

WEB SERVICES: componentes reutilizáveis no processo de desenvolvimento orientado ao reuso***WEB SERVICES: reusable components in the reuse-oriented development process***

Yuri Leal Gomes – muod.equipe@gmail.com
Faculdade de Tecnologia de Taquaritinga (Fatec) – Taquaritinga – SP – Brasil

Daniela Gibertoni – daniela.gibertoni@fatectq.edu.br
Faculdade de Tecnologia de Taquaritinga (Fatec) – Taquaritinga – SP – Brasil

DOI: 10.31510/infa.v19i2.1531

Data de submissão: 01/09/2022

Data do aceite: 28/11/2022

Data da publicação: 20/12/2022

RESUMO

A reutilização de *softwares* ou componentes preexistentes é uma abordagem que pode ser utilizada no processo de desenvolvimento com o intuito de aumentar a confiança, reduzir os riscos e prazos. Nesse contexto, *web services* são considerados componentes com uma alta possibilidade de reusabilidade, todavia, seu processo de implementação deve ser orientado à reusabilidade, possibilitando assim, sua inclusão em um processo de desenvolvimento orientado ao reuso. Sendo assim, o presente trabalho visa investigar através da revisão bibliográfica, a relação entre o desenvolvimento de *web services* reusáveis com o processo de desenvolvimento orientado ao reuso. Concluiu-se que o processo de desenvolvimento de *web services* pode ser conduzido através da engenharia de serviços. Ao estabelecer atividades responsáveis pela projeção de serviços reusáveis, haverá a disponibilização de mais componentes reutilizáveis. Portanto, ao oferecer uma quantidade grande de componentes reutilizáveis, há um estímulo para utilizar-se de abordagens de desenvolvimento orientadas ao reuso.

Palavras-chave: *Software*. Reúso. Engenharia de *software*. Componentes reusáveis.

ABSTRACT

The reuse of pre-existing software or components is an approach that can be used in the development process in order to increase trust, reduce risks and deadlines. In this context, web services are considered components with a high possibility of reusability, however, their implementation process must be reusability-oriented, thus enabling their inclusion in a reuse-oriented development process. Therefore, the present work aims to investigate through the literature review, the relationship between the development of reusable web services with the reuse-oriented development process. It was concluded that the web services development process can be conducted through service engineering. By establishing activities responsible for designing reusable services, more reusable components will be made available. Therefore,

by offering a large number of reusable components, there is an incentive to use reuse-oriented development approaches.

Keywords: Software. Reuse. Software Engineering. Reusable components.

1 INTRODUÇÃO

Atualmente as empresas estão inseridas em um cenário global, sendo este afetado constantemente por mudanças rápidas. Assim sendo, as organizações necessitam atender novas oportunidades e mercados, mudanças nas condições econômicas e ainda, ao surgimento de novos produtos e serviços concorrentes (SOMMERVILLE, 2011; PRESSMAN, 2011).

Produtos de *software* estão inseridos na maior parte dos processos de negócio nas organizações e, portanto, o seu desenvolvimento e suas entregas rápidas se tornaram fatores críticos para que se consiga garantir resposta às novas oportunidades e pressões competitivas que atingem as organizações.

Frente à demanda constante de desenvolvimento de produtos de *software* cada vez mais complexos em menor tempo e com qualidade, a comunidade de Engenharia de *Software* (ES) vem buscando desenvolver novas abordagens que atendam a este cenário (PRESSMAN, 2011; BORGES E FALBO, 2001).

Uma abordagem proposta em 2001 a essas exigências foi o desenvolvimento baseado em reuso. Entretanto, segundo McILROY (1968 apud SOMMERVILLE, 2011), essa estratégia de desenvolvimento já havia sido proposta há mais de 40 (quarenta) anos.

Segundo Krueger (1992), a reutilização de *software* pode ser definida como sendo o processo de criação de *software* a partir de *software* preexistente. Nesse sentido, Sommerville (2011) especifica três tipos de componentes de *software* que podem ser utilizados em um processo orientado ao reuso, sendo eles: *web services*, coleções de objetos e sistemas de *software stand alone*.

