

EMILIANE SILVA SOARES

**SWREQUIREMENT: UMA PROPOSTA DE INTEGRAÇÃO DE TÉCNICAS DE
ELICITAÇÃO DE REQUISITOS AO PROCESSO DE LEVANTAMENTO E
ANÁLISE DE REQUISITOS**

Monografia de graduação apresentada ao Departamento de Ciência da Computação como parte das exigências do Curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

LAVRAS
MINAS GERAIS-BRASIL
2007

EMILIANE SILVA SOARES

**SWREQUIREMENT: UMA PROPOSTA DE INTEGRAÇÃO DE TÉCNICAS DE
ELICITAÇÃO DE REQUISITOS AO PROCESSO DE LEVANTAMENTO E
ANÁLISE DE REQUISITOS**

Monografia de graduação apresentada ao Departamento de Ciência da Computação como parte das exigências do Curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Área de concentração:

Engenharia de Software

Orientador:

Prof. Dr. Heitor Augustus Xavier Costa

LAVRAS
MINAS GERAIS – BRASIL
2007

**Ficha Catalográfica preparada pela Divisão de Processo Técnico da Biblioteca
Central da UFLA**

Soares, Emiliane Silva

SWREQUIREMENT: Uma Proposta de Integração de Técnicas de Elicitação de Requisitos ao Processo de Levantamento e Análise de Requisitos / Emiliane Silva Soares – Minas Gerais, 2007, 47p.

Monografia de Graduação – Universidade Federal de Lavras. Departamento de Ciência da Computação.

1. Elicitação de Requisitos. 2. Técnicas de Elicitação de Requisitos. 1. SOARES, E. S. II. Universidade Federal de Lavras. III. Título.

EMILIANE SILVA SOARES

**SWREQUIREMENT: UMA PROPOSTA DE INTEGRAÇÃO DE TÉCNICAS DE
ELICITAÇÃO DE REQUISITOS AO PROCESSO DE LEVANTAMENTO E
ANÁLISE DE REQUISITOS**

Monografia de graduação apresentada ao Departamento de Ciência da Computação como parte das exigências do Curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Aprovada em 06 de agosto de 2007

Prof. M. Eng. Paulo Henrique de Souza Bermejo

Prof. Dr. Antônio Maria Pereira de Resende

Prof. Dr. Heitor Augustus Xavier Costa
(Orientador)

LAVRAS
MINAS GERAIS – BRASIL
2007

Dedico esse trabalho a minha mãe Maria da Conceição Silva Soares, por sempre ter me apoiado em todos os momentos da minha vida, proporcionando que essa conquista fosse realizada. Mãe amo você!

*Agradeço primeiramente a Deus por sua grandeza e sua infinita bondade e compaixão,
sem o qual nada seria possível.*

A minha mãe por todo apoio e amor incondicional.

*Agradeço aos meus irmãos Janinha e Josinha tão solidários e afetuosos, por sempre
acreditarem em mim.*

*Agradeço por toda a alegria proporcionada pelas três pessoas mais lindas do mundo Neto,
Lucas e Ana, meus amores.*

*A Rachel por ser tão amiga, compreensiva e ter colaborado para que esse trabalho fosse
realizado.*

Ao meu cunhado Agildo pelo carinho e a amizade.

A toda minha família sempre tão alegre.

A Família 303 com quem compartilhei inúmeras emoções e convivi com muita alegria.

A minha turma da computação que me ensinou muitas lições.

A SWFactory por proporcionar novas oportunidades de aprendizagem e amadurecimento.

Aos meus amigos e companheiros de trabalho, por tudo.

Aos meus amigos do Brejão, grandes momentos.

*Queria agradecer a alguns amigos muito presentes neste meu momento: Michelle por todo
companheirismo e amizade, Lucinéia por sempre ser mais que uma amiga, Bianca tão
amiga e atenciosa, ao grande afeto demonstrado por Milena, Quedes, Raquel, Felipe,
Karen, Vladimir, Ana, Fábio, Gleimar, Gregório, Viviane, Flávia, Sheila, Glasi, Paula,
Breno, André, Leonardo e Janaína.*

*A todos os meus amigos que engrandecem minha vida, fazendo com que as dores passem
rapidamente e as alegrias se eternizem em cada sorriso.*

*Ao meu orientador, professor Heitor Augustus Xavier Costa, por toda a paciência,
disponibilidade e principalmente por me apoiar e auxiliar na construção desse trabalho.*

*Aos demais professores do Departamento de Ciência da Computação da UFLA, a Ângela
e Deivson.*

Enfim, agradeço de coração a cada um que participou desta conquista.

Só enquanto eu respirar vou me lembrar de vocês!

SWREQUIREMENT: UMA PROPOSTA DE INTEGRAÇÃO DE TÉCNICAS DE ELICITAÇÃO DE REQUISITOS AO PROCESSO DE LEVANTAMENTO E ANÁLISE DE REQUISITOS

RESUMO

Na atualidade, as empresas desenvolvedoras possuem dificuldade em especificar os requisitos dos sistemas para atender aos seus clientes de forma satisfatória e apresentar aos desenvolvedores o conteúdo necessário para o desenvolvimento. Um processo de elicitação de requisitos bem definido e estruturado pode auxiliar na busca de requisitos detalhados claramente, facilitando a fase de desenvolvimento do software. Esta monografia apresenta **SWRequirement**, resultado da integração de algumas técnicas de elicitação de requisitos (entrevista, questionário, JAD, *brainstorming* e prototipação) acomodada no processo de levantamento e análise de requisitos, tentando minimizar os principais problemas que aparecem durante esse processo. Para alcançar esse propósito, foi feito um estudo detalhado das técnicas de elicitação de requisitos mais usadas e mais aplicáveis ao processo de elicitação de requisitos sugerido, essas técnicas foram divididas em etapas as quais foram integradas ao processo conforme a qualificação e adaptação a ele.

Palavras-chave: Elicitação de Requisitos, Técnicas de Elicitação de Requisitos, Processo de Levantamento e Análise de Requisitos.

SWREQUIREMENT: A PROPOSAL OF INTEGRATION OF REQUIREMENT ELICITATION TECHNIQUE TO REQUIREMENT ANALYSES PROCESS

ABSTRACT

Nowadays, software developer organizations have difficulty to specify software requirements to support the client needs. A well-defined and well-structured requirement analysis process can help in the search of more detailed requirement; so, it facilitates the software development. This work shows **SWRequirement**, which is the result of the integration of some requirement elicitation technique (interview, questionnaire, JAD, *brainstorming* e prototyping). To obtain a important result, it was accomplished a study detailed of requirement elicitation technique more used and more applied to requirement analysis process suggested. These techniques were divided in phases to integrate to the process conform the quality and the adaptation to it.

Key word: Requirement Elicitation, Requirement Elicitation Technique, Requirement Analyses Process

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1	INTRODUÇÃO	1
1.1	Motivação	1
1.2	Objetivo	2
1.3	Metodologia.....	2
1.3.1	Tipo de Pesquisa.....	2
1.3.2	Procedimento Metodológico	3
1.4	Estrutura do Trabalho	4
2	ELICITAÇÃO DE REQUISITOS.....	5
2.1	Considerações Iniciais	5
2.2	Definições.....	5
2.2.1	Requisitos	5
2.2.2	Engenharia de Requisitos	7
2.2.3	Elicitação de Requisitos.....	9
2.3	Importância.....	14
2.4	IEEE 1233, IEEE 830 e ISO-9000-3	16
2.5	Considerações Finais	18
3	TÉCNICAS DE ELICITAÇÃO DE REQUISITOS	19
3.1	Considerações Iniciais	19
3.2	Entrevista.....	19
3.3	Questionários	22
3.4	JAD – <i>Joint Application Development</i>	24
3.5	<i>Brainstorming</i>	27
3.6	Prototipação	29
3.7	Considerações Finais	32
4	SWREQUIREMENT	33
4.1	Considerações Iniciais	33
4.2	Visão Geral do SWRequirement	33
4.3	Fases do Processo de Levantamento e Análise de Requisitos.....	35
4.3.1	Compreensão do Domínio	35
4.3.2	Coleta de Requisitos	35
4.3.3	Classificação	37
4.3.4	Resolução de Conflitos	37
4.3.5	Definição de Prioridade	38
4.3.6	Verificação de Requisitos.....	39
4.4	Considerações Finais	40

5	CONSIDERAÇÕES FINAIS	42
5.1	Conclusões.....	42
5.2	Contribuições.....	43
5.3	Trabalhos Futuros	43
	REFERÊNCIAS BIBLIOGRÁFICAS.....	44

Lista de Figuras

Figura 2-1 – Processo de Engenharia de Requisitos [Fonte: Sommerville (2003)].....	9
Figura 2-2 – Relação das Dimensões das Atividades de Elicitação de Requisitos	11
Figura 2-3 – Levantamento e Análise de Requisitos [Fonte: Sommerville (2003)]	13
Figura 3-1 – Diagrama da Visão Geral do JAD [Fonte: Costa (1994)]	25
Figura 4-1 – Fluxo de Atividades do SWRequirement	41

Lista de Tabelas

Tabela 4-1 – Atividades, Técnicas Indicadas e Artefatos de Software Gerados por SWRequirement	33
--	-----------

1 INTRODUÇÃO

Na atualidade, as empresas desenvolvedoras possuem dificuldade em especificar os requisitos dos sistemas para atender aos seus clientes de forma satisfatória e apresentar aos desenvolvedores o conteúdo necessário para o desenvolvimento. Um processo de elicitação de requisitos bem definido e estruturado pode auxiliar na busca de requisitos detalhados claramente, facilitando a fase de desenvolvimento do software. Uma empresa envolvida no desenvolvimento de sistemas pode ter como possível objetivo garantir que os requisitos serão implementados atendendo aos anseios dos clientes e dos usuários, da melhor maneira possível.

A busca por qualidade tornou-se essencial no processo de desenvolvimento de sistema de software visto as exigências do mercado e a qualificação crescente do setor de informática. Pode-se considerar que um bom levantamento de requisitos é um passo importante para alcançar essa qualidade, pois a especificação do sistema é definida nessa fase do processo de produção de software.

1.1 Motivação

Para realizar o processo de desenvolvimento de software, é necessário fazer alguns levantamentos para identificar e definir as necessidades do projeto que começará. Esses levantamentos estão compreendidos na elicitação de requisitos, que não é uma tarefa simples de ser realizada, mas de suma importância para o sucesso do projeto.

Segundo Sommerville (2003), o levantamento e análise de requisitos compõem um processo difícil, por diversas razões:

- As pessoas envolvidas nesse processo normalmente não sabem o que querem do sistema, elas podem achar difícil demonstrar o que desejam do sistema e fazer pedidos não realistas, por não terem noção do custo de suas solicitações;
- As pessoas envolvidas no sistema expressam naturalmente os requisitos em seus próprios termos e com conhecimento implícito de sua área de atuação. Os engenheiros de requisitos que não têm experiência do domínio do cliente devem compreender esses requisitos;
- Diferentes pessoas têm em mente diferentes requisitos do sistema e podem descrevê-los de maneira distinta. Os engenheiros de requisitos precisam descobrir as fontes de requisitos possíveis e encontrar os pontos comuns e os conflitos;

- Fatores políticos podem influenciar os requisitos do sistema;
- O ambiente econômico e de negócios onde a análise ocorre é dinâmico, podendo levar a mudanças desse ambiente durante o processo de análise. Assim, tem-se como consequência a alteração na importância dos requisitos e o surgimento de novos requisitos solicitados por novas pessoas que foram envolvidas.

Segundo Camacho (2005), o processo de engenharia de requisitos como um todo é muito mais complexo do que apenas uma atividade de transferência de conhecimento entre os clientes e os engenheiros de requisitos. Pois, os clientes dificilmente têm uma idéia clara do que realmente querem e diferentes pessoas dentro da organização normalmente possuem requisitos conflitantes, além de existir na maioria dos casos limitações tecnológicas influenciando os requisitos, entre outros.

Dentro deste contexto, sugerir um processo integrado a técnicas de elicitação de requisitos é relevante, por auxiliar a elicitação de requisitos, pois através deste processo resultados importantes podem ser alcançados, tornando o desenvolvimento do software fácil e coerente com a realidade dos clientes e usuários.

1.2 Objetivo

O objetivo geral deste trabalho é apresentar uma integração de algumas técnicas de elicitação de requisitos ao processo de levantamento e análise de requisitos, proposto por Sommerville (2003), para este processo seja aplicado em qualquer empresa envolvida com desenvolvimento de software.

Como objetivos específicos deste estudo estão:

- Exploração do conhecimento sobre elicitação de requisitos na literatura;
- Análise de técnicas de elicitação de requisitos existentes;
- Busca pelas melhores práticas destas técnicas.

1.3 Metodologia

1.3.1 Tipo de Pesquisa

Para classificar o tipo de pesquisa adotado, deve-se considerar o enfoque abordado, os interesses, as áreas, as condições, os objetivos, as metodologias entre outros. É responsabilidade do pesquisador, escolher qual método melhor se aplica a sua pesquisa.

O método de pesquisa utilizado neste trabalho é a pesquisa exploratória, embasado em pesquisa bibliográfica e documental.

A pesquisa exploratória objetiva delimitar o tema a ser estudado, buscar as informações sobre o assunto, definir os objetivos, determinar as possibilidades e avaliar a viabilidade da realização da pesquisa. A utilização deste tipo de pesquisa possibilita o aprofundamento do conhecimento da realidade, explicando as razões dos fatos (Lakatos, 1983).

