

Uma Ontologia que Estabelece os Relacionamentos de Dependência entre as Práticas de Gerência de Projetos constantes nos modelos CMMI-DEV e MR-MPS-SW.

Ewelton Yoshio Chiba Yoshidome¹, Sandro Ronaldo Bezerra Oliveira¹

¹Programa de Pós-Graduação em Ciência da Computação – Universidade Federal do Pará (UFPA), Rua Augusto Corrêa, 1 – Guamá – 66075-110 – Belém – PA – Brasil

ewelton.yoshio@gmail.com, srbo@ufpa.br

Abstract. *This paper presents the ontology about project management, which establishes the dependency relationships between the practices (specific practices and expected results) of this process included in CMMI-DEV and MR-MPS-SW models, which aims to supporting beginner's consultants these models in understanding the subject. Thus, this research will help consultants and software development companies that aim to improve on their process. For this, we did a survey on three companies what have official evaluation, in order to collect evidences to instantiate and evaluate the ontology.*

Resumo. *Este artigo apresenta uma ontologia de gerência de projetos, na qual estabelece os relacionamentos de dependência entre as práticas (práticas específicas e resultados esperados) deste processo constantes nos modelos CMMI-DEV e MR-MPS-SW, com o objetivo de apoiar consultores-implementadores iniciantes destes modelos no entendimento sobre o assunto. Desta forma, esta pesquisa irá auxiliar consultores e empresas desenvolvedoras de software que almejam a melhoria em seu processo. Para isto, foi realizada uma pesquisa de campo em três empresas com avaliações oficiais com o intuito de coletar evidências para realizar a instanciação e avaliação da ontologia.*

1. Introdução

Empresas desenvolvedoras de software que desejam implementar programas de melhoria em seus processos organizacionais podem se deparar com o problema de conciliar entre 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, 2007].

Existe no mercado um número considerável de empresas desenvolvedoras de software que buscam por avaliação de melhoria do processo em modelos constantes nos programas MPS.BR – Melhoria do Processo de Software Brasileiro [SOFTEX, 2012a], e CMMI – *Capability Maturity Model Integration* [SEI, 2010]. Para que estas empresas consigam estas avaliações, é necessário que a empresa contemple um determinado conjunto de requisitos de cada processo 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. Este problema complica-se com o fato das constantes atualizações que os modelos de qualidade sofrem ao longo do tempo, com a adição de novas práticas e/ou processos em sua estrutura [Sharinfloo *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, estas 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 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 e Falbo, 2000].

A pesquisa descrita em Leal (2012), apresenta a necessidade de adotar um mecanismo para a seleção e implementação gradual de boas práticas a micro e pequenas empresas. Para isto, a abordagem utilizada foi a definição de uma ontologia contendo as referidas práticas. Entretanto, esta 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 MR-MPS-SW – Modelo de Referência do MPS para Software. Estes relacionamentos estão compatíveis com o modelo CMMI-DEV. A compatibilidade é evidenciada no Guia de Implementação Parte 11 do MPS.BR [SOFTEX, 2012b]. Devido a essa compatibilidade e a existência de um documento que garante a sua compatibilidade, optou-se em utilizar apenas os dois modelos citados. 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 ontologia serve como um mecanismo de disseminação de conhecimento sobre boas práticas de engenharia de software para consultores implementadores iniciantes, para apoiá-los na identificação e levantamento dos principais produtos de trabalho utilizados para contemplar o processo de gerência de projetos.

O escopo da ontologia desta pesquisa está restrito ao processo de Gerência de Projetos (GPR) do MR-MPS-SW e as áreas de processo de Planejamento do Projeto e Monitoramento e Controle do Projeto do CMMI-DEV. Entretanto, esta pesquisa está inserida no contexto de uma Dissertação de Mestrado defendida e aprovada do PPGCC-UFPA, na qual abrange todo o nível de maturidade G do MR-MPS.SW e parte do nível de maturidade 2 do CMMI-DEV. A ontologia do processo de Gerência de Requisitos está em processo de publicação na Revista Abakós [Yoshidome e Oliveira, 2014].

O objetivo do processo de Gerência de Projetos (GPR) é fornecer subsídios para o estabelecimento e manutenção de planos visando definir as atividades do projeto. Além disto, este processo é responsável em realizar o acompanhamento do projeto,

provendo visibilidade para a implementação de ações corretivas em caso de desvios no plano [SOFTEX, 2011; SEI, 2010].

A escolha do nível G do MR-MPS-SW (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 os consultores implementadores iniciantes atuarão, principalmente, nos processos desse nível de maturidade.

Além desta seção introdutória, este artigo está dividido nas seguintes seções: a Seção 2 descreve alguns dos trabalhos relacionados a esta pesquisa; a Seção 3 descreve a metodologia utilizada para este trabalho; a Seção 4 apresenta a ontologia de gerência de projetos desta pesquisa; a Seção 5 descreve as instanciações realizadas para apoiar na avaliação desta pesquisa; e, por fim, a Seção 6 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, esta 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, 2012]. A ontologia definida nesta pesquisa busca adequar-se à versão 1.3 do CMMI, o qual está compatível com a versão de 2012 do MPS.BR.

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

Colenci Neto (2008) propõe um modelo de referência para que as próprias empresas realizem uma avaliação 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 uma forma 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 estas 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 *et al.* (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 pelos modelos de qualidade de software.

