ENGENHARIA DE REQUISITOS: UM FUNDAMENTO NA CONSTRUÇÃO DE SISTEMAS DE INFORMAÇÃO

WAGNER ZAPAROLI
Mestre em Ciência da Computação - MACKENZIE; Consultor de Sistemas; Colunista em Ciência e Tecnologia do jornal *Gazeta de Bebedouro*; Professor de Lógica de Programação na UNINOVE

Resumo

Os indícios da crescente utilização da tecnologia da informação por vários segmentos da sociedade, como parte integrante e definitiva de suas atividades, aliados à complexidade das soluções propostas aos problemas do cotidiano, fizeram surgir uma disciplina dentro da Engenharia de Software, denominada Engenharia de Requisitos. Este artigo analisa os motivos pelos quais sugere-se a utilização irrevogável dessa disciplina no processo de desenvolvimento de sistemas de informação. São apresentados como justificativas alguns números que indicam os prováveis responsáveis pelos fracassos de projetos sistêmicos, as principais características da Engenharia de Requisitos, seu mecanismo de utilização e, por fim, o esforço evolutivo para o bom uso das técnicas disponíveis.

Palavras-chave: *requisito; sistema; software.*

Abstract

The signs of the increasing use of information technology by several segments of society, as a definitive component of their activities, allied to the complexity of the proposed solutions for the day-to-day issues, brought a new subject within Software Engineering, called Requirement Engineering. This paper details the reasons for the irreversible use of this subject in the process of information systems development. Some numbers are presented to point the most probable factors for IT project failures. The main Requirement Engineering features, its usage and its evolution are presented as well.

Key Words: *requirement; system; software.*

Fundamentos históricos

É pouco provável que, nos dias atuais, engenheiros consigam construir prédios denominados 'arranha-céus', como o *Empire State* em Nova Iorque, sem uma base metodológica forte que os guie de forma precisa na formação de cada elemento dessas gigantescas construções. Analogamente à construção civil, poder-se-ia dizer o mesmo para os construtores aeronáuticos e navais. A complexidade sistêmica envolvida em cada projeto, o excessivo número de detalhes, as possibilidades explícitas de falhas – o que poderia pôr em risco a vida das pessoas – são motivos suficientes para que os profissionais envolvidos em tais empreitadas utilizem métodos e padrões de desenvolvimento rigorosos.

A engenharia civil, por ser uma disciplina milenar (as pirâmides do Egito são um bom exemplo), hoje possui um grau de evolução invejável em termos metodológicos. O número de falhas em construções de alto risco é mínimo e ocorre normalmente não por problemas de projeto, mas por uso de materiais indevidos ou incompletos. A engenharia aeronáutica, principalmente no que concerne à construção de foguetes, sofreu inúmeras perdas materiais e humanas antes de chegar a um nível evolutivo que lhe permitisse, por exemplo, enviar homens para além das fronteiras da atmosfera terrestre. Esse conhecimento adquirido empiricamente – entre erros e acertos – por longas décadas, culminou no surgimento de um conjunto de técnicas, métodos, padrões e ferramentas, nas mais diversas áreas do conhecimento, que auxiliam positivamente não só a construção de novos projetos, mas também, e principalmente, a sua manutenção.

Muitas das técnicas que auxiliam a construção e manutenção de um sistema, independentemente de sua natureza, são baseadas em modelos que representam (ou tentam representar) os conceitos, características e necessidades que ele deverá cumprir. Yourdon (1990) sugere três motivos gerais para a utilização de modelos na construção de sistemas:

a) o modelo focaliza as características importantes do sistema, dando menos atenção às de menor importância;

b) o modelo permite a discussão de modificações e correções das necessidades do usuário, com baixo custo e mínimo risco;

c) o modelo permite ao analista do sistema verificar se conhece corretamente o ambiente do usuário e se o documentou de tal maneira que os projetistas e desenvolvedores (programadores) possam construir o sistema.