A pesquisa bibliográfica foi utilizada neste trabalho com o objetivo de rever a bibliografia existente sobre o tema. Segundo Lakatos (1983), pesquisa bibliográfica é desenvolvida a partir de um material elaborado. As fontes bibliográficas que podem ser: i) livros de leitura corrente (fontes secundárias); ii) livros de referência – informativa (dicionários, enciclopédia e anuários) e remissiva (que remetem a outras fontes-catálogos e repertórios); iii) publicações periódicas (jornais e revistas) e impressos diversos.

Inicialmente, realizou-se a revisão bibliográfica sobre os aspectos da elicitação de requisitos e, em seguida, foram estudadas algumas técnicas de levantamento de requisitos. Foram consultados livros, teses e dissertações disponibilizadas na Internet e na literatura de modo geral. Em seguida, foi elaborada uma estratégia de elicitação de requisitos, baseada nas técnicas estudadas e no processo de levantamento de requisitos proposto.

1.3.2 Procedimento Metodológico

O estudo realizado seguiu o seguinte procedimento metodológico. Inicialmente, foi realizada uma pesquisa em artigos na Internet, dissertações, teses, livros da biblioteca central da UFLA, da biblioteca da PUC Minas campus Coração Eucarístico, da biblioteca do Instituto de Ciências Exatas da UFMG e da biblioteca do Departamento de Ciência da Computação sobre engenharia de software, mais especificamente engenharia de requisitos e técnicas utilizadas para levantar requisitos.

Após a reunião dos materiais citados, foi realizado um estudo aprofundado sobre elicitação de requisitos, assim como as técnicas mais utilizadas e metodologias específicas.

Em seguida, propôs-se uma combinação das técnicas apresentadas focando na suas práticas e observando quais destas práticas melhor se adaptam ao processo de levantamento e análise de requisitos, proposto por Sommerville (2003).

1.4 Estrutura do Trabalho

No capítulo 2, são citados os principais conceitos de elicitação de requisitos, mostrando as dificuldades, a importância e as boas práticas para essa área de conhecimento. Também, são descritos alguns padrões e normas comumente utilizadas.

No capítulo 3, são apresentadas cinco técnicas de elicitação de requisitos, sendo feito o detalhamento destas técnicas e as suas divisões em etapas e realizando uma classificação segundo parâmetros propostos.

No capítulo 4, foi sugerida a integração das técnicas de elicitação de requisitos estudadas ao processo de levantamento de requisitos proposto por Sommerville (2003).

No capítulo 5, são apresentados as conclusões, as contribuições e os possíveis trabalhos futuros.

2 ELICITAÇÃO DE REQUISITOS

2.1 Considerações Iniciais

Neste capítulo, são abordados os conceitos envolvidos na elicitação de requisitos, bem como a sua importância e as normas e os padrões adotados neste processo. Primeiramente, é definido o que é requisito – o objeto principal de estudo. Em seguida, faz-se uma abordagem geral sobre o estado da arte da engenharia de requisitos e explora mais especificamente a elicitação de requisitos, mostrando a definição de alguns autores da área. Posteriormente, é citada a importância do processo de elicitação de requisitos, sendo mostrados os problemas que envolvem essa etapa do desenvolvimento de software. Para finalizar, as principais normas e padrões encontrados são citados para dar sustentação à estratégia de elicitação que será proposta.

2.2 Definições

2.2.1 Requisitos

Os requisitos para um sistema de software estabelecem o que o sistema deve fazer e definem restrições sobre sua operação e sua implementação (Sommerville, 2003).

Kontonya & Sommerville (1998) definem requisitos como a descrição de uma facilidade no nível do usuário; uma propriedade geral do sistema; uma restrição específica sobre o sistema; a especificação de um algoritmo particular, que deve ser empregado em cálculos particulares; enfim, uma restrição aplicável ao processo de desenvolvimento do sistema.

Os requisitos podem ser classificados em funcionais e não funcionais. De acordo com Sommerville (2003), requisitos funcionais são declarações de funções que o sistema deve fornecer ou descrições de como alguns cálculos devem ser realizados. Os requisitos de domínio são requisitos funcionais, provenientes de características do domínio de aplicação. O autor descreve requisitos não funcionais como requisitos do produto, que restringem o sistema a ser desenvolvido, os requisitos de processo que se aplicam ao processo de desenvolvimento e os requisitos externos. Eles, geralmente, relacionam-se às propriedades emergentes do sistema e, portanto, são aplicados ao sistema como um todo.

Os requisitos são as características que definem os critérios de aceitação de um produto. Lopes (2002) define requisitos como alguma coisa que o produto tem que fazer ou

alguma qualidade que precise estar presente. Sob essa perspectiva, têm-se duas categorias de requisitos: i) aqueles responsáveis pela funcionalidade do sistema; e ii) aqueles responsáveis pela qualidade que devem estar presentes, tais como desempenho, integridade, disponibilidade e segurança. Os primeiros são denominados requisitos funcionais e os últimos requisitos não funcionais.

A norma IEEE Std 1233 *IEEE Guide for Developing System Requirements Specifications* (IEEE 1233, 1998), em sua seção de definições, em seu item 3.15, define requisitos como uma condição ou capacidade necessária para o usuário resolver um problema ou atingir um objetivo; uma condição ou uma capacidade que precise ser atendida ou estar presente em um sistema ou em um componente, para satisfazer um contrato, uma norma, uma especificação ou outro documento imposto formalmente; enfim uma representação documentada de uma condição ou capacidade, tal como definidas anteriormente.

Jackson (1995) afirma que requisitos tratam de fenômenos do domínio de aplicação e não da máquina, a máquina pode garantir que estes requisitos sejam satisfeitos, pois a máquina e os requisitos compartilham parte do domínio de 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.

Pode-se perceber que há diversos autores conceituando requisitos, há muitos outros que não foram mencionados, contudo as definições estão muito próximas. Assim, no contexto deste trabalho, requisitos são considerados como as descrições das necessidades as quais o sistema deve atender, tendo como objetivo orientar os desenvolvedores e tornar claro o escopo do sistema para o cliente. Assim, é importante que estejam escritos de forma clara aos interessados. Os requisitos funcionais definem o comportamento do sistema, o que o software fará, especificando as funções que ele ou seus componentes deverão ser capazes de implementar, usando as entradas dos processos para obter as saídas esperadas. Os requisitos não funcionais expressam de que forma o sistema fará aquilo que foi definido pelos requisitos funcionais, apontando as restrições de qualidade, abrangência e operação que o sistema deverá satisfazer.

2.2.2 Engenharia de Requisitos

A Engenharia de Requisitos (ER), dentro da área de Engenharia de Software, trata uma parte fundamental no processo de produção de software, que é a definição do que se quer produzir. Cabe a ER propor métodos, técnicas e ferramentas que facilitem o trabalho de definição do que se quer de um software. A sistematização da etapa de definição de requisitos é necessária, porque a complexidade dos sistemas atuais exige maior atenção ao correto entendimento do problema antes do comprometimento de uma solução. Para que a definição de requisitos seja a mais eficaz possível, cabe aos engenheiros de software entender o ambiente no qual o software irá funcionar e escolher o modelo ou os modelos que melhor representam o ambiente (Bortoli, 2000).

Engenharia de requisitos é um amplo campo de estudo, inserido no contexto da Engenharia de Software, relacionado à identificação, à validação e à documentação das funções e das restrições que precisam ser respeitadas por um software em sua construção e sua operação (Lopes, 2002).

A engenharia de requisitos é o estudo que procura sistematizar o processo de definição de requisitos. Segundo Leite (2001), esta atividade é de extrema importância, pois a complexidade dos sistemas exige 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 gerenciais, organizacionais, econômicos e sociais, e está intimamente associada à qualidade do software.

A engenharia de requisitos estabelece o processo de definição de requisitos como 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 em um contexto previamente definido chamado Universo de Informações (Leite, 2001).

Macaulay (1996) aponta que Engenharia de Requisitos refere-se à análise de um problema utilizando uma perspectiva futura. A ER é uma atividade iterativa, uma vez que informações que afetam os requisitos podem ser identificadas em fases avançadas do processo de desenvolvimento e, portanto, caracteriza-se por sua natureza dinâmica. Assim, é necessário que o processo de engenharia de requisitos seja sistemático, iterativo, repetível e controlável.

Segundo Pressman (2006), a engenharia de requisitos fornece um mecanismo adequado para entender o que o cliente deseja analisar as necessidades, avaliar a exequibilidade, negociar uma solução razoável, especificar a solução de maneira não-ambígua, validar a especificação e administrar os requisitos à medida que eles são transformados em um sistema em operação. O processo de engenharia de requisitos pode ser descrito em cinco passos distintos: i) elicitação de requisitos; ii) análise e negociação de requisitos; iii) especificação de requisitos; iv) modelagem do sistema; v) validação de requisitos; e v) gestão de requisitos.

Para Sommerville (2003), o processo de engenharia de requisitos inclui:

- Estudo de viabilidade. Nesta atividade, é realizado um estudo breve e direcionado, sobre as contribuições do sistema para a empresa, a tecnologia utilizada e a possível integração com outros sistemas, que tem como entrada uma descrição geral do sistema e de como ele será utilizado dentro de uma organização. Sua saída será um relatório que decidirá se o sistema proposto é viável do ponto de vista comercial e se poderá ser desenvolvido;
- Levantamento e análise de requisitos. Nesta atividade, os membros da equipe técnica de desenvolvimento de software trabalham com o cliente e os usuários finais do sistema para descobrir mais informações sobre o domínio da aplicação, quais serviços o sistema deve fornecer, o desempenho exigido do sistema, as restrições de hardware e assim por diante;
- Especificação de requisitos. Nesta atividade, são traduzidas as informações coletadas durante a atividade de análise em um documento que define um conjunto de requisitos;
- Validação de requisitos. Nesta atividade, são mostrados os requisitos que definem o sistema desejado pelo cliente. Durante esse processo, inevitavelmente, são descobertos erros na documentação de requisitos. Assim, estes requisitos devem ser modificados a fim de corrigir esses problemas.

A

Figura 2-1 apresenta o processo de engenharia de requisitos segundo Sommerville (2003). Esse processo começa com a fase de estudo da viabilidade gerando um relatório de viabilidade (1), este artefato determina se a próxima fase será realizada. Caso o relatório indique que o sistema é viável, o levantamento e análise de requisitos devem ser feitos (2), gerando o modelo do sistema (3). Com as informações levantadas na etapa anterior, inicia-

se a especificação de requisitos (4), construindo o documento com os requisitos do usuário e do sistema (5). Caso haja necessidade, volta-se para a fase anterior para complementar os requisitos (6). Esse documento com os requisitos é validado, corrigido na fase de validação de requisitos (7). Havendo necessidade, volta-se para a fase de elicitação de requisitos (8). Ao fim do processo, uma documentação de requisitos é produzida, ou seja, tem-se a especificação do sistema (9).

Os requisitos usualmente são apresentados em dois níveis de detalhes. Os usuários e os clientes necessitam de uma especificação de alto nível dos requisitos; os desenvolvedores do sistema precisam de uma especificação mais detalhada. A análise de requisitos continua durante a definição e a especificação de requisitos. Portanto, as atividades de análise, de definição e de especificação são intercaladas.

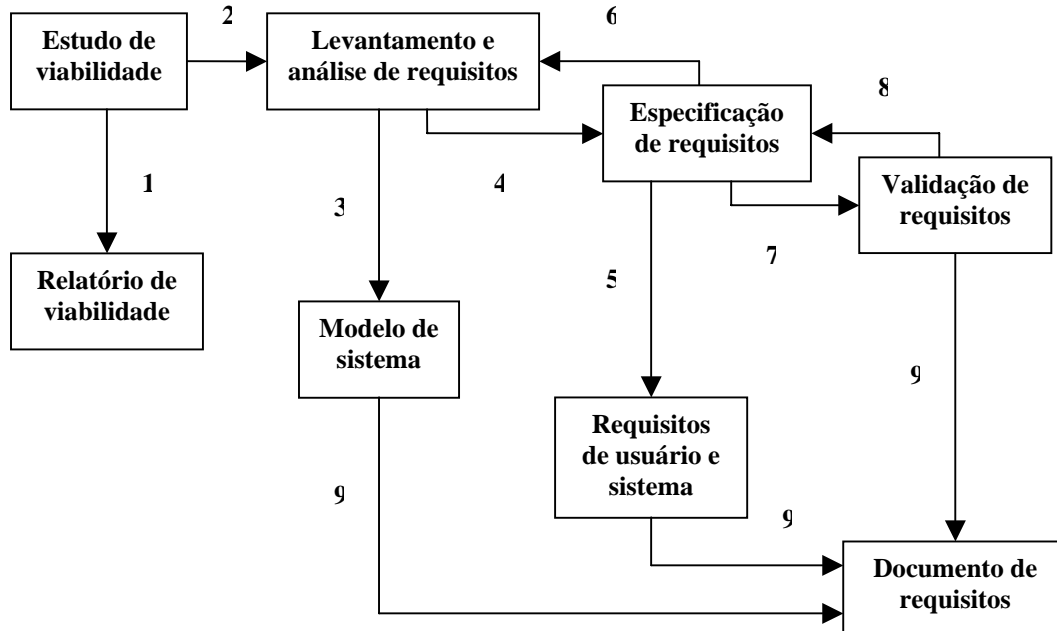


Figura 2-1 – Processo de Engenharia de Requisitos [Fonte: Sommerville (2003)]

A engenharia de requisitos propõe um conjunto de métodos, técnicas e ferramentas para dar apoio ao processo de definição de requisitos. Em suma, o conjunto de técnicas de levantamento, documentação e análise de requisitos forma a engenharia de requisitos.

