

Universidade Federal de Pernambuco
Centro de Informática

REQE – Uma Metodologia para Medição de Qualidade de Aplicações Web na Fase de Requisitos

Por

Tiago Pessoa Costa Reis

Dissertação de Mestrado

Recife

Junho de 2004

TIAGO PESSOA COSTA REIS

REQE – Uma Metodologia para Medição de Qualidade de Aplicações Web
na Fase de Requisitos

*Dissertação apresentada à Coordenação da
Pós-Graduação em Ciência da Computação
do Centro de Informática como parte dos
requisitos para obtenção do título de Mestre
em Ciência da Computação.*

Orientador: Jaelson Freire Brelaz de Castro

Recife
Junho, 2004

Dedico este trabalho aos meus maravilhosos pais, Carminha e Cleofas Reis, e à minha querida esposa, Pollyanna, que sempre me apoiaram e me deram forças para as conquistas da vida.

Resumo

O desenvolvimento de aplicações na internet tornou-se uma prática na comunidade de Tecnologia da Informação. As metodologias de desenvolvimento devidamente reconhecidas pela comunidade muitas vezes não são utilizadas. De forma análoga, os modelos de processo e metodologias de avaliação que promovem a compreensão e a melhora na qualidade desses projetos também não são adotados. Esta ausência de processos acarreta práticas de desenvolvimento *ad hoc*, prejudicando tanto as previsões de custo e tempo, quanto à qualidade do produto final.

Com o intuito de melhorar esse cenário é necessário investir em qualidade. Contudo, não podemos medir o que não podemos compreender, bem como não é possível medir o que não podemos controlar. Também não podemos melhorar um software, a menos que possamos medi-lo. Avaliar é medir. Por meio da medição é possível analisar e compreender o que está errado ou ausente. Neste contexto, esta dissertação propõe uma metodologia baseada na Engenharia de Requisitos para avaliar a qualidade de aplicações Web.

As metodologias de avaliação de qualidade de aplicações Web em geral efetuam as medições dos produtos nas fases finais do processo de desenvolvimento. A principal contribuição desta pesquisa é a metodologia REQE (Requirements Engineering Quality Evaluation), que visa antecipar esta avaliação. Como esta metodologia se propõe a avaliar a qualidade de aplicações Web numa fase inicial do processo de desenvolvimento, temos como benefício a possibilidade de descobrir erros de uma forma antecipada.

A partir de um documento de requisitos bem elaborado algumas características do sistema em desenvolvimento já podem ser previamente identificadas e avaliadas. À medida que os erros são identificados, tem-se a possibilidade de corrigi-los. Portanto, o objetivo da metodologia proposta é além de avaliar a qualidade de aplicações Web na fase de requisitos, contribuir com a construção de produtos de melhor qualidade.

Abstract

The development of applications on the Internet has become a standard practice in the academic Information Technology community. The development methodologies duly recognized by that community are very often not used. Similarly the process models and the assessment methodologies that promote comprehension and the improvement in quality of those projects are not adopted either. This lack of process results in practices of ad hoc development which damages not only forecasts of cost and time but the quality of the final product as well.

In order to improve this scenario, it is necessary to invest in quality. However, it is not possible to measure what cannot be understood, in the same way as it is not possible to measure what cannot be controlled. In addition, a piece of software cannot be improved unless it can be measured. It is by means of measurement that it is possible to analyze and understand what is wrong or what is lacking. In this context, this paper proposes a methodology based on Requirements Engineering to appraise the quality of Web applications.

In general, the quality assessment methodologies of Web applications evaluate the measurements of products during the final phases of the process being development. The main contribution of this research is the methodology called REQE (Requirements Engineering Quality Evaluation), which aims at anticipating this evaluation. As our proposition is to evaluate the quality of Web applications at an initial phase of the development process, the advantage is the possibility of foreseeing mistakes.

From a well-prepared requirements document some characteristics of the system under development can be identified and evaluated. This means that as soon as those mistakes are identified, it is possible to correct them. Therefore, besides appraising the quality of Web applications during the requirements phase, another aim of this paper is to contribute to the construction of better quality products.