Sendo assim, este artigo possui como objetivo realizar uma revisão bibliográfica que visa investigar a relação do desenvolvimento de *web services* como componentes reusáveis no processo de desenvolvimento orientado ao reuso.

O presente trabalho está organizado em 5 (cinco) seções: a seção 2 (dois) apresenta a fundamentação teórica deste trabalho, a seção 3 (três) os procedimentos metodológicos utilizados em sua elaboração, tendo como sequência na seção 4 (quatro) os resultados alcançados. Por fim, a seção 5 (cinco) apresenta as conclusões referentes a este trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Processos e modelos de *software*

Segundo Fuggetta (2000), um processo de *software* pode ser definido como sendo, um conjunto de políticas, estruturas organizacionais, tecnologias, procedimentos e artefatos que são necessários para projetar, implementar, implantar e manter um produto de *software*.

Há vários processos de *software* diferentes, entretanto, Sommerville (2011), afirma que todos devem possuir 4 (quatro) atividades que são de fundamental importância para a engenharia de *software*: especificação: etapa em que é realizado o processo de levantamento de requisitos; projeto e implementação: produção do *software*; validação: etapa em que se verifica se houve o atendimento correto às demandas do cliente; e evolução: o *software* necessita evoluir para que assim, atenda às necessidades de mudança dos clientes.

Vários autores afirmam que há relação direta entre a qualidade do processo de desenvolvimento com a qualidade do produto (FUGGETTA, 2000; PRESSMAN, 2011; WOO et. al 2006). Bertollo e Falbo (2003) afirmam que o principal estopim de problemas em um ciclo de desenvolvimento de produtos de *software* é a ausência de processos claramente definidos e efetivos. Além disso, ao utilizar-se de uma metodologia de forma consistente, há a possibilidade de se economizar tempo e dinheiro durante o projeto (WOO et. al, 2006).

Neste aspecto, um modelo de processo de *software*, é definido como sendo, uma representação simplificada do processo de *software* (SOMMERVILLE, 2011; PRESSMAN, 2016). Para tanto, foram analisados os principais modelos e metodologias existentes, sendo elas: Cascata (SOMMERVILLE, 2005); Incremental (PRESSMAN, 2006); Prototipagem (PRESSMAN, 2016); *Extreme Programming* (BECK, 2004; LARMAN, 2003); Processo unificado (KRUCHTEN, 2003); *Scrum* (SCHWABER, 2002) e o Processo orientado ao reúso (SZYPERSKI, 1999; SOMMERVILLE, 2011; PRESSMAN, 2016).

O enfoque do presente trabalho será o processo de *software* orientado ao reúso, sendo este, assunto do próximo item.

2.2 Engenharia de *software* orientada ao reúso

A engenharia de *software* baseada em reúso pode ser definida como sendo, uma estratégia em que o processo de desenvolvimento é orientado ao reúso de *software* já existente (SOMMERVILLE, 2011). De modo geral, é a utilização de conhecimentos e artefatos de engenharia existentes com o intuito de realizar uma construção de novos sistemas de *software* (FRAKES E FOX, 1996).

Um modelo deste processo de desenvolvimento é apresentado na Figura 1.

Figura 1: Modelo de processo de desenvolvimento de *software* orientado ao reúso.



Fonte: Adaptado de Sommerville (2011).

Conforme apresentado na Figura 1, pode-se constatar que o estágio de especificação de requisitos iniciais e o estágio da validação do sistema são comparáveis a outros processos. Entretanto, os demais não são, sendo eles: análise de componentes: de acordo com os requisitos levantados, realiza-se uma busca nos componentes pré-existentis; alterações nos requisitos: os requisitos são analisados com base nas informações sobre os componentes identificados.

Posteriormente, os requisitos serão alterados (se possível) para refletir os componentes disponíveis; projeto de sistema com reúso: neste estágio, inicia-se o processo de construção do *framework* do sistema ou algum componente é reusado; e desenvolvimento e integração: durante esta etapa, os *softwares* que não foram obtidos externamente são implementados e os componentes e sistemas COTS são integrados com o intuito de criar o novo sistema (SOMMERVILLE, 2011).