Yourdon cita também que os motivos sugeridos servem ao desenvolvimento de qualquer tipo de sistema, desde os tradicionais sistemas de informação, passando pelas grandes edificações, até chegar às máquinas voadoras, como os ônibus espaciais. Entretanto, deve-se ressaltar que nem todos os modelos podem cumprir suas finalidades, principalmente se não forem bem construídos. Por exemplo, poderiam "(1) tornar completamente obscuras *todas* as características do sistema, (2) tornar a construção do sistema mais cara do que o próprio sistema, (3) falhar na verificação do entendimento do analista de sistemas sobre as reais necessidades do usuário" (YOURDON, 1990:83). Isso significa que construir um modelo de sistema é tão importante quanto construir o próprio sistema. A escolha de ferramentas de modelagem e metodologias para amparálas é um trabalho fundamental e delicado, pois muitas vezes, devido à complexidade dos sistemas a serem desenvolvidos, é necessário utilizar várias ferramentas com capacidade de interação entre si. Embora fornecedores

costumeiramente prometam tal interação, é necessário efetuar testes antes de adotá-las, verificando principalmente se atendem às necessidades do projeto.

Os sistemas computacionais

Com o surgimento dos computadores no âmbito comercial, na segunda metade do século XX, surgiram também os sistemas computacionais para suportá-los, cuja evolução, mesmo de forma precoce e rápida, seguiu de perto a dos próprios computadores, que também foi muito rápida em comparação, por exemplo, à evolução da engenharia civil ou das ciências médicas. No início, os sistemas computacionais serviam apenas para fazer os computadores funcionarem, sem muitos recursos visuais ou mesmo de processamento; resumiam-se a cálculos simplórios. Entretanto, com a introdução da microinformática nos anos 80, os sistemas computacionais requereram uma reformulação completa, desde a capacidade de armazenamento de informações até a de interatividade com os usuários. Foram criadas as denominadas 'interfaces amigáveis' para minimizar as dificuldades que pessoas leigas teriam em relação à operação dos computadores. A sofisticação dos computadores e seus sistemas chegaram ao ponto de os usuários interagirem com eles por meio de comandos de voz, sem a necessidade da operação manual.

Toda essa evolução implicou a criação de sistemas maiores e mais complexos. As exigências em termos de qualidade, segurança e facilidade de manutenção fizeram com que os elaboradores lançassem mão de algum tipo de apoio para desenvolver seus aplicativos. O mercado fornecedor, por sua vez, fez o seu papel: disponibilizou novas metodologias, novas técnicas e novas ferramentas capazes de suprir essa demanda. Entretanto, mesmo com toda a gama de ferramentas existentes, não é trabalho fácil escolhê-las e empregá-las no desenvolvimento de um projeto. As variáveis que impactam uma decisão de escolha são muitas e os riscos grandes, principalmente se o sistema a ser desenvolvido for complexo e de alta prioridade para o cliente.

Nessa grande 'torre de babel', alguns órgãos reguladores como a OMG (*Object Management Group*) e o PMI (*Project Management Institute*), em conjunto com a ABNT (Associação Brasileira de Normas Técnicas), anteciparam-se às dificuldades e criaram guias para difundir práticas consagradas de desenvolvimento e gerenciamento de projetos. O que ressalta às nossas vistas é o fato de os guias serem independentes de qualquer ferramenta de fornecedores específicos e, principalmente, possuírem abrangência global: podem ser utilizados no Brasil ou em qualquer outro lugar do mundo nos mesmos moldes e características.

Muitos dos conceitos apresentados nos guias fornecidos pelos órgãos reguladores têm ajudado a refinar e a completar o próprio conceito da Engenharia de *Software*. Dentre os vários assuntos tratados por esta área, um deles, a Engenharia de Requisitos – tema desse artigo –, acabou ganhando atenção especial, isso em razão da grande importância que o assunto 'requisitos' possui num processo de desenvolvimento de sistemas de informação. Normalmente um dos primeiros passos, quando se desenvolve um sistema, é o denominado 'elicitação dos requisitos', sinônimo para o levantamento das necessidades dos usuários do futuro sistema. A importância desse passo se deve aos custos de manutenção por erros cometidos nessa etapa, os quais são considerados os mais altos de todo o projeto, principalmente se forem descobertos nas fases finais do desenvolvimento.