Índice

1 INTRODUÇÃO.....	1
1.1 MOTIVAÇÃO	2
1.2 ESTRUTURA DA DISSERTAÇÃO	4
2 AVALIAÇÃO DE QUALIDADE DE APLICAÇÕES WEB: DIFICULDADES E CARACTERÍSTICAS.....	7
2.1 INTRODUÇÃO	8
2.2 SITUAÇÃO	8
2.3 SISTEMAS BASEADOS NA WEB VERSUS SISTEMAS TRADICIONAIS	9
2.4 TIPOS DE APLICAÇÕES WEB.....	12
2.5 INTERFACE COM O USUÁRIO	14
2.6 DIFICULDADES NA AVALIAÇÃO DE QUALIDADE DE APLICAÇÕES WEB.....	16
2.7 SUBJETIVIDADE E OBJETIVIDADE NOS SISTEMAS DE AVALIAÇÃO	19
2.8 NECESSIDADE DE AVALIAR PARA COMPREENDER E MELHORAR	20
2.8.1 Avaliar a Qualidade no Produto	20
2.8.2 Avaliar a Qualidade na Fase de Requisitos	21
2.9 ESTADO DA ARTE DA AVALIAÇÃO DE QUALIDADE.....	22
2.9.1 Web-site QEM.....	26
3 REQUISITOS.....	34
3.1 INTRODUÇÃO	35
3.2 CLASSIFICAÇÃO DOS REQUISITOS	35
3.3 ENGENHARIA DO DOMÍNIO	38
3.4 REQUISITOS NÃO-FUNCIONAIS	44
3.4.1 Requisitos Não-Funcionais para Web	51
3.4.2 Abordagens para Modelar Requisitos Não-Funcionais	54
3.4.2.1 NFR Framework	58
3.4.3 Modelos de Qualidade	62
3.5 DOCUMENTO DE REQUISITOS	63
3.5.1 Template do Documento de Requisitos Proposto	69
4 REQE – UMA METODOLOGIA PARA MEDIÇÃO DA QUALIDADE DE APLICAÇÕES WEB NA FASE DE REQUISITOS	71
4.1 INTRODUÇÃO	72
4.2 REPRESENTAÇÃO DAS CARACTERÍSTICAS, SUBCARACTERÍSTICAS E ATRIBUTOS DE QUALIDADE	73
4.2.1 Escolha do Domínio da Aplicação e do Perfil do Interessado.....	74

4.2.2 Especificação da Árvore de Características, Subcaracterísticas e Atributos de Qualidade.....	76
4.3 ESPECIFICAÇÃO DESCRITIVA DA ÁRVORE DE CARACTERÍSTICAS, SUBCARACTERÍSTICAS E ATRIBUTOS DE QUALIDADE	78
4.4 ASSOCIAÇÃO DE PESOS AOS NÓS	82
4.5 ASSOCIAÇÃO DE ESCORES AOS ATRIBUTOS	87
4.6 CÁLCULO GERAL	90
5 EXEMPLO	95
5.1 INTRODUÇÃO	96
5.2 REPRESENTAÇÃO DAS CARACTERÍSTICAS, SUBCARACTERÍSTICAS E ATRIBUTOS DE QUALIDADE	97
5.3 ESPECIFICAÇÃO DESCRITIVA DA ÁRVORE DE CARACTERÍSTICAS, SUBCARACTERÍSTICAS E ATRIBUTOS DE QUALIDADE	101
5.4 ASSOCIAÇÃO DE PESOS AOS NÓS	107
5.5 ASSOCIAÇÃO DE ESCORES AOS ATRIBUTOS	113
5.6 CÁLCULO GERAL	122
6 CONCLUSÕES E TRABALHOS FUTUROS.....	126
6.1 PRINCIPAIS CONTRIBUIÇÕES	127
6.2 TRABALHOS FUTUROS	128
REFERÊNCIAS BIBLIOGRÁFICAS.....	130
APÊNDICE A - DOCUMENTO DE REQUISITOS PROPOSTO	141
APÊNDICE B - DOCUMENTO DE REQUISITOS DO EXEMPLO	152