3. Metodologia de Pesquisa

A metodologia de pesquisa utilizada neste trabalho foi baseada na metodologia proposta por Falbo (1998). Falbo propôs sua metodologia baseando-se nas práticas presentes nas principais metodologias de construções de ontologia da literatura. Além disso, a metodologia de Falbo foi elaborada para o contexto de engenharia de *software*.

A primeira etapa compreende a Pesquisa Bibliográfica, na qual foram realizadas as atividades de identificação, análise e seleção de trabalhos relacionados e bibliografias disponíveis sobre o tema. Outra atividade realizada durante esta etapa foram pesquisas sobre os conceitos relacionados à melhoria de processo de *software* e as práticas recomendadas, além de analisar como estas práticas relacionam-se.

Ainda, na primeira etapa, também foi definido um mapa mental (disponível em http://spider.ufpa.br/projetos/spider_ontologia/SPIDER_MapaNental_GPRGRE.png), que descreve as dependências entre as práticas presentes no MR-MPS-SW. Por meio deste mapeamento, foi possível iniciar a modelagem e a definição da ontologia. Vale ressaltar que o Guia de Implementação Parte 11 estabelece a compatibilidade entre as práticas do MR-MPS-SW e CMMI-DEV.

A segunda etapa consistiu na elaboração da ontologia de dependência entre as práticas do CMMI-DEV e MR-MPS-SW. Nesta etapa, foi definido o objetivo da ontologia, além de identificar as questões que a ontologia deve responder.

Na terceira etapa, a captura e formalização da ontologia, foi estruturada uma modelagem da ontologia por meio da linguagem UML – *Unified Modeling Language*. A partir da modelagem foram definidos axiomas por meio de lógica de primeira ordem, buscando descrever a ontologia em uma linguagem formal. O uso de lógica de primeira ordem para a definição de axiomas proporciona uma estrutura matemática das regras do universo em discurso, restringindo o comportamento e os relacionamentos dos conceitos envolvidos. Uma dedução em linguagem natural, geralmente, envolve pressuposições implícitas que entram despercebidas no processo de dedução [Carnap, 1958].

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 desta pesquisa foi identificar os nomes dos principais produtos de trabalho utilizados para contemplar as práticas do MR-MPS-SW. Além disso, os produtos de trabalhos coletados durante essa pesquisa contribuíram para apoiar na avaliação da ontologia, por meio das instanciações dos axiomas definidos. Como perfil resumido dessas empresas tem-se:

- A Empresa-A é uma empresa que oferece serviços de desenvolvimento de software, possui como nicho de mercado a comunidade acadêmica e o serviço público. Seu corpo de desenvolvimento é composto por 5 (cinco) integrantes. Encontra-se no nível de maturidade G;
- 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, esta empresa é focada no desenvolvimento de sistemas e páginas web, e sua equipe de

desenvolvimento é composta por 10 (dez) pessoas. Encontra-se no nível de maturidade F;

- 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. Encontra-se no nível de maturidade F.

Em seguida, foram realizadas duas revisões sobre a ontologia: a primeira sobre a modelagem e a segunda sobre os axiomas. A primeira revisão foi realizada em uma reunião junto a um especialista denominado de Especialista-A. Este 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. Nesta reunião foi explicado e justificado cada conceito e relacionamento modelado na ontologia. Ao final da revisão, o especialista apresentou suas considerações e necessidades de mudança.

A segunda revisão teve como objetivo buscar problemas sintáticos e semânticos na estrutura dos axiomas da ontologia. Esta revisão foi realizada em uma reunião junto ao Especialista-B. Este especialista possui ampla experiência em assuntos relacionados à lógica de primeira ordem, com inúmeros trabalhos publicados nesta área, e é professor do Programa de Pós-Graduação da UFPA. Além disso, o especialista atua na área de inteligência computacional aplicada e tecnologias de educação à distância. No final da revisão, o especialista propôs sugestões de melhoria e correções, as quais foram contempladas.

Por fim, a última etapa está relacionada à instanciação da ontologia desta pesquisa, como forma 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.

4. A Ontologia para o Processo de Gerência de Projetos

Como foi mencionado, o domínio de interesse desta pesquisa consiste em definir os relacionamentos de dependência entre as práticas do processo Gerência de Projetos (GPR) e áreas de processo de PP e PMC. As evidências destas práticas são produzidas pelas empresas desenvolvedoras de software a partir de sua institucionalização. Por este motivo, definiu-se que o universo de discurso da ontologia refere-se às práticas presentes no processo/áreas de processo.

A elaboração das questões de competência para esta pesquisa é baseada no modelo descrito por [Falbo, 1998]. Estas questões são:

1. Como o planejamento do projeto é estruturado?
2. Como o projeto é estimado?
3. Como são planejados os recursos humanos e de infraestrutura do projeto?
4. Como é definido o ciclo de vida do projeto?
5. Como é definido o cronograma?
6. Como o custo e orçamento são definidos?
7. Como são definidos os riscos do projeto?
8. Como são definidos os dados relevantes do projeto?
9. Como é definido o planejamento da comunicação do projeto?
10. Como é estabelecido o plano do projeto?
11. Como o projeto é revisado com todos os interessados?

12. Como é realizada a análise de viabilidade do projeto?
13. Como são realizados os monitoramentos dos parâmetros do projeto?
14. Como são realizados os monitoramentos dos recursos e dados relevantes do projeto?
15. Como são realizados os monitoramentos dos riscos do projeto?
16. Como são realizados os monitoramentos em marcos do projeto?
17. Como é realizado o gerenciamento dos desvios do projeto?