A necessidade da Engenharia de Requisitos

O termo requisito não é novo na engenharia de *software*. Yourdon (1990) já descrevia a seqüência de passos de um modelo em cascata para o desenvolvimento de sistemas, como mostra a Figura 1.

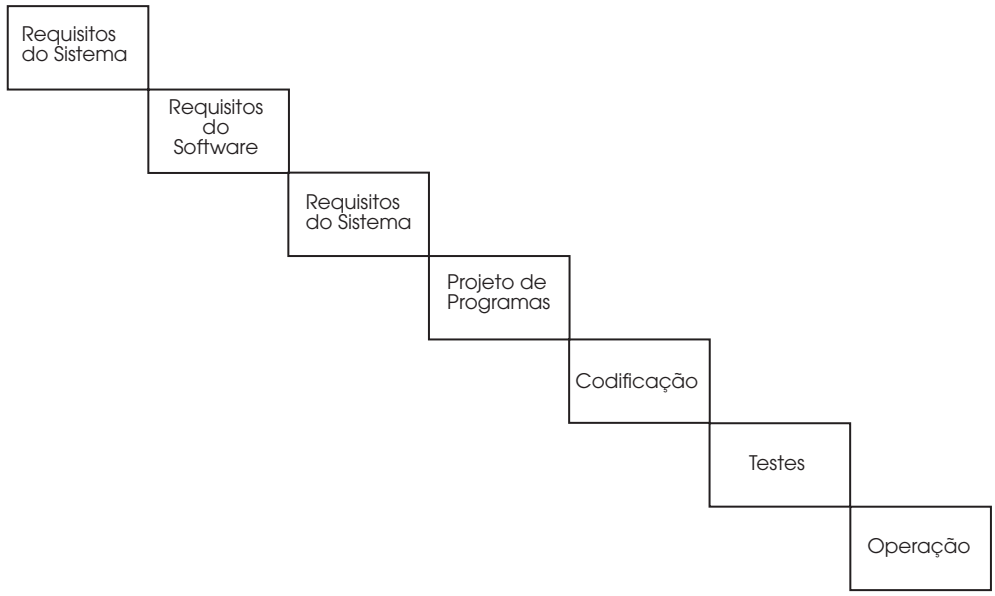

Figura 1 – Fases do Desenvolvimento de um Projeto em Cascata (YOURDON, 1990)
Fonte: elaboração própria

Apesar de ser um modelo superado para muitos dos projetos realizados atualmente, o desenvolvimento em cascata, quando criado, demonstrava explicitamente a preocupação com os requisitos, haja vista que, dos sete itens existentes no modelo, dois eram relacionados a eles.

Com a introdução do desenvolvimento iterativo, adjacente ao surgimento dos sistemas orientados a objetos, os requisitos foram abordados de uma forma mais enfática, sistemática e detalhada. Começaram não só a fazer parte do ciclo de vida do desenvolvimento de sistemas, mas também a implicar diretamente na arquitetura do projeto. Concomitantemente, surgiu a necessidade de realizar a gestão dos requisitos, principalmente pelas evidências fornecidas por pesquisas sobre custos de manutenção de erros, decorrentes de uma elicitação de requisitos deficiente. Alguns exemplos são claros: Pressman (1995) apresenta números do impacto de mudanças sobre os custos do projeto, comparando as principais fases do desenvolvimento de sistemas:

a) Definição: impacto de 1x;

b) Desenvolvimento: impacto de 1,5 – 6x;

c) Manutenção: 60 – 100x.

Por uma análise simples dos números, observa-se que, quanto mais tarde se detectarem erros ou necessidades de mudanças, maior será o impacto sobre os custos do projeto.

Castro (1995) nos fornece outros números sobre impactos de erros. Dos totais que ocorrem em projetos, normalmente 40% deles têm sua origem na fase de especificação; 30%, na fase de projeto, e 30%, na fase de codificação. Além disso, um erro que ocorre na fase de especificação custa muito mais do que um que ocorre na de codificação.