2.2.3 Elicitação de Requisitos

Dentro da engenharia de requisitos, cabe à elicitação de requisitos 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, trivial para quem tem o conhecimento, mas não é para quem o procura, sendo dificilmente lembrado e, portanto, não transmitido (Goguen, 1994).

Leite (2001) chama de elicitação de requisitos as atividades que envolvem a descoberta de requisitos de um sistema, como a identificação das fontes de informação, a coleta de fatos e a comunicação.

Para Camacho (2005), para tentar encontrar problemas a serem solucionados, serviços que o sistema deve oferecer requisitos de performance, restrições de hardware e outros, os engenheiros e os desenvolvedores de software devem trabalhar diretamente com os clientes e os usuários finais. Para conseguir obter informações dos clientes e dos usuários, não basta questioná-los sobre o que eles desejam, é necessário fazer uma análise muito mais detalhada da organização, do domínio da aplicação e dos processos de negócios onde o sistema será usado.

A elicitação de requisitos é designada por esse nome por conta do termo em inglês *elicit*, significando obter informações ou reações, esta fase do processo relaciona-se à obtenção dos requisitos do sistema. Assim, analistas e engenheiros de software trabalham com os clientes e os usuários finais para descobrir o problema a ser resolvido, os serviços do sistema, o desempenho necessário, as restrições de hardware e outras informações (Kontonya & Sommerville, 1998).

Kontonya & Sommerville (1998) argumentam a existência de quatro dimensões na atividade de elicitação de requisitos:

- Entendimento do domínio de aplicação. Entendimento geral da área na qual o sistema será aplicado;
- Entendimento do problema. Entendimento dos detalhes do problema específico a ser resolvido com o auxílio do sistema a ser desenvolvido;
- Entendimento do negócio. Entendimento da contribuição do sistema para que sejam atingidos os objetivos gerais da organização;

- Entendimento das necessidades e das restrições dos *stakeholders*¹. Entendimento detalhado das necessidades de apoio a serem promovidas pelo sistema à realização do trabalho e aos interesses de cada um dos *stakeholders*, dos processos de trabalho a serem apoiados pelo sistema e do papel de eventuais sistemas existentes na execução dos processos de trabalho.

A Figura 2-2 mostra a relação das dimensões da atividade de elicitação, essa relação é cíclica, pois, à medida que cada dimensão é conquistada as outras são fortalecidas e o seu entendimento é facilitado. A busca pelo entendimento do domínio de aplicação é ponto inicial desse ciclo, proporcionando o início do entendimento do problema, depois desse entendimento o conhecimento do domínio da aplicação é intensificado. Nesse estágio de compreensão, o entendimento do negócio é realizado, possibilitando que os outros entendimentos sejam melhorados. Para fechar o ciclo, o entendimento das necessidades e das restrições dos *stakeholders* é alcançado.

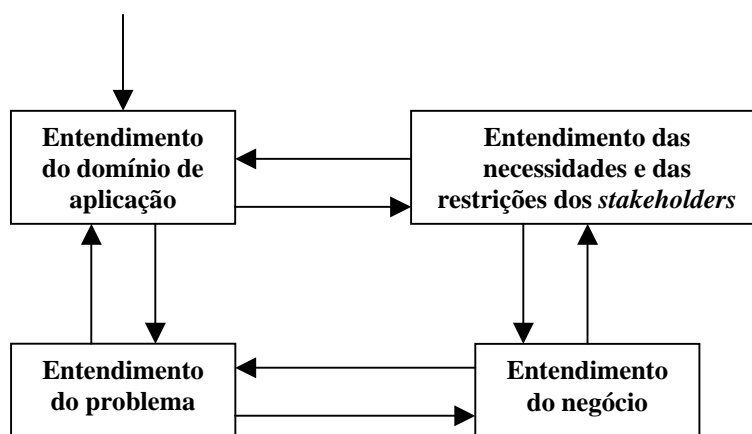


Figura 2-2 – Relação das Dimensões das Atividades de Elicitação de Requisitos

Pressman (2006) sugere um conjunto de diretrizes detalhadas para a elicitação de requisitos, a saber:

- Avaliar a viabilidade técnica e de negócios do sistema proposto;
- Identificar as pessoas que vão ajudar a especificar os requisitos e a compreender seu viés organizacional;
- Definir o ambiente técnico no qual o sistema ou o produto vai ser colocado;
- Identificar restrições de domínio que limitam a funcionalidade ou o desempenho do sistema ou do produto a ser construído;

¹ Termo usado para referir a qualquer pessoa que terá alguma influência direta ou indireta sobre os requisitos do sistema (Sommerville, 2003).

- Definir um ou mais métodos de elicitação de requisitos;
- Solicitar a participação de pessoas de modo que os requisitos sejam definidos de diferentes pontos de vista; certificar-se de identificar a razão de cada requisito registrado;
- Identificar requisitos ambíguos para candidatos à prototipagem;
- Criar cenários de uso para ajudar clientes/usuários a melhor identificar requisitos-chave.

Os produtos de trabalho produzidos como conseqüências das atividades de elicitação de requisitos vão variar de acordo com o tamanho do sistema ou do produto a ser construído. Pressman (2006) afirma que, para a maioria dos sistemas, os produtos de trabalho incluem:

- Uma declaração da necessidade e da viabilidade;
- Uma declaração delimitada do escopo do sistema ou do produto;
- Uma lista de clientes, usuários e outros interessados que participaram da atividade de elicitação de requisitos;
- Uma descrição do ambiente técnico do sistema;
- Uma lista de requisitos e as restrições do domínio que se aplicam a cada um;
- Um conjunto de cenários de utilização que fornece entendimento do uso do sistema ou do produto em diferentes condições de operação;
- Quaisquer protótipos desenvolvidos para melhor definir os requisitos.

Cada um desses produtos de trabalho é revisado pelo pessoal que participou da elicitação de requisitos (Pressman, 2006).

Ao escolher a técnica de elicitação, são considerados as fontes dos requisitos, a disponibilidade das pessoas, os tipos de requisitos a serem elicitados entre outros fatores. A escolha da técnica apropriada para elicitar requisitos depende do tempo e dos recursos disponíveis e do tipo de informação necessária.

Uma vez a necessidade identificada e um plano inicial desenvolvido, a equipe de requisitos tenta identificar quais propriedades o sistema deveria ter para satisfazer tal necessidade. As propriedades relevantes vão desde requisitos de alto nível a tempo de resposta, segurança, portabilidade, confiança e manutenibilidade.

Segundo Leite (2001), o processo de definir requisitos é incompleto, tendo em vista a grande complexidade do mundo real. No entanto, é indicado procurar obter requisitos mais completos possíveis.

Pode-se compreender porque os clientes comumente não sabem precisamente o que querem, pois, entre outras causas, eles podem necessitar de ver modelos, explorar alternativas e prever novas possibilidades, em conjunto com os engenheiros de requisitos. Essas causas estão imersas em fatores legais, sociais, políticos, financeiros e psicológicos (Goguen, 1994).

Para Sommerville (2003), um modelo genérico do processo de levantamento e análise de requisitos é mostrado na Figura 2-3.

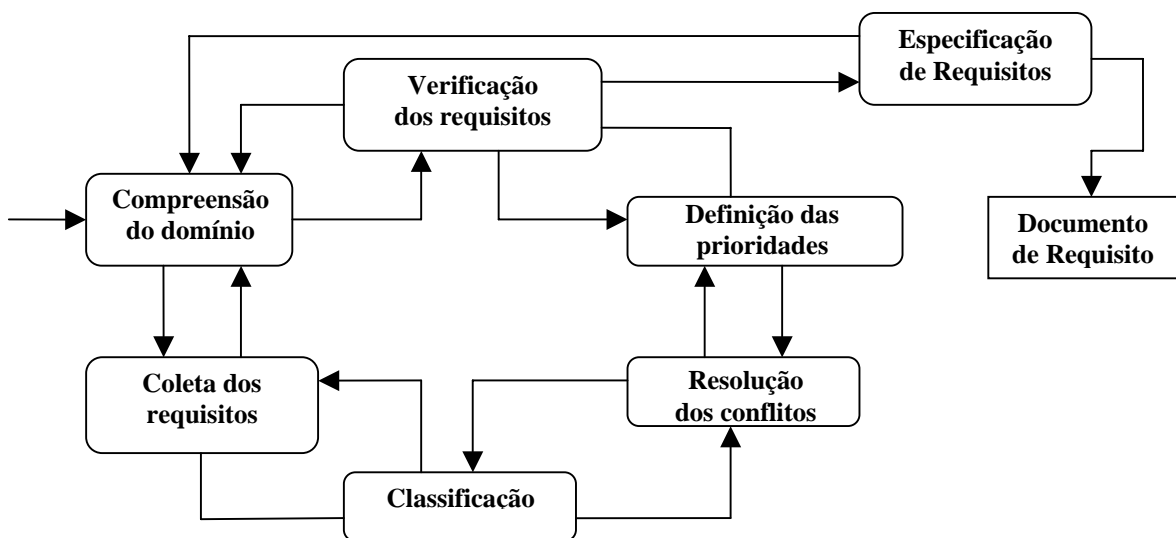


Figura 2-3 – Levantamento e Análise de Requisitos [Fonte: Sommerville (2003)]

Cada organização terá sua própria versão ou uma versão mais definida desse modelo geral, dependendo de fatores locais, como a perícia da equipe, o tipo de sistema em desenvolvimento e os padrões utilizados. As atividades do processo são:

- Compreensão do domínio. A atividade desenvolvida pelo analista para conhecer e compreender o domínio da aplicação, ou seja, entender o contexto que o sistema será empregado. Por exemplo, se o sistema a ser desenvolvido for para uma rede de supermercados, o analista deve descobrir como funcionam os supermercados;
- Coleta de requisitos. A atividade onde são definidos os requisitos do sistema, através da interação com os *stakeholders* do sistema. Cabe aos responsáveis por essa atividade descobrir os requisitos que devem ter no sistema para satisfazer as necessidades dos usuários e dos clientes. A coleta de requisitos proporciona aos analistas e aos desenvolvedores maior entendimento da aplicação do sistema;

- Classificação dos requisitos. A atividade de agrupamento dos requisitos do sistema, considerando suas características e sua funcionalidade. O conjunto de requisitos não estruturado e coletado é transformado em subconjuntos de requisitos organizados com coerência;
- Resolução de conflitos. A atividade realizada para encontrar e solucionar conflitos, para obter requisitos coerentes entre si. Devido aos diferentes interesses dos usuários envolvidos no processo, estes conflitos podem aparecer;
- Definição de prioridade. A atividade de classificação de requisitos considerados mais importantes, pois, durante o processo de construção de um sistema, não é viável atender a todos ao mesmo tempo. Essa classificação é definida para evidenciar os requisitos a serem finalizados primeiro e os requisitos mais representativos para o sistema. Portanto, deve-se determinar o grau de prioridade de cada requisito, consultando os *stakeholders*;
- Verificação de requisitos. A atividade certificadora de que os requisitos estão completos e consistentes e, principalmente, se atendem as necessidades dos usuários e transmitem corretamente os anseios do cliente em relação ao sistema. Essa atividade valida os requisitos coletados e detalhados nas etapas anteriores.

A Figura 2-3 mostra que o levantamento e análise de requisitos são um processo iterativo, com *feedback* contínuo de cada atividade para as outras. O ciclo começa com a compreensão do domínio e termina com a verificação dos requisitos. A compreensão dos requisitos pelo analista melhora a cada fase do ciclo. O processo de levantamento de requisitos, quando bem aplicado, resultará em um documento de requisitos detalhado, claro e consistente, que auxiliará na documentação e no desenvolvimento do sistema.

O processo de levantamento de requisitos apresentado por Sommerville (2003) foi o foco deste trabalho. Foram identificadas atividades de técnicas de elicitação de requisitos que mais se adequaram às propostas das fases deste processo.

2.3 Importância

A tarefa de análise é um processo de descoberta, refinamento, modelagem e especificação. A análise e a especificação podem parecer tarefas fáceis, mas o conteúdo de comunicação é elevado e complexo. As chances de interpretações errôneas e informações falsas são grandes, levando a uma especificação incorreta e causando insatisfação do

cliente, re-trabalho, perdas financeiras entre outros problemas (Pressman, 2006). Com intuito de minimizar esses problemas e alcançar uma análise que condiz com as reais necessidades e anseios do cliente, técnicas de elicitação de requisitos devem ser conhecidas.

Segundo Curtis *et al* (1988), os principais problemas na elicitação de requisitos são: i) o conhecimento parcial ou incompleto do domínio da aplicação; ii) requisitos oscilantes e conflitantes; e iii) falhas na coordenação e na comunicação de atividades. Ao analisar esses problemas, verifica-se a importância da elicitação de requisitos, ou seja, existe ganhos com o comprometimento em fazer um levantamento de requisitos preciso adotando padrões, normas e técnicas apropriadas.