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

- Estrutura do Planejamento do Projeto (questões 1 e 10);
- Definição das Estimativas (questão 2);
- Definição dos Recursos do Projeto (questão 3);
- Definição do Cronograma do Projeto (questões 4 e 5);
- Definição do Custo e Orçamento (questão 6);
- Definição dos Riscos do Projeto (questão 7);
- Definição dos Dados Relevantes (questões 8);
- Definição da Comunicação do Projeto (questões 9);
- Comprometimento com Planejamento do Projeto (questão 11);
- Análise de Viabilidade do Projeto (questão 12);
- Monitoramentos do Projeto (questões 13, 14, 15 e 16);
- Acompanhamento dos Desvios do Projeto (questão 17).

Vale ressaltar que a 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 esta linguagem. Somando-se a isto, o formalismo presente na linguagem é capaz de expressar grande parte dos relacionamentos existentes entre as práticas constantes no processo de Gerência de Projetos. Além disso, os axiomas da ontologia foram estruturados por meio de lógica de primeira ordem, com o objetivo de consolidar e restringir relacionamentos nas quais a linguagem UML não é capaz de definir.

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 citada anteriormente, será utilizado o documento “Plano do Projeto” para contemplar a prática referente à definição do escopo do projeto, definido pelo predicado *escopo(e,*,comb-req)*. Nesta situação, o primeiro parâmetro do predicado é preenchido por “Plano do Projeto”. Como, neste caso, o documento “Plano do Projeto” pode conter outras informações, tais

como: escopo, estimativas, cronograma, entre outros; um segundo parâmetro estabelecendo a localização da descrição do escopo deve ser informado ao predicado, como por exemplo, uma seção denominada de “Escopo do Projeto”. Por fim, pelo fato do documento “Plano do Projeto” envolver várias práticas esperadas em programas de melhoria, deve-se definir o terceiro parâmetro, “ppEscopo”. Este parâmetro tem como objetivo prover a unicidade da prática de definição do escopo. Assim, ao final o predicado foi definido como *escopo(Plano do Projeto, Escopo do Projeto, ppEscopo)*.

Nas subseções seguintes serão apresentados a ontologia de gerência de projetos, por restrições de espaço, apenas parte da pesquisa é apresentada. Salienta-se que esta ontologia é parte de uma dissertação de mestrado, o qual foi defendido em 16 de abril de 2014. Assim, o conteúdo completo pode ser encontrado no referido documento. A dissertação pode ser encontrado em [http://www.spider.ufpa.br/projetos/spider_ontologia/SPIDER_Uma Ontologia que Estabelece os Relacionamentos de Dependência entre as Práticas de Gerência de Requisitos e Gerência de Projetos Constantes nos Modelos MR-MPS-SW e CMMI-DEV.pdf](http://www.spider.ufpa.br/projetos/spider_ontologia/SPIDER_Uma_Ontologia_que_Estabelece_os_Relacionamentos_de_Dependência_entre_as_Práticas_de_Gerência_de_Requisitos_e_Gerência_de_Projetos_Constantes_nos_Modelos_MR-MPS-SW_e_CMMI-DEV.pdf).

4.1. Estrutura do Planejamento do Projeto

Inicialmente, estabelece-se o conjunto de todos os planejamentos específicos que compõem o planejamento do projeto. Esta composição foi definida, pois os programas de melhoria recomendam o estabelecimento de um plano geral do projeto contendo todos os seus planejamentos específicos. Os referidos planejamentos são: as estimativas, os riscos, o ciclo de vida, o cronograma, o planejamento dos recursos e dados, o escopo do projeto, o custo e o orçamento. Os planejamentos específicos, citados anteriormente, foram agrupados em quatro categorias [SEI, 2010]: planejamento de parâmetros do projeto; planejamento de recursos e dados do projeto; planejamento dos riscos do projeto; e planejamento da comunicação do projeto. A Figura 1 apresenta este agrupamento.

O planejamento de parâmetros do projeto (classe “PlanejamentoParametros”) está composto pelo escopo, ciclo de vida, cronograma, estimativas (tamanho e esforço), custo e orçamento. O planejamento dos recursos e dados do projeto (classe “PlanejamentoRecursoDados”) é composto pelos conceitos referentes a recursos (humanos e infraestrutura) e dados relevantes do projeto. O planejamento dos riscos (classe “PlanejamentoRiscos”) representa os riscos do projeto. Por fim, o planejamento da comunicação (classe “PlanejamentoComunicacao”) representa como os envolvidos irão interagir.

O objetivo do agrupamento dos planejamentos específicos provém da necessidade de definir regras mais genéricas quando estes planejamentos interagirem com as etapas de monitoramento do projeto, que são apresentadas mais adiante.

Para representar o relacionamento de composição de “PlanejamentoParametros”, “PlanejamentoRecursoDados”, “PlanejamentoRiscos” e “PlanejamentoComunicacao” com a classe “PlanejamentoProjeto”, foram definidos, respectivamente, os predicados *pParametros(ppa, *, comb-ppa)*, *pRecursosDados(prd, *, comb-prd)*, *pRiscos(pri, *, comb-pri)*, *pComunicacao(pcom, *, comb-pcom)* e *pProjeto(pp, *, comb-pp)*. Baseado nestes predicados, os axiomas A1, A2, A3 e A4 foram formulados, denotando a relação de composição.