O processo de desenvolvimento demonstrado através deste modelo, integra-se a tese da perspectiva técnica da reutilização apresentada por Biggerstaff e Richter (1989 apud KRUEGER, 1992). Nela, os processos de desenvolvimento orientados ao reúso compartilham 4 (quatro) conceitos básicos, sendo eles: abstração: compreender se um determinado artefato

de *software* pode ser reutilizado e se atenderá a expectativa do utilizador; seleção: realizar a busca dos componentes em bibliotecas de *software*, de modo a localizar o que atenderá o reutilizador; adaptação: realização de alterações no componente, de modo que este artefato possa ser reutilizado em um contexto diferente daquele em que fora projetado inicialmente; e integração: etapa em que se realiza a união final contendo os artefatos reutilizados (BIGGERSTAFF; RICHTER, 1989 apud KRUEGER, 1992).

Neste sentido, segundo Sommerville (2011), existem 3 (três) tipos de componentes de *software* que podem ser utilizados em um processo orientado ao reúso, são eles: *web services*: são desenvolvidos em conformidade com os padrões de serviço (HANSEN et. al., 2002); coleções de objetos: desenvolvidos como um pacote que será integrado com um *framework* de componentes, como *.NET* ou *J2EE*; e sistemas de *software stand alone*: que são configurados para utilização em um ambiente particular (SOMMERVILLE, 2011).

Como o enfoque deste trabalho está em torno de *web services*, no item seguinte estará sendo discorrido acerca deste tema.

2.3 Web services

Na visão de Hansen e Pinto (2003), *web services* podem ser definidos como componentes de *software* que são independentes da implementação e da plataforma. São descritos através de uma linguagem de descrição de serviços, publicados em um registro e descobertos através de um mecanismo padrão. Podem ser acionados também a partir de uma *Application Program Interface* (API) através da rede e serem compostos juntamente com outros serviços (HANSEN e PINTO, 2003; HANSEN et. al., 2002; KREGER, 2001).

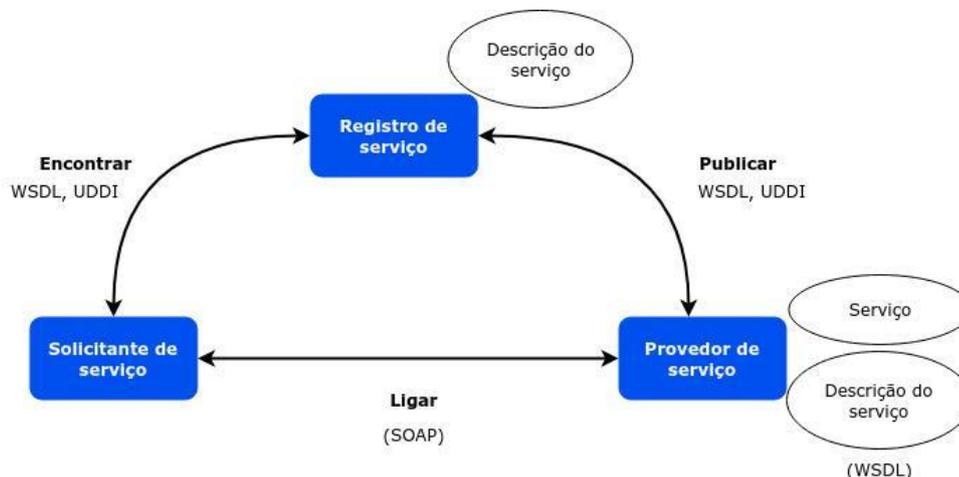
De forma a apresentar e descrever a estrutura de *web services*, no item subsequente será apresentada, de modo simplificado, a sua arquitetura.

2.4 Arquitetura de web services

De modo geral, a Figura 2 sintetiza a ideia de uma arquitetura orientada a serviços (SOA, *Service - Oriented Architecture*). Provedores projetam e implementam seus serviços, especificando posteriormente a sua interface. Publicam também informações sobre esses serviços em um registro acessível. Os solicitantes ou clientes de serviço que desejam utilizar um serviço descobrem a sua especificação e localizam o provedor de serviço. Após isso, ele

pode ligar sua aplicação ao serviço específico e com isso, comunicar-se com ele através dos protocolos de serviço - padrão (SOMMERVILLE, 2011; HANSEN e PINTO, 2003).

