

## 2- Engenharia de requisitos

Este capítulo contém os principais conceitos sobre engenharia de requisitos. Apresentamos as principais atividades do processo de definição de requisitos e ressaltamos o uso da técnica de reuniões como uma estratégia para elicitação de requisitos.

### 2.1. Importância

A visão do software como um produto [Gunther 78] permite organizar o seu processo de criação de forma mais estruturada. Segundo [Leite 94], esta visão facilita também a melhor identificação dos aspectos que são software e dos aspectos que pertencem ao universo no qual o software será inserido.

Podemos notar que cada dia é mais freqüente encontrarmos o produto software inserido em uma série de outros produtos, desde carros, telefones até sistemas de informação organizacionais. Como de um produto exige-se preço e qualidade, o software tem que ter o nível de qualidade exigido e tentar ser desenvolvido com o menor custo possível. De acordo com [Leite 94], “esta é exatamente a função da engenharia, procurar sistemas de melhor qualidade dentro de um custo compatível com essa qualidade, otimizando a redução de custos”.

Porém, garantir a qualidade nos produtos e processos de engenharia de software não é uma atividade fácil. Existem inúmeros fatores que atrapalham no alcance dos objetivos de qualidade. No entanto, é necessário que estes objetivos sejam atendidos, já que não podemos negar o quanto é decepcionante entregar um software que não corresponda às necessidades dos clientes. Como um dos causadores desses problemas podemos citar a falta de atenção dedicada à tarefa de definir e acompanhar a evolução dos requisitos durante o processo de construção do software [Leite 01].

É fundamental que o software seja confiável. Isso significa ser eficaz e cumprir com os padrões exigidos pelo universo que será inserido. Freeman [Freeman 87] subdivide a qualidade em duas vertentes: básica e extra. Ele

identifica qualidade básica como: funcionalidade, confiabilidade, facilidade de uso, economia e segurança de uso. E qualidade extra como: flexibilidade, facilidade de reparo, adaptabilidade, facilidade de entendimento, boa documentação e facilidade de adicionar melhorias. O setor de software vem sendo muito pressionado pela sociedade para que se invista alto no quesito qualidade. Normas internacionais como a ISO e iniciativas como a da SEI (*Software Engineering Institute*) [Fiorini 98] são exemplo disso.

“A engenharia de requisitos é a disciplina que procura sistematizar o processo de definição de requisitos” [Leite 94]. Segundo o autor, [Leite 94] esta atividade é de extrema importância já que a complexidade dos sistemas exige uma maior atenção ao correto entendimento do problema antes do acordo de uma solução final. Esta é uma atividade que trata de conhecimentos não apenas técnicos, mas também gerenciais, organizacionais, econômicos e sociais [Castro 95], e estar intimamente associada a qualidade do software [Lee 98].

“A engenharia de requisitos estabelece o processo de definição de requisitos como um processo no qual o que deve ser feito é elicitado, modelado e analisado. Este processo deve lidar com diferentes pontos de vista, e usar uma combinação de métodos, ferramentas e pessoal. (...) Este processo acontece num contexto previamente definido a que chamamos de Universo de Informações” - segundo [Leite 94].

A engenharia de requisitos propõe um conjunto de métodos, técnicas e ferramentas para dar apoio ao processo de definição de requisitos. Neste capítulo descreveremos as principais atividades da engenharia de requisitos, explicando sua importância dentro do ciclo de desenvolvimento de um sistema.

## **2.2. Conceitos gerais**

Iniciaremos discursando um pouco sobre alguns conceitos básicos da área de engenharia de requisitos.

### **2.2.1. Especificações**

Segundo Alan Davis [Davis 90], em seu livro *Software Requirements*, uma *especificação* é simplesmente:

“A document; for example, a requirements specification is a document containing requirements; a design specification is a document containing the design; a test specification is a document containing the testing process”.

Já o autor Carlo Ghezzi [Ghezzi 91], em *Fundamentals of Software Engineering*, é mais exato:

“In general, we can view a specification as the statement of an agreement between a producer of a service and a consumer of the service, or an implementer and a user”.

*Especificações* podem ser entendidas, em uma idéia formal, como ambos requisitos e programa. É um requisito porque trata do contexto, e é um programa porque também trata de características da máquina. Isto ocorre porque especificações formam uma ponte entre requisitos e programas [Jackson 95].

De acordo com [Lawsweerde 02], de maneira geral uma *especificação* formal é a expressão, em alguma linguagem formal e de alto nível de abstração, de uma coleção de propriedades que determinado sistema deve satisfazer. Este sistema pode ser um modelo descritivo de um domínio de interesse, um determinado modelo de software e seu ambiente, a arquitetura do software ou um modelo de algum processo a ser seguido. As propriedades consideradas acima podem referenciar objetivos de alto nível, requisitos funcionais, requisitos não funcionais, hipóteses do ambiente, entre outros.

### 2.2.2. Requisitos

*The IEEE Standard Glossary of Software Engineering Terminology* [IEEE 97] define *requisito* como uma condição ou capacidade necessária para um usuário resolver um problema ou alcançar um objetivo.

Já Goguen [Goguen 94] define *requisito* como propriedades que um sistema de software deve ter em busca do seu êxito no ambiente onde será utilizado.