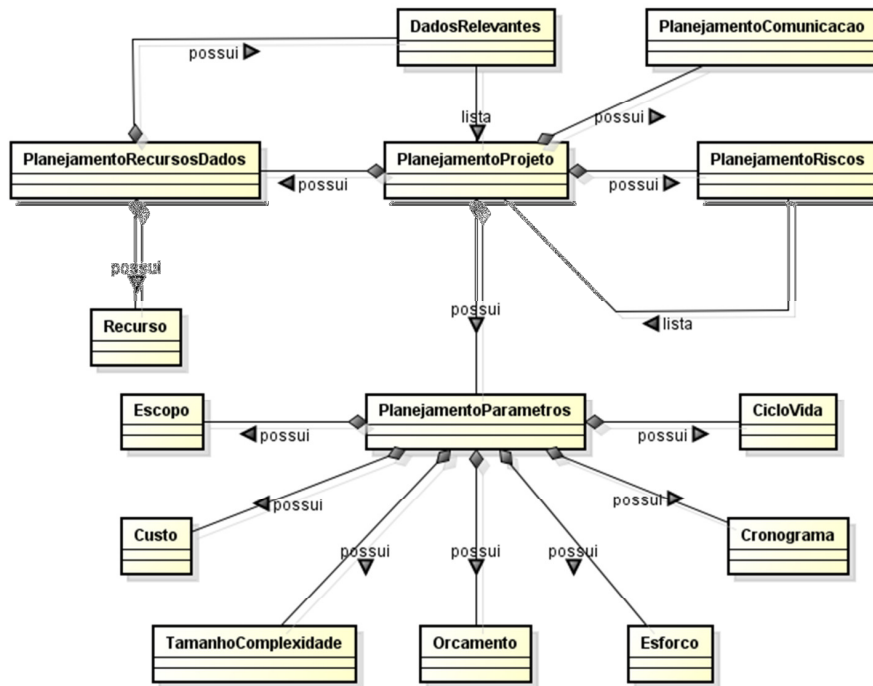


Figura 1. Estrutura do Planejamento do Projeto

$(\forall pp, comb - pp, ppa, comb - ppa)(possui(comb - pp, comb - ppa) \rightarrow pProjeto(pp, *, comb - pp) \cap pParametros(ppa, *, comb - ppa))$	A1
$(\forall pp, comb - pp, prd, comb - prd)(possui(comb - pp, comb - prd) \rightarrow pProjeto(pp, *, comb - pp) \cap pRecursosDados(prd, *, comb - prd))$	A2
$(\forall pp, comb - pp, pri, comb - pri)(possui(comb - pp, comb - pri) \rightarrow pProjeto(pp, *, comb - pp) \cap pRiscos(pri, *, comb - pri))$	A3
$(\forall pp, comb - pp, pcom, comb - pcom)(possui(comb - pp, comb - pri) \rightarrow pProjeto(pp, *, comb - pp) \cap pComunicacao(pcom, *, comb - pcom))$	A4

Como mencionado, a classe “PlanejamentoParametros” é composta pelos conceitos escopo, ciclo de vida, cronograma, tamanho, esforço, custo e orçamento. Estes conceitos foram denotados, respectivamente, pelos predicados *escopo*(*e*, *, *comb-e*), *cicloVida*(*cv*, *, *comb-cv*), *cronograma*(*cr*, *, *comb-cr*), *tComplexidade*(*tc*, *, *comb-tc*), *esforco*(*esf*, *, *comb-esf*), *custo*(*c*, *, *comb-c*) e *orcamento*(*o*, *, *comb-o*). Os axiomas de A5 a A11 descrevem a relação de composição da classe “PlanejamentoParametro”.

$(\forall ppa, comb - ppa, e, comb - e)(possui(comb - ppa, comb - e) \rightarrow pParametros(ppa, *, comb - ppa) \cap escopo(e, *, comb - e))$	A5
$(\forall ppa, comb - ppa, cv, comb - cv)(possui(comb - ppa, comb - cv) \rightarrow pParametros(ppa, *, comb - ppa) \cap cicloVida(cv, *, comb - cv))$	A6

$(\forall ppa, comb - ppa, cr, comb - cr)(possui(comb - ppa, comb - cr) \rightarrow pParametros(ppa, *, comb - ppa) \cap cronograma(cr, *, comb - cr))$	A7
$(\forall ppa, comb - ppa, tc, comb - tc)(possui(comb - ppa, comb - tc) \rightarrow pParametros(ppa, *, comb - ppa) \cap tComplexidade(tc, *, comb - tc))$	A8
$(\forall ppa, comb - ppa, esf, comb - esf)(possui(comb - ppa, comb - esf) \rightarrow pParametros(ppa, *, comb - ppa) \cap esforco(esf, *, comb - esf))$	A9
$(\forall ppa, comb - ppa, c, comb - c)(possui(comb - ppa, comb - c) \rightarrow pParametros(ppa, *, comb - ppa) \cap custo(c, *, comb - c))$	A10
$(\forall ppa, comb - ppa, o, comb - o)(possui(comb - ppa, comb - o) \rightarrow pParametros(ppa, *, comb - ppa) \cap orcamento(o, *, comb - o))$	A11

No contexto da classe “PlanejamentoRecursosDados”, os conceitos presentes em sua composição são todos os recursos do projeto e dados relevantes produzidos no projeto. Estes conceitos foram definidos por meio das classes “Recurso” e “DadosRelevantes”. Assim, os predicados derivados a partir destas classes foram: *recurso(r, *, comb-r)* e *dadosRelevantes(dr, *, comb-dr)*. Os axiomas A12 e A13 descrevem o relacionamento de composição da classe “PlanejamentoRecursosDados”.

