PUC Minas
A B A K Ó S
Instituto de Ciências Exatas e Informática

Licença Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported

Uma Proposta de Ontologia para Gerência de Requisitos: Relacionamentos de Dependência entre as Práticas do CMMI-DEV Nível 2 e MR-MPS-SW Nível G*

A Proposal of Requirements Management Ontology: Dependency Relationships among Practices included at CMMI-DEV Level 2 and MR-MPS-SW Level G

Ewelton Yoshio Yoshidome[1]
Sandro Ronaldo Bezerra Oliveira[2]

Resumo

Este artigo apresenta uma proposta de ontologia que estabelece os relacionamentos de dependência entre as práticas específicas da área de processo Gerência de Requisitos do modelo *Capability Maturity Model Integration for Development* e os relacionamentos de dependências entre os resultados esperados do processo de Gerência de Requisitos do Modelo de Referência de Melhoria do Processo de *Software* para *Software*. O objetivo desta pesquisa consiste em prover um mecanismo para apoiar implementações e avaliações oficiais a partir do uso desses modelos. Desse modo, esta pesquisa busca auxiliar consultores durante a implementação do processo de gerência de requisitos em uma organização de *software*. Para que a ontologia esteja aderente aos programas de melhoria, foi realizada uma pesquisa de campo em três empresas desenvolvedoras de software para coletar dados para auxiliar a fase de validação desta pesquisa.

Palavras-chave: Ontologia. Qualidade de Processo de *Software*. Relacionamento. Melhoria de Processo.

*Submetido em 26/10/2013 – Aceito em 25/02/2015

[1]Mestre em Ciência da Computação pelo Programa de Pós-graduação em Ciência da Computação da Universidade Federal do Pará (UFPA), Brasil– ewelton.yoshio@gmail.com

[2]Professor pelo Programa de Pós-Graduação em Ciência da Computação da Universidade Federal do Pará (UFPA), Brasil– srbo@ufpa.br

Abstract

This paper presents an ontology that establishes the dependency relationships between the specific practices of the Requirements Management process area included in Capability Maturity Model Integration and the dependency relationships between the expected results of the Requirements Management process included in Reference Model of Software Process Improvement for Software. The goal of this research is to provide a mechanism to assist the official implementations or evaluations in the use of these models. Thus, this research will support the consultants during the implementation of requirements management process in software development companies. For the ontology to adhere to software improvement programs, we conducted a survey in three software development companies to collect data to support the validation phase of this research.

Keywords: Ontology. Software Process Quality. Relationship. Process Improvement.

1 INTRODUÇÃO

Empresas desenvolvedoras de *software* que desejam implementar programas de melhoria em seus processos organizacionais podem se deparar com o problema de conciliar a execução das atividades dos projetos em desenvolvimento com as constantes melhorias que estão sendo implementadas em seu ambiente. Somando-se a isto, existe o fato de que o tempo de implementação de programas de melhoria geralmente é extenso (MORGADO et al., 2007).

Existe no mercado um número considerável de empresas desenvolvedoras de *software* que buscam por avaliação de melhoria do processo em modelos como Melhoria do Processo de *Software* Brasileiro (MPS.BR), e *Capability Maturity Model Integration for Development* (CMMI-DEV). Para que essas empresas consigam estas avaliações, é necessário que seja satisfeito determinado conjunto de requisitos estabelecidos pelo modelo. Segundo Soydan e Kokar (2006), por esses modelos agregarem um grande número de conceitos em sua estrutura, empresas que desejam receber avaliações de melhoria, geralmente, contratam especialistas (consultores em implementação). Mesmo para esses especialistas, o processo de verificar todos os relacionamentos entre os componentes do modelo não é uma tarefa trivial. Esse problema se complica com o fato das constantes atualizações que os modelos de qualidade sofrem ao longo do tempo, com adição de novas práticas e/ou processos em sua estrutura (SHARIFLOO et al., 2008).

Segundo Huang e Zhang (2010), existem poucos recursos humanos que possuem conhecimento sobre qualidade de *software* nas empresas de desenvolvimento. Além disso, mesmo possuindo conhecimento sobre melhoria de processo de *software*, essas pessoas possuem uma carência no entendimento correto de certos conceitos sobre o assunto.

Em Schots et al. (2011), são descritas experiências de implementações de melhoria do processo em várias empresas desenvolvedoras de *software*. Os autores relatam a necessidade de se apresentar e ensinar conceitos de engenharia de *software* para auxiliar na compreensão das atividades do processo Para apoiar no ensino e no estabelecimento do entendimento desses conceitos, a definição de uma ontologia pode ser utilizada (DUARTE; FALBO, 2000).

A pesquisa descrita em Leal et al. (2012) apresenta a necessidade de adotar um mecanismo para a seleção e implementação gradual de boas práticas à micro e às pequenas empresas. Para isto, a abordagem utilizada foi a definição de uma ontologia contendo as referidas práticas. Entretanto, essa ontologia não considera todas as práticas descritas em programas de melhoria.

Para esta pesquisa, foi desenvolvida uma ontologia que estabelece os relacionamentos de dependência entre as práticas existentes no Modelo de Referência do MPS para *Software* (MR-MPS-SW). Esses relacionamentos estão compatíveis com o modelo CMMI-DEV. Um relacionamento de dependência existe quando um ativo de modelo (resultado esperado/prática específica) depende de informações ou serviços providos por outro ativo de modelo para ser iniciado. Por exemplo: o método de estimativa de esforço do projeto possui dependência com

os dados/valores de tamanho do projeto.

A modelagem da ontologia servirá como um mecanismo de apoio durante uma implementação ou avaliação de programas de melhoria, auxiliando no levantamento de possíveis produtos de trabalho que podem ser utilizados em um processo de desenvolvimento de *software*.

Os relacionamentos de dependência entre as práticas presentes nos modelos foram estabelecidos, pois em documentações oficiais da Associação para a Promoção da Excelência do *Software* Brasileiro (SOFTEX), (SOFTEX, 2011), e do *Software Engineering Institute* (SEI), (SEI – SOFTWARE ENGINEERING INSTITUTE, 2010), é citada a existência das dependências entre os resultados esperados/práticas específicas (um resultado observável do sucesso do alcance do propósito do processo (ISO/IEC – INTERNATIONAL ORGANIZATION FOR STANDARDIZATION/ THE INTERNATIONAL ELECTROTECHNICAL COMISSION., 2004). No entanto, esses relacionamentos não são formalmente evidenciados, contendo apenas indicações de processos/áreas de processo que estão relacionados. A compatibilidade entre esses dois modelos pode ser garantida a partir de Mello (2011) e SOFTEX (2012a). Deve-se salientar que o modelo de referência MR-MPS-SW está compatível com o CMMI-DEV, ISO/IEC 12207 e ISO/IEC 15504 (SOFTEX, 2012b).

Existe um grande número de ferramentas para modelagem e disponibilização de ontologias que facilitam a disseminação para a comunidade interessada. Exemplos de ferramentas são (ALMEIDA, 2006): Protéjé, WebODE, Ontolingua Server, OilEd, WebOnto, entre outras. Muitas dessas ferramentas possuem a característica de terem o custo acessível e não há a necessidade de contratar consultores para seu uso (SOYDAN; KOKAR, 2006). Desse modo, essas ferramentas poderão ser utilizadas em futuros trabalhos desta pesquisa.

O escopo da ontologia desta pesquisa está restrito ao processo de Gerência de Requisitos (GRE) do MR-MPS-SW e à área de processo Gerência de Requisitos do CMMI- DEV. Entretanto, esta pesquisa está inserida no contexto de uma Dissertação de Mestrado na qual abrange todo o Nível G do MPS.BR e parte do nível 2 do CMMI.

A escolha do nível G do MPS.BR (compatível com as áreas de processo Planejamento do Projeto, Monitoramento e Controle do Projeto, Gerência de Requisitos do CMMI) consiste no fato de ser o primeiro nível de maturidade e empresas brasileiras desenvolvedoras de *software* que pretendem implementar melhoria em seu processo, ainda estão na fase de adaptação e aprendizagem.

Este artigo está dividido nas seguintes seções: a Seção 2 descreve alguns dos trabalhos relacionados a esta pesquisa; a Seção 3 apresenta, brevemente, os objetivos e características do contexto desta pesquisa; a Seção 4 descreve a metodologia utilizada para este trabalho; a Seção 5 apresenta a ontologia de gerência de requisitos esta pesquisa; a Seção 6 descreve as instanciações realizadas para apoiar na avaliação desta pesquisa; e, por fim, a Seção 7 apresenta as conclusões deste trabalho.

2 TRABALHOS RELACIONADOS

Como um trabalho relacionado, pode-se citar a pesquisa de Soydan e Kokar (2006), na qual é realizada uma definição de uma ontologia para representar o modelo CMMI-DEV. Porém, essa pesquisa define as regras que regem a estrutura do modelo, não buscando verificar o relacionamento das práticas específicas entre as áreas de processo. Além disso, a ontologia está baseada na versão 1.1 do CMMI, a qual não é compatível com a atual versão do MPS.BR (SOFTEX, 2012b).

De forma similar, existe a pesquisa de Sharifloo et al. (2008), na qual é definida uma ontologia para o modelo *Capability Maturity Model Integration for Acquisition* (CMMI- ACQ). Contudo, o domínio de conhecimento definido em sua pesquisa abrange apenas a área de processo de aquisição.

Neto (2008) propõe um modelo de referência para que as próprias empresas realizem uma avalição de seu processo de desenvolvimento de *software* para analisarem a sua aderência ao modelo MR-MPS. Entretanto, a solução proposta não se preocupa em definir um modo de disponibilizar as informações (técnicas, artefatos, procedimentos, entre outros) de como cada resultado esperado está sendo contemplado.

Duarte e Falbo (2000) propõem uma ontologia para qualidade de *software* voltada para o ensino sobre conceitos no domínio de qualidade de *software*, mas sua pesquisa não está focada em definir as principais práticas de qualidade de *software* e como essas práticas estão relacionadas entre si.

Na tese de doutorado de Almeida (2006), é definida uma ontologia para auxiliar no aprendizado e disseminação do conhecimento organizacional. Porém, o escopo de seu trabalho não possui aderência às normas e modelos de qualidade de *software*.

Em Guizzardi, Falbo e Guizzardi (2008) é descrita uma proposta de modelagem de ontologia em processo de *software* por meio de uma ontologia de fundamentação. Entretanto esse trabalho não possui foco em definir a ontologia considerando práticas sugeridas em modelos de qualidade de *software*.

O trabalho proposto em Leal et al. (2012) descreve uma ontologia baseada em boas práticas de processo de *software* em micro e em pequenas empresas. Entretanto, em sua pesquisa, o processo de gerência de requisitos não é detalhado como definido na literatura especializada, como em Pressman (2006), SOFTEX (2012b), SEI – Software Engineering Institute (2010). Além disso, não existe uma relação explícita entre o processo de gerência de requisitos com as práticas relacionadas à gestão de mudança.

