

# A Process Model for Standardization and Increase in the Requirements Quality

J. A. Dorigan and R. M. de Barros

**Abstract**— One of the biggest problems encountered in Requirements Engineering is the fact that the requirements are poorly specified, inconsistent with the stakeholder's needs or badly written. This study aims to present a model of Requirements Engineering Process allowing requirements description standardization, seeking to improve the software requirements specification quality. The proposed process will operate in the Requirements Specification and Validation assisting the Requirements Engineer on writing requirements in Natural Language. A comparative study between the proposed process and others who offer assistance to Requirements Engineering is also presented, so benefits of using the proposed process could be identified in software development.

**Keywords**— Requirements Engineering Process, Requirements Specification and Validation, requirements standardization, quality on the description of the requirements.

## I. INTRODUÇÃO

**L**EVANTAR corretamente os requisitos é uma tarefa muito importante na construção de um software. Quando expressos em termos do comportamento do software, esse comportamento deve ser percebido por um observador externo ao software [7] [9].

Aos Requisitos estão associados os principais problemas do desenvolvimento de software, na maioria dos casos eles não refletem as reais necessidades dos usuários, por serem incompletos ou inconsistentes [7].

Outra principal dificuldade é fazer com que a especificação esteja em conformidade com a necessidade do cliente. Na maioria das vezes acontece uma interpretação errada por parte do Engenheiro de Requisitos (neste trabalho será usada a sigla ER) ou o *Stakeholder* (Cliente) não consegue expressar de forma clara suas ideias. De toda forma, estes problemas criam **requisitos despadronizados e inconsistentes**.

Um modo de evitar estas inconsistências seria se as organizações dispusessem de um processo de Engenharia de Requisitos definido, oferecendo padronização na descrição e aumentando a qualidade da Especificação de Requisitos.

O objetivo deste trabalho, dentro deste cenário, é propor um modelo de Processo de Engenharia de Requisitos que forneça subsídios para a padronização da descrição de requisitos, criada por um ER, buscando aumentar a qualidade da especificação, em acordo com as necessidades do cliente.

A estruturação deste trabalho é feita da seguinte forma: na seção II são apresentadas abordagens teóricas que serão

utilizadas no decorrer do artigo, a seção III apresenta trabalhos relacionados, referentes a processos e técnicas que auxiliam a especificação e validação de requisitos de software. Na seção IV é exposto o modelo de processo de Engenharia de Requisitos proposto por este artigo, a seção V mostra uma Avaliação Comparativa do processo proposto e identifica benefícios de sua utilização. Por fim, a seção VI apresenta as Conclusões e Trabalhos Futuros.

## II. FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta os principais conceitos utilizados neste trabalho, tais como: Engenharia de Requisitos, Revisão de Requisitos e Inspeção de Software, Processamento de Linguagem Natural.

### A. Engenharia de Requisitos

Para elaborar e manter uma especificação de requisitos é necessário que os desenvolvedores executem um conjunto estruturado de atividades destinadas a identificar, analisar, especificar e validar requisitos. Este conjunto de atividades, iterativas por natureza, recebe o nome de Processo de Engenharia de Requisitos [9].

A Fig. 1 mostra como é definido este processo e os relacionamentos entre suas áreas.

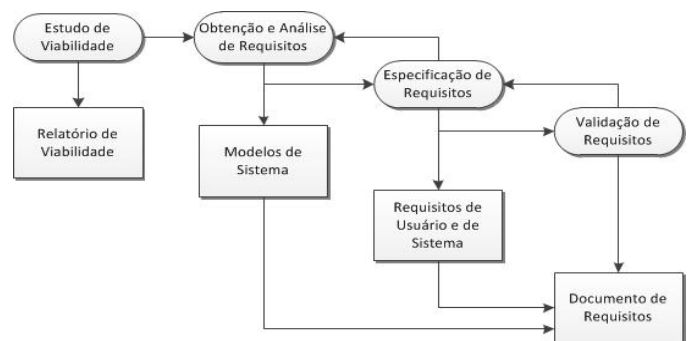


Figura 1. Processo de Engenharia de Requisitos [9].

