

A percepção de desenvolvedores frente a utilização de uma ferramenta de apoio ao reuso de software: um estudo de caso em uma pequena empresa de desenvolvimento de software

Analysis of the behavior of the developers in the implementation and use of a tool to support the reuse of software

Análisis del comportamiento de los desarrolladores en la implementación y uso de una herramienta de apoyo a la reutilización de software

Recebido: 08/09/2021 | Revisado: 14/09/2021 | Aceito: 22/09/2021 | Publicado: 24/09/2021

Marcos Vinicius Rossetto

ORCID: <https://orcid.org/0000-0002-6310-5913>

Universidade de Caxias do Sul, Brasil

E-mail: mvossetto@ucs.br

Daniel Luis Notari

ORCID: <https://orcid.org/0000-0001-6135-3043>

Universidade de Caxias do Sul, Brasil

E-mail: dlnotari@ucs.br

Scheila de Avila e Silva

ORCID: <https://orcid.org/0000-0002-3472-3907>

Universidade de Caxias do Sul, Brasil

E-mail: sasilva6@ucs.br

Resumo

Um dos desafios para desenvolvedores e empresas de desenvolvimento de software é a implementação de códigos-fonte de forma padronizada. Para auxiliar neste processo, existem alguns padrões de projetos que fornecem direções como, por exemplo, o reuso de software. Nesse sentido, o presente estudo buscou avaliar o comportamento de desenvolvedores durante a implantação de um ferramenta de apoio ao reuso de software. A avaliação foi realizada por meio da aplicação de questionários antes e depois da implantação da ferramenta de apoio ao reuso de software. Após a realização das análises dos questionários, pode-se perceber que os desenvolvedores compreendem o conceito de reuso de software. Entretanto, realizam a utilização da ferramenta de forma inapropriada. Com isso, percebe-se a necessidade de um empenho de toda a equipe de desenvolvedores juntamente com os gerentes de projeto.

Palavras-chave: Reuso de software; Comportamento dos programadores; Padrões de projeto; Ferramentas de reuso.

Abstract

One of the challenges for developers and software development companies is the implementation of source codes in a standardized way. To assist in this process, there are some design patterns that provide directions such as software reuse. In this sense, this study aimed to assess the behavior of developers during the implementation of a software reuse support tool. The evaluation was carried out through the application of questionnaires before and after the implementation of the software reuse support tool. After performing the analysis of the questionnaires, it can be seen that the developers understand the concept of software reuse. However, they use the tool inappropriately. With this, it is clear the need for a commitment from the entire developer team together with the project managers.

Keywords: Software reuse; Behavior of programmers; Design patterns; Reuse tools.

Resumen

Uno de los desafíos para los desarrolladores y las empresas de desarrollo de software es la implementación de códigos fuente de manera estandarizada. Para ayudar en este proceso, existen algunos patrones de diseño que brindan instrucciones, como la reutilización del software. En este sentido, este estudio buscó evaluar el comportamiento de los desarrolladores durante la implementación de una herramienta de soporte a la reutilización de software. La evaluación se realizó mediante la aplicación de cuestionarios antes y después de la implementación de la herramienta de soporte de reutilización de software. Después de realizar el análisis de los cuestionarios, se puede ver que los desarrolladores comprenden el concepto de reutilización de software. Sin embargo, usan la herramienta de manera inapropiada. Con esto, queda clara la necesidad de un compromiso de todo el equipo de desarrolladores junto con los jefes de proyecto.

Palabras clave: Reutilización de software; Comportamiento de programadores; Patrones de diseño; Herramientas de reutilización.

1. Introdução

Um software pode ser visto com uma coleção de produtos criados durante a execução de um conjunto determinado de passos: (i) o levantamento de requisitos com o cliente; (ii) a definição da arquitetura que será utilizada para o desenvolvimento; (iii) a escolha de uma metodologia para realizar o desenvolvimento; (iv) o processo de desenvolvimento em si; (v) a fase de testes; (vi) a entrega do software para o cliente (vii) e o acompanhamento do produto para manutenção e ajustes (Sommerville, 2011).

De modo a contribuir para a qualidade no desenvolvimento de software, podem ser empregadas técnicas de reúso de software, as quais tem o objetivo de melhorar a qualidade e facilitar o desenvolvimento de novos projetos (Pressmann & Maxim, 2016). O reúso de software é dependente do entendimento da importância deste processo por parte do desenvolvedor. Deste modo, pode-se dizer que o desenvolvedor apresenta um papel significativo no sucesso do processo de reúso de software (Sommerville, 2011). Diante disso, o objetivo deste trabalho foi analisar o comportamento dos desenvolvedores durante a implantação e utilização de uma ferramenta de apoio ao reúso de software em uma pequena empresa de desenvolvimento de sistemas de gestão.

