

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA - DAELN  
GESTÃO DA TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO**

**AFRO NETTO NUNES FARIA**

**O USO DE TÉCNICAS DE LEVANTAMENTO DE REQUISITOS NO  
MERCADO DE TRABALHO ATUAL**

**MONOGRAFIA**

**CURITIBA**

**2016**

**AFRO NETTO NUNES FARIA**

**O USO DE TÉCNICAS DE LEVANTAMENTO DE REQUISITOS NO  
MERCADO DE TRABALHO ATUAL**

Monografia apresentada como requisito parcial à obtenção do título de Especialista no MBA- Gestão da Tecnologia da Informação e Comunicação, do Departamento Acadêmico de Eletrônica - DAELN, da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dra. Rosangela de Fatima Stankowitz

**CURITIBA**

**2016**



Ministério da Educação  
Universidade Tecnológica Federal do Paraná  
Nome do Campus

Nome da Diretoria / Coordenação / Departamento  
Preencher com o Nome do Curso



## TERMO DE APROVAÇÃO

**(A SER FORNECIDA PELA SECRETARIA DO CURSO)**

O USO DE TÉCNICAS DE LEVANTAMENTO DE REQUISITOS NO MERCADO DE  
TRABALHO ATUAL

por

AFRO NETTO NUNES FARIA

Esta **Monografia** foi apresentado a em 10 de Outubro de 2016 como requisito parcial para a obtenção do título de Especialista em GESTÃO DA TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO . O(a) candidato(a) foi arguido(a) pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

---

Prof. Dra. Rosangela de Fatima Stankowitz  
Prof.(a) Orientador(a)

---

Prof. Dr. Alexandre Jorge Miziara  
Membro titular

- O Termo de Aprovação assinado encontra-se na Coordenação do Curso -

Dedico este trabalho à minha família,  
pelos momentos de ausência. À  
professora Prof. Dra. Rosangela de  
Fatima Stankowitz pela paciência na  
orientação e incentivo que tornaram  
possível a conclusão desta monografia.

## **AGRADECIMENTOS**

Certamente estes parágrafos não irão atender a todas as pessoas que fizeram parte dessa importante fase de minha vida. Portanto, desde já peço desculpas àquelas que não estão presentes entre essas palavras, mas elas podem estar certas que fazem parte do meu pensamento e de minha gratidão.

Agradeço a minha orientadora Prof. Dra. Rosangela de Fatima Stankowitz, pela sabedoria com que me guiou nesta trajetória.

Agradeço também pelos momentos descontraídos e sábios proporcionados pelo Prof. Dr. Alexandre Jorge Miziara.

A minha esposa sempre presente, amável e prestativa que foi o alicerce principal de força e resiliência que tonaram o presente trabalho possível.

A minha mãe, exemplo de superação e conquistas que me guiam a cada desafio.

Ao meu Padrasto/Pai Carlos Alberto Assis presente em todas fases de minha vida.

Ao meu irmão, eterno companheiro e amigo.

Gostaria de deixar registrado também meus agradecimentos à Secretaria do Curso, composta por Reneudo de Albuquerque Junior sempre prestativo e competente dando todo apoio a seu alcance.

Enfim, a todos os que por algum motivo contribuíram para a realização desta pesquisa.

## RESUMO

NETTO NUNES FARIA, Afro. **O Uso das Técnicas de Levantamento de Requisitos no Mercado Atual**. 2016. 61 folhas. Trabalho de Conclusão de Curso de Especialização MBA - Gestão da Tecnologia da Informação - Universidade Tecnológica Federal do Paraná. Curitiba, 2016.

A fase de levantamento de requisitos é uma das mais importantes fases do desenvolvimento de software. A engenharia de requisitos é composta de inúmeras técnicas de elicitação de requisitos as quais são ferramentas indispensáveis para os profissionais do mercado de tecnologia da informação. Identificar as principais técnicas de levantamento de requisitos utilizadas pelos analistas no desenvolvimento de um sistema é o objetivo deste trabalho. Para isso, foi feito um levantamento bibliográfico seguido de aplicação de questionário para cinquenta profissionais envolvidos com desenvolvimento de software. Os principais resultados revelam que as principais técnicas de levantamento de requisitos utilizadas são prototipagem e casos de uso. Revelam também o fato de que profissionais de empresas de pequeno porte fazem uso de um apanhado maior de técnicas em relação a aqueles que trabalham em empresas de grande porte. O contrário se dá quando a variável é o porte do cliente, isto é, quando o cliente atendido pela equipe de desenvolvimento apresenta um porte grande, mais técnicas de elicitação de requisitos são aplicadas. Por conseguinte, a pesquisa permitiu inferir que diversos fatores influenciam na frequência com a qual as técnicas são utilizadas. O entendimento de tais fatores contribui de forma a proporcionar base para compreensão de critérios de aperfeiçoamento dos cursos relacionados à tecnologia da informação, dos ambientes e costumes empresariais e da relação cliente x empresa desenvolvedora de software. Esta compreensão pode melhorar a qualidade dos softwares e diminuir de custos por meio do uso adequado e suficiente da engenharia de requisitos e suas especificações.

**Palavras-chave:** Engenharia de Requisitos. Engenharia de Software. Técnicas de Levantamento de Requisitos. Requisitos de Software.

## ABSTRACT

NETTO NUNES FARIA, Afro. The Use of Requirements Techniques in Today's Market. 2016. 61 pages. MBA Specialization Course Completion - Information Technology Management - Federal Technological University of Paraná. Curitiba, 2016.

The requirements elicitation phase is one of the most important phases of software development. The requirements engineering is composed of numerous techniques of requirements elicitation which are indispensable tools for professionals in the information technology market. Identify key requirements elicitation techniques used by analysts in the development of a system is the objective of this work. For this, it made a literature review followed by a questionnaire to fifty professionals involved with software development. The main results shows that the main techniques are prototyping and use cases. It also reveled the fact that small business professionals usually apply a larger overview of techniques compared to those working in largers companies. The opposite occurs when the variable is the client's size, more elicitation techniques requirements are applied. Therefore, research also has shown that several factors influence the frequency with which the techniques are used. Understanding these factors helps to provide the basis for understanding improvement criteria for courses related to information technology, the environment and business customs and customer relationship x developer software company. This understanding can improve software quality and reduce costs through the appropriate and adequate use of engineering requirements and specifications.

**Keywords:** Requirements Engineering. Software Engineering. Requirements Survey techniques. Software requirements.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Ciclo de vida do projeto de software .....	19
Figura 2 - Tipos de definição de requisitos.....	22
Figura 3 - Tipos de requisitos não funcionais.....	25
Figura 4 - Critérios de medição de usabilidade .....	26
Figura 5 - Fatores de desempenho .....	28
Figura 6 - Tipos de Segurança .....	30
Figura 8 - Exemplo Gráfico - Caso de Uso.....	36
Figura 9 - Fluxos de Eventos de um caso de uso .....	38
Figura 10 - Atributos de perguntas fechadas e abertas.....	42
Figura 11 - Uso de questões subjetivas e objetivas em questionários .....	43
Figura 12 - O modelo de ciclo de vida em Espiral .....	47
Figura 13 - Técnicas de Levantamento de Requisitos mais utilizadas .....	51
Figura 14 - Pontuação Média por Formação Acadêmica .....	53
Figura 15 - Pontuação Média por Cargo .....	56
Figura 16 - Pontuação Média pelo Porte da Empresa.....	57
Figura 17 - Pontuação Média Pelo Porte do Cliente .....	58
Quadro 1 - Cenário Atual de Desenvolvimento .....	17
Quadro 2 - Custo proporcional de manutenção de software .....	20
Quadro 3 - Fluxo de Exceções do caso de uso Abrir Conta.....	39
Quadro 4 - Pontuação de Frequência .....	49
Quadro 5 - Média ponderada: Uso das Técnicas de Levantamento de Requisitos...52	
Tabela 1 - Frequência de Uso de Acordo com a Formação Acadêmica .....	54
Tabela 2 - Pontuação Final por Formação Acadêmica .....	54



## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>13</b>
1.1 JUSTIFICATIVA.....	14
1.2 ESTRUTURA DO TRABALHO .....	14
<b>2 ENGENHARIA DE REQUISITOS.....</b>	<b>16</b>
<b>3 REQUISITOS DE SOFTWARE .....</b>	<b>21</b>
3.1 REQUISITOS FUNCIONAIS.....	22
3.2 REQUISITOS NÃO FUNCIONAIS .....	23
3.2.1 Requisitos de Produto.....	26
3.2.2 Usabilidade .....	26
3.2.3 Manutibilidade.....	27
3.2.4 Confiabilidade .....	27
3.2.5 Desempenho.....	28
3.2.6 Portabilidade.....	29
3.2.7 Reusabilidade .....	29
3.2.8 Segurança .....	30
3.3 LEVANTAMENTO DE REQUISITOS.....	31
3.4 TÉCNICAS DE LEVANTAMENTO DE REQUISITOS.....	32
3.4.1 Entrevistas .....	33
3.4.2 Cenários .....	34
3.4.3 Casos de Uso .....	35
3.4.3.1 Propriedades do Caso do Uso .....	37
3.4.3.2 Fluxo de eventos.....	37
3.4.3.3 Fluxos de exceção .....	39
3.4.4 Etnografia .....	39
3.4.5 Validação .....	40
3.4.6 Questionários.....	41
3.4.6.1 Elaboração do Questionário.....	41
3.4.6.2 Aplicação do Questionário .....	43
3.4.7 Workshops.....	44
3.4.7.1 JAD – <i>Joint Application Development</i> – (Desenvolvimento conjunto de aplicação).....	45
3.4.7.2 Brainstorming.....	46
3.4.8 Prototipagem.....	46
<b>4 PROCEDIMENTOS METODOLÓGICOS DA PESQUISA .....</b>	<b>48</b>
4.1 AMOSTRA .....	48
4.2 COLETA DE DADOS .....	48
4.2.1 Tratamento dos dados .....	49
<b>5 ANÁLISE E DISCUSSÃO DOS RESULTADOS .....</b>	<b>51</b>

5.1 TÉCNICAS DE LEVANTAMENTO DE REQUISITOS.....	51
5.2 FORMAÇÃO ACADÊMICA .....	53
5.3 CARGO .....	56
5.4 PORTE DA EMPRESA .....	57
5.5 PORTE DO CLIENTE .....	58
<b>6 CONCLUSÃO.....</b>	<b>59</b>
6.1 OBJETIVOS GERAL E ESPECÍFICOS .....	59
6.2 SUGESTÕES DE TRABALHOS FUTUROS.....	60
<b>REFERÊNCIAS.....</b>	<b>62</b>
<b>APÊNDICE A - Questionário de Pesquisa .....</b>	<b>65</b>

## 1 INTRODUÇÃO

Em se tratando de sistemas de informação, requisitos são, basicamente, as necessidades que um sistema deve atender e os serviços esperados pelos seus futuros usuários. Refletem a expectativa do cliente em relação às funcionalidades que o produto (sistema) final deve conter. A fase de levantamento de requisitos é aquela na qual o profissional de tecnologia da informação é responsável por reunir todas as ambições do cliente, entendê-las e documentá-las.

No entanto, trabalhar com requisitos vai além da primeira fase do projeto. Os processos de engenharia de requisitos visam também adiantar-se em relação às possíveis alterações de escopo. Supõe-se que por meio de entrevistas e apresentação de protótipos é possível instigar o futuro usuário a encontrar funcionalidades necessárias não previstas em uma análise prévia.

É de conhecimento geral que em praticamente todos os sistemas os requisitos mudam. As pessoas envolvidas desenvolvem uma melhor compreensão do que querem do software a cada entrega parcial, a organização que compra o sistema também muda da mesma forma que modificações são feitas no hardware, no software e no ambiente organizacional do sistema. (SOMMERVILLE, 2011).

O objetivo geral deste estudo é identificar as principais técnicas de levantamento de requisitos utilizadas pelos analistas no desenvolvimento de um sistema. Para atingir este Objetivo, estabeleceram-se os seguintes objetivos específicos:

- I. Realizar um levantamento bibliográfico sobre as técnicas de levantamento de requisitos;
- II. Elaborar um questionário listando as técnicas de levantamento de requisito;
- III. Aplicar o questionário a um conjunto de profissionais que tenham participação na fase de levantamento de requisitos;
- IV. Entender se existem fatores (e caso existam, quais são eles) que influenciam no uso adequado e satisfatório das técnicas de levantamento de requisitos apresentadas na revisão bibliográfica desta monografia.

## 1.1 JUSTIFICATIVA

O interesse por este estudo surgiu da necessidade identificada pelo pesquisador de entender o contexto que envolve projetos de software bem-sucedidos em decorrência de um trabalho bem elaborado e consistente na fase de levantamento de requisitos, bem como as consequências do contrário.

O estudo tem foco no profissional e se buscou identificar o perfil daqueles mais completos em se tratando de conhecimentos e uso de técnicas de levantamento de requisito. No entanto, vale ressaltar que não foram levados em conta fatores relacionados às empresas e seus modelos de trabalho, o que certamente também tem influência na qualidade da elicitação de requisitos de software.

É válido à comunidade de tecnologia da informação visto que dispõe a fase de elicitação dos requisitos como determinante no sucesso dos projetos de software, por conseguinte leva o profissional a olhar para si e à cultura de trabalho da empresa na qual está inserido de forma analítica e crítica.

## 1.2 ESTRUTURA DO TRABALHO

Este estudo está organizado em 6 capítulos conforme a descrição a seguir:

Capítulo 1 – Contextualização do tema juntamente com a apresentação do objetivo geral e objetivos específicos. Traz também a justificativa e a importância do estudo.

Capítulo 2 – Apresenta os conceitos da Engenharia de Requisitos e seus pormenores, assim como dados que comprovam a importância de sua aplicação.

Capítulo 3 – Aprofunda o entendimento de requisito de sistema apresentando conceitos chave, bem como as técnicas de elicitação de requisitos, e suas definições, provenientes de levantamento bibliográfico.

Capítulo 4 – Detalha os procedimentos metodológicos utilizados na pesquisa.

Capítulo 5 – Apresenta os dados coletados na aplicação do questionário analisando-os através do cruzamento destas informações.

Capítulo 6 – Considerações finais referentes ao objetivo geral e aos objetivos específicos, juntamente com propostas de possíveis trabalhos futuros que podem complementar este estudo.

## 2 ENGENHARIA DE REQUISITOS

A Engenharia de Requisitos é o ramo da Engenharia de Software que engloba as atividades relacionadas com a elicitação de requisitos de um sistema, aplicadas durante o ciclo de vida do software (KOTONYA; SOMMERVILLE, 1998).

Segundo Nuseibeh e Easterbrook (2000), a engenharia de requisitos é o processo para descobrir o propósito do sistema por meio da identificação dos *stakeholders* acompanhados de seus problemas e necessidades a serem atendidos. Em seguida, de forma organizada, é preciso documentar as informações visando a conveniência da análise durante o desenvolvimento do sistema, bem como o auxílio na comunicação entre as partes envolvidas. Os autores afirmam que a engenharia de requisitos é a fase inicial que busca levantar informações importantes no entendimento da organização até a especificação detalhada do software a ser implantado. Assim, envolve conteúdo técnico como também fatores organizacionais e sociais.