$(\forall pdr, comb - pdr, r, comb - r)(possui(comb - pdr, comb - r) \rightarrow pRecursosDados(pdr, *, comb - pdr) \cap recurso(r, *, comb - r))$	A12
$(\forall pdr, comb - pdr, dr, comb - dr)(possui(comb - pdr, comb - dr) \rightarrow pRecursosDados(pdr, *, comb - pdr) \cap dadosRelevantes(dr, *, comb - dr))$	A13

A classe “PlanejamentoRiscos” representa o conceito que evidencia a identificação de um conjunto de riscos do projeto. Como esta classe é composta apenas pelos riscos do projeto, não houve a necessidade de definir uma composição.

Finalmente, a classe “PlanejamentoComunicacao” representa o conceito que indica o planejamento da forma de comunicação e interação entre os envolvidos do projeto. De forma similar aos riscos do projeto, esta classe é composta apenas pelo planejamento da comunicação, por este motivo, não houve a necessidade de criar um relacionamento de composição.

4.2. Monitoramento do Projeto

Os monitoramentos do projeto são atividades responsáveis por verificar se o planejamento está sendo cumprido devidamente. Os programas de melhoria recomendam que seja monitorado o planejamento do projeto em marcos e pontos de controle. Nos marcos e em pontos de controle são realizadas revisões buscando encontrar desvios no planejamento (escopo, riscos, estimativas, entre outros). Os marcos são, geralmente, referentes ao início de cada fase do projeto que buscam encontrar desvios no planejamento de todo o projeto. Os pontos de controle são monitoramentos mais frequentes, focando-se, geralmente, em planejamentos específicos do projeto. Para isto, definiu-se uma classe denominada de “Monitoracao”. Esta classe possui duas sub-

classes: “MonitoracaoMarcos” e “MonitoracaoPeriodica”, referentes, respectivamente, aos marcos e pontos de controle.

Em programas de melhoria, como o CMMI e MPS.BR, são explicitados que cada planejamento específico (escopo, estimativas, custo, orçamento, cronograma, etc.) do projeto deve ser monitorado. Assim, foram criadas quatro classes filhas para a classe “MonitoracaoPeriodica”: “MonitoracaoComunicacao”, “MonitoracaoParametros”, “MonitoracaoRecursosDados” e “MonitoracaoRiscos”. A Figura 2 contém a referida estrutura.

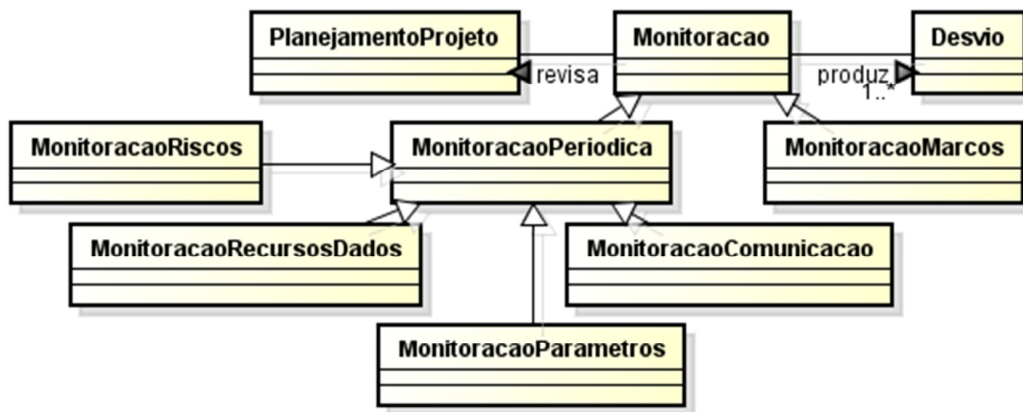


Figura 2. Monitoramento do Projeto

A “MonitoracaoComunicacao” busca monitorar o planejamento da comunicação do projeto. A classe “MonitoracaoParametros” busca encontrar os desvios no planejamento de escopo, estimativas, atividades, custo, orçamento e cronograma. A classe “MonitoracaoRecursosDados” denota a necessidade de verificar o planejamento dos dados relevantes do projeto, além dos recursos humanos e físicos alocados. Por fim, a classe “MonitoracaoRiscos” busca verificar por desvios no planejamento dos riscos do projeto.

Para definir os axiomas, foram utilizados os predicados $monitoracao(m, *, comb - m)$, denotando a classe “Monitoracao”; $mMarcos(m, *, comb - m)$ e $mPeriodica(m, *, comb - m)$, denotando, respectivamente, os monitoramentos em marcos e pontos de controle. Por fim, definiram-se os predicados $mParametros(m, *, comb - m)$, $mRiscos(m, *, comb - m)$, $mComunicacao(m, *, comb - m)$ e $mRecursosDados(m, *, comb - m)$, descrevendo, os monitoramentos em pontos de controle que verificam, respectivamente, os planejamentos de parâmetros, riscos, comunicação e recursos e dados relevantes do projeto.

Baseada na estrutura ilustrada na Figura 2, foram definidos os axiomas B1 e B2, para descrever as subclasses de “Monitoracao” e os axiomas B3 a B6, para descrever as subclasses de “MonitoracaoPeriodica”.