O reúso de software é uma técnica que propõe que, soluções implementadas em projetos anteriores, possam ser utilizadas no desenvolvimento de novos projetos. As principais estratégias para a utilização do reúso de software são: (i) padrões de projeto, (ii) frameworks e (iii) componentes (Pressmann & Maxim, 2016). Segundo Jha et al (2011), a reutilização de software “tem sido considerada como uma das áreas mais importantes para melhorar a produtividade de desenvolvimento e da qualidade do software”.

Um dos principais tipos de reúso é o baseado em frameworks, os quais fornecem recursos genéricos, passíveis de serem implementados em todos os softwares similares (Pressman & Maxim, 2016). Como exemplo, pode-se citar um framework de interface de usuário, que fornece apoio para tratamento de eventos de interface e uma estrutura de apoio à construção de displays. Assim, entende-se um framework como uma ferramenta que auxilia no desenvolvimento de software, fornecendo recursos para o desenvolvimento de aplicações de um software (Sommerville, 2011).

O desenvolvimento baseado em componentes é outra estratégia de reúso de software. Sommerville (2011, p. 315) define que os componentes “são abstrações de nível mais alto do que os objetos e são definidos por suas interfaces, geralmente, eles são maiores que objetos individuais e todos os detalhes de implementação são escondidos de outros componentes”. Então, pode-se afirmar que um componente é um artefato de software com uma interface definida, para acesso aos serviços oferecidos pelo componente. Isto auxilia no desenvolvimento de software uma vez que não é necessário recriar novas funcionalidades e sim usar um componente que fornece essa funcionalidade. Consequentemente, reduzindo o tempo de desenvolvimento (Rossi, 2004).

Outra forma de reúso de software são os padrões de projeto. Um padrão de projeto é uma solução já desenvolvida e reutilizável para um conjunto de problemas que poderão ocorrer durante o desenvolvimento de um software. Assim, um padrão é um modelo de como pode ser resolvido um problema. Segundo Gamma (2000, p.18), “Os padrões de projeto tornam mais fácil reutilizar projetos e arquiteturas bem-sucedidas. Expressar técnicas testadas e aprovadas as torna mais acessíveis para os desenvolvedores de novos sistemas”.

Considerando que não há uma única abordagem para a reutilização de software, o sucesso da aceitação do reúso de software dentro de uma organização deve considerar algumas medidas. Inicialmente deve-se realizar uma análise organizacional, na qual, avalia-se a posição da empresa perante o reúso, facilitando a escolha da técnica correta do reúso para cada organização. Subsequentemente deve-se realizar uma conscientização dos envolvidos na adoção do reúso de software. Assim, a aceitação da

mudança pode ocorrer com menos impacto aos desenvolvedores. E por último, é necessário cuidado na escolha da ferramenta que será utilizada para o reúso, pois é ela que fornecerá o suporte aos envolvidos (Sherif & Vince, 2003).

1.1 Reúso nas organizações

O reúso de software apresenta vários benefícios para uma organização. No entanto, existem alguns obstáculos que necessitam serem superados para a obtenção de todas as vantagens do reúso de software. Nesta subseção serão apresentados tanto os fatores que dificultam o reúso quanto os que promovem o reuso nas organizações.

Moriso et al. (2002), citou algumas variáveis que facilitam a adoção do reúso, dentre as quais destacaram-se: o tipo de projeto, as linhas de produto e a experiência dos desenvolvedores. Já um dos fatores que dificulta a adoção do reúso, é exemplificado com uma expressão utilizada por Agresti (2010), o “not invented here” (Não inventado aqui). Conforme Ferreira e Naves (2011), os desenvolvedores “tendem a reescrever componentes, pois acreditam que podem melhorá-los”. Entretanto, ao recriar as funcionalidades, os desenvolvedores enfrentam as mesmas dificuldades que já foram encontradas e superadas anteriormente. Deste modo, deve-se conscientizar os desenvolvedores a reutilizar código (Fraga et al, 2019).

Outro obstáculo citado por Ezran et al. (2002 apud Ferreira 2011), é a criação e manutenção de uma biblioteca de componentes. Na implantação do reúso de software dentro de uma organização, o tempo de catalogação de um componente pode ser muito maior do que o desenvolvimento da funcionalidade fornecida por ele. Porém, a utilização de um componente já desenvolvido em um projeto novo, torna o projeto mais estável, diminuindo a incidência de erros já tratados no componente, assim, aumentando sua qualidade.

A forma de gerenciamento dos projetos também é uma barreira na adoção do reuso. De acordo com Sherif e Vince (2003), existem problemas decorrentes da incompatibilidade com a estrutura das organizações tradicionais, por essas utilizarem metodologias tradicionais como o modelo cascata de desenvolvimento. Outro fator relevante são as ferramentas de apoio ao reúso, as quais podem ser uma barreira técnica a ser enfrentada, devido à dificuldade de se encontrar uma ferramenta que possibilite a localização de ativos reusáveis adequados para o desenvolvimento (Keswani, 2014; Padhy et al, 2019). Considerando esse cenário, a próxima seção descreve algumas ferramentas de apoio ao uso de software disponíveis.

