

PONTIFÍCIA UNIVERSIDADE CATÓLICA

BRUCE DEON JASPER

**UM ESTUDO SOBRE OS OBSTÁCULOS NA ELICIAÇÃO DE REQUISITOS DE
SOFTWARE**

São Paulo
2014

BRUCE DEON JASPER

**UM ESTUDO SOBRE OS OBSTÁCULOS NA ELICIAÇÃO DE REQUISITOS DE
SOFTWARE**

Trabalho de Conclusão de Curso apresentado
à Pontifícia Universidade Católica como requisito
parcial para obtenção do título de Especialista em
Engenharia de Software.

Orientador: Prof. Dr. Ítalo Santiago Vega

São Paulo
2014

A meus avós, sem o qual eu jamais poderia ter chegado onde estou; a meus pais e a meu irmão por todo apoio e confiança na realização deste trabalho.

AGRADECIMENTOS

A Deus, fonte de toda sabedoria, pela força e pela coragem que me concedeu, permanecendo ao meu lado em todo o percurso desta caminhada.

A Jesus, por sua companhia diária em minha vida. Meu exemplo e inspiração.

Ao Dr. Ítalo Santiago Vega minha eterna gratidão, por ter sido orientador persistente e presente, que, com diretrizes seguras, embora às vezes enigmáticas, me orientou na direção de um trabalho qualificado e diferenciado, saindo do lugar comum e das respostas fáceis e incentivando o raciocínio lógico, mostrando-me o caminho do conhecimento.

Ao Ms. Luiz Antonio de Lima, por me orientar em meu trabalho de graduação, como se passasse os primeiros ensinamentos a um filho. Seu esforço e dedicação me capacitaram a chegar até aqui.

Ao Dr. Carlos Eduardo de Barros Paes por todo suporte ao longo da elaboração deste trabalho.

Ao Ms. Wilson Wistuba, que colaborou e me aconselhou ainda na fase de escolha da pesquisa.

Ao Ms. Aluizio Saiter, que colaborou e me aconselhou ainda na fase de escolha da pesquisa.

À Amaryllis Beltrani Rochet por todo carinho, apoio e incentivo durante a elaboração desta pesquisa.

A todos que colaboraram, direta ou indiretamente, no desenvolvimento desta pesquisa.

De tempos em tempos, os homens tropeçam na verdade, mas a maioria deles se levanta e segue adiante como se nada tivesse acontecido (Winston Churchill).

RESUMO

Ao longo das décadas incontáveis métodos e ferramentas foram desenvolvidos buscando resolver os problemas observados da engenharia de software. Devido à sua reconhecida importância, a área da engenharia de requisitos tem recebido atenções cada vez maiores dos pesquisadores. Pesquisas demonstram a importância de se elicitar e gerenciar os requisitos de software adequadamente e os impactos desta etapa nos projetos de TI. Práticas recomendadas e Fatores de Sucesso são estudados buscando evolução científica e sucessos nos projetos empreendidos nas mais diversas áreas. Todavia, ainda existem muitos problemas não resolvidos na execução das atividades desta área, principalmente na eliciação de requisitos de software. Esta pesquisa buscou compreender os obstáculos existentes na engenharia de requisitos e contribuiu ao apresentar como os métodos formais podem auxiliar, em especial os métodos orientados a metas, uma abordagem que permite elicitar e raciocinar sobre os requisitos técnicos e relacioná-los a metas de negócio dos *stakeholders*. Em conjunto com métodos tradicionais, esta abordagem pode atuar para resolver alguns dos problemas apresentados. Como sugestões de prosseguimento foram recomendadas a aplicação prática de abordagens de engenharia de requisitos orientadas a metas na eliciação de requisitos de software, entre outras etapas do ciclo de desenvolvimento, apresentando estudos de casos reais, a execução de pesquisas direcionadas a mostrar vantagens e desvantagens e fatores de sucesso de abordagens de engenharia de requisitos orientados a metas e a execução de estudos orientados à construção de uma linguagem de modelagem de metas que unifique as diversas metodologias existentes, o que deve favorecer seu aprendizado e aplicação pelos engenheiros de requisitos.

Palavras-chave: Engenharia de Requisitos, Eliciação de Requisitos, Engenharia de Requisitos Orientada a Metas, Formalização de Requisitos

ABSTRACT

Over the decades countless methods and tools have been developed aiming to solve the observed problems of software engineering . Because of its recognized importance, the area of requirements engineering has received increasing attention from researchers. Research demonstrates the importance of eliciting and managing software requirements properly and the impacts of this step in IT projects. Best Practices and Success Factors are studied seeking scientific developments and successes in projects undertaken in several areas. However, there are still many unsolved problems in the implementation of activities in this area, mainly in software requirements elicitation. This research sought to understand the obstacles in requirements engineering and contributed to present how formal methods can help, especially goal-oriented methods, an approach that allows eliciting and reasoning about technical requirements and relate them to business goals of stakeholders. In conjunction with traditional methods, this approach can serve to resolve some of the problems presented. As suggestions for further research were recommended the practical application of goal-oriented requirements engineering approaches in software requirements elicitation, among other steps of the development's cycle, presenting real case studies, performing research directed to show advantages and drawbacks and successful approaches to goal-oriented requirements engineering and execution of studies oriented to building a modeling language that unifies the various existing methodologies, which should facilitate their learning and application by requirements engineers.

Keywords: Requirements Engineering, Requirements Elicitation, Goal-Oriented Requirements Engineering, Requirements Formalization

LISTA DE FIGURAS

Figura 1 - Visão Geral da Monografia.....	15
Figura 2 - Modelo do Processo de Engenharia de Requisitos	19
Figura 3: Tipologia dos Desafios de Comunicação na Eliciação de Requisitos	25

LISTA DE TABELAS

Tabela 1 - Dificuldades na Eliciação de Requisitos25

Tabela 2 - Melhores Práticas e Benefícios na Engenharia de Requisitos.....28

Tabela 3 - Vantagens e Desvantagens dos Métodos Semiformais30

SUMÁRIO

<u>INTRODUÇÃO</u>	11
JUSTIFICATIVA DA PESQUISA	14
ESTRUTURA DOS DEMAIS CAPÍTULOS DA MONOGRAFIA	15
DELIMITAÇÕES DA PESQUISA	16
<u>CAPÍTULO 1 – REVISÃO BIBLIOGRÁFICA</u>	17
1.1 O QUE É UM REQUISITO	17
1.2 O QUE É A ENGENHARIA DE REQUISITOS	18
<u>CAPÍTULO 2 – OBSTÁCULOS DA ENGENHARIA DE REQUISITOS</u>	21
2.1 CONTEXTO	21
2.2 O FATOR HUMANO	21
2.3 OBSTÁCULOS NA ENGENHARIA DE REQUISITOS	22
2.4 PROBLEMAS DA ELICIAÇÃO DE REQUISITOS	23
2.5 TÉCNICAS E FATORES DE SUCESSO	26
<u>CAPÍTULO 3 – ELICIAÇÃO DE REQUISITOS UTILIZANDO LINGUAGEM ORIENTADA A METAS</u>	29
3.1 FORMALIZAÇÃO DE REQUISITOS	29
3.2 ENGENHARIA DE REQUISITOS ORIENTADA A METAS	30
3.3 GORE APLICADA À ELICIAÇÃO DE REQUISITOS	32
<u>CONCLUSÃO</u>	35
CONTRIBUIÇÕES DESTA PESQUISA	35
LIMITAÇÕES DO ESTUDO E SUGESTÕES PARA NOVAS PESQUISAS	35
CONSIDERAÇÕES FINAIS	36
<u>REFERÊNCIAS BIBLIOGRÁFICAS</u>	37

INTRODUÇÃO

A importância da correta definição dos requisitos em um sistema de software já era demonstrada desde antes da criação do termo “Engenharia de Software” (BOEHM, 2006). Todavia, esta tarefa tem se demonstrado uma das mais árduas e complexas ao longo da história. Glass (2005) pontua diversas evoluções e revoluções ocorridas na história do desenvolvimento de software, onde a introdução de novas ferramentas e tecnologias foi cercada de promessas que não se cumpriram em sua totalidade, como a programação estruturada e as abordagens orientadas a objeto. Barry (2004) analisa que no momento que um novo método que é introduzido ele consegue solucionar com sucesso alguns problemas para o qual foi concebido mas acaba agregando também alguma etapa que se torna difícil de ser executada, sendo evitada a sua execução, concluindo que novas linguagens, modelos de desenvolvimento e ferramentas criadas com o intuito de aperfeiçoar a programação não devem ser o foco e que os esforços de estudo devem ser direcionados à área de requisitos, entre outras.

Ao refletir a complexidade em se entregar um projeto de software no tempo e custo previsto, DeMarco (2010, p. 95) expõe que “O desenvolvimento de software é e sempre será de alguma forma algo experimental. A forma atual de construção de software não é necessariamente experimental, mas sua concepção é”. Estudo de Glass e Vessey (1994) já evidenciava a importância da criatividade e do intelecto para o desenvolvimento de software. Filman (2005, p. 4) é categórico; “O desenvolvimento de software é uma arte”.

Morgan (2005, p. 130) questiona “Como pode uma indústria sobreviver ou continuar a demandar bilhões de dólares gastos pelo setor público e privado sem algum entendimento concreto e consenso sobre como ele funciona?”. Bjørner (2005) afirma que a noção de software defeituoso tornou-se natural levando a um entendimento de que é uma característica do produto e trazendo descrédito à área de engenharia de software.

