

Elicitação de requisitos orientada por ontologias de domínio

Requirements elicitation oriented by domain ontologies

Rafael Wobeto Motta^a, Edilson Ferneda^b, Fábio Bianchi Campos^c,
Viviane Judith Conte de Oliveira Pinto^d

^a *rafbase@gmail.com*

^b *eferneda@gmail.com*

^c *fabio.bianchi.campos@gmail.com*

^d *vivianejudith@hotmail.com*

Resumo: Levantamento de Requisitos é um processo dedicado a compreender as necessidades de clientes e usuários, e transformar estas necessidades em projetos de desenvolvimento de sistemas. Para capturar tais necessidades, o engenheiro de requisitos precisa interagir com pessoas de negócio das mais diversas áreas. Essa interação requer intensa comunicação e é bastante afetada pelas diferenças de vocabulário e interpretação que existe entre a equipe de negócio e a equipe de desenvolvedores do sistema. O uso de métodos de Arquitetura da Informação e de ontologias trazem novas alternativas para tratar tais problemas. O método de Arquitetura da Informação MAIA mostrou-se aderente ao Levantamento de Requisitos, possibilitando uma nova visão sobre o processo. Foram identificados estudos que abordam o uso de ontologias e do método MAIA que podem ser usados para apoiar o levantamento de requisitos. O objetivo principal deste trabalho é propor o modelo de Elicitação de Requisitos em Ciclos (ERC), criando uma nova abordagem para a especificação de sistemas, que garanta um documento de requisitos completo, consistente e não ambíguo, e que permita suporte automatizado de conteúdo. Esta abordagem é baseada em um conjunto de melhores práticas que visam melhorar o suporte a comunicação entre as equipes e, consequentemente, o processo de Levantamento de Requisitos como um todo.

Palavras chave: Engenharia de Requisitos, Levantamento de Requisitos, Arquitetura da Informação, Comunicação, Ontologia.

Abstract: The Requirements Elicitation is a process dedicated to understanding the needs of customers and users, and transform these needs into system development projects. To capture such needs the requirements engineer needs to interact with business people and the most diverse areas. This interaction requires intense communication, and is greatly affected by differences in vocabulary and interpretation that exists between the business team and the team of system developers. However, the use of Architecture Information methods and ontologies bring new alternatives to treat such problems. The Architecture Information method MAIA appears to be adherent to the Requirements Elicitation, providing new insight into the process. Therefore, studies have been identified which address the use ontologies and MAIA method that can be used to support the requirements specification. The main objective of this study is to propose the requirements elicitation in cycles (REC) model, creating a new approach to the system specification, which ensures a complete, consistent and unambiguous requirements document, and which allows automated content support. This approach is based on a set of best practices aimed at improving support for communication between teams and consequently the requirements survey process as a whole.

Keywords: Requirements Engineering, Requirements Survey, Information Architecture, Communication, Ontology.

1. Introdução

Os sistemas computacionais estão cada vez mais presentes nas estratégias de negócio das empresas, aumentando as exigências e expectativas sobre seu processo de desenvolvimento. Neste cenário, sabe-se que, mesmo um software aparentemente simples, pode envolver comportamentos complexos e surpreen-

dentes (Jackson, 2010). Para enfrentar esta nova realidade, nos últimos anos houve um grande crescimento no uso dos métodos ágeis pela indústria de software (Vallerão & Roses, 2013). No entanto, independentemente do método de desenvolvimento adotado, a aquisição de requisitos é considerada como o maior gargalo no desenvolvimento de sistemas.

Depara-se cada vez mais com projetos de

Tecnologia da Informação (TI) envolvendo uma quantidade e complexidade de requisitos cada vez maiores. Além disso, é comum que os stakeholders concernentes estejam geograficamente distribuídos e tenham diferentes formações. A consequência são problemas devido a choques quanto ao vocabulário (conceitos ou terminologias) (Farfeleder et al., 2011), o que cria uma barreira de comunicação e entendimento entre os stakeholders (Li et al., 2011).

Compreender os termos do domínio do negócio do cliente é fundamental, pois uma das principais medidas do sucesso de um produto de software é o grau com que ele atende aos objetivos e requisitos para os quais foi construído (Martins, Nardi & Falbo, 2006). Além disso, a qualidade do software é medida verificando se o projeto foi concluído a tempo, dentro do orçamento e se cumpre os requisitos do usuário (Soomro et al., 2014).

O desenvolvimento de sistemas tem o desafio de juntar o formalismo necessário para especificar um software com a natureza informal do ser humano e do mundo real, o que, na prática, envolve uma atividade de descrição e raciocínio que raramente é totalmente formal (Jackson, 2013).

A comunicação deve ser clara e livre de ambiguidades para evitar que processos complexos se tornem incompreensíveis (Araujo, Simanski & Quevedo, 2012), pois, problemas não tratados nesta etapa, tendem a se agravar ao longo da fase de desenvolvimento (Malanovicz & Brodbeck, 2010). Por isso, é essencial que as interações informais sejam precedidas de algum formalismo, para ser possível moldar a estrutura de todo o conjunto de artefatos de desenvolvimento, orientar o processo de aprendizagem e compreensão, e documentar as propriedades e comportamentos dos domínios de problemas (Jackson, 2013).

Soares e Moura (2015) propõem uma metodologia com uso de padrões linguísticos para criar uma semântica do domínio, e subsidiar a construção de um documento de requisitos completo, consistente e não ambíguo, e que permita suporte automatizado de conteúdo. Os autores afirmam que a definição da semântica, além de trazer bons resultados, é relativamente simples de implementar.

Neste contexto, surge o problema do presente trabalho: Como reduzir o impacto dos problemas de comunicação entre os stakeholders na qualidade dos requisitos gerados du-

rante o processo de especificação de sistema?

Existem ferramentas que auxiliam na construção da documentação de requisitos de sistemas, tais como *Open Source Requirements Management Tool*¹, *Rational RequisitePro*² e *HP Quality Center*³. No entanto, esses aplicativos buscam apenas fornecer uma interface para a estrutura básica necessária, não se preocupando com os demais aspectos do processo (Ito et al., 2011).

O objetivo deste trabalho é propor um modelo capaz de reduzir o impacto dos problemas de comunicação entre os *stakeholders* na qualidade dos requisitos gerados, através da formação de uma estrutura semântica a partir dos requisitos da especificação, que permita verificar se o documento de requisitos está completo, consistente e não ambíguo.

Acompanhar a qualidade durante o levantamento e análise dos requisitos é relevante, pois quanto mais cedo se mede a qualidade e se encontram e corrigem erros no produto, menor será o custo destas correções (Ramos et al., 2006). A Figura 1 representa a relação entre custo das mudanças e o tempo de projeto.

Ainda há carência de ferramentas no mercado com o objetivo de dar apoio às atividades de identificação e validação de requisitos junto aos clientes (Ito et al., 2011). Por outro lado, apesar da carência por ferramentas de suporte mais efetivas, a Engenharia de Requisitos (ER) é uma subdisciplina bastante madura em Engenharia de Software, ainda que um grande número de sistemas seja desenvolvido sem rigoroso controle de qualidade em seu estágio de planejamento (Capilla, Babar & Pastor, 2012). Os autores ressaltam que muitas vezes os atributos de qualidade são utilizados de maneira informal, justamente em decorrência da falta métodos e ferramentas que os suportem.

2. Referencial teórico

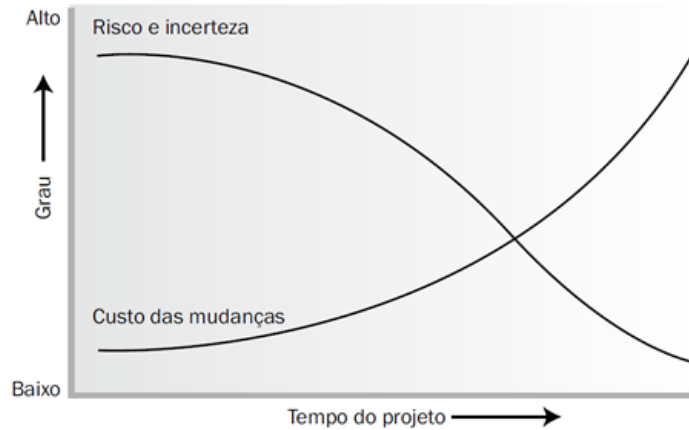
2.1. Engenharia de requisitos

A Engenharia de Requisitos (ER) tem por função compreender, documentar, comunicar

¹ <http://sourceforge.net/projects/osrmt/>

² http://www-01.ibm.com/software/awdtools/reqpro/features/?S_CMP=rnav

³ <http://www8.hp.com/br/pt/software-solutions/quality-center-quality-management/>

Figura 1. Custos das mudanças e correções de erros com base no tempo decorrido do projeto

Fonte: PMI (2013)

e implementar as necessidades dos clientes (Farfeleder et al., 2011). A construção de uma especificação de requisitos é um processo iterativo de quatro etapas (Jin et al., 2003): (i) determinação das necessidades dos clientes; (ii) definição dos requisitos da aplicação; (iii) validação da especificação de requisitos; e (iv) demonstração da proposta de sistema alvo para a obtenção da aprovação dos clientes. Apenas após os clientes concordarem com o sistema alvo proposto as fases de design e programação são iniciadas.

Os requisitos dividem-se em funcionais (funcionalidades e serviços do sistema) e não funcionais (características de qualidade e restrições do sistema) (Sommerville, 2011), sendo o objetivo do levantamento de requisitos identificar onde, como e com quem serão coletados os requisitos de um software (Costa, Zoucas & Alves, 2012). Por isso, é importante que se saiba a origem, os *stakeholders* e os responsáveis por cada requisito (Nardi & Falbo, 2006).

Os *stakeholders*, ou partes interessadas, são indivíduos ou organizações envolvidos em um projeto de software ou cujos interesses são afetados *pele* projeto (Hofmann & Lehner, 2001). É preciso considerar que estes vários atores envolvidos no processo de levantamento de requisitos possuem diferentes pontos de vista, modelos mentais e expectativas (Góes, Silva & Barros, 2013). Isso faz com que muitos *stakeholders* tenham dificuldade em articular claramente seus requisitos e suas intenções em relação ao projeto (Al-Hothali; Al-Zubaidi; Subbarao, 2012).

