

UNIVERSIDADE PRESBITERIANA MACKENZIE

EDUARDO KIEFER

LEVANTAMENTO DE REQUISITOS: TÉCNICAS E FERRAMENTAS PARA
LEVANTAMENTO DE REQUISITOS EM DESENVOLVIMENTO DE SISTEMAS.

São Paulo

EDUARDO KIEFER

LEVANTAMENTO DE REQUISITOS: TÉCNICAS E FERRAMENTAS PARA
LEVANTAMENTO DE REQUISITOS EM DESENVOLVIMENTO DE SISTEMAS.

Monografia de Conclusão de Curso apresentado ao Programa de Pós-graduação Lato Sensu da Escola de Engenharia da Universidade Presbiteriana Mackenzie, como requisito parcial para a obtenção do Título de Especialista em Gestão de Projetos.

ORIENTADORA: Prof. Kassya Rigolon de Andrade

São Paulo

2013

As minhas filhas Thais e Manuela e minha amada esposa Camila pelo apoio em todas as minhas atividades e compreensão da minha ausência.

AGRADECIMENTOS

Agradeço primeiro a Deus, meu guia, por sua grandeza e bondade sempre me trazendo conforto e proteção durante toda a minha vida.

A minha esposa por me apoiar em todas as minhas decisões e ter me dado duas filhas maravilhosas, Thais e Manuela.

As minhas amadas filhas Thais e Manuela, que apesar da pouca idade aceitaram minhas ausências noturnas durante o curso e a elaboração desse trabalho.

Aos meus colegas de Pós-Graduação pelo aprendizado, oportunidades e companheirismo.

A todos os meus professores que auxiliaram na minha educação e na construção do meu caráter.

A professora Kassya Rigolon de Andrade pela paciência, disponibilidade e principalmente pelas orientações que ajudaram na concretização desse trabalho.

Aos demais professores do curso de Gestão de Projetos por compartilhar seus conhecimentos e experiências durante todo o curso.

RESUMO

O processo de levantamento de requisitos é fundamental para o sucesso ou fracasso de um sistema. Quando bem definido e estruturado pode auxiliar na busca de requisitos detalhados com clareza, facilitando a fase de desenvolvimento do *software*. O objetivo deste trabalho é apresentar um estudo dos conceitos de Engenharia de Requisitos, assim como as técnicas de levantamento de requisitos. As técnicas são descritas e seus processos relacionados, assim como as situações em que são mais indicadas. Escolher a técnica correta de levantamento de requisitos cria condições para que os requisitos sejam mais bem especificados e para que as possíveis falhas no desenvolvimento de sistemas fiquem dentro de uma faixa aceitável e tolerável, tendo como resultado final, *softwares* com muito mais qualidade.

Palavras-chave: Engenharia de Requisitos. Levantamento de requisitos.

Técnicas de levantamento de requisitos. Escopo.

ABSTRACT

The process of requirements gathering is critical to the success or failure of a system. When well defined and structured can help find the detailed requirements clearly, facilitating the development phase of the software. The objective of this paper is to present a study of the concepts of Requirements Engineering as well as the technical requirements elicitation. Such techniques are described and its related process, as well as the situations that are most suitable. Choose the correct technique for gathering requirements creates conditions for the requirements are further specified and the possible failures in systems development are within an acceptable range and tolerable, with the final result, with much higher quality software.

Keywords: Requirements Engineering. Gathering requirements.

Technical requirements elicitation. Scope.

LISTA DE ILUSTRAÇÕES

Gráfico 1	Problemas com projetos.	11
Fluxograma 2	Tipos de requisitos não funcionais.	19
Quadro 3	Exemplo de métricas não funcionais.	20
Quadro 4	Um requisito de domínio de um sistema de trens.	21
Figura 5	Processo de Engenharia de Requisitos.	25
Figura 6	Os componentes do processo de coletar requisitos.	27
Figura 7	Estrutura Pirâmide de entrevista	30
Figura 8	Estrutura Funil de entrevista	30
Figura 9	Estrutura Diamante de entrevista	30
Figura 10	Diagrama da visão geral do JAD	48

LISTAS DE ABREVIATURAS E SIGLAS

CASE Computer-Aided Software Engineering

IBM International Business Machines

JAD Joint Application Development

PMI Project Management Institute

TI Tecnologia da Informação

SUMÁRIO

1	INTRODUÇÃO	10
1.1	OBJETIVO	13
1.2	METODOLOGIA.....	14
1.2.1	Tipo de pesquisa	14
1.2.2	Procedimento metodológico	14
2	REQUISITOS DE SISTEMAS	15
2.1	CONCEITOS.....	15
2.1.1	Requisitos funcionais.....	16
2.1.2	Requisitos não funcionais.....	17
2.1.3	Requisitos de domínio	21
2.2	ENGENHARIA DE REQUISITOS.....	22
3	TÉCNICAS PARA COLETAR REQUISITOS.....	26
3.1	ENTREVISTAS	28
3.1.1	Descrição	28
3.1.2	Processo	32
3.1.3	Vantagens e pontos fortes.....	36
3.1.4	Desvantagens	36
3.1.5	Melhores práticas.....	37
3.2	CRIATIVAS DE GRUPO	41
3.2.1	BRAINSTORMING.....	42
3.2.1.1	Descrição	42
3.2.1.2	Processo	42
3.2.1.3	Vantagens e pontos fortes.....	44
3.2.1.4	Desvantagens	44
3.2.1.5	Melhores práticas.....	45
3.2.2	JAD (Joint Application Development).....	47
3.2.2.1	Descrição	47
3.2.2.2	Processo	50
3.2.2.3	Vantagens e pontos fortes.....	54
3.2.2.4	Desvantagens	56
3.2.2.5	Melhores práticas.....	58
3.3	PROTOTIPAÇÃO.....	60

3.3.1	Descrição	60
3.3.2	Processo	62
3.3.3	Vantagens e pontos fortes.....	63
3.3.4	Desvantagens.....	64
3.3.5	Melhores práticas.....	66
3.4	OBSERVAÇÃO	68
3.4.1	Descrição	68
3.4.2	Processo	69
3.4.3	Vantagens e pontos fortes.....	71
3.4.4	Desvantagens.....	72
3.4.5	Melhores práticas.....	72
4	CONSIDERAÇÕES FINAIS.....	75
4.1	CONCLUSÕES	75
4.2	FUTURAS PESQUISAS.....	75
	REFERÊNCIAS	77

1 INTRODUÇÃO

"Um projeto bem-sucedido é aquele que é realizado conforme o planejado."
(Vargas, Ricardo Viana 2005, p.15).

Em projetos de desenvolvimentos de sistemas várias etapas precisam ser seguidas e muitas são as dificuldades podendo causar o fracasso do projeto. Quando este projeto realmente termina sem cumprir seu objetivo final e a equipe avalia quais foram os pontos falhos, quase sempre a má definição do escopo ou a compreensão errada de seus requisitos são listadas como principais motivos de insucesso.

O problema de falta de entendimento e má definição no escopo acontecem, geralmente, no início dos trabalhos. Porém, causarão um grande impacto durante sua execução ou somente no final do projeto, que muitas vezes será tarde demais, comprometendo todo o projeto ou onerando muito mais tempo e custo que o planejado.

Segundo Sommerville (2003), durante todo o ciclo de desenvolvimento de sistema, as empresas enfrentam problemas das mais diversas naturezas. Falhas com a comunicação e dificuldades com a identificação das reais necessidades do cliente são as principais e que estão relacionados a aspectos da Engenharia de Requisitos. Os problemas nesta fase são mais caros de corrigir do que nas fases subsequentes do desenvolvimento de *software*, tendo influência direta no atraso do cronograma do projeto. Em algumas situações, é muito comum, as dificuldades terem origem nas questões humanas e sociais inerentes ao processo, em outros momentos em problemas relacionados à própria empresa. O fato é que tais dificuldades persistem no cotidiano das empresas e necessitam de uma análise detalhada com a finalidade de serem gerenciados e minimizados.

Esse trabalho apresentará as principais técnicas e ferramentas utilizadas para um melhor entendimento dos requisitos assim como as principais características de cada uma delas, auxiliando os profissionais da área de desenvolvimento de *software* a coletar com mais riqueza e confiabilidade os requisitos com os *stakeholders*¹.

¹ *Stakeholders* – Termo utilizado para referir as pessoas, grupos de pessoas, como clientes, patrocinadores, organizações executoras e o público, que estejam ativamente envolvidas no projeto ou cujos interesses possam ser afetados de forma positiva ou negativa pela execução ou término do projeto. (PMBOK®, 2008).

Em todas as empresas, sejam elas de pequeno, médio ou grande porte, privadas ou governamentais existem um ou mais projetos sendo desenvolvidos. Algumas empresas têm como foco de seu negócio desenvolver projetos, outras utilizam projetos como diferencial competitivo ou até mesmo utilizam para demonstrar e enaltecer suas estratégias. Mesmo com benefícios gerados pelos projetos, a grande quantidade de profissionais envolvidos e altos investimentos na área, a grande maioria desses projetos falha, atrasam ou são cancelados antes do término. A Gráfico 1 demonstra graficamente uma pesquisa de como os projetos estão sendo finalizados desde 1994.

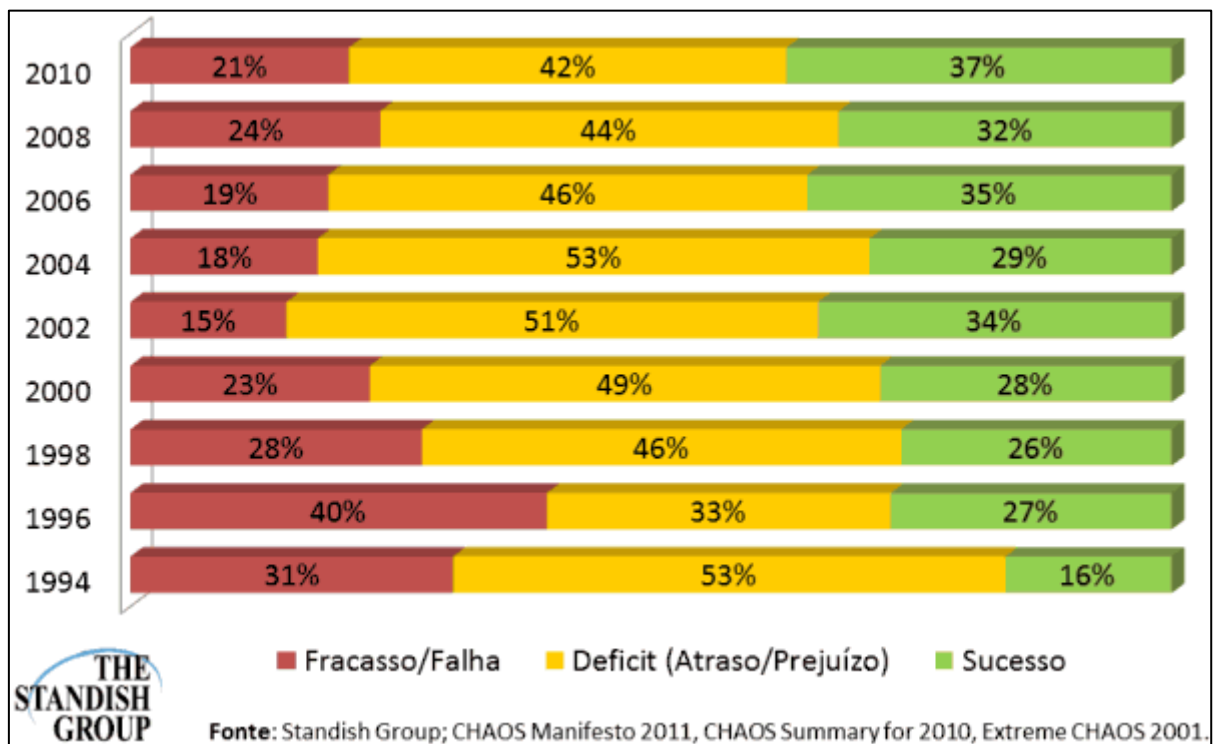


Gráfico 1: Problemas com projetos.

Fonte: Standish Group, CHAOS Manifesto 2011, CHAOS Summary 2010, Extreme CHAOS 2001.

Para Vargas (2005), muitas falhas são provenientes de causas externas, fora do controle da equipe. São elas as seguintes:

- Alterações na estrutura organizacional;
- Riscos incontroláveis no meio ambiente;
- Mudanças imprevisíveis na tecnologia disponível;
- Alterações imprevistas no prazo e preços;

- Cenário político e econômico.

Porém a grande maioria das falhas em projetos é proveniente de causas internas a organização, são as falhas gerenciais, que podem ser totalmente previstas e evitadas. Sendo as principais delas:

- Metas e objetivos mal ou não comprometidos;
- Falta de compreensão da complexidade do projeto;
- Grande quantidade de atividade e prazo incompatível;
- Estimativas financeiras pobres e incompletas;
- Projeto é baseado em dados insuficientes, ou inadequados;
- Controle inadequado;
- Falta ou excesso de gerentes;
- Expectativas cliente/equipe conflitantes ou opostas;
- Falta de conhecimento dos pontos-chave do projeto;
- Falta de liderança.

Segundo Furlan (2012, p.3), "Diversos autores e profissionais da área de Gestão de projetos atestam que a falta de uma definição de projetos e de escopo adequados são duas das principais razões de fracassos em projetos."

Principalmente em desenvolvimento de *software* o maior problema enfrentado é definição e a compreensão correta dos requisitos por completo, tanto pela equipe de desenvolvedores quanto por seus usuários. Para Pressman (1995, não paginado),

uma compreensão completa dos requisitos de *software* é fundamental para um projeto bem-sucedido. Um problema mal analisado e especificado desapontará o usuário e trará aborrecimentos ao desenvolvedor.

Sendo assim, a comunicação é fundamental no levantamento de requisitos, porém, estabelecer e manter uma comunicação efetiva para o total entendimento do escopo é cheia de

obstáculos, contudo, segundo Kendall (2010) para auxiliar nesse levantamento, existem várias técnicas que podem ser utilizadas. A proposta desta monografia é demonstrar as características e a utilização das mesmas. Técnicas bem aplicadas podem obter um melhor entendimento dos problemas resultando em soluções bem estruturadas, o que aumenta a chance de sucesso ao final do projeto.

Segundo o guia PMBOK® (2008), gerenciamento de escopo são "Os processos necessários para assegurar que o projeto inclui todo o trabalho necessário, e apenas o necessário, para terminar o projeto com sucesso.". Essas técnicas que serão apresentadas poderão evitar que esforços e recursos sejam desperdiçados em atividades que não resolverão os problemas dos patrocinadores/clientes.

Serão demonstradas diversas técnicas, porém uma análise deve ser feita para avaliar qual melhor se aplica em determinada situação, pois os *stakeholders*, o ambiente, a disponibilidade, recursos e localização física dos detentores da informação podem ser diferentes para cada projeto.

As técnicas apresentadas irão contribuir muito para o levantamento de requisitos de uma forma mais correta e satisfatória, porém, a utilização de uma técnica específica isoladamente pode não ser suficiente para uma definição satisfatório do escopo.

1.1 OBJETIVO

Tendo como base as definições da Engenharia de Requisitos e seus princípios, este trabalho tem como objetivo geral demonstrar uma integração de algumas técnicas de levantamento de requisitos ao processo de análise de requisitos em projetos de desenvolvimento de *software*.

Como objetivos específicos deste estudo estão:

- Apresentar e explorar o conhecimento sobre levantamento de requisitos na literatura;
- Analisar as técnicas de levantamento de requisitos existentes;
- Buscar as melhores práticas para estas técnicas.

1.2 METODOLOGIA

1.2.1 Tipo de pesquisa

Para que seja possível classificar o tipo de pesquisa utilizado, é necessário considerar o enfoque abordado, os interesses, as áreas, condições, objetivos, as metodologias entre outros. É de responsabilidade do pesquisador, adotar qual método melhor se adéqua a sua pesquisa.

O método de pesquisa utilizado neste trabalho é a pesquisa exploratória, pois visa propor um novo ou, a formalização de uma melhora do processo de levantamento de requisitos, com base nos padrões existentes. A pesquisa baseia-se em referências bibliográficas, trabalhos acadêmicos, teses e dissertações. Foram referenciados livros, artigos, sites de internet e documentos contendo padrões existentes e técnicas de levantamento de requisitos.

A pesquisa exploratória tem como objetivo delimitar o tema a ser estudado, buscar as informações sobre o assunto, determinar os objetivos, definir as possibilidades e avaliar a viabilidade da realização da pesquisa.

A utilização da pesquisa bibliográfica foi escolhida para este trabalho com o objetivo de rever a bibliografia existente sobre o tema. Inicialmente, foi feita a revisão bibliográfica sobre os aspectos do levantamento de requisitos e, posteriormente, foram estudadas algumas técnicas de levantamento de requisitos.

1.2.2 Procedimento metodológico

O estudo realizado seguiu os procedimentos metodológicos a seguir. Inicialmente, foi realizada uma pesquisa em dissertações, teses, documentos e livros das bibliotecas de Engenharia e Administração da Universidade Presbiteriana Mackenzie sobre engenharia de software, Engenharia de Requisitos e técnicas utilizadas para coletar requisitos.

