

Gestão do escopo para o projeto Biblioteca de reuso de requisitos de *software* para diferentes domínios

Scope management for the project Library of reuse of software requirements for different domains

Mariana Abreu Gualhano Mestre, Instituto Federal Fluminense (IFFluminense), Brasil – mariana_gualhamo@hotmail.com
Simone Vasconcelos Silva Doutora, Instituto Federal Fluminense (IFFluminense), Brasil – simonevs@iff.edu.br
Aline Pires V. de Vasconcelos Doutora, Instituto Federal Fluminense (IFFluminense), Brasil – apires@iff.edu.br

RESUMO

O processo de desenvolvimento de *software* requer o cumprimento de várias etapas, onde a gestão de requisitos mostra-se como uma das mais complexas e importantes. Uma plataforma que possibilite o acesso de equipes de desenvolvimento de *software* a uma biblioteca *on-line* contendo um conjunto de requisitos para o domínio que se deseja implementar, proporcionando o reuso destes requisitos, pode trazer muitos benefícios em relação à qualidade, tempo e custos. Neste contexto, um projeto de estruturação de uma biblioteca de reuso de requisitos de *software* para diferentes domínios é de grande interesse para a comunidade de desenvolvimento de *software* (desenvolvedores e/ou organizações). Este artigo tem como objetivo propor o projeto "Biblioteca de reuso de requisitos de *software* para diferentes domínios" através da elaboração de dois documentos essenciais da gestão de escopo, a Estrutura Analítica do Projeto – EAP (escopo do projeto) e a Lista de Requisitos (escopo do produto). Para elaboração deste trabalho foram realizadas as seguintes etapas: (i) estudo sobre gestão de escopo e engenharia de requisitos com ênfase na elicitação de requisitos e bibliotecas de reuso; (ii) elaboração e validação da EAP através de entrevistas com pesquisadores sêniores da área de Computação; (iii) elaboração da Lista de Requisitos; (iv) validação da Lista de Requisitos através de pesquisa de campo, utilizando questionários, com profissionais da área de Tecnologia da Informação (TI). Através deste trabalho pode-se concluir que a Biblioteca proposta, representada pelos documentos de gestão do escopo (projeto e produto), mostrou-se importante e capaz de gerar benefícios para área de TI.

Palavras-chave: Gestão. Biblioteca de Reuso. Escopo. Requisitos.

ABSTRACT

The software development process requires the fulfillment of several steps, where the management of requirements proves to be one of the most complex and important. A platform that allows the access of software development teams to an online library containing a set of requirements for the domain that one wishes to implement, and that provides the reuse of these requirements can bring many benefits in terms of quality, time and costs. In this context, a project of structuring a library of reuse of software requirements for different domains is of great interest to the software development community (developers and / or organizations). The purpose of this article is to propose the project "Library of reuse of software requirements for different domains" through the elaboration of two essential documents of scope management, the Work Breakdown Structure - WBS (project scope) and the List of Requirements (product scope). In order to elaborate this work, the following steps were performed: (i) study on scope management and requirements engineering with emphasis on requirements elicitation and libraries of reuse; (ii) elaboration and validation of the WBS through interviews with senior researchers in the Computing area; (iii) preparation of the List of Requirements; (iv) validation of the List of Requirements through field research, using questionnaires, with professionals in the area of Information Technology (TI). The research led to the conclusion that the proposed Library, represented by the documents of scope management (project and product), proved to be important and capable of generating benefits for the TI area.

Keywords: Management. Reuse Library. Scope. Requirements.

Recebido em 14/02/2019. Aprovado em 21/05/2019. Avaliado pelo sistema *double blind peer review*. Publicado conforme normas da APA.
<http://dx.doi.org/10.22279/navus.2019.v9n4.p80-93.915>

1 INTRODUÇÃO

Ao considerar que a velocidade das mudanças no setor tecnológico mundial é elevada e a necessidade de adaptação a essas é cada vez maior, é inconcebível uma organização de médio e grande porte não utilizar algum Sistema de Informação (SI), ou seja, um *software* automatizado para processar e gerenciar os dados relacionados ao seu negócio.

O desenvolvimento de um *software* é composto de várias etapas: elicitacão e validacão de requisitos, documentacão, desenvolvimento, testes e implantacão. Conforme Azevedo Junior e Campos (2008) a elicitacão de requisitos mostrou-se uma das etapas de maior importacão no desenvolvimento de *software*, pois a descoberta de grande parte dos problemas é originada nas etapas iniciais do desenvolvimento.

Vale ressaltar que é necessário aumentar concomitantemente a produtividade e qualidade, além da reduçã de custos no desenvolvimento desses *softwares*. De acordo com Hauksdóttir, Mortensen e Nielsen (2014) a reutilizacão de requisitos se mostrou uma soluçã alternativa para aprimorar a qualidade dos requisitos, além de diminuir os custos de desenvolvimento e tempo de mercado, e é de fundamental importacão na concretizacão dos objetivos supracitados

Reutilizar requisitos pode auxiliar na resoluçã de problemas comuns, relacionados a especificaçõ dos mesmos, como, por exemplo, a falta de padronizacão e completude, reiteram Palomares, Franch e Quer (2014), assim, a resoluçã desses problemas, resulta em uma reduçã no esforço de desenvolvimento e no tempo de trabalho.

Segundo Kaiya *et al.* (2010), a obtençã de requisitos de *software* se trata de uma abordagem complexa, visto que deve haver conhecimento sobre um domínio no qual o *software* é aplicado em suas reais necessidades.

A gestã de requisitos é um problema em qualquer projeto, não só na área de *software*. Estudo do *Project Management Institute* (PMI), desde o ano de 2012, apontam que o gerenciamento de requisitos é um grande obstáculo na área de projetos. Essa informacão acerca da complexidade da elicitacão de requisitos também é corroborada pela PMI (2017) que afirma que dos projetos iniciados nas organizaçõs, a segunda principal causa de falhas é a coleta imprecisa de requisitos.

No que se refere à importacão de reutilizar requisitos, Palomares, Quer e Franch (2017) afirmam que a Engenharia de Requisitos é uma área com vários desafios a superar, e um desses é a implementacão de abordagens de reutilizacão de requisitos, e completam atestando que embora existam várias propostas teóricas no que tange ao reuso de requisitos, as abordagens práticas atualmente adotadas são pouco conhecidas e estudadas.