Segundo Pressman [7], quando a organização não dispõe deste processo definido formalmente e amplamente divulgado, os desenvolvedores elaboram as especificações de forma empírica, executando atividades não padronizadas e definidas individualmente.

Se isto ocorre, a qualidade da especificação dependerá exclusivamente da experiência e formação das pessoas, com uma elevada probabilidade de ocorrerem conflitos.

As atividades relacionadas ao processo de Engenharia de Requisitos, apontadas por Sommerville [9], e sendo

J. A. Dorigan, Universidade Estadual de Maringá (UEM), Umuarama, Paraná, Brasil, jadorigan@gmail.com

R. M. de Barros, Universidade Estadual de Londrina (UEL), Londrina, Paraná, Brasil, rodolfo@uel.br

consideradas um conjunto de boas práticas são apresentadas a seguir:

1. **Identificar:** Corresponde à atividade de obter e compreender os requisitos de um sistema, através de entrevistas com os interessados, da análise do domínio do problema ou de estudos de mercado. Na identificação de requisitos podemos considerar como boas práticas:
  - A redação de uma declaração de visão e escopo do sistema;
  - A definição dos procedimentos para desenvolvimento dos requisitos;
  - A identificação das classes de usuários e dos diferentes grupos de interesse;
  - A identificação dos casos de uso;
  - A análise dos fluxos de trabalho dos usuários;
  - A definição dos atributos de qualidade do sistema;
  - O desenvolvimento de mecanismos que possibilitem o reuso de requisitos.
2. **Analisar:** Na análise, os requisitos identificados são compreendidos e detalhadamente analisados por todos os interessados no sistema. Nesta atividade surgem muitos conflitos, sendo comum a necessidade de negociação para que os requisitos sejam aceitos por todos. Boas práticas de análise referem-se à:
  - Elaboração de um diagrama de contexto do sistema;
  - Criação de protótipos;
  - Análise de viabilidade;
  - Priorização dos requisitos.
3. **Especificar:** Uma vez compreendidos, analisados e aceitos, os requisitos devem ser documentados com um nível de detalhamento adequado, produzindo a especificação de requisitos de software. Pode ser utilizada a linguagem natural ou diagramas, como os propostos pela *Unified Modeling Language* (UML) [1].
4. **Validar:** Após terem sido documentados, é necessário que os requisitos sejam cuidadosamente validados, principalmente quanto à consistência e a completude. Esta atividade visa identificar problemas nos requisitos, antes do início da implementação. A importância da validação é caracterizada pelo fato de que a correção de um erro nesta fase possui um custo muito inferior à correção do mesmo erro nas fases posteriores do processo de desenvolvimento.

Neste cenário, o processo proposto atuará entre as áreas de Especificação e Validação de Requisitos, como mostrado na Fig. 1, analisando a linguagem natural escrita e buscando com que a descrição dos requisitos obtenha um nível de qualidade adequado, diminuindo os riscos do desenvolvimento.

#### B. Revisão de Requisitos e Inspeção de Software

Em Engenharia de Software, segundo Sommerville [9], Revisão de Requisitos é uma atividade onde os desenvolvedores do projeto se reúnem com os *stakeholders* para verificar os produtos de desenvolvimento, nesse caso as especificações de requisitos, com a finalidade de detecção de defeitos, comentários e aprovação.

No mesmo sentido, Inspeção de Requisitos, que deriva da Inspeção de Software, é uma atividade estática que se baseia no exame visual de produtos do desenvolvimento para detectar defeitos, violações de padrões de desenvolvimento, e outros problemas [9]. O método de inspeção de software foi desenvolvido por Michael E. Fagan, em 1976, para a IBM.