A elicitação de requisitos é uma das partes mais difíceis e mais críticas do desenvolvimento de software (Li et al., 2011). Por isso, é

preciso empenho para facilitar a sua validação e reutilização (Chikh et al., 2011). E, como todas as atividades podem influenciar na qualidade dos requisitos, este estudo considera a utilização de métricas que atuem na avaliação dos requisitos em qualquer momento do processo de desenvolvimento de produtos de TI, inclusive após a fase de levantamento de requisitos.

O Documento de Especificação de Requisitos de Software (DERS) é uma declaração oficial do que é exigido do sistema, servindo como um acordo entre o cliente e a equipe de desenvolvimento (Soomro et al., 2014). Por isso, o documento deve estar estruturado de forma a permitir um entendimento claro por parte dos desenvolvedores sobre o comportamento esperado do sistema (Jackson, 2010). A padronização do documento de requisitos facilita a comunicação entre os *stakeholders* e melhora sua consistência e inteligibilidade (Soares & Moura, 2015).

Nas metodologias ágeis de desenvolvimento, a documentação é organizada de forma a acompanhar as mudanças constantes do processo de especificação de requisitos (Semedo, 2012), com desenvolvimento cíclico, comunicação e *feedbacks* constantes, permitindo identificar problemas e soluções de forma mais rápida e evitar desperdícios (Bassi Filho, 2008).

A volatilidade dos requisitos é uma questão presente na Engenharia de Software, podendo afetar o cronograma, e diminuir a qualidade dos serviços prestados (Shaban-Nejad et al., 2009). Por isso, é necessário haver rastreabilidade dos mesmos, para que haja controle das alterações e versões de um requisito e o

acompanhamento de seu estado. Isto só é possível quando há ligações explícitas entre requisitos e outros elementos do processo de software (Nardi & Falbo, 2006).

Para implementar a rastreabilidade, é preciso uma estrutura semântica, com vocabulários específicos, contendo informação relevante e permitindo a interligação destas informações com o domínio (Chikh et al., 2011). Tal estrutura viabiliza a análise de consistência e completude, que, aliada ao uso de atributos de qualidade e técnicas de rastreabilidade automatizadas, podem eliminar conflitos e cooperação falsamente identificados de forma eficiente (Egyed & Grunbacher, 2004).

A categorização dos requisitos permite distinguir uma variedade de objetos, permitindo que suas interações sejam capturadas e compostas. Interações equivalentes podem ser diferenciadas, identificando padrões de interação entre objetos, assim como algumas invariantes em sua ação. Desta forma, um objeto complexo pode ser decomposto em componentes básicos (Ehresmann & Vanbreemsch, 2006).

2.2. Ontologias e engenharia de requisitos

Ontologia é uma conceituação básica acerca de um universo de discurso (Martins, Nardi & Falbo, 2006). Pode também ser definida como um vocabulário específico usado para descrever certa realidade, com um conjunto de decisões explícitas, definindo de forma rigorosa o significado pretendido para o vocabulário (Nardi & Falbo, 2006). Existem ontologias para vários domínios (E-commerce, engenharia, turismo, etc.), usadas, entre outras aplicações, para aumentar a interoperabilidade entre as fontes (Khouri et al., 2012).

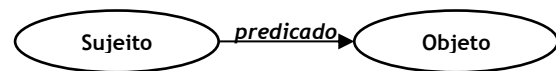
As ontologias são utilizadas para compartilhamento de vocabulários comuns, fornecendo uma estrutura para organizar o conhecimento (Shaban-Nejad et al., 2009). Ontologias têm um escopo determinado e buscam responder questões específicas (questões de competência).

Considerando que os requisitos são o foco da ER, é fundamental que haja uma compreensão clara do requisito em si e de como ele se relaciona com outros elementos (Nardi & Falbo, 2006). As tecnologias ontológicas provêm um meio eficiente para dar suporte a estas tarefas (Li et al., 2011), sendo cada vez maior o interesse pela aplicação de ontologias de domínio

no âmbito da ER (Chikh et al., 2011).

Pode-se representar uma ontologia através do RDF (*Resource Description Framework*) (Honkola et al., 2010). O RDF é um *framework* para representar informações, onde cada item é definido no formato de triplas (sujeito, predicado, objeto), ou ainda em triplas (sujeito, propriedade, valor) (Neumann & Weikum, 2008). Além disso, o RDF facilita enormemente a ligação entre diferentes ontologias, fortalecendo a interoperabilidade (Honkola et al., 2010). A Figura 2 representa o conjunto formado por cada tripla, formando um grafo RDF.

Figura 2: Um grafo com dois nodos (Sujeito e Objeto) e um arco conectando eles (Predicado)



O uso de ontologias tem se popularizado nos ambientes de desenvolvimento, pois seu objetivo é justamente promover um entendimento comum e compartilhado sobre um domínio, que pode ser comunicado entre pessoas e sistemas de aplicação (Nardi & Falbo, 2006).

A seguir são apresentadas algumas iniciativas que aplicam ontologias nos processos de ER.

2.2.1 Espaço conceitual de três camadas

O Espaço conceitual de três camadas (Jin et al., 2003) sugere a adoção de três camadas para o levantamento de requisitos: a camada Conceito Meta-Nível, a camada Conceito de Domínio e a camada Conceito de Aplicação. Cada camada contém um conjunto de conceitos e um conjunto de associações de conceitos, inclusive com conexões entre diferentes camadas, formando uma hierarquia de conceitos. A camada Conceito Meta-Nível representa a ontologia empresarial e fornece o consenso empresarial genérico para ajudar os analistas a entender as declarações dos clientes.

Este modelo permite analisar a consistência e a completude da arquitetura de informação de requisitos. Essa consistência pode ser verificada por meio da análise da hierarquia formada pelos pares de conceitos entre os diferentes níveis. Problemas de completude podem ser identificados analisando-se os nós sem correspondência (Jin et al., 2003).

2.2.2 Ontologia de Requisito de Software

A Ontologia de Requisito de Software (Nardi

& Falbo, 2006) utiliza o método SABiO (*Systematic Approach for Building Ontologies*) para propor uma ontologia de requisitos de software. A construção da ontologia inicia-se pela especificação de requisitos, listando as questões de competência levantadas:

- QC₁ O que é um requisito?
- QC₂ Qual a natureza de um requisito?
- QC₃ Quais são os requisitos de um projeto de software?
- QC₄ Para que módulos do sistema um requisito é alocado?
- QC₅ Quais os responsáveis por um requisito?
- QC₆ Quais os interessados (*stakeholders*) em um requisito?
- QC₇ Qual a origem de um requisito?
- QC₈ Qual o estado de um requisito?
- QC₉ Como um determinado requisito é decomposto em outros requisitos?
- QC₁₀ Que requisitos são dependentes de um determinado requisito?
- QC₁₁ Que requisitos são conflitantes entre si?
- QC₁₂ O que é uma especificação de requisitos de software (ERS)?
- QC₁₃ Que artefatos descrevem, modelam ou implementam um requisito?
- QC₁₄ Como gerenciar mudanças nos requisitos e nos artefatos a eles relacionados?
- QC₁₅ Como avaliar a qualidade em requisitos?

As questões de competência listadas acima são divididas em quatro tópicos (Nardi & Falbo, 2006): (i) Definição e Taxonomia de Requisitos (questões 1 a 4 e 12); (ii) Aprovação e Interesse em Requisitos (questões 5 e 6); (iii) Gerência de Requisitos (questões 7 a 11, 13 e 14); (iv) Qualidade em Requisitos (questão 15).

De acordo com Nardi e Falbo (2006), a ontologia de requisitos de software pode servir de base para o desenvolvimento de ferramentas de apoio a ER em um ambiente de desenvolvimento de software baseado em ontologia. Neste ambiente, segundo Falbo et al. (2004), as ontologias são usadas para estabelecer uma forma padrão de comunicação entre agentes.

2.2.3 ReqODE

O ReqODE é uma ferramenta de apoio à ER, com funcionalidades de controle e rastreabilidade de requisitos, construída com base em uma ontologia de requisitos (Martins, Nardi & Falbo, 2006).

A ferramenta ReqODE permite uma rastreabilidade abrangente dos requisitos, pois permite identificar requisitos dependentes ou conflitantes, associações com casos de uso, com artefatos do contexto do projeto e com o contexto de criação do requisito (Martins, Nardi & Falbo, 2006).

Para facilitar o gerenciamento de requisitos, cada projeto tem seu escopo dividido em módulos, os requisitos são alocados nestes módulos, e recursos humanos são indicados como responsáveis e interessados. A partir dos elementos anteriores, é criado um Documento de Rastreabilidade que pode ser utilizado para gerenciar mudanças. Assim, para qualquer alteração em um requisito, é possível identificar os demais elementos afetados, possibilitando analisar o impacto da alteração.

2.2.4 A proposta de Shaban-Nejad et al

O estudo de Shaban-Nejad et al (2009) propõem a utilização de ontologias em um sistema Web de gerenciamento de informações de laboratório (LIMS). Utiliza-se uma abordagem baseada em agentes para análise e gerenciamento de requisitos voláteis e dinâmicos (Shaban-Nejad et al., 2009).

O modelo de categorização criado representa os requisitos com base em suas dependências e interdependências, o que permite a detecção de mudanças de requisitos. Tais mudanças são monitoradas para verificar a coerência dos requisitos com o quadro categorial existente no contexto do sistema.

2.2.5 O uso de clichês

Outra prática adotada é o clichê (*boilerplate*), uma abordagem que utiliza uma sequência de atributos e elementos de sintaxe fixa. Um clichê comum é “< sistema > deve < ação >”, sendo < sistema > e < ação > atributos e “deve” um elemento fixo da sintaxe. Esta abordagem visa minimizar a quantidade de clichês a serem utilizados, mas mantendo uma elevada flexibilidade (Farfeleder et al., 2011).

Segundo Farfeleder et al. (2011), uma ferramenta de apoio precisa atender a dois pontos para acelerar o processo de especificação de requisitos de alta qualidade: (i) o sistema deve propor automaticamente, ao menos, partes dos requisitos, utilizando informações provenientes de uma ontologia de domínio, e (ii) o sistema deve explorar as relações e axiomas da ontologia de domínio.