De acordo com PMI (2013) o escopo do produto é definido como “características e funções que descrevem um produto, serviço ou resultado” e o escopo do projeto é definido como “o trabalho que deve ser realizado para entregar um produto, serviço ou resultado com as características e funções especificadas”. Embora exista uma forte relaçã entre o escopo do produto e do projeto, ambos são documentados em artefatos distintos, de forma que o documento mais importante na documentacão do escopo do projeto é a Estrutura Analítica do Projeto (EAP) e o documento mais importante na documentacão do escopo do produto é a lista de requisitos. A EAP é uma ferramenta essencial da área de conhecimento de gerenciamento de escopo de projetos, na qual é elaborado o processo de subdivisã das entregas e do trabalho do projeto em componentes menores e mais facilmente gerenciáveis.

Para Marquioni (2008) o escopo do produto identifica os requisitos que fazem parte do produto (o que deve ser entregue) e o escopo do projeto informa quais são as atividades que fazem parte do projeto (quais atividades devem ser executadas para que os requisitos sejam entregues). Para que a formalizacão em artefatos distintos não provoque dificuldades no tratamento em conjunto dos dois tipos de escopo que devem ser abordados em todo e qualquer projeto de *software*, uma alternativa viável é considerar a Engenharia de Requisitos como um ponto em comum entre as formas de gestã de escopo.

Os requisitos são a base para o desenvolvimento de produtos de *software*, portanto a precisã na elicitacão de requisitos desde o início do processo de desenvolvimento de *software* representa um enorme ganho no que diz respeito à qualidade, tempo e custos. Desta forma, uma plataforma que possibilite o acesso

de equipes de desenvolvimento de *software* a uma biblioteca *on-line* contendo um conjunto de requisitos para o domínio que se deseja implementar, proporcionando o reuso destes requisitos, pode agregar muitos benefícios ao processo de desenvolvimento de *software*, visto que a etapa de elicitação de requisitos é considerada, muitas das vezes, pela maioria dos projetos a etapa mais crítica, devido a sua complexidade e subjetividade, onde problemas nesta etapa podem gerar enormes prejuízos e até mesmo impossibilitar o projeto.

Nesse contexto, o presente trabalho tem como objetivo propor a estruturação de uma biblioteca de reuso de requisitos de *software* para diferentes domínios, com a finalidade de auxiliar as organizações e/ou desenvolvedores na fase inicial de um projeto de desenvolvimento de *software*, onde os mesmos poderão analisar e reutilizar através do acesso e busca na biblioteca proposta um conjunto de requisitos já elicitados e validados por outras organizações e/ou desenvolvedores para o domínio desejado. Portanto, este artigo apresenta a gestão do escopo do projeto "Biblioteca de Reuso de Requisitos de *Software* para Diferentes Domínios", sendo o escopo do projeto elaborado através de uma EAP e o escopo do produto elaborado através de uma Lista de Requisitos.

O proposto trabalho está dividido como se segue: na Seção 2 é feita uma revisão bibliográfica sobre gerenciamento de escopo de projeto, engenharia de requisitos e reuso, assim como também são levantados os trabalhos relacionados existentes na literatura; na Seção 3 é definida e explicada a metodologia utilizada; na Seção 4 são explicitados os resultados obtidos e por fim, na Seção 5 a conclusão após a realização desse trabalho.

2 REVISÃO DE LITERATURA

Os temas relacionados a gerência de requisitos e engenharia de requisitos têm sido amplamente estudados, visto que a coleta e gerenciamento são aspectos de fundamental importância para o sucesso de um processo e de acordo com Pacheco *et al.* (2017) a elicitação de requisitos mostrou-se uma das etapas de maior importância no desenvolvimento de um projeto.

A partir disso, considerando um levantamento bibliométrico na base de dados SCOPUS, utilizando com filtro "artigo de periódico e artigo de conferência" observou-se que no que se refere a interseção dos assuntos Gerência de Projetos, Engenharia de Requisitos e Biblioteca de Reuso, trata-se de uma abordagem ainda pouco explorada, pois na base utilizada não foram encontrados artigos que tratassem ao mesmo tempo os três assuntos.

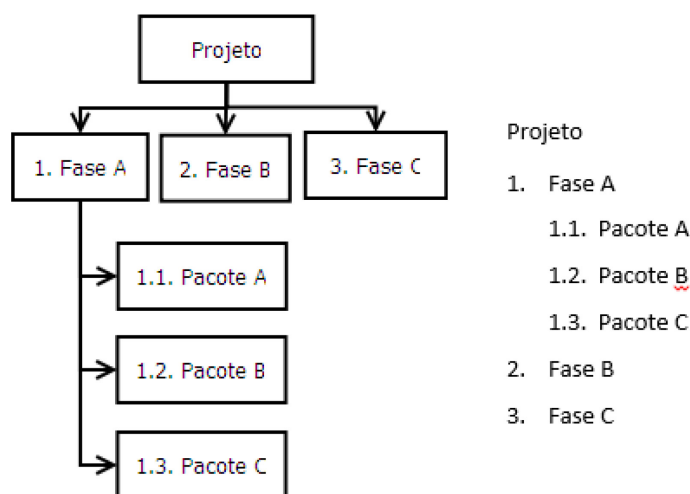
2.1 Gerenciamento de Escopo de Projetos

De acordo com o PMI (2013), em seu guia de boas práticas na gestão de projetos o *Project Management Body of Knowledge* (PMBOK), a gerência de projetos possui dez áreas de conhecimento (integração, escopo, tempo, custo, qualidade, recursos humanos, comunicação, riscos, partes interessadas e aquisição), onde uma delas é o gerenciamento de escopo. Nesta área, o escopo do projeto (trabalho que deve ser realizado para entregar um produto, serviço ou resultado com as características e funções especificadas) é gerenciado através de cinco processos: planejar o gerenciamento do escopo (elabora como será a definição, controle e validação do escopo), coletar os requisitos (documenta as necessidades das partes interessadas para atingir os objetivos do projeto), definir o escopo (desenvolve a descrição do projeto e do produto), criar a EAP (subdivide os produtos do projeto e o trabalho do projeto em componentes menores e gerenciáveis), validar o escopo (formaliza a aceitação dos produtos do projeto), e controlar o escopo (monitora o *status* do escopo do projeto e do produto, gerenciando a linha de base de escopo). Neste artigo serão abordados dois destes processos: coletar os requisitos e criar a EAP.

Martins e Melo (2016) afirmam que a EAP é uma decomposição hierárquica do escopo total do trabalho que deve ser realizada pela equipe do projeto a fim de alcançar os objetivos do projeto e criar as entregas requeridas, e tem como principal benefício o fornecimento de uma visão estruturada do que deve ser entregue.