Figura 2: Estrutura de *web services* - papéis, operações e artefatos.



Fonte: Adaptado de Sommerville (2011) e Kreger (2001).

Portanto, a base de um serviço é o seu fornecimento independente da aplicação que o usa (TURNER et. al., 2003 apud SOMMERVILLE, 2011).

Para garantir o desenvolvimento de serviços orientados ao reuso, a engenharia de serviços apresenta um conjunto de atividades específicas a esta finalidade. Um breve relato sobre esse processo de desenvolvimento está sendo apresentado no item a seguir.

2.5 Engenharia de serviços

Engenharia de serviços pode ser definida como um processo que orienta o desenvolvimento à criação de serviços reutilizáveis. Nesse sentido, engenheiros de serviço necessitam garantir que o serviço a ser desenvolvido represente uma abstração reutilizável e útil em diferentes sistemas (SOMMERVILLE, 2011).

Um modelo desse processo de desenvolvimento pode ser encontrado na Figura 3.

Figura 3: Modelo de processo de engenharia de serviços.

Fonte: Adaptado de Sommerville (2011).

Sommerville (2011), cita que há 3 (três) estágios lógicos presentes no processo de engenharia de serviços, conforme apresentado na Figura 3. São eles:

1. Identificação de serviço candidato: etapa de definição dos possíveis serviços que podem ser implementados e definição dos requisitos de serviço;
2. Projeto de serviço: estágio em que se projeta a lógica e as interfaces de serviço WSDL;
3. Implementação e implantação de serviço: fase em que ocorre a implementação e teste dos serviços, tornando-os disponíveis para uso (SOMMERVILLE, 2011).

Conclui-se que essa disponibilização de serviços reutilizáveis pode garantir, assim, um aproveitamento de processos de desenvolvimento baseados em reúso, visto que há uma maior disponibilização de componentes reusáveis.

Na seção subsequente, está sendo apresentada a metodologia de pesquisa utilizada durante a elaboração deste trabalho.

3 PROCEDIMENTOS METODOLÓGICOS

Na elaboração deste artigo foi utilizada a metodologia de pesquisa bibliográfica, que é definida segundo Gil (2008), como sendo uma pesquisa de caráter exploratório em que os dados são obtidos através de buscas. Os meios utilizados na obtenção destes dados podem ser livros ou artigos científicos (GIL, 2008).

Na próxima seção, estão sendo apresentados os resultados e conclusões obtidos durante a elaboração do presente trabalho.

4 RESULTADOS E DISCUSSÃO

Abordagens orientadas ao reúso de *software* necessitam de uma ampla base de componentes reusáveis (SOMMERVILLE, 2011). Assim sendo, desenvolver componentes que possam ser reutilizados em diferentes projetos se torna um fator crítico para garantir uma maior abrangência na utilização de processos orientados ao reúso de *software*.

Segundo Councill e Heineman (2001 apud SOMMERVILLE, 2011), um componente pode ser definido como sendo um elemento de *software* que está em conformidade com um modelo de componente padrão e que pode ser implantado e composto de forma independente.

Já Szyperski (2002 apud SOMMERVILLE, 2011), define um componente como sendo uma unidade de composição que possui uma interface contratualmente especificada e apenas dependências de contexto explícitas. Além disso, um componente de *software* pode ser implantado de forma independente e está sujeito a ser composto por partes de terceiros.

Sommerville (2011), afirma que na comunidade de CBSE não há consenso em relação à definição de componentes, porém, ambas as definições unidas são segundo sua visão a melhor forma de definição de componentes.

De maneira a complementar o entendimento acerca de componentes, a CBSE (*Component Based Software Engineering*) pode ser definida como sendo uma abordagem de desenvolvimento de *software* baseada na reutilização de componentes. Essa abordagem apresenta dois tipos de processos: o “desenvolvimento para reúso” que enfatiza o desenvolvimento de componentes ou serviços que serão reutilizados em outros projetos, e o “desenvolvimento com reúso” que possui como enfoque o desenvolvimento utilizando-se componentes e serviços já existentes (SOMMERVILLE, 2011).