O conhecimento parcial ou incompleto do domínio da aplicação é problema constante devido à falta de integração entre a própria equipe sobre o conhecimento individual do domínio do negócio ou pelo fato dos usuários apresentarem resistência em compartilhar com o engenheiro de requisitos as informações necessárias. Isso acontece, pois, às vezes, associa-se essa informação ao poder dentro da organização ou à estabilidade de seu emprego.

Requisitos oscilantes e conflitantes surgem devido aos *stakeholders* terem pontos de vista diferentes e, como o levantamento de requisitos geralmente é feito por meio de entrevistas individuais, esses problemas se agravam, ou seja, os requisitos definidos por um usuário entrevistado anteriormente podem gerar conflitos com requisitos definidos por outro posteriormente. Contudo, métodos integrativos devem ser adotados para conseguir uma solução para todos. Outro motivo pode ser a falta de uma boa definição dos objetivos do sistema (Camacho, 2005).

A falha de coordenação e de comunicação das atividades é gerada, porque o engenheiro de software pode não utilizar os documentos formais e confiar em rascunhos ou mesmo na própria memória. Essas falhas também ocorrem porque informações importantes são ocultadas por fatores gerais, sendo desconhecidas pelo engenheiro de requisitos atingindo diretamente a especificação dos requisitos. Outro motivo pode ser o fato do engenheiro de requisitos optarem por trabalhar sozinho, especificando um software que não atende ao usuário, mas a ele mesmo (Camacho, 2005).

Observando os problemas citados, consegue-se perceber que a elicitação de requisitos é crucial no processo de desenvolvimento de software. Sua qualidade pode

comprometer o restante do desenvolvimento do sistema, acarretando graves perdas de custo e de trabalho.

2.4 IEEE 1233, IEEE 830 e ISO-9000-3

A IEEE Std 1233 *IEEE Guide for Developing System Requirements Specifications* (IEEE 1233, 1998) é um guia para desenvolvimento de especificação de requisitos de sistemas. Este guia fornece a orientação para o desenvolvimento de um conjunto de requisitos que, quando realizados, satisfarão a uma necessidade identificada. Neste guia, o conjunto de requisitos será chamado de especificação de requisitos do sistema (ERS).

Desenvolver uma ERS inclui a identificação, a organização, a apresentação e a modificação dos requisitos. Este guia conduz às condições para incorporar conceitos organizacionais e restrições do projeto e projetar requisitos operacionais na especificação. Este guia indica também as características e as qualidades necessárias dos requisitos individuais e ao conjunto dos requisitos (IEEE 1233, 1998).

A norma IEEE Std 1233 é um guia para a construção de uma especificação de requisitos que pode ser utilizada juntamente com a norma IEEE Std 830, pois essa última define a estrutura de um documento de requisitos gerado na especificação de requisitos.

A norma IEEE Std 830 *IEEE Recommended Practice For Software Requirements Specifications* (IEEE 830, 1998) define a estrutura básica do documento de requisitos, especificando como deve ser organizado e fornece um modelo para sua confecção. Contudo, o conteúdo do documento depende de diversos fatores como a natureza do sistema que está sendo especificado, o nível de detalhamento dos requisitos, as práticas organizacionais e as restrições orçamentárias e de tempo aplicáveis ao processo de Engenharia de Software.

Segundo a norma IEEE Std 830 (IEEE 830, 1998), para clientes, fornecedores e outros indivíduos, uma boa especificação de requisitos deve fornecer diversos benefícios específicos, tais como:

- Estabelecer um acordo entre os clientes e os fornecedores de software;
- Reduzir o esforço do desenvolvimento;
- Fornecer uma estimativa de custos e cronograma;

- Produzir uma *baseline*² para validação e verificação;
- Facilitar transferência;
- Servir como base para a otimização do sistema.

Esta norma descreve boas práticas recomendadas para a especificação de requisitos do software. Ela é baseada em um modelo no qual o resultado é um processo de especificação de software que tem como saída um documento de requisitos não ambíguo e completo.

Para garantir a qualidade da especificação de requisitos e, como consequência do documento de requisitos, validarem os requisitos desse documento, o padrão ISO-9000-3 faz algumas considerações importantes.

O padrão ISO-9000-3 (ISO-9000-3, 2001) representa uma adaptação do ISO-9001 para o desenvolvimento, o fornecimento e a manutenção de software, levando em conta certas características que são específicas deste ramo industrial. No item 5.3 dessa norma, é descrita a elicitação de requisitos de software e algumas considerações, a saber:

- A fim de continuar com o desenvolvimento do software, o fornecedor deve ter um conjunto completo e sem ambigüidades dos requisitos funcionais. Além disso, os requisitos devem incluir aspectos necessários à satisfação das necessidades do comprador, aspectos estes como desempenho, segurança contra riscos, privacidade e confiabilidade. Tais requisitos devem ser estabelecidos de maneira precisa suficiente para permitir a validação durante a aceitação do produto;
- A especificação dos requisitos do comprador deve registrar estes requisitos. Em alguns casos, esta especificação é fornecida pelo comprador. Caso contrário, o fornecedor deve desenvolver estes requisitos em estreita cooperação com o comprador e obter a aprovação da especificação, antes de passar ao estágio de desenvolvimento. A especificação dos requisitos do comprador deve estar sujeita ao controle de documentação e à gestão de configuração, como parte da documentação de desenvolvimento;
- As interfaces entre o produto de software ou de hardware devem ser plenamente especificadas, diretamente ou por meio de referência, na especificação dos requisitos do comprador;

² Uma espécie de referencial que é utilizado no processo de mudança (Leite, 2001).

- Durante o desenvolvimento da especificação dos requisitos do comprador, recomenda-se dar atenção às seguintes questões:
 - ✓ Designação de pessoas (de ambas as partes) responsáveis pelo estabelecimento da especificação dos requisitos do comprador;
 - ✓ Métodos para obter acordo quanto aos requisitos e à aprovação das alterações;
 - ✓ Esforços para evitar mal-entendido, tais como definições de termos e explicação de fundamentos dos requisitos;
 - ✓ Registro e análise crítica dos resultados de discussões por ambas as partes.

2.5 Considerações Finais

Neste capítulo, pôde-se observar a importância da elicitação de requisitos, devido aos problemas que surgem caso esse processo não seja bem desenvolvido. Foram citados vários autores que mostram como realizar o levantamento e análise de requisitos, utilizando processos definidos e concisos. São obtidas neste capítulo, as saídas possíveis para um processo de elicitação bem sucedido, revendo as normas e os padrões e evitando os erros comuns à elicitação de requisitos.

3 TÉCNICAS DE ELICITAÇÃO DE REQUISITOS

3.1 Considerações Iniciais

No presente capítulo, apresenta-se uma visão detalhada das técnicas de elicitação de requisitos. São consideradas cinco técnicas: i) entrevista; ii) questionário; iii) JAD; iv) *brainstorming*; e v) prototipação. A entrevista é uma técnica conhecida e muito útil, questionários são eficientes por permitirem ao respondente a liberdade da resposta. JAD é uma técnica bem detalhada e estruturada, embasada em reuniões com os envolvidos no processo de desenvolvimento do software. O *brainstorming* sustenta a teoria de liberdade de idéias. A prototipação incentiva a construção de um protótipo que simula o sistema para estimular uma definição precisa dos requisitos.

3.2 Entrevista

A entrevista é uma das técnicas mais utilizadas para detalhar e incorporar os requisitos do cliente. A entrevista é uma técnica simples e direta que pode ser facilmente usada em diversas situações no processo de engenharia de software (Carvalho, 2003).

O uso de entrevistas é imprescindível na etapa de elicitação de requisitos por ser umas das formas de comunicação mais naturais que existe. Um dos aspectos fundamentais deste tipo de técnica é garantir que as idéias e as predisposições naturais do entrevistador não interfiram com uma livre troca de informação. Isto pode ser algo difícil de conseguir, uma vez que cada pessoa é influenciada por um conjunto de vivências pessoais, que podem dificultar a compreensão de idéias vindas de outras pessoas. Essa técnica estruturada pode ser aprendida e os desenvolvedores podem conquistar a aptidão com treinamento e prática (Carvalho, 2003).

Orienta-se a começar com elaboração de perguntas livres de contexto, concentradas no cliente, buscando as suas metas e os benefícios posteriores. Com o uso dessa técnica, objetiva-se obter conhecimentos sobre um domínio através de perguntas feitas aos usuários especialistas deste domínio por um ou mais entrevistadores; o uso da técnica permite também que os desenvolvedores entendam os processos atuais da organização, percebam o que está faltando no sistema existente e as expectativas dos usuários do novo sistema. As entrevistas são úteis quando usuários possuem muitos conhecimentos subjetivos e estão

dispostos a serem entrevistados. Caso o entrevistador seja experiente, é possível alcançar resultados mais eficientes.

O engenheiro ou analista de requisitos discute o sistema com diferentes usuários; a partir dessa discussão, elabora-se um entendimento de seus requisitos. Há basicamente, segundo Kontonya & Sommerville (1998), dois tipos de entrevista:

- Entrevistas fechadas, onde o engenheiro de requisitos prepara as perguntas dentro de um conjunto pré-definido de questões;
- Entrevistas abertas, onde não há agenda pré-definida e o engenheiro de requisitos discute de modo aberto o que os usuários querem do sistema.

Uma entrevista é uma tentativa sistemática de adquirir informação de uma pessoa. Entrevistar é uma habilidade importante para os analistas de sistemas porque o sucesso da técnica depende de uma habilidade para identificar: i) fluxos de trabalho; ii) fatores que influenciam as operações do sistema; e iii) elementos que compõem os sistemas (Kremer, 1998).

O processo de entrevista pode ser dividido em seis etapas segundo Kremer (1998):

- Preparar-se para a entrevista: ao prepara-se para a entrevista, antes de sua realização, o analista deve ter um bom conhecimento sobre a organização, o seu ambiente industrial, o escopo do projeto e os seus objetivos. Para isso, ele deve analisar os relatórios da organização, a documentação disponível de planejamento, os objetivos dos setores, os manuais existentes de procedimentos da empresa e os documentos dos sistemas usados e relacionados com o que será desenvolvido. Os analistas devem compreender termos comuns usados na empresa e ser familiar às regras de negócio da empresa;
- Planejar e programar a entrevista: a segunda etapa, planejamento e programação da entrevista, consiste em elaborar um plano de aplicação das entrevistas, de maneira que o processo comece de cima para baixo, ou seja, os diretores e os chefes de departamento são entrevistados antes que seus funcionários, pois esses conhecem um pouco as necessidades dos seus colaboradores. Deve-se preparar uma lista dos tópicos e das perguntas a serem abordadas, para ajudar a assegurar que pontos importantes não sejam esquecidos e a entrevista siga uma seqüência lógica. Os responsáveis pela aplicação das entrevistas devem explicar a sua finalidade, as áreas gerais que serão abordadas e a quantidade aproximada de tempo necessária;

- Iniciar a entrevista: no início da entrevista, o entrevistador deve apresentar-se, mostrar a finalidade da entrevista e abordar alguns interesses dos entrevistados. O entrevistador deve ainda explicar que serão anotadas algumas partes da entrevista e, depois de organizadas, as anotações serão divulgadas para os entrevistados. Geralmente, os entrevistados ficam com receio de que o analista procurar falhas na maneira como eles trabalham. Para amenizar essa sensação, é interessante falar dos processos que são conhecidos pelo entrevistado;
- Conduzir a entrevista: as melhores entrevistas são aquelas em que o entrevistado é quem mais fala; por isso, o analista terá que buscar uma forma de aproximar do entrevistado para que ele possa se expressar melhor. Ao conduzir a entrevista, o entrevistador deve-se empenhar para isso. Na utilização dessa técnica, tem-se que evitar perguntas restritivas, pois os resultados desse tipo de abordagem são respostas breves e aumento do distanciamento entre o entrevistado e o entrevistador. Boas práticas para obter bons resultados são: i) fazer perguntas abertas que incentive o entrevistado a falar; ii) usar palavras e frases apropriadas; iii) dar sugestões de aceitação das respostas; iv) repetir as respostas do entrevistado para verificar se houve completa compreensão;
- Fechar a entrevista: a etapa de fechar a entrevista envolve momentaneamente sumarizar os pontos que foram discutidos, destacando os fatos importantes e a sua compreensão. Isto deixa o entrevistado saber que o entrevistador o ouviu com cuidado durante a entrevista e fornece uma oportunidade de esclarecimento dos equívocos. Durante o fechamento, assim como durante a entrevista inteira, o analista deve adotar uma postura objetiva e evitar comentários, observações ou conclusões pessoais;
- Manter contato para maiores esclarecimentos: Finalmente, deve-se agradecer ao entrevistado pelo tempo dedicado e, para manter o contato, perguntar se uma entrevista mais breve pode ser programada em outro dia, caso seja necessário.

A entrevista pode apresentar algumas vantagens como: i) maior abertura para os entrevistados; ii) exposição verbal das idéias; iii) conhecimento mais profundo para o analista; iv) acesso a informações armazenadas informalmente; v) melhor julgamento sobre os envolvidos; e vi) incentivo ao raciocínio do entrevistado e do entrevistador. As desvantagens pode ser dificuldade para registrar a entrevista e surgimento de palpites e adivinhações.

3.3 Questionários

De acordo com Bastos Junior (2005), os questionários são um dos possíveis meios utilizados para a aquisição de dados na elicitação de requisitos. Um questionário é um documento usado para guiar uma ou mais pessoas a responder a uma ou mais perguntas. A elaboração de um questionário é um processo bem mais complexo do que possa aparentar devido às informações buscadas serem implícitas. Por isso, aplicar tempo e esforço adequados para a construção do questionário é uma necessidade e um fator de diferenciação favorável.