Por fim, tem-se a tese de doutorado de Falbo (1998), que define uma ontologia que estabelece os relacionamentos entre os ativos de um processo de *software*, com o objetivo de integrar o conhecimento em um ambiente de desenvolvimento. Porém, tal pesquisa não se preocupa em

alinhar com os conceitos constantes nos modelos de qualidade de *software* existentes.

3 O PROCESSO DE GERÊNCIA DE REQUISITOS

O objetivo do processo de GRE é realizar o gerenciamento dos requisitos do projeto e verificar suas inconsistências com os demais produtos de trabalho (SOFTEX, 2011; SEI – SOFTWARE ENGINEERING INSTITUTE, 2010).

A área de processo gerência de requisitos está presente no nível de maturidade 2 do CMMI-DEV. No contexto do MPS.BR, o processo de gerência de requisitos está presente no nível de maturidade G do MR-MPS-SW.

No CMMI-DEV, a área de processo de gerência de requisitos está dividida em cinco práticas específicas *(specific practices – SP)*. O MR-MPS-SW possui um conjunto de cinco resultados esperados (RE). Uma SP "é um componente esperado do modelo que é considerado importante para satisfazer à meta específica associada" (SEI – SOFTWARE ENGINEERING INSTITUTE, 2010).

Os programas de melhoria sugerem cinco práticas necessárias para contemplar o processo/área de processo de GRE, as quais são descritas como SP (CMMI-DEV) e RE (MR-MPS-SW). Essas práticas são:

- Trabalhar com os provedores de requisitos para obter um melhor entendimento do significado dos requisitos (SP 1.1 e GRE1);
- Obter comprometimento dos participantes do projeto com os requisitos (SP 1.2 e GRE2);
- Gerenciar mudanças nos requisitos à medida que evoluem durante o projeto (SP 1.3 e GRE5);
- Manter a rastreabilidade bidirecional dos requisitos e produtos de trabalho (SP 1.4 e GRE3);
- Identificar inconsistências entre os planos de projeto, produtos de trabalho e requisitos (SP 1.5 e GRE4).

4 METODOLOGIA DA PESQUISA

A construção da ontologia de relacionamentos entre as práticas do MPS.BR baseou-se no método sugerido por Falbo (1998). A elaboração da ontologia iniciou com a especificação dos requisitos, sendo identificadas as questões que a ontologia deve responder.

Em seguida, foram realizadas pesquisas sobre conceitos relacionados à melhoria do processo de *software*. Neste momento, também foi definida uma meta-ontologia modelada em linguagem *Unified Modeling Language* (UML). Meta-ontologia é a ontologia da representação de termos que serão utilizados para representar a ontologia principal. Os principais termos definidos em uma meta-ontologia são entidades, relacionamentos, atores, papéis (USCHOLD; GRUNINGER, 1996). A partir da modelagem, foram definidos axiomas por meio de lógica de primeira ordem. O uso de axiomas proporciona uma definição matemática das regras do universo em discurso, desse modo restringindo o comportamento e os relacionamento dos conceitos envolvidos no universo de discurso.

Paralelamente à etapa de definição dos conceitos, foi realizada uma pesquisa de campo buscando coletar informações dos processos de três empresas desenvolvedoras de *software* (Empresa-A, Empresa-B e Empresa-C), localizadas na cidade de Belém do Pará, que receberam avaliações oficiais do MPS.BR nos níveis G e F. O objetivo dessa pesquisa foi identificar os nomes dos principais produtos de trabalho utilizados para contemplar as práticas do MPS.BR. Como perfil resumido dessas empresas tem-se:

- A Empresa-A é uma empresa que oferece serviços de desenvolvimento de *software*, que possui como nicho de mercado a comunidade acadêmica e o serviço público. Seu corpo de desenvolvimento é composto por 5 (cinco) integrantes;
- A Empresa-B é uma empresa desenvolvedora de *software* baseada em soluções de *software* livre. A sua carteira de clientes é composta por empresas de vários segmentos do mercado, inclusive órgãos públicos. Atualmente, essa empresa é focada no desenvolvimento de sistemas e páginas *web*. Atualmente, sua equipe de desenvolvimento é composta por 10 (dez) pessoas;
- A Empresa-C é uma empresa focada em desenvolvimento multiplataforma. Suas soluções priorizam tecnologias baseadas em padrões abertos e *software* livre. Sua equipe de desenvolvimento é composta por aproximadamente 60 (sessenta) profissionais.

Na terceira etapa, foi realizada uma revisão por pares por meio de um especialista em Engenharia e Qualidade de *Software*. Esse especialista possui ampla experiência na implementação e avaliação dos modelos MR-MPS-SW e CMMI-DEV, e é instrutor oficial da SOFTEX para o modelo MR-MPS-SW. Durante essa etapa foi apresentada a meta-ontologia e explicado cada um dos conceitos e relacionamentos presentes na modelagem. Ao final dessa etapa, foram sugeridas melhorias na modelagem, as quais foram contempladas.

A quarta etapa consistiu na realização de uma revisão com um especialista no contexto de ontologias, buscando avaliar a consistência sintática e semântica das formalizações definidas na ontologia, a partir da realização de provas matemáticas sobre os axiomas gerados para a ontologia. O especialista possui ampla experiência em assuntos relacionados à lógica de pri-

meira ordem. Além disso, o especialista atua na área de inteligência computacional aplicada e tecnologias de educação à distância.

Por fim, a última etapa está relacionada à instanciação da ontologia desta pesquisa, como meio de avaliação dos conceitos e relacionamentos definidos na ontologia. Neste momento, foram utilizadas as informações coletadas durante a pesquisa de campo para realizar a instanciação da ontologia.

5 UMA ONTOLOGIA PARA GERÊNCIA DE REQUISITOS

Como foi mencionado, o domínio de interesse desta pesquisa consiste em definir os relacionamentos de dependência entre as práticas específicas do processo/área de processo de GRE. As evidências destas práticas são produzidas pelas empresas desenvolvedoras de *software* a partir da institucionalização das boas práticas de gerência de requisitos. Por este motivo, definiu-se que o universo de discurso da ontologia refere-se às práticas presentes no processo/área de processo.

Vale ressaltar que a meta-ontologia foi modelada utilizando a linguagem UML. Foi escolhida a linguagem UML para facilitar o entendimento dos conceitos e relacionamentos definidos nesta pesquisa, pois os especialistas da área de Engenharia e Qualidade de *Software* estão fortemente familiarizados com essa linguagem.

Salienta-se que existe uma ontologia referente às áreas de processo Planejamento do Projeto e Monitoramento e Controle do Projeto (processo de Gerência de Projeto no MPS.BR). O conteúdo referente à ontologia de gerência de projetos pode ser encontrado em *http://www.cin.ufpe.br/~srbo/PPGCC_Dissertacao_EweltonYoshidome.docx*.

A elaboração das questões de competência para esta pesquisa é baseada no modelo descrito por Falbo (1998). Essas questões são:

1. Como são identificados os requisitos?
2. Como são garantidos os entendimentos dos requisitos?
3. Como a equipe técnica compromete-se com os requisitos?
4. Como é realizada a rastreabilidade dos requisitos?
5. Como são realizadas as revisões de inconsistências dos requisitos?
6. Como são tratadas as mudanças de requisitos?

Baseando-se nas questões de competência, podem-se obter aspectos relacionados à (ao):

- Entendimento dos Requisitos junto aos Fornecedores de Requisitos (Questão 1 e 2);
- Comprometimento dos Requisitos pela Equipe Técnica (Questão 3);
- Definição da Rastreabilidade Bidirecional (Questão 4);
- Revisão de Inconsistências (Questão 5);
- Acompanhamento de Mudanças (Questão 6).

Frisa-se que as instâncias das classes para esta ontologia são representadas por produtos de trabalho institucionalizados no processo da organização. Como normalmente um único produto de trabalho é capaz de contemplar várias práticas esperadas pelos modelos ao mesmo tempo, decidiu-se definir os predicados em lógica de primeira ordem de cada conceito da ontologia com três parâmetros. O primeiro parâmetro representa o produto de trabalho, o qual contempla a prática. O segundo parâmetro representa a informação/conteúdo/seção do produto de trabalho, o qual contempla a prática esperada. Por fim, o terceiro parâmetro representa a combinação dos dois parâmetros anteriores, que é utilizado para realizar os relacionamentos entre os conceitos da ontologia.

Para exemplificar a utilização da estrutura dos predicados citados anteriormente, será utilizado o documento Plano do Projeto para contemplar a prática referente à definição dos requisitos de projeto. Nesta situação, o primeiro parâmetro do predicado é preenchido pelo "Plano do Projeto". Como, nesse caso, o documento de Plano do Projeto pode conter outras informações, tais como, escopo, estimativas, cronograma, entre outros, um segundo parâmetro estabelecendo a localização da descrição dos requisitos deve ser informado ao predicado, como por exemplo, a seção referente aos requisitos. Por fim, pelo fato do documento Plano do Projeto envolver várias práticas esperadas em programas de melhoria, deve-se definir o terceiro parâmetro. Esse parâmetro tem como objetivo prover a unicidade da prática de definição dos requisitos.

5.1 Entendimento dos Requisitos junto aos Fornecedores de Requisitos

Inicialmente, estabelece-se o escopo do projeto, junto aos clientes. A definição do escopo não é influenciada por nenhum conceito na ontologia. Entretanto, a partir do escopo, podem-se definir muitos elementos referentes ao projeto, tais como, conhecer a sua natureza, os seus requisitos, dados para realizar a sua estimativa, as habilidades necessárias para participar do projeto, entre outros.

A partir do escopo do projeto, pode-se definir um conjunto de requisitos. Na ontologia, para representar esse comportamento, estabeleceu-se que a classe "Escopo" influencia na classe

"RequisitoCliente". A classe "RequisitoCliente" é o resultado do levantamento e da consolidação das necessidades, expectativas, das restrições e das interfaces entre as partes interessadas, e que esteja aceitável ao cliente. Esse requisito é conhecido como requisito de cliente (SEI – SOFTWARE ENGINEERING INSTITUTE, 2010). A Figura 1 ilustra o relacionamento entre escopo e requisito de cliente.

Figura 1 – Definição dos requisitos de cliente e garantia de entendimento dos requisitos

Fonte: Dados da Pesquisa

Vale ressaltar a existência de empresas que definem o escopo e os requisitos de cliente como o mesmo produto de trabalho. Porém, decidiu-se separar estes conceitos, pois a ontologia pretende apresentar as práticas recomendadas em programas de melhoria e suas evidências, além de tornar a ontologia mais genérica possível.