1.2 Ferramentas de apoio ao reuso de software

Existem, no ambiente de desenvolvimento de software, algumas ferramentas específicas que apoiam o reúso de software. Nesta subseção serão apresentadas ferramentas que auxiliam na adoção do reúso de software.

A FARCSOft foi elaborada por Rossi (2004) com objetivo de auxiliar no desenvolvimento baseado em componentes, uma vez que possibilita o armazenamento de informações e apresenta um repositório de componentes reutilizáveis. A FARCSOft facilita o gerenciamento, a busca, o entendimento de como fazer a integração do componente no software que está sendo desenvolvido. Conforme descrito por Rossi (2004), esta ferramenta aceita fragmentos de códigos, funções, classes, frameworks, executáveis e até outros elementos de sistema. Para que seja possível ser feito o compartilhamento desses ativos reutilizáveis, é preciso definir e documentar as informações, a fim de auxiliar no entendimento de sua funcionalidade e na forma de integração do artefato, na construção de outro software.

Santos (2006) desenvolveu o BART (Basic Asset Retrieval Tool) para auxiliar na recuperação de artefatos para reúso, os quais podem ser documentos, códigos, planilhas, entre outros. A pesquisa desses artefatos pode ser por uma interface web ou a partir de extensões para as ferramentas Eclipse®, Visual Studio® e Microsoft Word®. Conforme o autor, essa ferramenta apresenta baixo nível de alteração organizacional dentro da empresa, além de um formato de busca inteligente. A ferramenta se divide em duas partes, a de busca e a de administração. A interface de busca provê acesso aos artefatos indexados. Os artefatos são mantidos em espaços de trabalho (workspaces) em um formato lógico, os quais garantem os controles de acessos e a

organização dos dados. Os acessos são definidos pela interface de administração. Com isso, apenas os usuários que têm as permissões podem acessar as informações armazenadas no espaço de trabalho.

O AMT (Asset Management Tool), implementada por Wang (2008), tem como seu objetivo facilitar o desenvolvimento de software baseado em componentes (Component Based Software Development - CBSA) utilizando uma gerência de artefatos baseado em RAS. Para isso se tornar possível, a ferramenta implementa o perfil padrão de componente (Default Component Profile) do RAS, adicionando as seções: Domain, Architecture, Analysis, Patterns e Test Cases. O objetivo principal dessas seções é permitir o desmembramento de mais artefatos reutilizáveis dentro de um mesmo artefato (Almeida, 2009).

Moura (2013) desenvolveu uma solução chamada Software Profile RAS, a qual tem como objetivo criar um repositório de ativos reutilizáveis. O objetivo da solução foi contribuir com a padronização das informações e artefatos para auxiliar no processo de reúso de software. A solução é baseada em RAS (Reusable Asset Specification ou Especificação de Ativos Reutilizáveis), o qual consiste em um conjunto de diretrizes e recomendações sobre a estrutura, conteúdo e descrições dos ativos, com objetivo de promover a padronização do empacotamento do ativo para reduzir os conflitos na reutilização (OMG, 2005). O autor salienta que a ferramenta possui funcionalidades como: classificação pelo tipo do ativo, classificação com relevância pela qualidade avaliada pelos usuários, classificação por custos e esforços de desenvolvimento, classificação por palavras-chave (tags), uso com perfis de usuários publicadores, autores e consumidores, comentários de usuários, para a comunicação de dúvidas, sugestões ou defeitos pelos usuários. No entanto, a ferramenta apresenta algumas limitações, como o grau de acessibilidade do ativo relacionado ao nível de experiência do usuário, a automatização das tarefas de compilação (deploy) de artefatos e de execução dos ativos.

A Rational Asset Manager (RAM), é uma ferramenta desenvolvida pela IBM que oferece uma biblioteca definitiva para ajudar as organizações a gerenciar e controlar os ativos técnicos e de negócios envolvidos na entrega de software e sistemas (IBM, 2015). A RAM utiliza o padrão RAS para estruturar o metadados, além de possuir um sistema de pontuação para melhor identificar os artefatos, essa pontuação também é utilizada no sistema de busca (Almeida, 2009). Adicionalmente, a ferramenta apresenta um cenário de uso diferente para cada perfil de usuário que podem ser: analistas, arquitetos, desenvolvedores e testadores.