Após reunir os materiais citados acima, foi realizado um estudo aprofundado sobre levantamento de requisitos, assim como as técnicas mais utilizadas e metodologias específicas. Em seguida, foi feita uma apresentação das técnicas focando nas suas características e melhores práticas.

2 REQUISITOS DE SISTEMAS

2.1 CONCEITOS

A definição de **requisitos** é muito ampla, porém, a grande maioria converge em um conceito comum: Levantamento de requisitos é um processo de descoberta das necessidades baseado na comunicação entre a necessidade conceitual para o total entendimento do problema e o real desenvolvimento de ideias para realizá-los, ou ainda, é o “Processo de documentar as funções e funcionalidades do projeto e do produto necessárias para atender às necessidades e expectativas das partes interessadas.” (PMBOK®, 2008, p. 93). Segundo Sommerville (2003), “Requisito é uma declaração abstrata, de alto nível, de uma função que o sistema deve fornecer ou de uma restrição do sistema”, ou, “Requisito é uma definição detalhada, matematicamente formal, de uma função do sistema.” (DAVIS, 1987).

Esta disciplina, de **requisitos**, reúne as atividades necessárias com a finalidade de entender o enunciado completo, claro e objetivo dos requisitos de um *software*. Para Pádua (2011), as seguintes definições de requisitos são aplicáveis,

Condição ou potencialidade de que um usuário necessita para resolver um problema ou atingir um objetivo.

Condição ou potencialidade que um sistema, componente ou produto deve possuir para que seja aceito (isto é, satisfaça a um contrato, padrão, especificação ou outro documento formalmente imposto).

Expressão documentada dessa característica.

Estes requisitos, segundo Pádua (2011, p.164), deve ser levantado pela equipe de projetos, em conjunto com usuários, representantes do cliente e especialistas na área. O conjunto de técnicas utilizadas para o levantamento, detalhamento, documentação e validação dos requisitos deste *software* forma a disciplina Engenharia de Requisitos, simplesmente chamada de Requisitos.

Desenvolvedor, analista de sistemas e cliente possuem um papel fundamental na análise e especificação desses requisitos. O cliente apresenta funcionalidades e características do *software*, muitas vezes obscuro e pobre em detalhes. O desenvolvedor e analista de sistemas age como um pesquisador, consultor e solucionador de problemas. A análise e especificação dos requisitos, a princípio, parecem uma atividade simples, porém, as

aparências enganam. A quantidade de informação é muito grande. As chances de um entendimento ambíguo, interpretação errônea e informações falsas são bem prováveis.

Requisitos são considerados de alta qualidade, segundo Pádua (2011, p. 165), quando “[...] são claros, completos, sem ambiguidade, implementáveis, consistentes e testáveis.”. Requisitos que não possuem estas características são considerados problemáticos, possuindo um alto risco de insucesso, devendo ser revisto e reformulados com os *stakeholders*. O sucesso no desenvolvimento de *software* é diretamente influenciado pela atenção dada em seu levantamento e dependente de uma completa análise dos requisitos de *software*.

Para Sommerville (2003), os requisitos de *software* são classificados como funcionais, não funcionais ou como requisitos de domínio:

2.1.1 Requisitos funcionais

Determina as funcionalidades que o *software* deve satisfazer, ou seja, descreve como ele deve reagir a determinadas entradas.

Exemplo: “O *software* deve calcular as comissões diárias, semanais e anuais de cada vendedor.”.

Os requisitos funcionais podem ser descritos de várias maneiras. No exemplo abaixo, estão vários requisitos funcionais do sistema bibliotecário de uma universidade Kotonya (1998 apud SOMMERVILLE, 2003) para estudantes possam pedir livros e documentos de outra universidade:

- i. O estudante deverá ser capaz de buscar todo o conteúdo inicial de banco de dados ou selecionar um subconjunto a partir dele;
- ii. O sistema deverá possuir telas apropriadas para o estudante ler documentos no repositório de documentos;
- iii. Cada solicitação deverá ser relacionada a um único identificador (ORDER_ID), que o estudante poderá copiar para a área de armazenagem permanente da conta.

Esses requisitos funcionais de usuário determinam recursos específicos que devem ser atendidos pelo sistema. Comparando os requisitos i e iii, verifica-se que os requisitos funcionais podem ser discriminados em diferentes níveis de detalhamento.

Muitos problemas no desenvolvimento de sistemas possuem origem na imprecisão da especificação de requisitos. É muito comum o analista de sistemas interpretarem o requisito de forma ambígua com o objetivo de simplificar seu desenvolvimento. Diversas vezes, porém, não é o que o cliente estava esperando, resultando em um atraso nas entregas e aumento de custos. Como exemplo, considere o requisito 2 do exemplo acima, que refere a ‘telas apropriadas’. O sistema pode fornecer o conteúdo dos documentos em diversos formatos. Contudo, ele está escrito de maneira ambígua, uma vez que não deixa claro quais são os formatos de telas que devem ser utilizados. Um analista que está sob pressão ou com o desenvolvimento do *software* atrasado pode simplesmente fornecer uma tela de texto argumentando que o requisito foi atendido.

Para Sommerville (2003), a especificação de requisitos funcionais de um sistema, inicialmente, deve ser completa e consistente. Sendo que completeza significa que todas as funcionalidades solicitadas pelo usuário devem estar definidas. A consistência significa que o sistema não deve possuir definições que se contradizem. Para sistemas grandes e complexos, torna-se quase impossível atingir essa consistência e completeza nos requisitos. Isso ocorre, em parte, por causa da própria complexidade do sistema e, em parte, porque pontos de vista diferentes apresentam necessidades inconsistentes. Essas inconsistências podem não ser claras quando os requisitos são especificados inicialmente. Porém, após uma análise mais profunda, os problemas começam a aparecer. À medida que os problemas surgem durante a revisão ou em fases posteriores, os problemas no documento de requisitos precisam ser corrigidos.

2.1.2 Requisitos não funcionais

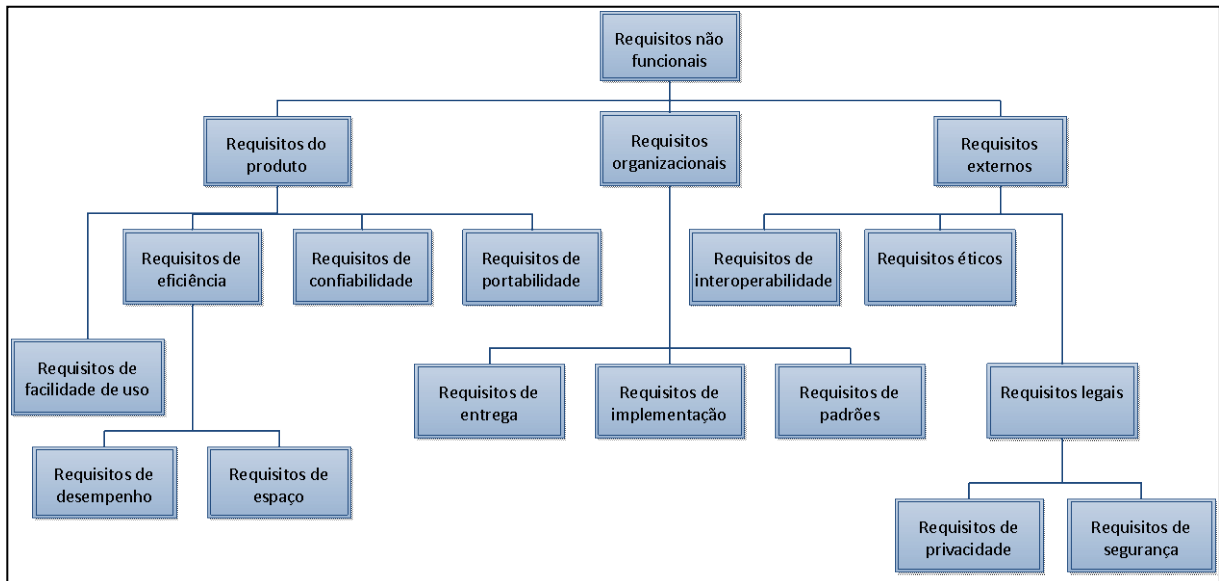
São requisitos que não descrevem diretamente às características específicas do sistema, são funções globais de um *software*, como usabilidade, desempenho, confiabilidade, custo, manutenção.

Exemplo: “O tempo de resposta do sistema não deve ultrapassar 15 segundos.”.

Muitas vezes, os requisitos não funcionais são mais importantes do que os requisitos funcionais individuais, pois muitos deles tratam questões que dizem respeito ao sistema como um todo, e não características individuais do sistema. Essa importância é dada pelo fato de que, se existir uma falha em cumprir um determinado requisito funcional individual, o sistema se degradará. Já a falha em cumprir com um requisito não funcional do sistema pode torná-lo inútil.

Contudo, os requisitos não funcionais não estão sempre relacionados ao *software* a ser desenvolvido. Alguns requisitos não funcionais podem, por exemplo, definir quais os processos, padrões e ferramentas devem ser utilizados para o desenvolvimento do *software* e até mesmo restringir o uso de uma determinada metodologia de desenvolvimento. (SOMMERVILLE, 2003 p.85)

A lista de requisitos não funcionais surge conforme a necessidade dos usuários ou do sistema, para atender restrições orçamentárias, legislação, política organizacional, limitação de hardware, integração com outros sistemas. O Fluxograma 2 classifica os diferentes tipos de requisitos não funcionais que podem ser relacionados.



Fluxograma 2: Tipos de requisitos não funcionais.

Fonte: Sommerville, 2003.

Os requisitos não funcionais demonstrados no Fluxograma 2, para Sommerville (2003, p.86), são classificados conforme a procedência:

- **Requisitos de produto:** São os requisitos que definem o funcionamento do produto. Por exemplo, requisitos de quanto de espaço em disco ele necessita, qual o tempo de resposta, taxa aceitável de falhas. Este requisito serve de apoio ao desenvolvimento do *software*;
- **Requisitos organizacionais:** São relacionados às políticas e procedimentos das organizações do cliente e do desenvolvedor do sistema. Como exemplos, estão os processos e padrões que devem ser utilizados, “[...] os requisitos de implementação, como a linguagem de programação ou método de projeto, os requisitos de fornecimento, que especificam quando o produto e seus documentos devem ser entregues.” (SOMMERVILLE, 2003, p.86);
- **Requisitos externos:** Abrange todos os requisitos com origem em fatores externo ao sistema. Entre os exemplos estão os requisitos que definem a integração do sistema com sistemas de outras empresas, os requisitos de obrigações legais e os requisitos éticos. Este ultimo, serve para assegurar que o sistema seja aceito pelos usuários.

Um problema comum nos requisitos não funcionais é que eles são, muitas vezes, de difícil verificação, pois são definidos por objetivos gerais, aberto à interpretação ambígua. Para isso, os requisitos não funcionais devem ser definidos de uma forma quantitativa,

utilizando métricas que possam ser objetivamente mensuradas e testadas. O Quadro 3 demonstra algumas métricas que podem ser utilizadas para especificar as propriedades do sistema, podendo ser utilizadas nos teste para verificar se os requisitos estão sendo cumpridos ou não.

Propriedade	Métrica
Velocidade	Transações processadas/segundo Tempo de resposta ao usuário/evento Tempo de <i>refresh</i> da tela
Tamanho	K <i>bytes</i> Quantidade de RAM
Facilidade de uso	Tempo de treinamento Número de frames de ajuda
Confiabilidade	Tempo médio para falhar Probabilidade de indisponibilidade Taxa de ocorrência de falhas Disponibilidade
Robustez	Tempo de reinício depois de uma falha Porcentagem de eventos que causam falhas Probabilidade de que dados sejam corrompidos por falhas
Portabilidade	Porcentagem de declarações dependentes de sistemas-alvo Número de sistemas-alvo

Quadro 3: Exemplo de métricas não funcionais.
Fonte: Sommerville, 2003.

A especificação quantitativa, mesmo sendo muito importante para traduzir os requisitos, não é fácil. Muitas vezes o cliente não consegue especificar as metas em requisitos quantitativos, e algumas, não possuem métricas para serem utilizadas, por exemplo, metas de facilidade de manutenção. Portanto, nos documentos de requisitos, as declarações de metas estão misturadas com os requisitos. Mesmo assim, estas métricas podem ser úteis para os desenvolvedores entenderem um pouco mais quais são as prioridades do cliente.

Para Sommerville (2003), os requisitos funcionais e não funcionais devem ser diferenciados em um documento, porém na prática, isso não é muito fácil. Se os requisitos forem definidos separadamente, será muito mais difícil entender quais são os relacionamentos entre eles. O ideal é encontrar um equilíbrio e isso depende do tipo de sistema que está sendo desenvolvido. Porém, os requisitos relacionados as características fundamentais do sistema, devem ser explicitamente destacados. Isso pode ser feito destacando-os em seções diferentes no documento de requisitos ou separá-los, de alguma forma, dos outros requisitos de sistema.

2.1.3 Requisitos de domínio

São requisitos que se originam do domínio da aplicação do sistema e que refletem características desse domínio. Podem ser novos requisitos funcionais, restringir um requisito funcional existente ou definir especificamente como deve funcionar esse requisito. Os requisitos de domínio são importantes por refletirem fundamentos do domínio da aplicação. Quando esses requisitos de domínio não são obedecidos, podem tornar o sistema impossível de funcionar. Para exemplificar a especificação de como o cálculo é realizado, definido pelos requisitos de domínio, Sommerville (2003), utiliza o Quadro 4, que é baseado na especificação de um sistema automatizado de controle de proteção de trens.

A desaceleração do trem será computada como:

$$D_{trem} = D_{controle} + D_{gradiente}$$

onde $D_{gradiente}$ é $9,81 \text{ ms}^2 * \text{gradiente compensado/alfa}$ e onde os valores de $9,81 \text{ ms}^2/\text{alfa}$ são conhecidos para diferentes tipos de trens.

Quadro 4: Um requisito de domínio de um sistema de trens.

Fonte: Sommerville, 2003.

“Esse sistema parará automaticamente um trem se ele atravessar um sinal vermelho. Esse requisito define como a desaceleração do trem é calculada pelo sistema e utiliza terminologia específica do domínio.” (SOMMERVILLE, 2003).

Esse exemplo demonstra o problema principal com os requisitos de domínio. Por serem expressos com uma linguagem específica do domínio da aplicação, pode dificultar o entendimento dessa linguagem por parte dos analistas de sistemas. O especialista de um domínio pode não passar todas as informações para o analista, simplesmente por achar que as informações são muito óbvias. Porém, as informações, são óbvias somente para o especialista de um domínio, para os desenvolvedores e analistas, essa informação além de nova, pode ser fundamental para o funcionamento correto do sistema.

Na prática, a classificação entre os tipos de requisitos não são tão clara como sugerem as definições acima. Um requisito de usuário relacionado à proteção, por exemplo, possui a princípio, características de um requisito não funcional. Porém, quando analisado mais detalhadamente, pode levar a outros requisitos que são classificados com funcionais, como a necessidade de incluir recursos de autenticação de usuários no sistema. Portanto, embora seja

importante classificar os requisitos dessa maneira, os analistas devem lembrar que essa é, na realidade, uma distinção artificial.

2.2 ENGENHARIA DE REQUISITOS

Engenharia de Requisitos, segundo Sommerville (2003, p. 103), “é um processo que envolve todas as atividades exigidas para criar e manter o documento de requisitos de sistema.”. Isto é, os requisitos definem o que o sistema deve fazer e as circunstâncias sob as quais deve operar. Possibilita a definição das funções e o desempenho de *software*, indica a interface do *software* com outros elementos do sistema e estabelecem quais são as restrições de projeto que o *software* deve possuir. De forma geral, a Engenharia de Requisitos é o processo de identificar todos os envolvidos, descobrir seus objetivos e necessidades e registrá-los de forma adequada para análise, comunicação e desenvolvimento.

É papel da Engenharia de Requisitos é propor métodos, ferramentas e técnicas que facilitem o levantamento da definição do que se quer de um *software*. O desenvolvimento da fase de definição de requisitos é necessário porque a complexidade dos sistemas atuais exige um correto entendimento do problema antes mesmo da compreensão de uma solução. Para que a definição dos requisitos seja mais eficaz, o engenheiro de *software* necessita entender o ambiente no qual o *software* irá atuar e escolher o modelo e método que melhor representam o ambiente.

Segundo Pressman (1995), a Engenharia de Requisitos fornece mecanismos adequados para o cliente entender as necessidades, avaliar a possibilidade de ser realizado, 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. O processo de Engenharia de Requisitos pode ser descrito em cinco passos distintos: i) levantamento 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.