Frequentemente pode ocorrer a confusão dos papéis das revisões e inspeções. A diferença é que as revisões são conduzidas durante o desenvolvimento dos requisitos para demonstrar e refinar abordagens, solicitar e comparar pontos de vistas, promover melhorias, identificar problemas e criar soluções para eles. Já as inspeções são conduzidas quando um artefato está completo e tem um único objetivo: encontrar defeitos. Para Fagan [5], o motivo em se manter esse objetivo exclusivo nas inspeções é que experimentos mostraram que quando as revisões são incluídas nas inspeções, ou quando mais de um objetivo é incluído, há diminuição da eficácia da detecção de defeitos das inspeções.

Tanto a inspeção como as revisões visam melhorar a qualidade do software antes de sua entrega. Ambos têm o propósito de encontrar defeitos para que possam ser corrigidos e ambos podem ser aplicados no início do desenvolvimento de software, podendo a inspeção ser aplicada também em planos e casos de testes.

#### C. Processamento de Linguagem Natural

Linguagem natural são linguagens utilizadas para comunicações no dia a dia dos seres humanos, línguas como o inglês ou português. Já as “linguagens artificiais” são interpretadas por computadores e máquinas em geral, como as linguagens de programação e notações matemáticas.

O Processamento de Linguagem Natural (PLN) é definido, segundo Costa [4], como qualquer tipo de manipulação computacional de linguagens naturais. Este campo da Inteligência Artificial está entre os grandes desafios da computação, pois envolve a junção dos conhecimentos linguísticos com os computacionais para prover mecanismos “inteligentes” e eficientes de interação homem-máquina.

Algumas contribuições importantes estão na área de tradução automática, que evoluiu consideravelmente nos últimos anos, na recuperação de informação e na engenharia de software, nesta última, com estudos de aplicações visando principalmente à etapa de especificação de requisitos de software.

Em Engenharia de Software, alguns trabalhos como o de Sayão [8] e Chen et al. [3], estudam a aplicação de técnicas de PLN na verificação e validação de requisitos com o objetivo de melhorar a qualidade e auxiliar o trabalho das equipes na análise de requisitos. A maioria das especificações de requisitos é escrita em linguagem natural, em alguns casos sendo complementadas por outros tipos de notações como diagramas, equações, modelos formais, etc.

Atualmente é comum que o desenvolvimento de software seja feito por equipes geograficamente distribuídas, para estes casos, a verificação e validação de requisitos são de alta complexidade.

As técnicas de PLN associadas ao levantamento de requisitos permitem a detecção de erros durante a fase inicial de especificação do projeto possibilitando a economia de tempo e trabalho, podemos perceber que estas técnicas podem ser aplicadas a documentos de requisitos e outros artefatos, apoiando o processo de validação de requisitos.

### III. TRABALHOS RELACIONADOS

Através da literatura pôde-se perceber que as áreas de especificação e validação de requisitos carecem de modelos de processos que ajudem o ER, a identificar, analisar, especificar e validar os requisitos.

Existem ferramentas disponíveis que propiciam auxílio à desenvolvedores na área de Engenharia de Requisitos. Muitas destas ferramentas são utilizadas sem o conhecimento, por parte do ER, dos processos, metodologias ou modelos que subsidiaram a criação das mesmas. Por isso, é de extrema importância o entendimento do processo utilizado para aumentar a qualidade da especificação de requisitos, diminuindo riscos nas fases posteriores do projeto.

No trabalho de Chen et al. [3] descreve-se uma técnica que utiliza um pré-processamento da linguagem natural usada na criação dos requisitos de um software. Esse pré-processamento faz uso de domínios gerais e específicos para separar os requisitos, após isso, a técnica faz uma busca por palavras chamadas de “objetivos” pelo autor, que são descritos como a parte central do requisito. Neste trabalho o autor deixa claro que o objetivo da técnica é auxiliar o ER provendo informações extraídas da escrita dos requisitos.