A Reusable Asset Management Tool ARCSeeker foi desenvolvida pela Sparx Systems (2016) e suporta o reúso de modelos UML desenvolvidos na ferramenta Enterprise Architect (EA), da Sparx Systems. A ferramenta possibilita ligar modelos UML com sua documentação, arquivos-fonte e armazená-los como componentes. A partir disso, esses componentes podem ser visualizados, pesquisados e recuperados. Podem ser destacadas as funcionalidades de criação de componentes e configuração de suas propriedades, armazenando em pastas virtuais, históricos de componentes e suas versões, navegação por categoria de componentes, busca por termos e notificações de alterações via e-mail. O ARCSeeker utiliza o padrão RAS para empacotar o modelo UML e os arquivos relacionados, oferecendo suporte à especificação.

Cada ferramenta descrita nessa seção foi analisada segundo algumas características elencadas como relevantes para os objetivos deste trabalho. Assim, avaliou-se os seguintes quesitos: (i) código aberto (pois permite liberdade de ajustes no software para suprir necessidades específicas); (ii) disponibilidade (a ferramenta deve ser possível de ser utilizada localmente); (iii) formas de busca do ativo dentro da ferramenta (facilidade para a localização dos ativos dentro da ferramenta), conforme apresentado na Quadro 1.

Quadro 1. Comparativo entre as ferramentas.

	Código aberto	Disponibilidade	Formas de busca
Profile RAS	X	X	tags, autores.
FARCSOft	--	--	--
BART	--	--	Palavras-chave, marcadores
RAM	--	X	--
ARCSeeker	--	--	--
AMT	X	--	--

Fonte: Autores.

Considerando as características das ferramentas analisadas, verificou-se a necessidade de criação de uma ferramenta própria devido às peculiaridades da organização na qual o estudo foi realizado. Outro obstáculo encontrado, foi a indisponibilidade das ferramentas para a estrutura de desenvolvimento da organização, visto que a linguagem Delphi é a utilizada para o desenvolvimento de seus produtos. Nesse contexto, a ferramenta desenvolvida supre todas as necessidades que foram destacadas na organização para a implantação de um repositório de reuso na empresa X.

O SisRepI (Sistema de Repositório Interno), é uma ferramenta desenvolvida pelo autor do presente trabalho, com o objetivo de prover o apoio ao reuso de software de uma forma simplificada. A ferramenta cria um repositório de ativos em um banco de dados SQL Server Express o qual possibilita o armazenamento de qualquer forma de ativos de reuso como: trecho de código, componentes e módulos. Além disso, como é de desenvolvimento próprio, qualquer necessidade de mudança pode ser realizada de forma rápida e com baixo custo. O cadastro do ativo na ferramenta se dá através de uma interface web, na qual o usuário preenche as informações relevantes em relação ao ativo. A localização dos ativos na ferramenta pode ser realizada através de palavras-chaves, data de criação, autor e descrição do ativo. O SisRepI, contém um controle de acesso dos usuários, o que garante uma maior segurança ao acesso das informações geradas internamente. O sistema não é baseado em RAS.

1.3 Trabalhos Relacionados

Digiampietri et al. (2013) abordam a combinação de workflows e semântica para facilitar o reuso de software. Os autores, nesse caso, propuseram que cada componente de software a ser reutilizado torna-se uma tarefa de um workflow e cada novo software desenvolvido é um novo workflow. Os autores justificam a utilização da semântica neste contexto como forma de auxiliar a descoberta, composição e orquestração automática de serviços, facilitando, contudo, o compartilhamento e reuso de dados. Para a análise do workflow proposto pelos autores, foram realizados dois estudos de caso: o primeiro estudo foi utilizado apenas para exemplificar e ilustrar o sistema desenvolvido; já o segundo estudo, apresentou uma solução aplicada a um problema de processamento de imagem. Com base nos resultados obtidos, os autores ressaltaram que existem vantagens potenciais no projeto de software, baseado em componentes. Exemplos citados foram: a redução do tempo de produção, a redução de custos e o aumento da qualidade.

Em seu estudo, Shiroky (2002) realizou a modelagem de um repositório de componentes que possuía uma gestão do gerenciamento através de serviços diversos executados de forma cooperativa e integrada. Para a realização de seu estudo o autor definiu três etapas: (i) etapa 1: estudos e definições dos serviços propostos; (ii) etapa 2: estudo sobre a linguagem Java e suas APIs (Application Program Interface), com destaque nas que implementam persistência, serialização de objetos e invocação remota de métodos; (iii) etapa 3: Na última etapa, foi realizada a construção do protótipo de um repositório de componentes de software. Ao implementar o modelo de repositório, o autor disponibilizou um ambiente de desenvolvimento integrado. Com isso,

garantiu que componentes de software possam ser desenvolvidos uma única vez e distribuídos em qualquer ambiente operacional, provendo um ciclo de vida consistente e serviços de controle de transações, persistência, distribuição e segurança.