Como definido no início desta subseção 2.2, o processo de descobrir, analisar, documentar e validar requisitos é chamado de processo de Engenharia de Requisitos. Para Sommerville (2003), são:

- **Estudo de viabilidade:** É onde o processo de Engenharia de Requisitos deve começar para os novos sistemas. É um estudo breve e direcionado que avaliará se o sistema contribui com os objetivos da empresa, a tecnologia aplicada e a possível integração com outros sistemas legados. O estudo de viabilidade envolve avaliar e coletar informações tendo como resultado, um relatório que decidirá se vale à pena realizar o processo de levantamento de requisitos e o desenvolvimento do sistema, ou seja, se do ponto de vista comercial o sistema é viável;
- **Levantamento e análise de requisitos:** É a próxima atividade após o estudo de viabilidade. Nesta etapa, os membros da equipe técnica de desenvolvimento de *software* trabalham com os *stakeholders* para obter mais informações sobre o domínio da aplicação, quais serviços o sistema deve fornecer, o desempenho esperado do sistema, as restrições de hardware e outras informações que afetam o funcionamento do sistema. O levantamento e a análise fazem parte de um processo difícil, pois: i) os *stakeholders* muitas vezes não sabem o que realmente querem do sistema, fazem pedidos que muitas vezes são impossíveis de serem desenvolvidos por possuírem um custo muito alto; ii) passam as informações para as analistas utilizando os próprios termos, tornando o entendimento por parte do engenheiro de *software* muito mais difícil; iii) e diferentes *stakeholders* têm uma visão diferente das funcionalidades que o sistema deve possuir, obrigando que o analista descubra todas as fontes e encontre os pontos em comuns assim como os conflitos. Existem ainda outros fatores que dificultam a análise e não estão ligados ao entendimento ou as informações dadas pelos *stakeholders*. São os fatores políticos que podem influenciar nos requisitos do sistema e o ambiente econômico que a análise de requisitos está atuando;
- **Especificação de requisitos:** Essa tarefa consiste em traduzir as informações coletadas durante a atividade de análise de requisitos e documentar definindo um conjunto de requisitos. É feito o detalhamento das características do sistema que o usuário considera como requisito. Nesta tarefa, podem-se utilizar algumas técnicas como: o levantamento orientado a ponto de vista, os cenários, etnografia e a prototipação. Segundo, Sommerville (2003), não existe uma abordagem perfeita e totalmente aplicável para a análise de requisitos. É preciso utilizar várias dessas técnicas para desenvolver um melhor entendimento e análise dos requisitos. Essas técnicas serão abordadas na próxima seção;

- **Validação de requisitos:** Nesta etapa, são mostrados os requisitos que definem o sistema desejado pelo cliente. Durante essa tarefa, é comum descobrir erros nos requisitos já documentados. Assim, esses requisitos devem ser revistos e modificados com o objetivo de corrigir esses erros. Essa validação é importante porque a surgimento de erros na documentação de requisitos pode levar a custos elevados relacionados ao retrabalho, quando esses erros são descobertos durante o desenvolvimento ou mais tarde quando o sistema já estiver implantado. O custo de fazer uma alteração no sistema, resultante de um problema de requisito, é muito maior do que corrigir erros de projeto ou de codificação.

A Figura 5 apresenta o processo de Engenharia de Requisitos segundo Sommerville (2003). Esse processo inicia 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 sinalize que o sistema é viável, o levantamento e análise de requisitos devem ser realizados (2), gerando o modelo do sistema (3). Com as informações obtidas na etapa anterior, inicia-se a especificação de requisitos (4), produzindo o documento com os requisitos do usuário e do sistema (5). Se necessário, 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). Caso haja necessidade, volta-se para a fase de especificação de requisitos (8). Ao término do processo, uma documentação de requisitos é produzida, ou seja, tem-se a especificação do sistema (9).

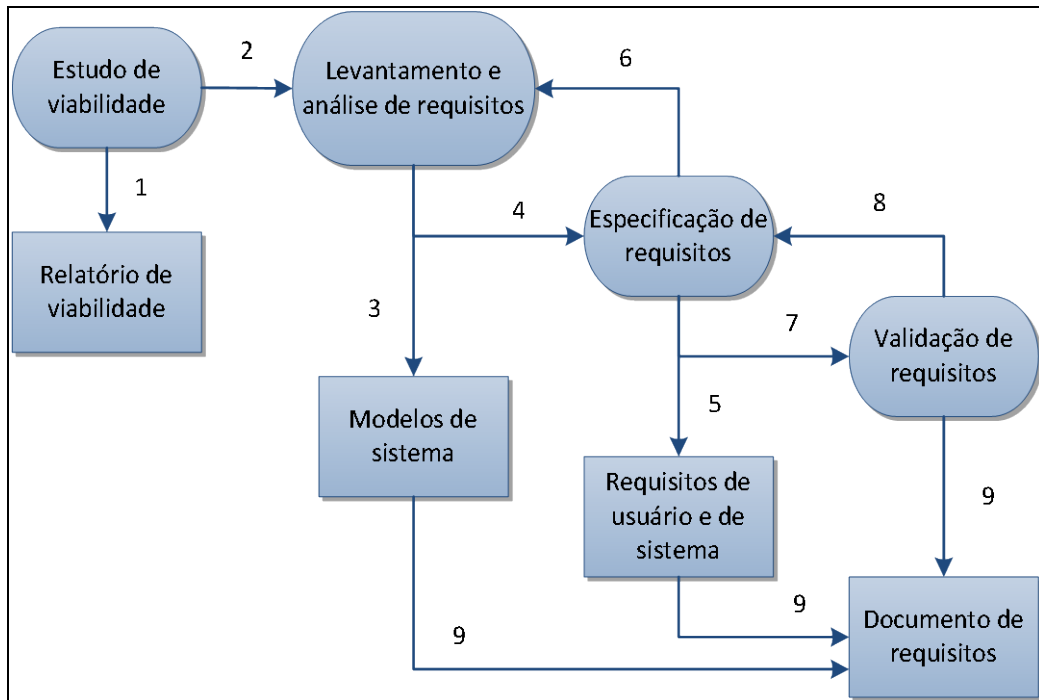


Figura 5: Processo de Engenharia de Requisitos.
Fonte: Sommerville, 2003.

Os requisitos usualmente são apresentados em dois níveis de detalhes. Os *stakeholders* necessitam de uma especificação de requisitos mais genérica, focado nas funcionalidades do sistema; os desenvolvedores precisam de uma especificação muito 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.

A Engenharia de Requisitos propõe um conjunto de métodos, ferramentas e técnicas que apoiam o processo de definição de requisitos. Em uma visão geral, o conjunto de técnicas de levantamento, documentação e análise de requisitos forma a Engenharia de Requisitos.

3 TÉCNICAS PARA COLETAR REQUISITOS

Para iniciar a coleta de requisitos com o cliente, é necessário que o analista entenda a sua necessidade, pois o que ele está solicitando pode ou não solucionar o problema. Por exemplo, um arquiteto, antes de iniciar a construção de uma casa, precisa saber a finalidade da casa junto ao cliente: será a casa onde o cliente irá morar, será a casa onde o cliente abrirá o seu negócio, qual será o negócio a ser oferecido nessa casa? Essa informação definirá qual é o objetivo do projeto ou sistema. Após essa informação, o arquiteto precisará se aprofundar nos detalhes, como por exemplo: qual o tamanho da casa, quantos cômodos, quantos ambientes são necessários, etc. As respostas servirão como base para que se elabore o plano da casa conforme o cliente deseja. Por último, informações ainda mais específicas irão ajudar na criação de uma casa ideal para determinado cliente, por exemplo, se ele possui filhos, qual a idade deles, ou ainda, qual o segmento de mercado o restaurante pretende atender, etc. Com essas informações, o arquiteto conseguirá construir a casa que irá atender exatamente às necessidades de seu cliente. (PÁDUA, 2011)

Com o objetivo de obter essas informações da melhor forma possível, ou seja, sem ambiguidade e rico em detalhes relevantes as necessidades do cliente, são aplicadas e desenvolvidas as técnicas de levantamento de requisitos. Precisamos conhecê-las no detalhe para avaliar qual técnica será mais eficaz para determinado grupo de *stakeholders* e sistema desejado.

As técnicas são um ponto fundamental no levantamento de requisitos, e fazem parte do processo de coletar requisitos definido no PMBOK® (2008), conforme

Figura 6.

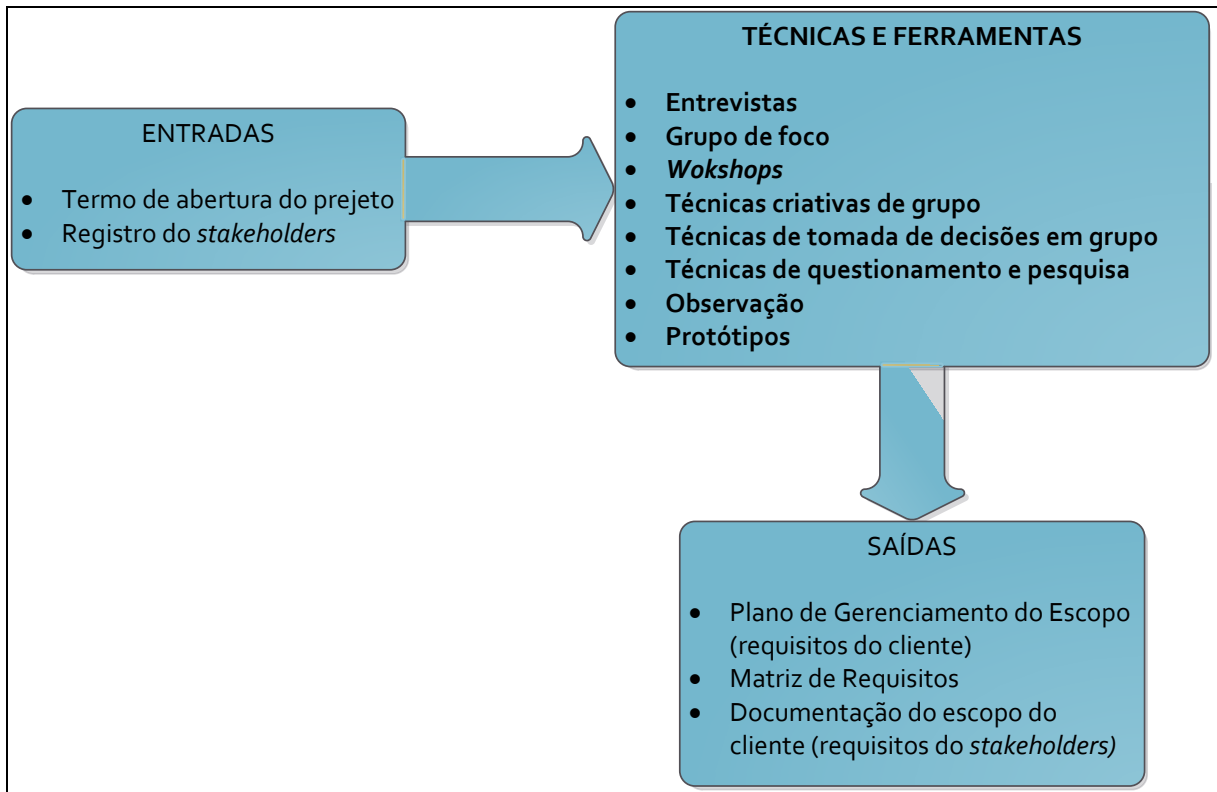


Figura 6: Os componentes do processo de coletar requisitos.

Fonte: PMBOK®, 2008.

Esta seção apresenta as técnicas que são aplicáveis no levantamento de requisitos. A aplicação dessas técnicas contribui para que os requisitos sejam levantados em menor tempo, com a participação ativa dos *stakeholders* e de uma forma mais concreta, diminuindo a ambiguidade de informações.

3.1 ENTREVISTAS

3.1.1 Descrição

A entrevista é uma das técnicas mais utilizadas para levantar os requisitos junto ao cliente. “Uma entrevista é uma conversa direcionada com um propósito específico, que utiliza um formato pergunta-resposta” (KENDALL e KENDALL, 2010). O PMBOK® (2008) define entrevista como “[...] um meio formal ou informal de se descobrir informações das partes interessadas através de conversas diretas com as mesmas.”. Ao entrevistar os principais envolvidos no projeto, os *stakeholders* e os especialistas na área que o sistema irá atuar o processo de identificar e definir os requisitos exigidos fluirá de uma forma muito mais rápida e confiável. Em sua maioria, são conduzidas individualmente, porém podem ser conduzidas por múltiplos entrevistadores e/ou entrevistados.

Para Carvalho (2001), entrevistar não é somente elaborar perguntas; é uma técnica estruturada que pode ser aprendida e na qual os analistas podem conquistar o conhecimento com treinamento e prática. Um dos principais aspectos dessa técnica é garantir que as ideias e as predisposições do entrevistador não interfiram nas reais necessidades e informações obtidas junto ao entrevistado. Isto pode ser algo difícil de obter, uma vez que cada analista sofre influência de um conjunto de vivências pessoais, que podem atrapalhar a compreensão de ideias vindas do entrevistado. É justamente por esse o motivo que a entrevista não é uma técnica tão simples como inicialmente pode aparentar.

Ao realizar a entrevista, o entrevistador deverá definir os objetivos, delimitando o que perguntar, não deixando de lado a formulação das perguntas e o tipo, sendo elas objetiva, subjetiva e aprofundamento. Outro aspecto relevante é verificar se o entrevistado é a pessoa apropriada a responder os esses questionamentos.

Conforme Braga (2008), a formulação das perguntas deve ser feita de acordo com o objetivo esperado. Podendo ser:

- **Questões subjetivas:** quando o entrevistador quer criar um ambiente mais espontâneo, obter maior riqueza de detalhes ou questionamentos mais abrangentes, porém, a utilização desta técnica pode fornecer detalhes irrelevantes e o descontrole da

entrevista, podendo passar a impressão que o entrevistador está sem foco, sem objetivo. Permite respostas mais abertas e podem gerar novos questionamentos.

Exemplos: O que você acha desse processo?, Explique como você vê esse processo?;

- **Questões objetivas:** quando se quer limitar as respostas possíveis, manter o controle da entrevista e ganhar tempo, uma vez que vão direto ao ponto em questão. Porém, pode tornar-se cansativo para o entrevistado, falhar na obtenção de detalhes relevantes e não constroem uma afinidade entre entrevistado e entrevistador.

Exemplos: Quem?, Quanto?, Quanto tempo?, Qual das seguintes informações?;

- **Questões de aprofundamento:** para um melhor detalhamento a respeito de um tópico. Contudo, a grande maioria dos autores defende que o mais importante é evitar perguntas que façam o entrevistado responder de forma específica e resumida ou perguntas que gerem respostas ambíguas.

Exemplos: Por quê?, Poderia exemplificar? Como o processo inicia?.

Ainda para Braga (2008), a utilização das perguntas acima descritas deve ser estruturada visando a melhor forma de conduzir a entrevista. Em situações onde o entrevistado não consegue abordar um determinado assunto, é indicada a utilização da estrutura **Pirâmide**, pois, é iniciada com questões bastante detalhadas, geralmente objetivas, e à medida que a entrevista progride, questões mais gerais e subjetivas são colocadas. Também pode ser utilizada se o analista de sistemas deseja finalizar um assunto específico.

A estrutura **Funil** inicia-se com questões mais abrangentes e à medida que a reunião avança são utilizadas questões mais específicas. É aconselhada quando se deseja começar uma entrevista de uma forma menos ameaçadora para o entrevistado ou para iniciar uma bateria de entrevistas. Permite levantar uma grande quantidade de informação detalhada, evitando longas sequencias de questões objetivas e de aprofundamento.

Na estrutura **Diamante** a entrevista começa com questões específicas, passa por questões mais abrangentes e finaliza com o entrevistador novamente abordando questões específicas. É considerada a combinação dos dois tipos de estrutura (Pirâmide e Funil), frequentemente é a mais utilizada, já que mantém a atenção do entrevistado em uma variedade de questões. Porém, tende a ser mais longa.

Abaixo, as Figura 7, Figura 8, Figura 9 demonstram os três tipos de estruturas:

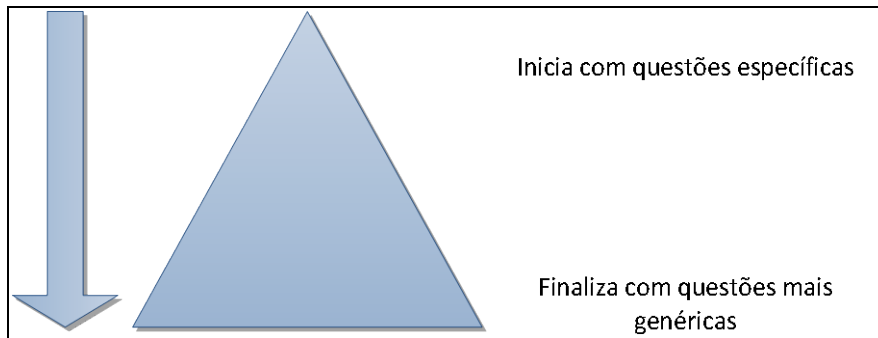


Figura 7: Estrutura Pirâmide de entrevista
Fonte: Falbo (2012).

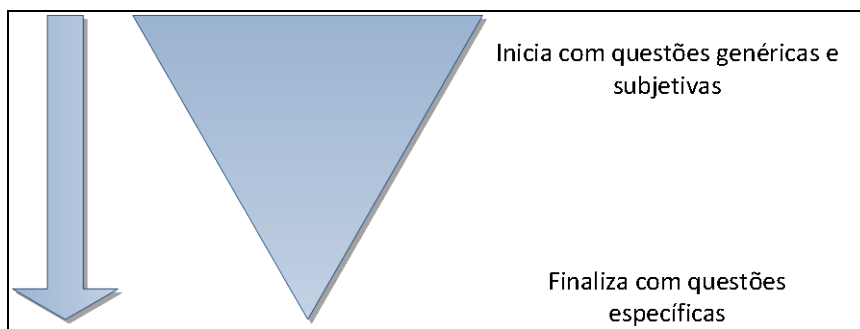


Figura 8: Estrutura Funil de entrevista
Fonte: Falbo (2012).