O PMI (2013) propõe dois tipos de EAP: orientada à entrega (construída em torno de resultados ou produtos desejados do projeto) e orientada à processos (semelhante a uma EAP orientada à entrega, exceto que ela é organizada, no nível mais alto por fases ou etapas de um processo e não por resultados). A EAP define as entregas do projeto e sua decomposição em pacotes de trabalho, possuindo duas formas de representação: forma gráfica (Figura 1a) para facilitar o entendimento e a visualização, e a forma de lista numerada (Figura 1b), para quando não for possível representá-la claramente na forma gráfica devido ao tamanho e complexidade da imagem. A EAP é fundamental para o projeto, pois fornece uma visão estruturada do que será entregue, facilitando o entendimento das partes interessadas em relação ao que deve ser feito (escopo) no projeto, além de servir de base para o planejamento de outras áreas de conhecimento.

Figura 1 – (a) Forma Gráfica (b) Forma de Lista Numerada.



Fonte: Autoria própria (2019).

2.2 Engenharia de Requisitos

Para a ISO/IEC/IEEE 24765 (2017) requisito de *software* é definido como a condição necessária para o usuário resolver um problema ou alcançar um objetivo; condição de ser atendido por uma solução para atender um contrato, especificação, padrão ou outros documentos formais; e documentação da representação das condições apresentadas nos dois itens anteriores.

Os requisitos de *software* são classificados de acordo com Sommerville (2011) em requisitos funcionais, não funcionais ou de domínio. Requisitos funcionais são as atividades que o sistema deve fornecer, como o sistema deve reagir a determinadas entradas e verifica o comportamento do sistema em determinadas situações. Requisitos não funcionais especificam as restrições sobre as atividades e funções fornecidas pelo sistema (tempo, processo de desenvolvimento, padrões de qualidade de *software*, entre outras). Requisitos de domínio são originados do domínio da aplicação do sistema e refletem as características desse domínio, podendo ser funcionais ou não funcionais.

Durante o processo de engenharia de requisitos é definido o que sistema deve fazer, quais são suas propriedades essenciais, suas restrições quanto a sua operação e o seu processo de desenvolvimento (SOMMERVILLE, 2011).

Pressman (2011) define as atividades da engenharia de requisitos em sete funções distintas: concepção, levantamento, elaboração, negociação, especificação, validação e gestão. E para Sommerville (2011) o processo de engenharia de requisitos possui quatro subprocessos: estudo de viabilidade, elicitação e análise, especificação e validação dos requisitos.

Uma abordagem que tem como objetivo reutilizar partes do processo de construção de um *software* que já foi utilizado, assim como o código e outras etapas do processo, para o desenvolvimento de outros *softwares*, de acordo com Ferreira e Neves (2011) é denominada Reuso de *Software*. Para Frakes e Kyo Kang

(2005) o conceito de reutilização de *software* significa reutilizar tanto conhecimento, quanto produtos de *software*, e não se trata de um conceito novo, foi criado por especialistas durante o desenvolvimento de *software*, ao mesmo tempo em que desenvolviam nova aplicação em contextos diferentes.

De acordo com Pacheco *et al.* (2017) a reutilização de *software* tem como principal finalidade aumentar a produtividade e a qualidade do mesmo, além de fornecer benefícios econômicos. Nesse cenário, os autores validam que a reutilização de requisitos de *software*, no contexto de Reuso de *Software* destacou-se entre as outras etapas, pois fornece um suporte sólido para desenvolver um *software* de qualidade, partindo da etapa inicial do seu ciclo de vida.

Srinivas, Radhakrishna e Rao (2015) salientam a importância da escolha eficiente dos componentes de *software* para reutilização. Em sua pesquisa, propõem agrupar os projetos e os componentes de *software* dos repositórios disponíveis e a partir de uma função de similaridade, conseguir a escolha do componente de *software* mais adequado ao problema, de forma rápida e eficiente.

Vijay Rao e Sarma (2014) propõem em seu trabalho uma abordagem para recuperar e reutilizar componentes de *software* em banco de dados. Afirmam que a classificação, o armazenamento e a seleção de componentes de uma biblioteca de reutilização são considerados fatores de sucesso chave para projetos.

Goldin e Berry (2015) corroboram que a reutilização de requisitos pode afetar de maneira significativa a reutilização nas outras etapas de desenvolvimento de *software*. E afirmam que a partir disso, a mesma se tornou uma opção muito cogitada para melhorar a produtividade e a qualidade do processo de *software*.

Na literatura encontra-se alguns modelos de bibliotecas e ferramentas de reuso de *software*. Pode-se apresentar alguns exemplos das mesmas:

- Biblioteca RCR Editor – composta de componentes reutilizáveis e baseada em XML (*eXtensible Markup Language*). Os componentes considerados para reutilização no trabalho são o controle de gestão, diagramas de análise, diagrama de estrutura, código e linguagem de programação, além de restrições do sistema operacional. A biblioteca possui uma interface amigável, na qual os usuários podem incluir, editar e utilizar os componentes (CHANG *et al.*, 2011);
- Biblioteca Brechó - permite o desenvolvimento de extensões. De acordo com Santos, Tostes e Werner (2013) a Brechó é um sistema de informação *Web* com uma base de aplicações, componentes, serviços, produtores/consumidores, com mecanismos de armazenamento, documentação, publicação, busca e recuperação. A ferramenta considera como componente, os artefatos produzidos no desenvolvimento (processo, modelos, manuais, código, binário, testes etc.) e, assim, permite a aquisição de diferentes conjuntos de artefatos vinculados a licenças personalizadas e configuráveis;
- Biblioteca Hunter - proposta pelos autores Wang *et al.* (2016), esta ferramenta tem como objetivo a reutilização do código fonte da linguagem JAVA, além disso, a biblioteca conta com a funcionalidade do alinhamento de interface e síntese do código reutilizável. Como mecanismo de busca do código fonte utilizaram um hibridismo de métodos, com banco de dados com aproximadamente 12 milhões de métodos Java coletados de repositórios de código aberto, como Github e Bitbucket;
- Ferramenta *CodeEase* – proposta pelos autores Abid *et al.* (2017), a ferramenta tem como principal objetivo a reutilização de código fonte, e foi desenvolvida como um *plugin* para o Eclipse. O mecanismo de busca está associado ao ambiente de desenvolvimento "Eclipse", assim quando um usuário escreve ou edita linhas de código em um método, o *CodeEase* fornece recomendações de conclusão para o mesmo, baseado na reutilização de código. Essa busca é realizada através da consulta em repositórios de código aberto, e um repositório local, o qual foi pré-populado com uma grande gama de códigos.

