



Engenharia de Requisitos

Definição de requisitos

The IEEE Standard Glossary of Software Engineering Terminology [IEEE97] define requisito como uma condição ou capacidade necessária para um usuário resolver um problema ou alcançar um objetivo (...).

Desta definição pode-se asserir que os requisitos de software são derivados de necessidades que usuários têm em resolver algum problema. Situa-se, então, o domínio do problema. O domínio do problema é uma parte particular do sistema real na qual o usuário está interessado, a parte relevante para seu problema específico [JAC95].

Definição de universo de informações

Porém, há consenso que a especificação de requisitos deve incluir não somente as especificações do domínio do problema, mas também qualquer tipo de informação que descreva o contexto do sistema [ALE99]. Esse contexto é conhecido como universo de informações, que é a realidade circunstanciada pelo conjunto de objetivos definidos pelos que demandam o software, e inclui todas as fontes de informação e todos os *stakeholders* [LEI94].

Definição de domínio do problema

O domínio do problema é o domínio de atuação do software e inclui todos os elementos que interagem com ele. O universo de informações, entretanto, é mais abrangente. É o contexto do sistema real, no qual encontram-se elementos e ocorrem fenômenos que provêm subsídios ao software e onde sentem-se seus efeitos e benefícios.

Uma analogia com um projeto de uma ponte sobre um rio pode ser considerada para ilustrar os conceitos. O domínio do problema restringe-se às especificações da ponte em si, como o objetivo em tê-la construída, o local e traçado desejado, o desenho arquitetônico, as especificações estruturais e de materiais, as especificações das fundações e do projeto necessário para sua construção. Já o universo de informações compreende todas as informações que influenciam a especificação, como o leito do rio, as margens, o curso d'água, as condições climáticas e geológicas e todos os outros fatores que vão interferir no projeto, e, também, todo o contexto que sofrerá seus efeitos, como a população e mercado beneficiados.

O domínio do problema é, portanto, o ambiente de atuação do software. Neste ambiente principiam as atividades da engenharia de software, a definição das necessidades do software. É tarefa dos engenheiros de requisitos entender o problema, na cultura e linguagem dos usuários, e definir um sistema que atenda às suas necessidades [LEF2000]. Para tal, o engenheiro de requisitos deve descobrir e estabelecer o universo de informações, de onde obtém os recursos na tarefa de elucidação do problema.

Definição do domínio da solução

Do contexto do problema parte-se para a abordagem da solução. No domínio da solução é enfocada a definição da solução aos problemas dos usuários. Neste espaço, os

desenvolvedores do software aplicam seus conhecimentos em busca da especificação do sistema a ser desenvolvido. Envolve as características da solução e os requisitos do sistema (figura 1). As características são descrições simples, na linguagem dos usuários, com o objetivo de comunicação entre os desenvolvedores e usuários sobre como o sistema trata o problema. Um característica é definida como um serviço que o sistema provê para cumprir uma ou mais necessidades dos *stakeholders* [LEF2000].

Uma vez estabelecido o conjunto de características com a concordância dos *stakeholders*, deve-se definir os requisitos mais específicos que serão necessários para a solução. Requisitos são propriedades que um sistema de software deve ter em busca do seu êxito no ambiente onde será utilizado [GOG94]. São especificações de o que deve ser desenvolvido e são condicionantes do processo da engenharia do software [SOM97].