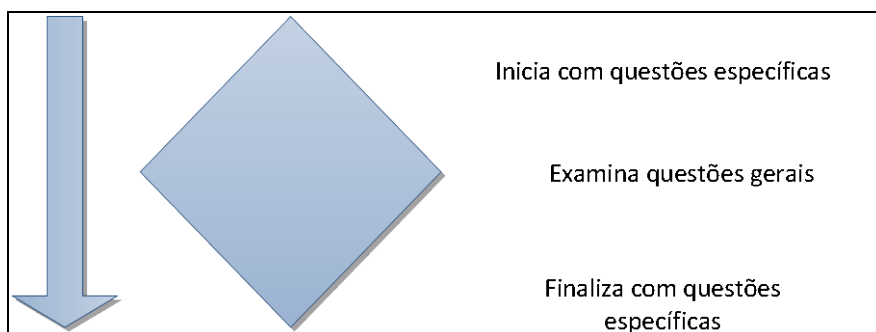


Figura 9: Estrutura Diamante de entrevista
Fonte: Falbo (2012).

Estruturar a entrevista é um meio de planejar, a princípio, a ordem das perguntas. Certamente, há a opção de não definir essa ordem com antecedência, porém as questões devem ser definidas e elaboradas antecipadamente, o que requer o planejamento.

Há basicamente, segundo MAGUIRE (1998 apud BATISTA 2003), três tipos de entrevistas: entrevistas não estruturadas, entrevistas semi-estruturadas e entrevistas estruturadas.

- **Entrevistas não estruturadas:** Este tipo de entrevista é indicado quando o entrevistador deseja manter a flexibilidade da entrevista para direcionar o questionamento no sentido que lhe parece mais apropriado, obtendo as informações contidas nas respostas ou nas conversas com os participantes. Com isso não é possível criar um conjunto de perguntas pré-definidas, ou seja, as questões são geradas de uma forma espontânea seguindo um fluxo natural de uma interação. Sua aparente simplicidade é devido ao fato de que entrevistar parece ser uma habilidade que as pessoas possuem devido à conversação social, ou seja, é a atividade básica e natural do ser humano de falar e comunica-se. É praticada em quase todas as áreas que o homem está presente;
- **Entrevistas semi-estruturadas:** Nas entrevistas semi-estruturadas existe um guia para o desenvolvimento da entrevista, onde é listado um conjunto previamente determinado de perguntas ou de itens que serão explorados no decorrer da entrevista. Este guia serve como uma lista de checagem, *checklist*, utilizada durante a entrevista e garante que a mesma informação seja extraída de pessoas diferentes. São consideradas semi-estruturadas, pois existe muita flexibilidade nas perguntas e a ordem e o mecanismo dessas questões não é definido com antecedência. Além disso, o entrevistador tem a liberdade para explorar determinadas perguntas com maior detalhe mesmo seguindo esse *checklist*;
- **Entrevistas estruturadas:** As entrevistas estruturadas são um conjunto de perguntas fechadas, pré-definidas e cuidadosamente elaboradas. As mesmas perguntas são feitas para cada entrevistado, mantendo, essencialmente, as mesmas frases e na mesma ordem. Esse tipo de entrevista é indicado quando existem muitos entrevistadores e deseja-se diminuir as variações das questões apresentadas. É apropriado também

quando se deseja ter a mesma informação de cada entrevistado em diferentes momentos ou quando o tempo para o levantamento dos requisitos não é suficiente.

Já para Sommerville (2003), existem apenas dois tipos de entrevista:

- **Entrevistas fechadas:** Onde o entrevistador prepara as perguntas seguindo um conjunto previamente definido de questões. Possui as características da entrevista estruturada definida acima;
- **Entrevistas abertas:** O entrevistador não elabora uma lista pré-definida, discutindo de forma aberta o que os *stakeholders* querem do sistema. Possui características semelhantes à entrevista não estruturada descrita acima.

As entrevistas geralmente são uma combinação desses dois tipos. Alguma resposta feita por uma pergunta previamente definida pode levar a outras questões que são discutidas de uma maneira menos estruturada. As entrevistas que são completamente abertas dificilmente são bem sucedidas. A grande maioria das entrevistas requer algumas perguntas que servem de ponto de partida e também para manter o foco em um ponto do sistema a ser desenvolvido (SOMMERVILLE, 2003).

3.1.2 Processo

A entrevista é uma forma sistemática de extrair informação de uma ou mais pessoas. Entrevistar é uma habilidade essencial para os analistas de sistemas, pois seu sucesso depende de uma habilidade para identificar: i) fluxo de trabalho; ii) fatores que afetam as operações do sistema; e iii) componentes que compõem os sistemas. (SOARES, 2007).

Existem basicamente quatro fases na entrevista segundo Batista (2003): identificação dos entrevistados, preparação, realização e análise das entrevistas.

- **Identificação dos entrevistados:** O levantamento dos requisitos inicia com a identificação dos *stakeholder* a ser entrevistado. O analista deve concentrar-se em selecionar as pessoas corretas com o intuito de diminuir a quantidade de entrevistas. É

aconselhável iniciar a entrevista pelos diretores, executivos ou *sponsor*² que oferecem uma visão global e, em seguida, os usuários podem fornecer informações mais detalhadas sobre o ambiente e o sistema novo ou atual. Com o decorrer das entrevistas o analista descobrirá se é necessário entrevistar mais pessoas, tanto novos usuários quanto pessoas que não irão interagir diretamente com o novo produto, mas que poderão de alguma forma, interagir com o novo sistema. Também é muito útil estudar os diferentes perfis dos entrevistados, buscando pontos comuns e divergentes com o entrevistador. Essa atitude pode ajudar a quebrar algumas barreiras no início da entrevista;

- **Preparação da entrevista:** Existem algumas atividades iniciais no preparo de uma entrevista: estudar o campo do problema, definir objetivos e a lista de perguntas e planejar e marcar os encontros. Conhecer o domínio do problema, conceitos e termos utilizados pelos usuários é essencial para compreender as necessidades deste setor. Isso fará com que os usuários tenham maior confiança de que o analista entendeu seus problemas. Para conseguir esse conhecimento sobre o domínio aconselha-se ler a documentação do sistema atual ou qualquer material relacionado ao novo sistema, como por exemplo, bibliografia. A documentação de projetos semelhantes dentro ou fora da organização e a conversa com outros analistas de sistemas também poderá auxiliar muito nesse entendimento. Todas essas ações irão guiar o entrevistador a preparar com antecedência uma lista de perguntas com muito mais qualidade e objetividade. Porém, não é possível preparar todas as perguntas com antecedência. As informações extraídas ao longo da entrevista servirão de base para a elaboração de novas questões, que serão formuladas à medida que a entrevista avançar. Para diminuir o tempo da entrevista devem-se apresentar claramente quais são os objetivos que se pretende atingir assim como a duração máxima da entrevista. Antes da entrevista pode-se enviar aos entrevistados uma lista de tópicos que serão tratados nos encontros, com isso as pessoas poderão ter tempo para se preparar, tornando as entrevistas muito mais proveitosas. Para marcar a data, hora, local e tempo da entrevista sempre se devem levar em consideração a agenda do entrevistado. O analista sempre deve

² Sponsor – A pessoa ou o grupo que fornece os recursos financeiros em dinheiro ou em espécie, para o projeto. (PMBOK®, 2008).

relembrar os entrevistados sobre o encontro, pelo menos um ou dois dias antes, aproveitando também para confirmar horário e local;

- **Realização e condução da entrevista:** A realização e condução da entrevista podem ser divididas em três etapas, segundo Batista (2003), a abertura, o desenvolvimento e encerramento da entrevista:

- **A fase de abertura:** É o aquecimento para a entrevista. As pessoas se apresentam e se cumprimentam, em seguida, o entrevistador expõe o motivo da entrevista, o que se espera conseguir, o formato das perguntas que serão apresentadas, como a informação será utilizada e outros detalhes relevantes para a entrevista. Se houver a necessidade de utilizar alguma fórmula matemática, financeira ou notação gráfica, o entrevistador deve explicá-las, se necessário, para que todos estejam alinhados com o mesmo conhecimento. É essencial causar uma boa impressão e transmitir segurança para os entrevistados;

- **O desenvolvimento:** É a principal atividade, quando o entrevistador estará explorando e fazendo as perguntas abertas sobre os tópicos, para que os entrevistados compreendam o alcance das respostas. Perguntas abertas não permitem respostas simples como “sim” ou “não”, incentivando maior comunicação e evitando a sensação de interrogatório. É muito importante nesta etapa que o entrevistador seja analítico e permaneça neutro. Quando possível é importante resumir o que já foi discutido para facilitar a compreensão. Questões genéricas incentivam respostas não reprimidas e podem gerar uma grande quantidade de informações. São muito úteis quando não se conhece o suficientemente o assunto para fazer questões mais detalhadas. Por outro lado, perguntas específicas são úteis quando se quer informar o usuário sobre um aspecto particular e incentivar uma resposta detalhada e precisa;

- **A fase de encerramento:** Pode-se encerrar uma entrevista quando as todas as perguntas forma respondidas, quando o tempo reservado, estipulado no início, tiver terminado ou quando o entrevistador sentir que os entrevistados estão cansados. Nesta etapa, também conhecida como etapa de relaxamento, pode-se fazer uma revisão para garantir que não há duplo entendimento nas

informações obtidas. As próximas ações são anotadas e o planejamento das próximas atividades é feito. O entrevistador deve agradecer a colaboração dos entrevistados, deixando aberta a possibilidade de um novo contato para esclarecer dúvidas que podem surgir ao analisar as informações obtidas ou ao realizar uma entrevista com outros usuários;

- **Análise da entrevista:** Finalizada a entrevista, a próxima atividade é ler as anotações, documentá-las, reorganizar as informações, compará-las com outras entrevistas ou fontes de informação e elaborar um resumo. Com essa análise podem-se descobrir informações conflitantes, ausentes ou ambíguas. Quando o resumo estiver pronto, pode-se enviá-lo aos entrevistados para que eles confirmem as informações. É muito importante reavaliar os procedimentos adotados para preparar e conduzir a entrevista, com o objetivo de melhorar a qualidade das entrevistas futuras.

Kendall (2010) sugere que o planejamento da entrevista siga os cinco passos abaixo:

- **Estudar material:** Esse passo tem a finalidade de otimizar o tempo das entrevistas, evitando perguntas básicas e superficiais. Para isso, o analista deve estudar previamente o material sobre o domínio e a organização. O entrevistador deve atentar-se à linguagem utilizada pelos usuários da empresa, procurando estabelecer um vocabulário comum que deve ser utilizado também nas questões apresentadas;
- **Definir objetivos:** De uma forma geral, o analista faz perguntas de algumas áreas, como formatos da informação, fontes de informação, frequência na tomada de decisão, estilo da tomada de decisão, etc.;
- **Selecionar os entrevistados:** É essencial chamar para a entrevista pessoas chave das áreas envolvidas no sistema. O próprio cliente pode auxiliar na seleção dessas pessoas;
- **Preparar o entrevistado:** A entrevista deve ser agendada com antecedência para que o entrevistado tenha tempo para se preparar. O entrevistador pode enviar o material para essas pessoas para auxiliá-los na preparação;

- **Preparar a entrevista:** O entrevistador decide, dentre outros, sobre os tipos de perguntas, a estrutura da entrevista, como será conduzida e o modo que a mesma será registrada.

3.1.3 Vantagens e pontos fortes

As entrevistas são úteis para obter informações detalhadas dos usuários e identificar possíveis áreas que podem ser analisada mais detalhadamente. Entrevistas são consideradas populares, bem conhecidas, muito bem aceitas e também úteis para descobrir eventos e processos que ocorrem esporadicamente. Resumindo, as duas principais vantagens da entrevista são i) é um processo basicamente simples de entender e aplicar; ii) é uma técnica muito bem aceita pelos usuários.

As questões abertas nas entrevistas são mais eficazes e permitem que o entrevistado expresse seu entendimento e sentimento sobre o assunto em pauta. Nas entrevistas não estruturadas, o ponto forte é que a entrevista é mais flexível e pode auxiliar os desenvolvedores a encontrar respostas para as diferenças em pontos específicos, para as alterações situacionais e para novas informações. O ponto forte nas entrevistas semi-estruturadas é que a utilização de um guia na entrevista possibilita entrevistar diferentes pessoas de uma maneira mais completa e sistemática, limitando os assuntos abordados na entrevista. A utilização de uma entrevista estruturada com questões fechadas possibilita ao entrevistador coletar sistematicamente informações detalhadas, facilitando a comparação das respostas de todos os entrevistados. (BATISTA, 2003).

3.1.4 Desvantagens

Segundo Kendall (2010), o processo de entrevistar pode tornar-se um processo longo e caro. As seguintes desvantagens são encontradas:

- Encontrar um horário na agenda das “pessoas-chave” para obter boas informações sobre o sistema pode uma tarefa muito difícil em algumas empresas;
- Geralmente muitos retornos e esclarecimentos de dúvidas são necessários, tornando o processo mais caro;

- Como não é possível entrevistar todas as pessoas de vários departamentos, devido ao tempo e custo, o risco de desvio de informação e parcialidade é alto;
- Mesmo com a elaboração de perguntas bem estruturadas e feitas corretamente para obter uma informação de qualidade, não existe um procedimento padrão para que a entrevista seja mais eficaz.

Na entrevista não estruturada podem ser gerados dados menos sistemáticos que são mais custosos para analisar e classificar. Uma das desvantagens da entrevista semi-estruturada é que não possibilita ao entrevistador desenvolver assuntos ou tópicos de interesse que surgem durante a entrevista, pois não foram relacionados quando o guia da entrevista foi criado. A flexibilidade do analista na elaboração e ordenação das questões pode resultar em respostas diferentes de pessoas diferentes diminuindo, assim, as formas de comparação das respostas. Já na entrevista estruturada o principal ponto fraco é não possibilitar que o entrevistador explore tópicos e itens que não foram contemplados quando as perguntas foram elaboradas. Essas entrevistas limitam o uso de direções alternativas de questionamentos com pessoas diferentes relatando seus pontos de vista e experiências. Com isso as diferenças e circunstâncias individuais não são inteiramente contempladas na avaliação. (BATISTA, 2003).

3.1.5 Melhores práticas

Para que a entrevista tenha uma boa fluidez, seja objetiva e obtenha informações úteis para o desenvolvimento do sistema, o entrevistador precisará se preparar para essa interação. O entrevistador necessita ao menos elaborar especificações do problema central que se deseja resolver. Precisa ter habilidades sociais, de entender e registrar as informações de uma forma organizada, podendo utilizar de meios digitais ou não. Abaixo duas das principais formas de registrar a entrevista segundo Kendall (2010):

- **Gravador:** Pode ser muito útil quando as respostas são longas ou possuir muitos termos técnicos. Requer a permissão dos entrevistados. Antes de iniciar a gravação é necessário solicitar aos participantes a permissão para gravar. Se todos autorizarem inicia-se a gravação dizendo os nomes de cada entrevistado e questionando novamente cada um deles sobre a autorização da gravação. Mantenha-o sempre em local visível;

Vantagens:

Registro completo do conteúdo da entrevista;
 Agilidade e melhor desenvolvimento;
 Reprodução fiel da entrevista para todos os outros membros da equipe.

Desvantagens:

Pode constranger o entrevistado;
 Pode diminuir a atenção do entrevistador;
 Pode ser necessária a transcrição do conteúdo da fita.

- **Anotações:** Pode utilizar um meio digital, ou seja, um documento texto ou digitalizado. Ou ainda, simplesmente, papel e caneta. Podem registrar as respostas fielmente como foram ditas ou ao menos os pontos importantes. Ao final de cada entrevista é aconselhável que os participantes assinem no final do documento para dar uma maior confiabilidade ao documento.

Vantagens:

Mantém o entrevistador sempre focado na entrevista;
 Pode ser usado para direcionar a entrevista em forma de um roteiro;
 Mostra interesse e preparo do entrevistado.

Desvantagens:

Pode causar a perda do objetivo da conversa;
 Excessiva atenção a fatos concretos e pouca aos sentimentos e opiniões;
 Pode ser necessária a transcrição do conteúdo da fita.