Rossi (2004), por sua vez, realizou um estudo com intuito de definir uma representação e um padrão de documentação de diferentes tipos de componentes de software em um repositório. Para isso, foi realizada a avaliação do meta-modelo, na qual assume o papel de descrever a estrutura do componente, seus relacionamentos e semântica dos dados armazenados. Os dados dos componentes foram mapeados, conforme ferramenta utilizada (FARCSOFT). A autora ressalta que existe uma necessidade de ferramentas que forneçam suporte às atividades de produção, consumo e gerência de componentes de softwares reutilizáveis. Desta forma, Rossi (2004) definiu a estrutura do repositório de componentes reutilizáveis, estabelecendo um conjunto de informações para representar e documentar diferentes tipos de software. Em relação às principais dificuldades encontradas no decorrer do trabalho, a autora descreve a falta de qualidade e de documentação para auxiliar no entendimento das funcionalidades dos componentes selecionados.

2. Metodologia

O estudo se caracterizou como uma pesquisa de natureza qualitativa e de nível exploratório. Considerando o objetivo deste trabalho, a pesquisa qualitativa é adequada visto que descreve a complexidade do problema e a interação com as variáveis. Assim, facilitando a compreensão e classificação de processos dinâmicos vivenciados em grupos (Gil, 2008). Em relação ao nível exploratório, este proporciona uma maior familiarização com o problema, para que com isso o problema se torne mais explícito (Diehl, 2004). Segundo Gil (2008, p.3), “A pesquisa exploratória proporciona maior familiaridade com o problema. Pode envolver levantamento bibliográfico, entrevistas com pessoas experientes no problema pesquisado”.

A estratégia de pesquisa utilizada foi a de estudo de caso, que segundo Diehl (2004) “apresenta uma série de vantagens, o que faz com que se torne o delineamento mais adequado em várias situações”. Yin (2001), complementa que, o estudo de caso não é um método simples e fácil de ser aplicado. Ele depende que o pesquisador consiga, por exemplo: (i) fazer boas perguntas, simples e diretas; (ii) ouvir com atenção, e saber separar suas ideologias e preconceitos; (iii) ser claro, ser teórico, político ou exploratório das questões que estão sendo estudadas. Assim, o estudo de caso foi realizado em uma empresa de desenvolvimento de software, aqui denominada Empresa X, localizada na região da serra do estado do Rio Grande do Sul.

O processo de coleta de dados deu-se através da aplicação de um questionário semiestruturado, análise de documentos e de observações diretas realizadas no ambiente do estudo de caso (Diehl, 2004). A construção do questionário foi realizada visando obter o máximo de detalhes do entendimento dos desenvolvedores em relação ao reuso de software e as perguntas foram validadas por um avaliador externo antes de sua aplicação. A aplicação do questionário foi realizada antes da implantação de uma ferramenta de apoio ao reuso. O questionário tinha duas perguntas principais: (i) “Você costuma reutilizar componentes de software? De que forma o realiza? Em caso contrário, qual o motivo?”; (ii) “Os seus colegas reutilizam componentes de software? De que maneira? Em caso contrário, você sabe o motivo?”.

O mesmo questionário foi reaplicado após a utilização da ferramenta SisRepI pelo período de 30 dias úteis, acrescido da seguinte pergunta: “O software implantado para incentivar a reutilização dentro da organização contribuiu para iniciar ou promover a reutilização dentro da organização?”. Os respondentes foram os desenvolvedores da empresa, totalizando seis questionários respondidos, sendo três antes e três após a utilização da ferramenta, a pesquisa ocorreu no período de 10 de abril de 2019 a 15 de maio de 2019.

O processo de análise de dados foi realizado através da técnica de análise de conteúdo. A análise de conteúdo pode ser descrita como um conjunto de técnicas, na qual, o objetivo é a identificação do que é dito a respeito em determinado tema (Minayo, 2001). A análise de conteúdo constitui uma técnica que trabalha os dados coletados, objetivando a identificação do que está sendo dito a respeito de um determinado tema (Vergara, 2005).

3. Resultados e Discussão

Após a aplicação da metodologia descrita, foi possível verificar como o reúso de software foi aplicado pelos desenvolvedores da Empresa X. Assim, esta seção está estruturada em três partes principais, que contemplam: a descrição da empresa, a apresentação dos dados obtidos por meio da aplicação dos questionários, e a discussão dos resultados obtidos.

3.1 Participantes do estudo

A empresa X foi fundada em 1994 por dois sócios. Atualmente conta com sete colaboradores, sendo dois do setor comercial, um do setor financeiro e três desenvolvedores. A empresa X é uma empresa de desenvolvimento de software de pequeno porte, sendo o seu principal produto o software para automação comercial e gestão industrial do ramo de moda íntima e jóias. O desenvolvimento dos produtos da empresa já utiliza o reúso através de componentes e criação de módulos. No entanto, antes da implantação do SisRepI como ferramenta de apoio, não existia uma organização no armazenamento dos ativos de software.