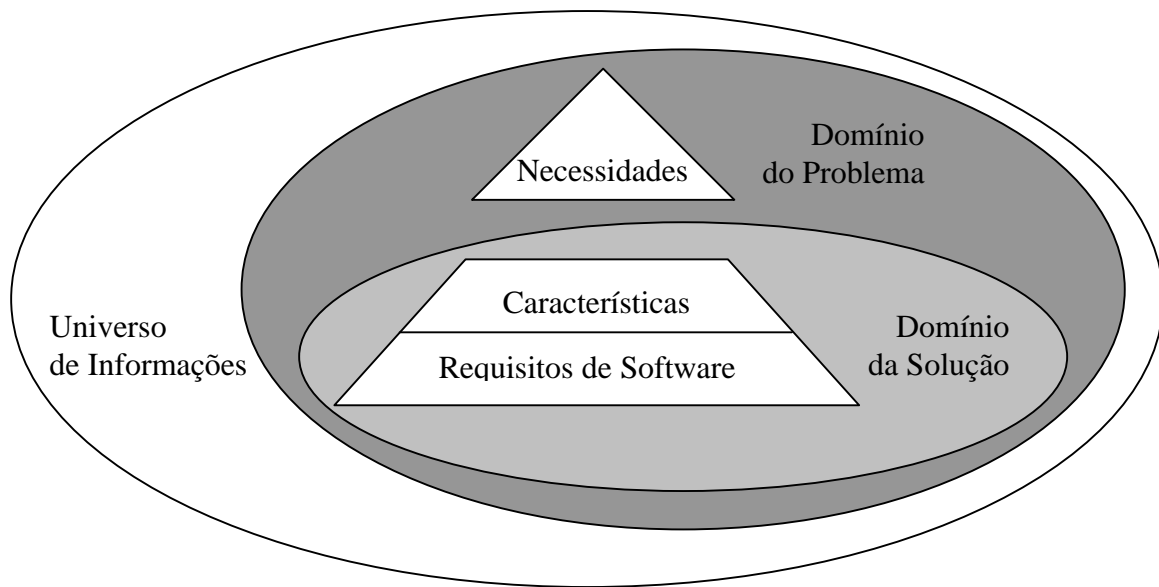


Figura 1. Os domínios da Engenharia de Requisitos

Níveis dos requisitos de software

Os requisitos de software incluem três níveis [WIE99]: os requisitos do negócio, os requisitos do usuário e os requisitos funcionais. Os requisitos do negócio identificam os benefícios primários que o sistema proverá aos clientes. São objetivos de alto nível ou requisições do cliente para o sistema de software.

Os requisitos do usuário são tarefas que os usuários serão habilitados a realizar com o sistema de software. Em geral, são capturados segundo abordagens de casos de uso e descrição de cenários, conforme será apresentado em capítulos subsequentes.

Os requisitos funcionais definem a funcionalidade que o software deve prover a fim de capacitar os usuários a realizar suas tarefas, satisfazendo os requisitos do negócio. Existem taxonomias que classificam em requisitos funcionais e não funcionais conforme seus objetivos [MAC99]. Basicamente, pode-se definir os requisitos funcionais como especificações de o que o software deve estar habilitado a fazer, e os requisitos não funcionais como especificações de o quão bem as funções devem ser realizadas (qualitativos), o custo empregado para criar e manter a funcionalidade e a qualidade (quantitativos) e as restrições impostas aos requisitos ou liberdade de desenvolvimento (restritivos) [GIL97]. Os requisitos de software, portanto, compreendem tanto as necessidades funcionais, que definem o comportamento e as propriedades

do sistema, como as necessidades não funcionais, que consistem em quesitos de qualidade e restrições operacionais ou do desenvolvimento do software.

Representação dos requisitos

The IEEE Standard Glossary of Software Engineering Terminology [IEEE97] define, ainda, requisito como uma representação documentada da condição ou capacidade (necessária para um usuário resolver um problema ou alcançar um objetivo).

Infere-se, segundo essa definição, a essencialidade de documentar as necessidades do usuário. A representação dos requisitos é um conceito chave para a prática da engenharia do software [WIE99], e visa descrever em modelos conceituais as condições ou capacidades do sistema. O objetivo principal da representação é propiciar uma melhor compreensão do domínio do problema e uma precisa definição do domínio da solução.

A representação dos requisitos é estabelecida em um nível conceitual de modelagem. Um modelo conceitual é uma estrutura cognitiva usada com o propósito de entendimento e comunicação de aspectos físicos e sociais do mundo que nos acerca [LOU95]. Os modelos produzidos, textuais ou em notação gráfica, descrevem os aspectos relevantes do domínio do problema e relata os componentes e comportamento esperado para a solução de software. Uma representação completa e correta é fundamental para que o desenvolvimento do software realmente atenda às necessidades dos clientes e usuários, considerando que a satisfação dos requisitos especificados é uma pré-condição básica para o sucesso de um software [CAL99].

A Engenharia de Requisitos

A engenharia de requisitos é a disciplina que procura sistematizar o processo de definição de requisitos [LEI94]. Aborda um ponto fundamental do desenvolvimento de software: a definição do que se produzir. Tem sido identificada como uma fase crucial por tratar de conhecimentos não apenas técnicos, mas também gerenciais, organizacionais, econômicos e sociais [CAS95], e estar intimamente associada a qualidade do software [LEE98].