Conforme alguns autores [Macedo 99] [Gilb 97], os *requisitos de software* podem ser tanto as necessidades funcionais, identificadas como o comportamento e as propriedades do sistema, quanto as necessidades não funcionais, caracterizadas pelos quesitos de qualidade e restrições operacionais ou do desenvolvimento do software.

Apesar dos requisitos serem sobre o domínio da aplicação, não sobre a máquina, a máquina pode garantir que estes requisitos sejam satisfeitos, pois a máquina e os requisitos compartilham parte do domínio da aplicação, como alguns eventos e estados. Quando um evento compartilhado acontece, ele acontece em ambos, quando um estado compartilhado muda de valor, ele muda em ambos [Jackson 95].

### 2.2.3. Domínio

Um *domínio* é um conjunto de funcionalidades comuns entre múltiplos sistemas [Simos 96].<sup>1</sup>

A engenharia de domínio tem como finalidade a criação de artefatos de software para um conjunto de aplicações de um domínio. Ela trabalha na criação de soluções genéricas que possam ser reutilizadas por todas as aplicações deste domínio.

O foco da análise de domínio é o suporte para o reuso sistemático e em larga escala, através da captura das características comuns e das variabilidades entre os sistemas do domínio.

A análise de domínio também pode ser definida como uma forma de criar uma boa infra-estrutura de reuso que permita a especificação e a implementação de aplicações dentro de um escopo definido (o *domínio*) [Arango 94].

## 2.3. Elicitação de requisitos

Dentro da engenharia de requisitos cabe à elicitação a tarefa de identificar os fatos que compõem os requisitos do sistema, de forma a prover o mais correto e mais completo entendimento do que é demandado daquele software. A captura de informações é um processo de descoberta que objetiva aumentar o conhecimento do objeto em questão, buscando o máximo de informações possível. Para isso é necessária uma habilidade para trabalhar com especialistas humanos e com o conhecimento tácito, que é trivial para quem tem o conhecimento, mas não é para

---

<sup>1</sup> Podemos notar que a definição de domínio de Jackson é diferente, pois ele define domínio como o contexto em que a aplicação se encontra.

quem o procura, sendo dificilmente lembrado e, portanto, não transmitido [Goguen 94].

Segundo [Leite 94] a seguinte citação é atribuída a Von Neumann. Na citação ele deixa de forma clara a importância da tarefa de elicitar na definição de requisitos:

*“There is no sense in being precise about something when you do not even know what you are talking about.”*

Leite [Leite 94] chama de *elicitação de requisitos* às atividades que envolvem a descoberta de requisitos de um sistema, como a identificação das fontes de informação, coleta de fatos e comunicação.

Engenheiros e desenvolvedores de software trabalham diretamente com clientes e usuários finais na tentativa de achar problemas a serem resolvidos, serviços que o sistema deve prestar, requisitos de performance, restrições de hardware e outros. Para isso não é necessário apenas perguntar para as pessoas o que elas querem, requer uma análise muito mais cuidadosa da organização, do domínio da aplicação e dos processos de negócios onde o sistema será usado.

O processo de engenharia de requisitos como um todo é muito mais complexo do que apenas uma atividade de transferência de conhecimento entre clientes e engenheiros de requisitos. Primeiro porque os clientes dificilmente têm uma idéia clara do que realmente querem, e segundo porque diferentes pessoas dentro da organização normalmente possuem requisitos conflitantes, além de existir na maioria dos casos limitações tecnológicas influenciando os requisitos, entre outros.

A principal atividade no processo de engenharia de requisitos é definir o que os usuários realmente necessitam. Pesquisas mostram que grandes projetos falham por causa de requisitos errados [Boehm 81], além disso, esse problema é freqüentemente causado por fatores social, político e cultural.

Existem quatro dimensões na elicitação de requisitos [Sommerville 98]:

1. Entendimento do domínio da aplicação

Entender o domínio da aplicação é ter o conhecimento geral da área onde o sistema é aplicado.

2. Entendimento do problema

É necessário entender os detalhes do problema específico do cliente onde o sistema será aplicado.

### 3. Entendimento do negócio

Geralmente sistemas pretendem contribuir de alguma maneira para o desenvolvimento de alguma organização ou negócio. É importante saber como o sistema afeta e interage com as diferentes partes do negócio e como ele pode contribuir para melhorar as expectativas do negócio.

### 4. Entendimento das necessidades e restrições dos interessados no sistema.

Os interessados no sistema são pessoas que de alguma maneira são afetadas pelo sistema. Elas podem ser usuários finais, gerentes, entre outros. O engenheiro de requisitos deve entender, em detalhes, as necessidades específicas deste grupo, principalmente os processos de trabalho que o sistema pretende suportar.

Uma vez a necessidade expressada e um plano inicial desenvolvido, a equipe de requisitos tenta identificar quais propriedades o sistema deveria ter para satisfazer tal necessidade.

Propriedades relevantes podem incluir não apenas requisitos de alto nível, mas também tempo de resposta, segurança, portabilidade, confiança e manutenibilidade.