2.3. Arquitetura da informação

A Arquitetura da Informação (AI) é uma disciplina que estuda e modela espaços informacionais utilizando a lógica e a linguagem (Siqueira, 2008), e refere-se a práticas e estudos engajados em resolver problemas de acesso e uso das informações (Resmini & Rosati, 2012). Pode ser definida como o projeto estruturado de um ambiente de compartilhamento de informações (Information Architecture Institute, 2015).

A AI visa estruturar, organizar, qualificar, localizar e gerenciar as informações de uma organização (Moresi, Ramos & Prado, 2010), possuindo uma relação bastante próxima com várias outras áreas (Trindade, Lima-Marques & Souza, 2009). Os conceitos da AI são aderentes aos processos de ER, uma vez que ambas as áreas visam a construção estruturada de espaços de informação para subsidiar os objetivos de uma organização. Por isso, para a construção do modelo proposto neste trabalho, será utilizado como referência o método de arquitetura da informação Maia.

O Método de Arquitetura de Informação Aplicada (MAIA) foi proposto a partir de investigações fenomenológicas (Lima-Marques, 2007 apud Costa, 2009), com o objetivo de criar um método voltado ao tratamento das informações de organizações (Costa, 2009). Seu uso é indicado para auxiliar na criação de um processo de desenvolvimento de sistema (Siqueira, 2012). O Maia tem seus elementos divididos em quatro momentos: o Escutar, o Pensar, o Construir e o Habitar. Estes elementos são ordenados de forma a permitir o desenvolvimento de um estado específico e organizado em um espaço de informação (Costa, 2009).

O método possui um caráter cíclico, mas não fechado, uma vez que não é possível delimitar os limites de cada momento. A ideia de desenvolvimento espiral do MAIA sugere que o resultado de um ciclo apresenta evidências que implicam em um novo ciclo de investigações. Segundo Melo (2010), há uma relação bidirecional entre o MAIA e ontologias, onde uma ontologia delimita o contexto inicial, que vai sendo alterado durante os momentos do MAIA. Com a ontologia pode-se ter uma visão geral do lugar onde ocorre o processo, das pessoas que o compõem, dos elementos destes processos e de sua finalidade (Melo, 2010).

2.4. Um modelo de aplicação prática de engenharia de requisitos em sistemas web

Uma primeira proposta de modelo de aplicação prática para ER foi proposta pelo autor deste trabalho (Motta, 2012). Este modelo foi construído com foco nos sistemas baseados em Web, utilizando princípios da ER e do desenvolvimento ágil.

O método utilizado pelo modelo é a Programação Extrema, ou XP. Seus valores fundamentais de comunicação, simplicidade, *feedback*, coragem e respeito são as referências para as funcionalidades do aplicativo. Apesar de não gerarem uma documentação tão detalhada, os métodos ágeis fornecem documentação suficiente para a gestão do sistema, e em um tempo bastante reduzido.

Como as partes envolvidas acompanham e interagem com o projeto através de um mesmo ambiente, o *feedback* é constante entre todos. A estrutura de análise gerada pelo modelo objetiva facilitar a tomada de decisões, pois com informações claras e organizadas a equipe tem mais facilidade de tomar ou mudar suas decisões.

3. Metodologia

Este trabalho faz uso da pesquisa descritiva, cujo propósito é expor as características de um determinado fenômeno (Moresi, 2003), podendo ser classificada como de abordagem qualitativa. No que diz respeito à sua natureza, esta pesquisa pode ser considerada como uma pesquisa aplicada, pois visa produzir conhecimento que subsidie uma aplicação prática do modelo proposto à solução de problemas específicos, envolvendo verdades e interesses locais (Moresi, 2003).

Foram coletados e analisados dados sobre a estrutura semântica formada pelos requisitos de uma especificação de *software*. A aplicação e avaliação do modelo foi feita em cenários reais de levantamento de requisitos, dentro da empresa onde o autor trabalha, uma grande instituição financeira brasileira, em projetos desenvolvidos tanto localmente quanto à distância, envolvendo diversas equipes e departamentos.

O modelo proposto utiliza o método MAIA para desenvolver uma ontologia de domínio específica para cada projeto, que pode ser conectada a uma ontologia de domínio da or-

ganização, criando um vocabulário comum e consensual visando reduzir o impacto da comunicação entre as diferentes equipes envolvidas no processo de levantamento de requisitos de sistema.

O processo de levantamento de requisitos dos projetos de desenvolvimento foi estruturado na forma de um modelo de elicitação de requisitos em ciclos, onde cada ciclo gerou informações para alimentar o próximo ciclo, seguindo a proposta do MAIA. Ao longo de cada ciclo foram aplicados os momentos do MAIA (Escutar, Pensar, Construir e Habitar) para o tratamento das informações, conforme proposto por Costa (2009), utilizando o método para auxiliar o processo de desenvolvimento, seguindo a indicação de Siqueira (2012), de forma a construir uma ontologia de domínio específica.

Os ciclos do modelo criam uma hierarquia de conceitos, como no Espaço conceitual de três camadas, o que permite analisar a consistência e a completude da arquitetura de informação dos requisitos (JIN et al., 2003). A organização de forma hierárquica permite rastrear mudanças nos requisitos (SHABAN-NEJAD et al., 2009). Estes requisitos irão compor tanto o documento de requisitos quanto a ontologia de domínio específica. Uma vez construída, esta ontologia de domínio específica é conectada à ontologia de domínio da organização, de forma que possa ser reutilizada em projetos futuros.

O levantamento de requisitos dos projetos foi feito com a utilização de técnicas já tradicionais: (i) entrevista; (ii) questionário; (iii) análise documental; (iv) reuniões. A técnica a

ser usada irá depender da necessidade de cada projeto e de cada etapa. Os requisitos coletados foram cadastrados no software de especificação de sistemas construído pelo autor, desenvolvido seguindo o modelo conceitual proposto neste trabalho.

Para avaliar o modelo, foram propostas análises pelo autor em relação a reusabilidade, rastreabilidade, consistência e completude. A reusabilidade só poderá ser observada no segundo e terceiro cenários, pois suas especificações são construídas a partir de um contexto inicial que reutiliza as definições a serem construídas no primeiro cenário. A rastreabilidade, a consistência e a completude foram verificadas através da análise dos artefatos gerados. A rastreabilidade, pela existência de relacionamentos explícitos entre os requisitos e demais elementos. A consistência, pela existência de relacionamentos entre requisitos de diferentes níveis conceituais. A completude, pela existência de relacionamentos que permitam apontar requisitos com relacionamentos incompletos.

A avaliação foi documentada por meio de uma métrica baseada em *check-list*, onde cada especificação de requisitos foi analisada por critérios de avaliação, marcados como “sim” ou “não” quanto ao atendimento do critério. A escolha desta métrica baseia-se no trabalho de Ito et al. (2011), que afirmam que essa técnica pode ser aplicada sem a necessidade de intervenção de um mediador. A avaliação e os critérios utilizados, com base no referencial teórico, são apresentados na Quadro 1.

O uso da análise de consistência e completude, aliada a técnicas automatizadas de ras-

Quadro 1. Aspectos e critérios de avaliação

Aspecto	Critério	Avaliação
Reusabilidade	Reutilizou conceitos da ontologia?	(sim/não)
	É possível propor automaticamente requisitos a partir da ontologia? (Farfeleder et al., 2011)	(sim/não)
	Explora as relações e axiomas da ontologia? (Farfeleder et al., 2011)	(sim/não)
Rastreabilidade	Há ligações explícitas entre requisitos e outros elementos? (Nardi & Falbo, 2006)	(sim/não)
	Há categorização dos requisitos? (Ehresmann & Vanbremeersch, 2006)	(sim/não)
	Um objeto complexo pode ser decomposto em componentes básicos? (Ehresmann & Vanbremeersch, 2006)	(sim/não)
Consistência	Há organização hierárquica dos requisitos? (Shaban-Nejad et al., 2009)	(sim/não)
	Permite análise de pares de conceitos entre os diferentes níveis? (Jin et al., 2003)	(sim/não)
Completude	Permite análise de nós sem correspondência? (Jin et al., 2003)	(sim/não)

Fonte: os autores

treabilidade, podem eliminar conflitos e cooperação falsamente identificados de forma eficiente (Egyed; Grunbacher, 2004). Assim, é possível avaliar a qualidade do requisito durante todo processo de elicitação de requisitos através de sua estrutura semântica, reduzindo o impacto da comunicação entre as equipes na qualidade da especificação.

Para verificar se o modelo permite implementar a rastreabilidade e avaliar a consistência e completude dos requisitos de forma satisfatória, a especificação gerada em cada projeto foi submetida às seguintes questões de competência da Ontologia de Requisitos de Software de Nardi e Falbo (2006): QC₃, QC₄, QC₅, QC₆, QC₇, QC₈, QC₉, QC₁₀, QC₁₁, QC₁₃, QC₁₄, QC₁₅.

As questões de competência QC₁ (O que é um requisito?), QC₂ (Qual a natureza de um requisito?) e QC₁₂ (O que é uma especificação de requisitos de software?), não foram adotadas por serem definições conceituais do requisito em si e do documento de requisitos, não contribuindo para a avaliação do modelo.

Partindo da premissa que as ontologias criam vocabulários que facilitam a comunicação entre pessoas (Falbo et al., 2004), a Ontologia de Requisitos de Software foi construída para facilitar a comunicação durante o processo de ER (Nardi & Falbo, 2006). Desta forma, assume-se que se é possível responder às questões de competência por meio da especificação de requisitos resultante do modelo, então se pode inferir que a proposta do modelo é capaz de reduzir o impacto dos problemas de comunicação entre as equipes de negócio e de TI na qualidade do levantamento de requisitos, uma vez que as questões de competência configuram as exigências de informação para criação de uma ontologia de requisitos de *software*.

Resultados esperados

Com esta pesquisa, pretende-se criar um modelo que possa auxiliar o processo de elicitação de requisitos a reduzir os problemas gerados pela falta de consenso entre as equipes de negócio e de TI. Espera-se que seja possível criar uma estrutura semântica que exija significados únicos, de forma a permitir verificar a qualidade dos requisitos em relação ao contexto no qual estão inseridos.