Fases da engenharia de requisitos

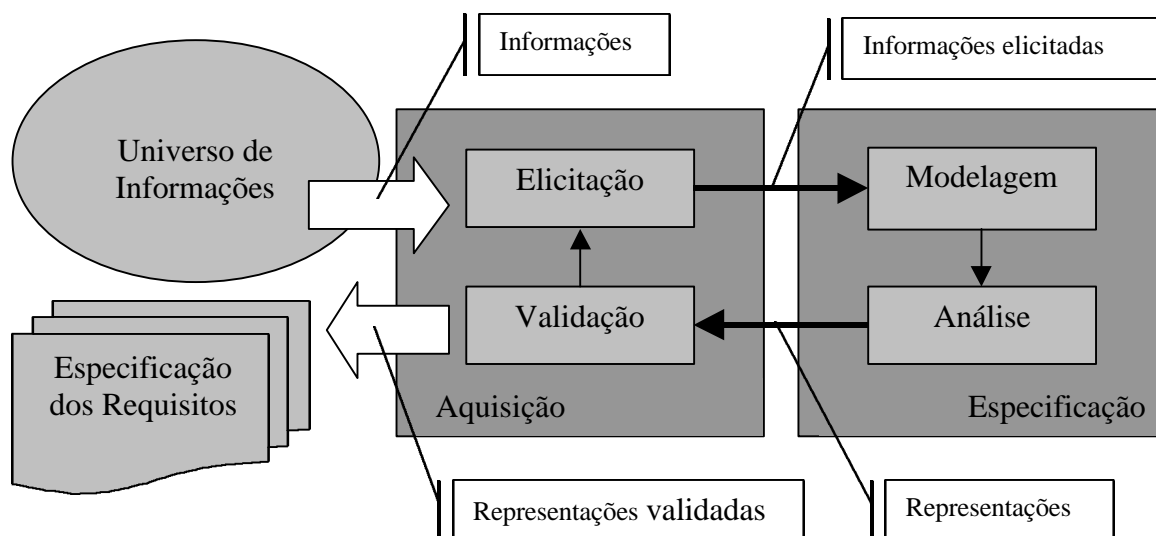


Figura 2. O processo da Engenharia de Requisitos

As atividades da engenharia de requisitos vão desde a idéia de desenvolver um sistema de software até a modelagem conceitual do que se vai construir. Aborda questões organizacionais, como identificação dos objetivos e definição do universo de informação; gerenciais, como determinação de diretrizes e análise de impacto do sistema; econômicas, como análise de custos, previsão e recursos necessários; sociais, como negociação, participação e comunicação entre os envolvidos; e técnicas, como coleta, organização, representação, análise e validação dos requisitos.

A engenharia de requisitos é responsável por propor um conjunto de métodos, técnicas e ferramentas que suportam o processo de definição de requisitos. Esse processo compreende duas fases com atividades distintas, porém interrelacionadas [CAS95]: uma fase de aquisição e uma fase de especificação dos requisitos (figura 2).

Fase de aquisição

A fase de aquisição tem como objetivo a obtenção do conhecimento do domínio do problema, junto às fontes do universo de informação, e a certificação entre os *stakeholders* de que as necessidades estão especificadas de forma correta e completa. Compreende as atividades de elicitación e validação.

A atividade de elicitación consiste na realização de ações para aquisição do conhecimento do domínio do problema. O objetivo principal é identificar e coletar informações do universo de informações, de forma a prover o mais completo e correto entendimento das necessidades que demandam o software.

A atividade de validação, no contexto da engenharia de requisitos, consiste na execução de ações com o objetivo de confirmar o conhecimento adquirido. Procura-se corroborar que o que foi especificado realmente represente o domínio do problema, isto é, retrate as necessidades. Sob esse ponto de vista, a validação pode ser considerada como uma atividade de negociação, no qual os clientes procuram explorar o completo entendimento dos requisitos, de acordo com o que eles querem e necessitam, enquanto que os desenvolvedores procuram a confiança em resolver o problema certo. Também busca-se assegurar que os requisitos funcionais sejam verificáveis, almejando garantir a correção da especificação, e que os requisitos não funcionais de qualidade, quantidade e restrições de software estejam contemplados na definição.