Como já foi dito, a tarefa de elicitação de requisitos está longe de ser uma tarefa fácil. Clientes podem mudar de idéia uma vez que vêem as possibilidades mais claramente, e descobertas feitas em fases muito tardias tendem a encadear uma volta para o aperfeiçoamento dos requisitos.

No contexto evolutivo é importante saber da impossibilidade da completeza de um conjunto de requisitos. Este conceito foi definido em [Leite 01] da seguinte forma:

“O processo de definir requisitos é inerentemente incompleto, tendo em vista a grande complexidade do mundo real. É obvio, no entanto, que sempre estaremos procurando ter requisitos os mais completos possíveis. A impossibilidade da completeza está relacionada a afirmação de Brooks [Brooks 87] sobre a bala de prata e a característica de homeostasia dos sistemas, ou seja, a característica evolutiva do software, que procura se adaptar ao seu macrosistema”.

É compreensível o motivo pelo qual os clientes freqüentemente não sabem exatamente o que necessitam. Isso pode ocorrer entre outros motivos, por uma necessidade de ver modelos, explorar alternativas e prever novas possibilidades,

atividades realizadas junto aos engenheiros de requisito. Frequentemente estas possibilidades são mescladas com fatores sociais, políticos, legais, financeiros, e psicológicos [Goguem 93].

Infelizmente um projeto pode vir a falhar porque nenhum sistema pode ser construído que satisfaça os requisitos solicitados, ou porque os requisitos acordados não refletem as necessidades reais [Goguem 93]. A Engenharia de requisitos tem como principal meta evitar tais problemas, o que frequentemente envolve significantes esforços na tarefa de elicitação de requisitos.

A seguir descreveremos como chegar às fontes de informação, apontaremos uma série de técnicas utilizadas para coletar os fatos do universo de informação (UdI), que são a base para descrever os requisitos do software, e também descrevemos técnicas que devem ser utilizadas na comunicação com os atores do UdI.

### **2.3.1. Identificação das fontes de informação**

Segundo Leite [Leite 94], o primeiro passo para identificação das fontes de informação é descobrir qual é o nosso Universo de Informações, que é o contexto no qual o software deverá ser desenvolvido e operado. Dele extrairemos as informações necessárias na tarefa de elicitação. A perfeita delimitação desse universo é função do macrosistema ao qual o software vai servir. O UdI contém todas as fontes de informação e todas as pessoas relacionadas com o software, também conhecidas como atores deste universo.

Em [Leite 94] é citado o trabalho de Burstin como uma técnica para identificar fontes de informação. Ele criou uma série de heurísticas para identificar atores e compor árvores desses atores. Basicamente, com o seu método, ele consegue mapear os atores de uma forma abstrata segundo uma hierarquia, na qual os atores que têm necessidades mais concretas aparecem nas folhas da árvore. Podemos visualizar melhor esta técnica através da Figura 1 tirada de [Leite 94].

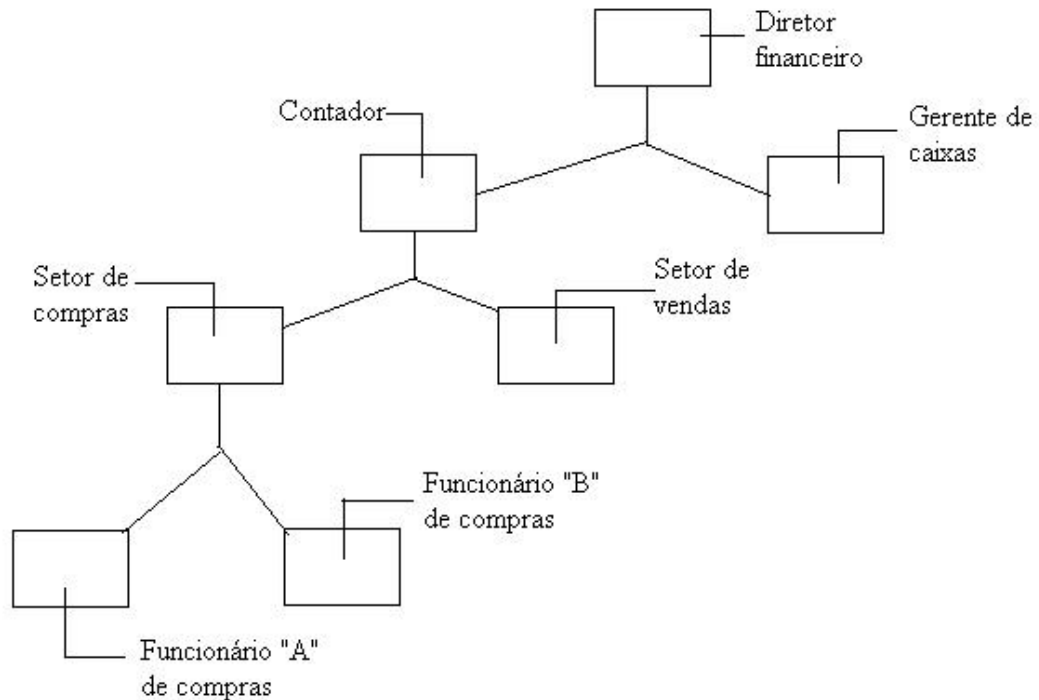


Figura 1 – Árvore abstrata de usuários [Leite 94].