Os questionários são utilizados em diversas áreas como Ciências Sociais e *Marketing*, sendo comum aplicá-los em pesquisas de opinião, censo e pesquisas eleitorais. Questionários também são utilizados como uma técnica auxiliar a outras técnicas de elicitação como entrevistas e reuniões, técnicas usadas com frequência na elicitação de requisitos (Bastos Junior, 2005).

Questionários podem ser elaborados e aplicados de diversas maneiras, de acordo com as necessidades que ele deve atender. Para Parasuraman (1991), as principais formas são:

- Auto-resposta via *e-mail* – torna fácil o envio do questionário a várias pessoas ao mesmo tempo, não requerendo muito tempo e possibilitando ao respondente enviar a resposta quando lhe for mais conveniente. Porém, desta maneira, tem-se uma taxa baixa de retorno e uma tendência a respostas curtas, inviabilizando perguntas que exijam respostas detalhadas;
- Auto-resposta em grupo – um grupo de pessoas é reunido seguindo alguns critérios de conveniência e as perguntas são feitas simultaneamente aos participantes, cada um responde individualmente ao questionário. Caso surja alguma dúvida de entendimento ou da finalidade, o participante poderá saná-la neste momento;
- Auto-resposta porta a porta – o investigador vai ao encontro do respondente, entrega e explica o questionário. Depois, volta para recebê-lo ou pede ao respondente para enviá-lo via correio. Este tipo de aplicação não é muito utilizado.

Parasuraman (1991), também citou a entrevista como um das possíveis formas de aplicar questionário, mas não será abordado neste tópico, pois esta técnica foi detalhada anteriormente.

De acordo com *Goode & Hatt* (1960), para construir um questionário válido, deve-se adotar um conjunto de procedimentos metodológicos e técnicos, que vão desde a

formulação do problema, a uma aplicação em uma pequena amostra, possibilitando a aquisição de dados empíricos que podem melhorar o questionário. Os principais procedimentos apontados pelo autor são:

- A formulação do problema deve ser feita a partir de uma questão inicial que servirá de base para elaboração do questionário;
- Essa questão deve ser clara, precisa, realista e pertinente;
- A explicação do objetivo da pesquisa;
- A formulação de hipóteses para que haja a interação entre a teoria e a verificação baseada na experiência, formando um guia de trabalho;
- Uma orientação para a aquisição de dados.

Os tipos de perguntas são importantes na coleta de informações devido a eles afetarem no tipo e na qualidade das informações afirma Mattar (2001). Os tipos mais usados em questionários são perguntas de resposta aberta e perguntas de resposta fechada. As perguntas de resposta aberta não oferecem alternativas para o respondente, este deve formular uma resposta com suas próprias palavras deixando claros o objetivo da resposta e a justificativa, mas o respondente tem total liberdade para responder. As perguntas de resposta fechada limitam o respondente a escolher umas das opções colocadas que retrata melhor sua opinião. Segundo Bastos Junior (2005), para decidir qual o tipo de perguntas é melhor para o questionário, é importante saber o modo como se planeja usar a informação obtida, pois a forma que é estruturada as questões determina a unidade de medida pelas quais as respostas serão classificadas.

Normalmente, a aplicação do questionário consiste em distribuir questões pré-definidas para uma amostragem significativa e representativa das partes interessadas e os resultados são avaliados. Questionários são úteis quando a quantidade de clientes e de usuários é extremamente grande. Uma vez que as questões podem ser pré-determinadas, o questionário é mais eficiente na avaliação de tendências de opiniões a respeito de requisitos específicos e bem definidos (Bastos Junior, 2005).

Camacho (2005) aponta como as principais vantagens da utilização de questionário: a padronização das perguntas e a possibilidade de tratamento estatístico das respostas. Como desvantagens, o autor aponta a limitação do universo de respostas e a pouca interação, visto a impessoalidade da técnica.

3.4 JAD – *Joint Application Development*

JAD, desenvolvido pela IBM em 1977, propõe a criação de sessões de trabalho estruturadas, por meio de dinâmicas de grupo e recursos visuais, em que usuários e analistas trabalham juntos para projetar um sistema, desde os requisitos básicos ao *layout* de telas e de relatórios, prevalecendo a cooperação e o entendimento. A equipe de desenvolvimento auxilia os usuários a elaborar os problemas e analisar prováveis soluções, comprometendo-os para que se sintam participantes do desenvolvimento (Loch *et al*, 2005).

Segundo Batista (2003), JAD baseia-se em quatro princípios básicos:

- Dinâmica de grupo, com a utilização de sessões de grupo facilitadas para aumentar a capacidade dos indivíduos;
- Uso de técnicas audiovisuais para aumentar a comunicação e o entendimento;
- Manutenção do processo organizado e racional;
- Utilização de documentação padrão, preenchida e assinada pelos participantes de uma sessão.

Para Reis (2000), JAD é uma técnica de reunião que têm como finalidade acelerar o projeto de desenvolvimento e de manutenção de software. Sua aplicação torna possível a concepção de sistemas mais eficientes em um tempo menor, tendo como um de seus benefícios a adesão que pode ter a vários métodos de desenvolvimento de software atualmente empregados, como análise estruturada, análise essencial e orientada a objetos.

A técnica pode ser usada tanto para elicitar quanto nas fases iniciais da implementação. Ela ajuda a identificar os assuntos que podem necessitar de rastreamento e fornece uma perspectiva multifacetada dos requisitos. Sessões JAD permitem aos analistas coletar simultânea e eficientemente uma grande quantidade de requisitos do sistema junto a uma gama de usuários-chave. Essas sessões são úteis por considerar necessidades específicas dos usuários. JAD também pode ser usada em conjunto com outras técnicas de elicitação, por exemplo, a prototipação. À medida que os requisitos são obtidos nas sessões, pode-se construir um protótipo que demonstre alguma funcionalidade destes requisitos.

JAD se ajusta à realidade da empresa. Quando aplicada corretamente, tende a aproximar os participantes e torna-se uma ferramenta para a conversão do conhecimento

implícito em conhecimento explícito. O comprometimento e o consenso entre os participantes criam um ambiente de respeito recíproco e de responsabilidade pelo sucesso do sistema (Loch *et al*, 2005).

Costa (1994) define JAD como método destinado a extrair informações de alta qualidade dos usuários em curto espaço de tempo, através de reuniões estruturadas que buscam decisões por consenso. Para o autor, o processo pode ser representado pela Figura 3-1.

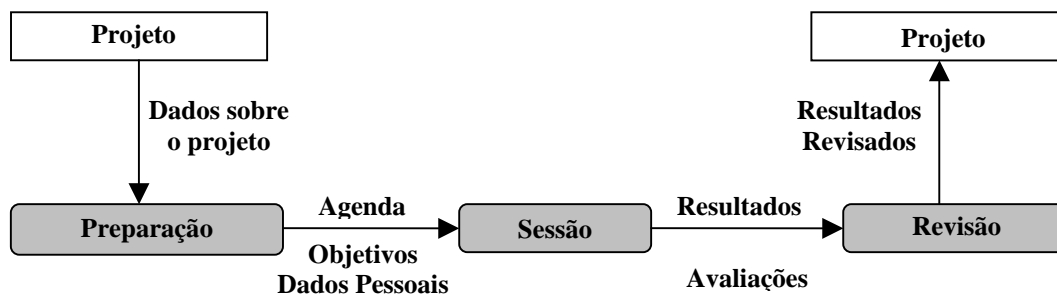


Figura 3-1 – Diagrama da Visão Geral do JAD [Fonte: Costa (1994)]

Para Costa (1994), o método é dividido em três fases preparação, sessão e revisão. JAD interage com o projeto recebendo dados necessários à preparação da sessão de trabalho e desenvolvendo, no final, os respectivos resultados. Esses resultados são partes dos recursos usados para o desenvolvimento. Nos próximos parágrafos, a descrição de cada fase segundo Costa (1994).

A etapa Preparação o objetivo de estabelecer as condições básicas para que a sessão de trabalho seja produtiva e completa. Sendo assim, devem ser escolhidos os participantes apropriados, uma agenda deve ser elaborada adequadamente e o material necessário deve ser providenciado. Para alcançar esses objetivos, a fase pode ser dividida em sub-etapas:

- Examinar em conjunto com o gerente do projeto, se é adequada à utilização de JAD: no caso, são analisados os riscos relativo ao tamanho do projeto, o domínio da metodologia e o espírito da equipe;
- Planejar as sessões: são definidas quantas sessões JAD serão realizadas, que tipo e quando;
- Elaborar a perspectiva gerencial: consiste em obter uma visão clara do sistema como um todo, com finalidade, escopo, objetivos e restrições;
- Familiarizar-se com áreas de negócios sob estudo: habilitar-se a fazer o papel de interface entre os usuários e os analistas;

- Preparar a agenda da sessão: indicar os passos a serem seguidos e mostrar onde o grupo se encontrará, para onde estão indo e o que deve ser alcançado;
- Preparar os participantes: apresentar a abordagem da sessão, identificar as informações que devem ser previamente providenciadas e distribuir a tarefa de produzir essas informações aos participantes;
- Preparar a agenda detalhada: roteiro que o facilitador usa durante a sessão;
- Preparar a ferramenta de documentação: estabelecer um padrão de documentação JAD.

A etapa Sessão consiste na realização das reuniões de trabalho com os participantes selecionados na fase Preparação, objetivando produzir as informações necessárias ao desenvolvimento do sistema. As suas sub-etapas são:

- Preparação do ambiente: a arrumação das mesas deve ser em forma de “U”, os equipamentos audiovisuais devem ser verificados, o *check-list* deve ser conferido, auxílio visuais devem ser instalados e recomenda-se preparação prévia de pastas para os participantes;
- Condução da sessão: começa com a apresentação dos participantes, seguido da explicação da agenda, revisão da perspectiva gerencial, desenvolvimento junto com o grupo de uma lista de regras da reunião e o início dos trabalhos seguindo a agenda;
- Documentação: é de responsabilidade de um documentador registrar tudo que acontece na sessão;
- Encerramento da sessão: revisão geral da agenda, avaliação da sessão e entrega da documentação produzida na sessão. Quando for a última sessão do projeto, é recomendável que seja feita a apresentação dos resultados alcançados.

A etapa Revisão tem como finalidade rever a documentação produzida na sessão e examinar possíveis melhorias na sistemática adotada. Essa etapa pode ser dividida em rever a documentação, examinar as avaliações e preparar a pasta do projeto JAD.

Segundo Sommerville (2003), JAD adota as seguintes diretrizes:

- Um encontro é realizado em um local neutro e conta com a participação dos desenvolvedores e dos clientes; regras para preparação e participação são estabelecidas; uma agenda detalhada o bastante para cobrir os pontos importantes, mas simples o suficiente para encorajar o livre fluxo de idéias, é sugerida; um moderador para controlar o encontro é escolhido; um mecanismo de definição é utilizado; e a meta é identificar o problema, propor elementos de solução, negociar diferentes abordagens e

especificar um conjunto preliminar de requisitos de solução em um clima que facilite a realização da atividade;

- A técnica de entrevista é usada nos primeiros encontros para definir o escopo do problema e a percepção global de uma solução e escrever a requisição do problema. Cada participante elabora uma lista de objetos que envolvem o sistema, (produzidos e usados por ele), uma lista de operações que interage com a lista de objetos e outra lista das restrições dos critérios de desempenho. Estas listas devem ser pequenas, mas que reflitam as perspectivas de cada um;
- No próximo encontro, apresenta-se a justificativa do produto para que os participantes saibam da sua necessidade, divulgam-se as listas para críticas e discussões. A partir dessa demonstração, cria-se uma lista com a conclusão do grupo, esta lista única abrange as outras. Neste encontro, deve-se dividir a equipe para desenvolver uma mini-especificação. Depois, cada sub-equipe apresenta sua mini-especificação e são feitas adições e supressões necessárias. Cada participante faz uma lista de critérios de validação, um ou mais participantes ficam responsáveis por escrever a especificação dos requisitos.

Segundo Paula Filho (2003), JAD apresenta as seguintes vantagens: i) comprometer os usuários com os requisitos que esses têm o poder de decisão sobre eles; ii) reduzir o prazo de levantamento da especificação dos requisitos; iii) eliminar requisitos de valor questionável; iv) reduzir diferenças de interpretação dos requisitos entre usuários e desenvolvedores; v) produzir um primeiro esboço das interfaces de usuário; e vi) trazer a tona, mais cedo possível, problemas políticos que possam surgir.

As desvantagens do JAD são: i) a exigência da disponibilidade de tempo dos participantes para as sessões; ii) se a preparação for insuficiente, a sessão pode não ter sucesso; iii) se o relatório de uma sessão estiver incompleto, pode por em risco a próxima sessão; e iv) a cultura organizacional pode não ser compatível com a aproximação JAD (Kontonya & Sommerville, 1998).

3.5 Brainstorming

Brainstorming, também chamada de tempestade de idéias, é uma técnica muito eficiente para grupos desenvolverem idéias criativas. Essa técnica cria idéias novas, resolve problemas e motiva e desenvolve equipes. Motiva porque envolve os membros de uma equipe em uma discussão gerencial maior e essa nova equipe trabalha junto.