Outra característica que deve ser ressaltada na relação entre as classes "Escopo" e "RequisitoCliente" é a possibilidade do requisito do cliente ser capaz de alterar o escopo do projeto. Isso ocorre devido ao fato de existirem mudanças nos requisitos ao longo do projeto que não estavam previstas no escopo inicial do projeto.

Uma prática esperada em gerência de requisitos está relacionada a um meio de garantir que os requisitos estejam entendidos por todos os envolvidos. O objetivo dessa prática consiste no estabelecimento de um mecanismo que garanta o entendimento homogêneo dos requisitos por todos os envolvidos. Esse mecanismo pode ser obtido a partir de contratos, gravações, assinaturas em atas, entre outros. Para representar a garantia de entendimento dos requisitos, foi definida uma classe denominada de "GarantiaEntendimentoRequisitos". Essa classe está relacionada à classe "RequisitoCliente" por meio de uma relação "garante".

Desta forma, os predicados relacionados aos conceitos escopo, requisito de cliente e garantia de entendimento dos requisitos são, respectivamente, *escopo(e,x,comb-e)*, *(req,y,comb-req)* e *gEntendimentoReq(ger,z,comb-ger)*.

A partir dos relacionamentos apresentados na Figura 1, podem-se obter os seguintes axiomas:

$$(\forall\, e, comb-e, req, comb-req)(influencia(comb-e, comb-req) \rightarrow escopo(e, *, comb-e) \ \wedge \ reqCliente(req, *, comb-req)) \quad \text{(A1)}$$

$$(\forall\, e, comb-e, req, comb-req)(influencia(comb-req, comb-e) \rightarrow escopo(e, *, comb-e) \ \wedge \ reqCliente(req, *, comb-req)) \quad \text{(A2)}$$

$$(\forall\, e, comb-e, req, comb-req)(garante(comb-ger, comb-req) \rightarrow gEntendimentoReq(ger, *, comb-ger) \ \wedge \ reqCliente(req, *, comb-req)) \quad \text{(A3)}$$

Deve-se observar que o escopo do projeto influencia na definição dos requisitos de cliente, porém, para estabelecer esses requisitos, é necessária a consolidação do seu entendimento

por todos os envolvidos. Assim, os dois relacionamentos devem ocorrer, simultaneamente, para a definição dos requisitos de cliente. A partir disto, obteve-se o axioma A4, o qual estabelece que existirá um requisito de cliente se, e somente se, existir um escopo que influenciará a definição dos requisitos e um mecanismo que garanta o entendimento de todos os requisitos.

$$(\exists\, req, comb-req)(reqCliente(req, *, comb-req) \leftrightarrow$$
$$(\exists\, e, comb-e, ger, comb-ger)escopo(e, *, comb-e)$$
$$\wedge gEntendimentoReq(ger, *, comb-ger) \wedge influencia(comb-e, comb-req)$$
$$\wedge(garante(comb-ger, comb-req) \quad \text{(A4)}$$

5.2 Comprometimento dos Requisitos pela Equipe Técnica

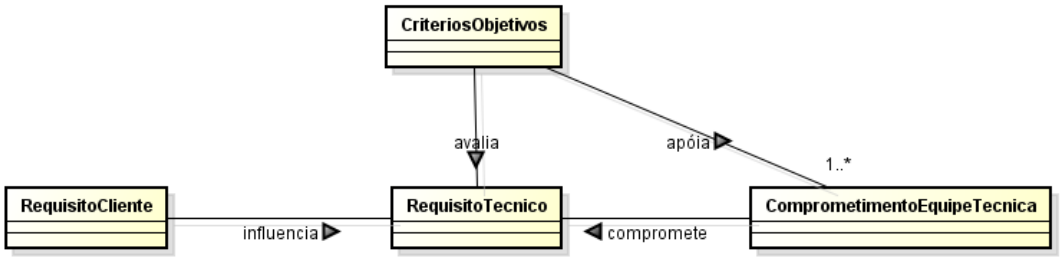
Além de garantir o entendimento dos requisitos entre os envolvidos do projeto, é necessário, também, definir mecanismos que garantam o comprometimento da equipe técnica em implementar os requisitos do projeto. Para isto, a equipe técnica realiza uma avaliação dos requisitos definidos a partir de um conjunto de critérios objetivos. Essa avaliação busca detectar inconsistências, ambiguidades, além de verificar a viabilidade de implementar os requisitos.

Normalmente, os requisitos analisados pela equipe técnica são refinados em uma linguagem mais funcional. Entretanto, existem empresas nas quais esse refinamento não é realizado, utilizando o próprio documento de escopo ou requisitos coletados diretamente do cliente.

O objetivo dessa prática é garantir que a equipe técnica busque por possíveis problemas durante a coleta de requisitos, verificando a existência de possíveis impedimentos para a sua implementação.

Como mencionado, os requisitos de cliente são refinados para um requisito em linguagem técnica. Para representar isto, definiu-se um relacionamento "influencia" entre as classes "RequisitoCliente" e "RequisitoTecnico", como apresentado na Figura 2.

Figura 2 – Refinamento dos Requisitos e Comprometimento da Equipe Técnica



Fonte: Dados da Pesquisa

Outra relação estabelecida para esta prática foi um relacionamento "compromete" entre as classes "ComprometimentoEquipeTecnica" e "RequisitoTecnico". Deve-se salientar que o comprometimento da equipe deve ser apoiado por um conjunto de critérios objetivos (classe "CriteriosObjetivos"). Esses critérios objetivos servem como um mecanismo de avaliação dos requisitos por parte da equipe técnica. Para isto, definiu-se o relacionamento "avalia" entre as

classes “CriteriosObjetivos” e “RequisitoTecnico”.

Assim, para esta prática, foram definidos os predicados *reqTecnico(reqt,x,comb-reqt), cTecnico(ct,y,comb-ct), cObjetivos(cobj,z,comb-cobj)*. Esses predicados representam, respectivamente, as classes “RequisitoTecnico”, “ComprometimentoEquipeTecnica” e “CriteriosObjetivos”.

Por meio da Figura 2, podem ser estabelecidos os axiomas epistemológicos B1, B2 e B3. Um axioma epistemológico descreve as restrições impostas pela forma de estruturação dos conceitos (FALBO, 1998).

$$(\forall\, req, reqt, comb-req, comb-reqt)(influencia(comb-req, comb-reqt) \rightarrow reqCliente(req, *, comb-req) \wedge reqTecnico(reqt, *, comb-reqt)) \quad \text{(B1)}$$

$$(\forall\, ct, reqt, comb-ct, comb-reqt)(compromete(comb-ct, comb-reqt) \rightarrow cTecnico(ct, *, comb-ct) \wedge reqTecnico(reqt, *, comb-reqt)) \quad \text{(B2)}$$

$$(\forall\, cobj, ct, comb-cobj, comb-ct)(apoia(comb-cobj, comb-ct) \rightarrow cObjetivos(cobj, *, comb-cobj) \wedge cTecnico(ct, *, comb-ct)) \quad \text{(B3)}$$

Além dos axiomas B1, B2 e B3, foi estabelecido outro axioma que consolidará os relacionamentos existentes entre as classes dessa prática (B4). Esse axioma define que, para realizar o comprometimento dos requisitos por parte da equipe técnica, é necessário o apoio de um conjunto de critérios objetivos. Vale ressaltar que a atividade relacionada ao comprometimento dos requisitos busca avaliar questões como ambiguidade, inconsistência, testabilidade, viabilidade, entre outros.

$$(\forall\, comb-ct, comb-reqt)(compromete(comb-ct, comb-reqt) \leftrightarrow (\exists\, cobj, comb-obj)(cObjetivos(cobj, *, comb-cobj) \wedge apoia(comb-cobj, comb-ct)) \quad \text{(B4)}$$

Outro axioma de consolidação, além de B4, refere-se à necessidade de estabelecer que o comprometimento da equipe com o requisito ocorre somente perante a sua aprovação por parte da equipe técnica, ou seja, o conjunto de critérios objetivos somente apoiará o comprometimento dos requisitos se, e somente se, esses critérios forem utilizados para avaliar os requisitos. Essa proposição é denotada com o axioma B5. Os axiomas de consolidação definem a coerência das informações de uma ontologia.

$$(\forall\, cobj, reqt, comb-obj, comb-ct, comb-reqt)(avalia(comb-cobj, comb-reqt) \leftrightarrow (apoia(comb-cobjt, comb-ct) \wedge\ compromete(comb-ct, comb-reqt) \quad \text{(B5)}$$

Salienta-se, também, que foi definido o predicado *avalia-condicao* para os axiomas B6 e B7. Esse predicado possui três parâmetros: *comb-obj*, que representa os critérios objetivos; *comb-reqt*, que representa os requisitos que estão sendo avaliados; e *comb-ct*, que registra se o requisito foi aprovado ou reprovado pela avaliação.

Os axiomas B6 e B7 têm a função de fortalecer a regra da necessidade dos requisitos

serem aprovados pelo conjunto de critérios objetivos.

$$(\forall\ cobj, reqt, comb-obj, comb-ct, comb-reqt)$$
$$(avalia-condicao(comb-cobj, comb-reqt, aprovado) \rightarrow$$
$$compromete(comb-ct, comb-reqt) \quad \text{(B6)}$$

$$(\forall\ cobj, reqt, comb-obj, comb-ct, comb-reqt)$$
$$(avalia-condicao(comb-cobj, comb-reqt, reprovado) \rightarrow$$
$$compromete(comb-ct, comb-reqt) \quad \text{(B7)}$$

5.3 Definição da Rastreabilidade Bidirecional

Os requisitos do projeto podem assumir diferentes abstrações ao longo do projeto. Assim, os requisitos podem estar descritos como necessidades, restrições, estórias. Além disso, muitos produtos de trabalho são derivados a partir dos requisitos, tais como: código-fonte, diagramas, casos de testes, entre outros (SOFTEX, 2011).

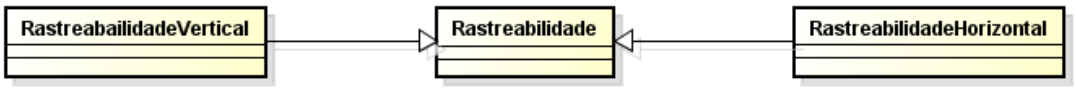
Baseado nesse cenário, uma das práticas presentes em modelos de qualidade é a institucionalização de um mecanismo para rastrear as dependências entre os produtos de trabalho.

A definição da rastreabilidade bidirecional permite rastrear as dependências entre os requisitos e os produtos de trabalho. A rastreabilidade apoia as avaliações de impacto das mudanças de requisitos que surgem ao longo do tempo.

Como a rastreabilidade bidirecional é uma prática constante em programas de melhoria, como CMMI e MPS.BR, foi definida uma classe denominada de “Rastreabilidade”. O conceito de rastreabilidade bidirecional abrange dois tipos de rastreabilidade: a rastreabilidade vertical e a rastreabilidade horizontal.