Além de atores, outras importantes fontes de informação podem ser: documentos do UdI, livros sobre temas relacionados, outros sistemas já existentes na empresa e outros sistemas já existentes no mercado.

### 2.3.2. Coleta de fatos

Requisitos de software são sentenças que expressam as necessidades dos clientes e que determinam a qualidade do software. Eles podem ser classificados como: requisitos funcionais, que estão diretamente ligados a funcionalidade do software; e não funcionais, que expressam restrições que o software deve atender ou qualidade específicas que o software deve ter [Leite 01].

Desta forma, os critérios de qualidade que o software deverá ter é definido através dos requisitos não funcionais. É frequente o problema do não tratamento dos aspectos relacionados a qualidade do produto no início do processo de produção do software [Cysneiros 99] [Chung 00]. A estratégia de tratar a qualidade no próprio processo de definição do software integra de maneira coerente aspectos funcionais e não funcionais, podendo em um nível mais alto



trabalhar opções de solução que ficavam escondidas em muitos casos até a etapa de programação [Leite 01].

Para realizar a tarefa de elicitação dos requisitos, o engenheiro de requisitos pode necessitar usar diferentes técnicas para descobrir as informações necessárias para o entendimento do domínio da aplicação e do problema. A maior parte do conhecimento na elicitação dos requisitos vem da leitura de documentos sobre o sistema e de conversas (entrevistas) com pessoas que estão envolvidas com o sistema como usuários, gerentes, e outros. Isto resulta em um grande volume de informação que deve ser organizada para torná-lo compreensível.

Existem várias técnicas de coleta de fatos como leitura de documentos, observação, entrevistas, questionários, análise de protocolo, participação ativa dos atores do UdI, enfoque antropológico, reuniões, reutilização, recuperação do desenho de software, entre outras. Neste trabalho discutiremos mais detalhadamente de duas delas: questionários e reuniões.

### **2.3.2.1. Reuniões**

Realizar uma reunião é uma técnica que prevê a participação coletiva dos envolvidos, sendo uma maneira mais efetiva de negociar requisitos e resolver seus problemas. Elas têm a vantagem de permitir uma interação mais natural entre pessoas, do que as entrevistas estruturadas e até mesmo as não estruturadas e dispor de múltiplas visões sobre a questão abordada. Seu produto final é um documento que contém definições do software.

Reuniões 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 informação. Os participantes da reunião devem ser o engenheiro de requisitos que identificou algum problema nos requisitos e os interessados no sistema que podem ajudar a resolver os problemas identificados. É importante chamar pessoas de diferentes culturas e perspectivas para trazerem diferentes habilidades, conhecimentos de domínio e experiência para a reunião. Como resultado, os desenvolvedores aumentam seus conhecimentos sobre o domínio da aplicação e os usuários tornam-se mais envolvidos no processo de desenvolvimento.

A reunião como outras técnicas de elicitação tem que lidar com o problema do conhecimento tácito. Durante a reunião o engenheiro de requisitos deve explicitar cada problema encontrado nos requisitos e todos os interessados devem ter a oportunidade de comentar. Porém certamente os participantes serão incapazes de lidar com o conhecimento tácito. Mesmo com a ajuda de facilitadores não existe garantia que os participantes irão compartilhar conhecimentos com os outros. Além disso, podem existir na reunião participantes com hierarquias diferentes dentro da empresa, correndo o risco de alguns não se sentirem com liberdade pra dizer o que realmente pensam [Goguem 93].

Em uma reunião é também uma boa oportunidade para dar prioridades aos requisitos e decidir qual requisito pode ser eliminado e quais devem ser incluídos na especificação final do sistema.

Podemos classificar reuniões com base na maneira como elas são organizadas, podendo ser apenas uma extensão do conceito de entrevista ou como uma maneira de se ter a participação ativa dos atores do UdI. Técnicas de *brainstorm* podem ser utilizadas para a fase de exploração dos requisitos. Já métodos como JAD (*Joint Application Design*) [Andrews 91] são utilizados mais na fase de especificação de requisitos, por ser um método que estrutura a reunião com o objetivo de conseguir um clima de cooperação, utilizando representações de Engenharia de Software.

Segundo Leite [Leite 94], as principais vantagens da utilização de reuniões são a possibilidade de se dispor de múltiplas opiniões e da criação coletiva. Porém existem algumas desvantagens que são a possibilidade de dispersão e o custo.

Vamos voltar a tratar com mais detalhes sobre reuniões na seção 2.6.

### **2.3.2.2. Questionários**

Os questionários são mais utilizados quando é preciso ter acesso a um grande número de pessoas. São importantes na tentativa de compreender melhor como as pessoas percebem certos aspectos do universo de informação do software.

Segundo Leite [Leite 94], as principais vantagens da utilização de questionário são a padronização das perguntas e a possibilidade de tratamento

estatístico das respostas. A limitação do universo de respostas e a pouca interação, visto a impessoalidade da técnica, ficam entre as principais desvantagens.

### **2.3.3. Comunicação**

A comunicação eficiente entre clientes e engenheiros de requisitos é de extrema importância para o sucesso da atividade de elicitação. A existência de culturas diferentes entre clientes e engenheiros de software gerando diferentes conhecimentos por parte de cada um é um problema de difícil resolução [Leite 94].