Como apresentado por Bergamini (2002), o gerente tem o trabalho de fazer com que a organização mantenha sua identidade, controlando os prazos e mantendo um bom funcionamento do sistema. Ademais, desempenha o papel de intérprete, coordenando sua equipe pelos caminhos escolhidos através de suas decisões. Apesar de essa ser a função da gerência, os gestores da Empresa X não executam esta atividade permitindo que os desenvolvedores tenham total autonomia em seus trabalhos, sem seguir padrões definidos de implementação, validação, testes, orçamento, etc. O Quadro 2 apresenta as características formação, cargo, tempo de empresa e idade dos desenvolvedores da Empresa X.

Quadro 2. Perfil dos desenvolvedores do estudo de caso.

Desenvolvedor	Formação	Cargo	Tempo Empresa	Idade
A	Curso técnico em análise e desenvolvimento de sistemas	Desenvolvedor e suporte	11 anos	33
B	Superior incompleto em análise e desenvolvimento de sistemas	Desenvolvedor e suporte	2 anos	20
C	Graduação em análise e desenvolvimento de sistemas	Desenvolvedor e suporte	5 anos	27

Fonte: Autores.

Para minimizar os impactos da implantação de ferramenta de apoio ao reúso de software foram realizadas reuniões com os desenvolvedores a fim de promover a compreensão dos conceitos do reúso de software e as vantagens na utilização. Após a implantação da ferramenta foi realizado o treinamento sobre a forma como os ativos deveriam ser inseridos e catalogados.

3.2 Análise dos questionários

Antes de implantar uma ferramenta SisRepI, os desenvolvedores foram questionados sobre suas noções de reúso de software. Após a aplicação do primeiro questionário, pode-se perceber que os desenvolvedores não apresentaram a compreensão sobre a complexidade do processo do reúso de software. Quando questionados se realizavam esse processo e de que forma, os desenvolvedores responderam de formas distintas:

“Sim, na forma de padrão de projeto. Desenvolvo o projeto em questão baseado em um padrão pré-definido, desta forma, o tempo de desenvolvimento do projeto se torna menor e o código fica melhor compreensível para o caso de um colega utilizar o mesmo projeto.” (Desenvolvedor A)

Entretanto outro desenvolvedor respondeu:

“Sim, costumo reutilizar código. Usamos alguns projetos individuais de cada cliente, assim, quando algum cliente solicita algo novo semelhante, faço a reutilização do código fonte. Também faço o mesmo quando devo iniciar um novo projeto. Utilizo também componentes prontos no desenvolvimento, os quais tornam o trabalho mais eficiente e prático.” (Desenvolvedor B)

Podemos perceber que, os desenvolvedores têm percepções diferentes sobre o que é a reutilização de software, no qual, o Desenvolvedor A, busca a reutilização através de padrões de projetos pré-definidos, o Desenvolvedor B, evidencia a reutilização por meio de projetos já desenvolvidos anteriormente.

Após realização da análise dos dados obtidos com a aplicação do primeiro questionário, pode-se perceber que os desenvolvedores não conseguem compreender o conceito do reuso de software que é: reciclar o design, código e outros componentes de um software, reduzir o custo, o tempo e melhorar a qualidade do produto. Conforme afirma Keswani (2014), uma das razões para essa dificuldade está no fato que os desenvolvedores envolvidos no processo, carecem de competências necessárias para criar ou utilizar os ativos de reuso (Keswani, 2014).

Outra questão aplicada aos desenvolvedores, foi se eles sabiam se os seus colegas utilizavam o reuso de software e de que maneira a faziam. Para essa questão, todos os desenvolvedores afirmam que tinham a percepção que os seus colegas também utilizavam técnicas de reutilização de software.

Após a implantação da ferramenta e de sua utilização pelos desenvolvedores pelo período de 30 dias, o questionário foi reaplicado. Neste momento, foi possível perceber uma alteração nas respostas dos desenvolvedores, na qual destaca-se a seguinte ideia: “Sim, realizamos a reutilização de software através de componentes de terceiros, módulos de desenvolvimento próprio e funções já existentes” (Desenvolvedor B).

Desta forma, podemos ressaltar que, existe um alinhamento entre o que foi respondido pelo Desenvolvedor B e o que foi descrito por Rossi (2004) e Sommerville (2011). No qual os autores descrevem a reutilização utilizando componentes. Os quais podem ser desenvolvidos internamente ou por terceiros.

Os desenvolvedores identificaram, também, que os seus colegas realizam a reutilização de software da mesma forma. Adicionalmente às questões sobre reuso, os desenvolvedores responderam à seguinte questão: “O software implantado incentiva a reutilização dentro da organização contribuiu para iniciar ou promover a reutilização dentro da organização? Comente.”.