Engenheiros de software trabalham com clientes e usuários finais do sistema para obter informações sobre o domínio da aplicação, os serviços que o sistema deve oferecer, o desempenho do sistema, restrições de hardware e assim por diante (SOMMERVILLE, 2011).

Atualmente é de amplo conhecimento da comunidade da engenharia de software que a definição adequada dos requisitos de um sistema tem grande impacto na qualidade final do produto. Apesar de parecer uma tarefa simples, a elicitação de requisitos é um processo extremamente complicado, visto que frequentemente o usuário não sabe descrever seu papel no processo e pode deixar informações muito importantes de lado, ou talvez sejam relutantes ou incapazes de cooperar (DIESTE, 2008).

É normal que os processos de engenharia de requisitos sejam diferentes entre organizações ou até mesmo entre diferentes áreas de uma mesma empresa devido às características dos projetos. A definição adequada dos processos proporciona à organização inúmeros benefícios, pois uma boa descrição do mesmo viabilizará orientações adequadas de forma a reduzir a probabilidade de esquecimento ou de uma execução superficial. No entanto, não se pode falar em processo ideal ou definir algum e impô-lo a uma organização. O correto é que a

empresa inicie com um processo genérico e passe a adaptá-lo no decorrer de sua experiência buscando sempre moldá-lo de acordo com suas necessidades específicas (SOMMERVILLE & SAWYER, 1997).

Quando o levantamento de informações é feito de forma aprofundada e a documentação é bem elaborada, os requisitos implícitos são minimizados, além de aumentar consideravelmente as chances de os desenvolvedores entenderem com mais precisão as demandas dos usuários. Algumas técnicas e práticas ajudam na produção de um documento de requisito mais completo e preciso. O conjunto dessas técnicas é chamado de Engenharia de Requisitos (DE PÁDUA, 2000).

Entretanto, muitos clientes não entendem a necessidade das especificações de requisitos. Pior ainda, muitos desenvolvedores de software e gerentes também não. É uma situação complexa. Seria como querer resolver um problema sem escrever o enunciado, o que iria aumentar o risco de tentar resolver o problema errado. Por outro lado, é possível também a existência de requisitos que não correspondam às necessidades reais dos clientes e usuários. Esta falha de engenharia de requisitos indica que não foi feita uma análise do valor de cada requisito, do ponto de vista da missão que o produto deve cumprir (DE PÁDUA, 2000).

Ávila e Spínola (2008) definem três principais objetivos que a definição dos processos envolvidos no desenvolvimento do software busca entregar:

- Qualidade de software;
- Produtividade no desenvolvimento, operação e manutenção do sistema;
- Oferecer aos profissionais maiores condições para que possam controlar os prazos, custos e a qualidade do software.

Apesar disso o cenário do mercado atual de desenvolvimento não tem se aproximado satisfatoriamente do cenário idealizado pela engenharia de requisitos (ÁVILA & SPÍNOLA, 2008). O Quadro 1 traz dados que mostram como os custos de correções de problemas encontrados na fase de levantamento de requisitos são baixos, se comparados aos apontados na fase de desenvolvimento do código.

**Quadro 1 - Cenário Atual de Desenvolvimento**

	<b>% do Custo de Desenvolvimento</b>	<b>% dos erros introduzidos</b>	<b>% dos erros encontrados</b>	<b>Custo relativo de correção</b>
<b>Análise de Requisitos</b>	5	55	18	1

<b>Projeto</b>	25	30	10	1-1.5
<b>Codificação e Teste de Unidade</b>	50			
<b>Teste</b>	10	10	50	1-5
<b>Validação e Documentação</b>	10			
<b>Manutenção</b>		5	22	10-100

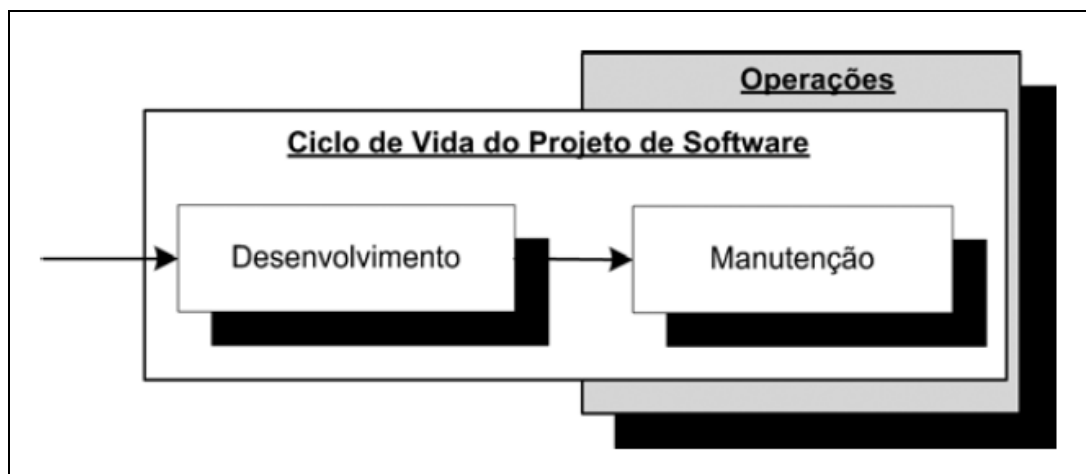
Fonte: Ávila e Spínola (2008)

Como demonstrado no Quadro 1, a área de levantamento e análise de requisitos recebe por volta de 5% dos orçamentos dos projetos, mesmo que seja responsável por mais da metade (55%) dos erros dos softwares. De acordo com Ávila e Spínola (2008) esses erros, se descobertos e tratados no início, ainda na fase de análise de requisitos, têm um custo significativamente baixo (custo relativo de correção igual a 1). No entanto, se não encontrados a tempo na fase de requisitos, são replicados nas fases de desenvolvimento (codificação) do projeto, na qual o custo de manutenção e correção passa a ser multiplicado em até 100 vezes (custo relativo de correção de 10 a 100) o custo inicial.

Estudos descritos na literatura revelam estatísticas contundentes sobre a importância da manutenção de software. Sutherland (1995), citado por Engholm Junior (2010), estimou que as organizações, somente nos Estados Unidos, gastaram com a manutenção de software aproximadamente 70 bilhões de dólares por ano. Neste mesmo país, o Bug do milênio gerou gastos de 8,3 bilhões de dólares ao governo durante cinco anos e 90 bilhões de dólares à Nokia com as correções preventivas (Engholm Junior, 2010),

Apesar dos custos de desenvolvimento de software apresentarem uma redução gradativa, a complexidade tem aumentado. O volume enorme de informações e de regras de negociação geram problemas como imprecisão na estimativa de prazos e custos, dificuldade na captação adequada de recursos e conseqüentemente problemas em atender as demandas. A engenharia de software atua oferecendo ferramentas e técnicas que visam aumentar a produtividade e qualidade dos sistemas (ENGHOLM JUNIOR, 2010).





**Figura 1 - Ciclo de vida do projeto de software**  
 Fonte: Engholm Junior (2010 p.26)

A engenharia de software não busca extinguir essa fase, mas sim diminuir ao máximo os seus custos. A Figura 1, encontrada em Engholm Junior (2010), divide o projeto em apenas duas fases principais, a de desenvolvimento e a de manutenção, as quais formam o ciclo de vida do sistema.

Engholm Junior (2010) classifica, ainda, a manutenção do software em três principais tipos:

- Corretiva: modificações que visam corrigir falhas;
- Evolutiva: relativas à inclusão de novas funcionalidades ou alterações relativas ao desempenho;
- Adaptativa: relacionada à adaptação do sistema a diferentes ambientes operacionais;

O autor explica que com base nos tipos de manutenção citados, as manutenções estão associadas a pelo menos um dos itens a seguir:

- Remoção de defeitos;
- Adição de funcionalidades no sistema;
- Alteração de funcionalidades;
- Evolução das funcionalidades;
- Ajustes;
- Atualização;
- Melhoria de desempenho.

Engholm Junior (2010) aponta alguns estudos que mostram que mais de 50% do tempo dedicado a manutenção dos sistemas é gasto pelos desenvolvedores na

tentativa de entender códigos mal ou não documentados. O autor traz ainda dados referentes aos custos de manutenção em diversos anos entre 1979 e 2000 representados no Quadro 2.

**Quadro 2 - Custo proporcional de manutenção de software**

Ano	Proporção de custos de manutenção de software	Definição
2000	>90%	Custo de software relacionado à manutenção e evolução de sistemas/custo total de software
1993	75%	Manutenção de software/orçamento de sistemas de informação (em 1000 empresas)
1990	>90%	Custo de software relacionado à manutenção e evolução de sistemas/custo total de software
1990	60-70%	Manutenção de software/orçamento total de operação de sistemas da informação ( <i>Management Information Systems – MIS</i> )
1988	60-70%	Manutenção de software/orçamento operacional do total de sistemas de informação gerencial ( <i>Management Information Systems – MIS</i> )
1984	65-75%	Esforço gasto na manutenção de sistemas/total disponível relacionado a engenharia de software
1981	>50%	Tempo gasto pela equipe de manutenção/tempo total (em 487 organizações)
1979	67%	Custos de manutenção/custo total de software

Fonte: ENGHOLM JUNIOR (2010 p.31)

O Quadro 2 mostra o aumento significativo dos custos de manutenção entre os anos de 1979 e 2000. Para Engholm Junior (2010), a explosão da alta dos custos originou-se do aumento da demanda e da sua complexidade aliadas à inexistência de técnicas apropriadas. Tais dificuldades deram origem ao termo *Crise do Software* que o autor descreve como papel da engenharia de software em relação à crise do software em que

A engenharia de software se concentra nos aspectos práticos da produção de um sistema de software sendo criada na tentativa de tratar a crise do software (...) realizando tratamento de engenharia sistemático e controlado no desenvolvimento de softwares (Engholm Junior, 2010, p. 35).

As atividades do processo de engenharia de requisitos proposto nesta pesquisa são discutidas na próxima seção de forma mais detalhada.

### 3 REQUISITOS DE SOFTWARE

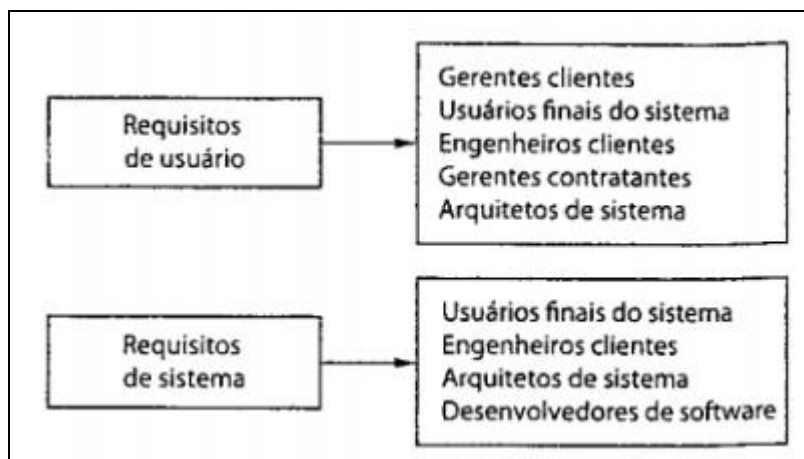
O que exatamente um sistema de software deve fazer? Quais os serviços e funcionalidades precisam ser oferecidos? Deve estar em conformidade com quais restrições? De acordo com Sommerville (2011), as respostas para tais perguntas são, basicamente, requisitos do sistema. O autor afirma que estes requisitos retratam as necessidades dos clientes para serem atendidas pelo sistema.

De Pádua (2000, p. 13) faz uso do termo característica para definir requisito dizendo que: “Os requisitos são as características que definem os critérios de aceitação de um produto”. Da mesma forma, Ávila e Spínola (2008) afirmam que o requisito é uma característica do sistema ou de algo que ele deverá estar apto a realizar para alcançar seus objetivos. Os autores complementam com as seguintes considerações:

- Requisitos são explicações das restrições e atribuições do sistema;
- Um requisito é uma funcionalidade que o software possui para solucionar determinado problema no mundo real;
- Uma regra que deve ser atendida de forma a satisfazer um contrato, padrão, documento entre outras formalidades.

Em se tratando do produto final a ser entregue, a análise dos requisitos possibilita a concordância entre a empresa desenvolvedora do software e o cliente por meio da geração de documentos que permite a validação do software de acordo com critérios estabelecidos na fase inicial do projeto, além de assegurar uma redução dos custos com retrabalho (ÁVILA; SPINOLA, 2008).

A Figura 2 mostra quais os grupos de envolvidos no projeto os requisitos de usuários e requisitos de sistemas são voltados, de acordo com Sommerville, 2011.



**Figura 2 - Tipos de definição de requisitos**  
**Fonte: Sommerville (2011, p. 59)**

Para Sommerville (2011), os requisitos de usuários geralmente são voltados aos gerentes de projetos, escritos em uma linguagem natural, expondo quais necessidades devem ser atendidas. São usados, muitas vezes, em licitações. Já os requisitos de sistema são escritos em uma linguagem técnica que define exatamente o que o sistema deve fazer, voltado aos arquitetos e desenvolvedores. A Figura 2 traz esses dois tipos de requisitos e seus respectivos públicos-alvo.

A Engenharia de Requisitos divide os Requisitos de Software em dois grandes grupos: funcionais e não-funcionais. Estes são descritos de forma detalhada no próximo tópico.

### 3.1 REQUISITOS FUNCIONAIS

Os requisitos funcionais podem descrever o que o sistema deve fazer de forma abrangente e abstrata ou de forma detalhada, de acordo com o público para o qual está voltado. Caso seja para que os usuários entendam suas funcionalidades, a descrição será simplista. Porém quando o alvo é a equipe de desenvolvimento, os requisitos funcionais abrangem informações e detalhes mais específicos como suas funções, entradas e saídas e em como o sistema deve se comportar e determinadas situações. (SOMMERVILLE, 2011).

Ávila e Spínola (2008, p 48) definem os requisitos funcionais resumindo a funcionalidades que o software deve apresentar e citam os seguintes exemplos:

- “O software deve permitir o cadastro de clientes”;

- “O software deve permitir a geração de relatórios sobre o desempenho de vendas no semestre”;
- “O software deve permitir o pagamento das compras através do cartão de crédito”.

Sommerville (2011) exemplifica requisitos funcionais fazendo uso de um sistema da área da saúde chamado MHC-PMS, utilizado para manter informações de pacientes com deficiências mentais:

- Um usuário deve ser capaz de pesquisar as listas de agendamentos para todas as clínicas;
- O sistema deve gerar a cada dia, para cada clínica, a lista dos pacientes para as consultas daquele dia;
- Cada membro da equipe que usa o sistema deve ser identificado apenas por seu número de oito dígitos.