Problemas na elicitação de requisitos são naturais e inevitáveis, pois eles refletem o fato que diferentes envolvidos no sistema tem diferentes necessidades e prioridades. Se não forem realizadas negociações abertas e explícitas, alguns interessados podem ficar decepcionados e hostis com o sistema.

O uso de técnicas de comunicação aumenta a possibilidade de uma comunicação mais clara e efetiva entre as partes. Apesar da ênfase dada a linguagem verbal é também importante nos preocuparmos com o uso da linguagem escrita, principalmente quando usamos a linguagem natural.

## **2.4. Análise de requisitos**

Nós discutimos a análise como se fosse uma atividade separada que segue a elicitação. Porém, na realidade, eles são processos interligados. Durante o processo de elicitação os problemas podem ser imediatamente reconhecidos, discutidos e resolvidos.

A análise e a negociação de requisitos têm o objetivo de estabelecer um acordo para a conclusão de um conjunto de requisitos o mais completo e consistente possível. Durante o processo de análise são identificados requisitos esquecidos, ambíguos, sobrepostos, conflitantes e não realistas. Se existe algum desses problemas com os requisitos, os interessados no sistema devem negociar e acordar modificações e /ou simplificações para eles.

As negociações dificilmente utilizam apenas argumentos lógicos e técnicos, pois também são influenciadas por considerações organizacionais e políticas e pela personalidade das pessoas envolvidas. Por exemplo, uma forte personalidade

pode forçar que suas prioridades prevaleçam em cima das dos demais, requisitos podem ser aceitos ou rejeitados por fortalecerem as influências políticas de alguns dentro da organização, usuários finais podem estar resistentes a mudanças e por isso bloquear requisitos, etc.

A tarefa de análise de requisitos pode ser dividida em: identificação de partes, verificação e validação [Leite 94]. Porém, existe um grande problema nesta divisão, que é como separar exatamente o que vem a ser verificação e o que vem a ser validação.

As atividades de validação e verificação têm muito em comum. Na primeira julgamos se os requisitos estão com as descrições feitas apropriadamente às necessidades dos interessados e a outra verifica problemas nos requisitos. Porém existem importantes diferenças para considerarmos elas separadamente.

O foco durante a verificação dos requisitos deve ser mais relacionado com a maneira que os requisitos são descritos. O objetivo é verificar a consistência, completeza e precisão no documento de requisitos.

Enquanto que o processo de validação deve garantir que os requisitos reflitam as necessidades dos interessados ao invés dos detalhes da sua descrição.

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 analisar se o que foi especificado realmente representa as necessidades do cliente.

Se não ocorre o processo de validação e o desenvolvimento começa muito cedo, alguns reparos podem ser inevitáveis. Como consequência de erros nos requisitos, os erros identificados na fase de programação podem custar 100 vezes mais para reparar [Sommerville 98]. Ocorre uma redução importante do re-trabalho, se estes erros e problemas são descobertos durante a validação dos requisitos [Sommerville 98].

Uma maneira de imaginar os processos de elicitação, análise e negociação de requisitos são como segmentos em uma espiral, como mostra a figura 2 retirada de [Sommerville 98], que foi inspirada no trabalho de [Boehm 88] sobre desenvolvimento de software.

Primeiro o engenheiro de requisitos descobre alguma informação sobre os requisitos. É feita uma análise, posteriormente uma negociação e então outro giro da espiral começa.

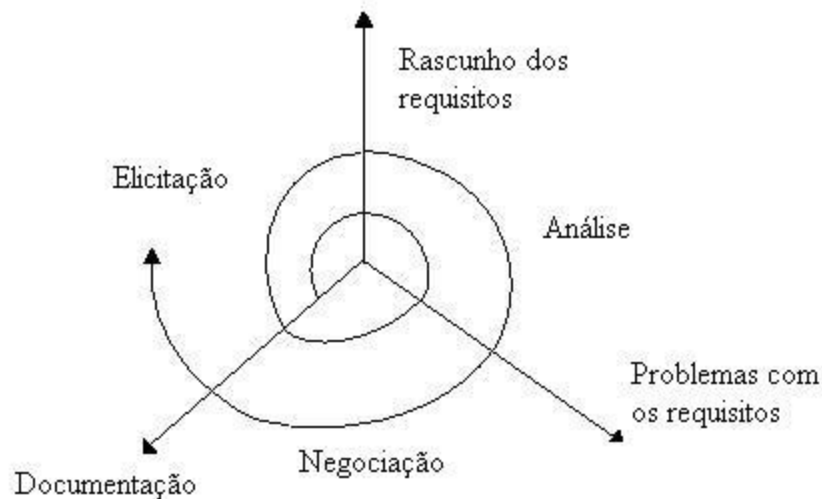


Figura 2 – O espiral da elicitação, análise e negociação

## 2.5. Principais problemas na elicitação de requisitos

Segundo [Curtis et al 88] os principais problemas na elicitação de requisitos são o conhecimento parcial ou incompleto do domínio da aplicação, requisitos oscilantes e conflitantes e falhas na coordenação e comunicação de atividades.