Um Estudo do Standish Group (1994) procurou entender quais seriam os fatores para um projeto exceder prazo, custo e/ou não entregar todas as funcionalidades especificadas. Dos fatores levantados, a falta de informações dos usuários (12.8%), os requisitos e as especificações incompletas (12.3%) e a mudanças nos requisitos e especificações (11.8%)

foram os fatores mais citados. Analisando esses dados, Leffigwell e Widrig (2003, p. 7) concluem: “Aparentemente ao menos um terço dos projetos de desenvolvimento são afligidos por problemas diretamente relacionados à obtenção, documentação e gerenciamento de requisitos”. Outros estudos também endossam esta afirmação:

- Estudo realizado sobre a prática da engenharia de software em doze empresas britânicas indicou que problemas relacionados a requisitos respondem por 40% de todos os problemas no desenvolvimento de software (Hall et. al, 2002).
- Weinberg (1997) mostra que até 60% dos erros de software se originam do processo de engenharia de requisitos.
- Pesquisa do Instituto Europeu de Software em 3800 organizações em 17 países concluiu que a maioria dos problemas de software percebidos são na área de especificação e gerenciamento de requisitos (European Software Institute, 1996, citado em Van Lamsweerde, 2000).

Embora não haja consenso sobre a aplicabilidade das técnicas utilizadas na pesquisa do Standish Group (EVELEENS e VERHOEF, 2010; GLASS, 2006; EBERT, 2005), é comprovado dentro da engenharia de software a dificuldade de se gerenciar os requisitos. Recentemente, estudo do IAG (2009) levantou que a principal causa de falha de 68% dos projetos de TI são requisitos pobres. Tran e Kasser (2005) pontuam que requisitos escritos de forma pobre são aqueles que não atendem aos requisitos aplicáveis à escrita de requisitos. Wiegers (1999) e Hooks (1993) tratam desta discussão.

Ebert (2005) aponta que as alterações de requisitos estão quase sempre correlacionadas a atrasos nos projetos que o sucesso de um projeto depende de um bom gerenciamento de requisitos. Wahono (2003) pontua os problemas mais sérios relacionados ao desenvolvimento de software estão relacionados aos requisitos. Van Lamsweerde (2008) aponta que os tipos de erros mais sérios envolvem requisitos incompletos, inadequados, inconsistentes, não mensuráveis ou ambíguos, causando atrasos na entrega dos projetos, excesso de custos, falha no atendimento às expectativas e degradação no ambiente controlado pelo software.

Este tópico foi discutido em Brooks (1987, p. 17) onde era caracterizado que “a parte mais difícil de construir um sistema de software é decidir com precisão o que construir. Nenhuma outra parte do trabalho conceitual é tão difícil como definir os requisitos técnicos detalhados,

incluindo todas as interfaces com pessoas, máquinas e outros sistemas de software. Nenhuma outra parte do trabalho compromete o sistema resultante se realizada errado. Nenhuma outra parte é mais difícil de corrigir posteriormente”. Complementando este raciocínio, Leffigwell e Widrig (2003) conceituam que os erros relacionados aos requisitos são provavelmente o tipo de erro mais comum, permeando todas as fases posteriores à sua definição como também são os tipos de erros mais caros para se corrigir. Esta afirmação se torna ainda mais importante com estudos como o de Lientz e Swanson (1978) apontando que 80% das alterações realizadas durante a fase de manutenção são alterações de requisitos, sendo os outros 20% correções de erros do software.

Os requisitos de software são objeto de estudo da área da engenharia de requisitos, definida por Zave (1995, p. 214) como: "ramo da engenharia de software preocupado com as metas do mundo real para funções e restrições de um sistema de software. Também se preocupa com os relacionamentos desses fatores para especificar exatamente o comportamento do software e sua evolução ao longo do tempo e entre famílias de software.". Em Arif et. al (2010) é encontrado o levantamento e análise de vários processos de engenharia de requisitos existentes (Linear, Iterativo e Espiral), sendo que Pandey et. al (2010) divide o processo em quatro fases: Eliciação e Desenvolvimento de Requisitos, Documentação dos Requisitos, Validação e Verificação dos Requisitos e Planejamento e Gerenciamento dos Requisitos.

O foco desta pesquisa está na fase de eliciação dos requisitos, particularmente apresentando os problemas existentes nesta etapa e como uma abordagem formal pode auxiliar neste processo. Rehman et al. (2013, p. 46) realiza análise de metodologias, técnicas e ferramentas utilizadas na engenharia de requisitos e conclui que a "Eliciação de Requisitos é mais uma arte do que uma ciência devido às habilidades e experiência necessária para realizar esta tarefa em uma variada natureza de projetos de desenvolvimento de software.". Brooks (1987) enfatiza que este processo é a função mais importante executada pelo construtor do software para o cliente.

Partindo da discussão apresentada, o problema de pesquisa que este trabalho busca discutir é:

“Em que aspectos uma abordagem de formalização de requisitos pode auxiliar no processo de eliciação de requisitos?”

JUSTIFICATIVA DA PESQUISA

Interesse acadêmico: Esta pesquisa aborda questões relativas às dificuldades encontradas no processo de eliciação de requisitos de software e como a utilização de uma abordagem formal pode auxiliar neste processo. A literatura evidencia que estes assuntos tem obtido grande importância na área de engenharia de software. Um dos principais fatores é o atual insucesso dos projetos de desenvolvimento de software. Diversas pesquisas identificam os requisitos como sendo o ponto crítico neste processo. Davis et. al. (2006, p.78) expõem que “apesar de anos de prática e um conjunto de metodologias de análise e design, ferramentas e técnicas, os sistemas desenvolvidos frequentemente falham em alcançar as funcionalidades desejadas por seus usuários”. A pesquisa visa acrescentar subsídios para o estudo e aplicação de uma abordagem formal, orientada a metas, no processo de eliciação de requisitos, visando aperfeiçoar esta etapa do processo.

Interesse pessoal: O interesse vai além do aprimoramento pessoal, mas também na produção de uma obra em sintonia com os estudos atuais na área de engenharia de software. Em minha vivência profissional pude vivenciar na prática vários problemas abordados neste trabalho. Nota-se na literatura a necessidade de estudos mais aprofundados e interdisciplinares sobre os requisitos, como apontado por Nuseibeh e Easterbrook (2000), que ressaltam que devido ao não atendimento de requisitos do cliente devido, parcialmente, a uma engenharia de requisitos ineficiente, a visão desta área como uma atividade burocrática, contratual e que apenas consome tempo está se transformando, sendo reconhecida como crítica no processo de desenvolvimento de sistemas.

ESTRUTURA DOS DEMAIS CAPÍTULOS DA MONOGRAFIA

O capítulo 1 aborda a revisão bibliográfica sobre o tema dessa pesquisa e está subdividida em duas seções: A seção 1.1 explora o que seria um requisito de software e a seção 1.2 busca explicar o que seria a engenharia de requisitos.

O capítulo 2 trata dos obstáculos existentes na engenharia de requisitos e se encontra dividido em 5 seções: a 2.1 contém a contextualização deste tópico, a 2.2 explana sobre os fatores humanos do processo, a 2.3 apresenta os obstáculos e dificuldades da engenharia de requisitos, a 2.4 se aprofunda nas dificuldades da etapa de eliciação de requisitos e a 2.5 discute sobre as técnicas de eliciação e fatores de sucesso nas tarefas de engenharia de requisitos.

O capítulo 3 apresenta na seção 3.1 a discussão sobre a formalização de requisitos de software. Na seção 3.2 é apresentada a engenharia de requisitos orientada a metas e na seção 3.3 é discutida a utilização de uma abordagem orientada a metas no processo de eliciação de requisitos e sua atuação sobre alguns dos problemas apresentados nesta pesquisa.

Na conclusão são apresentadas as considerações sobre a pesquisa e são propostas questões correlatas ao tema de pesquisa que poderão vir a ser abordadas em pesquisas futuras.

A Figura 1 apresenta a visão geral dos assuntos abordados nesta monografia.

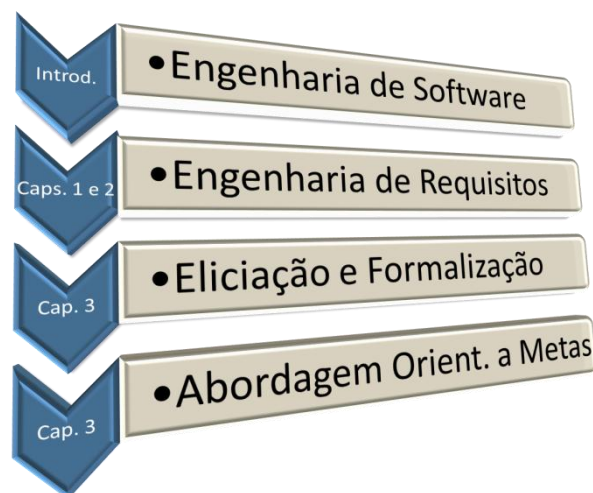


Figura 1 - Visão Geral da Monografia

DELIMITAÇÕES DA PESQUISA

Esta pesquisa contemplou os obstáculos no processo de engenharia de requisitos com foco na eliciação e formalização de requisitos de software, visando à utilização de uma abordagem formal para auxiliar este processo no intuito de obter um artefato que expresse melhor o entendimento dos requisitos. O foco não foi o de utilizar a linguagem formal ao longo do ciclo de vida do desenvolvimento de software nem o gerenciamento de suas alterações. Fugiu da abrangência dessa pesquisa a aplicação prática desta dos métodos propostos assim como estudos em profundidade sobre quais seriam os fatores causadores dos problemas de eliciação de requisitos, tópico este de grande abrangência e que necessitaria de estudos em ciências humanas como a psicologia e a sociologia vista a natureza desta atividade.