Outros trabalhos, na literatura mostram a importância da reutilização de componentes de *software* de requisitos, um exemplo é o trabalho dos autores Haddad e Ordóñez (2007), onde bibliotecas de reutilização estão associadas ao gerenciamento, armazenamento e recuperação de artefatos. E afirmam que é necessária uma escalabilidade na biblioteca, de acordo com o aumento da complexidade da manutenção e do gerenciamento dessa. Dessa maneira, os autores propõem em seu trabalho uma ferramenta que promove e facilita o projeto e o desenvolvimento de componentes de *software* reutilizáveis específicos do domínio.

3 METODOLOGIA DE PESQUISA

A metodologia de pesquisa utilizada neste trabalho foi dividida em quatro etapas, iniciando com a pesquisa bibliográfica (Etapa 1), na qual foi realizada uma pesquisa nas bases científicas Scopus e ScienceDirect, relacionada com aspectos fundamentais de bibliotecas de reuso, gerenciamento de projetos, engenharia de requisitos e seus respectivos tesouros, de modo a conferir a estrutura bibliográfica para fundamentar a investigação científica. Em seguida, foi elaborado o escopo do projeto (Etapa 2) através da criação da EAP para o projeto "Biblioteca de Reuso de Requisitos de *Software* para Diferentes Domínios", executando, portanto, o processo "Criar EAP" da área de conhecimento "Gerenciamento de Escopo". Através da EAP foi elaborada a decomposição hierárquica do escopo necessário para propor a estruturação do projeto com o objetivo de obter uma visão estruturada do que deve ser concretizado, além de alinhar o entendimento do projeto. A EAP elaborada foi classificada como orientada a processos, ou seja, organizada no nível mais alto (macroprocessos) por fases ou etapas de um processo e cada fase foi dividida em pacotes de trabalho. Para elaboração das fases (macroprocessos) e pacotes de trabalho que compõem a EAP foram realizadas validações através de entrevistas com pesquisadores sêniores na área de computação com *expertise* em reuso e requisitos de *software*.

Dando continuidade à metodologia, foi construído o escopo do produto (Etapa 3) através da elaboração da lista de requisitos necessários para implementação do projeto "Biblioteca de Reuso de Requisitos de *Software* para Diferentes Domínios", executando, portanto, o processo "Coletar os Requisitos" da área de conhecimento "Gerenciamento de Escopo". Para elaboração da lista de requisitos foram realizadas entrevistas com os pesquisadores sêniores na área de computação que participaram da Etapa 2. E para finalizar o processo metodológico, realizou-se a validação do escopo do produto, ou seja, da lista de requisitos (Etapa 4), na qual foi elaborada uma pesquisa com profissionais da área de TI através de um questionário (Quadro 1) com o objetivo de validar importância do produto resultante da lista de requisitos, biblioteca de reuso, para a área de TI. As questões 1 e 2 são relacionadas ao processo de desenvolvimento de *software*, mais especificamente sobre o ciclo de vida e as questões de 3 a 6 são relacionadas à utilização da biblioteca.

Quadro 1 – Questionário para validação do escopo do produto

Questões	Opções de Respostas
Q1: "Qual etapa do ciclo de vida do <i>software</i> demanda mais tempo?"	Elicitação (Levantamento e Validação) de Requisitos; Documentação; Desenvolvimento; Testes e Implantação.
Q2: "Na etapa de Elicitação de Requisitos, se a mesma for feita de forma inadequada, quanto isso prejudica o andamento do projeto?"	1 "Não Prejudica"; 2 "Prejudica Pouco"; 3 "Prejudica Razoavelmente"; 4 "Prejudica"; 5 "Prejudica Muito" e 6 "Prejudica Extremamente".
Q3: "A biblioteca agilizará o processo de desenvolvimento de <i>Software</i> ?"	1 "Não Agiliza"; 2 "Pouco"; 3 "Razoavelmente"; 4 "Agiliza"; 5 "Agiliza Muito" e 6 "Agiliza Extremamente".
Q4: "Você utilizaria essa biblioteca?"	Sim; Sim, apenas gratuita; Não.
Q5: "Além da Elicitação de Requisitos, em qual outra etapa do ciclo de vida a biblioteca seria benéfica?"	Projeto e Documentação; Desenvolvimento; Testes e Implantação; Em nenhuma etapa.
Q6: "Possui conhecimento de outra biblioteca que auxilie na Elicitação de Requisitos?"	Sim ou Não? Se sim, qual o nome da mesma."

Fonte: Autoria própria (2019).

As etapas do ciclo de vida utilizadas no Quadro 1 estão mais diretamente relacionadas ao ciclo de vida clássico de *software* (cascata), mas isso não é impedimento para que processos de desenvolvimento que utilizem outros formatos de ciclos de vida (RUP, Prototipação, Espiral, e outros) (Pressman, 2011) possam

participar da validação do escopo do produto. Nestes casos é necessário fazer um paralelo entre as etapas do ciclo de vida utilizadas no Quadro 1 e as etapas do ciclo de vida utilizado.

4 RESULTADOS E DISCUSSÃO

Os resultados gerados pelo presente trabalho podem ser divididos em três partes, a primeira consiste na elaboração do escopo do projeto “Biblioteca de Reuso de Requisitos de *Software* para Diferentes Domínios”, através da criação da EAP que descreve as fases que devem ser seguidas, a fim de executar o projeto proposto de uma maneira eficaz e com menor chance de desperdício de tempo e trabalho. A segunda parte apresenta à elaboração do escopo do produto através da lista de requisitos desejáveis para a biblioteca. E a terceira parte aborda a validação da lista de requisitos através de um questionário aplicado aos profissionais da área de Tecnologia da Informação (TI).

4.1 Escopo do Projeto: EAP da Biblioteca de Reuso de Requisitos de *Software*

O projeto da Biblioteca de Reuso de Requisitos de *Software* para Diferentes Domínios tem como objetivo disponibilizar um *framework on-line* capaz de proporcionar aos usuários o acesso aos requisitos de *software* por domínio desejado. A base de dados será alimentada pelo administrador da biblioteca, pela busca automática e mineração de texto na *web*, e pelos usuários que desejarem compartilhar conhecimento e contribuir para comunidade de desenvolvimento de *software*.