Serviços que são componentes com possibilidade de reutilização são considerados de baixo acoplamento e reusáveis; ou seja, serviços sempre devem operar da mesma forma, sendo independentes de seu ambiente de execução (SOMMERVILLE, 2011).

Entretanto, há a necessidade de se garantir que o processo de desenvolvimento utilizado para a criação de serviços seja voltado para a reutilização em diferentes sistemas.

Para isso, utiliza-se da engenharia de serviços como processo de desenvolvimento, garantindo assim que o serviço a ser desenvolvido represente uma abstração reusável que poderá ser útil em diferentes sistemas (SOMMERVILLE, 2011).

A forma em que o reúso está presente na engenharia de serviços está sendo apresentada no item a seguir.

4.1 Visão de reuso na etapa de identificação de serviço candidato

Cada organização possui uma diversidade de processos, nesse sentido, existem muitas possibilidades de implementação de serviços para apoio.

A identificação de serviço candidato envolve, segundo Sommerville (2011), compreensão e análise dos processos de negócios da organização com o intuito de decidir quais serviços reusáveis seriam úteis ao suporte destes processos se fossem desenvolvidos.

Erl (2005 apud SOMMERVILLE, 2011) especifica 3 (três) tipos de serviços que podem ser identificados nesta etapa, sendo eles:

1. Serviços utilitários: os que implementam alguma funcionalidade geral, que poderá ser utilizada por diferentes processos de negócios (SOMMERVILLE, 2011);
2. Serviços de negócio: serviços que estão ligados a uma função específica dos negócios (SOMMERVILLE, 2011); e
3. Serviços de coordenação ou de processo: serviços que suportam amplos processos de negócios, envolvendo geralmente atividades e atores diferentes (SOMMERVILLE, 2011).

Erl (2005 apud SOMMERVILLE, 2011) apresenta ainda, a possibilidade de classificação de serviços como orientados a tarefas: são aqueles associados com alguma atividade; e os orientados a entidades: são como objetos.

O objetivo nesta etapa deve ser o de identificar serviços que são logicamente coerentes, independentes e reusáveis (SOMMERVILLE, 2011).

Segundo Sommerville (2011), identificar candidatos a serviços pode ser uma tarefa difícil, pois se faz necessário considerar a forma de uso destes serviços. Neste cenário, é preciso pensar nos possíveis candidatos e, com isso, analisar uma série de questões referentes a eles para verificar se tendem a ser serviços úteis.

As possíveis questões apresentadas por Sommerville (2011) para auxiliar no processo de identificação dos serviços potencialmente reusáveis estão adaptadas e apresentadas no Quadro 1.

Quadro 01: Questões de identificação de serviços potencialmente reutilizáveis.

Para serviços orientados a entidades, esse serviço é relacionado apenas com uma entidade lógica utilizada em vários processos de negócios? Quais são
--

as operações executadas normalmente sobre essa entidade que deverá ser apoiada?
Para um serviço orientado a tarefas, é uma tarefa realizada por pessoas diferentes na organização? Essas pessoas estarão dispostas a aceitar a padronização inevitável que ocorre quando um único suporte é prestado?
O serviço é independente?
Para o funcionamento, o serviço precisa manter o estado? Se forem necessárias as informações de estado, um banco de dados deverá ser usado, o que pode limitar a reusabilidade do serviço.
O serviço poderia ser utilizado por clientes fora da organização?
Os diferentes usuários de serviço podem possuir diferentes requisitos não funcionais?

Fonte: Adaptado de Sommerville (2011).

As respostas obtidas durante essa etapa e por meio das questões apresentadas auxiliarão no processo de seleção e refinamento de abstrações que poderão ser implementadas como serviços (SOMMERVILLE, 2011).

A identificação de serviços reusáveis gera nesse processo uma disponibilização de componentes que fomentam a utilização de processos orientados a reuso.

A seção seguinte traz as conclusões obtidas através desse trabalho, juntamente com as considerações finais.

5 CONCLUSÃO

O desenvolvimento de *software* baseado em reuso é uma abordagem presente na Engenharia de *Software* (ES) que visa a implementação de *software* utilizando-se de componentes ou *softwares* já existentes. Os seus benefícios incluem maior confiança, redução de riscos, conformidade com padrões e o tempo de desenvolvimento reduzido.