$(\forall m, comb - m)(mMarcos(m, * comb - m) \rightarrow monitoracao(m, *, comb - m))$	B1
$(\forall m, comb - m)(mPeriodica(m, * comb - m) \rightarrow monitoracao(m, *, comb - m))$	B2
$(\forall m, comb - m)(mParametro(m, * comb - m) \rightarrow mPeriodica(m, *, comb - m))$	B3
$(\forall m, comb - m)(mRiscos(m, * comb - m) \rightarrow mPeriodica(m, *, comb - m))$	B4

$(\forall m, comb - m)(mComunicacao(m, * comb - m) \rightarrow mPeriodica(m, *, comb - m))$	B5
$(\forall m, comb - m)(mRecursosDados(m, * comb - m) \rightarrow mPeriodica(m, *, comb - m))$	B6

Como mencionado, os monitoramentos buscam verificar desvios no planejamento do projeto. Assim, para denotar esta relação, estabeleceu-se um relacionamento do tipo *revisa* entre as classes “Monitoracao” e “PlanejamentoProjeto”.

Outra característica que deve ser citada durante os monitoramentos do projeto, refere-se à necessidade de registrar todos os desvios encontrados. Por este motivo, estabeleceu-se um relacionamento do tipo *produz* entre as classes “Monitoracao” e “Desvio”. A classe “Desvio” denota o registro de todos os problemas, inconsistências, mudanças, não conformidades, defeitos, etc., que são encontrados ao longo do projeto. A Figura 2, também ilustra os relacionamentos da classe “Monitoracao”.

Salienta-se que a cardinalidade entre as classes “Monitoracao” e “Desvio” é 1-n. Isto se deve ao fato de que os programas de melhoria evidenciam a realização dos monitoramentos em marcos e pontos de controle por meio dos registros de desvios do projeto. Assim, sendo necessário o registro de pelo menos um desvio para cada monitoramento, como forma de evidência durante avaliações oficiais de melhoria de processo. Além disso, se a cardinalidade do relacionamento *produz* fosse 0-n, não seria possível definir um axioma [Falbo, 1998].

Para denotar as classes “PlanejamentoProjeto” e “Desvio”, foram utilizados, respectivamente, os predicados $pProjeto(pp, *, comb - pp)$ e $desvio(d, *, comb - d)$.

Vale ressaltar que a classe “Monitoracao” possui duas sub-classes: “MonitoracaoMarcos” e “MonitoracaPeriodica”. Assim, pode-se detalhar o relacionamento *revisa* na perspectiva de cada uma das sub-classes.

Os monitoramentos em marcos do projeto buscam verificar os desvios de todo o planejamento do projeto. Portanto, ligou-se as classes “MonitoracaoMarcos” e “PlanejamentoProjeto” por meio do relacionamento *revisa* diretamente, como descrito no axioma B7.

$(\forall m, comb - m, pp, comb - pp, cr, comb - cr)(revisa(comb - m, comb - pp) \rightarrow pProjeto(pp, *, comb - pp) \cap mMarcos(m, *, comb - m) \cap possui(comb - cr, comb - m) \cap cronograma(cr, *, comb - cr))$	B7
---	----

Vale ressaltar que os monitoramentos em marcos estão relacionados à classe “Cronograma” por meio do relacionamento *possui*. Definiu-se um axioma restringindo que *o planejamento do projeto será monitorado nos marcos se, e somente se, existir um cronograma e os marcos do projeto fizerem parte desse cronograma*.

$(\forall m, comb - m, pp, comb - pp, cr, comb - cr)(revisa(comb - m, comb - pp) \cap pProjeto(pp, *, comb - pp) \cap mMarcos(m, *, comb - m) \leftrightarrow (\exists cr) cronograma(cr, *, comb - cr) \cap possui(comb - cr, comb - m))$	B8
--	----

Os monitoramentos em pontos de controle buscam verificar os planejamentos específicos do projeto. Como visto anteriormente, a classe “MonitoracaPeriodica” foi

classificada em quatro subtipos: “MonitoracaoComunicacao”, “MonitoracaoParametros”, “MonitoracaoRecursosDados” e “MonitoracaoRiscos”. Assim, para denotar os monitoramentos em pontos de controle, o relacionamento *revisa* parte das quatro sub-classes de “MonitoracaoPeriodica”.

Como mencionado, as sub-classes de “MonitoracaoPeriodica” buscam verificar um conjunto específico do planejamento total do projeto. Devido a isto, o relacionamento foi feito baseado nas definições dos A1 ao A4. Assim, as classes “MonitoracaoParametros”, “MonitoracaoRecursosDados”, “MonitoracaoRiscos” e “MonitoracaoComunicacao” relacionam-se, respectivamente, às classes “PlanejamentoParametros”, “PlanejamentoRecursosDados”, “PlanejamentoRiscos” e “PlanejamentoComunicacao”, como podem ser vistos nos axiomas de B9 a B12.