Existem algumas habilidades que o entrevistador deve possuir, preparos para a entrevista e algumas características relacionados ao local onde a entrevista vai ocorrer que contribuem muito para o bom desenvolvimento da entrevista. A seguir algumas orientações e melhores práticas para a aplicação da técnica segundo Kendall (2010) e Batista (2003):

- No dia que antecede a entrevista, o entrevistador deve entrar em contato com os entrevistados ou com um representante, para confirmar horário e local do encontro;

- O local da entrevista, de preferência, deve ser confortável e agradável, ou seja, os entrevistados devem sentir-se tranquilos para conseguirem concentra-se nas questões levantadas;
- Se o local escolhido for uma sala de reunião da própria empresa, que é muito comum, deve solicitar-se que os entrevistados sejam interrompidos menos possíveis, tanto por ligações telefônicas como por outros colegas de trabalho fora da entrevista;
- De preferência os telefones celulares devem ser desligados ou silenciados;
- Antes do início da reunião o entrevistador deve definir os objetivos e os limites da entrevista para minimizar os desvios;
- Estipular um tempo máximo de duração da entrevista; (não superar 1 hora);
- A entrevista deve ser conduzida de uma forma amigável, porém profissional;
- O entrevistador deve ser claro ao apresentar as perguntas;
- O entrevistador deve disponibilizar tempo suficiente para o entrevistado responder a questão antes de realizar a próxima pergunta;
- Deve prestar atenção em tudo que o entrevistado fala;
- Deve evitar conduzir o entrevistado;
- O entrevistador deve estar preparado para alterar as perguntas e se aprofundar em uma nova linha discussão, desde que esteja alinhada ao objetivo inicial da entrevista e com o tipo de entrevista que está sendo utilizada;
- Se a entrevista possuir mais de um entrevistado, é aconselhável ter um número maior de entrevistadores que nunca deve superar o de entrevistados em mais de um;
- Se existir mais de um entrevistador, é interessante que eles tenham características diferentes para que o resultado da entrevista seja mais produtivo. Por exemplo, um dos entrevistadores poderia registrar as respostas enquanto outro poderia conduzir a entrevista, cuidando para que ela não fuja do objetivo inicial;

- O entrevistador deve perguntar aos presentes se há outra pessoa com a qual ele deve falar;
- Antes do término pergunte aos entrevistados se existe algo relevante sobre o assunto que eles desejam expor;
- No término da entrevista, os entrevistadores devem reunir os principais pontos e apresentar um resumo das respostas dos entrevistados agrupados por assunto; e
- Finalmente, o entrevistador deve agradecer aos entrevistados pela atenção e tempo que foi dedicado.

Todas essas as ações podem ser informadas aos envolvidos salientando como objetivo melhorar a qualidade da entrevista, ou seja, obter da melhor forma possível o entendimento do problema. Visa obter as informações mais próximas da realidade, pois a qualidade dessas informações depende muito de como são transmitidas pelo entrevistado e como o entrevistador as compreendem.

3.2 CRIATIVAS DE GRUPO

Segundo Xavier (2009), existem várias técnicas de levantamento de requisitos que exploram dinâmicas de grupos para a descoberta e o desenvolvimento de requisitos, as mais empregadas são: *Brainstorming* e *JAD (Joint Application Development)*. Pressman (1995), destaca além dessas duas a técnica de *Workshops* de Requisitos.

As técnicas de grupo são particularmente efetivas, pois envolvem e estabelecem o compromisso diretamente com os interessados e porque incentivam a cooperação. Todas elas, de alguma maneira, aplicam as diretrizes básicas a seguir: i) os encontros reúnem representantes de diferentes grupos de interesse, sendo definidas regras de preparação e participação; ii) um facilitador, podendo ser o analista ou um dos participantes, controla a reunião; iii) para registrar as ideias levantadas, são utilizados alguns meios como, *post-its*, quadro branco, anotações projetadas para acompanhamento de todos os participantes, etc. e iv) o objetivo é identificar e discutir um problema, propor possíveis soluções, negociar diferentes estratégias e especificar um conjunto inicial de requisitos da solução. (PRESSMAN, 1995).

O facilitador, geralmente, desempenha um papel fundamental no planejamento, escolha dos participantes e, principalmente, na condução da reunião, com o propósito de encorajar a participação ativa de todos os participantes para que se atinjam os objetivos de modo consensual. Se um dos participantes estiver participando pouco, é função de o facilitador trazê-la para a discussão, estimulando sua manifestação. FALBO (2012).

Em alguns contextos, as dinâmicas de grupo se sobrepõem ao trabalho individual, como as entrevistas. Isso porque, um grupo de pessoas reúne informações de várias origens, centralizam esses dados, discutem vários pontos de vistas, melhora o entendimento coletivo e atinge um consenso de uma maneira que muitas vezes não é possível atingir utilizando outras técnicas de levantamento de requisitos. FALBO (2012).

A seguir específico as principais técnicas, segundo Pressman (1995) e Xavier (2009), que exploram as dinâmicas de grupo.

3.2.1 BRAINSTORMING

3.2.1.1 Descrição

A técnica *Brainstorming*, que traduzindo significa tempestade de ideias, é uma técnica que tem como finalidade explorar a criatividade dos participantes gerando assim, através dos diferentes pensamentos, novas ideias e respostas. Além disso, motiva e desenvolve a equipe, pois, envolve as pessoas em uma discussão gerencial maior. É amplamente utilizada na criação de novos produtos ou realizar melhorias significativas nos existentes, resolução de problemas complexos, gestão de projetos e Engenharia de Requisitos. BRAGA (2008).

Segundo Soares (2007), o *brainstorming* é uma das técnicas de reuniões de grupo mais conhecida e mais antiga, utilizada para organização de ideias. Basicamente essa técnica tem como princípio reunir um grupo de pessoas que são especialistas no sistema e no negócio para que cada participante estimule o outro a criar ideias que ajudem a encontrar a melhor solução para o problema. Essa solução pode não acontecer na primeira reunião, podendo ser necessários outros encontros, porém respeitando a mesma ideia básica. Ainda segundo Soares (2007), todas as ideias apresentadas por qualquer dos participantes não podem ser recriminadas ou julgadas. A técnica é aconselhada a ser utilizada no início do desenvolvimento, pois ainda sabe-se pouco do projeto e sendo necessárias novas ideias.

Brainstorming é utilizada para gerar novas ideias, com isso a mente fica livre para aceitar outras ideias sugeridas e, com isso, incentivar a criatividade. O resultado positivo de uma sessão bem sucedida de *brainstorming* é um conjunto de boas ideias e o sentimento de que todos os participantes contribuíram para a solução do problema ou surgimento de um novo produto.

3.2.1.2 Processo

Braga (2008) resume o processo da seguinte forma, reúnam-se de uma até dez pessoas de especialidades e áreas diferentes, para que suas experiências distintas contribuam com a diversificação de opiniões. Deve existir um facilitador que terá como função, registrar todas as ideias sem exceção e um líder, que conduzirá a reunião e apresentará os temas a serem desenvolvidos. É fundamental que nenhuma ideia seja considerada errada ou absurda, contribuindo para o surgimento de novas ideias e principalmente nenhum dos participantes se

sinta constrangido. A duração de uma sessão de *brainstorming* deve ser de aproximadamente trinta minutos. Quando as ideias se esgotam, o líder reúne todas as ideias anotadas, as distribui para um grupo menor (entre três e sete participantes) solicitando que sejam escolhidas as melhores ideias, que depois de selecionadas, devem ser compartilhadas com todos os demais participantes.

As pessoas são as "peças" mais importantes para o êxito desta técnica, com isso é necessário um grupo bem definido de pessoas. As pessoas mais criativas e com certa experiência nesta área podem ajudar para a obtenção de um melhor resultado. Não existe um número certo para a realização de uma sessão, porém cerca de cinco a doze pessoas podem participar, ou um número suficiente de pessoas para a criação de novas ideias.

São necessários que todos os participantes tenham acesso as ideias geradas durante a sessão. Como alternativa ao quadro pode ser utilizado um *flip chart* para registrar as ideias na forma que elas surgem colando-as na parede, por exemplo. Para agrupar as ideias semelhantes pode ser aplicado algum critério de classificação, porém este não deve ser o principal objetivo. Sempre que possível, o facilitador deve ser uma pessoa bastante experiente.

Para Batista (2003), o processo possui as seguintes etapas:

- Definir os objetivos da sessão de *brainstorming* e seus participantes;
- Ao convidar os participantes, devem ser apresentados quais os tópicos serão abordados assim como o formato da reunião; Atentando-se ao fato que se utilizado alguma ferramenta de gravação, por exemplo, vídeo ou áudio, será necessário a concordância prévia de todos os participantes;
- Produzir um cronograma para a sessão e fazer uma sessão teste para verificar se o cronograma é coerente. Se for necessária alguma informação individual complementar, deve-se preparar um questionário adequado para ser aplicado a todos os participantes antes ou depois da sessão; e
- No decorrer da sessão, cabe ao líder ser ativo em conduzir a discussão e reunir os resultados no final de cada tópico. É essencial diferenciar entre o que é consenso do grupo e o que é a opinião dos participantes.

3.2.1.3 Vantagens e pontos fortes

A técnica *brainstorming* se torna eficiente quando cada participante possui uma parte do conhecimento de algum ponto específico do problema.

Soares (2007), destaca algumas vantagens:

- O fato de não existir ideias erradas nem julgamentos, ajudam a abolir dificuldades que outras dinâmicas de grupo podem gerar;
- Incentivar uma interação social mais amigável do que outras técnicas mais estruturadas podendo ser aprendida, com baixo investimento;
- Dar a todos os participantes uma visão mais ampla do problema ou do tema abordado.

Segundo Batista (2003), os pontos fortes podem ser destacados abaixo:

- No processo de *brainstorming*, todos os participantes podem levar o crédito pelas boas ideias que surgiram; não se gasta muito tempo para obter informações muito detalhadas e a sessão não tem a necessidade de se estender por mais de uma hora;
- O processo de grupo cria um sentimento de posse ou propriedade do resultado, pois todos de alguma forma participaram do desenvolvimento da solução ou inovação;
- Geração de diversas novas ideias; e
- Geralmente, boas ideias surgem da combinação de ideias que a princípio eram ruins ou sem relação com o tema, e surgiram de participantes que foram encorajados a apresentar ideias, muitas vezes incomuns.

3.2.1.4 Desvantagens

Para Soares (2007), a principal desvantagem é, por ser um processo relativamente não estruturado, não gera a mesma quantidade ou nível de detalhes de outros processos.

Batista (2003), aponta três principais desvantagens:

- Todos os participantes do grupo devem estar bem compenetrados e sentir-se à vontade para a sessão produza resultados satisfatórios;
- Geralmente sessões de *brainstorming* são caras e devem ser usadas de forma moderada, se possível somente realizar uma sessão durante todo o processo. Isso porque nos dias atuais o tempo das pessoas é sempre muito escasso e reuni-las por pelo menos uma hora pode ser um processo bastante custoso e difícil; e
- Nas sessões o analista deve cuidar para que as ideias geradas não sejam superficiais, não agregando informação para o que se deseja obter.

3.2.1.5 Melhores práticas

Soares (2007) destaca algumas boas práticas para conduzir uma sessão de *brainstorming* criativa e bem-sucedida:

- Estabelecer claramente os papéis do líder;
- Definir a tarefa e manter o controle na mão do facilitador;
- Definir o objetivo da sessão de *brainstorming* no seu início;
- Criar um ambiente positivo, onde as pessoas se sintam confortáveis para falar e apresentar suas ideias;
- Registrar de uma forma clara as ideias apresentadas;
- Incentivar o fluxo de novas ideias; e
- Acompanhar com muita atenção as comunicações verbais e não verbais.

Para evitar problemas originados de uma sessão de *brainstorming* mal elaborada e mal conduzida, Soares (2007), chama a atenção para os seguintes fatores:

- Local onde a sessão será conduzida;
- Ter um propósito específico e bem definido;

- Saber o que está sendo buscado e para que buscá-lo;
- Ser um facilitador em todos os aspectos;
- Listar os principais pontos obtidos;
- Estabelecer todas as regras para as pessoas participantes; e
- Não utilizar críticas durante a sessão.

Batista (2003) lista boas práticas muito semelhantes aos itens listados por Soares (2007) logo acima. Abaixo destacamos esses procedimentos práticos:

- Não permitir críticas às ideias que surgirem durante a sessão;
- Encorajar a participação de todos do grupo para a geração de uma grande quantidade de ideias;
- Não desprezar as ideias que a princípio pareçam irrelevantes;
- Manter as ideias curtas e objetivas;
- Listar as ideias levantadas;
- Evitar associar a ideia com a pessoa que a apresentou; e
- Tentar combinar as ideias.

Com essas melhores práticas apresentadas pelos autores Soares (2007) e Batista (2003) fica claro que um dos pontos mais importantes para o sucesso de uma sessão de *brainstorming* é relevar todas as ideias apresentadas mesmo que inicialmente pareça ser fora do contexto.

3.2.2 JAD (Joint Application Development)

3.2.2.1 Descrição

A técnica JAD, sigla para a expressão *Joint Application Development*, foi desenvolvida pela IBM em 1977 como parte de uma série de experimentos. Após três anos, JAD foi divulgada oficialmente pela IBM do Canadá. Inicialmente, o processo JAD era descrito como um conjunto de entrevistas realizadas com cada *stakeholder* de forma individual com a finalidade de obter os requisitos do sistema. Entre os anos de 1981 e 1987, o escopo de JAD cresceu e começou a abranger o planejamento e a análise. Foram introduzidas várias técnicas de engenharia para fazer parte de JAD. Além disso, foi adicionado o conceito de grupos dinâmicos nas sessões e ferramentas CASE foram introduzidas para facilitar a documentação. Essa nova fase de JAD tem como principal característica o maior envolvimento do usuário no qual ele assume mais firmemente o controle das decisões. Do ano de 1988 até os dias atuais, JAD continua se desenvolvendo, pois foram introduzidas técnicas como a prototipagem com o objetivo de incrementar a produtividade e a qualidade dos sistemas desenvolvidos. ALVES (1999).

Com o passar dos anos, JAD passou a ser conhecida mais como uma marca do que com um método específico, pois existem várias versões adaptadas conforme as necessidades dos clientes. À medida que novas técnicas estão surgindo, elas são incluídas nas sessões de JAD para garantir que essa técnica seja eficiente no levantamento de requisitos do sistema. ALVES (1999).

Braga (2008), descreve a técnica JAD como uma sessão realizada com um grupo de *stakeholders* de diversas áreas que já estão ou serão envolvidas pelo projeto. Essa técnica possui a finalidade de esclarecer dúvidas que são levantadas em diferentes áreas como: Desenvolvimento, Bando de Dados e Usuários do Sistema.

Soares (2007, p. 25) define a técnica 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." Ele representa o processo pela Figura 10.

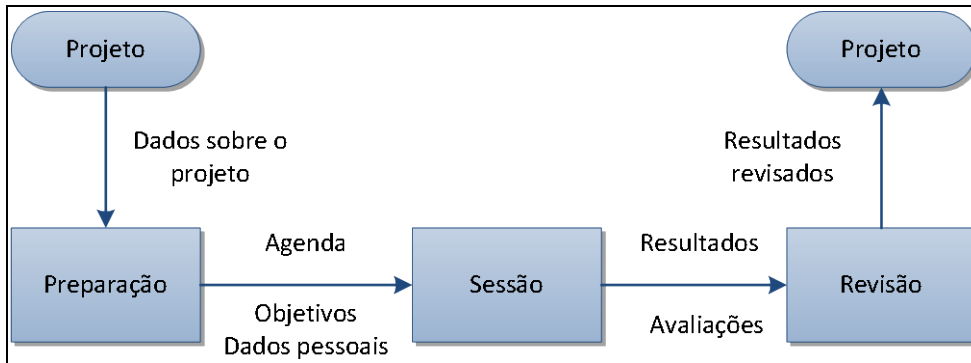


Figura 10: Diagrama da visão geral do JAD

Fonte: Soares (2007).

A técnica JAD visa criar sessões de trabalho estruturadas, utilizando uma dinâmica de grupo, em que analistas e usuários trabalham em conjunto para criar um sistema, iniciando com requisitos básicos e avançando até a definição de *layout* de telas e relatórios, focando principalmente a cooperação e o entendimento. Os desenvolvedores auxiliam os usuários a formular problemas e definir possíveis soluções. Com isso os usuários ganham um sentimento de envolvimento, posse e responsabilidade com o sucesso do sistema, participantes ativos do desenvolvimento. BATISTA (2003).

Para Alves (1999), as sessões de grupo surgiram para solucionar problemas de comunicação entre as pessoas que desenvolvem os sistemas. O propósito geral dessa abordagem é reunir pessoas em reuniões de trabalho, definir metas, incentivar a troca de ideias tendo como resultado um documento final concordado por todos os participantes.

JAD é uma sessão de grupo muito utilizado atualmente e tem como principal objetivo extrair informações dos usuários, em pouco tempo, através de reuniões estruturadas que buscam decisões e soluções por consenso. JAD é diferente de outras técnicas de desenvolvimento de sistema no sentido de envolver tanto usuários como profissionais de TI em uma mesma reunião de trabalho. ALVES (1999).

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

- Dinâmica de grupo: são realizadas reuniões guiadas por um líder experiente, analistas, usuários e gerentes, unindo forças e incentivando a criatividade dos participantes. A sessão terá como resultado final a definição dos objetivos e requisitos do sistema;

- Uso de técnicas visuais: utilizadas para melhorar a comunicação e o entendimento dos participantes;
- Manutenção do processo organizacional e racional: a técnica JAD utiliza a análise top down e atividades bem definidas, possibilitando assim, uma análise completa diminuindo as chances de falhas ou lacunas no projeto e cada nível de detalhe recebe a atenção necessária; e
- Utilização de documentação padrão: devidamente preenchida e assinada por todos os participantes de uma sessão. Este documento gerado garante a qualidade desejada no projeto e aumenta a confiança dos participantes.