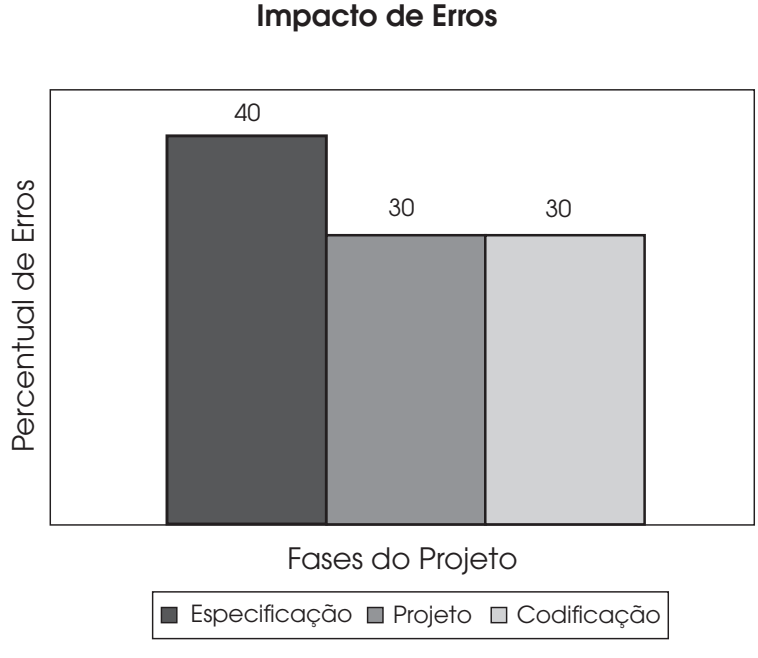

Figura 2 – Detalhe da ocorrência de erros por fase de projeto (CASTRO, 1995)
Fonte: elaboração própria

Para completar, o *Standish Group*, por meio de seu *Chaos Report* de 2000, informa que 74% dos projetos de *software* desenvolvidos falharam (RATIONAL, 2000). Os motivos relacionam-se, em grande parte, com objetivos não esclarecidos, ausência de planejamento, requisitos e especificações incompletos, ou falta de controle na mudança destes.

O resultado negativo das pesquisas, adicionado às dificuldades que as empresas têm em lidar com os requisitos, culminaram na necessidade de criação de uma disciplina denominada Engenharia de Requisitos. O objetivo prioritário dessa disciplina, regida pela Engenharia de *Software*, é normatizar o uso e a gestão dos requisitos.

Conceitos básicos da Engenharia de Requisitos

Um dos principais conceitos que envolvem a Engenharia de Requisitos é o do próprio requisito. Tal conceito assume objetivos e características variadas, dependendo de quem o utiliza e em que projeto o requisito é utilizado. Para Sommerville (1992), um requisito pode descrever uma propriedade geral do sistema, sua restrição específica ou ainda uma restrição no seu desenvolvimento. Para o *Institute of Electrical and Eletronics Engineers* (1998), um requisito pode ser uma representação documentada de uma condição ou capacidade da qual o usuário necessite para poder solucionar um problema ou alcançar um objetivo. Macauly (1996) faz uma referência simples ao requisito: "uma necessidade do cliente". Já Meyer (1988) afirma que elaborar um documento de requisitos é definir, de forma completa e não ambígua, as características externas do *software* oferecidas aos usuários e a forma pela qual este será integrado ao sistema.

Ainda em relação aos requisitos, a Rational (2000) afirma que projetos de *software* eficazes possuem as seguintes características:

a) os requisitos realmente devem refletir as necessidades dos clientes;

b) os requisitos devem ser compreendidos de forma coesa;

c) as expectativas dos clientes devem ser gerenciadas com eficácia;

d) as mudanças de requisitos devem ser gerenciadas.

Para que essas características sejam satisfeitas, o conceito de gerenciamento (ou análise) de requisitos deve ser claramente estabelecido. Segundo Pressman (1995: 232), a análise de requisitos de *software* pode ser dividida em cinco áreas: "(1) reconhecimento do problema, (2) avaliação e síntese, (3) modelagem, (4) especificação e (5) revisão".