Fase de especificação

A fase de especificação tem como objetivo descrever conceitualmente o conhecimento elicitado em modelos que expressem o domínio do problema, conforme o interesse dos *stakeholders*, e avaliar a descrição visando a correção, completude e consistência das necessidades. Compreende as atividades de modelagem e análise.

A atividade de modelagem visa a representação dos requisitos em modelos conceituais que descrevem as necessidades. O objetivo principal é representar o domínio problema para propiciar uma melhor compreensão das propriedades desejadas para o software. Os modelos produzidos, textuais ou em notação gráfica, descrevem os requisitos, relatando os componentes e o comportamento do software. A facilidade de leitura e a não ambigüidade são características importantes a serem consideradas nos modelos.

A atividade de análise compreende ações de examinar os modelos buscando detectar e resolver inconsistências e omissões. A meta da análise é analisar se o domínio do problema está representado e entendido coerentemente. Essa atividade antecede a atividade de validação e, ao contrário desta última que conta com a participação dos clientes, é desempenhada pelos engenheiros de requisitos.

Uma propriedade importante para uma análise bem sucedida é o rastreamento, que consiste na ligação da necessidade do domínio do problema ao requisito representado nos modelos [LEI94]. Esse rastreamento deve perdurar durante todo o ciclo de desenvolvimento, facilitando o gerenciamento do processo de produção e a validação do software. Como requisitos são produzidos não apenas na fase de aquisição, mas também durante todas as fases subsequentes [PIN98], a introdução desta propriedade aos modelos dos requisitos agrega flexibilidade ao desenvolvimento do software e a manutenções futuras.

Métodos de Elicitação de Requisitos

A elicitación envolve todo um conjunto de ações, que ocorrem no universo de informações, visando capturar e registrar as informações que subsidiarão o entendimento do domínio do problema e a conseqüente especificação dos requisitos. A captura é um processo de descoberta no qual procura-se obter o máximo de informações para o conhecimento do objeto em questão. Requer uma habilidade em trabalhar com especialistas humanos e com o conhecimento tácito, que é trivial para quem conhece a informação, mas não é trivial para quem procura obtê-la, de forma que dificilmente é lembrado e, portanto, não é transmitido [GOG94].

Não é uma tarefa fácil obter esse conhecimento, pois os envolvidos têm experiências, conhecimentos, preconceitos e terminologias diferentes. Algumas técnicas das ciências sociais, como psicologia e sociologia, têm sido estudadas e utilizadas nesta atividade, que envolve fatores comportamentais e de relacionamento humano [BER98] [DEB99].

Entre as técnicas mais comuns para a aquisição do conhecimento estão: análise de documentos, entrevistas, reuniões e observações. Outras técnicas, como etnografia e *inquiry cycle model*, também são apresentadas como convenientes a sistemas mais críticos, porém requerem maior tempo para suas aplicações.

Análise de documentos

A análise de documentos é uma técnica usualmente aplicada na qual explora-se o conhecimento escrito encontrado no universo de informações. Na modelagem dos requisitos, segundo o modelo proposto, essa técnica é muito útil para a definição dos objetos que compõem o modelo. A análise dos documentos permite um contato com o vocabulário utilizado no domínio do problema e auxilia na construção do glossário de termos especializados, que tem por objetivo definir os objetos e equalizar o conhecimento dos *stakeholders*.

Técnicas de entrevistas

Entrevista é uma técnica de interação entre entrevistado (especialista do conhecimento) e entrevistador (engenheiro de requisitos) buscando revelar conceitos, objetos e a organização do domínio do problema, além de buscar soluções ou projeções de soluções que comporão o domínio da solução.

As entrevistas mais usuais são as tutoriais, informais e estruturadas [OLI96]. Nas entrevistas tutoriais o entrevistado fica no comando, praticamente lecionando sobre um determinado assunto. Nas entrevistas informais ou não estruturadas, o entrevistador age espontaneamente, perguntando ao entrevistado sem obedecer a nenhuma organização. Esse tipo de entrevista oferece flexibilidade ao entrevistador e, normalmente, é utilizado no início do processo de elicitación. Já as entrevistas estruturadas são preparadas pelo entrevistador, que define previamente o andamento do procedimento de aquisição de conhecimento. Um fator importante a ser considerado nas entrevistas é o registro das informações coletadas, que pode ser

realizado através de anotações ou gravações de áudio ou vídeo. O material produzido deve ser organizado e serve como base para a preparação da próxima entrevista.