Outra definição para requisitos funcionais é a de Mendes Silva Filho (2008), que além de focar no usuário traz os termos entradas e saídas:

Um requisito de sistema de software especifica uma função que o sistema ou componente deve ser capaz de realizar. Estes são requisitos de software que definem o comportamento do sistema, ou seja, o processo ou transformação que componentes de software ou hardware efetuam sobre as entradas para gerar as saídas. Estes requisitos capturam as funcionalidades sob o ponto de vista do usuário. (Mendes Silva Filho, 2008, p. 5).

### 3.2 REQUISITOS NÃO FUNCIONAIS

Os requisitos não funcionais não estão relacionados aos serviços ou necessidades que o sistema deve atender, mas sim a propriedades específicas de funcionamento, como disponibilidade, tempo de resposta, desempenho, segurança e confiabilidade, entre outras características. Estes requisitos são, na sua maioria, mais importantes e críticos que os requisitos funcionais que, em determinadas situações, permitem aos usuários encontrar maneiras de contornar uma função do sistema que não atenda suas necessidades (SOMMERVILLE, 2011).

Ávila e Spínola (2008, p 48) classificam os requisitos não funcionais como “condições que o software deve atender ou qualidades específicas que o software deve ter.” E vão além: “Em vez de informar o que o sistema fará, os requisitos não-

funcionais colocam restrições no sistema”. Como exemplo de restrições provenientes de requisitos não-funcionais, os autores citam as seguintes condições:

- O sistema deve ser compatível com os navegadores Internet Explorer versão 5.0 ou superior e Firefox 1.0 ou superior;
- O tempo de consulta não pode ser maior que cinco segundos.

Lauesen (2002) separa os requisitos não funcionais em três principais frentes: performance, usabilidade e manutibilidade.

Performance refere-se ao quão eficiente o sistema deve ser com o hardware. Em outras palavras: quão rápido o sistema deve ser para executar determinada tarefa. E mais, quão acurados devem ser os resultados e quanto de dados deve ser possível armazenar.

Usabilidade define o quão amigável o sistema deve ser ao usuário. Quanto tempo de aprendizado é necessário para fazer uso deste. Também pode constar o quão eficiente ele é no uso diário.

Por fim, a manutibilidade do sistema aponta o quão fácil de executar correções ou inserir novas funções deve ser.

Seguindo uma outra linha de raciocínio, Sommerville (2011) qualifica os requisitos não funcionais da seguinte forma:

- Requisitos de produto - definem ou delimitam o comportamento do software. Refere-se, por exemplo, à velocidade, à confiabilidade, taxa de falhas aceitável, facilidade de uso (usabilidade).
- Requisitos organizacionais - são derivados das políticas das empresas do cliente e do desenvolvedor. Pode-se citar exemplos como: linguagem usada, procedimentos e normas de desenvolvimento e processos.
- Requisitos externos - provenientes de requisitos externos ao sistema, como normas regulamentares, leis, ética, dentre outros.

Já De Pádua (2000) separa os requisitos em tipos, sendo os três abaixo:

- Explícitos são descritos em um documento de especificação de requisitos;
- Normativos são provenientes de leis, regras, padrões, entre outras diversas normas que o sistema deve sujeitar-se;
- Implícitos são referentes às expectativas dos clientes e usuários, apesar de estas não serem documentadas.

Os requisitos não funcionais, diferente dos funcionais, não dizem o que o sistema deve fazer, mas sim como ele fará. Por conseguinte, acaba por ter relação direta com a arquitetura assumindo o papel de critérios a serem satisfeitos, motivo pelo qual também são denominados atributos de qualidade. (MENDES SILVA FILHO, 2008).

Mendes Silva Filho (2008) traz em detalhes uma adaptação de Sommerville (2011) dos tipos de requisitos não funcionais adicionando alguns requisitos de produto a mais, como pode ser observado na Figura 3.

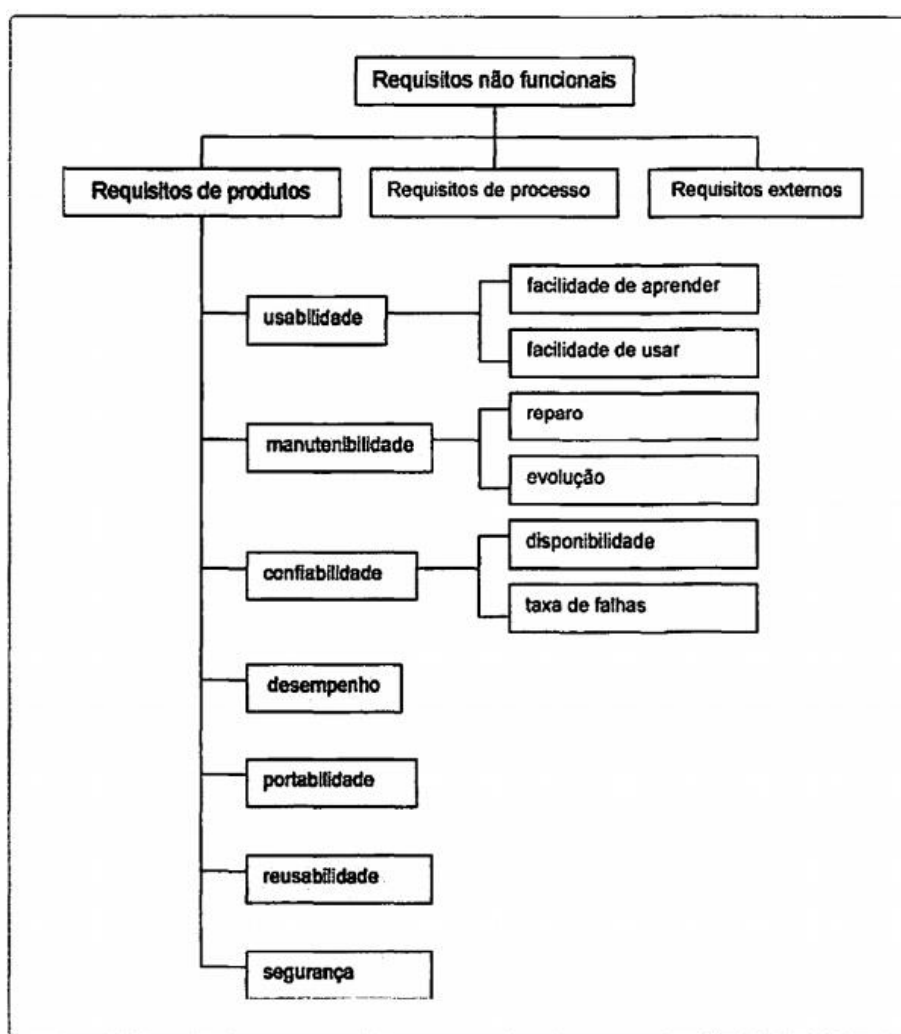


Figura 3 - Tipos de requisitos não funcionais  
Fonte: Mendes Silva Filho (2008 p. 6)

Os tópicos a seguir trazem com mais detalhes os tipos de requisitos não-funcionais presentes na Figura 3.

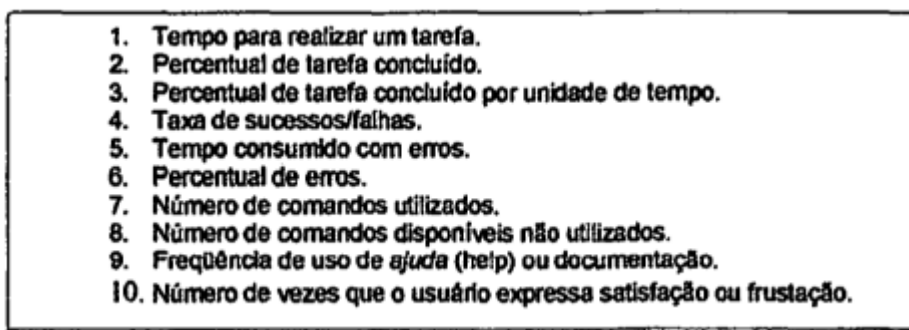
### 3.2.1 Requisitos de Produto

Os requisitos de produto são classificados como não-funcionais pois tratam exclusivamente das qualidades do sistema, assim implicam diretamente na determinação e construção da arquitetura. Mendes Silva Filho (2008) descreve, com base em Sommerville (2014), as subdivisões dos requisitos de produtos.

### 3.2.2 Usabilidade

Requisito aplicado a sistemas nos quais uma iteração entre seres humanos e sistema. Refere-se a quão fácil é o aprendizado e uso do software. Pode-se usar o adjetivo agradável também para classificar um sistema com boa usabilidade.

Existem critérios uteis à medição da usabilidade de um sistema. Mendes Silva Filho (2008) as enumera na figura a seguir:

- 
1. Tempo para realizar um tarefa.
  2. Percentual de tarefa concluído.
  3. Percentual de tarefa concluído por unidade de tempo.
  4. Taxa de sucessos/falhas.
  5. Tempo consumido com erros.
  6. Percentual de erros.
  7. Número de comandos utilizados.
  8. Número de comandos disponíveis não utilizados.
  9. Frequência de uso de ajuda (help) ou documentação.
  10. Número de vezes que o usuário expressa satisfação ou frustração.

**Figura 4 - Critérios de medição de usabilidade**  
Fonte: Mendes Silva Filho (2008 p. 6)

Para averiguação da qualidade do sistema quanto à usabilidade é necessário a criação de diversos cenários, analisando, por exemplo, se as informações são mostradas ao usuário nos momentos adequados, da mesma forma que se espera que as informações provenientes da iteração do cliente sejam encaminhadas corretamente (MENDES SILVA FILHO, 2008).



### 3.2.3 Manutibilidade

O termo manutibilidade é abrangente e se refere a qualquer correção ou alteração no sistema tanto na fase de desenvolvimento quanto após a entrega.

Um sistema com boa manutibilidade geralmente possui sua arquitetura separada em vários componentes. Desse modo, quando um reparo é necessário, a possibilidade de uma correção gerar problemas em outras partes do sistema é significativamente menor. Essa característica é chamada de modularidade.

Outro ponto importante da manutibilidade refere-se ao fato do arquiteto do sistema estar atento a possibilidade de evolução do sistema ao longo tempo. Se isso não ocorre aumentam as chances de futuras manutenções alavancarem outros problemas.

### 3.2.4 Confiabilidade

Segundo Mendes Silva Filho (2008), a confiabilidade como requisito não funcional remete a probabilidade de o sistema apresentar problemas durante certo período de tempo, dessa forma pode ser quantificada estatisticamente. Esse requisito de produto é visto como uma característica relacionada à qualidade de software, visto que implica averiguar se o sistema executará as tarefas como esperado.

Mendes Silva Filho (2008) enumera algumas métricas utilizadas para certificar a confiabilidade de um sistema:

- Disponibilidade – Medida que trata de “quão disponível o sistema estaria para uso, isto é, quão disponível o sistema estaria para efetuar um serviço solicitado por algum usuário” (MENDES SILVA FILHO, 2008, p.8);
- Taxa de ocorrência de falha - refere-se à frequência na qual o software apresenta problemas na execução de uma funcionalidade esperada pelo usuário;
- Probabilidade de falha durante fase operacional – “Esta é uma medida da probabilidade que o sistema irá comportar-se de maneira inesperada quando em operação. Esta métrica é de suma importância em sistemas críticos que requerem uma operação contínua” (MENDES SILVA FILHO, 2008, p.8);

- Tempo médio até ocorrência de falhas – essa métrica é responsável por analisar o tempo médio entre falhas. É importante pois mede o tempo que o sistema fica operacional sem apresentar problemas.

### 3.2.5 Desempenho

Essa métrica é bastante importante pelo fato de ter relação direta com a usabilidade do sistema. Operações podem deixar de ocorrer devido a uma lentidão do sistema, por exemplo. Diversos fatores implicam queda de desempenho, entre elas pode-se citar o espaço em disco e memória necessário para rodar e executar as tarefas de uma aplicação.

A Figura 5 mostra como o desempenho pode ser decomposto:

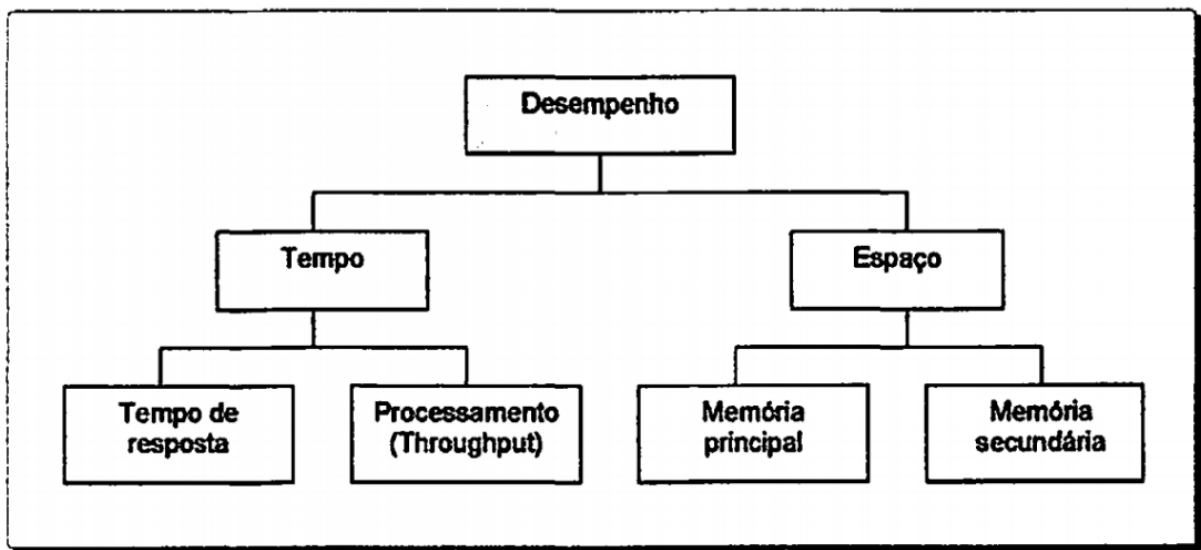


Figura 5 - Fatores de desempenho  
Fonte: Mendes Silva Filho (2008 p. 9)

- Resposta – define o tempo de resposta aceitável para que o sistema realize uma atividade;
- Processamento – define a quantidade mínima de dados que deveriam ser processados num determinado período de tempo;
- Temporização – “Este tipo de requisito especifica quão rapidamente o sistema deveria coletar dados de entrada de sensores antes que outras leituras de

dados de entrada, feitas posteriormente, sobrescrevam os dados anteriores” (MENDES SILVA FILHO, 2008, p.10);

- Espaço – refere-se ao uso máximo da memória do hardware pelo sistema. Pode ser separado em uso da memória principal e da memória secundária.

### 3.2.6 Portabilidade

Mendes Silva Filho (2008) diz que a portabilidade como requisito não funcional de produto refere-se à habilidade do software em ser executado em diferentes plataformas. Entende-se aqui plataforma tanto como um ambiente de software como um sistema operacional por exemplo, como também diferentes plataformas de hardwares, como computadores e celulares.