A rastreabilidade vertical é responsável por mapear as dependências entre produtos de trabalhos de granularidade distintos, por exemplo, o mapeamento entre os requisitos e os planos. A rastreabilidade horizontal é responsável por mapear as dependências entre produtos de trabalho de mesma granularidade, como por exemplo, a verificação de dependências entre os requisitos do projeto. Foram estabelecidas as classes “RastreabilidadeVertical” e “RastreabilidadeHorizontal” para representar, respectivamente, os conceitos de rastreabilidade vertical e rastreabilidade horizontal. A Figura 3 ilustra a estrutura de rastreabilidade bidirecional.

Figura 3 – Rastreabilidade Bidirecional



Fonte: Dados da Pesquisa

Assim, foi definido o predicado *rastreabilidade(ra,x,comb-ra)* para representar a rastreabilidade bidirecional. Para representar as rastreabilidades vertical e horizontal, foram definidos,

respectivamente, dois predicados *rVertical(ra,y,comb-ra)* e *rHorizontal(ra,z,comb-ra)*. Além disso, foram estabelecidos dois axiomas epistemológicos para representar o comportamento de herança do conceito de rastreabilidade bidirecional.

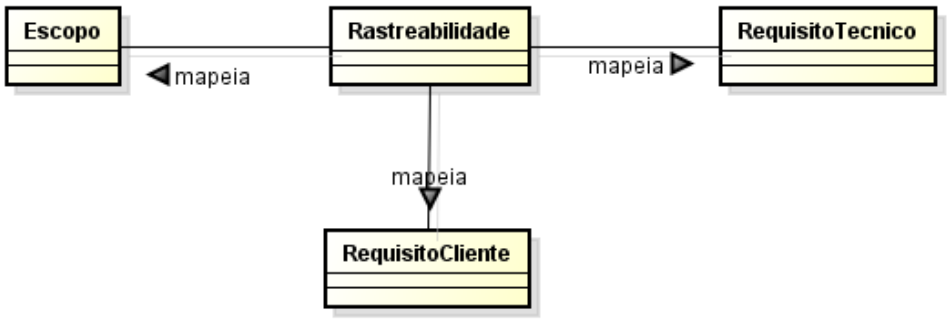
$$(\forall\, ra, comb-ra)(rVertical(ra,*,comb-ra) \rightarrow rastreabilidade(ra,*,comb-ra)) \quad \text{(C1)}$$

$$(\forall\, ra, comb-ra)(rHorizontal(ra,*,comb-ra) \rightarrow rastreabilidade(ra,*,comb-ra)) \quad \text{(C2)}$$

Como mencionado, a rastreabilidade busca estabelecer as dependências entre os produtos de trabalho de um projeto. Assim, para contemplar esse comportamento, definiu-se um relacionamento denominado de "mapeia". Este relacionamento liga a classe "Rastreabilidade" aos demais conceitos relacionados ao planejamento do projeto, tais como escopo e requisitos, além de conceitos que não fazem parte da gerência de requisitos, como planejamento de estimativas, custos, cronograma, recursos, entre outros.

Pelo fato do escopo desta pesquisa estar restrita ao processo/área de processo de GRE, os conceitos nas quais a classe "Rastreabilidade" estará relacionada são escopo, requisitos de cliente e requisitos técnicos. A Figura 4 ilustra esses relacionamentos.

Figura 4 – Relacionamento de Rastreabilidade com Escopo, Requisito de Cliente e Requisito Técnico



Fonte: Dados da Pesquisa

Os axiomas C3, C4 e C5 apresentam os relacionamentos presentes na Figura 4.

$$(\forall\, ra, e, comb-ra, comb-e)(mapeia(comb-ra, comb-e) \rightarrow (escopo(e,*,comb-e) \wedge rastreabilidade(ra,*,comb-ra)) \quad \text{(C3)}$$

$$(\forall\, ra, req, comb-ra, comb-req)(mapeia(comb-ra, comb-req) \rightarrow (reqCliente(req,*,comb-req) \wedge rastreabilidade(ra,*,comb-ra)) \quad \text{(C4)}$$

$$(\forall\, ra, reqt, comb-ra, comb-reqt)(mapeia(comb-ra, comb-e) \rightarrow (reqTecnico(reqt,*,comb-reqt) \wedge rastreabilidade(ra,*,comb-ra)) \quad \text{(C5)}$$

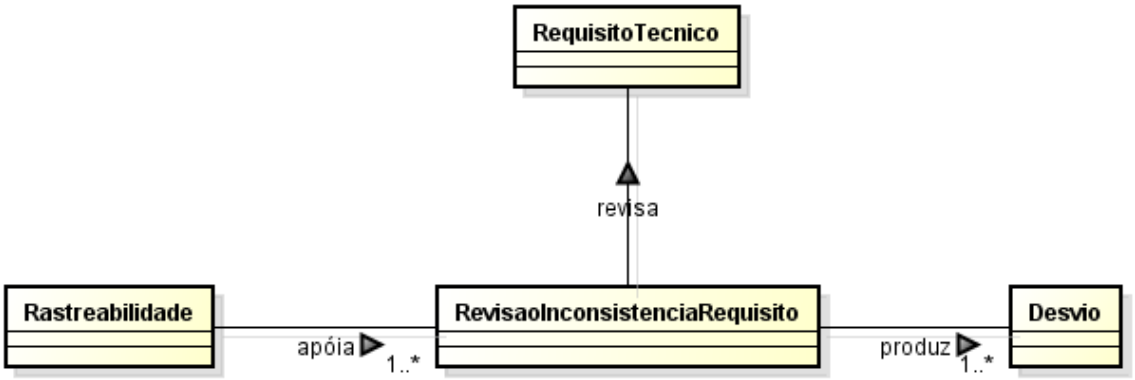
5.4 Revisão de Inconsistências

A revisão de inconsistência entre os requisitos e os produtos de trabalho do projeto é uma atividade realizada constantemente. Essa atividade pode ser realizada por meio de uma revisão ou mecanismo equivalente, buscando detectar problemas de inconsistência entre os requisitos e os demais produtos de trabalho, tais como planejamento do escopo, estimativas, cronograma, recursos, entre outros (SOFTEX, 2012b).

A importância da execução dessa atividade consiste no fato de que, ao longo do projeto, existem constantes mudanças nos requisitos. Por este motivo, é preciso examinar se os demais produtos de trabalho foram alinhados a essas mudanças. O mecanismo de rastreabilidade, citado na seção anterior, normalmente, é utilizado para auxiliar nestas revisões.

Para representar o conceito relacionado às revisões de inconsistências, estabeleceu-se uma classe denominada de "RevisaoInconsistenciaRequisito". Esse conceito relaciona-se à classe "RequisitoTecnico" por meio do relacionamento "revisa". Além disso, existe um relacionamento "apoia" entre as classes "Rastreabilidade" e "RevisaoIncosistenciaRequisito", denotando que o mecanismo de rastreabilidade auxiliará durante essas revisões. A Figura 5 ilustra esses relacionamentos.

Figura 5 – Revisão de Inconsistências e Registro de Desvios



Fonte: Dados da Pesquisa

Após a realização de uma revisão de inconsistências, um conjunto de inconsistências pode ser encontrado. Essas inconsistências, para esta ontologia, são definidas como desvios. Estes desvios devem ser registrados para evidenciar que a revisão foi realizada. Esse comportamento foi estabelecido por meio da relação "produz" entre as classes "RevisaoIncosistenciaRequisito" e "Desvio", conforme mostrado na Figura 5.

Pode-se notar, na Figura 5, que a cardinalidade entre "RevisaoIncosistenciaRequisito" e "Desvio" é do tipo *um para um ou mais* e não do tipo *um para muitos*. Modelou-se desse modo por três motivos: (1) durante uma avaliação oficial, é obrigatória a presença de desvios no projeto para evidenciar a realização das revisões de inconsistências; (2) a possibilidade de não serem geradas inconsistências é virtualmente impossível; e (3) segundo Falbo (1998), não

é possível elaborar um axioma a partir de um relacionamento do tipo *um para muitos*.

Desse modo, foram utilizados os predicados *rastreabilidade(ra,x,comb-ra), (reqt,y,comb-reqt), reqCliente(req,z,comb-req)* e *revisaoInconsistencia(rir,k,comb-rir)*, representando, respectivamente, a rastreabilidade bidirecional, os requisitos técnicos, os requisitos de cliente e a revisão de inconsistência dos requisitos. A seguir, são apresentados os axiomas que representam os relacionamentos ilustrados na Figura 5.

$$(\forall\, rir, reqt, comb-rir, comb-reqt)(revisa(comb-rir, comb-reqt) \rightarrow (revisaoInconsistencia(rir, *, comb-rir) \wedge reqTecnico(reqt, *, comb-reqt)) \quad \text{(D1)}$$

$$(\forall\, rir, ra, comb-rir, comb-ra)(apoia(comb-ra, comb-rir) \rightarrow (revisaoInconsistencia(rir, *, comb-rir) \wedge rastreabilidade(ra, *, comb-ra)) \quad \text{(D2)}$$

$$(\forall\, rir, d, comb-rir, comb-d)(produz(comb-rir, comb-d) \rightarrow (revisaoInconsistencia(rir, *, comb-rir) \wedge desvio(d, *, comb-d)) \quad \text{(D3)}$$

Um ponto importante a ser frisado nesta atividade consiste no fato de produção dos desvios, pois estes desvios serão apenas produzidos *se, e somente se, for realizada uma revisão de inconsistências, juntamente com o apoio do mecanismo de rastreabilidade.* O axioma D4 apresenta esta restrição.

$$(\exists\, d, comb-d)(\forall\, comb-rir, comb-ra)desvio(d, *, comb-d) \wedge (produz(comb-rir, comb-d) \rightarrow (revisao(comb-rir, comb-reqt) \wedge apoia(comb-ra, comb-rir)) \quad \text{(D4)}$$

5.5 Acompanhamento de Mudanças

Ao longo do projeto, os requisitos podem sofrer constantes mudanças por diferentes motivos, tais como alteração no escopo do projeto, adição de novas funcionalidades, ajustes provenientes de mudanças em outros requisitos, entre outros. Assim, uma das práticas sugeridas em programas de melhoria refere-se ao acompanhamento de todas estas mudanças desde sua solicitação até a sua conclusão.

O acompanhamento de mudanças de requisitos envolve atividades como registrar a solicitação de mudança, verificar o impacto gerado devido à mudança solicitada, buscar possíveis soluções para sua resolução, acompanhar e registrar todas as ações realizadas durante o ciclo de vida da solicitação de mudança.

No contexto desta pesquisa, o conceito "mudança" é um subtipo da classe "Desvio", definido com a classe "Mudanca". Essa classe representa as solicitações de mudanças realizadas ao longo do projeto. Além da classe "Mudanca", foi estabelecida a classe "Problema" como classe filha de "Desvio". A classe "Problema" representa todos os problemas, defeitos, inconformidades, entre outros.