Neste sentido, os colaboradores responderam:

Promove, pois esta reutilização já havia sido feita, porém, com a implantação do software profissionalizou este controle tornando-o muito mais confiável e seguro (Desenvolvedor A).

O desenvolvedor B, complementou:

[...] Muitas vezes não somos capazes de lembrar se o que nos foi solicitado já existe no sistema ou se existe algo semelhante. O software auxilia promovendo a reutilização do código e, com isso, ganhamos tempo para realizar as demais atividades (Desenvolvedor B).

Considerando a pergunta, “Você costuma reutilizar código? Se você reutiliza, de que forma faz isso? Se não reutilizar, qual o motivo por não reutilizar?”, foi possível verificar uma alteração nas respostas dos colaboradores antes e depois da implantação da ferramenta SisRepI, conforme Quadro 3.

Quadro 3. Comparação de respostas antes de depois da utilização da ferramenta.

Desenvolvedor	Antes da utilização da Ferramenta	Depois da utilização da Ferramenta
A	“Reutilizo código, basicamente aproveitando projetos existentes na empresa, juntamente com componentes criados por colegas”	“Realizamos a reutilização de <i>software</i> através de componentes de terceiros, módulos de desenvolvimento próprio e funções já existentes”
B	“Sim, costumo reutilizar código. Usamos alguns projetos individuais de cada cliente, assim, quando algum cliente solicita algo novo semelhante, faço a reutilização do código fonte. ”	“Sim, reutilizam, utilizando blocos de código fonte, funções ou procedimentos já existentes e modificando ou não, conforme necessário. ”
C	"Sim, procuro sempre analisar o que preciso desenvolver com o que já está em uso ou foi desenvolvido anteriormente, buscando funções, partes de código e até mesmo ideias que possam ser úteis."	"Sim, buscando funcionalidades que se encaixem nos desenvolvimentos em progresso"

Fonte: Autores.

De acordo com as considerações realizadas pelos colaboradores antes e depois da implantação da ferramenta de apoio ao reuso de software, pode-se perceber que os desenvolvedores conseguiram compreender o funcionamento e aplicação do reuso de software dentro da organização. Podemos ressaltar o pivotamento da resposta do Desenvolver A, quando ele consegue reconhecer os tipos de reuso que ele utiliza, destacando-se a seguinte frase: “Realizamos a reutilização de software através de componentes de terceiros, módulos de desenvolvimento próprio e funções já existentes”. Sendo que, anteriormente o Desenvolver A, apenas reconhecia como reuso os códigos criados por ele e seus colegas de empresa.

No entanto, ao analisar os dados inseridos na ferramenta de apoio, foram encontrados não só ativos de reuso, mas também tutoriais e manuais, assim fazendo que uma ferramenta específica de apoio ao reuso, se torne um repositório de documentos, alterando o objetivo específico da ferramenta. Como citado por Ezran et al. (2002 apud Ferreira, 2011), a manutenção da biblioteca e ativos se torna complexa aos desenvolvedores inserirem na ferramenta arquivos que não são ativos, desta maneira inflando o banco de dados da ferramenta e dificultando a localização dos ativos.

As necessidades específicas de cada cliente é outra adversidade apresentada na implementação do reuso dentro de uma organização. Quando existe uma heterogeneidade no tipo de cliente da empresa, algumas alterações requisitadas por eles podem não ser adequada para os outros clientes (Filho, 2013).

4. Conclusão

Este artigo apresentou a análise qualitativa do comportamento dos desenvolvedores durante a implantação e utilização de uma ferramenta de reuso de software. Nas entrevistas realizadas antes e depois da implantação da ferramenta, pode-se perceber que os desenvolvedores conseguiram compreender a essência do reuso, porém, os resultados mostraram outra perspectiva.

Os dados obtidos permitiram verificar que, mesmo realizando uma orientação do funcionamento da ferramenta, os desenvolvedores seguem apenas suas convicções. A alteração desse cenário exige o comprometimento dos gerentes de projeto, dos gestores e dos próprios desenvolvedores envolvidos no processo de implantação da ferramenta de apoio ao reuso de software.

Um ponto importante a ser ressaltado, é a falta de organização dos processos de desenvolvimento dentro da Empresa X, na qual, os desenvolvedores não seguem um padrão para realizar as alterações do sistema ou novos desenvolvimentos, deste modo, dificultando o uso de uma ferramenta de apoio ao reuso.

Como trabalhos futuros, a evolução da ferramenta desenvolvida pelo autor, para que ela se torne um extrator automático de código-fonte. Desta maneira, evitando que os usuários da ferramenta realizem um cadastramento incorreto dos ativos, consequentemente, tornando os dados mais confiáveis e corretos.