CAPÍTULO 1 – REVISÃO BIBLIOGRÁFICA

1.1 O que é um requisito

Para esta monografia o termo requisito será sempre referenciado como sendo um requisito de software. Algumas definições para este termo são:

- Um requisito é uma declaração que traduz ou expressa uma necessidade e suas restrições e condições associadas. (ISO/IEC/IEEE, 2011).
- “Um requisito de software é uma propriedade a qual deve ser exposto de forma a resolver um problema no mundo real” (IEEE, p. 33, 2004).
- “1) Uma condição ou capacidade necessária por um usuário para resolver um problema ou atingir um objetivo
2) Uma condição ou capacidade que deve ser atingida ou possuída por um sistema ou componente de sistema para satisfazer um contrato, padrão, especificação ou outro documento imposto formalmente.
Uma representação documentada de uma condição ou capacidade como em 1 ou 2”.
(IEEE, p. 62, 1990).

Para esta monografia será adotada a terceira definição por contextualizar o termo por parte do usuário e do sistema, citando explicitamente a existência de um documento, um meio de se comunicar as necessidades.

A função dos requisitos é dirigir a visão de um produto ou solução e são concebidos antes do produto ser definido (EBERT, 2005). Ernst et. al. (2011) declara que os requisitos são utilizados para unir o a visão dos problemas do mundo real, o ponto de vista dos *stakeholders*, ao mundo dos desenvolvedores, focado nas implementações.

Brooks (1987, p. 17) sumariza a importância da interação cliente e analista afirmando que “a função mais importante que o construtor de software executa para o cliente é a extração e refinamento dos requisitos do produto de forma iterativa.”.

1.2 O que é a Engenharia de Requisitos

Na literatura são encontradas várias definições para a área de engenharia de requisitos dentro da Engenharia de Software. Nuseibeh e Easterbrook (2000) expõem que a engenharia de requisitos é o processo de descobrir o propósito para qual um sistema é planejado, identificando os *stakeholders* e suas necessidades, documentando-as de forma passível a análise, comunicação e implementação. Este processo é multidisciplinar e focado nas interações humanas, utilizando várias técnicas e ferramentas em diferentes fases do desenvolvimento e para diferentes tipos de domínios de aplicação.

Van Lamsweerde (2000) expõe que o interesse da engenharia de requisitos está na eliciação, validação, especificação, consolidação e evolução de objetivos, funcionalidades, qualidades e restrições que um *software* deve ter para atingir um objetivo.

Pandey et. al. (2010) divide o processo em quatro fases:

1. Eliciação e Desenvolvimento de Requisitos: Possui foco em examinar e obter os requisitos desejados para o sistema de diferentes pontos de vistas. Com frequência os requisitos são mal interpretados pelo analista, tópico que será explorado no próximo capítulo.
2. Documentação dos Requisitos: Após a coleta dos requisitos é elaborado um documentação formal conhecida como especificação de requisitos de software. Macaulay (1996) pontua que este documento deve conter uma declaração concisa dos requisitos que o software deve satisfazer.
3. Validação e Verificação dos Requisitos: Após a elaboração do documento de especificação de requisitos as partes envolvidas precisam concordar que os requisitos corretos foram estabelecidos (validação) e que estes requisitos estão descritos corretamente (verificação).
4. Planejamento e Gerenciamento dos Requisitos: Esta fase se preocupa em controlar e rastrear as mudanças nos requisitos estabelecidos, as dependências entre os requisitos e entre os outros artefatos produzidos no processo de desenvolvimento. A rastreabilidade entre os artefatos e as mudanças nos requisitos deve ser gerenciada continuamente.

A Figura 2 detalha o processo exposto.

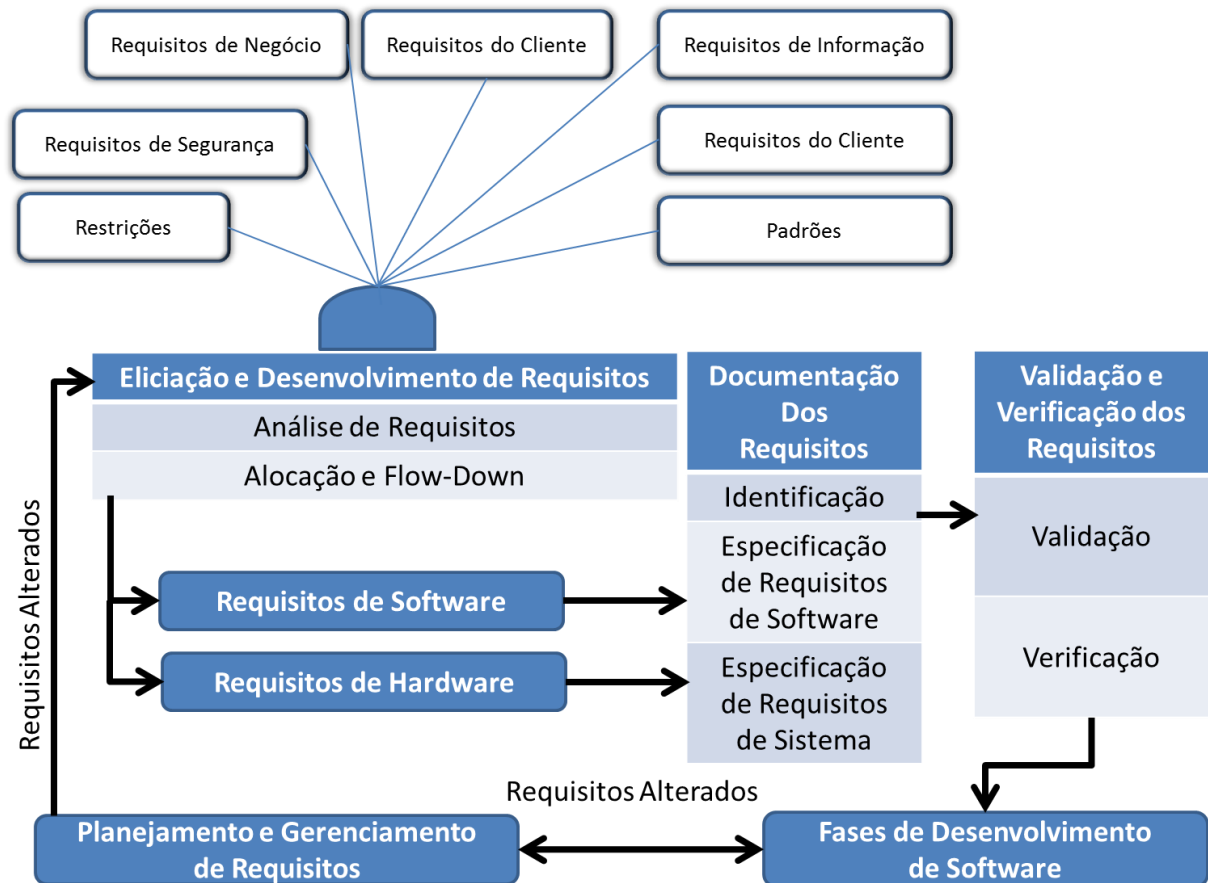


Figura 2 - Modelo do Processo de Engenharia de Requisitos

Fonte: Reproduzida de Pandev et. al. (2010, p. 288)

Em complemento à definição anterior segue-se a de Wahono (2003, p. 56), que pontua a questão das mudanças: “É uma forma sistemática de eliciar, organizar e documentar os requisitos de um sistema e um processo que estabelece e mantém concordância entre o cliente e o time de projeto sobre a mudança dos requisitos do sistema”.

Zave (1995, p. 214) oferece outra definição para a engenharia de requisitos: “É o ramo da engenharia de software preocupado com as metas do mundo real para funções e restrições de um sistema de software. Também se preocupa com os relacionamentos desses fatores para especificar exatamente o comportamento do software e sua evolução ao longo do tempo e entre famílias de software”. A preocupação com a evolução do software é citada por Hofmann e Lehner (2001), que pontuam que a engenharia de requisitos é um processo incremental que leva a descoberta progressiva dos requisitos do projeto.

Para esta monografia o entendimento dos conceitos apresentados é fundamental para os capítulos seguintes. A visão de que a engenharia de requisitos pode ser entendida como um processo com atividades claramente definidas na qual há a interação de seres humanos é necessária para o tópico abordado no próximo capítulo, que explanará sobre as dificuldades existentes no processo de engenharia de requisitos.

CAPÍTULO 2 – OBSTÁCULOS DA ENGENHARIA DE REQUISITOS

2.1 Contexto

Zave (1995) caracteriza a área de engenharia de requisitos como a mais informal, interdisciplinar e subjetiva das áreas pesquisadas pelos cientistas da computação. A literatura apresenta diversos fatores que agregam complexidade à execução das tarefas de engenharia de requisitos e suas consequências.

Yourdon (in Leffigwell e Widrig, 2003) contextualiza o problema dos requisitos como sendo “o problema da rocha”. Na analogia proposta, quando os requisitos descrevem “entregue-me uma rocha” e o desenvolvedor realiza esta entrega, a resposta do cliente é similar a “sim, mas o que eu realmente queria era uma pequena rocha azul”. Em projetos reais, o problema de se especificar uma “rocha aceitável” para ambos tem sua complexidade aumentada onde normalmente há mais de duas pessoas buscando essa solução. De acordo com Brooks (1987), o simples pedido de “faça o novo sistema trabalhar como o nosso antigo sistema” é, na verdade, muito simples e nunca se quer exatamente isso.