A seguir vamos explicar um pouco mais sobre cada um destes problemas.

### 2.5.1. Conhecimento parcial ou incompleto do domínio da aplicação

Mesmo que cada membro da equipe de elicitação de requisitos compreenda individualmente o domínio de uma aplicação, deve haver uma integração dos pontos de vista e uma difusão do conhecimento pelos demais participantes do projeto, porém isso normalmente não ocorre.

Não importa quanto um engenheiro de software sabe sobre técnicas de análise e programação, ele é um principiante no domínio da aplicação do negócio em questão.

Para construir um modelo do domínio que reflita um consenso entre os participantes devemos integrar os seus pontos de vista, que podem diferir no comportamento externo do sistema, no contexto onde o domínio está inserido ou

na representação mais apropriada do sistema. Um modelo integrado normalmente é uma solução mais completa e criativa do que cada solução individual.

Outro problema que causa uma informação parcial do domínio da aplicação e também é bastante freqüente, é a resistência apresentada por muitos usuários em compartilhar a informação necessária com o engenheiro de requisitos. Existe uma associação de poder e informação que geram nas pessoas um receio em compartilhar a informação para não perder poder [Mathias 94], seja para não perder o emprego ou mesmo para obter vantagens na organização.

### **2.5.2. Requisitos oscilantes e conflitantes**

Requisitos oscilantes são os que variam com o tempo devido às alterações externas à empresa (como mudanças na política econômica do país) ou devido a modificações estruturais internas (reestruturação da empresa ou corte de despesas) [Mathias 94].

Requisitos conflitantes são os que possuem inconsistência entre si e que precisam da utilização de técnicas de resolução de conflitos para integrá-los [Mathias 94].

Estes problemas são causados porque diferentes atores possuem diferentes pontos de vista e como normalmente requisitos são definidos com entrevistas individuais com os usuários, estes problemas se agravam. Dessa forma os requisitos levantados por um usuário que foi entrevistado mais tarde podem conflitar com os elicitados por outros anteriormente, necessitando assim de métodos integrativos para produzir uma solução global para todos.

Diversas podem ser as causas dos requisitos oscilantes, por exemplo, o usuário durante o processo de elicitação pode acabar aprendendo ou recordando detalhes do universo de informação, apresentando e alterando idéias já levantadas. Outro motivo pode ser a ausência de objetivos bem definidos para o sistema.

### **2.5.3. Falhas na comunicação e coordenação das atividades**

Muitos engenheiros de software durante a elicitação de requisitos não utilizam documentação formal, confiando na memória e em algumas anotações em rascunho, gerando uma documentação deficiente do sistema [Rosson et al 88].

Outro problema de comunicação é a falta de divulgação de algumas informações por desejo de obtenção de vantagens políticas, por necessidades de manter sigilo sobre partes do projeto, ou até mesmo por acharem elas desnecessárias. Isso influencia muito nos requisitos do sistema já que os engenheiros de requisitos ficam sem acesso ou tem o acesso tardio a algumas informações.

Muitos engenheiros de requisitos preferem trabalhar sozinhos e desenvolverem sistemas através de suas próprias idéias em vez de trabalhar em conjunto com o usuário. O que acarreta em um sistema que irá satisfazer as vontades do engenheiro, não do usuário. Portanto, o ideal é que o contato entre eles seja constante e direto.

O modelo de precisão, proposto por [McMaster, Grinder 80], utiliza técnicas de comunicação para facilitar a tarefa de elicitação de requisitos. As reuniões entre engenheiros de software e clientes são os principais focos desse modelo, que tenta compreender os processos de comunicação e de aquisição do conhecimento.

## **2.6. A reunião segundo o método de [Mathias 94]**

Como já foi dito, o método que estudamos nesta dissertação se baseia em um ciclo de reuniões de elicitação de requisitos apoiada pelo gerenciamento de conflitos, que tenta atuar nos problemas que listamos acima.

Falaremos nesta seção sobre conflitos, causas e métodos de resolução para facilitar o gerenciamento de conflitos, pois, muitas vezes, eles permanecem sem solução por desconhecimento das causas e técnicas adequadas.

Primeiro descreveremos o conceito de conflito, depois citaremos suas principais causas e logo a seguir as melhores técnicas para solucionar cada tipo de conflito existente. No final mostraremos um pouco de como o conflito tem sido tratado na atividade de elicitação de requisitos.

Um dos principais objetivos desta seção é mostrar que existe um tipo de conflito que deve ser estimulado e trabalhado como um importante trunfo para a aquisição do conhecimento.

### 2.6.1. O que é Conflito

Pode se chamar de conflito qualquer tipo de oposição ou interação antagônica originada por diversos motivos, tais como estruturas de valores diferentes, recursos ou posição social e disputa de poder [Robbins 74].

Na literatura sobre conflitos um dos pontos mais importantes é a distinção feita entre conflito funcional e conflito não funcional. Segundo [Robbins 74], o conflito funcional é aquele que o confronto beneficia ou auxilia nos objetivos da organização. Qualquer outro tipo de conflito que não se encaixe nesta definição não é um conflito desejável, deve ser tratado e eliminado, já que não objetiva trazer benefício para a organização, no máximo, para interesses individuais.