Índice de Figuras

Figura 3.1 – Engenharia do Domínio x Engenharia de Software	40
Figura 3.2 – Necessidades dos usuários x RNFs	45
Figura 3.3 – Classificação dos RNFs segundo Sommerville.....	47
Figura 3.4 - Abordagens para tratamento de RNF segundo [CYSNEIROS01].....	49
Figura 3.5 – Fatores que influenciam RNFs	57
Figura 3.6 – Propriedades mensuráveis dos RNFs	59
Figura 3.7 - Exemplo de grafo SIG.....	61
Figura 3.8 – RNFs vistos como características de qualidade	63
Figura 3.9 – Tabela das características e subcaracterísticas definidas pela ISO [ISO9126]	64
Figura 3.10 – Estrutura de um documento de requisitos segundo IEEE 830.	69
Figura 3.11 – Template do documento de requisito proposto	70
Figura 4.1 – Parte de uma <i>ACSAQ</i>	78
Figura 4.2 – Planilha de característica	80
Figura 4.3 – Planilha de subcaracterística.....	80
Figura 4.4 – Planilha de atributo de qualidade	81
Figura 4.5 – Exemplos de árvores permitidas.....	83
Figura 4.6 – Exemplos de árvores permitidas.....	84
Figura 4.7 – Exemplos de árvores proibidas.....	84
Figura 4.8 – Exemplos de árvores permitidas.....	85
Figura 4.9 – Exemplos de árvores proibidas.....	86
Figura 4.10 – Tabela de conformidade x escore	88
Figura 4.12 – Fórmula para cálculo da nota de nó pai.....	90
Figura 4.11 – Exemplo de uma <i>ACSAQ</i> com os atributos destacados.....	91
Figura 4.13 – Exemplo de cálculo geral	92
Figura 4.14 – Exemplo incorreto de cálculo geral.....	93
Figura 4.15 – Exemplo de cálculo geral	94
Figura 5.1 – <i>ACSAQ</i> do exemplo de sites acadêmicos, adaptada de [OLSINA99].....	99
Figura 5.2 – Planilha preenchida da característica <i>Usabilidade</i>	102
Figura 5.3 – Planilha preenchida da característica <i>Funcionalidade</i>	103
Figura 5.4 – Planilha preenchida da subcaracterística <i>Entendimento global do site</i>	103
Figura 5.5– Planilha preenchida da subcaracterística <i>Ajuda ao usuário</i>	104
Figura 5.6– Planilha preenchida da subcaracterística <i>Indicador de atualização</i>	104
Figura 5.7– Planilha preenchida da subcaracterística <i>Organogramas</i>	105
Figura 5.8 – Planilha preenchida do atributo <i>Mapa do Site</i>	105
Figura 5.9 – Planilha preenchida do atributo <i>FAQ</i>	105
Figura 5.10 – Planilha preenchida do atributo <i>Entrada</i>	106
Figura 5.11 – Planilha preenchida do atributo <i>Plano de ensino</i>	106
Figura 5.12 – Planilha preenchida do atributo <i>Das provas</i>	106
Figura 5.13 – Subcaracterísticas de <i>Usabilidade</i> e <i>Funcionalidade</i> com pesos	108
Figura 5.14 – Atributos de <i>Tarefas diversas</i> com pesos.....	109
Figura 5.15 – Parte da <i>ACSAQ</i> de <i>Informações da infra-estrutura acadêmica</i>	109
Figura 5.16 – Árvore de sites acadêmicos com pesos	110
Figura 5.17 – Árvore de sites acadêmicos com pesos e escores.....	113
Figura 5.18 – <i>ACSAQ</i> de sites acadêmicos com pesos e escores em todos os nós.....	122

1 Introdução

Neste capítulo introdutório apresentaremos as motivações deste trabalho e como está estruturada esta dissertação.

1.1 Motivação

O desenvolvimento de aplicações na internet tornou-se uma prática na comunidade de Tecnologia da Informação. No início eram apenas sites estáticos com objetivos de disseminar informações. Ao passar do tempo foram surgindo outros tipos de aplicações, tão complexas quanto qualquer outro software tradicional. Praticamente tudo o que pode ser desenvolvido *stand-alone* pode ser adaptado à Web. Tem-se que estar atento apenas a alguns fatores que funcionam como limitadores de certas aplicações Web, como, por exemplo, largura de banda. Entretanto, a cada dia esses recursos estão cada vez menos escassos.

Muito se falou da Crise de Software, gerada pelo uso inadequado de processos de desenvolvimento. As aplicações Web também sofreram e ainda sofrem desse “mal”. As metodologias de desenvolvimento devidamente reconhecidas pela comunidade muitas vezes não são utilizadas. De forma análoga, os modelos de processo e metodologias de avaliação que promovem a compreensão e a melhora na qualidade desses projetos também não são adotados. Esta ausência de processos acarreta práticas de desenvolvimento *ad hoc*, prejudicando tanto as previsões de custo e tempo, quanto à qualidade do produto final [GINIGE01].

Com o intuito de melhorar esse cenário faz-se necessário investir em qualidade. Contudo, não podemos medir o que não podemos compreender, bem como não é possível medir o que não podemos controlar. Também não podemos melhorar um software, a menos que possamos medi-lo. Avaliar é medir. Por meio da medição é possível analisar e compreender o que está errado ou ausente. Por exemplo, na Engenharia de Requisitos avaliamos a qualidade de um artefato de software confrontando um conjunto de características e atributos do software com os requisitos de qualidade estabelecidos.