A técnica JAD é composta por duas principais etapas: planejamento, que tem como objetivo especificar requisitos; e projeto, que trata do projeto de software.

Ainda segundo Batista (2003), os participantes de uma sessão JAD possuem seis papéis distintos:

- Líder de sessão: é o responsável direto pelo processo como um todo controlando a sessão, sendo o facilitador das reuniões, por isso deve ser competente, possuir bom relacionamento pessoal e grande capacidade de liderança. Deve ter o domínio de todos os aspectos da JAD e ter experiência na sua aplicação para, se necessário, planejar e entender as diversas tarefas da técnica JAD. Algumas habilidades são de extrema importância para o líder de sessão. São elas: compreender e desenvolver a dinâmica de grupo; iniciar e centralizar os debates; saber reconhecer quando a sessão está desviando do tema e corrigi-la; administrar as diferentes personalidades e características de cada participante; não permitir o desinteresse pela sessão mesmo que ela seja demorada e complexa; tudo isso é possível através da prática e da experiência na aplicação da técnica;
- Engenheiro de requisitos: é o participante responsável por produzir os documentos gerados durante as sessões JAD. Se durante as sessões são utilizadas ferramentas de software, essa pessoa deve saber utilizá-la eficientemente. Sempre que possível, deve ser um desenvolvedor experiente para compreender as questões técnicas e riqueza de detalhes discutidos durante a sessão. Deve possuir a habilidade de organizar bem as ideias e documentá-las por escrito com muita clareza;

- Executor: são os responsáveis pelo sistema sendo desenvolvido. Precisa fornecer aos demais participantes informações sobre a necessidade do novo sistema assim como os benefícios que se espera obter com ele. É responsável por tomar decisões executivas, como alocação de recursos e o desenvolvimento de novos sistemas;
- Representantes dos usuários: são as pessoas que atuam na empresa que irão utilizar o sistema. Durante a etapa de planejamento podem ser gerentes e pessoas-chave, com uma visão macro do sistema. Os representantes devem ser escolhidos de acordo com o conhecimento de suas necessidades na empresa, o entendimento da interação entre o seu departamento e os demais da empresa e algum conhecimento técnico de software. Podem ser incorporados alguns futuros usuários do sistema durante a fase de projeto da JAD;
- Representantes de sistemas de informação: são pessoas que estão totalmente familiarizadas com as capacidades dos softwares. Estes representantes devem auxiliar os usuários a entender o que é coerente ou possível fazer no novo sistema. Eles podem, por exemplo, apresentar para os usuários as consequências da escolha de um determinado caminho para a resolução de um problema; e
- Especialistas: são pessoas que podem fornecer informações detalhadas sobre um tópico específico, tanto com uma visão do usuário, pois conhece o funcionamento de uma parte da empresa, como de uma visão dos desenvolvedores, porque conhecem totalmente os aspectos técnicos da empresa.

O especialista de cada área possui características diferentes. Um especialista pode ser uma pessoa que utiliza determinado relatório. Um especialista de desenvolvimento pode ser um administrador de rede, onde sua interação com o projeto seria feita, por exemplo, durante a definição da infraestrutura que o novo sistema necessita.

3.2.2.2 Processo

A técnica JAD consiste em três fases, onde cada autor utiliza uma nomenclatura: Segundo Batista (2003), adaptação, sessão e finalização, para Braga (2008) e Pádua (2011) consistem em personalização, sessão e fechamento, já para Soares (2007), o método é dividido em três fases, preparação, sessão e revisão. Mesmo com a utilização de diferentes

nomenclaturas, todas convergem em uma mesma definição. Utilizarei para explicação do processo as seguintes expressões: adaptação, sessão e finalização.

A **fase de adaptação** visa à preparação para a sessão, ou seja, ocorre a definição da equipe, os participantes recebem orientação em relação à técnica, preparação do local das sessões e definição de particularidades do projeto. Sendo assim, devem ser escolhidos os participantes quanto à técnica de JAD, uma agenda deve ser desenvolvida adequadamente, e o material necessário deve ser providenciado.

É de responsabilidade do líder da sessão JAD, podendo ser auxiliado por um ou dois desenvolvedores, adaptar a técnica conforme o sistema a ser desenvolvido para que ela se torne mais efetiva. Segundo Pádua (2011), essa fase tem geralmente a duração de um a três dias e compreende os seguintes passos:

- A adaptação de iniciar pela definição em um nível macro, onde pode ser necessário entrevistas prévias com determinados clientes e usuário. Também é necessário obter informações referentes à organização para se familiarizar com a definição do problema. Pode ser utilizado, por exemplo, técnicas complementares como entrevista e observação;
- Obtida uma primeira compreensão dos objetivos do sistema, é preciso selecionar os participantes, convocá-los para participar das reuniões e disponibilizar para esses participantes uma lista prévia com os tópicos que serão tratados nas sessões para que eles possam estar mais preparados;
- O líder responsável pela sessão JAD deve definir a duração e a quantidade de sessões que serão realizadas e ajustar o formato dos documentos JAD as necessidade e características do sistema a ser desenvolvido; e
- O líder deve também preparar um material introdutório e todo conteúdo audiovisual que considere importante para ser apresentado durante as sessões.

Para atingir os objetivos dessa fase, Soares (2007), declara que as seguintes etapas devem ser seguidas:

- Junto com o gerente de projeto, analisar se é adequada a utilização de JAD. Para isso são examinados os riscos referentes ao tamanho do projeto, o domínio da metodologia e o "clima" na equipe;
- São definidas quantas sessões de JAD serão necessárias, que tipo e a agenda com as datas de realização;
- Elaborar a perspectiva gerencial, ou seja, obter uma visão clara do sistema como um todo, definindo os objetivos, escopo e restrições;
- Familiarizar-se com as áreas de negócio envolvidas, para isso, deve-se fazer a interface entre os usuários e os analistas;
- Elaborar a agenda da sessão indicando os passos a ser seguido, determinar onde o grupo se encontra no processo, para onde estão caminhando e o que ainda deve se alcançado;
- Preparar o grupo, apresentando uma abordagem da sessão, identificar as informações que devem ser obtidas previamente e distribuir a tarefa obter essas informações aos participantes;
- Definir a agenda detalhada, ou seja, preparar o roteiro que o facilitador utiliza durante a sessão de JAD; e
- Preparar a ferramenta que estabelecerá um padrão de documentação JAD.

Na **fase de sessão** os participantes apresentam e discutem suas ideias que são analisadas e debatidas até se chegar a um consenso. BATISTA (2003). Essa etapa consiste na realização das reuniões com os participantes selecionadas na fase anterior, de adaptação, tendo como objetivo produzir as informações detalhadas que são necessárias para o desenvolvimento do sistema. SOARES (2007).

Batista (2003) recomenda que sejam adotados os seguintes passos para este processo:

- **Apresentação:** os participantes são apresentados e o líder da sessão, assim como o patrocinador, dá boas-vindas a todos os participantes. Esse patrocinador apresenta resumidamente as necessidades que levaram ao desenvolvimento e os benefícios que

são esperados. O líder da sessão explica a dinâmica das sessões e o planejamento previamente montado;

- Definir os objetivos e requisitos: o líder da sessão incentiva a discussão para a elaboração dos objetivos ou requisitos gerais fazendo perguntas como: "Por que o sistema será desenvolvido?", "Quais os benefícios esperados?", "Como a empresa pode se beneficiar com o sistema?", "Quais as restrições relacionadas a normas ou leis afetam o sistema?". Conforme os requisitos vão se definindo, o analista os escreve de uma forma que fique visível para todos durante a reunião;
- Delimitar o escopo do sistema: com os requisitos definidos, é necessário organizá-los e chegar a um consenso sobre o âmbito do novo sistema. Para sistemas de informação é aconselhável identificar os usuários potenciais e determinar suas tarefas;
- Documentar tópicos abertos: os questionamentos que surgiram durante a sessão e não foram resolvidas devem ser documentadas para serem abordadas nas próximas sessões, devendo ficar com um dos participantes a responsabilidade de solucioná-la dentro de um prazo determinado; e
- Concluir a sessão: o líder da sessão JAD deve concluir a sessão revisando com todos os participantes as informações levantadas e as decisões acordadas. Nesse momento é dada oportunidade para que todos os participantes expressem qualquer consideração relevante, motivando por parte do líder da sessão o sentimento de propriedade e comprometimento dos participantes sobre os requisitos levantados.

Para Pádua (2011), a fase de sessão pode ser dividida em:

- Preparar o ambiente: as mesas devem ser organizadas em forma de "U" para que todos tenham uma visão clara dos tópicos, os equipamentos de audiovisual devem ser instalados e checados e recomenda-se preparação prévia do material apresentando aos participantes;
- Conduzir a sessão: inicia-se com a apresentação dos participantes, seguido da apresentação da agenda, revisão da perspectiva gerencial, definição junto aos participantes de uma lista de regras da reunião e o início dos trabalhos com base na agenda;

- Documentar: um dos participantes tem a responsabilidade de registrar tudo o que acontece na sessão JAD; e
- Encerrar a sessão: são revisados os tópicos da agenda, feita a avaliação da sessão e entrega da documentação gerada na sessão. Se for a última sessão do projeto, Pádua (2011), aconselha que seja feita a apresentação dos resultados alcançados.

Por último, a **fase de finalização**, onde deverá ser produzida uma lista de todos os pontos definidos durante as sessões JAD, um esboço da especificação de requisitos de *software*, que deverá ser demonstrada para os responsáveis com autonomia pela decisão de avançar com o projeto. Neste momento, também é essencial analisar as possíveis melhorias na aplicação da técnica. É necessário transformar todas as anotações, imagens e transparências geradas durante a sessão em documentos formais. Basicamente três passos devem ser seguidos BATISTA (2003):

- Completar a documentação: os analistas compilam a documentação gerada na sessão em documentos de acordo com as normas ou padrões da empresa;
- Revisar a documentação: toda a documentação gerada é encaminhada a todos os participantes para que façam uma análise e comentem. Se a quantidade de comentários for significativa, outra reunião é convocada para novas definições e discussões; e
- Validar a documentação: por último, quando todos os comentários já foram revisados, o líder da sessão JAD encaminha o documento ao patrocinador para que ele aprove. Assim que o documento for aprovado, cópias finais são encaminhadas a cada um dos participantes.

A sessão é finalizada quando se acredita que toda a informação importante foi definida.

3.2.2.3 Vantagens e pontos fortes

Segundo Pádua (2011), a técnica JAD apresenta as seguintes vantagens:

- Comprometer os usuários com os requisitos pelos quais eles têm o poder de decisão;
- Reduzir o tempo necessário para o levantamento da especificação dos requisitos;

- Excluir do projeto requisitos que possuam valor questionável;
- Esclarecer dúvidas relacionadas à interpretação de requisitos entre usuários e desenvolvedores;
- Elaborar um primeiro esboço das interfaces de usuários; e
- Identificar e discutir o quanto antes problemas políticos que possam atrapalhar o andamento do projeto.

Batista (2003), compara a técnica JAD com as entrevistas individuais e destaca as seguintes vantagens:

- Diminui a divergência de opiniões dos participantes, economizando tempo;
- A documentação é revisada não somente pelos desenvolvedores, mais por todo o grupo, inclusive clientes e futuros usuários do sistema; e
- Resulta em um maior envolvimento dos clientes e usuário na fase de desenvolvimento do sistema;
- A técnica JAD reduz o tempo relacionado ao processo de levantamento de dos requisitos, porém com um custo elevado, devido ao grande número de pessoas envolvidas. No período de dois a quatro semanas as informações não são somente levantadas, mas os requisitos, combinados por vários usuários do sistema, são identificados. Com um grupo experiente em JAD as empresas podem tornar seu processo de análise de sistema muito mais dinâmico;
- As sessões JAD auxiliam reunir especialistas, dando a eles possibilidade de compartilhar suas opiniões, entender outras e desenvolver o sentimento de posse do projeto;
- As formas de aplicação de JAD são bastante conhecidas, pois foi a primeira técnica de projeto acelerada disponível no mercado e, provavelmente, a mais conhecida podendo ser aplicada, sem muita complicação, por qualquer empresa;

- A técnica proporciona uma fácil integração de ferramentas CASE ao processo JAD, proporcionando uma melhora na produtividade das sessões e disponibiliza ao analista de sistemas um modelo já discutido e pronto para ser utilizado nas próximas fases;
- Proporciona ao analista ter visões de pessoas diferentes, aumentando a perspectiva do problema; e
- Na sessão muitos conflitos podem ser resolvidos, não permitindo que eles sejam levados para a fase de desenvolvimento.

3.2.2.4 Desvantagens

Os desenvolvedores de sistemas precisam ter em mente que não existe técnica completa para coletar requisitos, inclusive JAD. Esta pode ser ineficaz se não utilizada corretamente e alguns pontos específicos precisam ser resolvidos antes e durante as sessões JAD BATISTA (2003):

- "Dever de casa": Caso não seja feita uma preparação completa para a sessão de JAD, pode ser desperdiçado um tempo valioso dos profissionais, problemas errados podem ser apresentados, pessoas erradas podem ser convidadas a participar das sessões, recursos inadequados podem ser utilizados para a resolução dos problemas. Todos estes cenários podem aparecer caso os organizadores da sessão de JAD não estudem os aspectos do sistema que estão sendo avaliados;
- A equipe selecionada para participar da sessão de JAD deve incluir os empregados suficientemente capazes de fornecer o máximo de informação relacionado ao problema ou, pelo menos, das partes essenciais do problema. Por essa razão a seleção dos participantes é tão importante e deve ser feita com muita atenção. Os participantes devem ser formados não somente pelos empregados dos departamentos que utilizarão o sistema, mas também por outros usuários de diferentes níveis da organização. Esta variedade de pensamentos sobre o entendimento do processo refletirá algumas vezes em pontos de vista conflitantes, porém permitirá também aos participantes ver outras perspectivas;

- O facilitador atuando com motivador precisa se certificar de que todos os participantes, não somente os mais comunicativos têm a possibilidade de apresentar suas opiniões, ideias e ponto de vistas. Todos os especialistas do negócio na equipe JAD devem ser incentivados a oferecer sua ajuda, tornando as discussões mais produtivas. JAD também apresenta a falha mais comum das técnicas de desenvolvimento, que é acreditar na possibilidade de se obter dos usuários requisitos fundamentais, complexos e não ambíguos, utilizando modelos não executáveis;
- Outra desvantagem é o fato de que, teoricamente as pessoas da etapa de planejamento não são as mesmas da etapa do projeto, causando perda de informação obtida sobre o projeto, pois nenhum documento escrito tem a capacidade de transcrever fielmente as percepções obtidas pelas pessoas presentes nas reuniões;
- Alguns dos participantes podem não conseguir expressar suas ideias sobre o sistema ou podem se sentir inibidos por outros membros do grupo, principalmente colegas ou superiores hierárquicos. Outros participantes podem também não pensar criativamente quando estão em grupo, ou preferem uma conversa individual, ou ainda responder algum questionário em particular.

Pádua (2011), destaca as seguintes desvantagens:

- JAD exige a disponibilidade de tempo dos desenvolvedores, usuários e especialistas para as sessões;
- Caso o relatório uma sessão estiver incompleto, pode comprometer a qualidade da próxima sessão; e
- A cultura organizacional pode não estar compatível com a aproximação e o método JAD.

A técnica JAD deve ser aplicada com muita atenção, pois possuem pontos que podem comprometer sua eficácia como: "o não envolvimento de pessoas que desempenham papéis-chaves no processo de uso do sistema"; "A participação de pessoas sem comprometimento com o sistema" e "o número elevado de participantes (entre 8 a 15 para grupos experientes)", além disso, é de grande importância que o líder do JAD tenha experiência e envolva os participantes em todas as reuniões. Porém, as experiências de JAD's de sucesso, onde o índice

de mudança de requisitos é baixo e a redução de tempo gasto no levantamento de requisitos faz da técnica JAD uma técnica a ser sempre considerada. (PÁDUA, 2011).

3.2.2.5 Melhores práticas

Um dos pontos chave no processo JAD, e muitas vezes deixada de lado, é a preparação das sessões. De uma a três semanas antes de iniciar as sessões a equipe formada por executivos e desenvolvedores deve identificar os seguintes pontos, segundo Batista (2003):

- Projeto e seus objetivos: qual a essência do problema que os desenvolvedores e os participantes da sessão estão enfrentando;
- Objetivo e o que se espera das sessões: quais os pontos que devem esclarecidos nas sessões e quais os resultados que a equipe espera alcançar nas sessões JAD;
- Quais participantes serão beneficiados com o sistema: pessoas de que área e de que hierarquias deveriam participar para garantir a versatilidade da equipe de JAD; e
- Agendar as sessões com antecedência: escolher local, hora e ferramenta a ser utilizada que seja bom para todos. Sem uma preparação adequada e sem um plano claro de ações as sessões JAD podem tornar-se desorganizadas e resultar em desperdício de recursos.