Segundo Cabral et al. [2] a aplicação de técnicas de leituras sistemáticas como o *Perspective-Based Reading* (PBR) e não sistemáticas como Checklist durante a análise de requisitos trouxe bons resultados. Nestas técnicas vários inspetores inspecionam o documento de contexto de um software buscando erros ou inconsistências antes de transcrever o documento de requisitos. Esses erros são posteriormente avaliados para a comparação entre as duas técnicas. O trabalho também tratou da avaliação das perspectivas de inspetores, mostrando que diferentes papéis como ER, Arquitetos e Testadores podem apresentar diferentes resultados quando da utilização de uma técnica de leitura.

Um modelo de processo de Engenharia de Requisitos foi proposto no trabalho de Pandey et al. [6]. Os autores abrangem toda a área de Engenharia de Requisitos, propondo a divisão em quatro fases: Elicitação e Desenvolvimento de Requisitos, Documentação de Requisitos, Verificação e Validação de Requisitos e por fim Gerenciamento e Planejamento de Requisitos. Os requisitos ficam armazenados num *Software Requirement Specification* (SRS) e os autores apontam que o diferencial do trabalho é, além de abranger todas as áreas, possibilitar o gerenciamento de mudanças em requisitos já acordados dentro de um desenvolvimento de software.

Para melhor entendimento dos trabalhos aqui citados, é mostrada a Tabela I que demonstra um comparativo entre os trabalhos que se relacionam com este.

TABELA I. COMPARATIVO ENTRE TRABALHOS RELACIONADOS.

|  | Chen et al.  | Cabral et al.  | Pandey et al.   |
|--|--|--|---|
| <b>Identificação dos Requisitos</b>                | Stakeholders criam os requisitos.  | Não especificado no trabalho.  | São identificados requisitos “crus” de pontos de vista de stakeholders.           |
| <b>Análise de Requisitos</b>                       | Não é feita uma análise dos requisitos.  | Não é especificado no trabalho quem faz a análise dos requisitos.                                  | Um Analista de Sistema analisa os requisitos “crus” e cria requisitos de sistema. |
| <b>Especificação de Requisitos</b>                 | A especificação vem da identificação dos requisitos feita pelos stakeholders, escrita em LN. | Os requisitos já estão especificados, contidos no Documento de Requisitos.                         | A especificação é feita através do SRS.   |
| <b>Validação de Requisitos</b>                     | Não é objetivo do trabalho a validação os requisitos.  | Foca na validação dos requisitos utilizando para isso técnicas de leitura e revisão de requisitos. | A validação é feita analisando o SRS juntamente com stakeholders.                 |
| <b>Utiliza agrupamento de Requisitos</b>           | Sim, utiliza Domínios Gerais e Específicos.  | Não  | Sim, em níveis do sistema.  |
| <b>Utiliza padronização de Requisitos</b>          | Não  | Não  | Não   |
| <b>Qualidade dos Requisitos</b>                    | Sim, busca o aumento da qualidade dos próprios requisitos.                                   | Sim, busca o aumento da qualidade da validação dos requisitos.                                     | Sim, busca o aumento da qualidade do processo todo.                               |
| <b>Nível de Participação dos Stakeholders</b>      | Alto, identificando e especificando os requisitos.   | O trabalho não cita como foi a participação dos stakeholders.                                      | A participação dos stakeholders deve ser constante.                               |
| <b>Nível de Interação manual do modelo</b>         | Médio. O modelo utiliza ajuda computacional para o tratamento dos requisitos.                | Alto. O modelo utiliza técnicas que necessitam de muita interpretação humana.                      | Alto. O modelo não cita possibilidades de automação do processo.                  |
| <b>Reutilização Parcial ou Total de Requisitos</b> | Não  | Não  | Não   |

Esses fatos comprovam a necessidade da pesquisa e desenvolvimento de modelos de processos de apoio à Engenharia de Requisitos.

### IV. O PROCESSO PROPOSTO

Esta seção apresenta o processo, proposto por este artigo, para auxiliar o ER na especificação e validação de requisitos dentro de um desenvolvimento de software.

Primeiramente, são apresentados alguns conceitos e artefatos utilizados na descrição do processo.