O nosso trabalho, que foi desenvolvido no contexto do Projeto WEST (Web-Oriented Software Technology) [WEST01], baseia-se na metodologia *Web-site QEM* [OLSINA00b], que avalia os produtos de software na fase final do processo de desenvolvimento, isto é,

quando os mesmos já estão na fase operativa. A metodologia *Web-site QEM* foi desenvolvida com o propósito de ser uma estratégia eficaz para avaliar e analisar a qualidade de sites e aplicações Web em geral. Está baseada em um modelo hierárquico das características de mais alto nível previstas na norma ISO [ISO9126], a saber: Usabilidade, Funcionalidade, Confiabilidade, Eficiência, Manutenibilidade e Portabilidade. A partir destas características, se derivam subcaracterísticas e, a partir destas, seguindo um processo de decomposição hierárquico, se especificam atributos. O resultado do processo de avaliação (e eventualmente de comparação) pode ser interpretado como o grau de satisfação dos requisitos de qualidade. Nesta dissertação estamos propondo uma metodologia (*Requirements Engineering Quality Evaluation*) que visa antecipar essa avaliação de qualidade, tendo como um dos possíveis benefícios a descoberta de erros numa fase mais inicial do processo de desenvolvimento.

De acordo com Pressman [PRESSMAN02], o custo de correção de erros na fase de projeto é cerca de três a seis vezes mais alto do que na fase de definição de requisitos. E este custo aumenta ainda mais quando a correção de erros é realizada em fases mais avançadas do processo de desenvolvimento, como a fase de operação.

Portanto, por que esperar para a última fase do processo de desenvolvimento o que se quer medir, sabendo-se que quanto mais se distancia das fases iniciais mais caro ficará sua posterior correção? O leitor pode estar questionando que não há como medir as características na fase inicial uma vez que ainda não existe o produto. No entanto, existe um documento que pode representá-lo, o documento de requisitos, que especifica tanto os requisitos funcionais quanto os não-funcionais (por exemplo, vide [IEEE830]).

O documento de requisitos será decisivo para o desenvolvimento de um bom produto final. Estamos levando em consideração que o desenvolvimento vai realmente se apoiar neste documento. Tomando isso como premissa será possível fazer essa pré-avaliação. Os requisitos que forem surgindo durante o processo podem ser incorporados à metodologia.

Partimos do princípio da existência de um documento de requisitos bem elaborado, para que se tenha algo a ser analisado. Sabe-se que existem certas propriedades, chamadas de emergentes, pertinentes ao sistema como um todo que apenas surgem quando todos os subsistemas estão integrados, isto é, na existência do produto final. Essas propriedades realmente têm que esperar um pouco mais para serem investigadas, mas isso não invalida a nossa proposta, uma vez que muitas outras podem ser descobertas e corrigidas nas fases iniciais do processo de desenvolvimento.

1.2 Estrutura da Dissertação

Além deste capítulo introdutório, este trabalho possui outros seis capítulos e dois apêndices, como veremos a seguir:

Capítulo 2 - Avaliação de qualidade de Aplicações Web: Dificuldades e Características

No capítulo 2 realizamos uma breve discussão da situação atual das aplicações Web. Comparamos os sistemas baseados na Web com os sistemas tradicionais, identificando características particulares daqueles. Fazemos um levantamento sobre os tipos de aplicações Web e da importância da interface com o usuário desses sistemas. Além disso, analisamos a problemática da avaliação de qualidade de aplicações Web, citando fatores que influenciam a sua subjetividade. Fortalecemos a necessidade da utilização de métodos de avaliação para melhor compreender a qualidade dos sistemas. Apresentamos o estado da arte na avaliação de qualidade, bem como a metodologia *Web-site QEM* [OLSINA00b], sobre a qual se apóia este trabalho.

Capítulo 3 - Requisitos

Neste capítulo falamos de requisitos de uma forma geral. Também apresentamos os conceitos básicos da Engenharia do Domínio, uma vez que a nossa metodologia (REQE) recomenda a utilização de suas técnicas no sentido de identificar o que é importante no

contexto de um determinado domínio de aplicação. Nos concentramos em um tipo específico de requisitos que nos interessa bastante: os requisitos não-funcionais. Estudamos esse tipo particular de requisito inserido nas aplicações Web. Mostramos uma metodologia para modelar esse tipo de requisito. Por fim propomos um template para um documento de requisitos.