$(\forall m, comb - m, ppa, comb - ppa)(revisa(comb - m, comb - ppa) \rightarrow pParametros(ppa, *, comb - ppa) \cap mParametros(m, *, comb - m))$	B9
$(\forall m, comb - m, prd, comb - prd)(revisa(comb - m, comb - prd) \rightarrow pRecursosDados(prd, *, comb - prd) \cap mRecursosDados(m, *, comb - m))$	B10
$(\forall m, comb - m, pri, comb - pri)(revisa(comb - m, comb - pri) \rightarrow pRiscos(pri, *, comb - pri) \cap mRiscos(m, *, comb - m))$	B11
$(\forall m, comb - m, pcom, comb - pcom)(revisa(comb - m, comb - pcom) \rightarrow pComunicacao(pcom, *, comb - pcom) \cap mComunicacao(m, *, comb - m))$	B12

De forma similar aos monitoramentos em marcos, os monitoramentos em pontos de controle também estão relacionados à classe “Cronograma” por meio do relacionamento *possui*. Devido a isso, foram definidos os axiomas B13 a B16, restringindo que *os planejamentos específicos serão monitorados em pontos de controle se, e somente se, existir um cronograma e os monitoramentos em pontos de controle estiverem contidos nesse cronograma.*

$(\forall m, comb - m, ppa, comb - ppa, cr, comb - cr)(revisa(comb - m, comb - ppa) \cap pParametros(ppa, *, comb - ppa) \cap mParametros(m, *, comb - m) \leftrightarrow (\exists cr) cronograma(cr, *, comb - cr) \cap possui(comb - cr, comb - m))$	B13
$(\forall m, comb - m, prd, comb - prd, cr, comb - cr)(revisa(comb - m, comb - prd) \cap pRecursosDados(prd, *, comb - prd) \cap mRecursosDados(m, *, comb - m) \leftrightarrow (\exists cr) cronograma(cr, *, comb - cr) \cap possui(comb - cr, comb - m))$	B14
$(\forall m, comb - m, pri, comb - pri, cr, comb - cr)(revisa(comb - m, comb - pri) \cap pRiscos(pri, *, comb - pri) \cap mRiscos(m, *, comb - m) \leftrightarrow (\exists cr) cronograma(cr, *, comb - cr) \cap possui(comb - cr, comb - m))$	B15
$(\forall m, comb - m, pcom, comb - pcom, cr, comb - cr)(revisa(comb - m, comb - pcom) \cap pComunicacao(pcom, *, comb - pcom) \cap mComunicacao(m, *, comb - m) \leftrightarrow (\exists cr) cronograma(cr, *, comb - cr) \cap possui(comb - cr, comb - m))$	B16

Para denotar o relacionamento *produz* entre as classes “Monitoracao” e “Desvio”, foi estabelecido o axioma B17. Entretanto, deve-se salientar que o desvio surgirá somente quando um monitoramento for realizado no planejamento. Assim, pode-se definir que *um monitoramento produzirá um desvio se, e somente se, existir um planejamento na qual será revisado*, como descrito no axioma B18.

$(\forall m, comb - m, d, comb - d)(produz(comb - m, comb - d) \rightarrow desvio(d, *, comb - d) \cap monitoracao(m, *, comb - m))$	B17
$(\forall m, comb - m, d, comb - d)(produz(comb - m, comb - d) \cap desvio(d, *, comb - d) \cap monitoracao(m, *, comb - m) \leftrightarrow (\exists pp, comb - pp)revisa(comb - m, comb - pp) \cap pProjeto(pp, * comb - pp))$	B18

5. Instanciação

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 sobre as evidências coletadas durante a pesquisa de campo do processo de cada empresa, 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. Desta forma, os produtos de trabalho coletados durante a entrevista são utilizados como instâncias da ontologia. Por limitações de espaço, este trabalho apresenta uma parte da instanciação desta pesquisa, relacionado à instanciação do monitoramento do projeto na Empresa-A e Empresa-B, como apresentado no Quadro 1.

Quadro 1 – Instanciação do Monitoramento do Projeto na Empresa-A e Empresa-B

Empresa-A	Empresa-B
<i>pProjeto(Plano do Projeto, Plano do Projeto, ppPlanoProjeto)</i>	<i>pProjeto(Plano do Projeto, Plano do Projeto, ppPlanoProjeto)</i>
<i>cronograma(Plano do Projeto, Cronograma, ppCronograma)</i>	<i>cronograma(Plano do Projeto, Cronograma, ppCronograma)</i>
<i>possui(ppCronograma, ppMarcos)</i>	<i>possui(ppCronograma, ppMarcos)</i>
<i>possui(ppCronograma, ppPontosControle)</i>	<i>possui(ppCronograma, ppPontosControle)</i>
<i>mMarcos(Plano do Projeto, Marcos, ppMarcos)</i>	<i>mMarcos(Plano do Projeto, Marcos, ppMarcos)</i>
<i>mPeriodico(Plano do Projeto, Pontos de Controle, ppPontosControle)</i>	<i>mPeriodico(Plano do Projeto, Pontos de Controle, ppPontosControle)</i>
<i>revisa(ppMarcos, ppPlanoProjeto)</i>	<i>revisa(ppMarcos, ppPlanoProjeto)</i>
<i>produz(ppMarcos, rmDesvios)</i>	<i>produz(ppMarcos, fbIssue)</i>
<i>desvio(Relatório de Monitoração do Projeto, Desvios, rmDesvios)</i>	<i>desvio(Ferramenta de Bugtraking, Issue, fbIssue)</i>
<i>revisa(ppPontosControle, ppPlanoProjeto)</i>	<i>revisa(ppPontosControle, ppPlanoProjeto)</i>

<i>produz(ppPontosControle, arDesvios)</i>	<i>produz(ppPontosControle, fbIssue)</i>
<i>desvio(Ata de Reunião, Desvios, arDesvios)</i>	<i>desvio(Ferramenta de Bugtraking, Issue, fbIssue)</i>

Ao final da instanciação, alguns pontos foram observados. Nas três empresas o planejamento dos monitoramentos em marcos e pontos de controle é estabelecido no cronograma do projeto. Esses monitoramentos são executados e um conjunto de desvios é gerado. Assim, os axiomas B7 ao B18 são válidos.