Foi estabelecida uma diferenciação entre problema e mudança pelo fato de que mudanças nos requisitos podem acarretar em um grande impacto em todo o projeto, podendo acarretar em sua inviabilidade. Por este motivo, todo desvio referente à mudança deve estar associado a uma análise de impacto. Os problemas estão relacionados aos defeitos, falhas ou desvios de planejamento, que devem ser corrigidos para a correta continuidade do projeto.

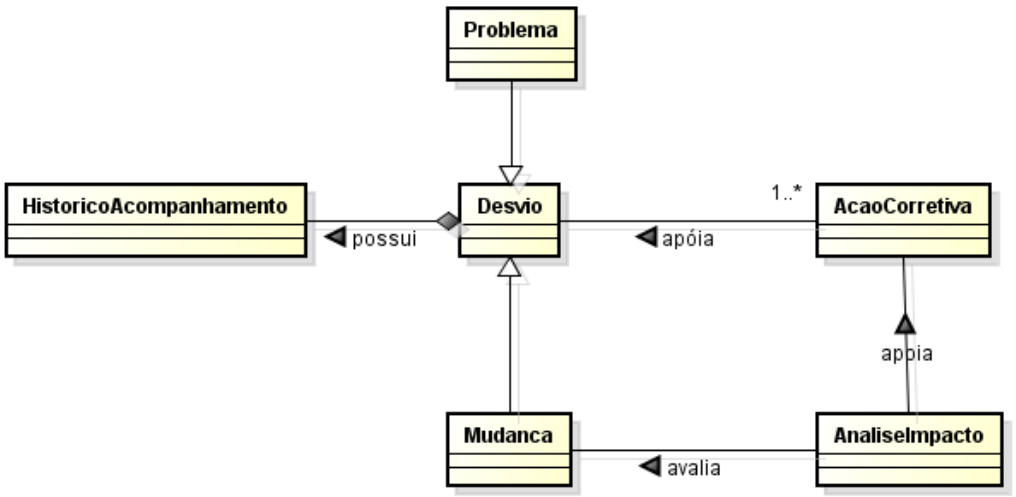
Quando um desvio é registrado, um conjunto de soluções deve ser proposto, com o objetivo de solucioná-lo. O conceito relacionado ao conjunto de solução para a resolução do desvio foi denominado de ação corretiva, modelado pela classe "AcaoCorretiva". Esta classe relaciona-se com a classe "Desvio" por meio da relação "apoia".

Como mencionado, toda solicitação de mudança está associada a uma análise de impacto. Por este motivo, modelou-se, na ontologia, uma classe denominada de "AnaliseImpacto". Essa classe tem a função de auxiliar na resolução de uma solicitação de mudança. Para isso, definiu-se o relacionamento "apoia" entre as classes "AnaliseImpacto" e "AcaoCorretiva". Esse relacionamento denota que os impactos da solicitação de mudança afetam as propostas de soluções do desvio (mudança). Além disso, foi definido o relacionamento "avalia" entre as classes "AnaliseImpacto" e "Mudanca". Esse relacionamento indica que a análise de impacto está associada apenas às solicitações de mudança.

Por fim, recomenda-se, também, que todas as ações tomadas sobre o desvio sejam registradas. Assim, faz-se necessário um histórico de acompanhamento de cada desvio. Para contemplar esta prática, modelou-se o relacionamento "possui" entre as classes "Desvio" e "HistoricoAcompanhamento". A Figura 6 ilustra as classes e os relacionamentos descritos até o momento.

Então, a partir das classes modeladas até o momento, podem-se estabelecer os seguintes predicados: *mudança(d,x,comb-d), problema(d,x,comb-d), acaoCorretiva(ac,y,comb-ac), hAcompanhamento(ha,z,comb-ha), analiseImpacto(ai,k,comb-ai)*. Esses predicados são referentes, respectivamente, aos conceitos mudança, problema, ação corretiva, histórico de acompanhamento e análise de impacto.

Figura 6 – Acompanhamento dos Desvios



Fonte: Dados da Pesquisa

Como as classes "Mudanca" e "Problema" são filhas de "Desvio", foram definidos os axiomas E1 e E2. Estes axiomas representam a estrutura de herança de classes. Além disso, foram elaborados os axiomas E3 e E4 para denotar o relacionamento entre análise de impacto e ação corretiva e o relacionamento entre análise de impacto e mudança.

$$(\forall\, d, comb-d)(mudanca(d,*,comb-d) \rightarrow desvio(d,*,comb-d)) \quad \text{(E1)}$$

$$(\forall\, d, comb-d)(problema(d,*,comb-d) \rightarrow desvio(d,*,comb-d)) \quad \text{(E2)}$$

$$(\forall\, ai, ac, comb-ai, comb-ac)(apoia(comb-ai, comb-ac) \rightarrow analiseImpacto(ai,*,comb-ai) \wedge acaoCorretiva(ac,*,comb-ac)) \quad \text{(E3)}$$

$$(\forall\, ai, d, comb-ai, comb-d)(avalia(comb-ai, comb-ac) \rightarrow analiseImpacto(ai,*,comb-ai) \wedge mudanca(d,*,comb-d)) \quad \text{(E4)}$$

Os axiomas referentes aos relacionamentos da classe "Desvio" foram definidos como E5 e E6.

$$(\forall\, ac, d, comb-ac, comb-d)(apoia(comb-ac, comb-d) \rightarrow acaoCorretiva(ac,*,comb-ac) \wedge desvio(d,*,comb-d)) \quad \text{(E5)}$$

$$(\forall\, ha, d, comb-ha, comb-d)(possui(comb-d, comb-ha) \rightarrow hAcompanhamento(ha,*,comb-ha) \wedge desvio(d,*,comb-d)) \quad \text{(E6)}$$

Deve-se salientar que todo desvio possui um histórico de acompanhamento, porém, esse histórico existirá apenas se os procedimentos de resolução do desvio foram executados. Assim, foi preciso definir o axioma de consolidação E7 para restringir essa necessidade.

$$(\exists\, ha, comb-ha)(\forall\, d, ac, comb-d, comb-ac)(possui(comb-d, comb-ha) \wedge hAcompanhamento(ha,*,comb-ha) \leftrightarrow (apoia(comb-ac, comb-d) \wedge problema(d,*,comb-d) \wedge acaoCorretiva(ac,*,comb-ac)) \quad \text{(E7)}$$

Como descrito anteriormente, um desvio do tipo mudança está associado a uma análise de impacto. Assim, foi necessário definir um axioma de consolidação (E8) para contemplar essa especificidade. Esse axioma descreve que *se uma análise de impacto avalia uma solicitação de mudança, então, essa análise de impacto deverá apoiar as propostas de solução dessa mudança.*

$$(\forall\, d, ac, ai, comb-d, comb-ac, comb-ai)(avalia(comb-ai, comb-d) \wedge mudanca(d,*,comb-d) \wedge analiseImpacto(ai,*,comb-ai) \rightarrow (apoia(comb-ai, comb-ac) \wedge acaoCorretiva(ac,*,comb-ac)) \quad \text{(E8)}$$

As Tabelas 1 e 2 contêm o dicionário de dados referentes às classes e aos relacionamentos, respectivamente, estabelecidos na ontologia desta pesquisa.

Tabela 1 – Dicionário de Dados das Classes

Classe	Significado
Escopo	Define as características e abrangência do projeto, tais como suas necessidades, expectativas e restrições.
RequisitoCliente	Define o resultado do levantamento e da consolidação das necessidades, expectativas, das restrições e das interfaces entre as partes interessadas e seja aceitável pelo cliente. Esses são os requisitos descritos em alto nível.
GarantiaEntendimentoRequisitos	É o mecanismo utilizado para garantir que todos os envolvidos possuem um entendimento comum de todos os requisitos. Isso pode ser implementado a partir de contratos, gravações, confirmações em fóruns, etc.
RequisitoTecnico	Propriedades de produtos e serviços a serem adquiridos ou implementados. São derivados dos requisitos de cliente.
ComprometimentoEquipeTecnica	Mecanismo utilizado para garantir que a equipe técnica compreendeu os requisitos e está comprometida em implementar estes requisitos. Isso pode ser implementado a partir de contratos, gravações, confirmações em fóruns, entre outros.
CriteriosObjetivos	Conjunto de questões respondidas de modo objetivo. Normalmente, utiliza-se o documento de *checklist* para evidenciar estes critérios.
Rastreabilidade	Mecanismo utilizado para realizar a associação entre duas ou mais lógicas, como requisitos, elementos de sistemas, verificações ou tarefas. Esse conceito é referente à rastreabilidade bidirecional.
RastreabilidadeVertical	Mecanismo utilizado para realizar a associação entre duas ou mais entidades de granularidade diferentes. Ex: rastreabilidade entre requisitos e plano do projeto.
RastreabilidadeHorizontal	Mecanismo utilizado para realizar a associação entre duas ou mais entidades de mesma granularidade. Ex: rastreabilidade entre requisitos.
RevisaoInconsistenciaRequisito	Atividade que evidencia a atividade de revisão, buscando encontrar inconsistências entre os requisitos e os demais produtos de trabalho do processo.
Desvio	Define um conjunto de desvios no planejamento do projeto. Esses desvios podem ser de natureza de problema, mudança, defeitos, inconformidades, entre outros.
Problema	Define um conjunto de problemas, defeitos e inconformidades registrados ao longo do projeto. Essa classe é subtipo de “Desvio”.
Mudanca	Define um conjunto de solicitações de mudanças nos requisitos do projeto. Diferentemente da classe “Problema”, uma mudança possui uma análise de impacto atrelado. Esta classe é subtipo de “Desvio”.
AnaliseImpacto	Recurso utilizado para apoiar as decisões de aprovação de uma solicitação de mudança.
HistoricoAcompanhamento	Descreve o registro de todas as ações tomadas durante o ciclo de vida de um desvio. O histórico de acompanhamento registra as ações tomadas, os agentes responsáveis pelas ações tomadas, a data/hora das ações, modificações realizadas etc.
AcaoCorretiva	Representa um conjunto de ações que podem ser tomadas para a resolução de determinado desvio.

Tabela 2 – Dicionário de Dados dos Relacionamentos

Relacionamento	Significado
influencia(a,b)	Indica que *a* é insumo para obter *b*.
revisa(a,b)	É utilizado quando *a* apresenta uma revisão sobre o conteúdo de *b*.
garante(a,b)	É utilizado quando um produto de trabalho *a* garante o entendimento de um conceito *b*.
compromete(a,b)	Define que um conceito *a* garante o comprometimento da execução de um conceito *b*.
apoia(a,b)	É utilizado quando um conceito *a* apoia no desenvolvimento/execução de *b*.
mapeia(a,b)	É utilizado quando *a* realiza o mapeamento de dependência do conceito *b*.
avalia(a,b)	É utilizado para indicar que o conceito *a* avalia o conteúdo *b*.
produz(a,b)	Indica que o produto de trabalho *a*, a partir de certas condições, produz o produto de trabalho *b*.
possui(a,b)	Indica que o produto de trabalho *b* está contido no produto de trabalho *a*.