Técnicas de reuniões

Reunião é uma técnica que prevê a participação coletiva dos envolvidos para discutir questões do domínio do problema. Esta prática permite uma interação mais natural entre os participantes e dispor de múltiplas visões sobre a questão abordada.

Participatory Design e *Joint Application Design* são metodologias de reuniões que enfatizam a participação coletiva na elicitação de requisitos. São práticas bem conhecidas de negociação de requisitos, que promovem a cooperação, entendimento e formação de equipes de trabalho entre os envolvidos no universo de informações.

Essas abordagens oferecem um ambiente apropriado para os desenvolvedores e usuários trabalharem em equipe, com o objetivo de compartilhar informações e idéias sobre os domínios do problema e da solução. Este processo auxilia a comunicação entre os envolvidos, que empenham-se em identificar necessidades, refinar requisitos, tomar decisões conjuntas, explorar possíveis soluções e selecionar alternativas apropriadas.

Joint Application Design (JAD) baseia-se em sessões estruturadas e disciplinadas, onde os envolvidos reúnem-se para desenvolver juntos o sistema de software. Em linhas gerais, essas sessões possuem uma agenda detalhada, recursos visuais para auxiliar a exposição de idéias, um moderador e um relator que registra as especificações de acordo comum entre os membros do grupo. O produto final é um documento que contém definições do software [DEB99].

Participatory Design (PD) é uma abordagem que concentra-se mais fortemente no envolvimento dos usuários, em relação ao *Joint Application Design*, por facilitar o processo de aprendizado entre desenvolvedores e usuários através de experiências conjuntas em situações de trabalho simuladas. Em linhas gerais, os usuários são introduzidos no ambiente dos desenvolvedores, conhecendo possibilidades técnicas e, da mesma maneira, os desenvolvedores colaboram com os usuários em suas tarefas. Ocorre um aprendizado mútuo que vem a contribuir no processo de definição dos requisitos [HER96].

JAD representa um movimento em direção a práticas mais colaborativas procurando viabilizar objetivos, enquanto que PD representa um movimento em direção a práticas mais técnicas procurando, também, viabilizar objetivos [CAR93]. Ambas metodologias são bem conhecidas por acentuar um alto grau de envolvimento dos usuários como imperioso para o desenvolvimento de um bom projeto de software. Como resultado, os desenvolvedores aumentam seus conhecimentos sobre o domínio da aplicação e os usuários tornam-se mais envolvidos no processo de desenvolvimento.

Técnica de observação

Observação é uma técnica na qual o engenheiro de requisitos procura ter uma posição passiva no domínio do problema, observando seus elementos e comportamentos. Esta estratégia visa obter um entendimento inicial sobre o contexto em estudo.

As observações consistem em observar alguém no momento da realização de suas tarefas rotineiras no ambiente real. O observador procura familiarizar-se com o domínio do problema e elicitar as informações necessárias para o seu entendimento. A aquisição desse conhecimento pode ocorrer com interrupção ou não das atividades do observador. Na observação simples o observador acompanha o raciocínio sem interromper o processo, enquanto que na análise por interrupção o observador pode suspender o processo a fim de esclarecer o raciocínio das atividades e operações realizadas [OLI96].

Outra técnica que pode ser aplicada de forma complementar com o intento de permitir uma visão mais completa e ajustada do domínio do problema é a etnografia. Etnografia é a coleta direta, e o mais minuciosa possível, dos fenômenos observados, por uma impregnação duradoura e contínua e um processo que se realiza por aproximações sucessivas [LAP88]. É originária da antropologia, em observações de práticas das sociedades. No contexto da engenharia de requisitos, é uma técnica que busca revelar informações sobre a estrutura, organização e práticas do domínio do problema [SOM93].