Impacto de Erros

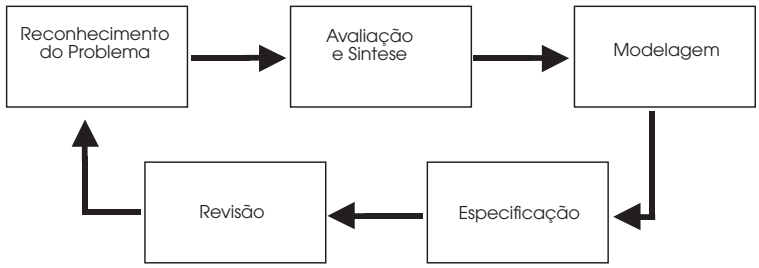

Figura 3 – Etapas para análise de requisitos (PRESSMAN, 1995)

Nessa mesma linha de raciocínio, a Rational (2000:5) define as chaves do gerenciamento de requisitos como sendo: "analisar o problema, compreender as necessidades dos envolvidos, definir o escopo do sistema, refinar a definição do sistema e gerenciar mudanças de requisitos".

Normalmente os envolvidos no processo de gerenciamento de requisitos são divididos em seis grupos: os executivos, os analistas, os desenvolvedores e projetistas, os responsáveis pelo controle de qualidade e testes, os documentadores e especialistas técnicos e, por fim, os gerentes de projeto.

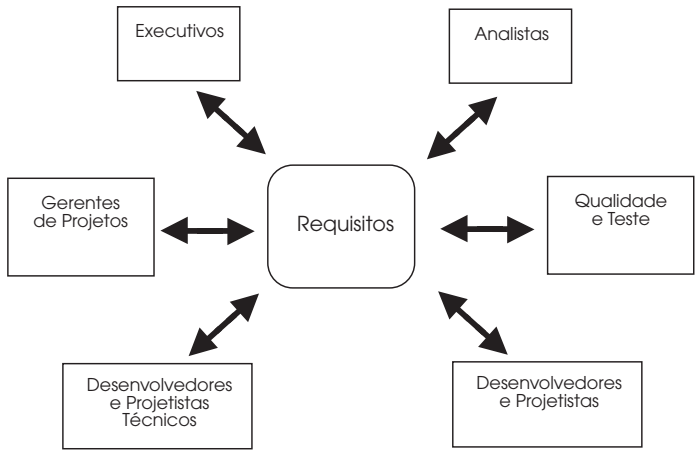

Figura 4 – Grupos de envolvidos no processo de gerenciamento de requisitos (RATIONAL, 2000)

Classificação dos requisitos

Em termos gerais, os requisitos são classificados em dois tipos: *funcionais* e *não funcionais*. Alencar (1999: 26) refere-se aos funcionais da seguinte forma: "dizem respeito à definição das funções que um sistema ou um componente de sistema deva fazer. Ou seja, descrevem as transformações a serem realizadas nas entradas de um sistema ou em um de seus componentes, a fim de que se produza

saídas". Já em relação aos requisitos não funcionais, o autor informa que eles "dizem respeito a restrições, aspectos de desempenho, interfaces com o usuário, confiabilidade, segurança mantenabilidade, portabilidade, padrões etc., bem como aspectos sociais e políticos" (ALENCAR, 1999:26). De forma geral, poder-se-ia associar o conceito de requisitos não funcionais à idéia de restrição sobre como os requisitos funcionais são implementados. Em outras palavras, os requisitos funcionais descrevem *o que* o sistema deve fazer, enquanto os não funcionais descrevem *como* os requisitos funcionais são implementados.

Numa linha de visão mais recente, que adiciona à engenharia de requisitos características relacionadas com as metas, políticas e estratégias da empresa, surgiu o conceito de requisitos organizacionais. Dobson (1994) define esse tipo de requisito como aquele que resulta de um sistema pretendido, considerando-se o contexto social no qual esse sistema está inserido. Alencar (1999: 28) cita alguns exemplos: "(a) obrigações e responsabilidades, (b) controle e autonomia, (c) valores e éticas etc., normalmente embutidos na estrutura e na política da empresa, de forma que não são diretamente observados ou facilmente articulados".