Quispe et. al. (2010) argumentam que a utilização incorreta de práticas de engenharia de requisitos tem como consequência o retrabalho, problemas de comunicação e coordenação e visão incorreta do estado do projeto.

2.2 O fator humano

Como pontuado no capítulo anterior, a engenharia de requisitos é uma atividade com participação de pessoas. Parnas e Clements (1986) expõem que a única forma de se eliminar erros cometidos por pessoas é evitar a utilização de pessoas, algo não passível de aplicação na forma pela qual os sistemas de software são desenvolvidos atualmente.

A literatura relata que a participação humana no processo é decisiva para a ocorrência de erros. Rupp (2002) aponta que a comunicação é o maior problema da engenharia de requisitos. Em estudo de caso, Bjarnason et. al. (2011) concluem que erros de comunicação

durante a fase engenharia de requisitos podem trazer sérias consequências, como esforço desperdiçado, problemas de qualidade, o não atingimento das expectativas dos clientes e a comunicação incorreta de quais requisitos o produto final irá possuir. Morley e Miller (1994) pontuam que os aspectos sociológicos são fundamentais para o desenvolvimento de software e a etapa de especificação de requisitos é crítica para o estabelecimento de relações entre usuários e técnicos, na qual podem ocorrer conflitos entre estes grupos. Thew e Sutcliffe (2008) pontuam que características como valores pessoais, motivações e emoções influenciam nas atividades de engenharia de requisitos.

Viller (1999) disserta sobre o relacionamento da engenharia de requisitos com ciências humanas como a psicologia cognitiva e apresenta três perspectivas de análise para os erros produzidos por seres humanos, aplicáveis não só a esta área:

- Erros Individuais: ocorrência de omissões, falsos positivos, falsos negativos, interpretação incorreta, falhas de memória, aplicação incorreta de procedimento ou regras que funcionaram no passado ou mesmo a aplicação de procedimentos ou regras incorretas, autoconfiança em excesso, generalização excessiva, entre outros.
- Falhas no desempenho do grupo e perdas do processo: facilitação ou inibição na execução de uma tarefa na presença de outros (um supervisor, por exemplo), alocação inapropriada de recursos humanos no grupo, problemas sócio motivacionais, liderança inapropriada entre outros.
- Problemas Organizacionais: metas incompatíveis, deficiências organizacionais, comunicação inadequada, planejamento ruim, controle e monitoramento inadequado, materiais impróprios, procedimentos operacionais ruins, entre outros.

2.3 Obstáculos na engenharia de requisitos

Van Lamsweerde (2008) lista fatores que tornam a atividade de engenharia de requisitos complexa:

- Cooperação com múltiplos stakeholders com diferentes conhecimentos prévios, interesses e expectativas, frequentemente conflitantes.
- O domínio do problema pode não ser familiar aos envolvidos e sua investigação é informal enquanto sua solução deve ser formal.

- Existência de necessidades implícitas, escondidas e premissas que devem ser formalizadas para compor uma especificação.
- Necessidade de ser avaliar várias alternativas para tomada de decisão sobre formas de se atingir os objetivos do sistema, delegar responsabilidades a cada parte do sistema, adotar medidas contra ameaças externas, entre outras.

Bell e Thayer (1976) concluem que a informação necessária para a construção de pequenos e grandes sistemas frequentemente está incorreta, ambígua, inconsistente ou simplesmente não existe, sendo necessário obtê-los e revisá-los continuamente. Barry (2004) pontua que o real problema da engenharia de software é como lidar com requisitos em constante mudança. McConnel (1996), em seu livro sobre desenvolvimento rápido, elabora uma lista de erros clássicos em projetos de software, entre eles a existência de mais requisitos do que o necessário.

Naz e Khokhar (2009) abordam outras dificuldades:

- As pessoas certas com experiência, conhecimento técnico e habilidades interpessoais podem não estar disponíveis para as atividades de engenharia de requisitos.
- As ideias iniciais sobre o que é necessário frequentemente estão incompletas, excessivamente otimistas, e fixadas nas mentes das pessoas.
- Dificuldade na utilização de ferramentas e métodos para realizar a eliciação de requisitos podem inviabilizar os benefícios de sua utilização.

Pode-se concluir que existe uma gama de problemas a serem estudados, de escopo interdisciplinar à engenharia de software. As seções abaixo analisarão em maior detalhe os tópicos de interesse para essa pesquisa, a etapa de eliciação de requisitos e sua formalização.

2.4 Problemas da eliciação de requisitos

Hickey e Davis (2003) conceituam a etapa de eliciação de requisitos como sendo a forma pela qual os analistas determinam os problemas e as necessidades dos clientes, permitindo aos desenvolvedores construir um sistema que resolva estes problemas e enderecem às necessidades atuais dos clientes. Este processo envolve o aprendizado das necessidades dos usuários e sua comunicação aos construtores de sistema. Kumari e Pillai (2013) apontam que esta fase é a parte mais complexa da engenharia de requisitos. Davis et. al. (2006) explica que

esta fase envolve a interação de usuários finais e analistas para identificar e capturar os processos e dados que devem existir em um novo sistema.

Christel e Kang (1993) agrupam os problemas na eliciação em três grupos:

- Problemas de Escopo: quando os requisitos possuem muito pouca ou excessiva informação. Neste caso se enquadram a má definição dos limites do sistema e detalhamento desnecessário de informações de design.
- Problemas de Entendimento: quando não há entendimento correto dentro de um grupo ou entre grupos, como usuários e desenvolvedores. Esta situação pode ocorrer quando:
 - 1) Os usuários possuem um entendimento incompleto de suas necessidades.
 - 2) Os usuários possuem pouco conhecimento sobre as capacidades computacionais e limitações.
 - 3) Os analistas possuem pouco conhecimento do domínio da aplicação.
 - 4) Usuários e analistas utilizam diferentes linguagens para comunicação.
 - 5) Há omissão de informações “óbvias”.
 - 6) Há visões conflitantes por parte dos usuários.
 - 7) Há requisitos expostos de maneira vaga e não testável (subjetivo).
- Problemas de Volatilidade: devido à natureza mutável dos requisitos permitindo sua evolução ao longo do tempo.

Tsumaki & Tamai (2006) também elencam e categorizam dificuldades apresentadas na eliciação de requisitos, expostas na Tabela 1.

Requisitos Incompletos	
	Entendimento incompleto das necessidades
	Conhecimento Incompleto do domínio da aplicação
	Pouca colaboração dos usuários
	Negligenciar premissas tácitas
Requisitos Incorretos	
	Limites do sistema mal definidos
	Mau entendimento do propósito do sistema
Requisitos ambíguos	
	Utilização de termos sinônimos e homônimos
	Termos não testáveis

Requisitos inconsistentes	
	Intenções não sólidas dos solicitantes
	Visões diferentes de usuários diferentes
Requisitos Mutáveis	
	Flutuação dos requisitos
	Aceitação contínua de mudança nos requisitos
Requisitos em Excesso	
	Desorganização e excesso de fontes de informação
	Requisitantes em excesso
	Compromissos não realizáveis feitos pelos vendedores
	Considerações desnecessárias sobre design

Tabela 1 - Dificuldades na Eliciação de Requisitos

Fonte: Reproduzida de Tsumaki & Tamai (2006, p. 51-52)

Em harmonia com estudos já citados, Davis et. al. (2006) apresentam que a comunicação é o principal foco de problemas na fase de eliciação de requisitos. A Figura 3 apresenta a tipologia dos desafios na comunicação, explicados na sequencia.

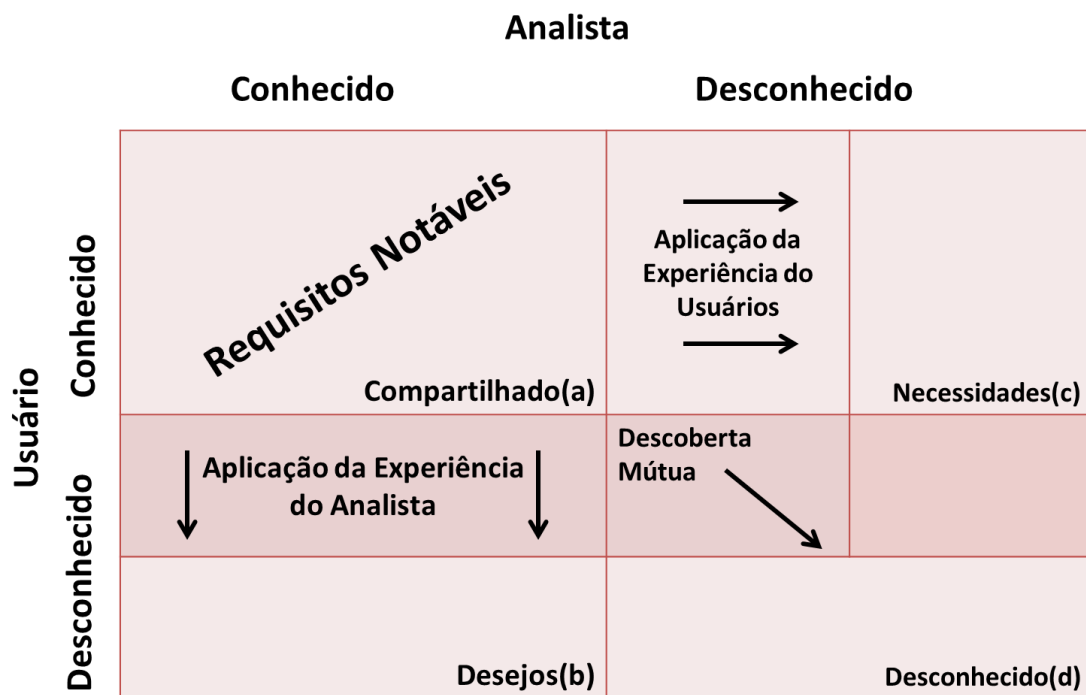


Figura 3: Tipologia dos Desafios de Comunicação na Eliciação de Requisitos

Fonte: Reproduzida de Davis et. al. (2006, p. 79)

No primeiro quadrante (a) se encontram os requisitos conhecidos pelo usuário e pelo analista baseado em sua experiência com o domínio da aplicação. O problema se encontra nos outros quadrantes. Em (b), residem os requisitos que o usuário não conhece, nos quais a aplicação do conhecimento no domínio e a experiência do analista são fundamentais. No quadrante (c) estão as necessidades que o usuário conhece, mas o analista as desconhece, sendo necessárias técnicas específicas para trazê-las ao conhecimento comum. Em (d) encontram-se os requisitos potenciais, desconhecidos pelo usuário e pelo analista, podendo ser descobertos somente quando o sistema estiver em uso.