Como resultado da avaliação do modelo, espera-se que os conceitos definidos em um primeiro cenário possam ser reaproveitados em

outros cenários, confirmando a capacidade de reusabilidade. Também se espera que seja possível construir relacionamentos explícitos entre todos os elementos da especificação de requisitos, permitindo analisar a rastreabilidade, a consistência e a completude dos artefatos gerados.

Para confirmar a eficiência do modelo em implementar a rastreabilidade e avaliar a consistência e completude dos requisitos, espera-se que a especificação gerada em cada projeto consiga responder a todas as questões de competência selecionadas para avaliar o modelo. Esta confirmação permite que se afirme que o modelo é capaz de contribuir para redução do impacto dos problemas de comunicação entre as equipes na qualidade dos requisitos gerados.

4. Modelo proposto

O modelo proposto neste estudo é o modelo ERC (Elicitação de Requisitos em Ciclos), que incorpora ao processo de levantamento de requisitos a aplicação de conceitos de AI e uso de ontologias de domínio.

O modelo ERC utiliza o método MAIA (Costa, 2009) para estruturar e organizar seus ciclos de elicitação de requisitos, de forma que seja possível a criação e atualização de uma ontologia de domínio ao longo do processo de levantamento de requisitos.

O MAIA, com sua proposta de desenvolvimento espiral para criar um espaço de informação (Melo, 2010), mostra-se aderente aos princípios de desenvolvimento iterativo e incremental da metodologia ágil, base do primeiro modelo, e bastante utilizada atualmente pela indústria de software (Vallerão & Roses, 2013). Essa compatibilidade do MAIA com as rotinas do processo de desenvolvimento de softwares reforça sua capacidade de contribuir para a ER.

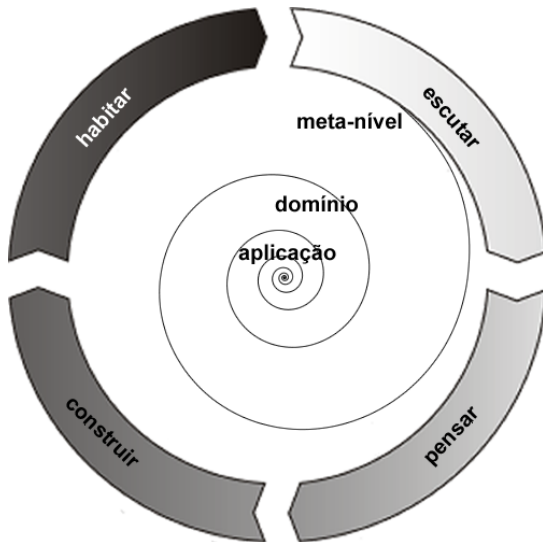
Além disso, o MAIA pode usar uma ontologia para delimitar o contexto inicial (MELO, 2010), o que pode ser de grande ajuda durante a especificação de requisitos. A Figura 3 ilustra a conexão criada entre os projetos de desenvolvimento e a ontologia de domínio da organização ao longo da especificação de um sistema.

A proposta de aprofundamento gradual dos conceitos envolvidos na especificação de sistemas baseia-se no Espaço Conceitual de Três Camadas (Jin et al., 2003), onde cada camada

representa conceitos que especializam os conceitos da camada anterior, de forma a permitir análises de completude e consistência.

A Figura 4 mostra a estrutura conceitual do modelo ERC, partindo dos conceitos mais genéricos, no meta-nível, até chegar aos conceitos mais específicos, na camada de aplicação.

Figura 4. Modelo ERC



Fonte: os autores

Ciclos de Elicitação

O modelo ERC segue as etapas da ER tradicional (Jin et al., 2003), porém, ao invés de seguir estas etapas de forma linear, o modelo possui comportamento cíclico, onde as etapas são percorridas inúmeras vezes, buscando a cada ciclo um maior aprofundamento dos conceitos adquiridos e analisados nos ciclos anteriores.

A cada novo projeto de desenvolvimento, os ciclos de elicitação são iniciados pela descoberta dos *stakeholders* do projeto, e definição do papel de cada um dentro do projeto. Este primeiro ciclo deve envolver apenas os principais *stakeholders* do projeto, como o patrocinador ou cliente, pois são eles que solicitam o projeto e possuem uma visão mais global das rotinas envolvidas. O início da coleta de requisitos visa um entendimento sobre o problema que o sistema deve resolver e qual seu contexto dentro da organização patrocinadora, além de identificar os produtos de software que devem ser desenvolvidos no projeto, descrevendo de forma sintética estes produtos.

As informações levantadas no primeiro ciclo irão nortear e subsidiar o próximo ciclo da especificação dos requisitos, e assim sucessi-

vamente, até que se obtenham requisitos suficientes para uma definição clara e correta da estrutura e do comportamento esperado para o sistema proposto.

Ao longo dos ciclos de elicitação, juntamente com as técnicas usadas no processo de levantamento de requisitos, é possível aplicar os momentos do MAIA para desenvolver a ontologia de domínio. Isto pode ser feito sem nenhum esforço adicional, registrando todas as informações em um ambiente compartilhado, de forma que os requisitos coletados sejam aproveitados simultaneamente para compor o documento de requisitos e a ontologia de domínio específica do projeto.

Logo no início da elicitação de requisitos já se pode iniciar também o desenvolvimento da ontologia de domínio. A Figura 5 ilustra como os requisitos são estruturados para que possam ser incluídos na ontologia.

Pode-se notar que, para o requisito ser inserido na ontologia de domínio, o mesmo é estruturado em triplas (sujeito, predicado e objeto). Isto permite identificar claramente as partes que compõe o requisito, e como estas partes se relacionam com outras definições da ontologia, formando uma rede semântica para dar significado a sujeitos e objetos dentro do contexto do domínio.

Tal estrutura assemelha-se ao uso de clichês (Farfeleder et al., 2011), sendo o predicado a parte fixa, e visa permitir a proposição automática de parte dos requisitos através da rede de conceitos construída na ontologia de domínio.

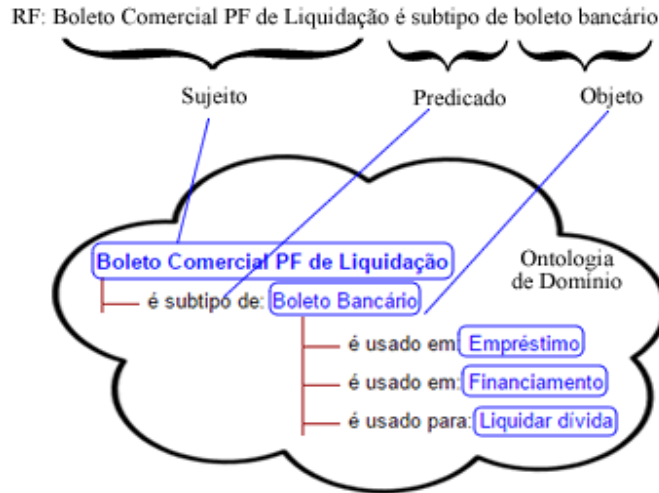
Ao final de cada ciclo, os dados resultantes devem servir de subsídio tanto para documentar a especificação do sistema alvo do projeto quanto para a ontologia de domínio criada ou alterada dentro do processo. Conforme novos ciclos são percorridos, aumenta o nível de detalhamento das especificações, com definições cada vez mais minuciosas sobre as rotinas de negócio abrangidas pelo escopo do projeto, e também sobre o papel de cada *stakeholder*.

Uma vez concluída a elicitação dos requisitos do sistema, seus conceitos passam a fazer parte da ontologia de domínio da organização, ficando disponíveis para serem reaproveitados em qualquer projeto de desenvolvimento futuro.

Documentação gerada

A documentação resultante do modelo ERC

Figura 5. Incorporação dos requisitos na ontologia de domínio



Fonte: os autores

organiza os requisitos do sistema conforme o esquema apresentado na Figura 6. É possível identificar os relacionamentos entre os elementos de forma explícita, e ver claramente toda composição do projeto e como seus elementos estão conectados. No esquema, a primeira linha de elementos representa os conceitos de meta-nível ou negócio, a segunda representa os conceitos de domínio, e a terceira e quarta linhas os conceitos de aplicação.

Analisando os relacionamentos do produto de software com os demais elementos, que possuem diversos níveis conceituais, nota-se claramente o suporte fornecido pelo modelo à

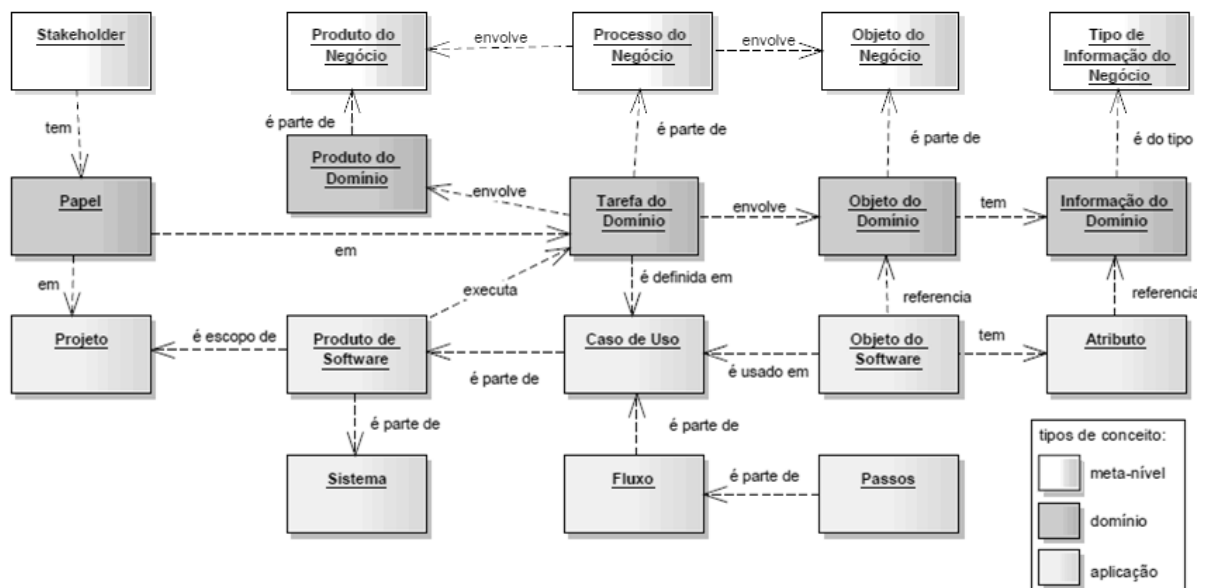
análise de consistência.