Referências

- Agresti, W. W. (2011). Software Reuse: Developers' Experiences and Perceptions. *Journal of Software Engineering and Applications*, 04(01), 48. <https://doi.org/10.4236/jsea.2011.41006>
- Almeida, T. O. (2009). *Ras4Nexus: Promovendo reuso utilizando o gerenciador de repositórios nexus com o padrão ras*. <https://lume.ufrgs.br/handle/10183/18552>
- Bergamini, C. W., & Beraldo, D. G. R. (2002). Cecília Whitaker. *O líder eficaz*. São Paulo: Atlas.
- Diehl, A. A., & Tatim, D. C. (2004). *Pesquisa em ciências sociais aplicadas: métodos e técnicas*. Pearson Brasil.
- Digiampietri, L. A., Araújo, J. C., Ostroski, É. H., Santiago, C. R. N., & Alcázar, J. D. J. P. (2013). Combinando workflows e semântica para facilitar o reuso de software. *Revista de Informática Teórica e Aplicada*, 20(2), 73-89. <https://doi.org/10.22456/2175-2745.23471>
- Ferreira, H., & Naves, T. F. (2011). Reuso de software: Suas vantagens, técnicas e práticas. *IX. Enacom*, 4.
- Fraga, A., Llorens, J., & Génova, G. (2019). Towards a methodology for knowledge reuse based on semantic repositories. *Information Systems Frontiers*, 21(1), 5-25. <https://doi.org/10.1007/s10796-018-9862-7>
- Gamma, E. (2000). Tradução de Luiz A. Meireles Salgado. *Padrões de Projeto: Soluções Reutilizáveis de Software Orientado a Objetos*. Porto Alegre: Bookman.
- Gil, A. C. (2002). *Como elaborar projetos de pesquisa* (Vol. 4, p. 175). São Paulo: Atlas.
- Ibm (2015). *Rational Asset Manager Family*. <http://www-03.ibm.com/software/products/pt/ratiassemanafami>
- Keswani, R., Joshi, S., & Jatain, A. (2014, February). Software reuse in practice. In *2014 Fourth International Conference on Advanced Computing & Communication Technologies* (pp. 159-162). IEEE.
- Jha, M., & O'Brien, L. (2011, December). A comparison of software reuse in software development communities. In *2011 Malaysian Conference in Software Engineering* (pp. 313-318). IEEE.
- Jones, C. (1993). Software return on investment preliminary analysis. Software Productivity Research. Inc., Burlington, MA.
- Minayo, M. C. D. S. (2001). O desafio do conhecimento: pesquisa qualitativa em saúde. São Paulo-Rio de Janeiro, HUCITEC-ABRASCO, 1992. *Pesquisa Social: teoria, método criatividade*. Petrópolis: Vozes.
- Morisio, M., Ezran, M., & Tully, C. (2002). Success and failure factors in software reuse. *IEEE Transactions on software engineering*, 28(4), 340-357.
- Moura, D. D. S. (2013). Software Profile RAS: estendendo a padronização do Reusable Asset Specification e construindo um repositório de ativos.
- Omg (2005). *Reusable Asset Specification. OMG Available Specification*. Version 2.2. Object Management Group, Inc. <http://www.omg.org/spec/RAS/2.2/>
- Padhy, N., Panigrahi, R., & Satapathy, S. C. (2019). Identifying the reusable components from component-based system: proposed metrics and model. In *Information systems design and intelligent applications* (pp. 89-99). Springer, Singapore. <https://doi.org/10.1007/s10586-017-1558-0>
- Pressman, R., & Maxim, B. (2016). *Engenharia de Software-(8ª Edição.)* McGraw Hill Brasil.
- Rossi, A. C. (2004). *Representação do componente de software na FARCSOft: ferramenta de apoio à reutilização de componentes de software* (Doctoral dissertation, Universidade de São Paulo).
- Santos, E. C., Durão, F. A., Martins, A. C., Mendes, R., Melo, C., Garcia, V. C., & Meira, S. D. L. (2006). Towards an effective context-aware proactive asset search and retrieval tool. *WDBC*, 6, 105-112.
- Schiroky, A. M. (2002). Modelagem e prototipação de um repositório extensível para componentes de software.
- Sparx, systems, *Reusable Asset Management Tool: ARCSeeker*. <http://www.arcseeker.com>
- Sommerville, I. Engenharia de Software. Tradução Ivan Bosnic e Kalinka G. de O. Gonçalves; revisão técnica Kechi Hiramã. (9ª. Ed.): Pearson Prentice Hall, 2011.
- Wang, Ze-Sheng., et al. *Design and implementation of ras-based reusable asset management tool*. In: *Internet Computing in Science and Engineering, 2008. ICICSE'08. International Conference on*. IEEE, 2008. p. 363-366.