A implementação da estrutura referente ao projeto Biblioteca de Reuso de Requisitos de *Software* para Diferentes Domínios foi feita a partir de um estudo bibliográfico sobre o processo de elicitação de requisitos e bibliotecas de reuso. A partir disso, foi determinado uma estrutura analítica desse processo, que pode ser utilizada para construção da Biblioteca de Reuso e no delineamento de compreensão desse processo. A Figura 2 apresenta a primeira versão da EAP desenvolvida neste trabalho e utilizou a representação por forma gráfica para facilitar o entendimento e visualização inicial, já que a mesma não possui alto grau de complexidade e tamanho. A EAP encontra-se decomposta em dois níveis hierárquicos, sendo composta de quatro macroprocessos que contêm as fases do projeto “Biblioteca de Reuso de Requisitos” e estes são divididos em pacotes de trabalho. Os macroprocessos são:

- Pesquisa Bibliográficas: composta por três pacotes de trabalho (Engenharia de Requisitos, Bibliotecas de Reuso e Ferramentas de Gerenciamento de Requisitos);
- Análise: composta por dois pacotes de trabalho (Modelo de descrição de requisitos funcionais e não funcionais e Métricas de qualidade de requisitos);
- Estruturação: composta por três pacotes de trabalho (Definição do modelo de requisitos funcionais e não funcionais, Estruturação da biblioteca e Protótipo);
- Validação: composta por dois pacotes de trabalho (Estudo de caso e Testes).

Para validação da EAP (Figura 2) foram realizadas entrevistas com dois pesquisadores sêniores na área de computação com expertise em reuso e requisitos de *software*, ambos com o seguinte perfil: mais de 20 anos de experiência acadêmica e profissional em Engenharia de *Software*, principalmente nas áreas de Requisitos, Gestão de Projetos e Reuso de *Software*; e com título de doutor na área de computação a mais de 10 anos. A partir da validação da primeira versão da EAP vários detalhamentos foram necessários para a correta compreensão das fases do projeto e serão descritos a seguir.

Figura 2 – Primeira Versão da EAP da Biblioteca de Reuso de Requisitos.



Fonte: Autoria própria (2019).

O primeiro macroprocesso consiste em Pesquisas Bibliográficas acerca da Engenharia de Requisitos, Bibliotecas de Reuso e Ferramentas de Gerência de Requisitos. Para tanto, bases acadêmicas conhecidas como Scopus e Science Direct, devem ser consultadas, de modo a conferir a estrutura bibliográfica e fundamentar a investigação científica.

No segundo macroprocesso, “Análise” é proposto a realização de uma pesquisa acerca dos modelos existentes de descrição de requisitos funcionais e não funcionais, buscando características em ambas as metodologias de desenvolvimento de *software* (tradicional e ágil) e, posteriormente, a verificação dos modelos que mais se adequam a Biblioteca de Reuso.

Além disso, ainda em relação ao macroprocesso “Análise”, o mesmo também consiste na exploração de mecanismos de aferição para a qualidade dos Requisitos, que estarão disponíveis na Biblioteca. Pesquisas como dos autores Haleem e Beg (2015); Bigio, Motta e Srteit (2017); e Hovorushchenko (2018) sugerem métodos para utilização de métricas no que diz respeito a qualidade dos requisitos: *Requirements Manager* (REM), Cobertura de rastreabilidade requisito-fonte (CRRF); além de outros mecanismos, os quais pode-se citar: avaliação por caso de uso e padrões, revisão por pares, avaliação por *feedback*, entre outros.

No macroprocesso “Estruturação” é abordado um estudo com o objetivo de encontrar um padrão para definir um requisito dentro da Biblioteca de Reuso. Assim, a inserção funcionará de duas formas diferentes, de acordo com a característica do requisito (funcional ou não-funcional). No que se refere a requisitos funcionais, a Biblioteca deverá disponibilizar diferentes modelos de descrição de acordo com o método de desenvolvimento (Ágil ou Tradicional). Já no que tange aos requisitos não funcionais, não há variação de acordo com a metodologia do desenvolvimento, mas sim de acordo com as características do próprio requisito. Assim, serão definidos *layouts*, funcionalidades, modo de interação com os usuários, entre outros aspectos voltados à usabilidade, o que será delineado de forma visível através do pacote de trabalho que consiste na implementação de um protótipo, que irá cobrir determinados modelos de descrição de requisitos, um mecanismo inicial de avaliação da qualidade, além de um mecanismo de busca dos requisitos cadastrados.

No último macroprocesso é realizada a validação, a qual segue duas abordagens: a primeira consiste em estudos de caso em setores de uma instituição que apresentam requisitos definidos para demanda de *software*, e no setor de TI dessas instituições para validar os mecanismos de aferição da qualidade e de busca na biblioteca; e, com base nesses estudos de caso, poderão ser realizados testes no protótipo da biblioteca, a fim de melhorar sua estrutura.

A partir do detalhamento das fases foi possível elaborar a versão final da EAP, a qual foi decomposta em três níveis hierárquicos para melhor compreensão do projeto e capacidade para gerenciar o mesmo. A

versão final da EAP está representada na forma de lista numerada para facilitar a visualização, devido ao considerável aumento do número de pacotes de trabalho em relação à primeira versão, conforme mostra o Quadro 2.

Quadro 2 – Versão Final da EAP da Biblioteca de Reuso

1. Elaborar as Pesquisas Bibliográficas
 - 1.1. Elaborar pesquisas bibliográficas em Engenharia de Requisitos
 - 1.1.1. Consultar as bases acadêmicas e científicas
 - 1.1.2. Elaborar o texto
 - 1.2. Elaborar pesquisas bibliográficas em Bibliotecas de Reuso
 - 1.2.1. Consultar as bases acadêmicas e científicas
 - 1.2.2. Elaborar o texto
 - 1.3. Elaborar pesquisas bibliográficas em Ferramentas de Gerência de Requisitos
 - 1.3.1. Consultar as bases acadêmicas e científicas
 - 1.3.2. Elaborar do texto
 - 2.1. Pesquisar modelos existentes de descrição de requisitos funcionais
 - 2.1.1. Pesquisar modelos nas metodologias tradicionais de desenvolvimento
 - 2.1.2. Pesquisar modelos nas metodologias ágeis de desenvolvimento de *software*
 - 2.1.3. Analisar os modelos que mais se adequam ao projeto da Biblioteca de Reuso
 - 2.2. Pesquisar modelos existentes de descrição de requisitos não funcionais
 - 2.2.1. Pesquisar modelos nas metodologias tradicionais de desenvolvimento
 - 2.2.2. Pesquisar modelos nas metodologias ágeis de desenvolvimento de *software*
 - 2.1.3. Analisar os modelos que mais se adequam ao projeto da Biblioteca de Reuso
 - 2.3. Elaborar as métricas de qualidade de requisitos
 - 2.3.1. Analisar os mecanismos de aferição para a qualidade dos requisitos
 - 2.3.2. Propor métricas de qualidade de requisitos para implementar na biblioteca
 - 2.4. Definir a lista de requisitos funcionais e não funcionais
 3. Elaborar a Estruturação da Biblioteca de Reuso
 - 3.1. Elaborar estudos com o objetivo de definir um padrão de requisito funcional
 - 3.1.1. Definir um padrão de requisito funcional, usando metodologias tradicionais
 - 3.1.2. Definir um padrão de requisito funcional, usando metodologias ágeis
 - 3.2. Elaborar estudos com o objetivo de definir um padrão de requisito não funcional
 - 3.3. Elaborar o protótipo da Biblioteca
 - 3.3.1. Definir o *layout* da Biblioteca
 - 3.3.2. Definir os padrões de usabilidade da Biblioteca
 - 3.3.3. Implementar a Biblioteca
 4. Realizar a validação
 - 4.1. Realizar o estudo de caso
 - 4.1.1. Definir as instituições e setores
 - 4.1.2. Executar e documentar o estudo de caso
 - 4.2. Realizar testes nos protótipos
 - 4.2.1. Executar os testes de acordo com os estudos de caso
 - 4.2.2. Documentar os testes

Fonte: Autoria Própria (2019).