- 1. Engenheiro de Requisitos (ER):** Responsável pela atividade de identificação, análise, especificação e validação dos requisitos;
- 2. Stakeholder (Cliente):** Responsável por prover informações claras das suas necessidades, para que os requisitos possam estar de acordo com elas e refleti-las dentro do software;
- 3. Requisitos:** São as necessidades que o *stakeholder* tem e que o software em desenvolvimento deve implantar. São descritos utilizando linguagem natural e inicialmente serão tratados como itens;
- 4. Nível de Rigoridade:** No processo propomos a utilização de um nível de rigoridade para a aceitação da descrição de um requisito. O nível de rigoridade utilizado fica a critério do ER, levando em conta as palavras identificadas e tendo em mente a obtenção de uma maior qualidade na padronização da descrição;
- 5. Lista de Requisitos:** Uma lista comum que contém todos os requisitos identificados do software que está em desenvolvimento;
- 6. Documento de Requisitos:** No processo proposto não faremos menção a um modelo de Documento de Requisitos de Software (DRS) específico, apenas apresentamos a possibilidade de criação deste documento.

O processo inicia-se quando *Stakeholder* e ER interagem em reuniões iterativas e incrementais para debaterem quais serão os requisitos do software. Nessas reuniões, primeiramente o ER deve definir dois tipos de contextos: **Contexto Geral** e **Contexto Específico**. Estes contextos servirão para delimitar a área de atuação e criar grupamentos de requisitos posteriormente.

Após esse primeiro passo, o ER relaciona todos os requisitos que conseguiu extrair do cliente, e classifica-os de acordo com a funcionalidade do requisito, tratando-os, de acordo com Sommerville [9], como **Requisitos Funcionais** e **Requisitos Não-Funcionais**.

Buscando um aumento na padronização da descrição dos requisitos que estão sendo especificados, o processo propõe que o ER tenha disponíveis opções de palavras válidas, já utilizadas e de mesmo contexto para sua escolha à medida que os requisitos são escritos, fazendo com que ocorra uma reutilização de palavras e a descrição dos requisitos tenha um maior nível de padronização e qualidade.

Em seguida, ocorre a fase de Validação, que de acordo com níveis de rigorosidade pré-definidos pelo ER, avaliam e validam a descrição dos contextos gerais e específicos e a descrição dos requisitos. Os níveis de rigorosidade levam em consideração a quantidade de palavras com inconsistências identificadas.

Caso essa validação seja **negativa**, serão oferecidas sugestões para que o ER altere os itens (contexto geral, específico e requisitos) que contém inconsistências, relativas à descrição dos requisitos, e após essas alterações possa fazer uma nova validação.

Caso a validação seja **positiva**, todos os itens poderão ser armazenados, primeiramente em uma Lista de Requisitos, para que o ER tenha disponível e com fácil visualização os requisitos que já foram especificados, e em seguida armazenados em um Banco de Dados, nomeado de Banco de Dados Histórico de Requisitos (BDHR).

Dentro da questão de validação, a rastreabilidade dos requisitos será feita através do contexto em que o requisito está inserido, do projeto ao qual está relacionado e das palavras-chaves que fazem parte da sua descrição. As palavras descritas nos contextos gerais e específicos poderão ser utilizadas para verificar a descrição dos requisitos, minimizando a chance de especificação incorreta de contextos e requisitos.

Após o término de todas as reuniões, e todos os requisitos definidos e validados seria então possível a criação de um Documento de Requisitos de Software (DRS).

Este processo não especifica um modelo ou template padrão, por não ser a intenção do trabalho, apenas oferece uma lista contendo todos os requisitos validados e armazenados anteriormente, atribuídos a um projeto de software.

A Fig. 2 exemplifica o processo proposto por este artigo.

