about-accenture/accentures-history-the-beginnings-in-the-early-1950s/>.

[11] WHITE, S. A.; CORP, I. B. M. Process modeling notations and workflow patterns. *Business*, Future Strategies Inc., v. 21, n. 1999, p. 1–25, 1999.

[12] VARIOS. *Soa Manifesto*. 2012. [Online; accessed 12-December-2011]. Disponível em: <http://www.soa-manifesto.org/>.

[13] O'BRIEN, L.; MERSON, P.; BASS, L. Quality attributes for service-oriented architectures. In: *Proceedings of the International Workshop on Systems Development in SOA Environments*. Washington, DC, USA: IEEE Computer Society, 2007. (SDSOA '07), p. 3–. ISBN 0-7695-2960-7. Disponível em: <http://dx.doi.org/10.1109/SDSOA.2007.10>.

[14] CLEMENTS, P.; KAZMAN, R.; KLEIN, M. *Evaluating Software Architectures: Methods and Case Studies*. [S.l.]: Addison-Wesley, 2001. ISBN 978-0-201-70482-2.

[15] KITCHENHAM, B.; CHARTERS, S. Guidelines for performing systematic literature reviews in software engineering. *Version*, v. 2, n. EBSE 2007-001, p. 2007?01, 2007.

[16] WIERINGA, R. et al. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements Engineering*, Springer London, v. 11, p. 102–107, 2006. ISSN 0947-3602. 10.1007/s00766-005-0021-6.

ANEXOS