A introdução desse novo tipo de requisito aumentou naturalmente a abrangência da engenharia de requisitos. Aspectos gerenciais, organizacionais, econômicos, sociais e ambientais, não contemplados pela visão tradicional, foram adicionados ao seu contexto.

A arquitetura da engenharia de requisitos

Existem três aspectos fundamentais para a engenharia de requisitos ponderar sobre o desenvolvimento de sistemas: compreensão, descrição e concordância de um problema. Diversos modelos arquiteturais são sugeridos para abarcar esses aspectos, cada qual embasado em metodologias e técnicas específicas. Um deles, sugerido por Sommervile (1992), descreve uma arquitetura simplificada, composta somente de duas fases: a *aquisição*, dividida em elicitação e validação, e a *especificação*, dividida em análise e modelagem.

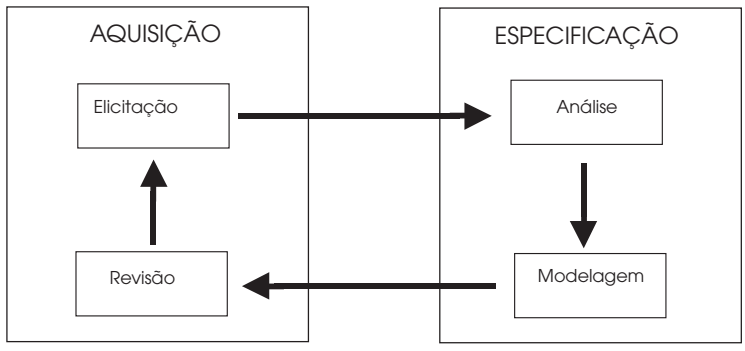

Figura 5 – Arquitetura Simplificada da Engenharia de Requisitos (SOMMERVILLE, 1992)

Em termos gerais, a fase de aquisição, como o próprio nome diz, é responsável pela aquisição (ou coleta) de informações referentes ao sistema a ser desenvolvido. O primeiro passo da aquisição é a elicitação, cujo objetivo principal é identificar os requisitos e respectivos detalhes. Entre os analistas, tal passo é popularmente conhecido como levantamento das necessidades. As técnicas mais comuns para a sua realização são: análise de conteúdo (dos documentos existentes), observação (do ambiente de trabalho dos usuários) e entrevistas (questionários, sessão JAD – *Joint Application Design* etc.). Yourdon (1990) cita os seguintes objetivos específicos desse passo: identificar os usuários responsáveis; desenvolver o escopo (inicial) do sistema; identificar as atuais deficiências no ambiente do usuário; estabelecer metas e objetivos para um novo sistema; determinar se é possível automatizar o sistema e, se assim for, sugerir alguns esquemas aceitáveis, e preparar uma previsão que será usada para conduzir o restante do projeto. Como produto do passo, tem-se uma lista dos requisitos obtidos. Tal lista poderia ser considerada, em alguns casos, um contrato entre desenvolvedor e usuário do sistema.

Ainda relacionado à aquisição existe um segundo passo denominado *validação*, responsável pela certificação dos requisitos obtidos, e fundamental para o desenvolvimento do sistema, considerando-se que, se algum erro persistir na fase de aquisição e se propagar para a de especificação, o custo do reparo será muito alto. A prototipação é uma das técnicas mais recomendadas para implementação desse passo.

A segunda fase da arquitetura da engenharia de requisitos – a especificação – tem como objetivo principal transformar os requisitos obtidos na fase de aquisição em produtos inteligíveis aos analistas implementadores. Tais analistas, de posse da documentação gerada, devem ser capazes de entender o sistema e implementá-lo (programá-lo) de acordo com as necessidades definidas pelo usuário. Os passos aqui executados – análise e modelagem – funcionam como se fossem um só. Na análise "são identificadas inadequações, contradições, ambigüidades e incompletudes que possam existir segundo a lista dos requisitos elicitados" (ALENCAR, 1999:24). A modelagem, por sua vez, é a representação (gráfica ou não) desses requisitos elicitados, numa linguagem que pode ser natural, rigorosa ou formal (*op.cit.*). As principais ferramentas disponíveis para a realização da especificação poderiam ser divididas em dois grupos distintos: as fornecidas pela análise estruturada e as sugeridas pela análise orientada a objetos. Em relação à análise estruturada, as ferramentas mais comuns são: diagrama de fluxo de dados, diagrama de entidades / relacionamentos, dicionário de dados e diagramas de transições de estado. Já em relação à análise