6 AVALIAÇÃO DA ONTOLOGIA

Nesta seção são descritos os procedimentos para a realização da avaliação da ontologia de relacionamentos de dependência entre as práticas de Gerência de Requisitos do CMMI e MPS.BR.

6.1 Contexto da Avaliação

A avaliação da ontologia desta pesquisa é baseada nos procedimentos de instanciação de conceitos (classes). Uma instanciação de uma ontologia consiste de um número de declarações sobre objetos do universo de discurso, usando os conceitos e as relações definidos na ontologia (VALENTE, 1995).

Essas instanciações são realizadas sobre os axiomas definidos na ontologia. Assim, espera-se que as instâncias de cada conceito presentes na ontologia possam ser traduzidas como produtos de trabalho. Foi realizada uma instanciação do processo de cada empresa descrita na Seção IV e, em seguida, foi realizada uma avaliação da ontologia baseada nos resultados da instanciação.

Os objetos de análise desta avaliação são os processos das empresas desenvolvedoras de *software* que foram entrevistadas durante a pesquisa de campo. Deste modo, os produtos de trabalho coletados durante a entrevista são utilizados como instâncias da ontologia.

Não foram utilizadas ferramentas para a definição das ontologias. A verificação da

consistência dos axiomas foi realizada por meio de revisão por pares por um especialista na área de ontologias. O motivo foi o fato de que não foi possível encontrar uma ferramenta que satisfizesse as necessidades desta pesquisa. Entretanto, futuramente, existe um planejamento de que uma ferramenta seja utilizada para auxiliar na verificação das consistências dos axiomas definidos nesta pesquisa.

6.2 Execução da Instanciação

Uma vez que a ontologia e seus axiomas foram devidamente definidos, iniciou-se a instanciação dos predicados referentes aos conceitos da ontologia desta pesquisa.

Além das instanciações dos conceitos da ontologia, esta etapa abrangeu também a instanciação dos relacionamentos. Para isto, nas subseções seguintes são apresentadas estas instanciações.

6.2.1 Entendimento dos Requisitos junto aos Fornecedores de Requisitos

Para apresentar a instanciação relacionada ao entendimento dos requisitos, foram utilizados os dados coletados a partir do processo institucionalizado na Empresa-A. Em seu processo, o escopo do projeto é definido em um produto de trabalho denominado de "Proposta Técnica". Paralelamente à definição do escopo do projeto, a empresa também realiza a coleta dos requisitos. Esses requisitos são registrados em um produto de trabalho denominado de "Lista de Requisitos".

Ainda, durante a reunião de definição do escopo, é verificado se os requisitos coletados estão perfeitamente entendidos por todos. Assim, para garantir este entendimento, todos os envolvidos assinam seus nomes no documento.

escopo(Proposta Técnica, Escopo, ptEscopo)
influencia(ptEscopo, lrRequisitos)
reqCliente(Lista de Requisitos, Requisitos, lrRequisitos)
gEntendimentoReq(Lista de Requisitos, Assinaturas, lrAssinaturas)
garante(lrAssinaturas, lrRequisitos)

A Empresa-B utiliza o artefato denominado de "Termo de Abertura" para estabelecer o escopo do projeto. Mais tarde, o escopo deste documento é refinado para o artefato "Lista de Requisitos".

O produto de trabalho "Lista de Requisitos" contém o conjunto de todos os requisitos coletados durante a fase de concepção do escopo. O entendimento dos requisitos é garantido a partir de assinaturas dos envolvidos no documento "Lista de Requisitos". Ressalta-se que este

documento está fortemente dependente das especificações descritas no escopo do projeto.

escopo(Termo de Abertura, Escopo, taEscopo)
influencia(taEscopo, lrRequisitos)
reqCliente(Lista de Requisitos, Requisitos, lrRequisitos)
gEntendimentoReq(Lista de Requisitos, Assinaturas, lrAssinaturas)
garante(lrAssinaturas, lrRequisitos)

Na Empresa-C, o escopo do projeto é definido no artefato "Proposta Técnica". Essa empresa interpreta o escopo do projeto como um conjunto de requisitos coletados durante as reuniões com o cliente. Desse modo, tanto o escopo como os requisitos fazem parte do mesmo documento. Como nessa situação o escopo e os requisitos são entendidos como o mesmo conceito, pode-se entender que os requisitos são definidos primeiramente. Assim, nesse caso, os requisitos do cliente definem o escopo do projeto.

Em relação à garantia de entendimento dos requisitos, a Empresa-C utiliza um contrato associado à "Proposta Técnica". Nesse contrato, são coletadas as assinaturas de todos os envolvidos.

reqCliente(Proposta Técnica, Requisitos, ptRequisitos)
influencia(ptRequisitos ,taEscopo)
escopo(Proposta Técnica, Escopo, ptEscopo)
gEntendimentoReq(Contrato, Assinaturas, cAssinaturas)
garante(cAssinaturas, ptRequisitos)

Notou-se que, nas empresas Empresa-A e Empresa-B, os requisitos do cliente foram definidos após o estabelecimento do escopo do projeto. Assim, pode-se notar a veracidade do axioma A1.

A Empresa-C define a lista de requisitos e o escopo do projeto como uma mesma entidade. Entretanto, como estabelecido no axioma A2, que define que os requisitos influenciam na definição do escopo, este prevê essa possibilidade.

Em relação à garantia de entendimento dos requisitos, nas três empresas são utilizadas assinaturas para confirmar que os requisitos estão devidamente entendidos por todos os envolvidos.

6.2.2 Comprometimento dos Requisitos pela Equipe Técnica

O SP 1.2 e o GRE2 sugerem que os requisitos sejam avaliados pela equipe técnica, com base em critérios objetivos, para verificar possíveis problemas (semânticos, ambiguidades, limitações tecnológicas, entre outros) nos requisitos. Caso os requisitos sejam aceitos pela equipe, deverá ser estabelecido um mecanismo para garantir que a equipe técnica comprometeu-se a implementar esses requisitos.

Na Empresa-A, os requisitos técnicos elicitados são registrados no documento "Lista de Requisitos". Desse modo, nesta empresa, os requisitos coletados durante a fase de definição do escopo são previamente descritos em uma linguagem técnica, não havendo duas fases distintas para o processo de refinamento dos requisitos.

Os critérios objetivos para avaliação por parte da equipe técnica são presentes na própria "Lista de Requisitos". Assim, os resultados da avaliação objetiva são registrados no próprio documento. Além disso, caso os requisitos sejam aprovados, a equipe técnica compromete-se a implementar estes requisitos a partir da assinatura no próprio documento.

reqCliente(Lista de Requisitos, Requisitos, lrRequisitos)
influencia(lrRequisitos, lrRequisitos)
reqTecnico(Lista de Requisitos, Requisitos, lrRequisitos)
cTecncio(Lista de Requisitos, Assinaturas, lrAssinaturas)
cObjetivos(Lista de Requisitos, Checklist, lrChecklist)
apoia(lrChecklist, lrAssinaturas)
avalia(lrChecklist, lrRequisitos)
apoia(lrChecklist, lrAssinaturas)
avalia-condicao(lrChecklist, lrRequisitos, aprovado)
compromete(lrAssinaturas, lrRequisitos)

Na Empresa-B, os requisitos técnicos são registrados no artefato "Lista de Requisitos". De modo similar à Empresa-A, não existe uma etapa definida na qual os requisitos de cliente são refinados para requisitos técnicos. Nessa empresa, os requisitos são registrados em linguagem técnica durante a elaboração do escopo do projeto. Adicionalmente, nesse mesmo documento, está presente o conjunto de critérios objetivos que são utilizados para a equipe técnica realizar a avaliação dos requisitos.

Quando os requisitos são avaliados e aprovados, a equipe técnica compromete-se em implementar esses requisitos assinando um documento denominado de "Plano do Projeto". Esse produto de trabalho está associado ao artefato "Termo de Abertura", que descreve o escopo do projeto.

reqCliente(Lista de Requisitos, Requisitos, lrRequisitos)
influencia(lrRequisitos, lrRequisitos)
reqTecnico(Lista de Requisitos, Requisitos, lrRequisitos)
cTecncio(Plano do Projeto, Assinaturas, ppAssinaturas)
cObjetivos(Lista de Requisitos, Checklist, lrChecklist)
apoia(lrChecklist, ppAssinaturas)
avalia(lrChecklist, lrRequisitos)
apoia(lrChecklist, ppAssinaturas)
avalia-condicao(lrChecklist, lrRequisitos, aprovado)
compromete(ppAssinaturas, lrRequisitos)

No contexto da Empresa-C, os requisitos em linguagem técnica são os mesmos requisitos coletados durante a definição do escopo. Como mencionado anteriormente, essa empresa interpreta o escopo como os requisitos do projeto. Então, o escopo do projeto é estruturado

em forma de requisitos em uma linguagem técnica. Desse modo, os requisitos em linguagem técnica são registrados no documento "Proposta Técnica".

Em relação à avaliação dos requisitos, a equipe técnica da Empresa-C realiza uma reunião e, a partir de um *checklist* presente no documento "Laudo de Avaliação dos Casos de Uso", os requisitos são avaliados. Ao final da avaliação, a equipe técnica descreve o resultado em uma "Ata de Reunião", comprometendo-se com assinaturas, caso os requisitos sejam aprovados.

reqCliente(Proposta Técnica, Requisitos, ptRequisitos)
influencia(ptRequisitos, ptRequisitos)
reqTecnico(Proposta Técnica, Requisitos, ptRequisitos)
cTecncio(Ata de Reunião, Assinaturas, arAssinaturas)
cObjetivos(Lauda de Avaliação dos Casos de Uso, Checklist, lacuChecklist)
avalia(lacuChecklist, ptRequisitos)
apoia(lacuChecklist, arAssinaturas)
avalia-condicao(lacuChecklist, ptRequisitos, aprovado)
compromete(arAssinaturas, ptRequisitos)

No caso das três empresas, a avaliação dos requisitos por parte da equipe técnica é realizada com o apoio de um *checklist*. Esse *checklist* contém um conjunto de critérios que verificam a consistência dos requisitos de modo objetivo. Assim, pode-se perceber claramente que os axiomas B3 e B5 estão condizentes.

Baseado no resultado do *checklist*, a equipe técnica das três empresas comprometem- se ou não com a implementação dos requisitos. Portanto, como existe a necessidade do *checklist* para o respaldo do comprometimento, os axiomas B2 e B4 são válidos.