Para o líder de sessão JAD é aconselhável adotar as seguintes práticas:

- Criar um bom ambiente;
- Proporcionar aos participantes um formulário simples para completar com detalhes pessoais antes que a reunião inicie. Essa prática pode ajudar a identificar uma atividade quando for preciso um último ajuste no ambiente, ou quando acontecer uma eventual espera pelos participantes atrasados;
- Auxiliar os participantes na formulação do problema e guiá-los quando necessário;
- Não permitir comportamentos destrutivos ou negativos por parte de algum dos participantes;

- Evitar polêmicas entre os participantes com ideias e opiniões divergentes;
- Não sugerir soluções para o problema e evitar uma avaliação das soluções propostas;
- Garantir-se que todos os participantes estão tendo oportunidade de contribuir com ideias e soluções;
- Quando possível, esclarecer os conceitos a serem explorados utilizando slides, quadros brancos, transparências ou outros meios para destacar aspectos do sistema a ser construído; e
- Apresentar diversas alternativas para enfatizar pontos que existam mais de uma solução e para estimular a discussão sobre temas comuns e problemas; e
- Preparação do ambiente: a disposição das mesas, quando possível, deve ser em forma de "U", os equipamentos audiovisuais devem ser revisados, o *checklist* deve ser conferido, auxílio visuais devem ser instalados e recomenda-se preparação antecipada de pastas para os participantes.

3.3 PROTOTIPAÇÃO

3.3.1 Descrição

Diversas vezes os usuários têm dificuldade em visualizar como um requisito especificado na forma de um documento escrito ou por um conjunto de modelos irá se concretizar em um sistema. É muito comum as pessoas não conseguirem descrever suas necessidades e aspirações sem ter algo tangível ao seu alcance. Para esses casos, se um protótipo de sistema for criado para demonstração dos requisitos, fica muito mais fácil para os usuários e *stakeholders* identificar problemas e sugerir melhorias nos requisitos. SOMMERVILLE (2003).

Kendall (2010), afirma que prototipação é "uma técnica que permite a obtenção rápida de sugestões, inovações ou mudanças e permite captar o sentimento do usuário em relação ao sistema em desenvolvimento".

Os protótipos possibilitam que os desenvolvedores criem um modelo do software que deverá ser desenvolvido. Desenvolvedores e usuário se reúnem e definem os objetivos gerais do *software*, identificam os requisitos que são conhecidos até aquele momento e decidem em quais áreas será preciso detalhar as definições. O passo seguinte é fazer um projeto rápido que tem como objetivo mostrar uma representação dos aspectos do *software* que serão visíveis ao usuário como, por exemplo, formatos de telas de entrada e saída de dados. Com este projeto é possível criar o protótipo que será validado pelo usuário e utilizado para refinar os requisitos do sistema a ser desenvolvido. Com as interações o protótipo é modelado para atender as necessidades do usuário ao mesmo instante em que o desenvolvedor compreende melhor o problema.

Na criação do protótipo, geralmente, não é necessário ser observado requisitos não funcionais, como por exemplo, *desempenho*, usabilidade e aspectos de segurança. Funcionalidades e flexibilidade podem ser deixadas de fora do protótipo e ferramentas de gerenciamento e garantias de qualidade podem ser desconsiderados. BATISTA (2003).

Embora os protótipos sejam desenvolvidos com o objetivo principal de apresentar para os usuários um esboço do sistema, deve ser classificado como descartável ou evolutivo.

Protótipo evolutivo: É desenvolvido com a finalidade de criar partes de um sistema funcional de forma interativa, possibilitando ao usuário o acompanhamento do desenvolvimento de forma que alguma alteração possa ser realizada com custos relativamente menores em relação ao sistema final. Esta abordagem propõe entregar de uma forma rápida ao usuário uma versão executável do sistema. Portanto, os requisitos que estão bem compreendidos são contemplados nas versões iniciais do sistema. Os requisitos que ainda possuem uma definição incompleta somente deverão ser implementados após grande utilização do protótipo.

De uma forma geral, Batista (2003), considera que todo o ciclo de vida de um sistema pode ser considerado como uma série incremental de protótipos acumulativos. A prototipação evolutiva tem como base o princípio de que os requisitos dos usuários serão alterados e sob uma visão evolutiva do ciclo de vida do *software* pode-se considerar que a primeira entrega é o protótipo inicial. Alterações e melhorias posteriores resultam em novas entregas de protótipos mais completos. Este ciclo continua até que se obtenha o produto final.

Para que um protótipo seja desenvolvido de uma forma rápida, existem três abordagens possíveis BATISTA (2003):

- Prototipação no papel é quando uma simulação do sistema é desenvolvida e utilizada para realizar experimentos, testes e simulação sobre o *software*;
- Prototipação "Mágico de OZ", quando uma pessoa simula as respostas possíveis do sistema para determinados *inputs* do usuário. O usuário interage com uma pessoa que representa o sistema, simulando as possíveis respostas geradas por esse sistema. Esta aplicação é muito útil quando um novo sistema precisa ser desenvolvido como uma continuação de um sistema já existente. Isso é possível, porque os usuários já conhecem a interface e podem ver as interações entre esta e a funcionalidade do sistema, simuladas pelo "mágico de OZ". Possui um baixo custo e também não necessita que se desenvolva um *software*, exceto o da interface com o usuário; e
- Prototipação automatizada, onde uma linguagem de quarta geração, ou outra ferramenta de rápido desenvolvimento, é utilizada para implementar um protótipo executável. É uma técnica mais com maior custo do que as duas anteriores.

Protótipo descartável (*throw-away*): Tem como finalidade demonstrar ao usuário, aspectos relacionados à interface de usuário, problemas de comunicação com outros sistemas e a viabilidade de *performance* rápida. BRAGA (2008).

Para Sommerville (2003), o protótipo descartável tem como principal objetivo esclarecer os requisitos e fornecer dados complementares para os gerentes avaliarem os riscos do processo. O protótipo será descartado logo após a análise dos requisitos e avaliação dos requisitos.

Para a construção de um protótipo descartável, podem ser utilizadas ferramentas de desenvolvimento que possuam biblioteca de componentes, processadores de texto, ferramentas de manipulação de imagens ou até mesmo um lápis e papel, porém, no caso de protótipos descartáveis, como o próprio nome define, é aconselhável que não o utilize para fins de implementação por não seguir padrões de qualidade, resultando em um aumento na manutenção. BRAGA (2008).

3.3.2 Processo

O processo de criação do protótipo inicia com uma análise preliminar dos requisitos dos usuários. Em seguida, inicia-se um processo iterativo de desenvolvimento do protótipo e validação com os usuários. Cada fase de repetição possibilita que o usuário compreenda melhor os requisitos, inclusive as implicações dos requisitos gerados nas interações anteriores. Geralmente, um conjunto final de requisitos pode ser levantado e o protótipo é descartado. CARVALHO (2001).

Segundo Batista (2003), um procedimento geral pode ser adotado para a prototipação rápida, segue abaixo:

- Inicialmente, deve reservar tempo suficiente para a elaboração do protótipo. Se existir a necessidade de avaliar o protótipo junto ao usuário, então será necessário reservar um tempo para projetar tarefas relevantes, reunir usuário, validar o protótipo e registrar os resultados;
- Montar o ambiente necessário para a criação do protótipo, incluindo o hardware, software e as ferramentas para o seu desenvolvimento;

- Analisar o plano de criação do protótipo e desenvolvê-lo;
- Selecionar os usuários chaves para testar e validar o protótipo. Um facilitador também poderá participar para orientar os usuário e proceder à avaliação;
- Realizar tarefas reais para serem feitas pelos usuários utilizando o protótipo;
- Acompanhar o procedimento de validação e garantir que o protótipo possa ser utilizado para cumprir as tarefas;
- Cada sessão deve ser conduzida pelo facilitador, que instrui os usuários para trabalhar nas tarefas selecionadas, interagindo e respondendo ao sistema apropriadamente;
- Caso necessário, junto à utilização do protótipo pelo usuário, pode ser realizada uma entrevista para obter informações adicionais;
- Agradecer ao usuário pelo tempo investido na utilização e validação do protótipo;
- Analisar as informações obtidas e registrar as observações e avaliações do usuário;
- Relacionar as implicações e recomendações para o projeto, assim como dar um *feedback* para a equipe de desenvolvimento; e
- Verificar onde é necessário melhorar o protótipo e repetir o processo descrito acima até que o protótipo preencha as principais dúvidas de requisitos.

3.3.3 Vantagens e pontos fortes

Para Batista (2003), a prototipação possui as seguintes vantagens:

- Desenvolver um protótipo proporciona que o usuário visualize um modelo de sua lista de requisitos, permitindo que revisões a este modelo sejam implementadas com baixo impacto no produto final;

- Construir um protótipo na fase de levantamento permite aos usuários finais do sistema testá-los através de *software* e descobrir o que realmente é preciso, entendendo como o sistema pode ser usado para apoiar o trabalho deles;
- É muito comum os usuários encontrarem dificuldade em visualizar como uma descrição textual de requisitos terá como resultado final um software executável. Com um protótipo em mãos para demonstrar os requisitos que eles não compreendem por completo, os usuários descobrem que é mais fácil também encontrar problemas e sugerir melhorias nos requisitos;
- Proporciona aos usuários uma apresentação palpável do que trata o sistema;
- Ajuda a definir todas as viabilidades e utilidades do sistema antes que o projeto entre na fase de altos custos de desenvolvimento;
- Ainda para Batista (2003), "Prototipação é a única maneira efetiva de desenvolver interfaces de usuários.";
- É possível utilizar o protótipo para desenvolver testes de sistema que podem ser usados mais tarde no processo de validação de sistema;
- O desenvolvimento de protótipos exige que seja feito um estudo detalhado dos requisitos. Isto em diversas vezes revela inconsistências, omissões e requisitos incompletos;
- Permite alterar o sistema com mais antecedência no desenvolvimento, deixando-o mais próximo às necessidades do usuário. Com isso o custo de alteração acaba sendo muito menor; e
- Possibilita descartar um sistema quando ele se mostra inadequado com mais antecedência. A prototipagem é muito utilizada na análise de viabilidade.

3.3.4 Desvantagens

Batista (2003), afirma que existem custos e problemas relacionados à prototipação, são eles:

- Custo de treinamento: Se uma empresa não possuir experiência com a elaboração de protótipos, os desenvolvedores devem receber treinamento no ambiente de prototipação;
- Custo de desenvolvimento: Varia de acordo com o tipo de sistema a ser prototipado e da técnica a ser aplicada. O custo varia de poucas pessoas por dia, para sistemas menores, a muitas pessoas por ano para sistemas grandes;
- Mudança de cronogramas: Muitas vezes os desenvolvimentos de protótipos podem ocasionar uma prorrogação de cronograma, conseqüentemente, o atraso na data de entrega do sistema. Por outro lado, o tempo investido na prototipação pode ser recuperado, evitando, por exemplo, a refação de uma atividade;
- Protótipos incompletos: Em algumas situações o protótipo pode apenas simular o sistema ou interface na sua fase final, sem passar por estágios intermediários. Isto oferece pouca ajuda na definição de requisitos que poderiam surgir na elaboração do protótipo;
- A prototipação pode induzir aos analistas e desenvolvedores a não criarem uma documentação formal do sistema;
- Um dos maiores problemas dessa técnica, ainda segundo Batista (2003) está no fato de os usuários ficarem satisfeitos com o protótipo e optarem por não avançarem até a versão final do sistema;
- Mesmo geralmente sendo considerada mais rápida, em alguns casos a prototipação pode levar mais tempo para trazer resultados que outras técnicas;
- O investimento exigido para elaborar um protótipo baseada em *software* e *hardware* são bem maiores do que o investimento para a prototipação em papel.

Para Kendall (2010), a prototipagem pode gerar problemas, dentre eles:

- Gerência dos projetos: Geralmente, várias iterações são necessárias para se ajustar um protótipo. Com isso, surge uma questão: quanto para? Se essa questão não for pensada com muita atenção, a prototipagem pode se estender indefinidamente. É fundamental, delimitar e seguir um planejamento para coletar, analisar e interpretar as

informações do usuário em relação ao protótipo. Além disso, Kendall (2010), assim como Batista (2003), afirma que em alguns casos, utilizar os protótipos pode ser caro e demandar muito tempo;

- Considerar o protótipo como a versão final do sistema: Assim como Batista (2003), Kendall (2010), destaca que o maior risco da prototipagem é que os usuários, ao terem contato com o protótipo rodando, concluem que o sistema está muito próximo do fim, achando que o protótipo é a versão final. Da mesma forma, os desenvolvedores podem se sentirem tentados em utilizar os protótipos descartáveis no sistema, não considerando que a qualidade do protótipo pode não ser a mais compatível com a qualidade de um sistema. Portanto, gerenciar expectativas é essencial para a utilização bem sucedida da prototipagem.

3.3.5 Melhores práticas

A prototipação traz muitas vantagens somente se o protótipo puder ser desenvolvido substancialmente mais rápido que o sistema final. Quando utilizada da forma correta, esta técnica é muito útil para superar as várias dificuldades relacionadas ao processo de levantamento de requisitos, principalmente as dificuldades de comunicação e de necessidades do usuário. Batista (2003), destaca os principais procedimentos práticos que podem auxiliar no sucesso da aplicação da técnica de prototipação, são elas:

- Evitar desperdiçar muito tempo no desenvolvimento do primeiro protótipo. O protótipo deve conter as principais funcionalidades do sistema e não gastar muito esforço com características particulares como, por exemplo, animação, cores e imagens;
- Evitar desenvolver o protótipo muito elaborado, pois isso pode gerar uma sensação ao usuário que o protótipo na verdade é o produto final;
- Evitar incluir características ou funcionalidades que irão aumentar as expectativas dos usuários, porém, muito provavelmente não serão implementadas em um sistema real, como por exemplo, tempo muito rápido de resposta às funcionalidades e gráficos muito complexos; e

- É fundamental que o protótipo seja desenvolvido rapidamente, pois devem estar disponíveis logo na fase inicial do processo de levantamento de requisitos. Isso auxiliará muito na obtenção das informações junto aos usuários e o entendimento do analista quanto às funcionalidades do sistema.

Segundo Kendall (2010), para o desenvolvimento de um protótipo funcional, as diretrizes abaixo podem ser muito úteis:

- Defina o objetivo do protótipo antes de iniciar o seu desenvolvimento. Estes objetivos podem ser definidos em conjunto com analista desenvolvedores e se necessário até alguns usuários chaves;
- Utilize módulos gerenciáveis: Para a prototipagem não é preciso e na maioria das vezes, nem esperado, desenvolver um sistema complexo. Devemos lembrar que o protótipo é para auxiliar o entendimento dos requisitos principais do sistema e não tem como objetivo atender a todos eles;
- Assim como Batista (2003), Kendall (2010), destaca que o protótipo deve ser desenvolvido rapidamente. O desenvolvimento de um protótipo durante as fases de levantamento de requisitos não pode consumir muito tempo, caso contrário perde o seu propósito. Para acelerar o seu desenvolvimento, é aconselhável utilizar ferramentas adequadas;
- Altere o protótipo em iterações sucessivas e contínuas: O protótipo deve ser alterado focando as necessidades do usuário. Cada alteração necessita uma nova análise; e
- Destaque a interface com o usuário: As interfaces utilizadas no protótipo devem permitir que o usuário interaja com o sistema de uma forma fácil. Possivelmente será necessário um treinamento mínimo para isso. É muito indicado na prototipagem que se utilize sistemas interativos com interfaces gráficas.

3.4 OBSERVAÇÃO

3.4.1 Descrição

Geralmente as pessoas não conseguem detalhar seu trabalho, pois para elas suas tarefas são quase um processo natural. Elas estão concentradas apenas em seu trabalho e podem não ter uma visão de como seu trabalho se relaciona com outras atividades da empresa. Porém, as questões sociais e organizacionais que influenciam o trabalho, somente tornam-se claras quando observados por uma pessoa imparcial. (SOMMERVILLE, 2003). Muitas vezes, a melhor forma de entender quais são suas atividades é observá-las no seu ambiente de trabalho realizando essas tarefas. (BATISTA, 2003).

Um exemplo clássico dessa definição é utilizado por Kotonya (1998 apud BATISTA 2003), “É muito mais fácil demonstrar o processo de fazer um laço no cordão do sapato do que descrevê-lo.”.

Para Kendall (2010), observar o comportamento e o ambiente de trabalho da pessoa pode ser uma maneira bastante eficaz de obter informações que, simplesmente, não são percebidas quando outras técnicas são utilizadas.

Segundo o guia PMBOK® (2008, p. 96),

As observações fornecem uma maneira direta de se examinar indivíduos em seu ambiente e como desempenhar o seu trabalho ou tarefas e executam processos.

“A **etnografia** é uma técnica de observação que pode ser utilizada para compreender os requisitos sociais e organizacionais.” (SOMMERVILLE, 2003), ou seja, entender a política da empresa e a cultura com o objetivo de familiarizar-se com o sistema e suas características.

A etnografia, segundo Falbo (2012), é o estudo de pessoas em seu ambiente natural de trabalho. Para Xavier (2009) esta técnica é aconselhável quando o processo a ser analisado é de grande complexidade de detalhes, ou quando os usuários que executam a tarefa têm dificuldade de definir as especificações de suas tarefas.

Nesta técnica, um analista observa o usuário realizando uma tarefa em particular no seu local de trabalho e anota as características e comportamentos envolvidos em cada atividade. Com a técnica é possível entender como os usuários interagem com os sistemas e com outros companheiros de trabalho.