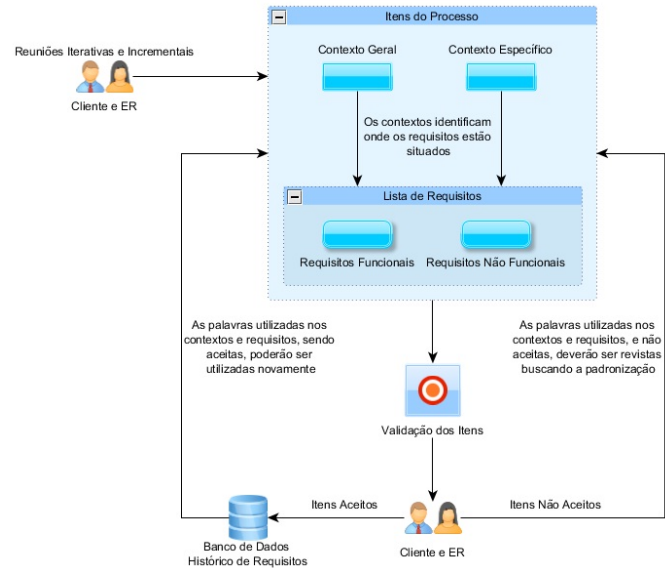


Figura 2. Processo de Engenharia de Requisitos Proposto.

A finalidade de armazenar os requisitos num BDHR é proporcionar à organização, e principalmente ao ER, a escolha de palavras, itens e contextos já utilizados e validados, diminuindo as chances de erros e inconsistências na descrição dos requisitos.

É importante salientar que o objetivo ao propor um processo que utilize um processamento de linguagem natural juntamente com uma revisão de requisitos é auxiliar ao máximo o ER no momento da criação dos requisitos, ou seja, quando eles estão sendo escritos pela primeira vez.

Utilizar puramente um método de processamento de linguagem natural que avalie as palavras e seu significado não é o objetivo deste trabalho, muito menos utilizar uma técnica de revisão de requisitos que apenas aponte erros e inconsistências.

A ideia central é proporcionar ao ER a utilização de palavras, em linguagem natural, que já tenham sido validadas, dentro de contextos de trabalho específicos e que apresentem a qualidade necessária para uma especificação de requisitos padronizada.

## V. AVALIAÇÃO COMPARATIVA

A análise comparativa entre o processo proposto e outros processos da literatura é mostrada nesta seção, demonstrando a importância a caracterização deste estudo. Juntamente com a Tabela I, apresentada na seção III, incluiu-se o processo proposto para facilitar a avaliação. A Tabela II mostra o comparativo entre os processos.

Utilizando este comparativo e as ideias aqui apresentadas, surgem alguns benefícios advindos da utilização do processo proposto num desenvolvimento de software.

TABELA II. COMPARATIVO ENTRE TRABALHOS.