Como os elementos dos diferentes níveis conceituais possuem relacionamentos explícitos, é possível implementar a rastreabilidade dos requisitos e analisar a consistência e completude dos mesmos.

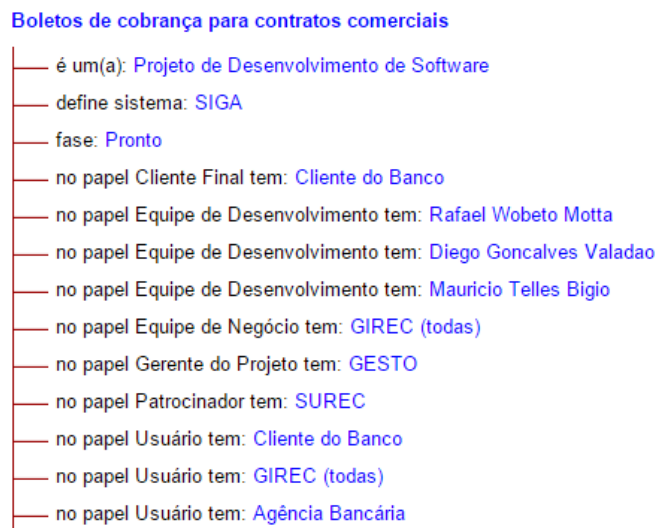
5. Aplicação do modelo

O modelo ERC foi aplicado em cenários reais de projetos de desenvolvimento de um grande banco brasileiro, na área de recuperação de crédito. Estes projetos envolvem a matriz do banco, suas filiais e agências bancárias. Será apresentada apenas uma parte dos dados gerados, de modo a deixar a análise do modelo

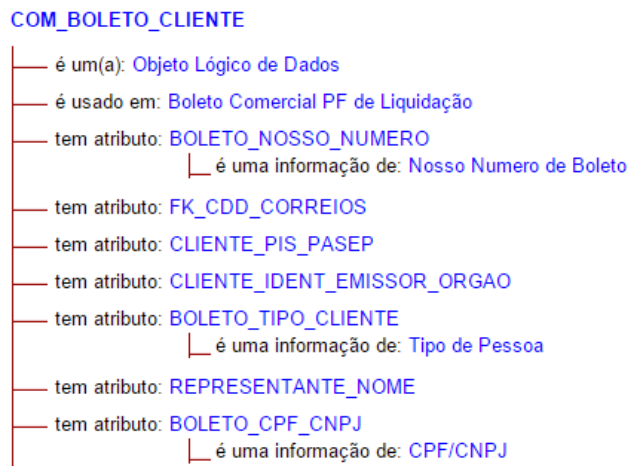
Figura 6: Estrutura do documento resultante do modelo ERC



Fonte: os autores

Figura 7. Ferramenta ERC - triplas ontológicas das informações iniciais do projeto

Fonte: os autores

Figura 8. Ferramenta ERC - triplas ontológicas da base de dados do produto

Fonte: os autores

mais didática.

5.1. Primeiro cenário: boletos de cobrança para contratos comerciais

O primeiro cenário envolve o desenvolvimento de uma ferramenta totalmente nova, para ser acessada na Intranet da empresa, tendo o envolvimento de empregados da organização na formação das equipes de negócio e de TI. Vale ressaltar que nenhum dos empregados da equipe de TI havia trabalhado antes com ferramentas de emissão de boletos.

As informações coletadas com a equipe de negócio foram cadastradas na ferramenta ERC, sendo tratadas nos momentos do MAIA. Os dados obtidos a respeito das informações iniciais do projeto são organizados e representados como triplas, conforme mostrado na Figura 7.

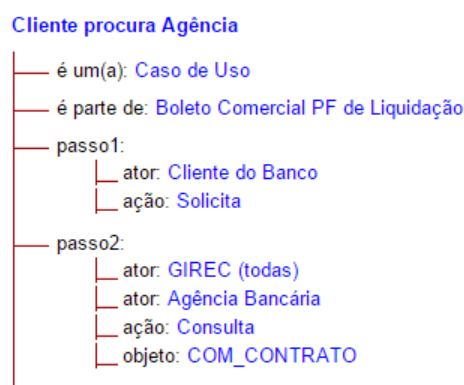
Como se trata apenas do início do projeto, os dados a respeito dos *stakeholders* podem sofrer alterações ao longo do tempo, fazendo com que tanto os requisitos quanto a ontologia sejam refinados.

Continuando a coleta de informações, passa-se para a especificação dos produtos de software que devem ser desenvolvidos, assim como os produtos e processos do negócio, *stakeholders*, e demais dados relacionados com o software. Desta forma, a cada novo ciclo de elicitação de requisito, as novas definições passam a compor as triplas que definem as relações dos objetos com outros elementos. Se já houver uma definição na ontologia para o tipo de informação apresentado, esta definição é exibida aninhada, conforme apresentado na Figura 8.

Os últimos ciclos têm como objetivo definir a utilização e o comportamento esperado dos produtos, em formato de caso de uso, em uma sequência de passos, onde cada passo representa uma ação única a ser realizada.

Tais definições também são convertidas em triplas, definindo o relacionamento do caso de uso com os demais elementos, como o produto de software do qual faz parte, os atores envolvidos, as ações executadas e os objetos usados, como ilustrado na Figura 9.

Figura 9. Ferramenta ERC - triplas ontológicas de caso de uso do produto



Fonte: os autores

Construídos os casos de uso que descrevem o comportamento de cada produto, a especificação dos requisitos do sistema alvo do projeto está completa. A documentação gerada possui uma definição clara de quais produtos de software devem ser entregues, quem são os responsáveis por especificar os requisitos de cada produto, quem irá utilizá-los, quais informações são necessárias para cada um e qual o comportamento esperado na utilização destes produtos. A avaliação do primeiro cenário é apresentada no Quadro 2.

Como os diferentes ciclos de captura de in-

formações utilizam informações uns dos outros, forma-se uma estrutura bastante consistente de rastreabilidade dos requisitos, possibilitando que se obtenha diversas informações sobre cada requisito e sua participação no sistema.

A aplicação do modelo ERC para a elicitação dos requisitos da ferramenta permitiu, ao término do projeto, obter a especificação do sistema e ainda criar uma ontologia de domínio para boletos de cobrança de contratos comerciais. Essa ontologia pode ser utilizada em projetos futuros de desenvolvimento ou aperfeiçoamento da ferramenta.

Neste projeto, foi possível documentar os requisitos funcionais e não funcionais do sistema, com especificações completas e detalhadas da estrutura, do comportamento e das restrições da ferramenta. Simultaneamente, pode-se construir uma ontologia de domínio específica para os boletos de cobrança, que absorveu os requisitos em um espaço de informação organizado em triplas RDF.

Após entrar em operação, os usuários relataram, através de mensagens formais e informais, melhorias promovidas pelos novos produtos de software nos processos de negócio envolvidos. Com base nos relatos dos usuários, pode-se observar que a qualidade empenhada no levantamento dos requisitos contribuiu para a qualidade dos produtos de *software* construídos.

5.2. Segundo cenário: boletos para agências de cobrança externas

O segundo cenário envolve o desenvolvimento de uma ferramenta com as mesmas funcionalidades do primeiro cenário, porém, para ser usada na Extranet da empresa por prestadores das empresas de cobrança tercei-

Quadro 2. Avaliação do primeiro cenário

Aspecto	Critério	Avaliação
Reusabilidade	Reutilizou conceitos da ontologia?	não
	É possível propor automaticamente requisitos a partir da ontologia?	não
	Explora as relações e axiomas da ontologia?	não
Rastreabilidade	Há ligações explícitas entre requisitos e outros elementos?	sim
	Há categorização dos requisitos?	sim
	Um objeto complexo pode ser decomposto em componentes básicos?	sim
Consistência	Há organização hierárquica dos requisitos?	sim
	Permite análise de pares de conceitos entre os diferentes níveis?	sim
Compleitude	Permite análise de nós sem correspondência?	sim

Fonte: os autores

rizadas contratadas para cobrar os clientes inadimplentes.

Neste projeto, graças à ontologia de domínio criada anteriormente, todos os conceitos envolvidos na emissão e tratamento de boletos de liquidação e renegociação de dívidas comerciais para clientes do tipo pessoa física puderam ser imediatamente incorporados. Este fato permitiu que o projeto fosse iniciado quase completo, reduzindo drasticamente o tempo necessário para a especificação da nova ferramenta.

A Figura 10 representa a incorporação de conceitos da ontologia de domínio ao novo projeto, após a inclusão das definições dos produtos de software envolvidos no projeto.

Figura 10. Segundo cenário - reutilização de conceitos

Boleto ACE de cobrança para contratos comerciais

- é um(a): Projeto de Desenvolvimento de Software
- define sistema: SINEB
- fase: Documentação
- no papel Equipe de Desenvolvimento tem: Joaquim Melo Amorim
- no papel Equipe de Desenvolvimento tem: Nilson Massarenti
- no papel Equipe de Negócio tem: GIREC (todas)
- no papel Gerente do Projeto tem: GESTO
- no papel Patrocinador tem: SUREC
- no papel Usuário tem: Cliente do Banco
- no papel Usuário tem: ACE

Fonte: os autores

Pode-se notar que foi adicionado ao projeto, no papel de usuário, o *stakeholder* “Cliente do Banco”. Este pode ser um detalhe pequeno, porém representa um ator fundamental: o usuário final do serviço fornecido através da ferramenta a ser desenvolvida.

O maior esforço necessário neste projeto foi

em relação ao uso e comportamento dos produtos no ambiente da Extranet, sendo necessário criar casos de uso específicos para a emissão de boletos neste ambiente. Para o restante da especificação, foram feitas apenas pequenas inclusões e alterações nos conceitos já presentes na ontologia.

A avaliação do segundo cenário é apresentada no Quadro 3.