6.2.3 Definição da Rastreabilidade Bidirecional

A rastreabilidade bidirecional na Empresa-A é implementada a partir do uso da "Matriz de Rastreabilidade". Nesse produto de trabalho são adicionadas as relações entre todos os produtos de trabalho presentes no processo.

Deve-se notar que não foi realizada a rastreabilidade entre os requisitos de cliente e os requisitos técnicos, pois, no contexto da Empresa-A, esses dois requisitos são a mesma entidade.

rVertical(Matriz de Rastreabilidade, Lista de Requisitos x Proposta Técnica,mrReqxPt)
rHorizontal(Matriz de Rastreabilidade, Lista de Requisitos x Lista deRequisitos, mrReqxReq)
reqCliente(Lista de Requisitos, Requisitos, lrRequisitos)
escopo(Proposta Técnica, Escopo, ptEscopo)
mapeia(mrReqxPt, lrRequisitos)
mapeia(mrReqxReq, lrRequisitos)
mapeia(mrReqxPt, ptEscopo)

Na Empresa-B, a técnica de rastreabilidade bidirecional utilizada, também, é a "Matriz

de Rastreabilidade". Como mencionado, os requisitos de cliente e requisitos técnicos representam o mesmo conceito para o contexto da Empresa-B. Por esse motivo, não foi realizado um mapeamento entre essas duas entidades.

rVertical(Matriz de Rastreabilidade, Lista de Requisitos x Termo de Abertura,mrReqxTa)
rHorizontal(Matriz de Rastreabilidade, Lista de Requisitos x Lista deRequisitos, mrReqxReq)
reqCliente(Lista de Requisitos, Requisitos, lrRequisitos)
escopo(Termo de Abertura, Escopo, taEscopo)
mapeia(mrReqxTa, lrRequisitos)
mapeia(mrReqxReq, lrRequisitos)
mapeia(mrReqxTa, taEscopo)

A Empresa-C, utiliza uma ferramenta de ambiente corporativo. Desse modo, grande parte das informações dos projetos é registrada nesta ferramenta. Por esse motivo, o mecanismo de rastreabilidade utilizado por esta empresa baseia-se nos vínculos de informação gerados pela própria ferramenta.

Ressalta-se que as informações registradas em documentos físicos são anexadas à ferramenta corporativa, permitindo, assim, os mapeamentos a partir da própria ferramenta. Salienta-se, ainda, que os programas de melhoria sugerem que a organização implemente, pelo menos, uma rastreabilidade vertical e um rastreabilidade horizontal, não exigindo que todos os produtos de trabalho sejam mapeados.

rVertical(Ferramenta Corporativa, Link entre informações, fcLink)
rHorizontal(Ferramenta Corporativa, Link entre informações, fcLink)
reqCliente(Proposta Técnica, Requisitos, ptRequisitos)
escopo(Proposta Técnica, Escopo, ptEscopo)
mapeia(fcLink, ptRequisitos)
mapeia(fcLink, ptEscopo)

Nas empresas Empresa-A e Empresa-B, é utilizado o mesmo mecanismo de rastreabilidade (matriz de rastreabilidade). No caso da Empresa-C, a rastreabilidade é realizada de modo sistemático, no qual uma ferramenta corporativa é responsável por ligar os conteúdos dos produtos de trabalho do projeto. Mesmo utilizando mecanismos diferentes para contemplar a rastreabilidade bidirecional, essas empresas almejam encontrar as dependências dos produtos de trabalho. Assim, tanto a matriz de rastreabilidade quanto os *links* gerados pela ferramenta corporativa servem para rastrear as dependências dos produtos de trabalho, desse modo, validando os axiomas C1 e C2.

6.2.4 Revisão de Inconsistências

Na Empresa-A, as revisões de inconsistências são realizadas periodicamente. Essa revisão ocorre simultaneamente com os monitoramentos em pontos de controle. Durante essas

revisões, a "Matriz de Rastreabilidade" é utilizada para apoiar na busca por inconsistências nos produtos de trabalho.

As inconsistências (desvios) encontradas durante as revisões são registradas no documento "Relatório de Monitoramento do Projeto".

reqTecnico(Lista de Requisitos, Requisitos, lrRequisitos)
rVertical(Matriz de Rastreabilidade, Lista de Requisitos x Proposta Técnica, mrReqxPt)
rHorizontal(Matriz de Rastreabilidade, Lista de Requisitos x Lista deRequisitos, mrReqxReq)
revisaoInconstencia(Relatório de Monitoramento do Projeto, RequisitosAnalisados, rmpReqAnalisados)
apoia(mrReqxPt, rmpReqAnalisados)
apoia(mrReqxReq, rmpReqAnalisados)
revisa(rmpReqAnalisados, lrRequisitos)
produz(rmpReqAnalisados, rmpListaInconsistencias)
desvio(Relatório de Monitoramento do Projeto, Lista de Inconsistências, rmpListaInconsistencias)

No contexto da Empresa-B, a revisão é realizada periodicamente pela equipe. Todas as revisões realizadas são apoiadas com o uso da "Matriz de Rastreabilidade" e as inconsistências detectadas são registradas em uma ferramenta de *bugtracking*. Nessa ferramenta, as atividades de revisão realizadas são registradas em um *ticket* e, para cada inconsistência, é criado um *ticket*.

reqTecnico(Proposta Técnica, Requisitos, ptRequisitos)
rVertical(Matriz de Rastreabilidade, Lista de Requisitos x Termo de Abertura,mrReqxTa)
rHorizontal(Matriz de Rastreabilidade, Lista de Requisitos x Lista deRequisitos, mrReqxReq)
revisaoInconstencia(Ferramenta de Bugtracking, Ticket de descrição darevisão, fbTicketRevisão)
apoia(mrReqxTa, fbTicketRevisão)
apoia(mrReqxReq, fbTicketRevisão)
revisa(fbTicketRevisão, ptRequisitos)
produz(fbTicketRevisão, fbTicketInconsistencia)
desvio(Ferramentade de Bugtracking, Ticket de Inconsistências, fbTicketInconsistencia)

Por fim, a Empresa-C utiliza a funcionalidade de *ticket* de uma ferramenta de *bugtracking* para registrar as atividades de revisão de inconsistências. Além disso, para cada inconsistência detectada, um *ticket* é produzido. Este *ticket* é relacionado ao *ticket* de registro de atividade.

Como grande parte das informações do projeto está presente na ferramenta corporativa e essas informações estão ligadas, é possível navegar entre as informações sequencialmente, auxiliando, desse modo, a atividade de revisão. Ressalta-se que a ferramenta corporativa não possui uma funcionalidade de verificação automática de inconsistência, sendo que essa verificação é realizada manualmente por um operador.

reqTecnico(Proposta Técnica, Requisitos, ptRequisitos)
rVertical(Ferramenta Corporativa, Link entre informações, fcLink)
rHorizontal(Ferramenta Corporativa, Link entre informações, fcLink)
revisaoInconstencia(Ferramenta de Bugtracking, Ticket de descrição darevisão, fbTicketRevisão)
apoia(fcLink, fbTicketRevisão)
apoia(fcLink, fbTicketRevisão)
revisa(fbTicketRevisão, ptRequisitos)
produz(fbTicketRevisão, fbTicketInconsistencia)
desvio(Ferramentade de Bugtracking, Ticket de Inconsistências, fbTicketInconsistencia)

Nas três empresas entrevistadas, notou-se que o mecanismo de rastreabilidade institucionalizado é utilizado para apoiar as revisões de inconsistências entre os requisitos e os demais produtos de trabalho. Assim, pode-se validar o axioma D2.

Durante as revisões de inconstâncias dos requisitos, um conjunto de inconsistências é detectado e registrado, como descrito nos axiomas D3 e D4. Neste contexto, a Empresa-A utiliza um documento denominado de "Relatório de Monitoramento do Projeto" e as empresas Empresa-B e Empresa-C utilizam uma ferramenta de *bugtracking*. Como se pode notar, nas três empresas, inconsistências (desvios) são detectadas e registradas.

6.2.5 Acompanhamento dos Desvios

A última prática sugerida para gerência de requisitos é relacionada ao acompanhamento de mudanças. Na Empresa-A, é utilizado um produto de trabalho denominado de "Solicitação de Mudança de Requisitos". Esse documento possui todas as solicitações de mudanças pertencentes ao ciclo de desenvolvimento. Além disso, neste documento, são registrados as propostas de solução e o histórico da resolução de cada solicitação de mudança.

Além do documento "Solicitação de Mudanças de Requisitos", existe um produto de trabalho denominado de "Análise de Impacto da Mudança de Requisitos". Nesse artefato são descritas todas as consequências geradas ao realizarem a solicitação de dada mudança.

mudanca(Solicitação de Mudanças de Requisitos, Lista de Solicitação de Mudanças, smrListaMudanças)
analiseImpacto(Análise de Impacto da Mudança de Requisitos, Descrição, aimrDescrição)
avalia(aimrDescrição, smrListaMudanças)
acaoCorretiva(Solicitação de Mudanças de Requisitos, Plano de Ação, smrPlanoAção)
apoia(aimrDescrição, smrPlanoAção)
apoia(smrPlanoAção, smrListaMudanças)
hAcompanhamento(Solicitação de Mudanças de Requisitos, Histórico de Acompanhamento, smrHistóricoAcompanhamento)
possui(smrListaMudanças, smrHistóricoAcompanhamento)

A Empresa-B utiliza uma ferramenta de *bugtracking* para realizar o acompanhamento das mudanças de requisitos. Nessa ferramenta, é instanciado um *ticket* para cada solicitação de mudança. Assim, a própria ferramenta registra todas as mudanças realizadas no *ticket* até a sua conclusão. O plano de ação para cada solicitação de mudança é descrito no próprio corpo de descrição de cada *ticket*.

Em relação à análise de impacto das solicitações de mudança, é utilizado um documento denominado de "Análise de Impacto e Avaliações de Mudanças". Esse documento descreve as possíveis complicações de cada solicitação de mudança.

mudanca(Ferramenta de Bugtracking, Ticket de solicitação de mudança, fbTicketMudança)
analiseImpacto(Análise de Impacto e Avaliações de Mudanças, Descrição, aiamDescrição)
apoia(aiamDescrição, fbTicketMudança)
acaoCorretiva(Ferramenta de Bugtracking, Plano de Ação, fbPlanoAção)
apoia(aiamDescrição, fbPlanoAção)
apoia(fbPlanoAção, fbTicketMudança)
hAcompanhamento(FerramentadeBugtracking, Acompanhamento, fbHistóricoAcompanhamento)
Históricodepossui(fbTicketMudança, fbHistóricoAcompanhamento)

Finalmente, no contexto da Empresa-C, o acompanhamento de mudanças é realizado por uma ferramenta de *bugtracking*. Nessa ferramenta, é instanciado um *ticket* para cada solicitação de mudança. No corpo descritivo de cada *ticket* também se registra um plano de ação para a sua resolução. Como o controle das mudanças é feito pela ferramenta, todas as modificações realizadas sobre cada *ticket* são registradas em um histórico de acompanhamento.