Um ponto que deve ser comentado refere-se à cardinalidade entre “Monitoracao” e “Desvio”, onde foi modelada como uma relação de um para um ou mais (1-n). Neste sentido, baseado no universo de possibilidades, é possível que durante um monitoramento nenhum desvio seja detectado, ou seja, uma cardinalidade 0-n. Entretanto, como em avaliações oficiais é necessária a existência do registro de pelo menos um desvio, então optou-se em utilizar a cardinalidade 1-n.

A instanciação da estrutura do planejamento do projeto apenas define os produtos de trabalho utilizado nas três empresas. Dessa forma, apenas os predicados dos conceitos e o relacionamento de herança foram instanciados. Assim, para verificar a validade dos axiomas A1 ao A13, utilizaram-se as recomendações constantes nos modelos MR-MPS-SW e CMMI-DEV. Esses modelos descrevem que deve ser definido um plano geral do projeto no qual integra os seus planos específicos.

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 GPR do MPS.BR e CMMI, buscando modelá-la baseada em processos de empresas de software oficialmente avaliadas. Assim, a ontologia ficou estruturada de forma mais genérica possível, sendo capaz de abranger o maior número possibilidade de implementações.

Deve-se salientar que a forma como as práticas sugeridas nos modelos de qualidade serão implementadas estão sujeitas a subjetividade do consultor implementador e a realidade/cultura da empresa. Desta forma, surgindo mecanismos de evidências incomuns, tais como as mídias visuais (fotos) e audiovisuais (gravação de áudio ou vídeo). Assim, pode-se surgir a necessidade de realizar evoluções na ontologia para contemplar as referidas evidências. Entretanto, uma das características presentes nas ontologias é sua capacidade evolutiva.

Com base nos resultados obtidos a partir das instanciações dos axiomas da ontologia desta pesquisa, pode-se notar a ampla complexidade em 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 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. Esta ferramenta possuirá um banco de dados de ativos de processo capazes de contemplar cada resultado esperado. A ferramenta facilitará o manuseio da ontologia, pois tornará o uso dos axiomas mais transparentes para o usuário. Assim, 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.

8. Agradecimentos

Este trabalho está inserido no escopo do Projeto SPIDER e recebeu o apoio financeiro do CNPq a partir de bolsa de estudo de Mestrado.

Referências Bibliográficas

- Almeida, M. B. (2006), “Um modelo baseado em ontologias para a representação da memória organizacional”. Tese de Doutorado – UFMG, Belo Horizonte – MG.
- Carnap, R. “Introduction to Symbolic Logic and Its Application”, Dover Publications, Inc., New York, 1958.
- Colenci Neto, A. (2008), “Proposta de um Modelo de Referência para Desenvolvimento de Software com Foco na Certificação do MPS.BR”. Tese de Doutorado – Instituto Alfredo Luiz de Coimbra - UFRJ/COPPE, Rio de Janeiro – RJ.
- Duarte, K. C., Falbo, R. A. (2000), “Uma Ontologia de Qualidade de Software”. WQS2000 – Workshop de Qualidade de Software. João Pessoa.
- Falbo, R. A. (1998), “Integração de Conhecimento em um Ambiente de Desenvolvimento de Software”, Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro - RJ.
- Guizzardi, G., Falbo, R. A., Guizzardi, R. S. S. (2008), “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.
- Huang, D. B., Zhang, W. (2010), “CMMI in Medium & Small Enterprises: Problems and Solutions”. The 2nd IEEE International Conference on Information Management and Engineering (ICIME). Chile.
- Morgado, G. P., *et al.* (2007), “Práticas do CMMI como Regras de Negócio”. São Paulo: Produção.
- Schots, N. C. L., *et al.* (2011), “Lições Aprendidas em Implementações de Melhoria de Processos em Organizações com Diferentes Características”. Anais do VII Workshop Anual do MPS.BR – WAMPS, Campinas – SP.
- SEI – Software Engineering Institute (2010), “Capability Maturity Model Integration for Development – CMMI-Dev. Versão 1.3”. Carnegie Mellon.
- Sharinfloo, A. A., *et al.* (2008), “An Ontology for CMMI-ACQ Model”. 3rd International Conference on Information and Communication Technologies: From Theory to Applications. Damascus.
- SOFTEX – Associação para Promoção da Excelência do Software Brasileiro (2011), “Melhoria do Processo de Software Brasileiro – Guia de Implementação - Parte 1: Fundamentação para Implementação do Nível G do MR-MPS”.
- SOFTEX – Associação para Promoção da Excelência do Software Brasileiro (2012a), “Melhoria do Processo de Software Brasileiro – Guia Geral MPS de Software”.
- SOFTEX – Associação para Promoção da Excelência do Software Brasileiro (2012b), “Melhoria do Processo de Software Brasileiro – Guia de Implementação de Software – Parte 11:2012”.
- Soydan, S. H., Kokar, M. M. (2006), “An OWL Ontology for Representing the CMMI-SW Model”. 2nd International Workshop on Semantic Web Enabled Software Engineering. Athens-GA.
- Valente, A. (1995), “Legal Knowledge Engineering - A Modelling Approach”. IOS Press.
- Yoshio, E., Oliveira, S. (2014) “Uma Ontologia que Estabelece os Relacionamentos de Dependência entre as Práticas de Gerência de Requisitos presentes no CMMI-DEV Nível 2 e MR-MPS-SW Nível G”. Revista Abakós. Em publicação.