A portabilidade de um componente ou sistema de software é proporcional à quantidade de esforço necessário para que funcione num novo ambiente. Se uma quantidade menor de esforço é exigida quando comparada ao trabalho de desenvolvimento, então o sistema é dito portátil (MENDES SILVA FILHO, 2008, p.10).

### 3.2.7 Reusabilidade

Essa característica remete em fazer uso de componentes de softwares existentes na construção novos sistemas, visando minimizar esforço e custos. O reuso, segundo Mendes Silva Filho (2008), pode ser classificado em três frentes:

- Aplicação - quando a aplicação inteira é reutilizada;
- Objetos ou módulos - os subsistemas mais importantes poderiam ser reutilizados;
- Funções - quando componentes com funções específicas são reutilizadas. Por exemplo uma função matemática.

Quando uma arquitetura é pensada de forma a promover qualidade através da criação de um software bastante modularizado, indiretamente tem-se como resultado um sistema com componentes reutilizáveis.

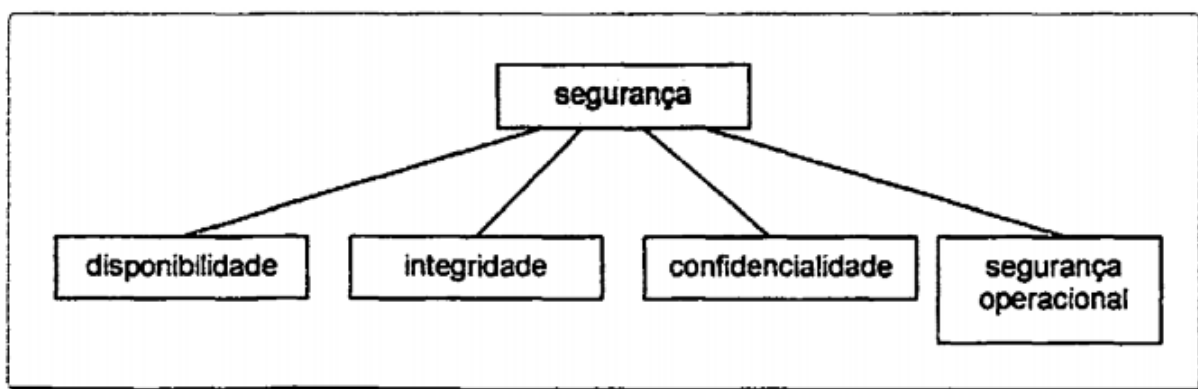
O uso de bibliotecas de componentes facilita no desenvolvimento de projetos com baixo acoplamento, além de facilitar o trabalho do desenvolvedor que terá acesso a implementações de código já testadas.

### 3.2.8 Segurança

A segurança de um sistema remete à probabilidade de um acesso não autorizado não ser barrado ou de dados se perderem. Mendes Silva Filho (2008, p. 11) complementa afirmando: “à medida que os sistemas de software se tornam distribuídos e conectados a redes externas, os requisitos de segurança vão se tornando cada vez mais importantes. ”.

Mendes Silva Filho (2008) exemplifica quatro requisitos de segurança:

- Acesso a aplicação e determinadas funcionalidades de acordo com as permissões do usuário autenticado;
- Controle das funcionalidades disponíveis aos usuários feita apenas por um administrador;
- Backup dos dados do sistema a cada 24 horas e armazenamento preferencialmente em local diferente da aplicação;
- Transferência de dados entre o servidor e o cliente criptografada.



**Figura 6 - Tipos de Segurança**  
Fonte: Mendes Silva Filho (2008 p. 11)

Mendes Silva Filho (2008) também traz o requisito de segurança separado em ênfases, como mostra a Figura 6, as quais são detalhadas com mais precisão abaixo.

- Disponibilidade - trata da segurança do sistema contra uma possível interrupção dos serviços;
- Integridade - busca assegurar que acessos ou atualização não permitidos e não ocorram;
- Confidencialidade - garantir que informações não autorizadas sejam reveladas;
- Segurança operacional - remete a fase considerada para o sistema em uso.

### 3.3 LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos diz respeito à fase inicial do processo de engenharia de requisitos e engloba os trabalhos de conhecimento e elicitação dos requisitos. Nessa fase, especialistas de domínio, clientes, futuros usuários interagem visando compreender a organização a nível de processos, necessidades, fraquezas dos softwares atuais, possíveis melhorias e restrições. Trata-se de um trabalho complexo o qual não se limita ao desejo do cliente, mas sim estudar de forma acurada a organização bem como o domínio da aplicação (KOTONYA; SOMMERVILLE, 1998).

Nesse contexto de descoberta, quatro dimensões se destacam:

- Domínio da aplicação: refere-se ao entendimento global da área na qual o sistema irá ser implantado;
- Problema a ser solucionado: entendimento do problema a ser solucionado por meio da implantação do sistema;
- Contexto do negócio: entender de que forma o sistema irá afetar a organização na qual será implantado;
- Necessidades e restrições dos envolvidos: entendimento do papel de cada grupo envolvido no desenvolvimento do sistema, suas respectivas necessidades a serem atendidas juntamente com as restrições a serem respeitadas.

### 3.4 TÉCNICAS DE LEVANTAMENTO DE REQUISITOS

Para a etapa de levantamento de requisitos é necessária a aplicação de uma ou mais técnicas visando à captura de informações sobre o produto final desejado e, principalmente, das necessidades dos clientes/usuários.

Kendall e Kendall (2010) separam os métodos de levantamento de requisitos em dois grupos: métodos interativos e métodos não obstrutivos. Os métodos interativos são aqueles que necessitam da iteração com os *stakeholders*, como é o caso de entrevistas e workshops de requisitos, por exemplo. Os métodos não obstrutivos são aqueles que não interferem no trabalho dos membros das organizações. Como exemplo tem-se o estudo de documentos.

Pressman (2011) defende a ideia que a primeira questão a ser resolvida antes do início das atividades encontra-se no modo como os envolvidos da empresa desenvolvedora do sistema e os *stakeholders* se relacionam. O autor afirma que frequentemente cada pessoa define seu território e se comunica com uso de ferramentas pouco produtivas, como notas formais e memorandos, levando a mal-entendidos e perda de informações. O que se busca é que os envolvidos se relacionem como um time único e busquem sinceramente um resultado final de sucesso, deixando de lado a mentalidade “nós e eles”.

Kanjirath Nandita (2011) afirma que não são raras as vezes que partes importantes interessadas do projeto são deixadas de lado no processo de elicitação de requisitos, fato que pode ser desastroso. Para evitar este problema a autora enumera as seguintes providencias:

- I. Identificar e listar as partes interessadas. Trata-se dos indivíduos e grupos que tem participação, seja ela positiva ou negativa, na implementação do sistema;
- II. Classificar os participantes listados por ordem de influência no projeto. Pode ser de 1 a 5, por exemplo;
- III. Mapear o relacionamento entre os envolvidos relacionando-os visando encontrar divergências entre as informações e visões levantadas;
- IV. Registrar o nível de participação esperado de cada indivíduo identificado.

Todo processo de levantamento de informações e tomada de decisões deve estar em conformidade com as normas e regulamentações regionais. O contrário

pode ser altamente custoso no que se refere a uma adequação tardia (KANJIRATH NANDITA, 2011).

Algumas das técnicas mais comuns de levantamento de requisitos com mais detalhes são:

### 3.4.1 Entrevistas

Existe um grande leque de técnicas de levantamento de requisitos. No entanto as entrevistas são quase que as únicas utilizadas na captura de informações, apesar de que nem sempre seja o meio mais adequado para entender as necessidades dos usuários (CARRIZO, 2008).

As entrevistas, de acordo com Sommerville (2011), podem ser de dois tipos. Fechadas, as quais seguem um conjunto de perguntas pré-definidas. Abertas, em que não se tem uma agenda pré-definida. Sommerville deixa claro que na prática, geralmente, as entrevistas são um mesclado dos dois tipos. Tendo uma linha de raciocínio bem definida, mas aberta a quaisquer novas questões que se apresentem de forma a ajudar na compreensão e eliciação dos requisitos. Vale reforçar: entrevistas estruturadas tendem a ter resultados mais produtivos em relação às não estruturadas (DIESTE, 2008).

Apesar de ser largamente utilizada e ajudarem no entendimento global da aplicação, as entrevistas apresentam alguns problemas em relação à compreensão dos requisitos do domínio da aplicação. De acordo com Sommerville (2011, p.72), “pode ser difícil elicitar conhecimento do domínio por meio de entrevistas por duas razões”, são elas:

Especialistas em aplicações fazem uso de terminologias e jargões específicos para um domínio. Para eles, é impossível falar dos requisitos de domínio de outra maneira. Eles normalmente se utilizam da terminologia de forma precisa e sutil, o que dificulta a compreensão dos engenheiros de requisitos.

O conhecimento do processo negocial é tão familiar aos *stakeholders* que estes têm dificuldade em explicá-lo, ou pensam que algumas informações são tão básicas que não vale a pena mencioná-las. “Por exemplo, para um bibliotecário, não é necessário dizer que todas as aquisições são catalogadas antes de serem

adicionadas à biblioteca”. Todavia, isso pode não ser evidente ao entrevistador, e acaba não sendo levado em conta nos requisitos.

Entrevistadores de sucesso evitam ideias preconcebidas sobre os requisitos e estão dispostos a ouvir os *stakeholders*, mesmo que estes apresentem requisitos-surpresa, eles estão abertos a novas perspectivas sobre o sistema. E mais, bons engenheiros de requisitos procuram estimular os entrevistados a seguir uma linha de raciocínio visual da futura aplicação, apresentando até protótipos, de forma a estimular o futuro usuário no garimpo de informações importantes ainda não levantadas. (SOMMERVILLE, 2011).

### 3.4.2 Cenários

A construção de cenários busca aproximar do mundo real o funcionamento do sistema a ser desenvolvido. Uma descrição dos fluxos de informações expondo de forma simplista o que os usuários esperam que aconteça em diversas situações é elaborada visando um melhor e total entendimento das necessidades e casos específicos que talvez não tenham sido colocados em outras fases do processo de elicitação de requisitos do sistema. De acordo com Sommerville, 2011, pessoas que não estão habituadas à área de tecnologia da informação, geralmente enfrentam dificuldades em assimilar documentos técnicos e se dão melhor com exemplos da vida real.

Normalmente, no início é elaborado um esboço da interação entre os atores dos sistemas. No decorrer da elicitação são adicionados mais detalhes destas (SOMMERVILLE, 2011).

Sommerville, 2011 exemplifica a constituição de um cenário:

- Uma descrição do que o sistema e os usuários esperam quando o cenário se iniciar;
- Uma descrição do fluxo normal de eventos no cenário;
- Uma descrição do que pode dar errado e como isso é tratado;
- Informações sobre outras atividades que podem acontecer ao mesmo tempo;
- Uma descrição do estado do sistema quando o cenário acaba;



### 3.4.3 Casos de Uso

A técnica de elicitação de casos de uso tem se consolidado como fundamental em aplicações orientadas a objetos e, conseqüentemente, acompanhando o uso de UML, do inglês *Unified Modeling Language*, (Linguagem de Modelagem Unificada) na elaboração de documentos de requisitos. (SOMMERVILLE, 2011).

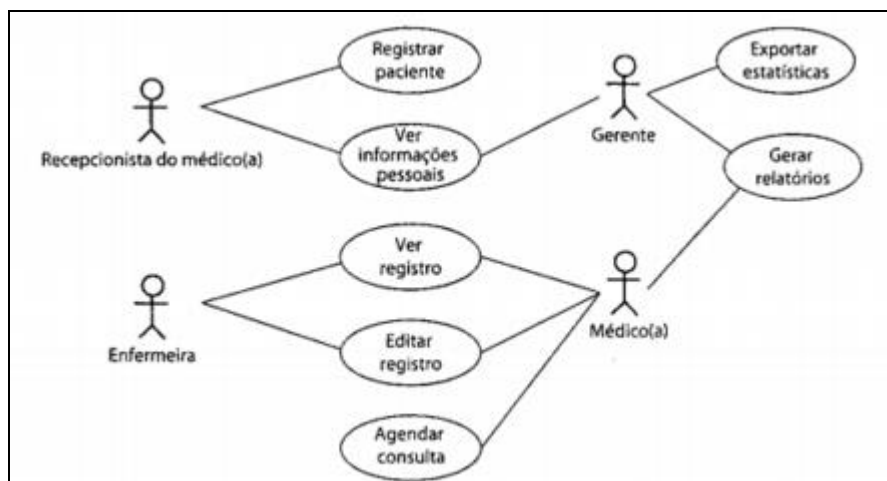
Mesmo que não oficializado, o que dá sentido aos casos de uso é a descrição textual em forma de cenários das iterações dos chamados “atores” (futuros usuários ou outros sistemas) com o software a ser construído.

Fowler (2005) explica a ideia de atores afirmando que estes são usuários, o que pode causar certo desconforto de forma que usuário aqui pode se referir a pessoas, objetos ou outros sistemas. A partir dessa premissa pode-se concluir que o termo ator não pode ser aplicado com a mesma eficácia que outros existentes na linguagem UML.

Sommerville (2011), resume casos de uso da seguinte forma:

O conjunto de casos de uso representa todas as possíveis interações que serão descritas nos requisitos de sistema. Atores, que podem ser pessoas ou outros sistemas, são representados como figuras 'palito'. Cada classe de interação é representada por uma elipse. Linhas fazem a ligação entre os atores e a interação. Opcionalmente, pontas de flechas podem ser adicionadas às linhas para mostrar como a interação se inicia (SOMMERVILLE, 2011, p.74).

A Figura 7 é proposta por Sommerville (2011) para demonstrar de forma gráfica o exposto acima.



**Figura 7 - Exemplo Gráfico - Caso de Uso**  
**Fonte: SOMMERVILLE (2011 p.75)**

A elaboração de casos de uso é uma boa prática em se tratando de levantar informações sobre as iterações entre os atores e o sistema, no entanto não é efetivo da mesma forma quando se busca “elicitare restrições ou requisitos de negócios e não funcionais em alto nível ou para descobrir requisitos de domínio”. (SOMMERVILLE, 2011 p.75).

Fowler (2005) explica que é possível inserir algumas informações importantes nos casos de uso, como as que se seguem:

- Pré-condição - o sistema deve garantir como verdadeiro antes de dar início ao caso de uso. Auxilia no desenvolvimento do sistema indicando aos analistas programadores regras que não precisam validar a nível de código;
- Garantia - algo que o sistema deve entregar ao final do caso de uso. As garantias de sucesso são aquelas que se espera após o término do caso de uso com sucesso. As garantias mínimas são aquelas esperadas após qualquer cenário;
- Gatilho - evento que dá início ao caso de uso;

A manutenção do caso de uso deve acontecer durante toda fase do projeto. Caso ocorram mudanças e/ou correções, estas devem ser atualizadas e marcadas como uma nova revisão, visando manter o caso de uso em acordo com o projeto em desenvolvimento. Vale ressaltar que é possível que o caso de uso faça referência a outros documentos, o que enriquece a elicitação como mais informações. (SOMMERVILLE, 2011).