Ao final deste projeto a ontologia de boletos de cobrança estava revisada e ampliada, e passou a unir as especificações de dois sistemas diferentes dentro da ontologia de domínio da organização. Ajudada pelo fato de estar parcialmente validada, a ferramenta de boletos para as empresas terceirizadas também foi implantada com sucesso, sendo rapidamente adotada nas rotinas de cobrança das empresas terceirizadas.

5.3. Terceiro cenário: boletos em lote

O terceiro cenário envolve o desenvolvimento de um novo módulo para a ferramenta do primeiro cenário, para enviar propostas preformatadas para os clientes inadimplentes, evitando a necessidade deste cliente procurar a empresa para regularizar suas dívidas. Assim como no segundo cenário, pode-se facilmente reutilizar os conceitos da ontologia de domínio, sendo necessário apenas complementar os requisitos com os novos comportamentos esperados para este módulo.

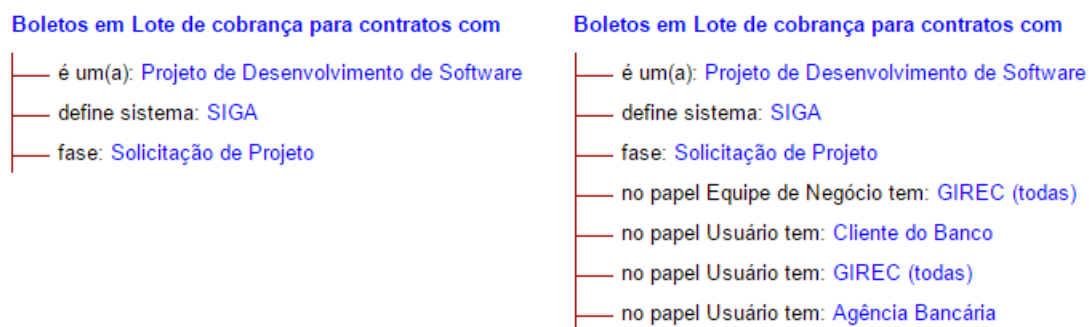
Para exemplificar o potencial da reusabilidade, na Figura 11, à esquerda, o projeto criado sem nenhum requisito e, à direita, os requisitos incorporados pela reutilização dos conceitos definidos no primeiro cenário.

Pode-se observar que os papéis diretamente envolvidos com a especificação dos produtos de software (Equipe de Negócio e Usuários)

Quadro 3. Avaliação do segundo cenário

Aspecto	Critério	Avaliação
Reusabilidade	Reutilizou conceitos da ontologia?	sim
	É possível propor automaticamente requisitos a partir da ontologia?	sim
	Explora as relações e axiomas da ontologia?	sim
Rastreabilidade	Há ligações explícitas entre requisitos e outros elementos?	sim
	Há categorização dos requisitos?	sim
	Um objeto complexo pode ser decomposto em componentes básicos?	sim
Consistência	Há organização hierárquica dos requisitos?	sim
	Permite análise de pares de conceitos entre os diferentes níveis?	sim
Compleitude	Permite análise de nós sem correspondência?	sim

Fonte: os autores

Figura 11. Terceiro cenário - reutilização de conceitos

Fonte: os autores

Quadro 4. Avaliação do terceiro cenário.

Aspecto	Critério	Avaliação
Reusabilidade	Reutilizou conceitos da ontologia?	sim
	É possível propor automaticamente requisitos a partir da ontologia?	sim
	Explora as relações e axiomas da ontologia?	sim
Rastreabilidade	Há ligações explícitas entre requisitos e outros elementos?	sim
	Há categorização dos requisitos?	sim
	Um objeto complexo pode ser decomposto em componentes básicos?	sim
Consistência	Há organização hierárquica dos requisitos?	sim
	Permite análise de pares de conceitos entre os diferentes níveis?	sim
Compleitude	Permite análise de nós sem correspondência?	sim

Fonte: os autores

foram preenchidos automaticamente. Como o modelo ERC permite reutilizar as ontologias de domínio, já no primeiro ciclo a especificação dos *stakeholders* do projeto pode ser preenchida completamente com base na ontologia, uma vez que eles já haviam sido definidos anteriormente.

Os requisitos de base de dados também já iniciam o processo praticamente prontos, sendo necessárias apenas pequenas alterações para adaptar a ferramenta de emissão individual à emissão massificada de boletos. Da mesma forma, os requisitos de casos de uso utilizam os conceitos já definidos na ontologia, sendo necessária a criação apenas dos novos conceitos envolvidos no projeto em andamento.

A avaliação do terceiro cenário é apresentada no Quadro 4.

6. Resultados e discussão

Durante a especificação deste projeto, ficaram claras as vantagens oferecidas pela ontologia de domínio para projetos ligados ao mesmo assunto. Assim como ocorreu nos boletos para as empresas terceirizadas, este projeto foi iniciado com grande parte de seus requi-

sitos já predefinidos, o que poupou tempo e recursos, sem que com isso fosse comprometido nenhum aspecto da qualidade do produto e da especificação.

Vários estudos sobre o processo de comunicação durante a especificação de um sistema chegaram a conclusões semelhantes (Carvalho & Mirandola, 2007; Malanovicz & Brodbeck, 2010; Araujo, Simanski & Quevedo, 2012; Helder et al., 2013): o maior obstáculo para a elicitação de requisitos de qualidade está na dificuldade de mútua compreensão sobre os problemas abordados por parte dos diferentes grupos de profissionais envolvidos. Outra dificuldade apontada pelos autores é a quantidade insuficiente de interações entre as equipes de negócio e de desenvolvimento de sistemas.

A complexidade e importância do processo de engenharia de requisitos exigem o apoio de ferramentas automatizadas (Martins, Nardi & Falbo, 2006). O uso de métodos e ferramentas é crucial para se ter sucesso nos processos de engenharia de requisitos, pois os requisitos devem ser especificados dentro de prazos e orçamentos definidos. Mas para serem efetivas, estas ferramentas precisam se apoiar em um entendimento de consenso acerca dos

elementos envolvidos (Nardi & Falbo, 2006), como os modelos baseados em ontologias.

Gerenciar a interação com o cliente considerando tanto um modelo orientado como a reutilização de conceitos aproxima o cliente do projeto de forma mais eficiente (JIN et al., 2003). O uso de ferramentas de gerenciamento de requisitos com um banco de dados de requisitos comuns, acessíveis a todos *stakeholders*, vem merecendo a atenção da comunidade envolvida com processos de elicitação de requisitos (Farfeleder et al., 2011).

O tratamento explícito do rastreamento de requisitos é essencial para medir a importância e a abrangência destes dentro do sistema como um todo (Shaban-Nejad et al., 2009). Identificar elementos associados aos requisitos permite uma análise de impacto mais eficiente (Martins, Nardi & Falbo, 2006). Os *logs* de alterações criados pelas ferramentas de gestão de requisitos acumulam dados importantes sobre vários tipos de mudanças, no entanto ainda faltam modelos formais com semântica clara e compreensível para acompanhar essas mudanças (Shaban-Nejad et al., 2009).

Ontologias de domínio podem ser utilizadas para formular e analisar requisitos, principalmente na especificação de requisitos dentro de um mesmo domínio (Farfeleder et al., 2011). O uso de tecnologias ontológicas se mostra extremamente vantajoso (Li et al., 2011). No entanto, as ferramentas de modelagem atuais ainda não favorecem sua utilização nesse contexto, sendo muito heterogêneas e carentes de métodos uniformes, o que resulta em pouca utilização por parte das organizações (Duarte, 2011).

Neste trabalho, a conjunção da ER com a Al resultou em melhores produtos, como já preconizava Morville e Rosenfeld (2006). O modelo ERC aqui proposto combina os conceitos utilizados com sucesso nas ferramentas e técnicas que já abordavam o uso de ontologias dentro do processo de elicitação de requisitos, proporcionando um modelo mais simples de utilização, e que suporta as ações envolvidas no levantamento de requisitos.

A abordagem ontológica do ERC cria um vocabulário com significados rigorosamente definidos, o que diminui os problemas de ambiguidade de entendimento sobre o mesmo requisito. As ontologias também ajudam na reutilização dos requisitos em diferentes projetos dentro do mesmo domínio. Além disso, a utiliza-

ção dos ciclos do MAIA com aumento gradual do detalhamento do projeto cria uma estrutura de rastreabilidade completa.

Ao permitir o aumento da interação entre todos os envolvidos por meio de um ambiente único, compartilhado e sempre disponível, o modelo ERC aproximou o cliente do projeto. A objetividade da documentação gerada facilitou a aprovação dos gestores para o escopo do projeto e demais requisitos. Essa rapidez na avaliação permitiu um refinamento bastante eficiente de toda documentação do projeto, corrigindo divergências e descartando requisitos que não faziam parte do escopo do projeto.

6.1. Avaliação dos cenários

Seguindo a metodologia proposta, analisou-se a reusabilidade da ontologia de domínio, e os aspectos de rastreabilidade, consistência e completude dos artefatos gerados pelos cenários.

No segundo e terceiro cenários, pode-se observar que o modelo ERC permite que a ontologia de domínio seja facilmente desenvolvida e reutilizada em qualquer projeto dentro do mesmo domínio. Esta capacidade de reutilização permite atualizar e ampliar a ontologia de domínio a cada novo projeto. E quanto mais ampla esta ontologia se torna, mais conceitos dos processos de negócio são descritos, criando um consenso para as definições dos processos da organização. A criação desse consenso reduz o impacto da comunicação entre as diferentes equipes envolvidas na elicitação de requisitos.