4.2 Escopo do Produto: Lista de Requisitos da Biblioteca de Reuso de Requisitos de Software

Foi elaborado o escopo do produto, Requisitos da Biblioteca de Reuso de Requisitos de *Software*, através da lista de requisitos funcionais para a implementação da Biblioteca, conforme o Quadro 3.

Quadro 3 – Lista de Requisitos Funcionais da Biblioteca de Reuso de Requisitos

REQ01 - Cadastro de perfis de acesso
REQ02 - Cadastro de tipos de usuários (individual, comunidade, organização etc.)
REQ03 - Cadastro de usuários
REQ04 - Cadastro de domínios
REQ05 - Cadastro de padrões da metodologia tradicional para descrição de requisitos
REQ06 - Cadastro de padrões das metodologias ágeis para descrição de requisitos
REQ07 - Cadastro (incluir, alterar e excluir requisitos) da descrição de requisitos funcionais por domínio utilizando padrões da metodologia tradicional
REQ08 - Cadastro (incluir, alterar e excluir requisitos) da descrição de requisitos funcionais por domínio utilizando padrões das metodologias ágeis
REQ09 - Cadastro de descrição dos requisitos não funcionais por domínio
REQ10 – Validação do cadastro (incluir, alterar e excluir requisitos) de requisitos funcionais e não funcionais
REQ11 - Cadastro das métricas de qualidade de requisitos
REQ12 - Avaliação da qualidade da descrição dos requisitos
REQ13 - Busca automática através da mineração de dados na <i>web</i> de requisitos funcionais por domínio utilizando padrões metodologia tradicional
REQ14 - Busca automática através da mineração de dados na <i>web</i> de requisitos funcionais por domínio utilizando padrões das metodologias ágeis
REQ15 - Consulta de requisitos funcionais e não funcionais por domínio e por metodologia
REQ16 - <i>Upload</i> de requisitos funcionais e não funcionais por domínio e por metodologia
REQ17 – Gerar os indicadores para apoio a tomada de decisão: Quantidade de usuários que executaram <i>upload</i> de requisitos por domínio; Quantidade de domínios com requisitos disponíveis na biblioteca; Quantidade de usuários que colaboraram com a inclusão de requisitos por domínio; Quantidade de atualizações de requisitos por domínio.

Fonte: Autoria Própria (2019).

4.3 Validação da Lista de Requisitos da Biblioteca de Reuso de Requisitos de Software

Para validação do escopo do produto, Biblioteca de Reuso de Requisitos de *Software*, foi aplicado o questionário apresentado no Quadro 1 aos profissionais que trabalham na área de TI, mais especificamente com desenvolvimento de *software*, nas seguintes instituições do estado do Rio de Janeiro: Algorich – Desenvolvimento de *Software*, localizada na cidade de Campos dos Goytacazes; DNA Sistemas, localizada na cidade de Santo Antônio de Pádua; e núcleos de desenvolvimento de *software* do Instituto Federal Fluminense (IFFluminense) dos campi Itaperuna, Campos Guarus e Campos Centro.

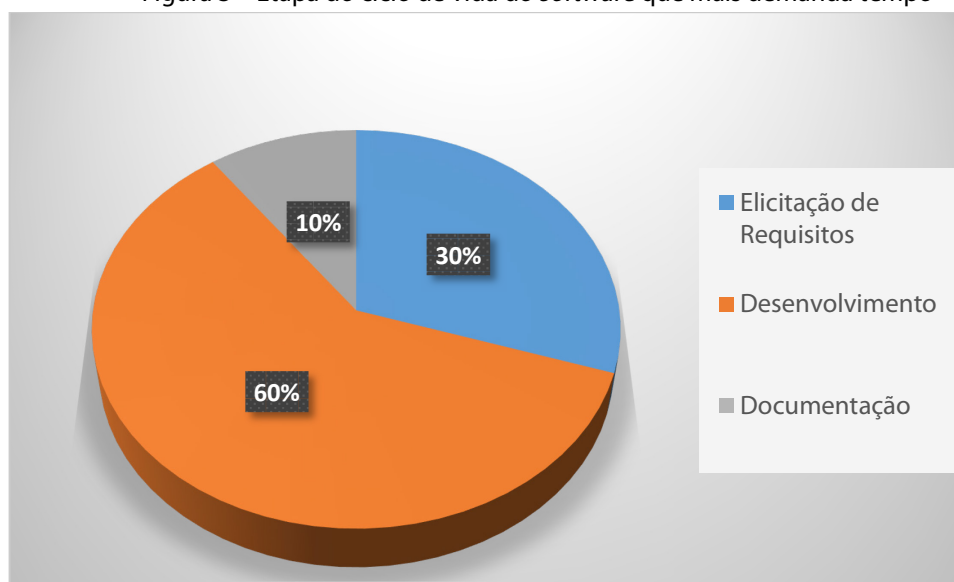
O questionário foi respondido por uma amostra de dez profissionais da área de TI, todos com mais de cinco anos de experiência na área. De acordo com a função exercida pelo profissional, a amostra contém: cinco Analistas de Sistemas; quatro Desenvolvedores de *Software* e um Coordenador de Redes. A lista de requisitos e a proposta do projeto de “Biblioteca de Reuso de Requisitos” foi apresentada a toda amostra por meio de uma reunião presencial e o questionário (Quadro 1) disponibilizado de forma *on-line* para acesso da amostra, onde a mesma respondeu as questões após o término da reunião.

A partir das respostas obtidas nos questionários, pode-se concluir que, em relação as etapas do ciclo de vida do *software*, conforme mostra a Figura 3, 60% da amostra considerou a etapa de Desenvolvimento como aquela que mais demanda tempo durante o ciclo de vida do *software*; 30% considerou a etapa de Elicitação de Requisitos e 10% considerou a etapa de Documentação. E conclui-se também que 90% da

amostra afirmou que se a etapa de Elicitação de Requisitos for realizada de forma inadequada no processo de desenvolvimento de um *software*, o andamento do projeto é extremamente prejudicado.