### 3.4.3.1 Propriedades do Caso do Uso

Larman (2010) traz outras propriedades importantes na elaboração do caso de uso.

- Ator principal: usuário responsável por chamar os serviços disponibilizados pela aplicação visando um objetivo;
- Lista de interessados/interesses: lista composta dos requisitos que originaram o caso de uso;
- Pré-condições: composta dos fatores que devem ser atendidos antes de se dar início ao caso de uso;
- Pós-condições: composta dos fatores que devem ser atendidos após a execução do caso de uso;
- Requisitos especiais: existem caso se busque relacionar algum requisito não funcional ao caso de uso;

### 3.4.3.2 Fluxo de eventos

De Sousa (2008), explica que a seção de fluxo de eventos é de extrema importância no caso de uso pois traz os passos entre o ator e o sistema. Cada cenário dentro do contexto da aplicação é representado por um fluxo de evento. Este por sua vez contém os passos da iteração “ator x sistema”. Cada caso de uso possui pelo menos um evento chamado (principal). Caso outros eventos existam, são chamados de fluxos alternativos (Figura 8).



**Figura 8 - Fluxos de Eventos de um caso de uso**  
 Fonte: adaptado de (DE SOUSA, 2008)

De Sousa (2008) descreve algumas regras da aplicação de fluxos de eventos em casos de uso:

- O fluxo alternativo pode ter origem do fluxo principal ou de outro fluxo alternativo. Ele se inicia quando seu gatilho é acionado a partir do momento que uma condição de negócio pré-determinada é tida como verdadeira;
- Todo fluxo alternativo, sem exceção, possui uma condição que será testada para que esse seja executado;
- Um fluxo alternativo proveniente de outro fluxo alternativo é chamado de sub-fluxo.

A seguir tem-se dois exemplos de fluxos de eventos propostos por De Sousa (2008):

Exemplo 1:

- Fluxo de Evento Básico: Abertura de Conta para um novo cliente;
- Fluxo de Evento Alternativo: Abertura de Conta para cliente já existente;

Exemplo 2:

- Fluxo de Evento Básico: Abertura de conta para um novo cliente.
- Fluxo de Evento Alternativo: Não informado Tipo da Conta

### 3.4.3.3 Fluxos de exceção

A seção de fluxos de exceção contém os caminhos que o sistema deve tomar caso regras não sejam tidas como verdadeiras. Geralmente mensagens são mostradas ao usuário quando isso ocorre, mas é possível também que outros casos de uso sejam disparados (DE SOUSA, 2008).

De Sousa traz o Quadro 3 que exemplifica de forma detalhada fluxos de exceção de um sistema bancário fictício.

**Quadro 3 - Fluxo de Exceções do caso de uso Abrir Conta**

<p><b>Regras</b></p> <p>H1 - CPF é obrigatório.</p> <p>R2 - Tipo da conta é obrigatório.</p> <p>R3 - Senha é obrigatória.</p> <p>R4 - Caso o cliente já possua uma conta do mesmo tipo na agência, não poderá abri-la.</p> <p><b>Fluxo de Exceções</b></p> <p>E1 - CPF não foi informado.</p> <p>Regra R1</p> <p>Ação: Sistema emite mensagem M1.</p> <p>E2 - Tipo da conta não foi informado.</p> <p>Regra R2</p> <p>Ação: Sistema emite mensagem M2.</p> <p>E3 - Senha não foi informada.</p> <p>Regra: R3</p> <p>Ação: Sistema emite mensagem M3.</p> <p>E4 - Cliente solicita abertura de conta de mesmo tipo que ele já possui na agência.</p> <p>Regra: R4</p> <p>Ação: Sistema emite mensagem M4</p>
--

Fonte: De Sousa (2008)

### 3.4.4 Etnografia

A técnica de se observar o ambiente de trabalho no qual o sistema irá ser implantado é chamada de etnografia. Com seu uso, o analista de sistemas busca,

por meio de imersão, entender a cultura de trabalho bem como a política organizacional da empresa. A ideia baseia-se no fato que requisitos não formais possam ter sido ignorados e através da observação dos futuros usuários do sistema seja possível ao analista de requisitos descobri-los.

Sommerville (2011), explica que a etnografia permite de descoberta principalmente de dois tipos de requisitos:

- Requisitos frutos do modo que os futuros atores realmente trabalham, não como é especificado nas normas e regras do processo que dizem como deviam trabalhar;
- Requisitos provenientes da cooperação de outras pessoas envolvidas no processo.

Como a etnografia tem foco no usuário final mostra-se como uma ótima técnica para completar com detalhes definições já levantadas por outras fases do processo de elicitação de requisitos. Nesse sentido tem-se essa técnica como complemento e não se recomenda o uso para definição de regras negociais da aplicação a ser desenvolvida.

Após o levantamento dos dados pelo analista, o próximo passo é atualizar as informações já levantadas em outras fases da elicitação. Uma desvantagem da etnografia é o fato de consumir um tempo considerável e de que o analista pode ser induzido ao erro em suas anotações.

#### 3.4.5 Validação

A validação das especificações aparece como fase essencial visto que estudos mostram que corrigir um defeito na fase de construção do sistema proveniente de um requisito mal definido pode custar até duzentas vezes mais caro caso encontrado na fase de validação. (LEFFINGWELL, D).

Sommerville (2011) cita verificações que podem ser aplicadas após o processo de levantamento de requisitos:

- Verificações de validade: visam analisar e verificar se as funções apontadas pelos usuários como necessária estão bem definidas. Após um estudo aprofundado e cruzamento de informações com *stakeholders* é possível que se identifique funções diferentes ou adicionais;

- Verificações de consistência: objetiva encontrar requisitos conflitantes ou repetidos;
- Verificações de completude: valida se o escopo pretendido foi alcançado no documento de requisitos;
- Verificações de realismo: visa responder, por meio de análise e conhecimento das tecnologias a serem utilizadas na construção do sistema, se é possível o desenvolvimento do pretendido levando em consideração as limitações tecnológicas e o cronograma visado.

#### 3.4.6 Questionários

De acordo com Bastos Junior (2005) questionário é, basicamente, um documento usado para direcionar uma ou mais pessoas a responder uma ou mais perguntas.

A técnica de aplicação de questionários que possibilita ao analista levantar informações de inúmeras pessoas que irão interagir com o sistema referentes aos comportamentos, crenças e funcionalidades que almejam no software a ser desenvolvido (KENDALL; KENDALL, 2010).

Camacho (2005) coloca como principais vantagens do uso de questionário: a padronização das perguntas e a possibilidade de tratamento estatístico das respostas. Já como desvantagens o mesmo autor aponta a restrição de conteúdo das respostas e a pouca interação, de forma essa técnica atua sem qualquer pessoalidade.

##### 3.4.6.1 Elaboração do Questionário

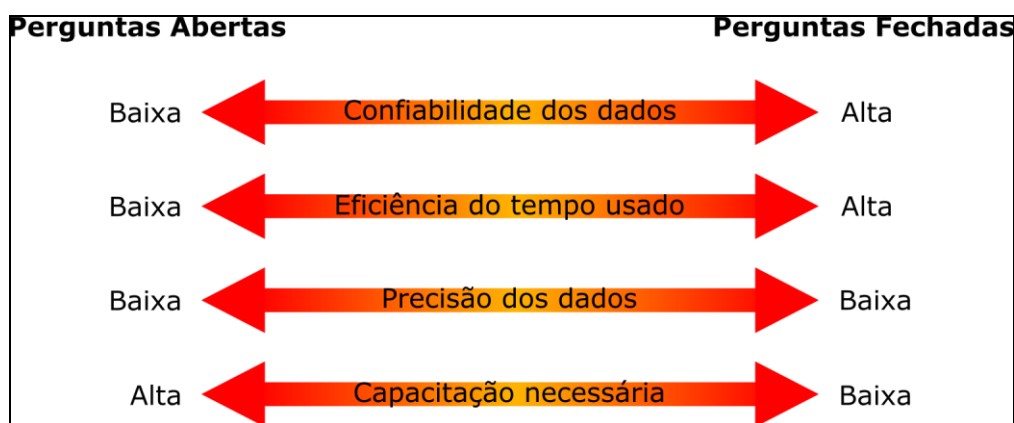
De acordo com Kendall e Kendall (2010), existe uma ótima oportunidade de produtividade no processo de levantamento de requisitos se entrevistas e questionários forem usados de forma inteligente:

- A criação de questionários mais ricos e completos com baseados em dados angariados após a aplicação de entrevistas;
- O analista pode usar questionários para quantificar informações emergentes de entrevistas;

- Refinar as respostas dadas nos questionários através da aplicação de entrevistas;

A elaboração das perguntas que vão compor o questionário é um processo engenhoso e complexo. Um questionário mal formulado pode, facilmente, levar a conclusões equivocadas que acabam sendo prejudiciais para o projeto em questão (BASTOS JUNIOR, 2005).

Existem dois principais tipos de pesquisa: pesquisas abertas e pesquisas fechadas. As abertas são as que possibilitam o respondente colocar de forma textual sua resposta, exigindo mais tempo a análise das respostas. Em contrapartida fornece dados mais ricos e possibilita o levantamento de informações não previstas. As perguntas fechadas são aquelas que o analista disponibiliza diversas alternativas entre as quais o respondente pode escolher uma ou mais, dependendo do enunciado. Esse método possibilita uma captação de dados mais rápida e em maior número, no entanto restringe a captação de informações não imaginadas antes da pesquisa. São largamente usadas através de perguntas de múltipla escolha (BASTOS JUNIOR 2005). A Figura 9 traz algumas propriedades desses grupos facilitando a comparação entre eles.



**Figura 9 - Atributos de perguntas fechadas e abertas**  
 Fonte: adaptado de (KENDALL; KENDALL, 2010)

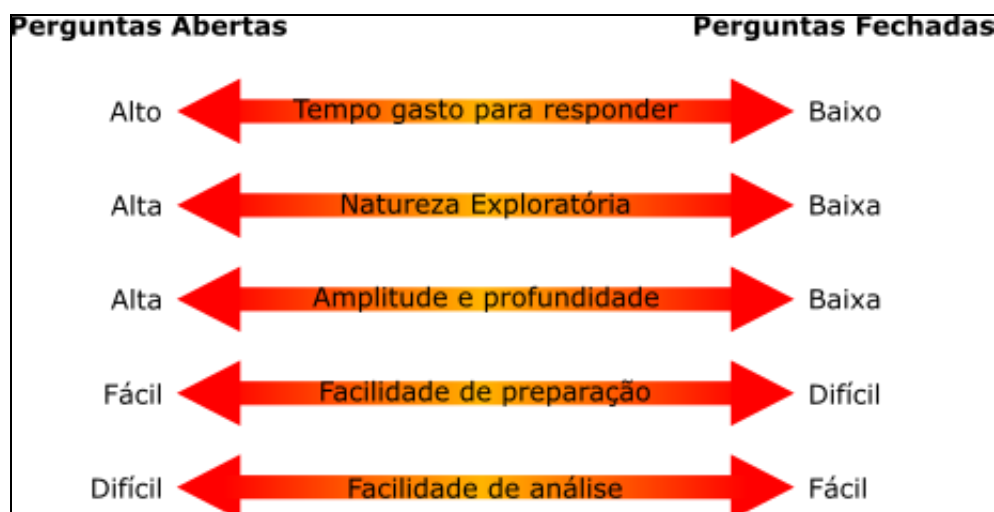
Segundo Bastos Junior (2005), a decisão de qual tipo de pergunta usar no questionário deve ter base no modo como se planeja usar a informação obtida, já que a forma que as questões são estruturadas delimita a unidade de medida pelas quais as respostas serão classificadas.

Kendall e Kendall (2010) afirma que as perguntas abertas, também conhecidas como subjetivas, são muito usadas quando se busca saber as opiniões



dos usuários de um determinado sistema, já que seria quase impossível listar em alternativas todas opiniões possíveis. Em se tratando das perguntas fechadas, estas são a melhor solução quando o objeto de estudo envolve um grande número de pessoas localizadas em diferentes filiais, por exemplo.

A Figura 10 compara perguntas fechadas e aberta em diversos aspectos na aplicação de questionários:



**Figura 10 - Uso de questões subjetivas e objetivas em questionários**  
 Fonte: adaptado de (KENDALL & KENDALL, 2010)

Kendall e Kendall (2010) trazem, também, alguns pontos de atenção e dicas em se tratando de elaboração de perguntas:

- Use a linguagem dos respondentes sempre que possível. Seja simples;
- Trabalhe buscando ser específico, de forma a evitar qualquer mal-entendido causados por uma pergunta vaga e/ou ambígua. Ao mesmo tempo não seja específico em demasia. O que se busca é um equilíbrio.
- Faça perguntas curtas;
- Envie as perguntas corretas de acordo com o respondente. Ele deve fazer parte do público-alvo para o qual a pergunta foi elaborada. Caso contrário ele não estará capacitado para responder corretamente.

#### 3.4.6.2 Aplicação do Questionário

A aplicação de questionários pode ser feita de diversas formas. Para Parasuraman (1991) as principais são:

- Auto resposta via e-mail:
  - Acesso facilitado a um grande número de pessoas;
  - Comodidade para o questionado enviar as respostas quando lhe for conveniente;
  - Baixa de retorno;
  - Respostas curtas, inviabilizando perguntas que exijam respostas detalhadas;
- Auto resposta em grupo:
  - Perguntas são aplicadas simultaneamente a um grupo de participantes selecionados de acordo com critérios pré-definidos de conveniência;
  - Cada participante responde individualmente;
  - Dúvidas dos participantes podem e devem ser discutidas e sanadas durante a aplicação;
- Auto resposta porta a porta:
  - O investigador vai ao encontro do respondente e deixa o questionário a ser respondido com o participante, o qual deverá, por sua vez, responder e enviar via correio. Esse método tem caído em desuso.

#### 3.4.7 Workshops

Workshop trata-se de uma técnica de levantamento de requisitos colaborativa, isto é, envolve a participação de grupos de interessados que colaboram entre si para coleta de dados e solução tanto de problemas quanto definição de estratégias (AURUM; WOHLIN, 2005). As três abordagens principais a serem tratadas a seguir aplicam de alguma forma a diretrizes básicas propostas por Pressman (2006) abaixo:

- I. As discussões terão como participantes representantes de diferentes grupos de interessados, previamente informados de regras de preparação e participação;
- II. Um facilitador que vai dar o tom da reunião e guiar os assuntos a serem tratados.
- III. Um meio de anotar tudo que está sendo tratado de forma que todos envolvidos tenham ciência do conteúdo. Uma boa ideia é uso de um

computador compartilhado no qual o facilitador anota as decisões tomadas de forma que todos acompanhem e tenham acesso ao mesmo documento após a reunião;

- IV. A meta pode ser identificar ou resolver uma questão pendente, tratar de decisões negociais, especificar requisitos, entre outras questões válidas do projeto.