|  | Chen et al. (2010)   | Cabral et al. (2008)   | Pandey et al. (2010)  | Processo Proposto   |
|--|--|--|---|---|
| <b>Identificação dos Requisitos</b>                | Stakeholders criam os requisitos.  | Não especificado no trabalho.  | São identificados requisitos "crus" de pontos de vista de stakeholders. | Stakeholders e ER durante reuniões identificam os requisitos.                                 |
| <b>Análise de Requisitos</b>                       | Não é feita uma análise dos requisitos.  | Não é especificado no trabalho quem faz a análise dos requisitos.                                  | Um Analista analisa os requisitos "crus" e cria requisitos de sistema.  | O ER faz a análise dos requisitos baseado na sua experiência e nos contextos.                 |
| <b>Especificação de Requisitos</b>                 | A especificação vem da identificação dos requisitos feita pelos stakeholders, escrita em LN. | Os requisitos já estão especificados, contidos no Documento de Requisitos.                         | A especificação é feita através do SRS.                                 | A especificação é feita pelo ER, utilizando LN, com a opção de escolha de palavras validadas. |
| <b>Validação de Requisitos</b>                     | Não é objetivo do trabalho a validação os requisitos.  | Foca na validação dos requisitos utilizando para isso técnicas de leitura e revisão de requisitos. | A validação é feita analisando o SRS juntamente com os stakeholders.    | Os requisitos são validados por Stakeholders e ER através de níveis de rigorosidade.          |
| <b>Utiliza agrupamento de Requisitos</b>           | Sim, utiliza Domínios Gerais e Específicos.  | Não  | Sim, em níveis do sistema.  | Sim, utilizam-se dois níveis: Contextos e Tipos.  |
| <b>Utiliza padronização de Requisitos</b>          | Não  | Não  | Não   | Sim   |
| <b>Qualidade dos Requisitos</b>                    | Sim, busca o aumento da qualidade dos próprios requisitos.                                   | Sim, busca o aumento da qualidade da validação dos requisitos.                                     | Sim, busca o aumento da qualidade do processo todo.                     | Sim, busca o aumento da qualidade na validação e descrição dos requisitos.                    |
| <b>Nível de Participação do Cliente</b>            | Alto, identificando e especificando os requisitos.   | O trabalho não cita como foi a participação dos stakeholders.                                      | A participação dos stakeholders deve ser constante.                     | Alto, identificando e validando os requisitos.  |
| <b>Nível de Interação manual do modelo</b>         | Médio. O modelo utiliza ajuda computacional para o tratamento dos requisitos.                | Alto. O modelo utiliza técnicas que necessitam de muita interpretação humana.                      | Alto. O modelo não cita possibilidades de automação do processo.        | Médio. O modelo apresenta possibilidade de automação para o tratamento dos requisitos.        |
| <b>Reutilização Parcial ou Total de Requisitos</b> | Não  | Não  | Não   | Sim   |

### A. Benefícios Esperados

Em estudos desenvolvidos, foram identificados benefícios dentro das fases de Especificação e Validação de Requisitos que podem ser esperados quando o processo proposto é aplicado em um processo de desenvolvimento de software.

Alguns desses benefícios esperados são:

- **Aumento na qualidade:** a qualidade dos requisitos aumentará significativamente através da Validação dos requisitos apontada pelo processo, pois dois ER diferentes poderão criar a mesma descrição de requisitos baseados no BDHR;
- **Diminuição de redundâncias:** os requisitos serão tratados através de contextos. Mesmo contendo muitas palavras idênticas, os contextos em que estarão inseridos irão determinar seu significado, tanto para o ER quanto para o *Stakeholder*;
- **Aumento na rastreabilidade:** os requisitos serão gerenciados dentro de contextos gerais e específicos, funcionalidades e projetos. Quando uma palavra-chave já validada é alterada existe a possibilidade de todos os requisitos que contenham a mesma palavra-chave dentro do mesmo contexto serem identificados;
- **Aumento da percepção:** no processo proposto é possível detectar uma situação já vivenciada e sugerir ao ER uma solução, esta solução fica aberta a aceitação ou não por parte do ER;
- **Diminuição de retrabalho:** a descrição e especificação dos requisitos poderão ser feitos na presença do *stakeholder*. Assim caso ocorram discrepâncias quanto ao requisito criado podem ocorrer alterações no mesmo

momento, resolvendo o conflito e melhorando a especificação;

- **Diminuição de riscos:** uma vez validados, e aprovados pelo *stakeholder*, os requisitos são armazenados no BDHR, diminuindo as chances de erros ou inconsistências durante as outras fases do desenvolvimento;
- **Aumento da memória organizacional da empresa:** um item muito importante e faltante em muitas empresas, pois seus projetos são tratados separadamente. Esse processo propõe manter todos os requisitos à disposição do ER no início de qualquer projeto e dentro do contexto especificado por ele.

## VI. CONCLUSÕES E TRABALHOS FUTUROS

Este artigo mostra como está o cenário de desenvolvimento da área de Engenharia de Requisitos, mais precisamente na Especificação e Validação de Requisitos. Muita pesquisa ainda é necessária para que descrição e a documentação dos requisitos sejam feitas com maior nível de qualidade, utilizando processos e técnicas para que *Stakeholder* e ER possam ficar seguros de que o requisito criado corresponde exatamente ao requisito desejado.