Levando para o contexto do levantamento de requisitos, envolve a participação direta ou indireta do analista nas atividades diárias dos usuários, durante um determinado tempo, enquanto são coletadas informações sobre os processos sendo realizados. A observação pode ser direta, quando o analista (observador) está presente enquanto o usuário executa suas tarefas, ou indireta, quando a tarefa é observada a distância através de outros meios como, por exemplo, a gravação de vídeo. Na observação direta, o observador pode acompanhar o trabalho da pessoa, sem interromper a execução da atividade. Caso necessite de uma explicação mais detalhada, o observador pode solicitar uma pausa na tarefa para realizar uma melhor análise. (BATISTA, 2003).

A técnica de observação é muito eficaz, segundo Sommerville (2003), na descoberta de dois tipos de requisito:

- Os requisitos derivados da maneira como os usuários realmente trabalham e não da maneira como os processos estão documentados ou explicados;
- Os requisitos derivados do relacionamento entre as pessoas que tomam decisões e outras pessoas da empresa. São derivados da colaboração e do conhecimento das atividades de outros usuários.

Por outro lado, a técnica de observação não é apropriada para obter requisitos organizacionais e de domínio por ter seu enfoque no usuário final. Também pode não ser fácil identificar novas características que deveriam ser acrescentadas ao sistema. Assim, pode ser combinada com outras técnicas de levantamento de requisito, podendo ser utilizada para confirmar ou negar informações obtidas nas entrevistas e /ou questionários.

Essa técnica pode ser utilizada quando não é possível a análise de documentos, ou seja, para sistemas que não possuem documentação podendo ser aplicada tanto em tarefas manuais, automatizadas ou em tarefas muito complexas para serem descritas.

3.4.2 Processo

Para Batista (2003), a observação é considerada mais uma arte do que ciência, porém alguns processos podem ser definidos. Uma decomposição das tarefas é amplamente empregada e analisar esta decomposição pode ser eficaz para planejar as sessões de observação.

O processo de observar inicia decidindo o que observar que muitas vezes, é mais difícil do que a própria observação. Inicialmente devem-se definir os objetivos e quais requisitos precisam ser levantados e se estas informações devem ser obtidas focando a quantidade ou o detalhamento. Em seguida, são definidos horários, locais e pessoas que serão observadas. Para garantir a transparência e evitar o constrangimento, os usuários devem ser avisados que serão observados. A cooperação dos usuários com o processo de observação deve ser o mais natural possível, ou seja, a rotina de trabalho da pessoa observada deverá sofrer o mínimo de interferência possível do entrevistador. Se for utilizado algum meio digital para registro da atividade, por exemplo, gravação em vídeo, os usuários devem autorizar por escrito, por razões legais.

Na próxima etapa, deve-se decidir qual a técnica de registro deve ser utilizada. Se o registro será feita por anotações à mão, maneira mais tradicional, gravações em áudio, vídeo ou audiovisual. Quanto mais detalhada tiver a gravação ou documentação da observação, mais tempo será utilizado para analisá-la.

Por fim, deve ser feito uma análise, resumo e relato da documentação de acordo com os objetivos definidos no início da sessão. Na prática, segundo Batistas (2003), a observação frequentemente é utilizada para complementar ou esclarecer as informações obtidas por uma entrevista, para que o analista tenha de uma forma prática e rápida uma visão dos tópicos abordados nas entrevistas. O usuário pode falar sobre as suas tarefas e, até mesmo, porque faz, e depois ele é observado realizando a tarefa descrita.

Com a utilização da técnica de observação, pode-e acabar gerando mais dados do que o necessário, pois é fácil produzir informações. Batista (2003) recomenda utilizar a observação de forma cuidadosa e concentrar-se nas tarefas críticas e interativas, e onde for possível, fazer uma abordagem de amostragem.

Ainda que a anotação seja a principal maneira de se registrar as informações de uma sessão de observação, o observador deve saber utilizar equipamentos de vídeo. A preparação para uma sessão de observação é fundamental para a obtenção do máximo possível de dados. Diferentes visões, obtidas na sessão de observação, resultam em conjuntos distintos de informações obtidas. Quando em uma sessão de observação é feita por uma equipe de desenvolvedores, deve ser atribuído com antecedência, qual o ângulo de observação cada membro da equipe deverá ter.

Os métodos de observação, conforme QREN (2013), envolvem os seguintes passos:

- Passo 1 - Escolha da situação: Os contextos para observação são definidos antes dos interesses das pessoas que solicitam a avaliação. O analista negocia os locais de observação e os pontos que serão observados junto com as partes interessadas;
- Passo 2 - Observação: O observador se atenta às tarefas do observado, tendo o cuidado de atrapalhar o menos possível o comportamento do usuário. Este trabalho, resumidamente, consiste em anotar ou gravar as situações encontradas pelo observador de uma forma mais discreta possível. O analista pode anotar assim que as atividades são realizadas ou longe do observador para não causar constrangimento.

Esse passo não se limita a uma simples observação, mas deve ser complementado por uma análise organizacional ou institucional para identificar como as atividades das pessoas se relacionam dentro da empresa;

- Passo 3 - Análise do material coletado: Uma forma de analisar e processar o material coletado é separar as atividades conforme suas características e sequenciá-las como um filme. O observador identifica nas anotações quais os pontos, explícitos ou implícitos, que agregam valor ao trabalho executado. As sequencias das atividades e o levantamento revelam quais são os pontos que realmente são relevantes para o desenvolvimento do sistema que será desenvolvido;
- Passo 4 - Análise da sequencia junto dos usuários: A sequencia das atividades e as avaliações feitas pelo observador são reescritas de forma que o observado permaneça anônimo. Posteriormente, são entregues a outras partes interessadas para que sejam reavaliadas e validadas por elas. Esta etapa é importante para garantir que o observador não teve um entendimento distorcido da sequencia das atividades e do seu contexto.

3.4.3 Vantagens e pontos fortes

Como defendido por diversos autores aqui citados, a principal vantagem da técnica de observação é permitir que o observador tenha a visão do que os usuários realmente fazem. Segundo Batista (2003), a observação direta permite focar as atenções em áreas específicas

que se deseja alguma informação. Já a observação indireta permite extrair a atividade que de alguma forma não foi possível ser registrada ou notificada.

Um ponto forte que deve ser destacado é que essa técnica é considerada simples de ser aplicada e não exige muito treinamento e preparação. Outra vantagem é que os usuários não precisam descrever suas atividades, diminuindo assim problemas relacionados à articulação de ideias e interpretação de informação. O observador pode obter informações de aspectos sociais e organizacionais, sem a necessidade de um relato do usuário, além de revelar dados que outras técnicas não conseguem fornecer. SOMMERVILLE (2003).

3.4.4 Desvantagens

A técnica de observação sofre algumas críticas. Para Belgamo (2000), pode provocar alterações no ambiente de trabalho ou no comportamento dos usuários observados. Outra crítica é a de que esse método baseia-se muito na interpretação pessoal do observador. Além disso, o grande envolvimento do analista pode distorcer a análise da atividade ou ainda levar a uma interpretação parcial da realidade.

Segundo Batista (2003), a observação pode atrapalhar e assim como citado por Belgamo (2000), alterar o comportamento do usuário devido à presença do observador no seu dia-a-dia. As anotações ou gravações feitas durante o processo de observação precisam ser analisadas pelo analista que a fez, podendo demandar mais tempo por não permitir que a análise seja feita por outros analistas. Outro ponto fraco é que ela pode obter uma grande quantidade de informações irrelevantes. Além disso, com essa técnica torna-se difícil capturar atividades cognitivas.

3.4.5 Melhores práticas

Assim como em outras técnicas de levantamento de requisitos, a observação envolve planejamento, desenvolvimento e registro dos resultados. No planejamento, o analista deverá definir o que deverá ser observado, quais usuários observar, quando, onde, porque e de que maneira. Sommerville (2003) orienta que não existe uma forma padrão de se conduzir um processo de observação, contudo, sinaliza algumas diretrizes para a aplicação da técnica:

- É essencial investir um tempo conhecendo as pessoas envolvidas para estabelecer uma relação de confiança;
- O analista deve assumir que os usuários sendo observados são competentes em seu trabalho e tentar capturar meios não padronizados de se trabalhar. Frequentemente, esses meios apontam para eficiência no processo de trabalho que foram desenvolvidos e incorporados a partir da experiência do usuário;
- Devem-se registrar de forma detalhada as práticas de trabalho durante todo o processo de observação e no final do processo, redigir um relatório contendo todos esses detalhes observados. Só será possível criar um quadro coerente do processo após diversos desses detalhes terem sido coletados;
- Antes de iniciar o trabalho, o analista deverá informar as pessoas e dizer como a observação vai ser conduzida assim como seu objetivo.

Referente à definição de quando realizar a observação é de total importância não apenas considerar se as pessoas que serão observadas estarão trabalhando nos processos de interesse no período combinado, mas também se esses processos terão uma ocorrência significativa no período agendado.

Quando os usuários observados se sentem constrangidos é muito comum que eles não desempenhem as atividades da maneira habitual. Algumas estratégias, segundo Batista (2003) pode auxiliar na diminuição desses problemas:

- Deixar bem claro quais são os motivos da observação;
- Fazer com que o processo de observação seja longo, para que se torne uma atividade normal e o usuário se acostume com ela;
- Fazer a observação "disfarçadamente", ou seja, não transformar esse período de levantamento em um processo burocrático e cansativo;
- Garantir que os usuários que estão sendo observados entendam a razão do levantamento e que não vejam o observador e analista de forma negativa;

- Fazer uma sessão piloto para testar o formato da observação e sentir qual é a expectativa. Isso ajudará também a determinar o tempo necessário para a realização da observação. Assegurar para que o tempo da interação acontecer seja suficiente;
- Tentar atrapalhar o menos possível;
- Registrar quaisquer atividades que não foram entendidas e esclarecê-las com o usuário assim que a sessão for finalizada;
- Se necessário, solicitar documentos e relatórios utilizados pelo usuário em suas tarefas;
- Encerrada a observação, registrar as primeiras impressões antes de iniciar a fase análise;

Alguns itens importantes, para Sommerville (2003), devem ser executados antes, durante e depois da observação:

- Antes, é preciso identificar as áreas dos usuários que serão observados; obter a autorização das respectivas gerências para executar as observações; obter nomes e cargos das pessoas chave envolvidas na observação; e apresentar a finalidade do estudo;
- Durante, é preciso familiarizar-se com o local de trabalho que será feita a observação. Para isso é necessário observar a estrutura organizacional atual; as facilidades automatizadas e manuais; coletar documentos e procedimentos que são utilizados em cada processo específico observado; e registrar dados estatísticos das tarefas, como: frequência que ocorrem, tempo de duração para cada pessoa observado, estimativas de volume. É muito importante, além de observar as tarefas normais acima, observar as exceções;
- Depois, é preciso documentar as informações coletadas nas observações feitas. É preciso revisar as informações com as pessoas observadas, superiores e usuários chaves, para consolidar o resultado da observação.

4 CONSIDERAÇÕES FINAIS

4.1 CONCLUSÕES

Escolher a técnica de levantamento de requisitos depende principalmente do tempo e de recursos disponível para o analista de sistema, do tipo de informações que precisa ser obtida e do tipo de sistema que deve ser desenvolvido.

Existem diversas técnicas, porém somente uma pequena quantidade delas são comprovadamente efetivas. No primeiro momento, a equipe de desenvolvimento deve dar preferência a técnicas que são conhecidas, familiares e utilizadas na empresa. Porém, é sempre importante inovar buscando novas técnicas que possam realmente trazer bons resultados à empresa. É possível encontrar muito material publicado sobre as técnicas, mas provavelmente, pouco seja realmente implementável em um projeto.

Não existe uma técnica universal criada sob medida para cada processo de levantamento de requisitos. Existem, porém, técnicas que são mais indicadas ou apropriadas para uma determinada situação ou sistema. A equipe de desenvolvimento, em último caso, deve encontrar meio para selecionar a que for mais conveniente e demonstrar estes meios foi a principal finalidade deste trabalho. Com isto, os desenvolvedores poderão desenvolver sistemas que melhor atendam as necessidades dos usuários, evitando aumento de custos e atraso de cronogramas. Neste ponto específico, acreditamos que a proposta do trabalho oferece a criação de condições e mecanismos para melhorar o processo de levantamento de requisitos contribuindo para que os possíveis erros no desenvolvimento de sistemas fiquem dentro de uma faixa aceitável e tolerável, tendo como resultado final, *softwares* com muito mais qualidade.

4.2 FUTURAS PESQUISAS

Com base neste trabalho, alguns outros poderão ser desenvolvidos para tentar resolver problemas no desenvolvimento de sistema e melhorar a qualidade final do *software*. Algumas das sugestões são:

- Como a comunicação e aspectos de caráter humano influenciam no levantamento de requisitos;

- As técnicas mais eficiente para desenvolvimento de sistemas que necessitam de alta *performance*; e
- Como e quais ferramentas CASE podem auxiliar e melhorar os aspectos relacionados à coleta de requisitos.

REFERÊNCIAS

ALVES, Carina Frota; GUEDES, Lucy Valença; PINTO, Rosa Candida. *Um Visão do Processo de Elicitação de Requisito*, Nov. 1999. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Pernambuco, 1999.

BATISTA, EDINELSON A. *Uma taxonomia facetada para técnicas de elicitação de requisitos*. Trabalho de conclusão de curso (Mestre em Computação na área de Engenharia de Computação) – UNICAMP, São Paulo, 2003.

BEDANI, Janaina. Artigo de Engenharia de software – Técnicas para levantamento de requisitos, São Paulo, mar. 2011.

BELGAMO, A. e MARTINS, L. E. G. Congresso de Iniciação Científica. *Um Estudo Comparativo sobre as Principais Técnicas de Elicitação de Requisitos do Software*. Out. 2000, Piracicaba, UNIMEP/CNPq.

BRAGA, Fabrício Portes. Técnicas de Levantamento de Requisitos. Artigo apresentado à diretoria de ensino de pós-graduação, pesquisa e extensão, da UPIS – União Pioneira de Integração Social, Brasília-DF em 30 abr. 2008.

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 Workshop em Engenharia de Requisitos, Buenos Aires, Argentina, 2001.

CHAOS Manifesto 2011. Standish Group, CHAOS Summary 2010, Extreme CHAOS 2001. Disponível em: <<http://blog.mhavila.com.br/2010/06/17/sucessos-e-falhas-em-projetos-de-ti>>. Acessado em: 15 dez. 2012.

COSTA, Osvaldo Wilson, *JAD Joint Application Design*. 2º Ed. Rio de Janeiro: Liv. E Editora Infobook, 1994.

DAVIS, William. *Análise e Projeto de Sistemas uma Abordagem Estruturada*. LTC - Livros Técnicos e Científicos Editora S.A. Rio de Janeiro, Brasil. 1987.

FALBO, Ricardo de Almeida. *Engenharia de Requisitos*. Trabalho de conclusão de curso (Mestre em Computação na área de Engenharia de Computação) – UFES – Universidade Estadual do Espírito Santo, Espírito Santo, 2012.

FURLAN, Cláudio J. G. *Gestão de escopo e tempo*. São Paulo, 2012. Apostila do Curso de Pós-Graduação em Gestão de Projetos da Universidade Mackenzie.

KENDALL, K.E., KENDALL, J.E. *Systems Analysis and Design*, 8th Edition, Camden, New Jersey, Prentice Hall, 2010.

PÁDUA, Wilson de. *Engenharia de software: Fundamentos, métodos e padrões*. 3º Edição, Rio de Janeiro: LTC, 2011. Capítulo 6: Requisitos.

PMBOK - *Um guia do conhecimento em gerenciamento de projetos*. PMI – Project Management Institute ed. 4º Edição PMI, 2008.

PRESSMAN, Roger S. Engenharia de software. São Paulo: MAKRON Books, 1995. Capítulo 6: Princípios fundamentais de requisitos.

QREN, Observatório. Manual Técnico - A Avaliação do Desenvolvimento Socioeconômico II - A Recolha de Dados: Técnicas de Observação. Disponível em: <<http://www.observatorio.pt/>>. Acesso em: 10 fev. 2013

SILVA, Thais Mara. *Uma proposta de novos estudos para melhorarias do processo de elicitação de requisitos para uma cooperativa de software livre*. 2009. Trabalho de conclusão de curso (Graduação em Ciência da Computação) - Universidade Federal de Lavras, Minas Gerais, 2009.

SOARES, Emiliane Silva. *SWRequirement: Uma proposta de integração de técnicas de elicitação de requisitos ao processo de levantamento e análise de requisitos*, 2007. Trabalho de conclusão de curso (Graduação em Ciência da Computação) - Universidade Federal de Lavras, Minas Gerais, 2007.

SOMMERVILLE, Ian Engenharia de software 6º Ed. São Paulo: Pearson Addison Wesley, 2003.

VARGAS, Ricardo Viana. Gerenciamento de projetos: Estabelecendo diferenciais competitivos. 6º Ed. São Paulo: Brasport, 2005. Capítulo 4: O processo de coletar os requisitos.

WOOD, J. Silver, D. Joint Application Development, 2º edição, John Wiley & Sons, 1996.

XAVIER, Carlos Magno da Silva. Gerenciamento de projetos: como definir e controlar o escopo do projeto. 2º Ed. São Paulo: Saraiva, 2009. p. 70-78.