2.5 Técnicas e fatores de sucesso

Davis et. al. (2006) expõe que o sucesso da fase de eliciação de requisitos ocorre quando os requisitos atendem às necessidades dos usuários. Apesar dos problemas apontados nos tópicos anteriores, pesquisas e estudos na área colaboraram para o desenvolvimento de técnicas, ferramentas e processos. Maiden (2005) pontua que pesquisas na área de requisitos são publicadas desde 1970. Maiden e Rugg (1996) elaboraram o *framework* ACRE (*ACquisition of REquirements*) unindo várias técnicas de eliciação indicando sua aplicação conforme o objetivo do processo. Encontram-se na literatura análises sobre a utilização das mais variadas técnicas (SAJID ET. AL. , 2010, REHMAN ET. AL. , 2013) sobre pontos positivos e negativos.

Hickey e Davis (2003) expõem que a habilidade do analista na seleção de técnicas de eliciação de requisitos pode aumentar o sucesso nos projetos, devendo ser utilizado os seguintes critérios (Ibid. p. 177):

- Para cada técnica de eliciação existe um conjunto de fatores circunstanciais que levam o analista a escolher aquela técnica.
- Para cada técnica de eliciação existe um conjunto adicional de predicados que caso sejam verdade levam o especialista alterarem sua escolha inicial.
- Para cada técnica de eliciação existe um conjunto de habilidades do analista que devem estar presentes para que o uso da técnica seja efetivo.
- Para cada técnica de eliciação existe um conjunto adicional de habilidades que não são universalmente uma necessidade, mas que podem ser necessárias durante a execução da técnica sem pré-conhecimento disso.

Rehman et. al. (2013) endossam a necessidade de se adotar critérios na escolha das técnicas de eliciação conforme o domínio da aplicação e o tipo de sistema que se está desenvolvido.

Rupp (2002) também aponta fatores humanos também aplicáveis à seleção de qual técnica utilizar:

- Relacionamentos de poder e confiança dentro do time
- Se a informação necessária se encontra no consciente, inconsciente ou subconsciente.
- Se o *stakeholder* possui bom desempenho para explicar seu conhecimento
- Se o sistema irá lidar com procedimentos dinâmicos ou com informação estática.

Em pesquisa feita com quinze times de engenharia de requisitos para diversos projetos, Hofmann e Lehner (2001) elencaram as melhores práticas utilizadas para atingir o sucesso, expostas na Tabela 2.

Prática	Benefícios
Envolver clientes e usuários no processo de engenharia de requisitos	Melhor entendimento das “reais” necessidades
Identificar e consultar todas as fontes possíveis de requisitos	Maior cobertura dos requisitos
Atribuir gerentes de projeto e membros de equipe experientes para as atividades	Desempenho mais previsível
Direcionar de 15% a 30% do total do esforço do projeto para a engenharia de requisitos	Manter uma especificação de alta-qualidade durante todo o projeto
Utilizar <i>templates</i> e exemplos de especificação	Aumentar a qualidade da especificação
Manter um bom relacionamento com os <i>stakeholders</i>	Satisfazer de uma melhor forma as necessidades do cliente
Priorizar requisitos	Focar a atenção nas necessidades mais importantes
Desenvolver modelos complementares junto de protótipos	Eliminar ambiguidades e inconsistências da especificação
Manter uma matriz de rastreabilidade	Obter um <i>link</i> explícito entre os requisitos e

	os entregáveis do projeto
Utilizar revisão por pares, cenários e orientações para validar e verificar os requisitos	Obter uma especificação mais precisa e maior satisfação do cliente

Tabela 2 - Melhores Práticas e Benefícios na Engenharia de Requisitos

Fonte: Sumarizado com base de Hofmann e Lehner (2001, p.65).

Batool et. al. (2003) comparam a utilização de técnicas de engenharia de requisitos tradicionais com técnicas ágeis e concluem que as técnicas ágeis são melhores para lidar com cenários de requisitos flexíveis.

Rehman et. al. (2013, p.46) aponta que “devido aos méritos relativos e limitações dos métodos existentes, é sempre sugerido que seja usado um conjunto de diferentes técnicas para obter a perfeição na fase de eliciação de requisitos. De fato, muitas das técnicas de eliciação necessitam intrinsecamente ser usadas em conjunto de outras técnicas devido a seus atributos complementares”.

Baseado no entendimento dos problemas e soluções propostas existentes, o capítulo seguinte buscará explicar como a utilização de uma abordagem formal pode auxiliar no entendimento dos requisitos.

CAPÍTULO 3 – ELICIAÇÃO DE REQUISITOS UTILIZANDO LINGUAGEM ORIENTADA A METAS

3.1 Formalização de Requisitos

No capítulo anterior pode-se observar que muitos dos problemas levantados possuem como origem problemas de comunicação entre os envolvidos. A necessidade de se obter entendimento comum é um dos principais motivos de existir uma especificação de requisitos (PARNAS e CLEMENTS, 1986).

Todavia, a utilização de linguagem natural para descrição dos requisitos não é um bom método de armazenar o entendimento do analista (JACKSON, 1996). Lamsweerde (2001a) sugere que a utilização de modelos formais seria a melhor alternativa na engenharia de requisitos para orientar a eliciação, especificação e análise dos requisitos. Heisel e Souquières (1999) expõem que a utilidade de uma especificação formal é cada vez mais aceita por pesquisadores e engenheiros de software.

Contudo, construir um modelo que possa ser compreendido pelo cliente, expondo sua visão do problema e ao mesmo tempo sendo útil aos analistas permitindo a extração dos requisitos e o entendimento do domínio do problema é algo complexo (GIBSON, 2000).

Métodos semiformais são utilizados para representação e comunicação de ideias, como o diagrama entidade-relacionamento, diagramas de estado e a linguagem de modelagem UML são utilizados para a modelagem de requisitos (LAMSWEERDE, 2001a). Suas vantagens e desvantagens são apresentadas na Tabela 3:

Vantagens	Desvantagens
Notações gráficas são fáceis de entender e comunicar	Em geral os métodos semiformais conseguem lidar apenas com aspectos funcionais.
Diferentes tipos de diagrama proveem visões complementares e inter-relacionadas do mesmo sistema.	Devido à captura somente de características declarativas do sistema, geralmente suportam formas muito limitadas de especificação e análise.
Notação padronizada	A semântica da notação gráfica se torna confusa, sendo que o mesmo modelo pode

	ser interpretado de formas diferentes por pessoas diferentes.
--	---

Tabela 3 - Vantagens e Desvantagens dos Métodos Semiformais

Fonte: Elaborado com base em Lamsweerde (2001a, p.5-7).

Devido à utilização de notação matemática, Lawrence (1996) expõe que os métodos formais são de difícil comunicação para o cliente. De acordo com Lamsweerde (2001a), a escrita e leitura de especificações formais também são complexas, exigindo um conhecimento especializado também para detalhar as entradas necessárias para as ferramentas de análise. Gibson (2000) esclarece que técnicas de desenvolvimento de software mais rigorosas são necessárias, embora utilizem mais tempo e cujo domínio é mais difícil.

Todavia, as vantagens da utilização de métodos formais para a especificação de requisitos também é apresentada na literatura (FURIA et. al. 2006, BALZER e GOLDMAN, 1981). Lamsweerde (2001a) aponta as seguintes vantagens na utilização de métodos formais:

- Fornecem regras precisas de interpretação de declarações;
- Oferecem suporte sofisticado para atividades de análise, como animação e verificação algorítmica;
- Permitem a geração automática de outros artefatos úteis como contraexemplos, cenários de falha, casos de teste, obrigações de prova, refinamentos, fragmentos de código e outros.

3.2 Engenharia de Requisitos Orientada a Metas

Uma dos principais métodos formais utilizados para a engenharia de requisitos são as abordagens orientadas a metas (GORE – *Goal-Oriented Requirements Engineering*) que fazem uso intenso de conhecimento do domínio para estabelecer metas que irão resultar nos requisitos do sistema (LAMSWEERDE, 2001b).

Kavakli (1999) expõe que a relação entre a engenharia de requisitos e o contexto organizacional tem atraído interesse de pesquisa, baseado na premissa de que no desenho de sistemas de software, os engenheiros de requisitos devem buscar aperfeiçoar situações organizacionais vistas como problemáticas, ou que estão em estado menor que a perfeição.