Basicamente, workshops de requisitos são reuniões com diversos *stakeholders* de diversas áreas, cujo objetivo é a descoberta e desenvolvimento de requisitos, podem ser combinados com diferentes técnicas, dependendo do seu objetivo (AURUM; WOHLIN, 2005).

#### 3.4.7.1 JAD – *Joint Application Development* – (Desenvolvimento conjunto de aplicação)

Aurum e Wohlin (2005) caracterizam o JAD como um processo de eliciação de requisitos no qual um grupo de *stakeholders* se reúne com o objetivo de investigar problemas e suas possíveis soluções. A vantagem aqui é que com a presença de diversos envolvidos, seja possível a tomada de decisões importantes rapidamente. Entretanto, vale ressaltar que o foco de uma reunião JAD é concentrado nos problemas negociais das partes envolvidas, deixando de lado questões técnicas.

De Pádua (2000) enumera as vantagens do uso do JAD:

- I. Comprometer com os requisitos os usuários com poder de decisão sobre estes;
- II. Reduzir o prazo de levantamento da especificação dos requisitos;
- III. Eliminar requisitos de valor questionável;
- IV. Reduzir diferenças de interpretação dos requisitos entre usuários e desenvolvedores;
- V. Produzir um primeiro esboço da interface de usuário;
- VI. Trazer à vista, o mais cedo possível, problemas políticos que possam interferir no projeto.

O processo de aplicação do JAD consiste em três principais fases (DE PÁDUA, 2000):

- I. Personalização: adaptação do método à tarefa específica, geralmente feita pelo líder do JAD;
- II. Sessões: parte principal do método, na qual participam todos os interessados na tarefa;
- III. Fechamento: produção dos documentos resultantes.

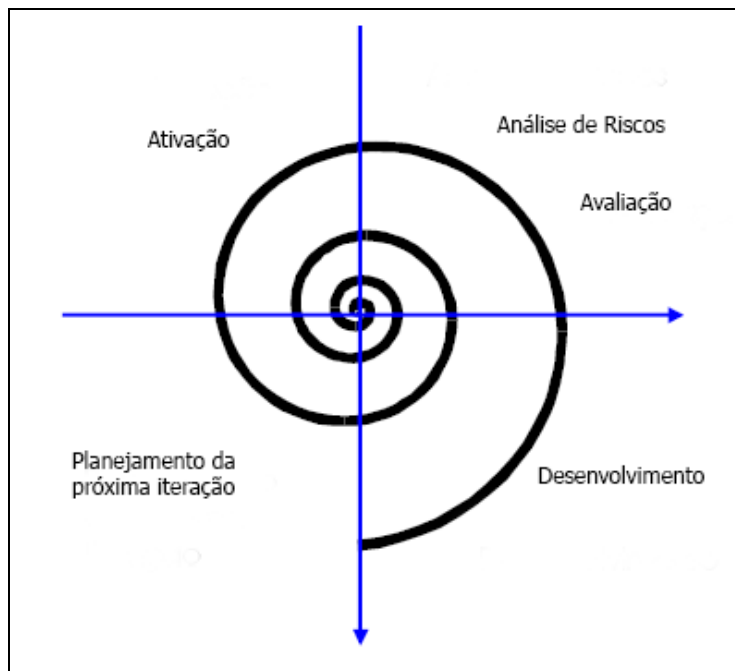
#### 3.4.7.2 Brainstorming

O Brainstorm (tempestades de ideias) é uma expressão largamente utilizada em diferentes áreas para busca de ideias criativas. Aurum e Wohlin (2005) definem Brainstorm como um processo no qual diferentes grupos de interessados e envolvidos no projeto participam de uma discussão informal visando gerar o máximo de ideias possíveis em pouco tempo. Não focar em críticas ou correções durante a discussão é crucial para manter o grupo relaxado e descontraído. Também vale ressaltar que não é o foco desse processo a tomada de decisões.

#### 3.4.8 Prototipagem

Segundo Sommerville (2010), “protótipo é uma versão inicial de um sistema usado para demonstrar conceitos, experimentar opções de projeto e descobrir mais sobre o problema e suas possíveis soluções“. O autor continua afirmando que o desenvolvimento do protótipo deve ter baixo custo e deve acontecer em pouco tempo, de forma a ser apresentado e discutido ainda no início do projeto.

Desenvolvido para ser uma representação dos requisitos levantados das fases iniciais, o protótipo assim que pronto é apresentado ao cliente que por sua vez avalia, corrige requisitos equivocados e sugere modificações que melhor atendem a suas necessidades. Os analistas recebem o feedback do cliente e, em seguida, corrigem o protótipo, atualizam a documentação de requisitos e os apresentam novamente ao cliente. Esse processo se repete até que estejam todos de acordo com a versão final do protótipo, que pode ser usada como documento de avaliação ao final do projeto (PRESMAN).



**Figura 11 - O modelo de ciclo de vida em Espiral**  
**Fonte: DE PADUA (2000 p. 25)**

Um processo semelhante chamado prototipagem evolutiva é proposto por De Pádua (2000). O autor descreve esse processo como sendo um desenvolvimento de uma série de versões provisórias seguindo o modelo de espiral. Os protótipos são avaliados pelo cliente e voltam para fase de desenvolvimento, cobrindo cada vez mais requisitos, até que se atinja o produto desejado. Esse modelo permite que os requisitos sejam definidos progressivamente de forma flexível e com uma produtiva iteração com o cliente como mostra a Figura 11.

## 4 PROCEDIMENTOS METODOLÓGICOS DA PESQUISA

Esta é uma pesquisa descritiva com delineamento de levantamento de dados, uma vez que a coleta de dados foi realizada por meio de questionário com aplicação online a profissionais de tecnologia da informação que tenham contato com os fundamentos da engenharia de requisitos no seu dia a dia. Em se tratando do método científicos, o escolhido foi o dedutivo, de forma que se objetivou descobrir se as seguintes premissas são verdadeiras:

- O perfil da amostra (formação acadêmica e cargo que ocupam) tem relação com a forma que os profissionais utilizam os fundamentos da engenharia de requisitos;
- O porte das empresas nas quais os profissionais da amostra trabalham tem influência nas técnicas de levantamento de requisitos utilizadas nos projetos de software;
- O porte dos clientes aos quais os profissionais da amostra prestam serviços tem influência na fase de levantamento de requisitos dos projetos de desenvolvimento de software.

### 4.1 AMOSTRA

A amostra consiste em cinquenta profissionais que atuam de alguma forma em desenvolvimento de projetos de software.

### 4.2 COLETA DE DADOS

O processo de coleta de dados foi realizado por meio de aplicação de questionário on-line (APÊNDICE A - por um período de um mês. A ferramenta escolhida para a aplicação foi a Survey Monkey. Trata-se de um site ([surveymonkey.com](http://surveymonkey.com)) que oferece uma plataforma para criação de questionários online e disponibiliza pacotes de serviços tanto gratuitos quanto pagos. Um diferencial do Survey Monkey é o fato de permitir apenas uma resposta por IP (Internet Protocol, endereço único de computador), o que evita que uma pessoa responda mais de uma vez.

O questionário elaborado foi postado em grupos de tecnologia da informação nas redes sociais Facebook e LinkedIn e solicitado a colaboração dos membros.

#### 4.2.1 Tratamento dos dados

Os dados coletados foram tabulados em uma planilha Excel agrupados da seguinte forma:

1. Três técnicas de levantamento de requisitos mais utilizadas;
2. Três técnicas de levantamento de requisitos menos utilizadas;
3. Curso de formação dos profissionais que utilizam mais técnicas de levantamento de requisito;
4. Curso de formação dos profissionais que utilizam menos técnicas de levantamento de requisitos;
5. Porte da empresa dos profissionais que utilizam mais técnicas de levantamento de requisitos;
6. Porte da empresa dos profissionais que utilizam menos técnicas de levantamento de requisitos;
7. Porte do cliente aos quais os profissionais que utilizam mais técnicas de levantamento de requisito prestam serviços;
8. Porte do cliente aos quais os profissionais que utilizam menos técnicas de levantamento de requisito prestam serviços.

Cada entrevistado recebeu uma pontuação de acordo com a frequência que faz uso das técnicas de levantamento de requisitos expostas no presente trabalho. Em seguida esta pontuação foi somada ao grupo do seu curso, cargo, porte da empresa na qual trabalha e porte médio dos clientes aos quais presta serviços. Assim obteve-se uma pontuação para cada um dos critérios.

A pontuação de frequência será aplicada de acordo com o Quadro 4 - Pontuação de Frequência.

**Quadro 4 - Pontuação de Frequência**  
Fonte: Autoria própria

Nunca	0 pontos
-------	----------

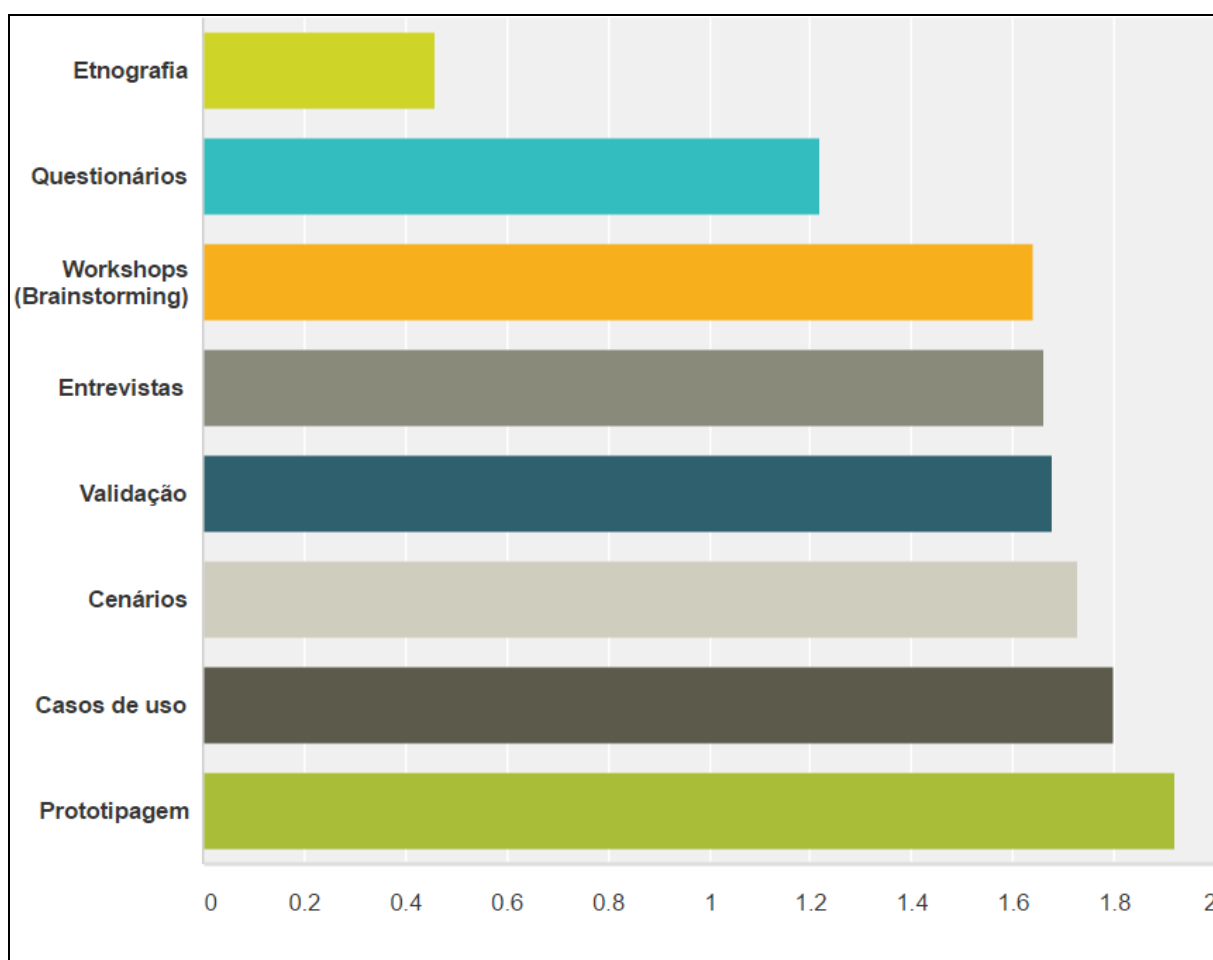
Raramente	1 ponto
Quase Sempre	2 pontos
Sempre	3 pontos



## 5 ANÁLISE E DISCUSSÃO DOS RESULTADOS

### 5.1 TÉCNICAS DE LEVANTAMENTO DE REQUISITOS

Em se tratando das técnicas de levantamento de requisitos mais utilizadas entre os entrevistados temos a Figura 12 que representa graficamente as médias ponderadas.



**Figura 12 - Técnicas de Levantamento de Requisitos mais utilizadas**  
Fonte: Pesquisa própria

A técnica prototipagem aparece como a mais utilizada, seguida de casos de uso e cenários. Dentre as menos utilizadas a etnografia se posiciona em primeiro, seguida de validação.