Capítulo 4 – REQE – Uma Metodologia para Medição da Qualidade de Aplicações Web na Fase de Requisitos

Este é o capítulo mais importante desta dissertação. Apresentamos aqui a metodologia **REQE** (Requirements Engineering Quality Evaluation) capaz de avaliar a qualidade de aplicações Web na fase de requisitos. Mostraremos passo a passo como isso é possível, explicando em detalhes todo o processo.

Capítulo 5 – Exemplo

Aqui, utilizamos nossa proposta para avaliar uma aplicação (site acadêmico).

Capítulo 6 – Conclusões e Trabalhos Futuros

No capítulo 6 expomos as conclusões e os futuros trabalhos a serem desenvolvidos.

Apêndice A – Documento de Requisitos Proposto

Neste apêndice apresentamos o template do documento de requisitos proposto por este trabalho.

Apêndice B – Documento de Requisitos do Exemplo

Apresentamos o documento de requisitos que serve de base para a aplicação da metodologia REQE no contexto do exemplo apresentado no capítulo 5.

2 Avaliação de Qualidade de Aplicações Web: Dificuldades e Características

Neste capítulo estudaremos as aplicações Web e como avaliar sua qualidade. Identificaremos algumas peculiaridades que não estão presentes nos sistemas tradicionais e avaliaremos alguns tipos de aplicações Web. Faremos um estudo da interface com o usuário, por ser uma característica de forte influência no contexto das aplicações Web. Descreveremos um pouco sobre as dificuldades de uma avaliação de qualidade, principalmente com relação ao fator subjetividade. Descreveremos o estado da arte na avaliação de qualidade, em particular a metodologia *Web-site QEM*.

2.1 Introdução

Ainda não existe uma metodologia de avaliação de qualidade de aplicações Web amplamente difundida, reconhecida e utilizada como padrão de avaliação na comunidade de Engenharia de Software. O uso de uma eficiente metodologia de avaliação possibilita a identificação de fatores e características de qualidade, que podem ser incorporados em projetos futuros, contribuindo com o desenvolvimento de artefatos Web de melhor qualidade.

Na Seção 2.2 apresentamos a situação atual das aplicações Web. Na Seção 2.3 fazemos um estudo comparativo dos sistemas tradicionais versus os sistemas desenvolvidos para a Web. Citamos algumas classificações existentes na literatura a fim de separar as aplicações Web em grupos (Seção 2.4). Destacamos ainda a interface com o usuário como uma área de especial importância no contexto das aplicações Web (Seção 2.5). Abordamos as dificuldades inerentes à atividade de avaliar aplicações Web (Seção 2.6), principalmente com relação ao componente subjetividade (Seção 2.7). Falamos da importância de se avaliar para melhor compreender a qualidade de um artefato de software (Seção 2.8). Apresentamos o estado da arte na avaliação de qualidade de aplicações Web na Seção 2.9 e em particular a metodologia *Web-site QEM* [OLSINA00b] (Seção 2.9.1).

2.2 Situação

O crescimento da internet tem tido um impacto significativo sobre os diversos setores de negócios, bem como nas áreas de educação, governo, entretenimento e sobre nossa vida pessoal e profissional [OLSINA00b]. Muitos sistemas de informação legados estão sendo migrados para o ambiente Web. A cada dia que passa, aumenta o percentual da população com acesso à internet. Surgem mudanças na nossa vida pessoal, no modo de nos relacionarmos. Às vezes deixamos de conversar com pessoas que estão a poucos metros de nós para enviarmos mensagens eletrônicas. Em contrapartida, temos a mesma facilidade de nos comunicar com pessoas independente de suas localizações, quer estejam próximas de nós, quer em outro continente. O comércio eletrônico, através da internet, cresceu e rompeu limites geográficos. Como estamos cada vez mais “conectados” a influência que a internet exerce sobre as pessoas tende a aumentar ainda mais. Utilizamos sites Web nas mais

variadas situações: para efetuarmos uma transação bancária, conversarmos com pessoas, nos mantermos informados etc. Começamos a ter uma certa “dependência” de várias aplicações da internet. Além das já existentes, uma vasta variedade de novas aplicações distribuídas está emergindo a partir deste novo ambiente. E esse conjunto de aplicações tende a crescer. É indiscutível o incremento da popularidade e aceitação de aplicações centradas na Web.