Em complemento, Almisned e Keppens (2010) expõem que o alinhamento proposto entre a análise de requisitos e as metas de negócio é essencial.

Segundo LAMSWEERDE (2001b, p. 250), “uma meta é um objetivo que o sistema sobre consideração deve atingir”. A necessidade de se conhecer as metas na engenharia de requisitos foi reconhecida por Ross e Schoman (1977) como essencial, onde a definição dos requisitos deve dizer o porquê um sistema é necessário. Fowler (2003, p.79), em seu livro sobre UML, já demonstrava a importância de se conhecer as metas: “Esta meta do usuário é a chave para os casos de uso. Um caso de uso é um conjunto de cenários unidos por uma meta de usuário comum”.

Peng (2009) descreve que as abordagens orientadas por metas questionam o porquê as funcionalidades são necessárias e como implementá-las, diferente dos métodos tradicionais focados nas funcionalidades e suas interações com os usuários.

Na GORE, um agente seria um componente ativo de sistema, como um ser humano, um dispositivo ou um software. Um requisito de software seria uma meta cujo atingimento está sobre responsabilidade de um único agente de software (LAMSWEERDE e LETIER, 2004).

A derivação dos requisitos a partir das metas segue um conceito lógico: um requisito só pode existir porque existe uma determinada meta sendo base para o mesmo, ou seja, os requisitos implementam as metas tal qual um programa implementa uma especificação de design (LAMSWEERDE, 2001b).

Lapouchnian (2005) apresenta as principais abordagens de GORE:

O *Framework* NFR: esta abordagem está focada na modelagem e análise de requisitos não funcionais e busca colocá-los

i/Tropos*: O *i** é um *framework* que pode ser usado não somente para engenharia de requisitos, mas também em outras áreas como reengenharia de processos de negócio e análise de impacto organizacional. Nas etapas iniciais da fase de requisitos, esta abordagem facilita a análise do domínio ao representar em diagramas os *stakeholders*, seus objetivos e relacionamentos. A Tropos é uma metodologia de desenvolvimento que utiliza o *i** como base.

GBRAM: Sigla para *Goal-Based Requirements Analysis Method*, este método se concentra em identificar e abstrair as metas de várias fontes de informação, assumindo que não há metas documentadas ou levantadas com os *stakeholders* e que podem ser usados para tal diagramas, documentos, entrevistas transcritas etc. já existentes.

KAOS: É uma metodologia composta por um conjunto de técnicas de análises formais. A sigla significa *Knowledge Acquisitos in autOated Specification* ou *Keep All Objects Satisfied*. Ela permite a combinação de diferentes níveis de expressão e raciocínio, como métodos semiformais para a modelagem e estruturação de metas e abordagens formais para um raciocínio mais preciso.

3.3 GORE aplicada à Eliciação de Requisitos

Deve-se observar que a utilização da orientação a metas é complementar à análise orientada a objetos, na qual a modelagem das metas foca principalmente nos estágios iniciais da análise de requisitos (MYLOPOULOS, CHUNG, & YU, 1999). Como exposto por Sommerville, Sawyer e Viller (1998), a utilização de uma abordagem que combine múltiplas perspectivas na engenharia de requisitos pode, potencialmente, levar a especificações com maior possibilidade de satisfazer as necessidades de um conjunto diverso de *stakeholders*.

Ao exemplificar o uso da metodologia KAOS, Lamsweerde e Letier (2004) expõem que a modelagem de metas difere da modelagem utilizando casos de uso por se iniciar em declarações de alto nível, genéricas e precisas das intenções dos *stakeholders* em vez de declarações muito específicas, operacionais e frequentemente imprecisas. Desta forma, os casos de uso surgem como um passo posterior, operacionalizando as metas atribuídas a agentes de software.

A literatura apresenta que a GORE pode contribuir na resolução de problemas citados no capítulo 2 na etapa de eliciação de requisitos, como no conhecimento incompleto das necessidades e do propósito do sistema, além de falhas de comunicação entre os *stakeholders* e requisitos em excesso.

Lamsweerde (2001b) lista alguns fatores para mostrar a importância das metas na engenharia de requisitos:

- As metas fornecem critérios precisos para se determinar a pertinência dos requisitos, evitando requisitos irrelevantes;
- As metas fornecem uma forma de comunicação para se explicar os requisitos. Desta forma, ao se refinar uma meta é possível obter rastreabilidade de um objetivo estratégico de alto nível até requisitos técnicos de um nível mais baixo. As metas podem ser utilizadas para relacionar o software futuro ao contexto organizacional e de negócio.
- A análise de alternativas através de metas fornecem o nível adequado de abstração para que tomadores de decisão consigam ser envolvidos para validar opções ou sugerir alternativas.
- As metas foram reconhecidas como por fornecer a base para identificar conflitos entre requisitos e por resolvê-los, em alguns casos.

Todavia, Lapouchnian (2005) apresenta que as metas não necessariamente estão explícitas e também precisam ser eliciadas. Neste processo, a análise do sistema/organização atual se mostra uma fonte importante na identificação de metas.

Regev e Wegmann (2002, p.2) apontam que quatro fatores podem afetar a identificação das metas:

1. O que é necessário nem sempre é desejado;
2. O que é desejado nem sempre é necessário;
3. O que é necessário ou desejado frequentemente não pode ser implementado;
4. O que é inicialmente sentido como necessário geralmente não é o que seria mais efetivo.

Lamsweerde (2001b) esclarece a necessidade de se refinar as metas em busca de submetas implícitas e que este refinamento deve prosseguir até que seja possível atribuir uma submeta a um agente de software ou do ambiente. A importância de se refinar as metas também é discutida em Estrada et. al. (2004) e Antón (1997). Em complemento, Lapouchnian (2005) expressa que a partir de algumas metas identificadas, o alvo é realizar o refinamento até que as metas sejam facilmente operacionalizadas e implementadas, sendo que este processo geralmente é feito perguntando-se questões do tipo “COMO” realizar a meta. Lamsweerde (2001b) apresenta que perguntas do tipo “PORQUE” também devem ser feitas nesta etapa.

É apresentando na literatura a eliciação de requisitos utilizando GORE e métodos ágeis (SEN e HEMACHADRAN, 2010), para geração de ideias na etapa de eliciação (OSHIRO, WATAHIKI e SAEKI, 2003) e na identificação de metas de um sistema crítico para controle da velocidade e aceleração de trens (LAMSWEERDE, 2001b).

Sommerville, Sawyer e Viller (1998) apontam que as abordagens orientadas a metas apresentam como vantagem a exposição de diferentes metas por diferentes *stakeholders*, fornecendo uma maneira estrutura de avaliar alternativas. Zowghi e Coulin (2005) expõem que na prática as abordagens GORE tem se mostrado uteis em situações onde somente se conhece as necessidades de alto nível do sistema e onde existe uma falta de entendimento de detalhes específicos dos problemas a serem resolvidos e suas possíveis soluções.

Contudo, ao analisar diversas abordagens GORE, Kavakli (2002) aponta que entre as existe um alto grau de sobreposição entre os diferentes métodos, na forma de que diferentes métodos fornecem estratégias alternativas que atuam sobre a mesma área da engenharia de requisitos. A busca por uma linguagem unificada, tal qual a UML, é apresentada por Patrício et. al. (2011), com a UGL (*Unified Goal-Oriented Language*), o que demonstra ser este um campo de pesquisa em evolução.

Após a explanação desta pesquisa, segue-se a conclusão desta monografia.

CONCLUSÃO

Contribuições desta pesquisa

As contribuições desta pesquisa podem assumir sentidos diferentes, dependendo do público e da área de pesquisa. Acredito que os estudantes de graduação e pós-graduação podem se beneficiar do texto e do referencial teórico como base para pesquisas subsequentes, amparado pelo conteúdo discutido e pela gama de tópicos estudados que se relacionem com o conteúdo abordado.

Para a Engenharia de Requisitos, esta pesquisa explorou uma visão do assunto, referente aos desafios existentes, com foco na eliciação de requisitos, e como abordagens orientadas a metas podem contribuir sobre os problemas apresentados na literatura. Nota-se, contudo que o problema de ser eliciar requisitos está longe de ser completamente conhecido e com soluções perfeitas. As abordagens orientadas a metas ainda se apresentam fragmentadas em diversos métodos e ferramentas. Para sua utilização e aprendizado em larga escala, esforços em pesquisa para a unificação das abordagens deve ser uma meta, o mesmo caminho seguido pela UML.

Ainda há muito a se desenvolver e espera-se que esta pesquisa possa ser mais um passo na direção do entendimento da questão. Interessante notar a ausência de literatura nacional e publicações de autores nacionais sobre este tópico, apesar de se mostrar um assunto de interesse científico há décadas.

Limitações do estudo e sugestões para novas pesquisas

As limitações deste estudo são muitas. A de maior consideração deve estar no fato de ser uma pesquisa de caráter teórico, cuja verificação prática não pode ser realizada por razões de tempo e recursos disponíveis.

Como sugestões para futuras pesquisas podem ser apontados várias recomendações em diversas áreas de pesquisa, sem esgotar, entretanto, o campo de pesquisa:

1. A aplicação prática de abordagens de engenharia de requisitos orientadas a metas na elicitação de requisitos de software, entre outras etapas do ciclo de desenvolvimento, apresentando estudos de casos reais.
2. A execução de pesquisas direcionadas a mostrar vantagens e desvantagens e fatores de sucesso de abordagens GORE.
3. Estudos orientados à construção de uma linguagem de modelagem de metas que unifique as diversas metodologias existentes. Nota-se que a complexidade vai além da etapa de modelagem, pois, como apontado por Lamsweerde (2001b, p. 252), “metas devem obviamente ser especificadas precisamente para suportar a elaboração, verificação/validação, gerenciamento de conflitos, negociação, explicação e evolução dos requisitos”.