Entretanto, o *brainstorming* não é simplesmente uma atividade aleatória, a técnica necessita ser estruturada e seguir regras (Maguire, 1998).

De acordo com Kirawowski (1997), o *brainstorming* é uma das várias técnicas de reuniões de grupo, possivelmente a mais conhecida e a mais antiga, utilizada para originar idéias. A essência básica dessa técnica é reunir um grupo de especialistas no sistema e no negócio para que cada participante possa estimular o outro a criar idéias que auxiliem a encontrar a solução do problema em uma ou mais reuniões. Ainda, segundo Kirawowski (1997), as idéias que surgirem e forem discutidas nestas reuniões não podem ser recriminadas ou julgadas. A técnica pode ser aplicada no começo da fase do desenvolvimento onde pouco do projeto é conhecido e são necessárias idéias novas.

Maguire (1998) afirma que *brainstorming* é empregada para gerar novas idéias, permitindo que a mente fique livre para aprovar as idéias que forem sugeridas e deixando liberdade para a criatividade. A consequência de um *brainstorming* realizada com sucesso são várias boas idéias e a percepção de que todos participaram da solução do problema.

Bellino & Case (2005) afirmam que o *brainstorming* é parte essencial do processo de pensamento de um negócio ou nova idéia em toda e qualquer organização. Quando esta técnica é adequadamente executada, ela pode conduzir a uma dinâmica e valiosa troca de idéias e iniciativas. Porém, quando mal conduzida, ela pode promover ressentimento, tensão interna e ser contra produtivo. Os autores destacam algumas boas práticas para conduzir uma sessão *brainstorming* para que seja ao mesmo tempo criativa e bem-sucedida:

- Estabelecer o papel do líder;
- Definir a tarefa e manter o controle na sua mão;
- Indicar o objetivo da sessão de *brainstorming*;
- Criar uma atmosfera positiva;
- Registrar as idéias;
- Incentivar o fluxo das idéias;
- Monitorar com cuidado uma comunicação verbal e não-verbal.

De acordo com Berkun (2004), para evitar problemas decorrentes de um *brainstorming* mal elaborado e mal conduzido, deve-se ficar atento a:

- Local;

- Ter um propósito específico;
- Saber o que buscar e para que buscá-lo;
- Saber facilitar;
- Colocar os pontos principais em uma lista;
- O conforto é chave;
- Estabelecer regras para o grupo;
- Adiar críticas.

Freitas (2006) resume *brainstorming* da seguinte forma: primeiramente, as idéias devem ser ditas em voz alta, com o intuito de influenciar os demais, e escritas, para que não sejam perdidas. As idéias não são descartadas. Depois, as idéias são apresentadas para o grupo novamente, que inicia um processo de seleção das idéias, podendo alterá-las ou mesclá-las. O *brainstorming* é mais eficiente quando cada participante possui uma parte do conhecimento de algum aspecto do problema.

As vantagens do *brainstorming* são: i) não ter críticas e julgamentos que ajudam a abolir algumas das dificuldades inerentes ao processo; ii) evitar a tendência a limitar o problema muito cedo; e iii) fornecer uma interação social mais confortável do que algumas técnicas de grupo mais estruturadas e pode ser aprendida, com muito pouco investimento. A desvantagem é, por ser um processo relativamente não estruturado, não produzir a mesma qualidade ou nível de detalhes de outros processos (Digiampietri, 2004).

3.6 Prototipação

A prototipação é uma técnica que consiste basicamente em implementar uma parte do sistema para que o usuário possa dar um retorno útil ao processo de requisitos. Essa implementação é feita a partir de uma especificação preliminar com objetivo de simular a aparência e a funcionalidade do software mesmo que de forma incompleta. A prototipação é utilizada no processo de levantamento de requisitos quando existe nível de incerteza elevado ou quando é preciso obter um rápido retorno dos usuários. O protótipo desenvolvido, depois de utilizado, pode proporcionar aprovação, interação, avaliação e alterações nas características de usabilidade e de funcionalidade do software proposto na especificação (Kontonya & Sommerville, 1998). Com o objetivo maior de esclarecer os requisitos do usuário e conseguir defini-los precisamente, recomenda-se o uso da técnica.

Para Souza (2006), a prototipação possibilita alcançar:

- Reações iniciais do usuário: observar e entrevistar os usuários para capturar o sentimento deles com o sistema;
- Sugestões do usuário para refinar ou alterar o protótipo: guiando os desenvolvedores para atender da melhor forma as necessidades dos usuários;
- Inovações: novas capacidades, não imaginadas antes da interação com o protótipo;
- Informações para revisão de planos: definir prioridade e alterar o planejamento.

As diretrizes para o desenvolvimento de um protótipo são (Souza, 2006):

- Trabalhar com módulos gerenciáveis: o protótipo não precisa ser o sistema completo;
- Construir o protótipo rapidamente: agiliza o processo utilizando ferramentas apropriadas;
- Modificar o protótipo em iterações sucessivas: o protótipo deve ser modificado para atender as necessidades dos usuários;
- Enfatizar a interface com o usuário: as interfaces do protótipo devem permitir que o usuário interaja facilmente com o sistema;
- Envolver os usuários: fazendo com que eles experimentem o protótipo, observando as reações e incentivando que sejam feitas críticas e sugestões.

Sommerville (2003) afirma que a prototipação é parte do processo de requisitos. Para o autor, é difícil para os usuários finais preverem como vão utilizar novos sistemas de software para dar apoio ao seu trabalho diário. Se esses sistemas forem grandes e complexos, provavelmente será impossível fazer uma avaliação antes que o sistema seja desenvolvido e implantado.

Para lidar com essa dificuldade, Sommerville (2003) aponta a abordagem evolucionária de desenvolvimento, ou seja, um sistema incompleto é oferecido ao cliente e, depois, o sistema é modificado e aumentado até que os requisitos dos usuários se tornem completamente claro. Outra alternativa apresentada pelo autor é o desenvolvimento de um protótipo descartável para ajudar na análise e na validação dos requisitos. Depois da avaliação, o protótipo é descartado e um sistema de qualidade é construído.

A prototipação evolucionária se baseia na idéia de desenvolver uma implementação inicial e expô-la aos usuários, para que eles possam avaliar e opinar sobre o sistema, aperfeiçoando essa implementação ao longo de vários estágios até um sistema apropriado seja construído. Com isso, é possível fornecer o sistema rapidamente e obter um alto

comprometimento do usuário com o software. Entretanto, quando os sistemas desenvolvidos são grandes, podem surgir problemas de gerenciamento, de manutenção e contratuais (Sommerville, 2003).

A prototipação descartável amplia o processo de análise de requisitos, com a finalidade de reduzir os custos totais do ciclo de vida. Sua principal função é esclarecer os requisitos e fornecer informações adicionais para os gerentes avaliarem riscos do processo. O protótipo será descartado, após a análise dos requisitos e a avaliação de riscos. Um problema da prototipação descartável executável é o modo de utilização do protótipo poder não corresponder a maneira como o sistema final entregue é utilizado. Os modelos de protótipos com base em papel, da interface com usuário do sistema, são eficientes para ajudar os usuários a melhorar um projeto de interface e trabalhar em cenários de uso comum (Sommerville, 2003).

Sommerville (2003) afirma que, para a prototipação de nível industrial, três técnicas de desenvolvimento rápido podem ser utilizadas: i) o desenvolvimento com linguagem dinâmica de alto nível; ii) a programação de banco de dados; e iii) a montagem de componentes e aplicações.

Essas técnicas de prototipação enfatizam a rapidez do fornecimento, em vez de outras características do sistema, como desempenho, facilidade de manutenção e confiabilidade.

Para Paula Filho (2003), o protótipo descartável tem por objetivo abordar aspectos críticos dos requisitos de um produto, implementando de maneira rápida um pequeno subconjunto de funções do sistema. O protótipo descartável pode ajudar a decidir questões que sejam cruciais para o sucesso de um sistema e que sejam tratáveis em um experimento de curta duração. O autor indica essa técnica, para os seguintes estudos: i) alternativas de interface de usuário; ii) problema de comunicação com outros sistemas; e iii) viabilidade de desempenho.

Paula Filho (2003) considera que o protótipo descartável pode ser construído no mesmo ambiente que o sistema final, em um ambiente de desenvolvimento mais rápido ou no mesmo ambiente considerado como ferramenta de documentação. Segundo o autor, alguns exemplos de áreas, onde protótipos são usados (com seus respectivos ambientes de prototipagem), são:

- Interface de usuário – protótipo visual em ambiente definitivo ou em ambiente de programação rápida ou ainda com ferramentas de desenho;
- Relatórios textuais – processador de textos ou ferramenta de geração de relatórios;
- Relatórios gráficos – ferramenta de desenho ou ambiente de programação rápida com biblioteca gráfica;
- Organização e desempenho de banco de dados – ferramenta de desenvolvimento rápido integrada ao sistema de gerência de banco de dados que se pretende usar;
- Cálculos complexos – planilha ou ferramenta matemática;
- Partes de tempo de resposta crítico, em sistemas de tempo real – pequeno programa de teste no ambiente alvo;
- Tecnologia no limite do estado-da-arte – pequeno programa de teste no ambiente que for mais adequado.

As desvantagens da prototipação enfatizadas por Anacleto (2005) são: i) a gerência do projeto por serem, normalmente, necessárias várias iterações para refinar um protótipo, surgindo a dificuldade sobre o término da prototipação; e ii) a possibilidade do protótipo ser considerado como o sistema final, visto que a qualidade pode não ter sido devidamente analisada.

Para Paula Filho (2003), as vantagens da utilização dessa técnica são: i) a redução dos riscos na construção; ii) aumento da manutenibilidade; iii) aumento da estabilidade dos requisitos e iv) oportunidade para treinamento dos programadores menos experientes.

3.7 Considerações Finais

Este capítulo teve a finalidade de esclarecer como cada técnica de elicitação é aplicada no processo de elicitação de requisitos e como otimizar essa aplicação para que se possa obter um conjunto de requisitos que defina corretamente o sistema. As técnicas foram detalhadas com objetivo de oferecer subsídios suficientes para realizar comparações entre elas e suas classificações. As cinco técnicas abordadas podem ser úteis, sendo um grande auxílio ao processo de levantamento de requisitos.

4 SWREQUIREMENT

4.1 Considerações Iniciais

Neste capítulo, as técnicas de elicitação de requisitos são integradas às atividades do processo de levantamento e análise de requisitos. Cada atividade é relacionada a uma técnica, tendo em vista os aspectos que a indicam mais aplicável.

Foi considerada cada atividade da etapa de levantamento e análise de requisitos do processo de engenharia de requisitos proposto por Sommerville (2003). Este processo foi escolhido por ser um processo bem definido, compostos por fases coerentes que se mostram necessárias para a realização de uma boa elicitação de requisitos. O processo também pode ser considerado simples e objetivo facilitando a integração das técnicas a suas fases bem como sua aplicação.

4.2 Visão Geral do SWRequirement

SWRequirement é uma estratégia de elicitação de requisitos, sendo um processo modificado para utilizar técnicas de elicitação em cada etapa do processo de levantamento e análise de requisitos, proposto por Sommerville (2003) (Tabela 4-1).

Tabela 4-1 – Atividades, Técnicas Indicadas e Artefatos de Software Gerados por SWRequirement

Atividade	Técnica Indicada	Artefatos de Software Gerados
Compreensão do domínio	Entrevista	Proposta de Especificação de Requisitos
Coleta de requisitos	JAD	Documento de Requisitos Detalhado do Sistema
Classificação	Questionário (Auto-Resposta em Grupo)	Documento de Requisitos Detalhado do Sistema Agrupados
Resolução de conflitos	<i>Brainstorming</i>	Lista de Conflitos
Definição de prioridade	Questionário (Auto-resposta via <i>e-mail</i>)	Lista de Prioridades
Verificação de requisitos	Prototipação	Documento de Requisitos Detalhado do Sistema Aprovado e Lista de Possíveis Riscos

SWRequirement começa com a Compreensão do Domínio, sendo realizada pela técnica de Entrevista resultando o artefato de software Proposta de Especificação de Requisitos. Em seguida, a Coleta de Requisitos é realizada utilizando a técnica JAD para elaborar o artefato de software Documento de Requisitos Detalhado do Sistema. Para

refinar a especificação, são realizadas: a Classificação, a Resolução de Conflitos, a Definição de Prioridades e a Verificação de Requisitos.

A Classificação é realizada utilizando a técnica de Questionários, sendo finalizada com a atualização do artefato de software Documento de Requisitos Detalhado do Sistema. A Resolução de Conflitos acontece por meio da técnica de *Brainstorming*, gerando o artefato de software Lista de Conflitos. A Definição de Prioridade é realizada usando a técnica de Questionários, resultando no artefato de software Lista de Prioridade. A Verificação de Requisitos utiliza a técnica de Prototipação, gerando o artefato de software Lista de Riscos.

A Figura 4-1 mostra o fluxo de atividades que compõem o **SWRequirement**, bem como os papéis e os artefatos de entrada e saída utilizados em cada atividade. A atividade inicial é Compreender o Domínio da aplicação, que deve ser realizada pelo gerente de projeto e pelo engenheiro de requisitos através de entrevistas com o cliente, gerando o artefato de software Proposta de Levantamento de Requisitos. A próxima atividade é Coletar Requisitos, essa necessita que os envolvidos participem das sessões JAD e o engenheiro de requisito prepare, planeje e lidere as sessões JAD; ao final desta atividade o artefato de software Documento de Requisitos Detalhado do Sistema é gerado. Em seguida, a atividade Classificar Requisitos é executada pelo engenheiro de requisitos, aplicando questionários aos usuários do sistema; desta atividade, o artefato de software Documento de Requisitos Detalhado do Sistema é atualizado.