Infelizmente, a verdade é que muitas das aplicações que utilizamos não foram construídas de forma “organizada”, centradas em processos e metodologias comprometidos com a qualidade, principalmente no que diz respeito à Engenharia de Requisitos [ESCALONA03] [ZANLORENCIO02]. Como consequência, encontramos muitos problemas no uso dessas aplicações. Com o intuito de melhorar esse panorama existem inúmeros grupos de pesquisa interessados em modelar e desenvolver aplicações Web [LIMA03] [PASTOR03] [SOUZA03]. A ausência de metodologias reconhecidas pela Engenharia de Software no desenvolvimento de aplicações Web está contribuindo para a Crise de Software [GIBBS94], oriunda dos anos sessenta, ceder espaço para a Crise de Software Web. Está ocorrendo uma verdadeira inversão das fases do processo de desenvolvimento. Frequentemente o que deveria ser uma manutenção, se transforma numa verdadeira reformulação dos requisitos básicos do sistema, acarretando além de trabalhos desnecessários, aumentos exorbitantes de custos e estouros nos prazos de conclusão.

A seguir faremos um estudo comparativo das aplicações Web com os sistemas tradicionais. Muitas técnicas válidas para o desenvolvimento e avaliação de qualidade destes também podem ser utilizadas no contexto das aplicações Web.

2.3 Sistemas Baseados na Web versus Sistemas Tradicionais

O desenvolvimento de aplicações Web envolve uma certa mudança na apresentação da informação. A quantidade de pessoas que têm acesso a um sistema disponibilizado na internet é bem maior do que em um sistema tradicional, convencional. Obviamente que esse público pode ser restrito através do uso de mecanismos de acesso. Mas quando realmente se quer atingir a maior quantidade de pessoas, as aplicações Web levam uma

enorme vantagem. De fato os sistemas Web estão se tornando cada vez mais universais. Muitas organizações têm demonstrado interesse em migrar seus sistemas tradicionais para a internet. As razões para esta mudança incluem:

- A alta popularidade da tecnologia entre clientes em potencial;
- O relativo baixo custo de investimento necessário, com relação aos resultados esperados;
- A alta possibilidade de alcance desses resultados;
- Incremento das relações comerciais, uma vez que estarão ao alcance de todos os usuários da internet [HECK98].

As principais diferenças entre aplicações Web e as tradicionais referem-se a questões sobre navegação, organização da interface e implementação. Algumas dessas características são [GINIGE01]:

- A maioria dos sistemas Web é orientada à documentação, contendo páginas Web estáticas e dinâmicas;
- Sistemas Web são focados na aparência, exigindo criatividade na apresentação visual e a incorporação de recursos multimídia;
- A maioria dos sistemas Web é dirigida a conteúdo, portanto, desenvolver aplicações Web inclui desenvolver o conteúdo a ser apresentado;
- Geralmente um site Web atende a uma multiplicidade de perfis de usuários, precisando satisfazer às necessidades de usuários com diferentes habilidades e capacidades, aumentando a complexidade da interação homem-máquina, a interface com o usuário e a apresentação das informações;
- A natureza e as características da mídia da Web não estão tão bem entendidas quanto a mídia dos softwares tradicionais;
- Em geral, um projeto Web exige uma maior multiplicidade de tipos e níveis de habilidades (e papéis) do que os participantes de um projeto de software tradicional;

- A maioria dos sistemas Web precisa ser desenvolvida em um curto espaço de tempo (*Web-time*), tornando difícil a adoção do mesmo nível de rigor no planejamento e teste utilizados em sistemas tradicionais;
- Os sistemas Web diferem de sistemas tradicionais quanto ao ambiente em que são disponibilizados, pois o meio onde se executa as aplicações baseadas na Web é mais imprevisível do que o meio onde se roda as aplicações de software tradicionais;

As fases do ciclo de um sistema de informação necessitam ser adaptadas para melhor atenderem ao desenvolvimento de aplicações Web, desde os requisitos (com grande atenção aos requisitos não-funcionais e de interface), até o tratamento de questões como audiência da aplicação e compatibilidade de ferramentas, além da preocupação excessiva com segurança. Deste modo, é necessário um forte investimento em processos de desenvolvimento de aplicações Web, para melhor atenderem às necessidades dos usuários.

Está evidente que o “mundo Web” possui características específicas [VASUDEVAN96]. Dentro do domínio de aplicações Web podemos ainda identificar tipos diferentes de sistemas [TURAU98] [GELLERSON97] que, apesar de possuírem características semelhantes, diferem substancialmente no objetivo e nas preocupações e princípios sobre os quais se baseiam.