Considerações finais

Considerado o objetivo estabelecido, este estudo analisou uma pequena parte da área de pesquisa em torno da engenharia de requisitos, de natureza complexa e interdisciplinar, e propicia um referencial para futuras pesquisas, de caráter mais extenso e exaustivo.

REFERÊNCIAS BIBLIOGRÁFICAS

ALMISNED Faisal, KEPPENS Jeroen. Requirements Analysis: Evaluating KAOS Models In: Journal of Software Engineering and Applications, v. 3 n. 9, p. 869-874, Setembro 2010.

ANTÓN Annie I. Goal Identification and Refinement in the Specification of Information Systems, 1997. 261 p. Tese (Doutorado em Ciência da Computação) – Instituto de Tecnologia da Geórgia, Universidade do Estado da Carolina do Norte, 1997.

ARIF.S., KHAN. Q. GAHYUR. S.A.K. Requirement Engineering Processes, Tools/Technologies, & Methodologies In: International Journal of Reviews in Computing (IJRIC), v.2, p. 41-56, Mar 2010.

BALZER, R. M, GOLDMAN, N.M, WILE, D.S. Operational Specification as the Basis for Rapid Prototyping In: ACM SIGSOFT Software Engineering Notes, v.7, n.5, p. 3-16, Dezembro 1982.

BARRY, Daniel M. The Inevitable Pain of Software Development: Why There Is No Silver Bullet In: WIRSING, Martin, KNAPP, Alexander, BALSAMO, Simonetta (Org.). Radical Innovations of Software and Systems Engineering in the Future. Venice: Springer, 2004, p. 50-74.

BATOOL, Asma, MOTLA Yasir Hafeez, HAMID Bushra, ASGHAR Sohail, RIAZ Muhammad, MUKHTAR Mehwish, AHMED Mehmood. Comparative Study of Traditional Requirement Engineering and Agile Requirement Engineering In: International Conference on Advanced Communication Technology, 15th, 2013, PyeongChang, Anais... Los Alamitos, IEEE Computer Society, 2013, p. 1006-1014

BELL, T. E. THAYER T.A. Software Requirements: Are They Really a Problem? In: International Conference on Software Engineering, 2nd, 1976, São Francisco, Anais...Los Alamitos, IEEE Computer Society, 1976, p. 61-68.

BJARNASON Elizabeth, WNUK Krzysztof e REGNELL Björn. Requirements Are Slipping Through the Gaps - A Case Study on Causes & Effects of Communication Gaps in Large-

Scale Software Development. In: IEEE International Requirements Engineering Conference, 19th, 2011, Trento, Anais... Los Alamitos, IEEE Computer Society, 2011, p. 37 - 46.

BJØRNER, Denis. A Cloverleaf of Software Engineering In: IEEE International Conference on Software Engineering and Formal Methods, 3rd, 2005, Koblenz, Anais...Nova Iorque, IEEE Press, 2005, p. 75-85.

BOEHM, Barry, A View of 20th and 21st Century Software Engineering In: International Conference on Software Engineering, 28th, 2006, Xangai, Anais...Nova Iorque, ACM New York, 2006, p. 12-29.

BROOKS, Frederick P. No Silver Bullet: Essence and Accident in Software Engineering In: IEEE Computer, v. 20, n. 4, p. 10-19, Abril 1987.

CHRISTEL, Michael G. KANG, Kyo C. Issues in Requirements Elicitation. Pittsburgh, Pensilvânia: SEI – Software Engineering Institute, 1982, 80p (Relatório Técnico: CMU/SEI-92-TR-012, ESC-TR-92-012).

DAVIS, Christopher J, TREMBLAY, Monica Chiarini, FULLER Robert M. BERNDT, Donald J. Communication challenges in requirements elicitation and the use of the repertory grid technique. In: Journal of Computer Information Systems, v.47, número especial, p. 78-86, Setembro 2013.

DEMARCO, Tom. Software Engineering: An Idea that has Come and Gone In: IEEE Software, v.26, n. 4, p.95-96, Jul-Aug 2009.

EBERT, Christof. Requirements Before the Requirements: Understanding the Upstream Impacts In: IEEE International Conference on Requirements Engineering, 13th, 2005, Paris, Anais... Los Alamitos, IEEE Computer Society, 2005, p. 117-124.

ERNST Neil A., BORGIDA Alexander, MYLOPOULOS John. Requirements Evolution Drives Software Evolution In: International Workshop on Principles of Software Evolution and the annual ERCIM Workshop on Software Evolution, 12th e 7th, 2011, Szeged, Anais...Nova Iorque, ACM, 2011, p.16-20.

ESTRADA Hugo, PASTOR Oscar, MARTÍNEZ Alicia, JIMENEZ Jose Torres. Using a Goal-Refinement Tree to Obtain and Refine Organizational Requirements In: LAGANÁ Antonio , GAVRILOVA Marina L. , KUMAR Vipin , MUN Youngsong , TAN C. J. Kenneth, GERVASI Osvaldo (Org.). Computational Science and Its Applications – ICCSA 2004 : Part IV, Assisi, Springer, 2004, p. 506-513.

EVELEENS, J. Laurenz, VERHOEF, Chris, The Rise and Fall of Chaos Report Figures In: IEEE Software, v.27, n. 1, p.30-36, Jan-Feb 2010.

FILMAN, Robert E. Postmodern Software Development. In: IEEE Internet Computing, v. 9, n. 1, p. 4-6, Jan-Feb 2005.

FOWLER, Martin. UML Distilled: A Brief Guide to the Standard Object Modeling Language. 3^a ed. Boston: Addison-Wesley Professional, 25 de Setembro 2003, 208 p.

FURIA Carlo A., ROSSI Matteo, STRUNK Elisabeth A., MANDRIOLI Dino, KNIGHT John C. Raising Formal Methods To The Requirements Level. Departamento de Eletrônica e Informação, Politécnica de Milão e Departamento de Computação, Universidade da Virgínia. Novembro, 2006, 43p (Relatório Técnico: CS-2006-24).

GIBSON, Paul J. Formal Requirements Engineering: Learning from the students In: Australian Software Engineering Conference, 2000, Canberra, Anais... Los Alamitos, IEEE Computer Society, 2000, p.171.180.

GLASS, Robert L. The Standish Report: Does It Really Describe a Software Crisis? In: Communications of the ACM – Music Information Retrieval, v. 49, n. 8, p. 15-16, Aug. 2006.

GLASS, Robert, L. Silver Bullet Milestones in Software History In: Communications of the ACM – Spyware, v. 48, n. 8, p. 15-18, Aug. 2005.

GLASS, Robert, L, VESSEY, Iris, Software Tasks: Intellectual, Clerical...or Creative? In: Hawaii International Conference on System Sciences, 27th, 1994, Maui, Anais...Nova Iorque, IEEE Press, 1994, p.377-382.

HALL, T, BEECHAM, S. RAINER, A. Requirements problems in twelve software companies: an empirical analysis In: IEE Anais... Software, v.149, n.5, p. 153-160, Oct 2002.

HEISEL, Maritta, SOUQUIÈRES Jeanine. Methodological Support for Requirements Elicitation and Formal Specification In: International workshop on Software specification and design, 9th, 1998, Mie, Anais... Los Alamitos, IEEE Computer Society, 1998, p. 153-155.

HICKEY, Ann M. DAVIS, Alan M. Elicitation Technique Selection: How Do Experts Do It? In: IEEE International Requirements Engineering Conference, 11th, 2003, Monterrey, Anais... Los Alamitos, IEEE Computer Society, 2003, p. 169 – 178.

HOFMANN, Hubert F., LEHNER, Franz. Requirements Engineering as a Success Factor in Software Projects In: IEEE Software, v.18, n. 4, p. 58 - 66, Jul-Aug 2001.

HOOKS, Ivy. Writing Good Requirements. In: International Symposium of the NCOSE - Volume 2, 3rd, 1993, Virginia, Anais... San Diego, INCOSE, 1993.

IAG CONSULTING – Mythbusting the Path to Requirements Success – Why Some Companies Will Always Waste Millions on Poor Requirements Quality. New Castle: IAG Consulting Inc. , 2009.

IEEE, The Institute of Electrical and Electronics Engineers. Guide to the Software Engineering Body of Knowledge. SWEBOK® Guide, 2004, 202p.

IEEE, The Institute of Electrical and Electronics Engineers. IEEE Std 610.12. IEEE Standard Glossary of Software Engineering Terminology, 1990, 84p.

ISO/IEC/IEEE - International Organization for Standardization / International Electrotechnical Commission / The Institute of Electrical and Electronics Engineers - ISO/IEC/IEEE 29148 – Systems and Software Engineering – Life cycle processes – Requirements Engineering, 2011, 95p.

JACKSON, Daniel. Requirements Need Form, Maybe Formality. In: IEEE Software, v. 13, n. 2, p. 21-22, Março 1996.

KAVAKLI, Evangelia. Goal-Oriented Requirements Engineering: A Unifying Framework In: Requirements Engineering, v. 6, n.4, p. 237-251, Janeiro 2002.

KAVAKLI, Evangelia. Goal-Driven Requirements Engineering: Modelling and Guidance. 1999. 234 p. Tese (Doutorado em Filosofia) – Departamento de Computação, Instituto de Ciência e Tecnologia, Universidade de Manchester, 1999.