**Quadro 5 - Média ponderada: Uso das Técnicas de Levantamento de Requisitos**  
**Fonte: Pesquisa própria**

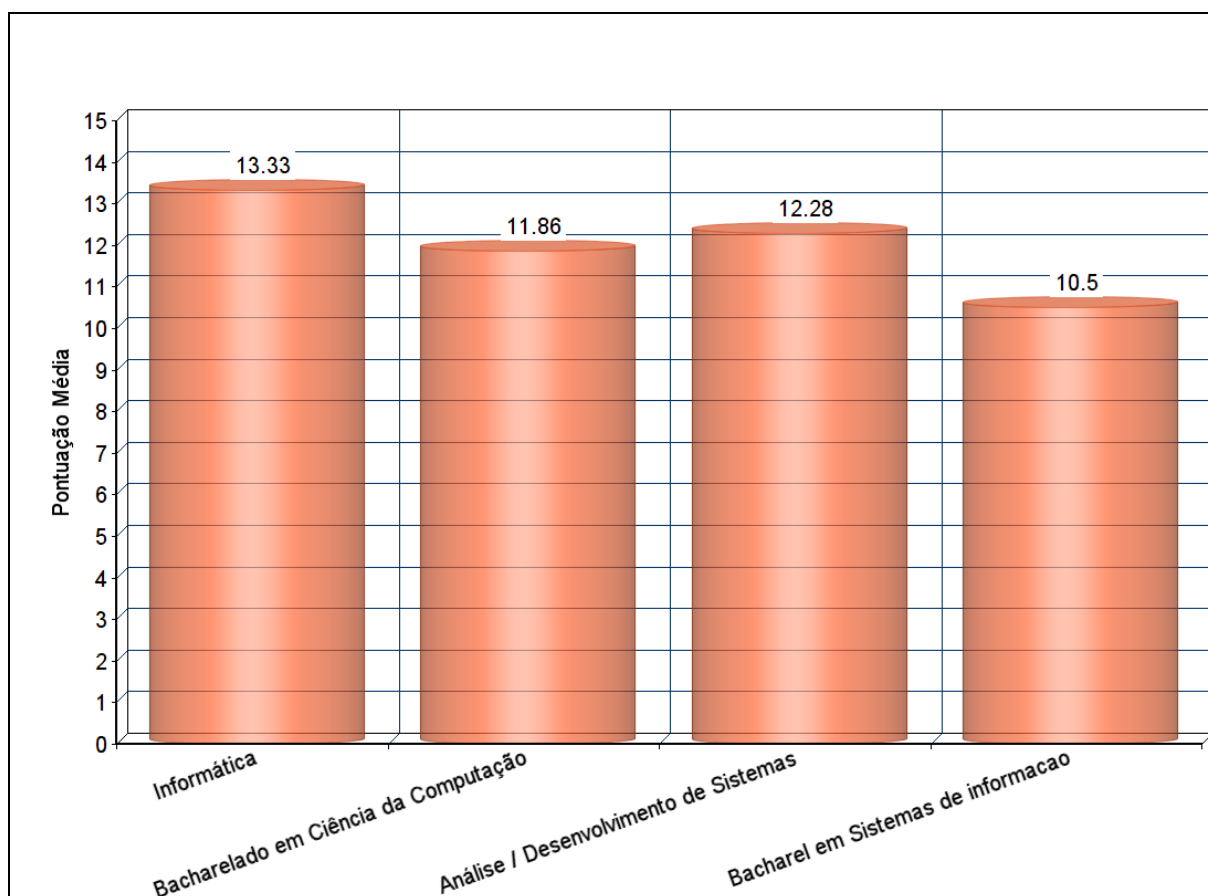
	Nunca	Raramente	Quase Sempre	Sempre	Total	Média ponderada
Etnografia <i>Total Respostas</i>	62,00% 31	30,00% 15	8,00% 4	0,00% 0	50	0,46
Questionários <i>Total Respostas</i>	28,00% 14	36,00% 18	22,00% 11	14,00% 7	50	1,22
Workshops (Brainstorming) <i>Total Respostas</i>	8,00% 4	40,00% 20	32,00% 16	20,00% 10	50	1,64
Entrevistas <i>Total Respostas</i>	8,00% 4	42,00% 21	26,00% 13	24,00% 12	50	1,66
Validação <i>Total Respostas</i>	8,00% 4	30,00% 15	48,00% 24	14,00% 7	50	1,68
Cenários <i>Total Respostas</i>	12,24% 7	18,37% 9	53,06% 26	16,33% 8	50	1,73
Casos de uso <i>Total Respostas</i>	8,00% 4	26,00% 13	44,00% 22	22,00% 11	50	1,80
Prototipagem <i>Total Respostas</i>	6,00% 3	20,00% 10	50,00% 25	24,00% 12	50	1,92

Fonte: Dados da pesquisa

O Quadro 5 mostra os dados coletados de forma mais detalhada e permite perceber que a prototipagem aparece fortemente nas opções de uso “Quase Sempre” e “Sempre”, fato que determinou sua primeira colocação dentre as técnicas.

## 5.2 FORMAÇÃO ACADÊMICA

Em relação a formação acadêmica, o curso de Informática aparece em primeiro lugar, com 13.33 pontos, seguido de Análise / Desenvolvimento de Sistemas (12.28) e Bacharelado em Ciência da Computação (11.86). Mais dados desse critério podem ser observados na Figura 13. Lembrando que essa pontuação teve origem no total de pontos dividido pelo número de entrevistados, por exemplo: um total de 28 entrevistados são formados em Análise / Desenvolvimento de Sistemas. A somatória desses foi 334. Para chegamos a uma pontuação média temos 334 dividido por 28, resultando em 12.28, presente na Figura 13.



**Figura 13 - Pontuação Média por Formação Acadêmica**  
Fonte: Pesquisa própria

Ao cruzarmos os dados obtidos é possível estabelecer a formação dos entrevistados que fazem uso de protótipos como ferramenta de levantamento de requisitos com mais frequência:

- Análise / Desenvolvimento de Sistemas, com 58 pontos (4 raramente + 14 quase sempre + 9 sempre);
  - Bacharelado em Ciência da Computação, com 24 pontos (6 raramente + 6 quase sempre + 2 sempre);
- Com menos pontos tem-se:
- Bacharel em Sistemas de Informação, com 8 pontos (4 quase sempre);
  - Informática com 5 pontos (1 quase sempre + 1 sempre)

**Tabela 1 - Frequência de Uso de acordo com a Formação Acadêmica**  
**Fonte: Pesquisa própria**

	FORMAÇÃO ACADEMICA															
	Análise / Desenvolvimento de Sistemas				Bacharelado em Ciência da Computação				Informática				Bacharel em Sistemas de Informação			
	N	R	QS	S	N	R	QS	S	N	R	QS	S	N	R	QS	S
<b>Técnicas</b>																
Entrevistas	2	14	7	5	3	4	4	4	0	1	0	2	0	1	2	1
Etnografia	17	9	2	0	8	6	1	0	2	0	1	0	4	0	0	0
Validação	3	7	14	4	1	5	6	3	0	1	2	0	0	2	2	0
Casos de uso	1	6	12	9	3	6	5	1	0	0	2	1	0	1	3	0
Cenários	3	5	16	4	1	4	8	2	1	0	2	0	1	0	2	1
Workshops (Brainstorming)	4	8	9	7	1	6	6	2	0	1	1	1	0	3	1	0
Prototipagem	<b>2</b>	<b>3</b>	<b>14</b>	<b>9</b>	<b>1</b>	<b>6</b>	<b>6</b>	<b>2</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>4</b>	<b>0</b>
Questionários	5	13	6	4	3	4	5	3	2	0	0	1	3	0	1	0

Legenda: N – Nunca, R – Raramente, QS – Quase Sempre, S – Sempre

Mais dados relativos às pontuações de cada formação acadêmica podem ser observados na Tabela 1. Pode-se definir a pontuação final de cada técnica por formação acadêmica, dando origem a Tabela 2:

**Tabela 2 - Pontuação Final por Formação Acadêmica**  
**Fonte: Pesquisa própria**

PONTUAÇÃO POR FORMAÇÃO ACADEMICA															
----------------------------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

	Análise / Desenvolvimento de Sistemas	Bacharelado em Ciência da Computação	Informática	Bacharel em Sistemas de Informação
<b>Técnicas</b>				
Entrevistas	43	24	7	8
Etnografia	13	8	2	0
Validação	47	26	5	6
Casos de uso	57	19	7	7
Cenários	49	26	4	7
Workshops (Brainstorming)	47	24	6	5
Prototipagem	58	24	6	8
Questionários	37	23	3	2

Um fato interessante a ser considerado é que as duas técnicas mais utilizadas (Prototipação e Casos de Uso) contam com uma grande iteração entre analista e cliente no processo. Em contrapartida, a última colocada, etnografia, caracteriza-se por ser uma técnica na qual o analista observa o cliente em seu ambiente de trabalho.

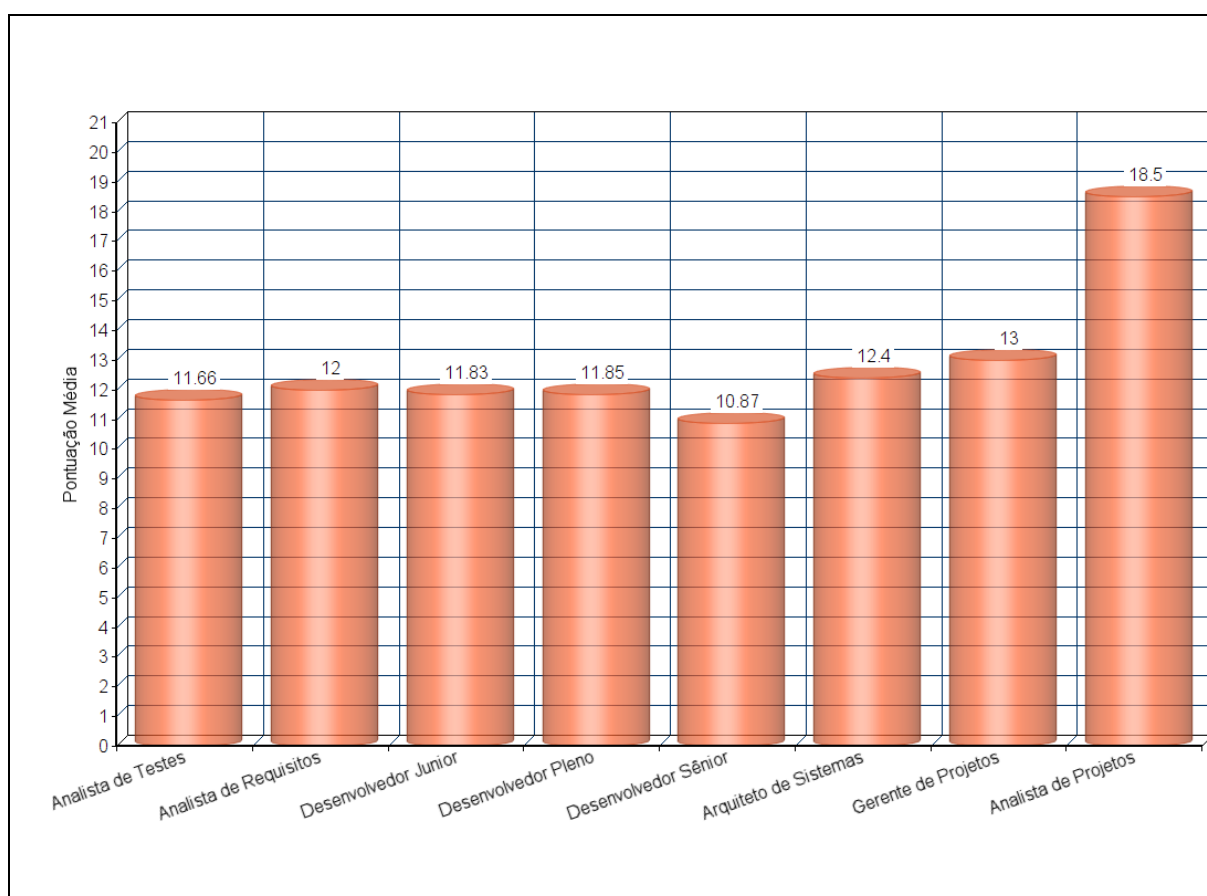
Segundo Sommerville (2009, p.75), “a etnografia é uma técnica de observação que pode ser usada para compreender os processos operacionais e ajudar a extrair os requisitos de apoio para esses processos. Um analista faz uma imersão no ambiente de trabalho em que o sistema será usado.” O autor completa ao afirmar que essa técnica tem como foco descobrir requisitos implícitos não antes revelados em fases anteriores do processo e descobrir também como as pessoas realmente trabalham, não como elas deveriam trabalhar.

Já a prototipação e casos de uso contam com uma participação direta e efetiva do cliente para que este valide e sugira correções, se necessárias, no modelo criado pelo analista. Isto corrobora com o pensamento de Sommerville (2009) que exalta o fato de que os protótipos possibilitam aos clientes e futuros usuários averiguarem o quanto o sistema a ser desenvolvido dará suporte ao seu trabalho. O resultado dessa análise é o surgimento de ideias e novos requisitos para o sistema. E mais, com os protótipos em mãos, os *stakeholders* podem apontar omissões e erros de requisitos.

Vale ressaltar que protótipos e casos de uso são usados como documentos de validação ao final do projeto, servindo de base de comparação do sistema entregue com o acordado na fase de levantamento.

### 5.3 CARGO

Entre os entrevistados, a ocupação que atingiu a maior pontuação média de uso das técnicas de levantamento de requisitos apontadas no presente estudo foi a de Analista de Projetos com 18.5 pontos, destacando-se significativamente quando comparada à segunda colocada, Gerente de Projetos, que atingiu apenas 13 pontos. Em último encontra-se o cargo de Desenvolvedor Sênior.



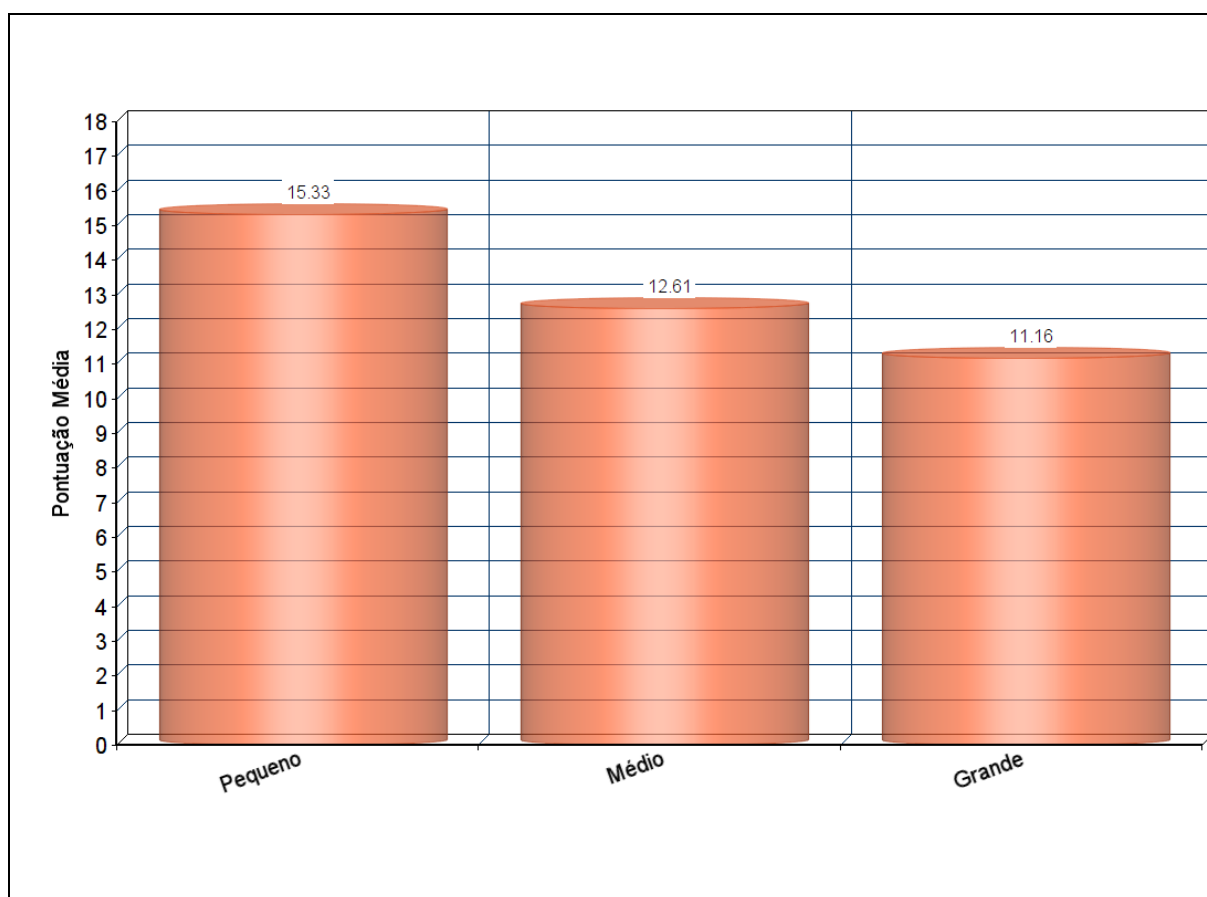
**Figura 14 - Pontuação Média por Cargo**  
Fonte: Pesquisa própria

A Figura 14 mostra todas pontuações de acordo com o cargo exercido. Segundo INFOJOBS, 2016 o Gerente Comercial é “o responsável por organizar os recursos humanos e materiais disponíveis, alocando-os de acordo às necessidades do cliente, além de elaborar um cronograma de ações e disponibilizá-lo para todos os envolvidos no serviço”. O site também afirma que esse profissional atua na área comercial com o cliente acarretando em uma grande responsabilidade, talvez por isso na presente pesquisa esse cargo se destacou exponencialmente.