Porém, essa abordagem necessita de uma quantidade considerável de componentes reusáveis para etapas de análises, sendo este um fator crítico para seu êxito.

Web services são componentes com uma grande possibilidade de reusabilidade, em que seu processo de especificação pode incluir atividades de modo a projetar serviços reusáveis para demais projetos.

A forma de desenvolvimento orientada ao reúso garante assim, uma maior disponibilização de componentes, o que fomenta a utilização de abordagens de desenvolvimento de produtos de *software* orientadas ao reúso.

REFERÊNCIAS

Beck, K. and Andres. C. (2004), **Extreme Programing explained: Embrace change**, 2.ed. Addison Wesley Professional.

Bertollo, G. e Falbo, R. A. (2003) “**Apoio Automatizado à Definição de Processos de Software em Níveis**”. In: II Simpósio Brasileiro de Qualidade de Software (SBQS), Fortaleza, Brasil.

BORGES, L. M. S. e FALBO, R. A. (2001) “**Gerência de Conhecimento sobre Processos de Software**”. In: XV Simpósio Brasileiro de Engenharia de Software (SBES), Rio de Janeiro, Brasil.

Frakes, W. and Fox, C. (1996). **Quality improvement using a software reuse failure modes model**. *IEEE Transactions on Software Engineering*, 22(4):274–279.

Fuggetta, A. (2000) “**Software Process: A Roadmap**”. In: 22nd International Conference on Software Engineering (ICSE), Proceedings of the Conference on The Future of Software Engineering, pp. 25-34., New York: ACM Press.

Gil, Antonio Carlos. **Método e técnicas de pesquisa social**. 6. ed. São Paulo: Atlas S.A, 2008.

HANSEN, R. P.; PINTO, S. C. S. C.. **Construindo Ambientes de Educação Baseada na Web Através de Web Services Educacionais**. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE), [S.l.], p. 61-70, nov. 2003. ISSN 2316-6533. Disponível em: <<https://www.br-ie.org/pub/index.php/sbie/article/view/236>>. Acesso em: 08 abr. 2022. doi:<http://dx.doi.org/10.5753/cbie.sbie.2003.61-70>.

Hansen, R.P. Santos, C.T. Pinto, S.C.C.S.; Lanius, G.L and Massen, F. **Web Services: An Architectural Overview**. First International Seminar on Advanced Research In E-Business-EBR 2002. PUC-RIO. November 2002.

Kreger, H. **Web Services Conceptual Architecture**. IBM Software Group, May 2001

Kruchten, P. (2003), **The Rational Unified Process: An Introduction**. 3th edition. New York: Addison – Wesley.

Krueger, C. W.(1992) “**Software Reuse**”, ACM Computing Surveys, 24(2), p. 131-184.

Larman, C. (2003), **Agile and iterative development: a manager's guide**. 1. ed. New York: Addison - Wesley.

Osterweil, L. (1987) “**Software Process Are Software Too**”. In: 9th International Conference on Software Engineering (ICSE), Monterey, Estados Unidos, p. 2-13.

Pressman, Roger S. **Engenharia de software:uma abordagem profissional**. 6.ed. –Porto Alegre: AMGH, 2006.

Pressman, Roger S. **Engenharia de software:uma abordagem profissional**. 7.ed. –Porto Alegre: AMGH, 2011.

Pressman, Roger S. **Engenharia de software:uma abordagem profissional**. 8.ed. –Porto Alegre: AMGH, 2016.

Schwaber, K. and Beedle, M. (2002) **Agile software development with Scrum**. Prentice Hall.

Sommerville, Ian. **Engenharia de Software**. 6. ed. São Paulo: Pearson Education do Brasil, 2005.

Sommerville, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson Education do Brasil, 2011.

Szyperski, C. (1999), “**Component software: beyond object-oriented programming**”. 1 edition. New York: Addison – Wesley, 411p.

Woo, F., Mikusauskas, R., Bartlett, D. and Law, R. (2006), “**A Framework for the Effective Adoption of Software Development Methodologies**”. In: ACM SE’06, March, Melbourne, Florida, USA.