A atividade seguinte, Resolver Conflitos, realizada com clientes e usuários pelo engenheiro de requisitos, utiliza a técnica de *Brainstorming*; os resultados desta atividade são os artefatos de software Lista de Conflitos e Documento de Requisitos Detalhado do Sistema atualizado. Ao término desta atividade, inicia-se a atividade Definir Prioridade, que consiste na definição das prioridades dos requisitos a serem implementados, realizada pelo engenheiro de requisitos e pelos usuários, utilizando a técnica de questionários; os resultados desta atividade são os artefatos de software Lista de Prioridade e Documento de Requisitos Detalhado do Sistema atualizado.

A última atividade do **SWRequirement** é Verificar Requisitos realizada pelo engenheiro de requisitos, utilizando um protótipo desenvolvido e disponibilizando-o para alguns usuários para que estes forneçam um *feedback* para o engenheiro de requisitos;

nesta atividade, o artefato de software Documento de Requisitos Detalhado do Sistema é atualizado e o artefato de software Lista de Possíveis Risco é gerado.

4.3 Fases do Processo de Levantamento e Análise de Requisitos

Essa seção apresenta as fases do processo de levantamento e análise de requisitos mostrando e justificando qual a técnica de elicitação é indicada e quais os artefatos de software são gerados em cada fase do processo. As fases detalhadas nesta seção formam o **SWRequirement**.

4.3.1 Compreensão do Domínio

Os analistas devem desenvolver sua compreensão do domínio de aplicação, ou seja, conhecer o contexto do negócio em que software atuará. Essa atividade consiste em um levantamento abrangente e pouco específico.

A utilização da técnica de entrevista poderia auxiliar nesta atividade, explicitando as informações obtidas com o cliente. Por seu caráter simples, a técnica é recomendada tendo em vista a agilidade que a atividade requer. O uso da entrevista aberta proporciona maior interação entre o cliente e o analista, deixando o entrevistado mais confortável para descrever o domínio de aplicação. Mesmo esta atividade não requerendo um grande detalhamento, é indicada que sejam seguidas as etapas da técnica descritas no referencial teórico.

Por fim, será obtido um resumo dos pontos discutidos para documentar essa atividade. Através da sua análise, será gerado um artefato de software chamado Proposta de Levantamento de Requisitos. Neste artefato de software, deve conter a missão do sistema, uma lista simplificada de funções, requisitos de qualidade (críticos), metas gerenciais e uma estimativa de custo e prazo para o levantamento de requisitos.

4.3.2 Coleta de Requisitos

A coleta de requisitos é o processo de interagir com *stakeholders* do sistema para descobrir seus requisitos; nesta atividade, os requisitos são definidos e detalhados. Esta é a atividade a qual se requer mais esforço durante o levantamento de requisitos, pois serão definidas as características e as funções do sistema. Na fase Elicitação de Requisitos, a

compreensão do domínio aumenta gradativamente à medida que os requisitos são identificados.

Nesta etapa do processo, o artefato de software Proposta de Levantamento de Requisitos elaborado na atividade Compreensão de Domínio deve ser usado para iniciar a coleta de requisitos e auxiliar durante essa atividade, fornecendo as informações primordiais para o entendimento do sistema.

A técnica que melhor atende a essa atividade é JAD por seu nível de interação e detalhamento. JAD mostra, pela sua própria definição (técnica destinada a extrair informações de alta qualidade dos usuários em curto espaço de tempo, através de reuniões estruturadas que buscam decisões por consenso), ser muito útil a essa atividade.

Sua característica de envolver os participantes do projeto, deixando transparente a sua importância no projeto, motiva-os a serem úteis, retirando barreiras que os participantes estabelecem para se comunicar. Essa ação torna os participantes mais propensos a contribuir com informações relevantes e necessárias.

Outros fatores que viabilizam a utilização dessa técnica na atividade de coleta de requisitos é JAD se ajustar à realidade de qualquer organização, aproxima os participantes e torna-se uma ferramenta para a conversão do conhecimento implícito em conhecimento explícito. Portanto, quando a técnica é bem aplicada, consegue-se unir os participantes para alcançar o sucesso do projeto.

A etapa Preparação de JAD possui algumas sub-etapas que não precisam ser feitas por serem atividades relacionadas com a compreensão do domínio, essas informações foram adquiridas na etapa correspondente. As outras etapas de JAD devem ser seguidas completamente para garantir a eficiência da técnica.

A coleta de requisitos deve originar um artefato de software Documento de Requisitos Detalhado do Sistema. Esse artefato de software deve conter uma introdução (objetivo do documento, escopo do sistema, materiais de referência, definições e siglas e a visão geral do documento), uma descrição geral do sistema e os requisitos específicos (requisitos de interface³, funcionais e não funcionais).

³ Requisitos referentes às entradas e às saídas do sistema (Paula Filho, 2003).

4.3.3 Classificação

A classificação é a atividade que considera o conjunto não estruturado dos requisitos e os organiza em grupos coerentes, ou seja, agrupa os requisitos relacionados ou tratam da funcionalidade do sistema que mantém alguma relação ou dependência entre elas.

Para realizar essa atividade, deve-se ter o artefato de software produzido na etapa anterior, pois os requisitos elicitados na coleta de requisitos serão classificados nesta atividade seguindo os critérios adequados.

A técnica indicada para essa atividade é o questionário, documento usado para conduzir uma ou mais pessoas a responder a uma ou mais perguntas, por ser uma técnica objetiva. Um fator determinante na escolha do questionário é as questões poderem ser pré-determinadas, tornando eficiente na avaliação de tendências de opiniões a respeito de requisitos específicos e bem definidos.

Os procedimentos que envolvem a elaboração de um questionário e sua aplicação devem ser seguidos. As perguntas devem ser fechadas para limitar a classificação dos requisitos observando a sua funcionalidade ou as suas dependências; as perguntas são elaboradas a partir do conjunto de requisitos definido na coleta de requisitos e da identificação de funções do software e problemas a serem resolvidos por ele.

Para aplicar os questionários, deve-se usar a forma definida como Auto-Resposta em Grupo. Essa forma de aplicação possibilita que sejam sanadas as dúvidas sobre as perguntas e a finalidade do questionário (caso surjam) e, também, para obter as respostas em curto prazo de tempo, agilizando a atividade de classificação, pois essa não deve demandar muito tempo.

Após a aplicação do questionário, as respostas devem ser analisadas e os requisitos agrupados conforme essa análise sugerir. Depois dessa análise, o artefato de software Documento de Requisitos Detalhado do Sistema deve ser atualizado, para que os requisitos sejam ordenados de acordo com a classificação no documento.

4.3.4 Resolução de Conflitos

O processo de levantamento de requisitos envolve vários *stakeholders* o que, geralmente, faz surgir conflitos entre os requisitos. Essa atividade do processo tem o objetivo de solucionar esses conflitos, tornando os requisitos coerentes com a real necessidade da organização.

A partir do artefato de software Documento de Requisitos Detalhado do Sistema devidamente atualizado será iniciada a resolução de conflitos, a ordem dos requisitos abordados será a do documento de requisitos, portanto, como eles estão ordenados por finalidade, funcionalidade e dependências entre eles, facilita a resolução de conflitos.

Um *brainstorming* pode ser uma boa técnica para essa atividade do processo, por seu caráter de liberdade de expressão e de abolição de críticas. Os conflitos serão debatidos com respeito e os participantes defenderão suas idéias sem constrangimento até que seja estabelecido um consenso em relação aos requisitos. Devem ser convidados para o *brainstorming*, os *stakeholders* envolvidos nos requisitos que apresentam conflito, tanto usuários quanto desenvolvedores.

Primeiramente, será realizada apenas uma reunião, mas, se não for suficiente para solucionar os conflitos, uma nova reunião deve ser marcada o mais breve possível, para finalizar a atividade de resolução de conflitos. É importante que as boas práticas indicadas para planejar e conduzir um *brainstorming* sejam realizadas, para manter a reunião bem estruturada.

O resultado dessa atividade é o artefato de software Lista de Conflitos que informa em quais requisitos ocorreu conflito e como eles foram solucionados. Usando esta lista como referência, deve atualizar o artefato de software Documento de Requisitos Detalhado do Sistema inserindo as informações provenientes dessa atividade.

4.3.5 Definição de Prioridade

Normalmente, em um conjunto de requisitos, alguns deles são mais importantes e devem ser finalizados antes. Por isso, esta atividade do processo de levantamento de requisitos se faz necessária, pois define as prioridades dos requisitos. Essas prioridades serão determinadas junto com os *stakeholders*.

O artefato de software Documento de Requisitos Detalhado do Sistema atualizado com as informações da etapa anterior é à entrada dessa atividade, onde os requisitos estão agrupados corretamente e, possivelmente, não existem conflitos entre eles, faltando definir a prioridade dos requisitos.

Nesta atividade, o uso de questionários é um facilitador, pois perguntas diretas podem ser elaboradas, facilitando para o respondente e tornando essa tarefa mais rápida. Os procedimentos de elaboração e de aplicação dos questionários sugeridos precisam ser

efetuados, para assegurar sua estrutura. O questionário deve conter perguntas fechadas e bem direcionadas para estabelecer as prioridades dos requisitos, podendo ser um questionário mais simples, para facilitar a sua análise. Neste caso, o questionário pode ser aplicado da forma conhecida por Auto-resposta via *e-mail*, devido a sua simplicidade e aos *stakeholders* estarem bem envolvidos e engajados no projeto.

Como resultado da definição de prioridade, o artefato de software Lista de Prioridades deve ser elaborado, informando qual a ordem de implementação dos requisitos do sistema.

4.3.6 Verificação de Requisitos

Nesta etapa, os requisitos são verificados, a fim de descobrir se eles são consistentes e completos e em concordância com o que realmente os *stakeholders* desejam do sistema. Essa atividade é extremamente importante, pois os equívocos, quando identificados neste estágio, evitam re-trabalho e mudanças nos requisitos após o início da implementação.

Na verificação de requisitos, serão usados os artefatos de software Documento de Requisitos Detalhado do Sistema e Lista de Prioridades. A partir desses documentos, os principais requisitos do sistema serão implementados, embora de forma incompleta.

A técnica que melhor atende a verificação de requisitos é a prototipação por tratar-se de uma amostra do sistema que simula parte da aparência e da funcionalidade do software. Portanto, com a utilização da prototipação, o usuário pode dar uma resposta mais significativa para o processo de levantamento de requisitos, mostrando como reage ao sistema. O protótipo de um sistema pode auxiliar o usuário a sugerir melhorias e alterações necessárias, que ajudarão os desenvolvedores a satisfazer o cliente.

A técnica é indicada, pois é necessário um retorno rápido dos usuários. A verificação de requisitos deve ser feita no menor tempo possível e de forma eficiente, pois essa ocorre na finalização do processo de levantamento de requisitos, sendo a etapa intermediária entre esse processo e o processo de implementação do software. Para obter esse retorno rápido, indica-se a construção de um protótipo descartável por não priorizar qualidade e desempenho; um modelo de protótipo de papel pode ser usado para auxiliar na validação da interface.

Após o uso do protótipo, o artefato de software Documento de Requisitos Detalhado do Sistema deve ser atualizado de acordo com o *feedback* dos *stakeholders* e aprovado pelo

cliente. Além disso, é gerado um artefato de software Lista de Possíveis Riscos baseado no uso do protótipo, facilitando o gerenciamento de riscos.

4.4 Considerações Finais

Neste capítulo, foi sugerida uma combinação das atividades realizadas nas técnicas de elicitação de requisitos apresentadas no capítulo 3. As atividades combinadas visam atender ao processo de levantamento e análise de requisitos de um sistema que foram detalhadas no capítulo 2. Assim, objetiva-se melhorar a elicitação de requisitos, fornecendo aos desenvolvedores um conjunto maior e mais preciso de informações. **SWRequirement** consta de atividades, de técnicas e de artefatos gerados a partir destas, com intuito de formalizar e otimizar a elicitação de requisitos feita por uma empresa desenvolvedora de software.