O conflito funcional estimula o interesse e a curiosidade, impede a estagnação, ajudando a realizar boas mudanças sociais e pessoais. Realizando a direta expressão de idéias opostas, os sistemas sociais reajustam suas estruturas, eliminando as fontes de insatisfação [Mathias 94].

É muito comum ocorrer a intolerância ao conflito na sociedade atual, pois a educação que nossos pais nos ensinam é que devemos nos relacionar bem uns com os outros, evitando qualquer tipo de confronto. Como por exemplo, discussões com colegas da escola ou com os próprios irmãos. Também podemos citar como exemplo algumas escolas da sociologia, onde o conflito é enxergado como o desequilíbrio de forças do sistema social que deveria estar em repouso, isto é, equilibrado quanto às forças que o compõem [Wikipedia 05]. Desta forma, encaramos o conflito como algo sempre negativo que deve ser eliminado, esquecendo e reprimindo os conflitos positivos, ou seja, os funcionais. Porém devemos compreender que através da aceitação da existência do “bom conflito” (conflito funcional) e do estudo deste, podemos evitar problemas destrutivos, oferecer um grande número de idéias a partir do dinamismo gerado em busca de soluções e estimular a percepção.

Segundo uma filosofia que reflete atitudes perante o conflito, chamada de interacionista [Robbins 74], o benefício do conflito deve ser reconhecido e encorajado. Ela define o gerenciamento de conflitos como uma técnica que combina as técnicas de estímulo e resolução dos mesmos. Pela visão interacionista existe uma quantidade ideal de conflito, necessitando de métodos de resolução



quando está alto demais ou de estímulo por técnicas adequadas quando está muito baixo.

A filosofia interacionista acredita que as organizações devem estimular o conflito para não sofrerem por causa de pensamentos estagnados, decisões inadequadas e desmotivação, pois as boas mudanças acontecem a partir da insatisfação, da vontade de progredir e do desenvolvimento de alternativas criativas.

## **2.6.2. Causas do Conflito**

Segundo [Robbins 74], o conflito pode ser causado por três motivos: fatores de comunicação, estruturais e de comportamento pessoal.

### **2.6.2.1. Fatores de comunicação**

A retroalimentação, que é a etapa em que o receptor se torna emissor e transmite o que compreendeu da mensagem, é uma atividade importante para verificar se o receptor entendeu o que foi passado. Porém ela não é garantia de sucesso na comunicação, pois como o receptor se torna um emissor, ele fica sujeito aos mesmos problemas do emissor original.

Problemas na comunicação podem ser causados porque a mensagem teve que passar por vários intermediários até alcançar o receptor final ou por ruídos no canal de comunicação. Alguns comportamentos do receptor como a desatenção, avaliação prematura da mensagem, a interrupção do interlocutor ou a existência de um antagonismo entre eles, também podem causar problemas na comunicação.

### **2.6.2.2. Fatores estruturais**

Estes conflitos são aqueles causados por questões relacionadas à própria estrutura da organização.

Segundo [Mathias 94], Katz distingue três tipos de conflitos com causas estruturais:

- Conflito causado pela existência de vários subsistemas dentro da organização.

Tipos distintos de orientação, para fora ou para dentro da organização, são fonte potencial de conflito, pois normalmente suas normas e padrões de referência entram em choque.

- Disputas internas entre unidades dentro de uma organização.

Organizações grandes normalmente contêm duas ou mais unidades com funções parecidas, o que pode, em muitos casos, causar conflitos não funcionais em busca de um maior destaque na organização.

- Conflito hierárquico por recompensas organizacionais.

Em uma organização comum é inevitável que indivíduos em cargos diferentes recebam diferentes salários. Esta distinção de remuneração pode gerar conflitos com o objetivo de obter mais poder, prestígio e vantagens e também dificultar eventuais alterações na estrutura organizacional, por parte de alguns resistentes a mudanças.

### **2.6.2.3. Fatores pessoais**

Podemos citar valores e pensamentos diferentes entre duas ou mais pessoas, fatores pessoais causadores de conflito, pois cada um tem o interesse que o seu valor ou pensamento seja o dominante.

Por exemplo, pessoas de personalidade agressiva ou dominadora podem causar conflitos não funcionais ao tentar dominar outros participantes. Já pessoas extremamente tímidas ou resistentes a mudanças podem dificultar o trabalho cooperativo, reduzindo a qualidade dos resultados de um trabalho em equipe.

Os membros de uma equipe podem ter personalidades diferentes, porém eles têm que saber trabalhar em grupo, ouvir com atenção, ser firme sem magoar, evitando assim reações emocionais prejudiciais. Desta forma conflitos interpessoais não funcionais certamente serão menos freqüentes.

### **2.6.3. Técnicas de resolução de conflitos**

Nesta seção abordaremos as principais técnicas de resolução de conflitos e suas características. Ao apresentar cada técnica, iremos comentar qual forma de conflito melhor seria resolvido com ela.

#### **2.6.3.1. Negociação**

A negociação tenta solucionar o conflito através da análise de todas as possibilidades existentes, na tentativa de descobrir uma solução que satisfaça a todos a medida do possível.