A Engenharia de Requisitos é uma das primeiras fases de um desenvolvimento de software, sendo talvez a mais importante, pois através dela o software é definido. Portanto, todo o auxílio para que os requisitos estejam em conformidade com o desejo do *stakeholder* é de grande valia tanto para analistas, engenheiros e desenvolvedores quanto para a organização em si.

O processo proposto visa atender esta demanda e prover todos os benefícios citados anteriormente.

Como trabalhos futuros busca-se desenvolver e incluir no processo um DRS padrão para armazenar a especificação de requisitos, prover alinhamento entre o processo proposto e a Gerência de Requisitos, para que, além de tratados, os requisitos possam ser gerenciados. A aplicação do processo a um desenvolvimento de software, utilizando uma ferramenta que implemente as ideias apresentadas também é uma ideia para a continuação deste trabalho, avaliando assim os resultados obtidos e medindo a efetividade e correspondência com todos os benefícios esperados.

## REFERÊNCIAS

- [1] G. Booch, J. Rumbaugh, Jacobson, I. *UML: Guia do Usuário*. 2 ed. Rio de Janeiro. Elsevier, 2005.
- [2] M. S. Cabral, F. Alencar, J. Castro, O. Pastor, J. Sánchez. "Aplicação de Técnicas de Leitura durante a Análise de Requisitos", WER08 - Workshop em Engenharia de Requisitos, Barcelona, Catalonia, Spain, Setembro 12-13, 2008, pp 193-204.
- [3] H. Chen, K. He, P. Liang, R. Li. "Text-based requirements preprocessing using nature language processing techniques", International Conference on Computer Design and Applications (ICDDA), Qinhuaodao, Hebei, China, June 25-27, 2010, pp 14-18.
- [4] E. Costa e A. Simões, *Inteligência Artificial - Fundamentos e Aplicações*, 2 ed. FCA - Editora de Informática, 2008.
- [5] M. E. Fagan, *A history of software inspections*. New York, NY, USA: Springer-Verlag, 2002.

- [6] D. Pandey, A. K. Ramani, U. Suman, "An Effective Requirement Engineering Process Model for Software Development and Requirements Management". International Conference on Advances in Recent Technologies in Communication and Computing, IEEE Press, 2010.
- [7] R. S. Pressman, *Engenharia de Software - Uma Abordagem Profissional*, 7 ed. McGraw-Hill, 2011.
- [8] M. Sayão, *Verificação e Validação em Requisitos: Processamento da Linguagem Natural e Agente*. Rio de Janeiro, Abril, 2007.
- [9] I. Sommerville, *Engenharia de Software*, 8 ed. Addison Wesley, 2007.



**José André Dorigan** é graduado em Ciência da Computação pela Universidade Estadual de Londrina (UEL), Londrina, Paraná, Brasil, em 2010. Concluiu o mestrado em Ciência da Computação pela Universidade Estadual de Londrina (UEL) em 2013. Entre os anos de 2009 e 2010 participou da Fábrica de Projetos de TIC - GAIA no Departamento de Computação, na Universidade Estadual de Londrina, desenvolvendo projetos na área de Engenharia de Software. Atualmente é Professor Colaborador da Universidade Estadual de Maringá (UEM), ministrando aulas nos cursos de graduação. Suas pesquisas se concentram na área de Engenharia de Software, na subárea de Engenharia de Requisitos.



**Rodolfo Miranda de Barros** é graduado em Ciência da Computação pela Universidade Federal de São Carlos (UFSCar) e em Administração de Empresas pela Universidade Estadual de Londrina (UEL). Concluiu o mestrado em Ciência da Computação pela Universidade Federal do Rio Grande do Sul (UFRGS) em 1997 e o doutorado em Engenharia Elétrica pela Universidade Estadual de Campinas (Unicamp) em 2008. Atualmente é Professor Adjunto B da Universidade Estadual de Londrina, ministrando aulas no curso de graduação e mestrado. Coordena os projetos Gerenciamento de Serviços de TI (GERTI) e a Fábrica de Projetos de TIC - GAIA.