Em um estudo de caso do projeto *Editorial Board*, da empresa *Film Board* [SIM97], foram aplicadas técnicas tradicionais, como reuniões, entrevistas e análises de documentos, e técnicas etnográficas, como observações e análises de gravações de vídeo, durante o projeto conceitual. Nesta aplicação pode-se concluir que as observações dos usuários envolvidos em suas atividades corriqueiras são necessárias para o estabelecimento de uma compreensão mútua do processo, no auxílio do entendimento da prática corrente e no desenvolvimento de uma visão geral dos objetivos do projeto. A utilização da etnografia também pode revelar múltiplos pontos de vista dos usuários sobre as atividades correntes e sobre os objetivos do projeto. Esses diferentes pontos de vista podem ser harmônicos ou problemáticos no sentido das possibilidades de integração em um sistema coerente, sendo que os conflitos devem ser analisados e apresentadas diferentes soluções e suas conseqüências, visando a busca de um consenso. A etnografia, segundo conclusão do projeto realizado, pode levar a obtenção de requisitos fechados à prática corrente, o conhecimento tácito.

A abordagem etnográfica aplicada em conjunto com outras técnicas de elicitação possibilita um entendimento completo das atividades realizadas e a minimização do conhecimento tácito, dada a grande integração ao ambiente. Trabalhos realizados utilizando-se essa abordagem têm comprovado a eficácia na obtenção dos requisitos pela participação do engenheiro de requisitos no ambiente de trabalho [DEB99].

Inquiry Cycle Model

O *Inquiry Cycle Model* é uma estrutura formal utilizada para descrever discussões sobre requisitos. Consiste em três atividades (figura 3) que se repetem sob uma estratégia de controle, nas quais os envolvidos propõem e discutem seus requisitos e refinam-nos quando aprovados [POT94].

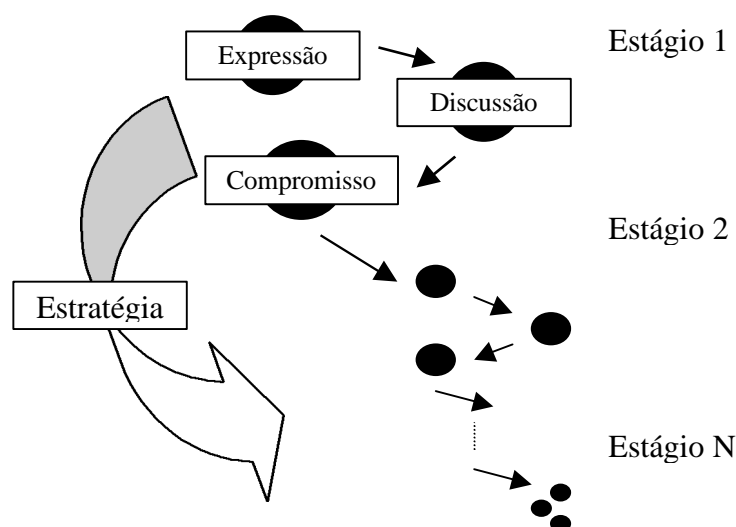


Figura 3. Fases do *Inquiry Cycle Model*

Na fase de *expressão* são apresentadas informações relacionadas ao domínio do problema, com o objetivo de analisá-las cooperativamente. Na fase de *discussão* são debatidos os comentários e anotações individuais sobre a análise da expressão. Na fase de *compromisso* são tomadas as decisões resultantes das discussões, acordadas as terminologias e comprometida a procura pelas informações ausentes. Cada ciclo completo pelas três fases finaliza um estágio de investigação. O progresso no cumprimento dos estágios segue uma estratégia que visa a aquisição completa e o refinamento dos requisitos capturados.

O *Inquiry Cycle Model* pode ser instanciado por diversas técnicas e métodos, dependendo da aplicação. É uma técnica adequada a ser aplicada visando a estruturação dos requisitos segundo o modelo proposto. O fato do modelo propor uma hierarquia de requisitos que parte de um objetivo até a especificação de operações de software é complacente à técnica, que estabelece uma estratégia de refinamento dos requisitos elicitados.