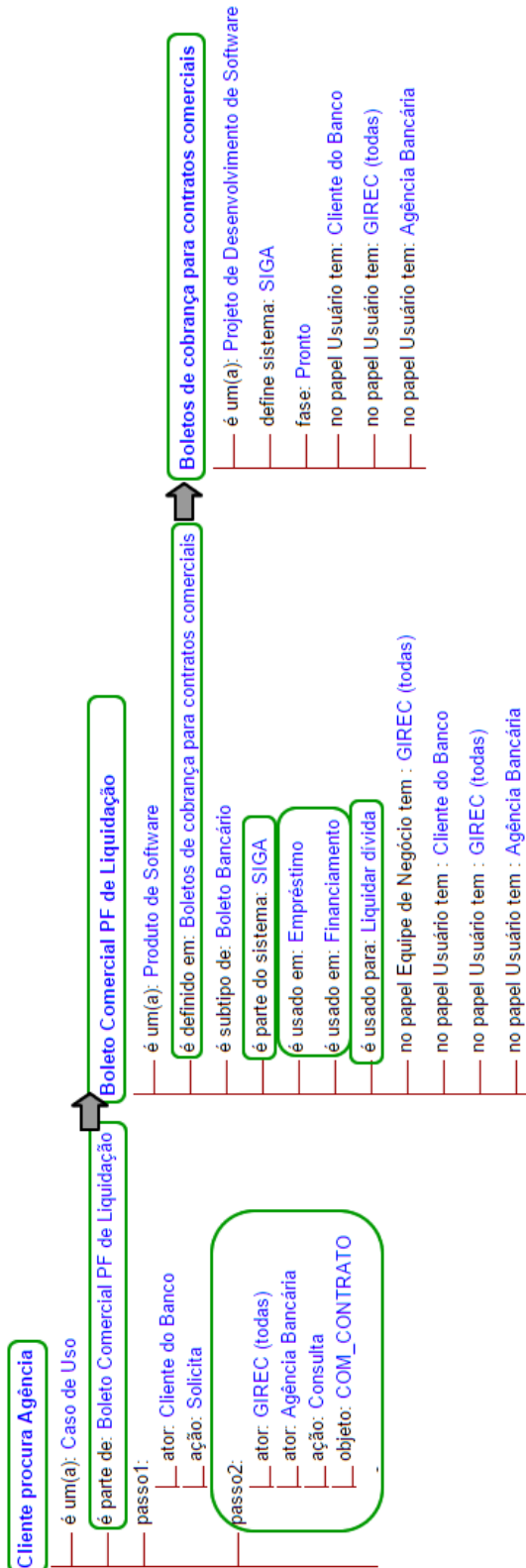
Para analisar a rastreabilidade, foram identificados os relacionamentos entre os requisitos e demais elementos da especificação, partindo do requisito mais específico da aplicação, que é um passo de um caso de uso. A Figura 12 ilustra alguns dos elementos rastreados a partir do “passo 2” do caso de uso “Cliente procura Agência”, construído no primeiro cenário.

Os itens circulados são os elementos rastreados, e as setas cinzas representam uma mudança de nível conceitual. A análise mostra que o modelo implementa uma estrutura de rastreabilidade dos requisitos, pois permite mapear todos os elementos ligados direta e indiretamente a um requisito específico.

A análise também mostra que o modelo permite verificar a consistência pelo relacio-

namento de requisitos de diferentes níveis conceituais, como no caso da hierarquia conceitual formada pelos elementos do caso de

Figura 12. Ferramenta ERC - análise de rastreabilidade do primeiro cenário



Fonte: os autores

uso “Cliente procura Agência”, do produto de software “Boleto Comercial PF de Liquidação” e do projeto “Boletos de cobrança para contratos comerciais”.

A análise mostra ainda que o modelo permite verificar a completude pela identificação de requisitos com relacionamentos incompletos, como no “passo 1” do caso de uso “Cliente procura Agência”, onde se pode constatar que o passo possui “ator” e “ação”, mas não possui um “objeto” vinculado a “ação”.

Para confirmar a efetividade do modelo em relação aos aspectos analisados, as especificações geradas nos cenários foram submetidas às seguintes questões de competência da ontologia de requisitos de Nardi e Falbo (2006):

- QC₃ Os requisitos do projeto são as definições contidas nos stakeholders e seus papéis, nos produtos, nos casos de uso, nos objetos e seus atributos.
- QC₄ Os requisitos são alocados nos produtos de software que compõem o sistema.
- QC₅ Os responsáveis pelos requisitos são os stakeholders com papel de Equipe de Negócio.
- QC₆ Os stakeholders de um requisito são aqueles vinculados ao mesmo, podendo atuar em diversos papéis.
- QC₇ A origem de um requisito pode ser encontrada rastreando seu relacionamento com os elementos do projeto.
- QC₈ O estado de um requisito pode ser criado, aprovado ou rejeitado.
- QC₉ Um determinado requisito pode ser decomposto em outros requisitos de nível menos abstrato, como o produto de software, que pode ser decomposto em casos de uso.
- QC₁₀ Os requisitos dependentes de um determinado requisito são identificados através dos relacionamentos entre eles, onde o atributo é dependente do objeto, que por sua vez é dependente do produto.
- QC₁₁ Requisitos conflitantes são identificados com a ajuda da ontologia de domínio, que não aceita definições ambíguas para formação de seu vocabulário.
- QC₁₃ Os artefatos que descrevem, modelam ou implementam um requisito são os objetos e os casos de uso.
- QC₁₄ O gerenciamento das mudanças nos requisitos e nos artefatos a eles relacionados é feito através da estrutura de ras-

treabilidade.

QC₁₅ A qualidade dos requisitos é avaliada através da análise de consistência e completude.

Portanto, é possível responder a todas as questões de competência através dos artefatos gerados pelo modelo, confirmando a capacidade do modelo de avaliar a qualidade dos requisitos durante um projeto, reduzindo o impacto da comunicação entre as equipes na qualidade dos requisitos.

6.2. Resultado da avaliação

Após a aplicação prática do modelo e análise dos cenários de desenvolvimento propostos, foram obtidos os seguintes resultados: (i) os conceitos definidos no primeiro cenário foram facilmente reaproveitados no segundo e terceiro cenários, confirmando a capacidade de reusabilidade da ontologia de domínio criada a partir do modelo; (ii) foi possível construir relacionamentos explícitos entre todos elementos da especificação de requisitos, permitindo analisar a rastreabilidade, a consistência e a completude dos artefatos gerados; (iii) a especificação gerada nos cenários de desenvolvimento consegue responder a todas as questões de competência selecionadas para avaliar o modelo; e (iv) constata-se que o modelo é capaz de contribuir na redução do impacto dos problemas de comunicação entre as equipes na qualidade dos requisitos gerados.

7. Conclusão

O objetivo deste trabalho é propor um modelo capaz de reduzir o impacto dos problemas de comunicação entre os *stakeholders* na qualidade dos requisitos gerados, através da formação de uma estrutura semântica a partir dos requisitos da especificação, que permita verificar se o documento de requisitos está completo, consistente e não ambíguo.

A importância de um acompanhamento constante da qualidade dos requisitos pôde ser confirmada por meio dos referenciais teóricos que mencionam os impactos dos requisitos nos projetos de desenvolvimento de sistema.

Foram perceptíveis as vantagens de incorporar conceitos de outras áreas ao processo de levantamento de requisitos, como método MAIA (Costa, 2009), o Espaço Conceitual de Três Camadas (Jin et al., 2003), e as questões de competência a serem respondidas na Onto-

logia de Requisitos de Software (Nardi & Falbo, 2006).

A aplicação do modelo mostrou-se bastante promissora, desenvolvendo uma ontologia de domínio e criando um ambiente de controle da qualidade dos requisitos que colaboraram na redução do impacto da comunicação entre equipes na qualidade dos requisitos.

A proposta do modelo de compartilhar informações em um ambiente único, onde os requisitos fossem declarados de forma estruturada, e simultaneamente transformados em triplas para compor a ontologia de domínio da organização mostrou-se viável e útil.

Foi possível definir claramente os diferentes elementos que compõem a especificação de um produto de software, tornando a comunicação entre as equipes de negócio e de TI mais objetiva e menos vulnerável a ambiguidades.

A organização do espaço de informação promovido pelo MAIA na criação da ontologia de domínio permitiu que os requisitos declarados em um projeto pudessem ser facilmente reutilizados e até melhorados em outros projetos.

O vocabulário com significados rigorosamente definidos, criado através da ontologia de domínio, a implementação da rastreabilidade e das análises de qualidade dos requisitos durante o processo de levantamento de requisitos, reduziram o impacto dos problemas de comunicação entre as diferentes equipes envolvidas nos projetos de desenvolvimento de software.

Farfeleder et al. (2011) afirmam que, para que o processo de especificação gere requisitos de alta qualidade, a ferramenta de apoio precisa atender dois pontos: (i) o sistema deve propor automaticamente, ao menos, partes dos requisitos, utilizando informações provenientes de uma ontologia de domínio, e (ii) o sistema deve explorar as relações e axiomas da ontologia de domínio. Como os projetos são construídos dentro de uma ontologia de domínio, todas suas definições estão interligadas por um vocabulário comum. Assim, conceitos podem ser reutilizados e atualizados a cada novo projeto, criando um consenso entre as equipes.

Quanto mais projetos dentro da organização utilizarem o modelo, mais ontologias de domínio específicas serão criadas, tornando a ontologia de domínio da organização cada vez mais precisa e ampla. Consequentemente, quanto

mais ampla for a ontologia de domínio, maior será a possibilidade de reutilização da mesma em outros projetos de desenvolvimento da organização.

Portanto, o objetivo principal deste trabalho foi alcançado. O modelo ERC mostrou-se uma interessante alternativa para dar suporte ao levantamento de requisitos, ajudando a

garantir uma especificação de requisitos de qualidade.

Como trabalho futuro, sugere-se a aplicação do modelo em outros projetos de desenvolvimento, para que se possa generalizar sua proposta para outros cenários, uma vez que o modelo foi testado em um conjunto pequeno de cenários.