Apesar das diferenças, muitos dos problemas relacionados a sistemas Web seguem estratégias de sistemas tradicionais, como descritos em [CLEMONS91] e [PORTER85]. Desta forma muitas das soluções relacionadas a sistemas tradicionais também serão aplicadas a sistemas Web [SOUZA02].

A seguir veremos dois exemplos de como as aplicações Web podem ser agrupadas. Essas não são as únicas formas de classificá-las. No entanto, independente da classificação adotada, as aplicações de um mesmo grupo devem possuir características semelhantes.

2.4 Tipos de Aplicações Web

Alguns pesquisadores classificaram as aplicações Web em categorias. Um deles foi Powell [POWELL98] que propôs cinco classes. Cada novo tipo de aplicação listado possui outras características além de todas as presentes no tipo anterior. Esta classificação foi extraída de [OLSINA00b]:

- Site estático: é o tipo mais simples de uma aplicação Web, correspondente a uma coleção de páginas estáticas. A partir de uma página é possível acessar as outras por meio de *links*. Como exemplo dessas aplicações temos os documentos, ou informações editadas e publicadas em formato HTML;
- Site estático com formulários de entrada: esse tipo de site provê um nível de interação básico implementado por meio de formulários de entrada. Isso permite uma maior interação por parte dos usuários. Exemplos desse tipo são questionários, livros de visita, comentários e sugestões etc;
- Site com acesso a dados dinâmicos: o usuário, por meio de consultas, pode acessar bases de dados. As respostas das consultas são geradas dinamicamente e geralmente apresentadas no formato de documentos HTML, DHTML ou XML;
- Site criado dinamicamente: geralmente são criados nos casos em que o site deve prover conteúdo personalizado, dependendo da instância do usuário;
- Aplicação de software baseada na Web: este tipo de aplicação pode ser entendido como um software qualquer acessado através da internet.

É importante salientar que esta não é uma classificação tida como padrão pela comunidade acadêmica, embora tenha sido utilizada em algumas situações. Achamos interessante apresentar ao menos outro tipo de classificação, que divide as aplicações em três categorias [ARAUJO01]:

- Aplicações de páginas de informação: também denominadas de hipertexto orientado a publicação ou aplicações tipo quiosque. Este tipo de aplicação Web é composto basicamente de páginas de informação, ou seja, de páginas HTML ou DHTML,

juntamente com componentes sockwave ou flash, animações, sons, vídeos, etc. Nenhuma computação efetiva que suporte a execução de um negócio é realizada. Estas aplicações têm a função de prover/disseminar informações, priorizando o conteúdo e a forma de apresentação. Exemplo: Web-sites institucionais e pessoais.

- Aplicações centradas em banco de dados: basicamente manuseiam dados armazenados em bancos de dados, não possuindo fluxos de processos ou transações complexas. Estas aplicações têm a função de realizar a manutenção e apresentação de dados armazenados em um banco de dados. Por exemplo, as informações são apresentadas como páginas HTML podendo ou não ser geradas dinamicamente, bem como a estrutura de links associada a estas páginas. A prioridade neste tipo de software é a organização e estruturação do banco de dados e a forma como os dados serão disponibilizados e apresentados. Exemplo: Web-sites para cadastro de currículos.
- Aplicações que suportam a realização de transações de negócio: são formadas por componentes de software, responsáveis pela execução de algum tipo de computação que auxilie a realização de um negócio, integrado a páginas de informação. Existe a junção de informações/conteúdo, associados a um negócio, e componentes de software altamente interativos e dinâmicos que se relacionam. Estas aplicações são de grande escala, podendo estar distribuídas através de vários sites, possuem uma arquitetura robusta e complexa, e geralmente possuem interfaces com aplicações legadas e com outras aplicações Web, como, por exemplo, bancos de dados, servidores de transações, etc. Através destes sistemas podem-se processar dados de negócio e providenciar a comunicação e colaboração de pessoas interessadas na realização do mesmo. Nesta categoria a prioridade é a definição da arquitetura do sistema e da aplicação, e o projeto de seus componentes. Exemplo: Web-sites de comércio eletrônico.

Na próxima seção falaremos um pouco sobre a interface com o usuário, assunto bastante presente em qualquer tipo de aplicação Web e de fundamental importância para o sucesso de um projeto.