A análise de impacto de cada solicitação de mudança é registrada em um produto de trabalho denominado de "Avaliação de Impacto".

mudanca(Ferramenta de Bugtracking, Ticket de solicitação de mudança, fbTicketMudança)
analiseImpacto(Avaliação de Impacto, Descrição, aiDescrição)
apoia(aiDescrição, fbTicketMudança)
acaoCorretiva(Ferramenta de Bugtracking, Plano de Ação, fbPlanoAção)
apoia(aiDescrição, fbPlanoAção)
apoia(fbPlanoAção, fbTicketMudança)
hAcompanhamento(FerramentadeBugtracking, Acompanhamento, fbHistóricoAcompanhamento)
Históricodepossui(fbTicketMudança, fbHistóricoAcompanhamento)

Para realizar o acompanhamento das mudanças nos requisitos, a Empresa-A utiliza um documento para registrar todas as mudanças realizadas do documento "Solicitação de Mudanças de Requisitos". Nesse mesmo documento, são descritas todas as ações cabíveis para resolução da solicitação de mudança.

No caso das empresas Empresa-B e Empresa-C, é utilizado o próprio fluxo do ciclo de

vida dos *tickets* instanciados na ferramenta de *bugtracking* institucionalizado. Essas ferramentas armazenam o histórico de todas as mudanças realizadas ao longo do projeto. Além disso, o conteúdo do plano de ação para cada solicitação de mudança é descrito no campo descrição de cada *ticket*. Assim, pode-se notar que os axiomas E5 e E6 são validados.

Em relação à análise de impacto, as três empresas utilizam um documento descrevendo as implicações que dada mudança de requisitos pode acarretar ao projeto, que apóia a resolução da solicitação de mudança, validando o axioma E8.

6.3 Avaliação e Interpretação dos Resultados

Durante as instanciações dos axiomas definidos nesta pesquisa, percebeu-se que as três empresas desenvolvedoras de *software* definem os requisitos em uma estrutura mais técnica possível, não existindo a transição entre requisitos de cliente e requisitos técnicos. Por esse motivo, ao instanciar estes conceitos, suas sintaxes ficam idênticas. Como consequência, durante a definição da rastreabilidade, não há a necessidade de mapear os requisitos de cliente aos requisitos técnicos. Entretanto, optou-se em manter a separação desses dois conceitos com o objetivo de manter uma estrutura genérica e compatível com os modelos CMMI-DEV e MR-MPS-SW. Nesses modelos, nota-se a evidente separação de requisitos de cliente e requisitos técnicos, obviamente, não obrigando que sejam entidades diferentes.

Outro ponto que deve ser comentado, refere-se à cardinalidade entre "RevisaoInconsistenciaRequisitos" e "Desvio", que foi modelada como uma relação de um para um ou mais. Nesse sentido, baseado no universo de possibilidades, é possível que, durante uma revisão de inconsistência de requisitos, nenhum desvio seja detectado.

Porém, durante as avaliações de programas de melhoria, é obrigatória a presença de, pelo menos, um desvio em cada projeto. Esse desvio é necessário para que o avaliador verifique se o processo de desenvolvimento da empresa é capaz de gerenciar os desvios detectados durante um projeto. Desse modo, serve como uma evidência de que a prática específica 1.3 e 1.5, ou os resultados esperados GRE4 e GRE5 estão satisfeitos.

Em relação à rastreabilidade, notou-se certa dificuldade durante a elaboração dos seus axiomas. Isso se deu pela própria natureza do conceito de rastreabilidade, que busca associar dependências entre duas ou mais entidades. Houve questionamentos da necessidade em definir axiomas de relacionamentos de dependência entre os conceitos mapeados pela rastreabilidade. Entretanto, decidiu-se não definir esses axiomas por dois motivos: (1) esses axiomas estariam no nível de conteúdo, ou seja, suas instâncias seriam conceitos/produtos intermediários, não fazendo parte do escopo desta pesquisa; e (2) a própria estrutura da ontologia permite a visualização das dependências dos produtos de trabalho (instâncias dos conceitos).

Por fim, notou-se que muitas das práticas recomendadas são satisfeitas por um mesmo

produto de trabalho. Por esse motivo, os predicados dos conceitos da ontologia foram definidos como uma tripla (produto de trabalho, conteúdo/seção, variável representativa), tornando as instâncias únicas.

7 CONCLUSÕES

Este artigo apresentou uma ontologia que define a representação dos relacionamentos de dependência entre as práticas presentes no processo/área de processo de Gerência de Requisitos do MPS.BR e CMMI, buscando o modelo baseado em processos de empresas de *software* oficialmente avaliadas. Assim, espera-se que a ontologia proposta nesta pesquisa fique nos moldes definidos nos modelos.

Com base nos resultados obtidos a partir das instanciações dos axiomas da ontologia desta pesquisa, pode-se notar a ampla complexidade para definir uma estrutura genérica, o suficiente para representar a interação entre as práticas do CMMI-DEV e MR-MPS-SW, devido à subjetividade da definição de um processo e de seus ativos.

Como trabalho futuro, planeja-se construir uma ferramenta para apoiar na implementação de processos/áreas de processo aderentes às práticas do MPS.BR e CMMI. Essa ferramenta possuirá um banco de dados de ativos de processo capazes de contemplar cada resultado esperado. Empresas que desejam definir um processo organizacional aderente ao CMMI ou MPS.BR poderão realizar, a partir da ferramenta, uma busca de sugestões de ativos para a implementação do seu processo, assim, apoiando implementações/avalições oficiais.

Como a avaliação dos axiomas foi realizada apenas a partir de revisão por pares, pretende-se, futuramente, converter a modelagem da ontologia em UML para uma linguagem voltada especificamente à modelagem de ontologias. Esta conversão evitará interpretações equivocadas devido à carga semântica inerente à linguagem UML.

8 AGRADECIMENTOS

Este trabalho recebeu apoio financeiro do CNPq por meio de bolsa institucional de pesquisa concedida ao PPGCC-UFPA. E agradece, também, aos especialistas que contribuíram para a revisão dos resultados parciais desta pesquisa e às empresas de *software* que permitiram a coleta dos produtos de trabalho usados nos seus processos de desenvolvimento de *software*. Este projeto faz parte do projeto *Software Process Improvement: DEvelopment and Research* (SPIDER), institucionalizado na Universidade Federal do Pará (UFPA).

Referências

ALMEIDA, M. B. **Um modelo baseado em ontologias para a representação da memória organizacional**. 2006. Tese de Doutorado — UFMG, Belo Horizonte – MG.

DUARTE, K. C.; FALBO, R. A. Uma ontologia de qualidade de software. In: **WQS2000 – Workshop de Qualidade de *Software***. João Pessoa: [s.n.], 2000.

FALBO, R. A. **Integração de Conhecimento em um Ambiente de Desenvolvimento de *Software***. dez. 1998. Tese de Doutorado — COPPE/UFRJ, Rio de Janeiro – RJ.

GUIZZARDI, G.; FALBO, R. A.; GUIZZARDI, Renata Silva Souza. A importância de ontologias de fundamentação para a engenharia de ontologias de domínio: o caso do domínio de processos de *Software*. **IEEE Latin America Transactions**, 2008.

HUANG, D.B.; ZHANG, W. CMMI in medium & small enterprises: Problems and solutions. In: **The 2nd IEEE International Conference on Information Management and Engineering (ICIME)**. Chine: [s.n.], 2010.

ISO/IEC – INTERNATIONAL ORGANIZATION FOR STANDARDIZATION/ THE INTERNATIONAL ELECTROTECHNICAL COMISSION. **ISO/IEC 15504-2: Information Technology - Process Assessment - Part 2: Performing an Assessment**. [S.l.]: Geneve, 2004.

LEAL, A.L. de Castro et al. Ontology based *software* development good practices selection for adoption in micro-enterprises. In: **XXXVIII Conferencia Latinoamericana en Informática – CLEI**. [S.l.: s.n.], 2012.

MELLO, M. S. **Melhoria de Processo de *Software* Multi-Modelos Baseada nos Modelos MPS e CMMI-DEV**. 2011. Dissertação de Mestrado — Escola de Engenharia de São Carlos – USP, São Carlos – SP.

MORGADO, Gisele P. et al. **Práticas do CMMI como Regras de Negócio**. São Paulo: Produção, 2007.

NETO, A. Colenci. **Proposta de um Modelo de Referência para Desenvolvimento de Software com Foco na Certificação do MPS.BR**. 2008. Tese de Doutorado — Instituto Alfredo Luiz de Coimbra - UFRJ/COPPE, Rio de Janeiro – RJ.

PRESSMAN, Roger S. **Engenharia de software**. Rio de Janeiro: McGraw-Hill, 2006. Tradução José Carlos Barbosa dos Santos.

SCHOTS, Natália Chaves Lessa et al. Lições aprendidas em implementações de melhoria de processos em organizações com diferentes características. In: VII WORKSHOP ANUAL DO MPS.BR – WAMPS. **Anais...** Campinas – SP, 2011.

SEI – SOFTWARE ENGINEERING INSTITUTE. **Capability Maturity Model Integration for Development – CMMI-Dev. Versão 1.3**. [S.l.]: Carnegie Mellon, 2010.

SHARIFLOO, Amir Azim et al. An ontology for cmmi-acq model. In: 3ND INTERNATIONAL CONFERENCE ON INFORMATION AND COMMUNICATION TECHNOLOGIES: FROM THEORY TO APPLICATIONS. **Anais...** Damascus, 2008.

SOFTEX. **Melhoria do Processo de Software Brasileiro – Guia de Implementação - Parte 1: Fundamentação para Implementação do Nível G do MR-MPS**. [S.l.], 2011.

SOFTEX. **Melhoria do Processo de Software Brasileiro – Guia Geral MPS de Software**. [S.l.], 2012.

SOFTEX. **Melhoria do Processo de Software Brasileiro – Guia de Implementação - Parte 1: Implementação e Avaliação do MR-MPS-SW:2012 em Conjunto com o CMMI-DEV v1.3**. [S.l.], 2012.

SOYDAN, Gokhan Halit; KOKAR, Mieczyslaw M. An owl ontology for representing the cmmi-sw model. In: 2ND INTERNATIONAL WORKSHOP ON SEMANTIC WEB ENABLED SOFTWARE ENGINEERING. **Proceedings...** Athens – GA, 2006.

USCHOLD, Mike; GRUNINGER, Michael. Ontologies: Principles, methods and applications. **Knowledge Engineering Review**, Canada, 1996.

VALENTE, André. Legal knowledge engineering. **A Modelling Approach, IOS Press**, 1995.