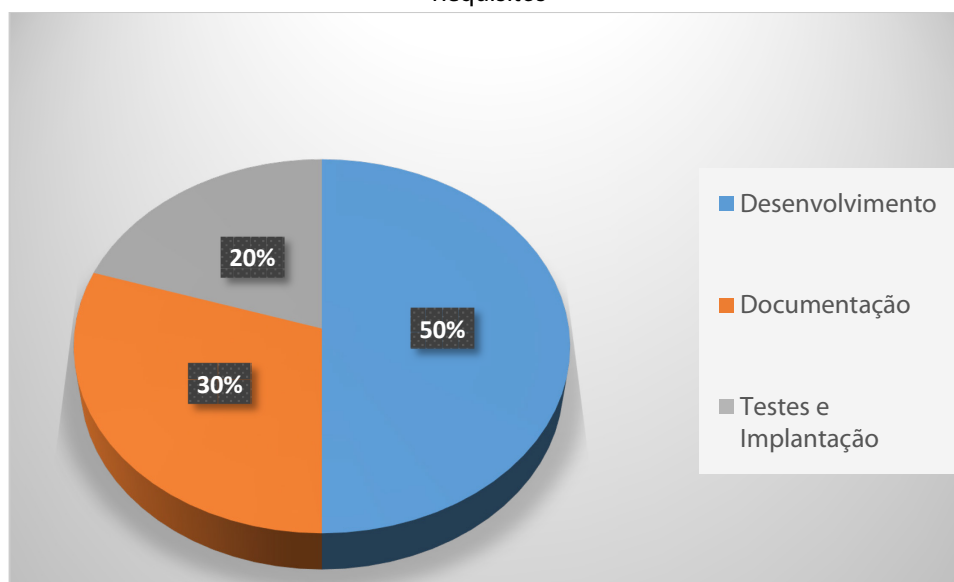
Ainda a partir das respostas obtidas nos questionários, mas em relação aos resultados referentes à estruturação da Biblioteca de Reuso de Requisitos de *Software* pode-se chegar aos seguintes resultados: 80% da amostra concordou que à utilização da biblioteca proposta agilizará o processo de desenvolvimento de *software*; toda a amostra optou por utilizar a biblioteca proposta, porém 70% utilizaria apenas se fosse gratuita; observa-se na Figura 4 que a amostra considerou que a biblioteca proposta pode contribuir, não só para a fase de Elicitação de Requisitos, mas também, para a fase de Desenvolvimento (50%), Documentação (30%), Testes e Implantação (20%); e que a amostra desconhece e/ou não utiliza bibliotecas para auxiliar à Elicitação de Requisitos no início de um projeto.

Figura 3 – Etapa do ciclo de vida do *software* que mais demanda tempo



Fonte: Autoria Própria (2019).

Figura 4 – Outra (s) etapa (s) do ciclo de vida onde a biblioteca seria benéfica, além da Elicitação de Requisitos



Fonte: Autoria Própria (2019).

5 CONCLUSÃO

Os requisitos são, cada vez mais, elementos essenciais à construção de bons produtos de *software*. A partir disso, uma boa e completa elicitação de requisitos, no início do processo de desenvolvimento, vem a ser um ganho no que diz respeito ao tempo e custos. A possibilidade de a equipe de desenvolvimento de *software* ter acesso ao reuso, através de uma biblioteca *on-line*, de uma boa descrição de requisitos para o domínio que deseja implementar pode ser considerada um benefício para todo o processo de desenvolvimento de *software* devido à esta etapa ser custosa e complexa.

A estrutura analítica de um projeto é um fator fundamental para uma boa hierarquização e um melhor entendimento do projeto, auxiliando de forma clara e concisa ao cumprimento dos objetivos.

Essa pesquisa atingiu seus objetivos quanto à elaboração do escopo do projeto e do produto para a estruturação da Biblioteca de Reuso de Requisitos de *Software* para Diferentes Domínios, sendo validada a sua importância através de um questionário aplicado aos profissionais da área de Tecnologia da Informação, que em sua maioria foram favoráveis a biblioteca proposta e que serão potenciais usuários da Biblioteca de Reuso de Requisitos após a sua implementação.

Espera-se que à execução deste do projeto possibilite a melhoria da qualidade de desenvolvimento de *software*, no que se refere a custo e tempo do processo de elicitação de requisitos. Além disso, foi corroborado que o Reuso de Requisitos é uma área da Engenharia de Requisitos na qual ainda há necessidade de pesquisa e de contribuição no que se refere ao processo de desenvolvimento de *software*. Como proposta para trabalhos futuros, sugere-se implementar a Biblioteca de Reuso de Requisitos e validar o uso da mesma em uma amostra de profissionais experientes na área de desenvolvimento de *software*.

REFERÊNCIAS

- ABID, S.; JAVED, S.; NASEEM, M.; SHAHID, S.; BASIT, H. A.; HIGO, Y. Codeease: harnessing method clone structures for reuse. //: INTERNATIONAL WORKSHOP ON SOFTWARE CLONES (IWSC), 3., Klagenfurt. **Proceedings of the 11th IWSC**. Klagenfurt, Austria: IEEE, 2017. p. 24-30. Disponível em: <http://ieeexplore.ieee.org/document/7880505/>. Acesso em: 11 mar. 2018.
- AZEVEDO JUNIOR, D. P. de; CAMPOS, R. de. Definição de requisitos de software baseada numa arquitetura de modelagem de negócios. **Production**, v. 18, n. 1, p. 26–46, 2008.
- BIGIO, M. T.; MOTTA, R. W.; SRTREIT, R. E. Quality metrics in requirements collection. **Revista Gestão do Conhecimento e Tecnologia da Informação**, v. 1, p. 88–97, 2017.
- CHANG, C-H. *et al.* XML-Based Reusable Component Repository for Embedded Software. //: ANNUAL COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE WORKSHOPS (COMPSACW), 35., Munich. **Proceedings of the IEEE 35th COMPSACW**. Munich, Germany: IEEE, 2011. p. 345-350. Disponível em: <http://ieeexplore.ieee.org/document/6032262/>. Acesso em: 19 fev. 2018.
- NAVES, T. F.; FERREIRA, H. N. Reuso de software: suas vantagens, técnicas e práticas. //: ENCONTRO ANUAL DE COMPUTAÇÃO (IX ENACOMP), 4., 2011, Catalão. **Anais [...]**. Catalão-GO: ENACOMP, 2011.
- FRAKES, W. B.; KYO K. Software reuse research: status and future. **IEEE Transactions on Software Engineering**, v. 31, n. 7, p. 529–536, 2005.
- GOLDIN, L.; BERRY, D. M. Reuse of requirements reduced time to market at one industrial shop: a case study. **Requirements Engineering**, v. 20, n. 1, p. 23–44, 2015.
- HADDAD, H. M.; ORDONEZ, M. J. Enhanced Component Reuse with Atomic Domains: Application Scenarios. //: INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY (ITNG'07), 4., 2007, Las Vegas. **Proceedings of the ITNG'07**. Las Vegas – USA: IEEE, 2007. p. 597-602. Disponível em: <http://ieeexplore.ieee.org/document/4151748/>. Acesso em: 30 jan. 2018.