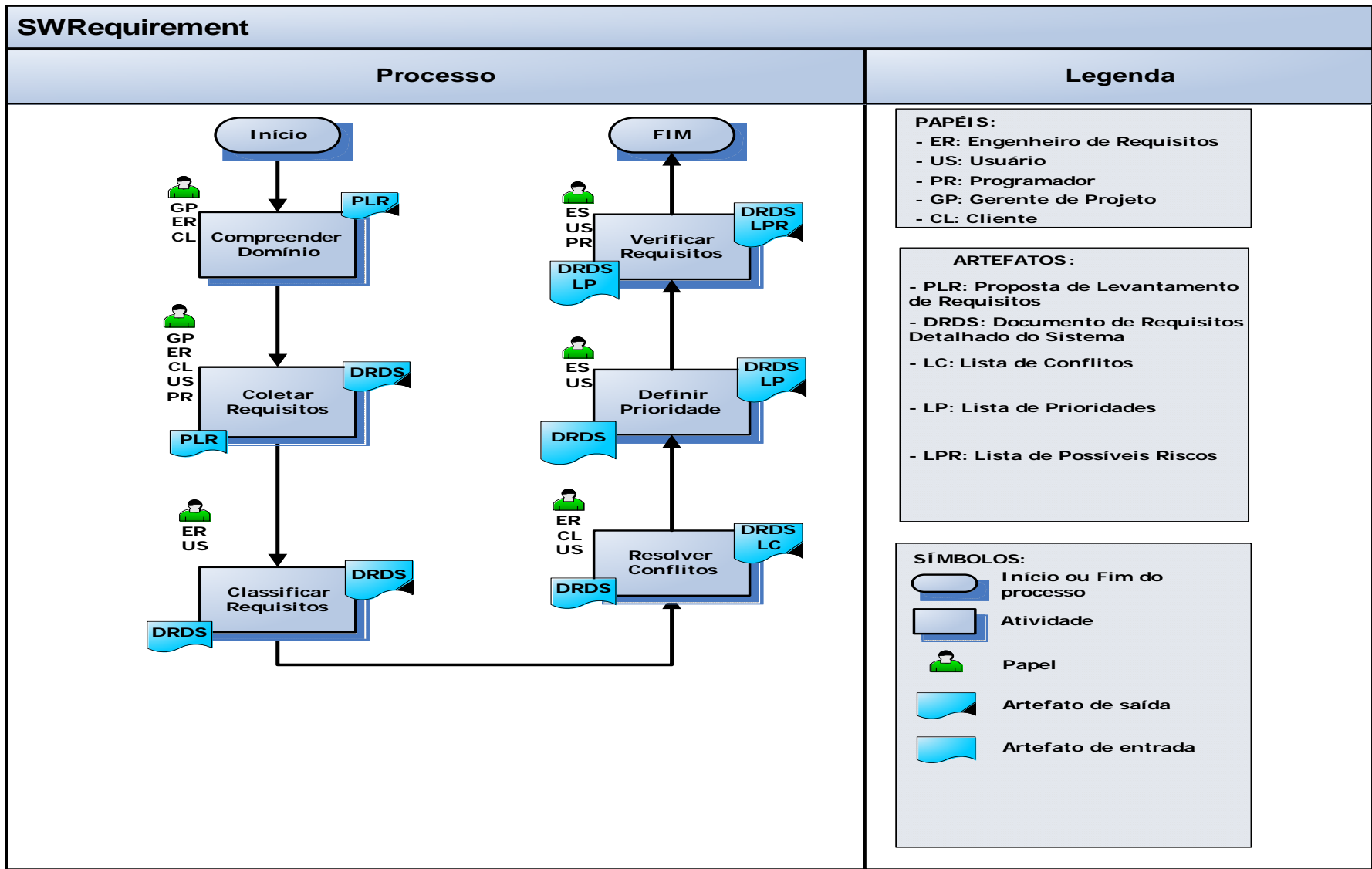


Figura 4-1 – Fluxo de Atividades do SWRequirement

5 CONSIDERAÇÕES FINAIS

5.1 Conclusões

Neste trabalho, foi proposta uma integração de algumas técnicas de elicitação de requisitos estudadas e apresentadas adequando-as ao processo de levantamento e análise de requisitos, proposto por Sommerville (2003) – **SWRequirement**. Buscou-se com essa iniciativa apoiar o engenheiro de software na elicitação de requisitos, facilitando a busca por informações necessárias para especificar um sistema. A principal característica de **SWRequirement** é a combinação de técnicas de elicitação para realizar as atividades do processo. Um processo de levantamento e análise de requisitos integrando as técnicas de elicitação de requisitos mostra-se importante, pois, através da sua utilização, resultados eficazes podem ser conquistados, levando a construção de sistemas de qualidade e condizentes com os anseios dos usuários.

Os requisitos descrevem o sistema e devem instruir os desenvolvedores para a construção do sistema, sendo necessário obtê-los com a maior precisão possível. Dispensar esforço e tempo buscando requisitos de qualidade é extremamente importante para o processo de software de maneira geral, pois possibilita a economia de trabalho e custo neste processo. Os engenheiros de requisitos necessitam de meios de definir os requisitos, o processo proposto pode ser visto como uma opção para alcançar requisitos com as características necessárias.

As informações necessárias para especificar um sistema não são conquistadas simplesmente questionando os clientes e os usuários sobre o que eles desejam, mas com análise, técnicas e processos bem estruturados e definidos no ambiente de desenvolvimento de software. Tem-se que realizar as atividades de levantamento de requisitos priorizando a qualidade das informações para garantir o possível sucesso do sistema. Essa importância pode ficar mais relevante visto que um sistema mal especificado dificilmente atenderá as expectativas dos usuários e clientes.

Pode-se concluir que **SWRequirement** objetiva evitar os problemas, que normalmente cercam a etapa de elicitação de requisitos de um sistema. **SWRequirement** tende a ser simples para possibilitar sua utilização em empresas de todos os níveis de maturidade, mas necessita de pessoas capacitadas para realizá-lo, devido a complexidade das informações buscadas e das técnicas que devem ser utilizadas. **SWRequirement**,

quando bem empregado, pode possibilitar inúmeros ganhos ao desenvolvimento de software, como a redução de custos e tempo de conclusão do sistema.

5.2 Contribuições

As contribuições desse trabalho são:

- **SWRequirement** auxilia e guia a elicitação de requisitos de um sistema, utilizando técnicas de elicitação de requisitos amplamente conhecidas para assegurar a qualidade das informações obtidas;
- O detalhamento das técnicas de elicitação escolhidas mostrando as melhores práticas para aplicá-las e conquistar melhores resultados na elicitação de requisitos.

Portanto, pode-se considerar que este trabalho contribui com a área de Engenharia de Software, mais precisamente em Engenharia de Requisitos, buscando mostrar sua importância e facilitar o levantamento e análise de requisitos de um sistema. Com a composição das técnicas de elicitação e sua utilização nas atividades do processo de levantamento e análise de requisitos, pode-se melhorar a capacidade de elicitação de requisitos de uma empresa de software, aumentando a qualidade de sua especificação.

5.3 Trabalhos Futuros

Como sugestão de trabalhos futuros, pensa-se em:

- Utilizar efetivamente o **SWRequirement** na elicitação de requisitos de um sistema real para verificar se seus objetivos serão alcançados com sucesso, trazendo os benefícios esperados ao desenvolvimento do sistema em questão;
- Definir o *lay-out* dos artefatos gerados por **SWRequirement**;
- Desenvolver uma ferramenta CASE (*Computer Aided Software Engineering*) para automatizar a construção dos artefatos de software gerados pelo **SWRequirement**.

Enfim, há possibilidades em dar continuidade a este processo, sendo utilizando-o de fato ou em um processo de desenvolvimento de software completo.

REFERÊNCIAS BIBLIOGRÁFICAS

- ANACLETO, J. C. **Técnicas de Levantamento de Requisitos**. Notas de aula. Disponível em: <http://www.dc.ufscar.br/~junia/aula%203.pdf>. Consulta em 10/02/2007.
- BASTOS JUNIOR, P. R. O. **Elicitação de requisitos de software através da utilização de questionários**. Rio de Janeiro: PUC-Rio, Departamento de Informática, 2005.
- BATISTA, E. A.; **Uma Taxonomia Facetada para Técnicas de Elicitação de Requisitos**. Dissertação de Mestrado, Instituto de Computação, UNICAMP, Campinas, 2003.
- BELGAMO, B. **Estudo Comparativo sobre as Técnicas de Elicitação de Requisitos do Software**. SBC, 2000.
- BELLINO, R., CASE, S.; **Como Conduzir Uma Sessão Criativa E Bem-Sucedida de "Brainstorming"?**. Disponível em: http://www.catho.com.br/jcs/inpuer_view.phtml?id=7252. Consulta em 10/03/2007.
- BERKUN, S. **How to run a brainstorming meeting**. Disponível em <http://www.scottberkun.com/essays/essay34.htm>. Consulta em 10/03/2007.
- BORTOLI, L. A.; PRICE, A. M. A. **O Uso de Workflow para Apoiar a Elicitação de Requisitos**, III Workshop de Engenharia de Requisitos, 2000.
- BROOKS, F. **No Silver Bullet: Essence and Accidents of Software Engineering**. Computer, Apr. 1987.
- CAMACHO, C. **Gerenciando Conflitos em Reuniões: Uma Estratégia para a Elicitação de Requisitos de Software**. Rio de Janeiro: PUC-Rio, Departamento de Informática, 2005.
- CARVALHO, A. E., TAVARES, H. C., CASTRO, J. **Uma Estratégia para Implantação de uma Gerência de Requisitos visando a Melhoria dos Processos de Software**. Anais do WER01 - Workshop em Engenharia de Requisitos, Buenos Aires, Argentina, Novembro, 2001.
- COSTA, O. W. D. da. **JAD, Joint Application Design**. Rio de Janeiro: Infobook, 1994.
- CURTIS, Bill; KRASNER, Herb e ISCOE, N. **A Field Study of the Software Design Process for Large Systems**. Communications of the ACM, November 1988, vol.31, N. 11.

- DIGIAMPIETRI, L. A. **Extração de requisitos**. Notas de aula 2004. Disponível em: <http://www.ic.unicamp.br/~luciano/mc426/modulo2-v.pdf>. Consulta em 20/06/2007.
- FREITAS, Danilo Pestana de. **Ampliando a Colaboração no Levantamento de Requisitos de Sistemas**. Dissertação de Mestrado em Informática – Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2006.
- GIL, A. C. **Como Elaborar Projetos de Pesquisa**. 3. ed. São Paulo: Atlas, 1991.
- GOGUEN, J. **Requirements Engineering as the Reconciliation of Social and Technical Issues. Requirements Engineering: Social and Technical Issues**. Academic Press Inc., London, England, 1994.
- GOODE, W. J.; HATT, P. K. **Métodos em Pesquisa Social**. 4a ed. São Paulo: Comp. Ed. Nacional, 1960.
- IEEE-SA STANDARDS BOARDS IEEE Std 1233-1998: **IEEE Guide for Developing System Requirements Specifications**. Dezembro 1998.
- IEEE-SA STANDARDS BOARDS IEEE Std 830-1998: **IEEE Recommended Practice For Software Requirements Specifications**. Junho 1998.
- ISO9000-3: **Quality Management and Quality Assurance Standards**. 2001.
- JACKSON, M. **Software Requirements and Specifications**. First edition, Addison Wesley, 1995.
- KIRAWOWSKI, J. **Requirements Engineering and Specification in Telematics: Methods for User-Orientated Requirements Specification**. Human Factors Research Group, Cork, Ireland, 1997. Disponível em <http://www.ejeisa.com/nectar/respect/3.2/>. Consulta em 16/12/2006.
- KONTONYA, P.; SOMMERVILLE, I. **Requirements Engineering: Processes and Techniques**. Viley 1998.
- LAKATOS, E. M. **Fundamentos da Metodologia Científica**. São Paulo: Atlas, 1991.
- LEITE, J. C. S. P. **Qualidade de Software: Teoria e Prática**. Orgs. Rocha, Maldonado, Weber, Prentice-Hall, São Paulo, 2001. Capítulo 17.
- LOCH, J. M.; Paludo, M. A.; e Reis, D. R. **A metodologia Joint Application Design - JAD como interface para a conversão de conhecimento tácito em explícito**. Um

- exemplo de aplicação em desenvolvimento de software.** Disponível em: <http://www.ppgte.cefetpr.br/docentes/permanentes/dalcio/jad.pdf>. Consulta em 01/01/2007.
- LOPES, L. T. **Uma Proposta para Processo de Requisitos em Ambientes de Desenvolvimento Distribuído de Software.** Anais do WER03 - Workshop em Engenharia de Requisitos, Piracicaba-SP, Brasil, Novembro, 2003.
- LOPES, Paulo S. N. **Uma taxonomia da pesquisa na área de Engenharia de Requisitos.** Dissertação de Mestrado, IME/USP, São Paulo, 2002.
- MACAULAY, L. **Requirements Engineering.** Springer, 1996.
- MAGUIRE, Martin C. **RESPECT Deliverable D 5.3. RESPECT User-Centred Requirements Handbook.** HUSAT Research Centre, Loughborough, 1998. Disponível em <http://www.ejeisa.com/nectar/respect/5.3/>. Consulta em 10/01/2007.
- MATTAR, F. N. **Pesquisa de marketing.** 3a. ed., São Paulo: Atlas, 2001.
- PARASURAMAN, A. **Marketing research.** 2. ed., Addison Wesley Publishing Company, 1991.
- PAULA FILHO, W. de P. **Engenharia de software: fundamentos, métodos e padrões.** Wilson de Pádua Paula Filho. 2. ed. Rio de Janeiro: Livros Técnicos e Científicos, 2003.
- PRESSMAN, R. S. **Engenharia de software.** São Paulo: Makron Books, 2006.
- REIS, D. R. **Contributos para a melhoria da eficiência e da eficácia nas relações de cooperação entre universidades e pequenas e médias empresas industriais brasileiras.** Aveiro/Portugal. Tese de Doutorado, 2000.
- SELLTIZ, C.; COOK, S. W. **Métodos de pesquisa nas relações sociais.** 2a.ed., São Paulo: E.P.U., 1987.
- SOMMERVILLE, Ian. **Engenharia de software.** 6. ed. São Paulo: Addison Wesley, 2003.
- SOUZA, Vitor. **Elicitação de Requisitos.** Notas de Aula da disciplina Análise e Projeto Orientado a Objetos, UFES, 2006.