orientada a objetos, constituem ferramentas principais os seguintes diagramas: diagrama de casos de uso, de classes, de colaboração, de seqüência, de estados, de atividades, de pacotes e diagrama de utilização.

Questões da engenharia de requisitos

A engenharia de requisitos, criada em 1993 com a realização do *I International Symposium on Requirements Engineering*, embora se tenha desenvolvido rapidamente com a agregação de metodologias, técnicas e ferramentas que tratam dos requisitos e do desenvolvimento de sistemas, ainda contabiliza alguns problemas que persistem em grande parte nos projetos atualmente desenvolvidos. Alencar (1999) cita alguns deles:

- falta de envolvimento das partes participantes;
- falta de gerenciamento e rastreamento de requisitos;
- falta de definição das responsabilidades;
- falta de comunicação entre os envolvidos.

Apesar de tais problemas estarem relacionados direta ou indiretamente aos requisitos, sabe-se que alguns deles existem muito antes do surgimento da própria engenharia de requisitos, principalmente os que se referem às responsabilidades e à comunicação entre participantes do desenvolvimento. Entretanto, de 1993 até os dias atuais, muito se tem feito com o intuito de minimizar os reflexos da má especificação dos requisitos. Conferências e simpósios internacionais, como o Ideas – *workshop* Ibero-Americano de Engenharia de Requisitos e Ambientes de *Software*, ICRE - IEEE *International Conference on Requirements Engineering*, JIRA – *Jornadas de Ingeniería de Requisitos Aplicada*, WER – *Workshop* de Engenharia de Requisitos, entre outros realizados no mundo inteiro com a participação de grandes universidades, institutos de pesquisa e empresas privadas, têm produzido bons frutos para a engenharia de requisitos.

Inúmeras sugestões para solução de problemas, novos métodos e técnicas e, principalmente, novas ferramentas estão sendo produzidas em grandes proporções e postas à disposição do mercado consumidor. No entanto, vale uma ressalva: não existem produtos ou soluções milagrosos. Se a complexidade dos sistemas aumenta geometricamente, as soluções e produtos assim também o fazem. A exigência de equipes e gestores competentes e bem treinados já não é uma simples possibilidade; é um pré-requisito.

Referências bibliográficas

ALENCAR, F. M. R. *Mapeando a Modelagem Organizacional em Especificações Precisas*. 1999. Tese de doutorado. Centro de Informática.UFPE, Recife.

CASTRO, J.F. *Introdução à Engenharia de Requisitos*. XV Congresso da Sociedade Brasileira de Computação. XIV Jornada de Atualização em Informática – JAI. Canela, RS. 1995.

DOBSON, J. E. *et al. The ordit approach to rganizational requirements*. London: Academic Press, 1994.

Disponível em <http://www.er.les.inf.puc-rio.br/er_portugues.htm.> Engenharia de Requisitos. Acessado em out. 2002.

MACAULAY, Linda A. *Requirements Engineering*.1.ed. Great Britain: Springer-Verlag London, 1996.

MEYER, B. *Object-Oriented Software Construction*. New Jersey: Prentice Hall, 1988.

PRESSMAN, R. S. *Engenharia de Software*. 3. ed. São Paulo: Makron Books, 1995.

RATIONAL Software Corp. *O sucesso começa com o gerenciamento de requisitos*. São Paulo: Rational, 2000.

SOMMERVILLE, Ian; SAWYER, P. *Requirements Engineering: A Good Practice Guide*. New Jersey: Prentice Hall. Englewood Cliffs, 1992.

YOURDON, Edward. *Análise Estruturada Moderna*. 3. ed. Rio de Janeiro: Campus, 1990.