Referências

- Al-Hothali, S. A., Al-Zubaidi, N. A. & Subbarao, A. (2012). Requirements Elicitation For Software Projects. *International Journal of Computer Science and Information Security*, 10(11), 64-71.
- Araujo, D. C., Simanski, E. S. S. & Quevedo, D. M. (2012). Comunicação interna: relação entre empresa e colaboradores, um estudo de caso. *Brazilian Business Review*, 9(1), 47-64.
- Bassi Filho, D. L. (2008). *Experiências com desenvolvimento ágil* (Dissertação de Mestrado em Ciência da Computação), São Paulo: Universidade de São Paulo.
- Capilla, R., Babar, M. A. & Pastor, O. (2012). Quality requirements engineering for systems and software architecting: methods, approaches, and tools. *Requirements Engineering*, 17(4), 255-258.
- Carvalho, M. M., Mirandola, D. (2007). A comunicação em projetos de TI: uma análise comparativa das equipes de sistemas e de negócios. *Produção*, 17(2), 330-342.
- Chikh, A., Abulaish, M., Nabi, Syed I. & Alghathbar, K. (2011). An ontology based information security requirements engineering framework. *Communications in Computer and Information Science*, 186, 139-146.
- Costa, I. M. (2009). *Um método para arquitetura da informação : fenomenologia como base para o desenvolvimento de arquiteturas da informação aplicadas* (Dissertação de Mestrado em Ciência da Informação). Brasília: Universidade de Brasília.
- Costa, L. A., Zoucas, A. C. & Alves, J. B. M. (2012). Elicitação de requisitos de software no setor público: lições aprendidas e recomendações para mitigação de riscos. *Revista Brasileira de Administração Científica*, 3(2), 214-226.
- Duarte, J. C. (2011). *Uma arquitetura ágil da informação organizacional* (Tese de Doutorado em Ciência da Informação). Brasília: Universidade de Brasília.
- Egyed, A. & Grunbacher, P. (2004). Identifying requirements conflicts and cooperation: How quality attributes and automated traceability can help. *IEEE software*, 21(6), 50-58.
- Ehresmann, A. C. & Vanbremeersch, J.-P. (2006). The memory evolutive systems as a model of Rosen's organisms- (metabolic, replication) systems. *Axiomathes*, 16(1-2), 137-154.
- Falbo, R. A., Ruy, F. B., Pezzin, J. & Moro, R. D. (2004). *Ontologias e Ambientes de Desenvolvimento de Software Semânticos*. Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento.
- Farfeleder, S., Moser, T., Krall, A., Stalhane, T., Omoronyia, I. & Zojer, H. (2011). Ontology-driven guidance for requirements elicitation. *Lecture Notes in Computer Science*, 6644, 212-226.
- Góes, A. S., Silva, J. P. & Barros, R. M. (2013). Melhoria no Processo de Levantamento de Requisitos para Software de Gestão Pública: Um Estudo de Caso utilizando Instruções de Trabalho. *Revista de Sistemas de Informação da FSMA*, 1(12), 21-32.
- Hedler, H. C., Cruz, K. C. B., Campos, R. P. & Alonso, L. B. N. (2013). Comunicação e compartilhamento do conhecimento entre equipes em automação de processos. *Comunicologia*, 6(2), 165-183.
- Hofmann, H. F. & Lehner, F. (2001). Requirements engineering as a success factor in software projects. *IEEE software*, 18(4), 58-66.
- Honkola, J., Laine, H., Brown, R. & Tyrkkö, O. (2010). Smart-M3 information sharing platform. *The IEEE symposium on Computers and Communications*, 1041-1046.
- Ito, M. L., Fuzii, R. Y. M., Souza, R. C. G., Valencio, C. R., & Tronco, M. L. (2011). Support tool to the validation process of functional requirements. *IEEE Latin America Transactions*, 9(5), 889-894.
- Jackson, M. (2010). Representing structure in a software system design. *Design Studies*, 31(6), 545-566.
- Jackson, M. (2013). Formalism and Intuition in Software Engineering. In: M. Jackson *Perspectives on the Future of Software Engineering*, 335-347.
- Jin, Z. Bell, D. A., Wilkie, F. G. & Leahy, D. G. (2003). Automated requirements elicitation: Combining a model-driven approach with concept reuse. *International Journal of Software Engineering and Knowledge Engineering*, 13(1), 53-82.
- Khouri, S., Boukhari, I., Bellatreche, L, Sardet, E., Jean, S. & Baron, M. (2012). Ontology-based structured web data warehouses for sustainable interoperability: requirement modeling, design methodology and tool. *Computers in Industry*, 63(8), 799-812.
- Li, G., Jin, Z., Xu, Y. & Lu, Y. (2011). An engineerable ontology based approach for requirements elicitation in process

- centered problem domain. International Conference on Knowledge Science, Engineering and Management. *Lecture Notes in Computer Science*, 7091, 208-220.
- Martins, A. F., Nardi, J. C. & Falbo, R. A. (2006). ReqODE: Uma Ferramenta de Apoio à Engenharia de Requisitos Integrada ao Ambiente ODE. XX Simpósio Brasileiro de Engenharia de Software - SBES.
- Melo, A. M. C. (2010). *Um modelo de Arquitetura da Informação para processos de investigação científica* (Dissertação de Mestrado em Ciência da Informação). Brasília: Universidade de Brasília.
- Moresi, E. (2003). *Metodologia da pesquisa*. Universidade Católica de Brasília. Disponível em: <http://ftp.unisc.br/portal/upload/com_arquivo/1370886616.pdf>.
- Moresi, E. A. D., Ramos, R. G. C. & Prado, H. A. (2010). Mapeamento de informações organizacionais: um estudo na Embrapa. *TransInformação*, 22(2), 101-110.
- Morville, P. & Rosenfeld, L. (2006). *Information architecture for the World Wide Web* (3ª ed). Sebastopol, CA: O'Reilly.
- Motta, R. W. (2012). *Engenharia de requisitos em sistemas Web: Aplicação Prática* (Monografia de Especialização em Engenharia de Sistemas). Brasília: Escola Superior Aberta do Brasil.
- Nardi, J. C. & Falbo, R. A. (2006). *Uma Ontologia de Requisitos de Software*. Conference: Memórias de la IX Conferencia Iberoamericana de Software Engineering (CIBSE). La Plata, Argentina.
- Neumann, T. & Weikum, G. (2008). RDF-3X: a RISC-style engine for RDF. *Proceedings of the VLDB Endowment*, 1(1), 647-659.
- Ramos, R. A., Carvalho, A., Monteiro, C., Silva, C., Castro, J., Alencar, F., Afonso, R. (2006). *Avaliação da Qualidade de Documentos de Requisitos Orientado a Aspectos*. Congresso Ibero-Americano em Engenharia de Software. La Plata: CIBSE.
- Resmini, A. & Rosati, L. (2012). A Brief History of Information Architecture. *Journal of Information Architecture*, 3(2), 33-46.
- Semedo, M. J. M. (2012). *Ganhos de produtividade e de sucesso de Metodologias Ágeis VS Metodologias em Cascata no desenvolvimento de projectos de software* (Dissertação de Mestrado em Engenharia Informática e Sistemas de Informação). Lisboa: Universidade Lusófona de Humanidades e Tecnologias,.
- Shaban-Nejad, A., Ormandjieva, O., Kassab, M. & Haarslev, V. (2009). Managing requirement volatility in an ontology-driven clinical LIMS using category theory. *International Journal of Telemedicine and Applications*, 2009.
- Siqueira, A. H. (2008). *A lógica e a linguagem como fundamentos da arquitetura da informação* (Dissertação de Mestrado em Ciência da Informação). Brasília: Universidade de Brasília.
- Siqueira, A. H. (2012). *Arquitetura da informação: uma proposta para a fundamentação e caracterização de uma disciplina científica* (Tese de Doutorado em Ciência da Informação). Brasília: Universidade de Brasília.
- Soares, H. A. & Moura, R. S. (2015). A methodology to guide writing Software Requirements Specification document. *Latin American Computing Conference (CLEI)*, 1-11.
- Sommerville, I. (2011). *Engenharia de Software* (9ª ed). São Paulo: Pearson Education.
- Soomro, S., Hafeez, A., Shaiikh, A., Musavi, S. H. A. (2014). Ontology Based Requirement Interdependency Representation and Visualization. *Communications in Computer and Information Science*, 414, 259-270.
- Trindade, T. L. N., Lima-Marques, M. & Souza, F. A. (2009). *Information architecture on cost management*. 6ª Conferência Internacional sobre Sistemas de Informação e Gestão de Tecnologia.
- Vallerão, A. G. & Roses, L. K. (2013). Monitoramento e Controle de Projetos de Desenvolvimento de Software com o Scrum: Avaliação da Produção Científica. *Revista de Gestão e Projetos*, 4(2), p. 100-127.

Sobre os autores

Rafael Wobeto Motta

Graduado em Ciência da Computação pela Universidade Federal do Mato Grosso (2005), Especialista em Engenharia de Sistemas pela Escola Superior Aberta do Brasil (2012) e Mestre em Gestão do Conhecimento e Tecnologia da Informação pela Universidade Católica de Brasília (2016).

Edilson Ferneda

Graduado em Tecnologia de Computação pelo Instituto Tecnológica de Aeronáutica (1979), Mestre em Sistemas e Computação pela Universidade Federal da Paraíba (1988) e Doutor em Ciência da Computação pelo *Laboratoire d'Informatique, Robotique et de Microélectronique de Montpellier*, França (1992). Entre 1986 e 2004, foi professor do Departamento de Sistemas e Computação da Universidade Federal da Paraíba (atual Universidade Federal de Campina Grande), tendo atuado nos cursos de Bacharelado em Ciência da Computação, Mestrado em Informática e Doutorado em Engenharia Elétrica. Desde 2001 é professor titular da Universidade Católica de Brasília, onde atua no Curso de Bacharelado em Ciência da Computação e no Mestrado em Gestão do Conhecimento e Tecnologia da Informação. Seus interesses incluem Inteligência Artificial e Gestão do Conhecimento.

Fábio Bianchi Campos

Graduado em Engenharia Elétrica pela Universidade de Brasília (1984), Mestre em Ciência da Computação pela Universidade de Brasília (1996) e Doutor em Engenharia de Sistemas e Computação pela UFRJ (2008), foi Professor da Universidade Católica de Brasília (UCB) entre 1999 e 2014, na área de Engenharia de Software e Governança de TI, atuando no Mestrado em Gestão do Conhecimento e Tecnologia da Informação e no Bacharelado em Ciência da Computação. Avaliador e implementador do modelo de melhoria do software Brasileiro MPS.BR. Consultorias, treinamentos e pesquisas diversas, atuando princi-

palmente nos seguintes temas: Melhoria de processos, Auditoria e Avaliação de Processos e Sistemas, Qualidade de Software, Gestão da Aquisição de Serviços e produtos de TI, indicadores e métricas em TI, e estimativas em projetos de TI. Atualmente, é Analista de Informática do PRODASEN, do Senado.

Viviane Judith Conte de Oliveira Pinto

Graduanda em Engenharia Civil pela Universidade Católica de Brasília.