Bibliografia

- [ALE99] Alencar, F.M.R. & Castro, J.F.B. *Integrating Early and Late-Phase Requirements: A Factory Case Study*. Anais do XIII Simpósio Brasileiro de Engenharia de Software, Florianópolis, Outubro, 1999.
- [BER98] Bertolin J.C.G. *Uma Análise de Técnicas da Psicologia para a Elicitação de Requisitos de Software*. Seminário de Andamento de Mestrado, III Semana Acadêmica, Programa de Pós-Graduação em Computação da UFRGS, Porto Alegre, Outubro, 1998.
- [CAL99] Caldas Jr., J. & Masiero, P.C. *Obtenção de Modelo de Análise OO a Partir de um Modelo de Requisitos Baseado em Cenários*. Anais do II Workshop Iberoamericano de Engenharia de Requisitos e Ambientes de Software, Alajuela, Costa Rica, Marzo, 1999.
- [CAR93] Carmel, E., Whitaker, R.D. & George, F.J. *Participatory Design and Joint Application Design: A Transatlantic Comparison*. Communications of the ACM, vol. 34, no. 4, June, 1993.
- [CAS95] Castro J.F.B. *Introdução a Engenharia de Requisitos*. Jornada de Atualização em Informática do XIV Congresso da Sociedade Brasileira de Computação, Canela, Agosto, 1995.
- [DEB99] De Bortoli, L.A. *Um Método de Trabalho para Auxiliar a Definição de Requisitos*. Dissertação de Mestrado, Programa de Pós-Graduação em Ciência de Computação da UFRGS, Porto Alegre, 1999.
- [GIL97] Gilb, T. *Towards the Engineering of Requirements*. Requirements Engineering Journal, vol. 2, no. 3, 1997.
- [GOG94] Goguen, J. *Requirements Engineering as the Reconciliation of Social and Technical Issues*. Requirements Engineering: Social and Technical Issues, Academic Press Inc., London, England, 1994.
- [HER96] Herlea, D.E. *Users' Involvement in the Requirements Engineering Process*. Knowledge Science Institute, University of Calgary, Canada, 1996.

- [IEEE97] IEEE. *IEEE Software Standards Collection*. IEEE Computer Society Press, Los Alamitos, USA, 1997.
- [JAC95] Jackson, M. *Software Requirements and Specifications - A Lexicon of Practice, Principles and Prejudices*. Addison-Wesley Publishing Company, USA, 1995.
- [LAP88] Laplatine F. *Aprender Antropologia*. Editora Brasiliense, São Paulo, 1988.
- [LEE98] Lee, W.J., Cha, S.D. & Kwon, Y.R. *Integration and Analysis of Use Cases Using Modular Petri Nets in Requirements Engineering*. IEEE Transactions on Software Engineering, vol. 24, no. 12, December, 1998.
- [LEF2000] Leffingwell, D. & Widrig, D. *Managing Software Requirements - A Unified Approach*. Addison-Wesley Longman Inc., USA, 2000.
- [LEI94] Leite, J.C.S.P. *Engenharia de Requisitos: Notas de Aula*. Pontifícia Universidade Católica, Rio de Janeiro, 1994.
- [LOU95] Loucopoulos, P & Karakostas, V. *System Requirements Engineering*. McGraw-Hill Inc., New York, USA, 1995.
- [MAC99] Macedo, N.A.M. & Leite, J.C.S.P. *Integrando Requisitos Não Funcionais aos Requisitos Baseados em Ações Concretas*. Anais do II Workshop Iberoamericano de Engenharia de Requisitos e Ambientes de Software, Alajuela, Costa Rica, Marzo, 1999.
- [OLI96] Oliveira, K.M. & Werneck, V.M.B. *Aquisição de Conhecimento: Velha Fórmula, Nova Aplicação*. Exame de Qualificação para Doutorado em Ciência da Computação, COOPE, Universidade Federal do Rio de Janeiro, 1996.
- [PIN98] Pinheiro, F.A.C. *Da Importância dos Modelos de Desenvolvimento de Software para a Engenharia de Requisitos*. Relatório Técnico, Departamento de Ciência da Computação, Universidade de Brasília, 1998.
- [POT94] Potts, C., Takahashi, K. & Antón, A.I. *Inquiry-Based Requirements Analysis*. IEEE Software, vol. 11, no. 2, February, 1994.
- [SIM97] Simonsen, J. & Kensing, F. *Using Ethnography in Contextual Design*. Communications of the ACM, vol. 40, no. 7, July, 1997.
- [SOM93] Sommerville, I. et al. *Ethnography into the Requirement Process*. Proceedings of the First International Symposium on Requirements Engineering, San Diego, USA, 1993.
- [SOM97] Sommerville, I. & Sawyer, P. *Requirements Engineering: A Good Practice Guide..* John Wiley & Sons, Chichester, England, 1997.
- [WIE99] Wiegers, K.E. *Software Requirements*. Microsoft Press, Washington, USA, 1999.