Entre os outros cargos não se obteve grandes diferenças entre as pontuações, o que leva a conclusão que, com exceção da posição de Gerente Comercial, os cargos não definem de forma significativa a frequência com a qual os profissionais fazem uso das técnicas de elicitação de requisitos pontuadas no presente estudo.

#### 5.4 PORTE DA EMPRESA

Quanto ao porte da empresa, a pesquisa mostrou que os profissionais das de porte pequeno (15.33 pontos) usam mais técnicas de levantamento de requisitos do que as de médio (12.61 pontos) e grande porte (11.16 pontos). A Figura 15 traz essas informações.



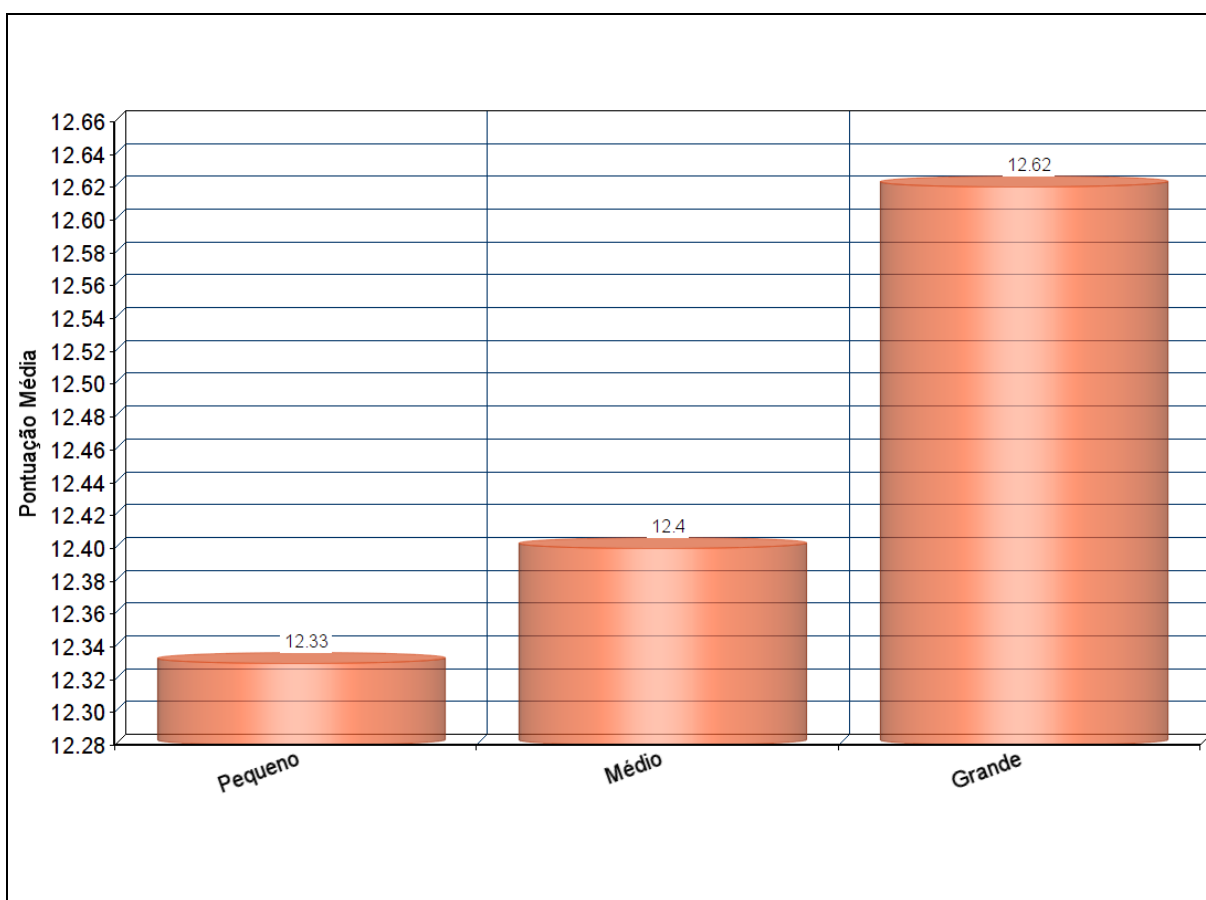
**Figura 15 - Pontuação Média pelo Porte da Empresa**  
Fonte: Pesquisa própria

Com base nos dados demonstrados na Figura 15 é possível afirmar que quanto menor a empresa, mais técnicas serão empregadas na fase de levantamento de requisitos.

## 5.5 PORTE DO CLIENTE

Em se tratando do porte dos clientes que os entrevistados costumam atender a pesquisa trouxe os seguintes dados:

- Pequeno porte, com 12.33 pontos;
- Médio porte, com 12.4 pontos;
- Grande porte, com 12.62.



**Figura 16 - Pontuação Média Pelo Porte do Cliente**  
Fonte: Pesquisa própria

Observa-se, então, na Figura 16 exatamente o contrário do constatado nos dados coletados referentes ao tamanho das empresas desenvolvedoras de software nas quais os entrevistados estão inseridos. Quanto maior o cliente atendido, mais técnicas de levantamento de requisitos são utilizadas.



## 6 CONCLUSÃO

Tendo em vista a literatura estudada e os dados coletados na pesquisa, chegou-se às seguintes considerações.

### 6.1 OBJETIVOS GERAL E ESPECÍFICOS

Respondendo ao Objetivo Geral desta pesquisa sobre identificar as principais técnicas de levantamento de requisitos utilizadas pelos analistas no desenvolvimento de um sistema, as técnicas que se destacaram foram a prototipação e casos de uso.

Fato interessante a ser observado é que ambas se caracterizam como técnicas que promovem uma grande iteração entre o cliente e a empresa desenvolvedora do software. E mais, ambas têm como resultado documentos ricos em detalhes os quais são frequentemente utilizados ao final do projeto como instrumentos de validação de sistema.

Quanto ao primeiro objetivo específico, o primeiro - levantamento bibliográfico sobre as técnicas de levantamento de requisitos - a pesquisa mostrou-se fundamental no que se diz respeito ao aprofundamento das características particulares de cada uma das técnicas de levantamento de requisitos. Percebeu-se também, que a internet está repleta de sites com informações sobre o assunto, no entanto autores clássicos como Sommerville e Mendes Silva Filho, entre outros, são as principais fontes das páginas da rede.

O segundo objetivo (elaboração do questionário) foi concluído com amparo no levantamento bibliográfico proposto pelo primeiro objetivo. Tais informações possibilitaram, também, a criação de hipóteses que podem responder os motivos pelos quais algumas técnicas são mais utilizadas em relação a outras.

O terceiro objetivo, que se refere a aplicação do questionário, foi atingido com a colaboração de 50 profissionais da área de tecnologia da informação por meio de redes sociais.

O quarto objetivo específico foi alcançado após o cruzamento dos dados coletados pelo questionário aplicado e permitiu identificar os seguintes fatores como influenciadores na aplicação de diferentes técnicas de levantamento de requisitos:

- I. Quanto maior o cliente atendido, mais técnicas de levantamento de requisitos são utilizadas;
- II. Quanto menor a empresa, mais técnicas serão empregadas na fase de levantamento de requisitos;
- III. Profissionais ligados à área comercial, quando responsáveis por elaborar documentos de requisitos, fazem uso de um leque maior de técnicas se comparados a outros cargos;
- IV. Profissionais formados no curso de Informática são os que utilizam mais técnicas no seu dia-a-dia, os formados em Bacharelado em Ciências da Computação, menos.

## 6.2 SUGESTÕES DE TRABALHOS FUTUROS

Como possíveis trabalhos futuros pode-se enumerar os seguintes:

- I. Comparar a grade curricular do curso de informática com as de outros cursos buscando entender por que profissionais formados nesse curso fazem uso de mais técnicas em relação aos outros;
- II. Estudar a fase de levantamento de requisitos no mercado atual. Entender quanto do tempo total dos projetos tem sido dedicado a ela. Analisar estudos de caso de projetos reais bem e malsucedidos e compara-los, analisando as técnicas utilizadas para levantamento de informações de escopo;
- III. Entender o porquê da diferença entre as empresas de grande e pequeno porte em relação ao uso das técnicas, respondendo perguntas como:
  - a. Profissionais de grandes empresas tem menos tempo para levantar requisitos?
  - b. Profissionais de empresas de pequeno porte são mais cobrados quanto a qualidade dos requisitos levantados?



## REFERÊNCIAS

- AURUM, A., WOHLIN, C., **Engineering and Managing Software Requirements**, SpringerVerlag, 2005
- NANDITA KANJIRATH, **Pontos cegos em Engenharia de Requisitos**, Biblioteca Virtual PMI, 2011
- ÁVILA, ANA LUIZA & SPÍNOLA, RODRIGO OLIVEIRA. Introdução à Engenharia de Requisitos. Disponível em: <http://www.devmedia.com.br/artigo-engenharia-de-software-introducao-a-engenharia-de-requisitos/8034>, 2008 Acesso em: 13 maio 2016.
- BASTOS JUNIOR, P. R. O. **Elicitação de requisitos de software através da utilização de questionários**. Rio de Janeiro: PUC-Rio, Departamento de Informática, 2005
- CAMACHO, C. **Gerenciando Conflitos em Reuniões: Uma Estratégia para a Elicitação de Requisitos de Software**. Rio de Janeiro: PUC-Rio, Departamento de Informática, 2005
- COCKBURN, A., **Escrevendo Casos de Uso Eficazes: Um guia prático para desenvolvedores de software**, Porto Alegre: Bookman, 2005.
- DE PÁDUA F WILSON. **Engenharia de Software: fundamentos, métodos e padrões**. 8ª ed. Editora LTC, 2000.
- DE SOUSA L. VINÍCIUS. Desenvolvimento de Software Dirigido por Caso de Uso. Disponível em: <http://www.devmedia.com.br/desenvolvimento-de-software-dirigido-por-caso-de-uso/9148>, 2008. Acesso em: 16 junho 2016
- ENGHOLM JUNIOR, H., **Engenharia de software na prática**. São Paulo: Novatec, 2010
- FOWLER, M. **UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos**. 3. ed. Porto Alegre: Bookman, 2005.
- GIL, Antônio Carlos. **Como Elaborar Projetos de Pesquisa**. 4. ed. São Paulo: Atlas, 2002
- [http://vqv.com.br/es/esm03\\_RequisitosNaoFuncionais.pdf](http://vqv.com.br/es/esm03_RequisitosNaoFuncionais.pdf), 2008. Acesso em: 25 Junho 2016

KENDALL, K.E., KENDALL, J.E. **Systems Analysis and Design**, Prentice Hall, 8. ed, 2010

KIRSCHBAUM, C. **Decisões entre pesquisas quali e quanti sob a perspectiva de mecanismos causais**. Revista Brasileira de Ciências Sociais, Vol. 28, 2013, p. 179-193.

KOTONYA, G., SOMMERVILLE, I., **Requirements Engineering: processes and techniques**. Chichester, Inglaterra: John Wiley, 1998.

LARMAN, C. **Utilizando UML e Padrões: uma introdução à análise e ao projeto orientados a objetos e ao Processo Unificado**. 2. ed. Porto Alegre: Bookman, 2004.

LAUESEN, S. **Software requirements styles and techniques: elicitation**. England: A Personal Education Limited, 2002

LEFFINGWELL, D. **Calculating the return on investment from more effective requirements management**. *American Programmer*, Arlington, v. 10, n. 4, p.13-16, 1997.

MALHOTRA, N.K. **Pesquisa de marketing: uma orientação aplicada**. 3.ed. Porto Alegre: Bookman, 2001.

MENDES DA SILVA FILHO, ANTONIO. Requisitos não funcionais. **Disponível em:**

Nuseibeh, B. & Easterbrook, S. **Requirements engineering: a roadmap**, in 'ICSE '00: Proceedings of the Conference on The Future of Engineering', ACM Press, New York, NY, USA, pp. 35-46, 2000

PARASURAMAN, A. **Marketing research**. 2. ed., Addison Wesley Publishing Company, p. 21-60, 1991

PRESSMAN, R. S. **Engenharia de Software: uma abordagem profissional**. 7. ed. Porto Alegre: McGraw-Hill, 2011.

ROESCH, S. M. A. **Projetos de Estágio e de Pesquisa em Administração**. 2. ed. São Paulo, Atlas, 1999

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011

SOMMERVILLE, I., SAWYER, P. **Requirements engineering: a good practice guide**. Chichester, England: John Wiley, 1997

INFOJOBS,

2016

[http://www.infojobs.com.br/artigos/Gerente\\_Comercial\\_Interno\\_2127.aspx](http://www.infojobs.com.br/artigos/Gerente_Comercial_Interno_2127.aspx)

Acesso em 18 agosto 2016

## APÊNDICE A - Questionário de Pesquisa

1. Qual sua formação acadêmica?	
	Ensino médio
	Análise / Desenvolvimento de Sistemas
	Bacharelado em Ciência da Computação
	Informática
Outro:	

2. Qual seu cargo?	
	Desenvolvedor Junior
	Desenvolvedor Pleno
	Desenvolvedor Sênior
	Arquiteto de Sistemas
	Analista de Requisitos
	Analista de Testes
	Gerente de Projetos
	Comercial
Outro:	

3. Qual o porte da empresa na qual você trabalha?	
	Pequeno
	Médio
	Grande

4. Qual o porte dos clientes que você geralmente atende?	
	Pequeno
	Médio
	Grande

5. Dentre as técnicas de levantamento de requisitos abaixo relacionadas, assinale a frequência com que você as utiliza:				
	<b>Nunca</b>	<b>Raramente</b>	<b>Quase sempre</b>	<b>Sempre</b>
Entrevistas				
Etnografia				
Validação				

Casos de uso				
Cenários				
Workshops (Brainstorming)				
Prototipagem				
Questionários				