HALEEM, M.; BEG, M. R. Impact analysis of requirement metrics in software development environment. *In*: IEEE INTERNATIONAL CONFERENCE ON ELECTRICAL, COMPUTER AND COMMUNICATION TECHNOLOGIES, 2015, Coimbatore. **Proceedings of the ICECT 2015**. Coimbatore, India: IEEE, 2015. p. 1-6. Disponível em: <http://ieeexplore.ieee.org/document/7226072/>. Acesso em: 10 mar. 2018.

HAUKSDÓTTIR, D.; MORTENSEN, N. H.; NIELSEN, P. E. Identified adjustability dimensions when generating a product specific requirements specification by requirements reuse. **Computers in Industry**, v. 65, n. 6, p. 952–966, 2014.

HOVORUSHCHENKO, T. Information Technology for Assurance of Veracity of Quality Information in the Software Requirements Specification. *In*: SHAKHOVSKA, N.; STEPASHKO, V. (ed.). **Advances in Intelligent Systems and Computing II - Selected Papers from the International Conference on Computer Science and Information Technologies (CSIT 2017)**. Lviv – Ukraine, Cham: Springer International Publishing, 2018. p. 166–185.

ISO/IEC/IEEE 24765. **Systems and software engineering — Vocabulary**. Geneva: International Organization for Standardization (ISO), 2017.

KAIYA, H.; SHIMIZU, Y.; YASUI, H.; KAJIRI, K.; SAEKI, M. Enhancing Domain Knowledge for Requirements Elicitation with Web Mining. *In*: ASIA PACIFIC SOFTWARE ENGINEERING CONFERENCE, 2010, Sydney. **Proceedings of the APSEC (2010)**. Sydney, NSW: IEEE, 2010. p. 3-12. Disponível em: <http://ieeexplore.ieee.org/document/5693175/>. Acesso em: 1 fev. 2018.

MARQUIONI, C. E. Escopo de Projeto X Escopo de Produto: A Engenharia de Requisitos como subsídio para a Gestão de Software. *In*: SIMPÓSIO INTERNACIONAL DE ADMINISTRAÇÃO E MARKETING/V CONGRESSO DE ADMINISTRAÇÃO DA ESPM, 3., 2008, São Paulo. **Anais [...]**. São Paulo: ESPM, 2008.

MARTINS, G. H.; MELO, M. Gestão Antecipada e Gestão de Projeto: Um estudo de caso na Indústria de Embalagens no Brasil. **Journal of Lean Systems**, v. 1, n. 3, p. 30–52, 2016.

PACHECO, C. *et al.* Reusing functional software requirements in small-sized software enterprises: a model oriented to the catalog of requirements. **Requirements Engineering**, v. 22, n. 2, p. 275–287, 2017.

PALOMARES, C.; FRANCH, X.; QUER, C. Requirements Reuse and Patterns: A Survey. *In*: SALINESI, C.; WEERD, I. VAN DE (ed.). **Requirements engineering: foundation for software quality**. Germany: Cham- Springer International Publishing, 2014. p. 301–308.

PALOMARES, C.; QUER, C.; FRANCH, X. Requirements reuse and requirement patterns: a state of the practice survey. **Empirical Software Engineering**, v. 22, n. 6, p. 2719–2762, 2017.

PMI. **PMBOK**: um guia do conhecimento em gerenciamento de projetos. 5. ed. Pennsylvania: Project Management Inst, 2013.

PMI. Success rates rise: transforming the high cost of low performance. *In*: 9TH GLOBAL PROJECT MANAGEMENT SURVEY, 2017, USA. **PMI's Pulse of the Profession**. USA: PMI, 2017. p. 1-32 Disponível em: <https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2017.pdf>. Acesso em: 12 fev. 2018.

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

SANTOS, R. P. DOS; TOSTES, L. R.; WERNER, C. M. L. A Brechó-EcoSys extension to support negotiation in the software ecosystems context. *In*: INTERNATIONAL CONFERENCE ON INFORMATION REUSE & INTEGRATION (IRI), 14., 2013, San Francisco. **Proceedings of the IEEE 14TH IRI 2013**. São Francisco, USA: IEEE, 2015. Disponível em: <http://ieeexplore.ieee.org/document/6642521/>. Acesso em: 19 fev. 2018.

SOMMERVILLE, Ian. **Engenharia de software**. 9. ed. São Paulo: Person Addison-Wesley, 2011.

SRINIVAS, C.; RADHAKRISHNA, V.; RAO, C. V. G. Clustering Software Project Components for Strategic Decisions and Building Reuse Libraries. //: INTERNATIONAL CONFERENCE ON ENGINEERING & MIS 2015, 2015, Istanbul. **Proceedings of the ICEMIS '15**. Istanbul, Turkey: ACM Press, 2015. Article No. 62. Disponível em: <http://dl.acm.org/citation.cfm?doid=2832987.2833075>. Acesso em: 30 jan. 2018.

VIJAY RAO, D.; SARMA, V. V. S. A Computational Intelligence Approach to Software Component Repository Management. //: BALAS, V. E.; KOPRINKOVA-HRISTOVA, P.; JAIN, L. C. (ed.). **Innovations in Intelligent Machines**. Berlin: Springer Berlin Heidelberg, 2014. p. 109–132.

WANG, Y.; FENG, Y.; MARTINS, R.; KAUSHIK, A.; DILLIG, I.; REISS, S. P. Hunter: next-generation code reuse for Java. //: ACM SIGSOFT INTERNATIONAL SYMPOSIUM ON FOUNDATIONS OF SOFTWARE ENGINEERING, 24., 2016, Seattle, WA. **Proceedings of the FSE 2016**. Seattle, WA: ACM Press, 2016. p. 1028-1032. Disponível em: <http://dl.acm.org/citation.cfm?doid=2950290.2983934>. Acesso em: 10 mar. 2018.