Esta técnica é indicada quando existe um relacionamento satisfatório entre os participantes, permitindo uma comunicação aberta. É uma técnica das mais democráticas, mas pode demorar muito para atingir solução, pelos inúmeros impasses que podem ocorrer durante a negociação.

#### **2.6.3.2. Técnicas com um participante externo**

Quando os participantes não conseguem sozinhos resolverem o conflito é necessário então o auxílio de uma fonte externa, que agirá com coerção sobre os participantes com o objetivo de chegar a uma solução.

O profissional ou pessoa atribuída para ser o participante externo deve ter algumas qualidades como elevada competência profissional em relação a processos sociais, bom conhecimento dos indivíduos envolvidos no conflito e saber manter a neutralidade para facilitar o desenvolvimento da confiança pelos participantes [Mathias 94].

Esta técnica é bastante indicada, desde que o mediador atenda aos requisitos citados anteriormente, pois como ele é imparcial tem mais facilidade para enxergar soluções mais eficientes, já que analisa objetivamente o problema.

#### **2.6.3.3. Esclarecimento do problema**

Os participantes, de forma cooperada, buscam toda a gama de alternativas e similaridades nos seus pontos de vista. Para desta forma, tentar entender o motivo

das dúvidas e mal entendidos que estão sendo gerados, tornando possível a resolução através do melhor conhecimento do contexto onde o conflito se encontra [Robbins 74].

Por se tratar de uma técnica que prioriza a comunicação, ela é altamente recomendável mesmo se as relações entre os envolvidos não for muito boa. Pois é possível durante o processo, que surja compreensão entre os participantes ou até que seja percebido que o conflito não passou de um mal entendido.

#### **2.6.3.4. Expansão e substituição de recursos**

Quando o problema for falta de algum recurso, ele é resolvido simplesmente aumentando o seu número. Porém dependendo do recurso escasso esta solução pode se tornar bastante cara ou, dependendo do caso, inviável.

Se a expansão for impossível pode ser tentada a substituição do recurso escasso por outro com características similares, mas corre-se o risco do recurso similar não ser aceito pelo grupo.

Esta técnica pode ser empregada também para o caso de um conflito gerado pela baixa qualidade de um recurso. Neste caso a chance de sucesso é muito maior, considerando que o novo recurso satisfaça os quesitos de qualidade desejados pelos envolvidos.

#### **2.6.3.5. Votação**

Esta é uma técnica bem simples onde cada pessoa envolvida deve manifestar de forma aberta ou secreta sua opção. Após todos os votos apurados a opção da maioria é adotada.

Esta técnica tem um ponto negativo, pois raramente satisfaz a todos. Ela só deve ser aplicada após o insucesso na tentativa de outros métodos, como a negociação, por exemplo.

#### **2.6.3.6. Alteração da variável humana**

Apesar de ser uma técnica lenta e cara, é recompensada pelos seus resultados significativos, pois atua na causa do problema [Robbins 74]. Ela visa

alterar o comportamento de uma ou mais pessoas envolvidas no conflito. Por isso ela é mais eficiente quando aplicada em conflitos devido a fatores pessoais.

#### **2.6.3.7. Alteração de variáveis estruturais**

Através de alguma alteração na estrutura da organização esta técnica tenta tratar principalmente os conflitos relacionados com causas estruturais, como alteração de normas vigentes, transferência de membros para outros departamentos, etc. Para maior eficiência, esta técnica deve ser realizada por uma pessoa com bom conhecimento da organização.

#### **2.6.4. O conflito na elicitação de requisitos**

O conflito está presente em todas as áreas que o homem atua. Aqui, destacaremos a sua importância na área de Informática, mais especificamente, na área de Elicitação de Requisitos.

No desenvolvimento de um sistema, surgem boas idéias por parte de cada participante no processo de aquisição de conhecimento. Porém estas idéias na maioria das vezes não se encaixam perfeitamente, deixam lacunas ou se sobrepõem em algumas partes, o que é normal.

Existem alguns conceitos que justamente visam ajudar neste processo de aquisição de conhecimento. Um deles é o conceito de pontos de vista, que trata das características particulares de cada pessoa. Segundo [Mathias 94], Nygaard diz que elas influenciam os aspectos que parecem relevantes a cada um, o que resulta em distintas conclusões sobre o mesmo tema. Desta forma, na tentativa de adquirir um conhecimento mais profundo podemos elaborar, comparar e combinar visões da mesma área de interesse a partir de diferentes ângulos [Leite 88].

Acredita-se [Mathias 94] que é importante dar mais atenção às relações contraditórias em uma organização com o objetivo de serem usadas como importante fonte de aquisição de conhecimento. Ao contrário das metodologias tradicionais, que sempre pregaram os conflitos como problemáticos, eles devem ser inseridos como parte integrante do desenvolvimento de sistemas. Sendo utilizado como uma alternativa para surgimento de novas idéias, ajudando a obter requisitos de melhor qualidade e mais adequados à organização. Um ótimo

exemplo é a técnica de reuniões conhecida como *brainstorming*, essa técnica é usada basicamente para maximizar a geração de idéias provenientes de um grupo de pessoas, sem nenhuma preocupação crítica, até que se esgotem as possibilidades.