KUMARI, Neetu, PILLAI, Anitha S. A Survey On Global Requirements Elicitation Issues And Proposed Research Framework In: IEEE International Conference on Software Engineering and Service Science, 4th, 2013, Pequim, Anais... Los Alamitos, IEEE Computer Society, 2013, p. 554 – 557.

LAPOUCHNIAN, Alexei. Goal-Oriented Requirements Engineering: An Overview of the Current Research. Toronto: Departamento de Computação, Universidade de Toronto, 2005, 32p (Relatório Técnico).

LAMSWEERDE, Axel Van. Requirements Engineering: From Craft to Discipline. In: International Symposium on the Foundations of Software Engineering, 16th, 2008, Atlanta, Anais...Nova Iorque, ACM, Novembro 2008, p. 238-249.

LAMSWEERDE, Axel Van, LETIER Emmanuel. From Object Orientation to Goal Orientation: A Paradigm Shift for Requirements Engineering In: WIRSING, Martin, KNAPP, Alexander, BALSAMO, Simonetta (Org.). Radical Innovations of Software and Systems Engineering in the Future. Venice: Springer, 2004, p. 325-340.

LAMSWEERDE, Axel Van. Building Formal Requirements Models for Reliable Software In: CRAEYNEST Dirk, STROHMEIER Alfred (Org.). Reliable Software Technologies Ada-Europe 2001. Leuven, Springer, 2001a, p. 1-20.

LAMSWEERDE, Axel Van. Goal-Oriented Requirements Engineering: A Guided Tour In: IEEE International Symposium on Requirements Engineering, 5th, 2001, Toronto, Anais... Los Alamitos, IEEE Computer Society, 2001b, p. 249 – 262.

LAMSWEERDE, Axel Van. Requirements Engineering in the Year 00: A Research Perspective In: International Conference on Software Engineering, 22nd, 2000, Limerick, Anais...Nova Iorque, ACM, 2000, p. 5-19.

LAWRENCE, Brian. Do We Really Need Formal Requirements? In: IEEE Software, v. 13, n. 2, p. 20-22, Março 1996.

LEFFINGWELL, Dean, WIDRIG, Don, Managing Software Requirements: A Use Case Approach, 2 ed. Westford: Addison-Wesley Professional, 2003.

LIENTZ, B. P. SWANSON, E.B. TOMPKINS, G. E. Characteristics of Application Software Maintenance In: Communications of the ACM, v. 21, n. 6, p. 466-481, Jun. 1978.

MAIDEN, Neil. What Has Requirements Research Ever Done for Us? In: IEEE Software, v. 22, n.4, p. 104-105, Julho/Agosto 2005.

MAIDEN, N. A. M. , RUGG, G. ACRE: Selecting Methods for Requirements Acquisition In: Software Engineering Journal, v.11, n. 3, p. 183-192, Maio 1996.

McCONNEL Steve, Rapid Development: Taming Wild Software Schedules. 1ª ed. Redmond, Washington: Microsoft Press, 09 de Julho 1996, 674 p.

MORGAN, Jeanette Nasem, Why the software industry needs a good Ghostbuster In: Communications of the ACM, v.48, n.8, p.129-133, August 2005.

MORLEY, JoAnn, MILLER Robert L. Tying Sociology to Information Systems Development In: IEEE National Aerospace and Electronics Conference, 1994, Dayton, vol.2 Anais... Los Alamitos, IEEE, 1994, p. 845 – 852.

MYLOPOULOS John, CHUNG Lawrence, YU Eric. From object-oriented to goal-oriented requirements analysis In: Communications of ACM, v. 42 n. 1, p. 31-37, Janeiro 1999.

NAZ, Humaira, KHOKHAR, Muhammad Nadeem. Critical Requirements Engineering Issues & their Solution. International Conference on Computer Modeling and Simulation, 1st, 2009, Macau, Anais... Los Alamitos, IEEE Computer Society, 2009, p. 218 – 222.

NUSEIBEH, Bashar e EASTERBROOK Steve. Requirements Engineering: A Roadmap In: International Conference on Software Engineering, 22nd, 2000, Limerick, Anais...Nova Iorque, ACM, 2000, p.35-46.

OSHIRO, Kazuya, WATAHIKI Kenji, SAEKI Motoshi. Goal-Oriented Idea Generation Method for Requirements Elicitation In: IEEE International Requirements Engineering Conference, 11th, 2003, Monterrey, Anais... Los Alamitos, IEEE Computer Society, 2003, p. 363 - 364.

PANDEY, D, SUMAN, U, RAMANI. An Effective Requirement Engineering Process Model for Software Development and Requirements Management In: International Conference on Advances in Recent Technologies in Communication and Computing, 2nd, 2010, Kottayam, Anais...Los Alamitos, IEEE Computer Society, 2010, p.287-291

PARNAS, David L. CLEMENTS, Paul C. A rational design process: how and why to fake it In: IEEE Transactions on Software Engineering, v. SE-12, n. 2, p. 251-257, Fev. 1986.

PATRÍCIO Pedro, AMARAL Vasco, ARAÚJO João, MONTEIRO Rui. Towards a Unified Goal-Oriented Language In: IEEE Annual Computer Software and Applications Conference, 35th, 2011, Munique, Anais... Los Alamitos, IEEE Computer Society, 2011, p.596 – 601.

PENG, Qian. Issues in Specifying Requirements for Adaptive Software Systems. 2009. 35 p. Dissertação (Mestrado em Ciência da Computação) – Escola de Matemática e Engenharia de Sistemas, Faculdade de Matemática, Ciência e Tecnologia, Universidade de Växjö, 2009.

QUISPE Alcides, MARQUES Maira, SILVESTRE Luis, OCHOA Sergio F., ROBBES Romain. Requirements Engineering Practices in Very Small Software Enterprises: A Diagnostic Study In: International Conference of the Chilean Computer Science Society, 29th, 2010, Antofagasta, Anais... Los Alamitos, IEEE Computer Society, 2010, p.81-87.

REGEV, Gil, WEGMANN, Alain. UML for Early Requirements Elicitation: A Regulation based Approach. Lausanne: Escola Politécnica Federal de Lausanne – Escola de Ciências da Computação e Comunicação, 2002, 15p (Relatório Técnico: EPFL-IC - IC/2002/013).

REHMAN, T, KHAN, M. N. A, RIAZ N. Analysis of Requirement Engineering Processes, Tools/Techniques and Methodologies In: International Journal of Information Technology and Computer Science, v.5, n.3, p. 40-48, 2013.

ROSS, D. T. SCHOMAN K.E, Structured Analysis for Requirements Definition In: IEEE Transactions on Software Engineering, v.3, n.1, p. 6-15, 1977.

RUPP, Chris. Requirements and Psychology In: IEEE Software, v.19, n.3, p.16-18, Maio/Junho 2002.

SEN A M, HEMACHANDRAN K. Elicitation of Requirements in Goal Oriented Requirement Engineering In: Assam University Journal of Science and Technology, v. 5, n. 2, 2010.

SOMMERVILLE, Ian, SAWYER P., VILLER S. Viewpoints for requirements elicitation: a practical approach In: IEEE International conference on Requirements Engineering, 3rd, 1998, Colorado, Anais... Los Alamitos, IEEE Computer Society, p. 74-81, 1998.

THE STANDISH GROUP - The CHAOS Report. West Yarmouth, MA: Standish Group International Inc. 1994.

THEW Sarah, SUTCLIFFE Alistair. Investigating the Role of ‘Soft issues’ in the RE Process In: IEEE International Requirements Engineering Conference, 16th, 2008, Catalunha. Anais... Los Alamitos, IEEE Computer Society, p. 63-66, 2008.

TRAN, Xuan-Linh, KASSER, Joseph E. Towards improving the recognition and correction of poor requirements In: Systems Engineering/Test and Evaluation Conference, 2005, Brisbane, Anais... Brisbane, ICE Austrália, p. 1-13 2005.

TSUMAKI, Toshihiko, TAMAI, Tetsuo. A Framework for Matching Requirements Engineering Techniques to Project Characteristics and Situation Changes In: Software Process: Improvement and Practice, v. 11, n. 5, p. 505-519, Setembro/Outubro 2006.

VILLER, Stephen Alexandre. Human Factors in Requirements Engineering: a method for improving requirements processes for the development of dependable systems. 1999. 314 p. Tese (Doutorado em Ciência da Computação) – Departamento de Computação, Universidade de Lancaster, 1999.

WAHONO, Romi Satria. Analyzing Requirements Engineering Problems In: IECI Japan Workshop, 4th, 2003, Tóquio, Anais... Tóquio, IEEE Computer Society, 2003, p. 55-58.

WEINBERG, Gerald M. Quality Software Management: Anticipating Change. Nova Iorque, Dorset House, Maio 1997. Volume 4.

WIEGERS, Karl E. Writing Quality Requirements In: Software Development, v. 7, n. 5, p. 44-48, Maio 1999.

ZAVE, Pamela. Classification of Research Efforts in Requirements Engineering In: IEEE International Symposium on Requirements Engineering, 2nd, 1995, Nova Iorque, Anais... Los Alamitos, IEEE Computer Society, 1995, p. 214-216.

ZOWGHI Didar , COULIN Chad . Requirements Elicitation: A Survey of Techniques, Approaches, and Tools In: AURUM Aybüke , WOHLIN Claes (Org.). Engineering and Managing Software Requirements. Berlim, Springer, 2005, p. 19-46.