2.5 Interface com o Usuário

Um dos pontos que merece especial atenção nas aplicações Web é a interface com o usuário. Por isso existe uma área de pesquisa, Interface Homem-Máquina, que estuda com profundidade este assunto. Segundo Landauer [LANDAUER95], 80% do custo total de desenvolvimento são gastos com a manutenção. Contudo, 80% dos custos de manutenção de software ocorrem devido a problemas na interação entre os usuários e o sistema, e não a problemas técnicos. Ele também afirma que 64% destes custos estão relacionados com problemas de usabilidade. Esta situação reforça a importância dos requisitos de interface no desenvolvimento de software. Isto significa que o desenvolvimento de software é uma atividade multidisciplinar, implicando na participação de profissionais de diferentes áreas, não apenas engenheiros de computação, mas também das áreas de design, marketing e psicologia, por exemplo.

Segundo, John Shiple, em [SHIPLE03], antes de construirmos a interface de uma aplicação Web, ou Web-site, deve-se:

- Identificar qual o objetivo da aplicação, depois qual a audiência da mesma, ou seja, avaliar a experiência dos usuários e definir seus papéis;
- Definir os cenários da interface, ou seja, com base nos papéis encontrados ao avaliar a audiência, devem ser identificados os cenários de uso da interface por cada tipo de usuário (papel) e então, escritas as histórias de como será a interação entre o usuário aplicação;
- Analisar interfaces ou sites, similares e/ou concorrentes;
- Identificar o conteúdo visual através da análise dos requisitos funcionais;
- Organizar o conteúdo e definir a estrutura do site;
- Definir a navegabilidade do site;
- Definir os templates que descrevem as páginas HTML;
- Definir o protótipo da aplicação.

Baresi, Garzotto e Paolini, em [BARESI01], propõem o Framework W2000 para projeto de aplicações Web. Este framework é uma integração entre UML e HDM – Hypermedia Design Model [GARZOTTO93] e consiste em: definir estereótipos e customização de diagramas para caracterizar HDM com UML; criar diretrizes para usar UML com o objetivo de especificar alguns dos aspectos dinâmicos e operacionais de aplicações Web; e refinar diagramas de Casos de Uso para descrever requisitos de alto nível, relacionados a aspectos funcionais e navegacionais. Neste framework, cada atividade produz um modelo (representado por diagramas de UML), que descreve alguns aspectos da aplicação.

A análise de requisitos inicia com a identificação das diversas categorias (papéis) de usuários que devem interagir com a aplicação. Esta identificação dos papéis ajuda a organizar os caminhos navegacionais e as operações a serem realizadas pelos usuários. Por exemplo, para um site de uma operadora de cartão de crédito: *Usuários Genéricos* navegam sobre as informações públicas; *Clientes* realizam consultas em seus cartões (saldo, extrato, etc), alteram senha e solicitam alterações cadastrais, segunda via de cartão e inclusão de dependentes; e *Futuros Clientes* submetem propostas e consultam suas propostas.

No Framework W2000 também é produzido o Modelo de Casos de Uso Funcionais, que identifica as principais funções e as associa com os papéis de usuários, e o Modelo de Casos de Uso Navegacional, onde as capacidades de navegação são associadas a cada papel. Os usuários genéricos que navegam pela aplicação Web, mas que não são representados por atores do sistema por não executarem nenhuma funcionalidade da aplicação, aparecem apenas no

Modelo de Casos de Uso Navegacional porque eles apenas navegam através das páginas públicas, não executando nenhuma ação em particular.

Juan Sánchez, propõe em [SÁNCHEZ02] um processo automático de geração dos componentes da interface de uma aplicação baseado em XML (*Extensible Markup Language*) que possibilita gerar as interfaces em diversas ferramentas de desenvolvimento, como Java, VisualBasic, Delphi, etc.

O método proposto por Sanchez chama-se VRU (*Validación de Requisitos de Usuario*) e pode ser integrado aos métodos de desenvolvimento de software que utilizem casos de uso, para identificar requisitos funcionais, e alguma variante dos diagramas de seqüência, para descobrir as interações internas do sistema.

Sanchez também utiliza *Message Sequence Chart (MSC)* [LADKIN94] para descrever e especificar, de um modo formal, os casos de uso da aplicação, do ponto de vista das interações internas, que devem incluir informações necessárias para gerar os formulários de interface com o usuário, associados a cada caso de uso.

Analisando o exposto acima concluímos que as aplicações de software requerem interfaces com usuários aptas a suportar uma grande variedade de tarefas, em diversos contextos. Os componentes de interface para aplicações interativas estão adquirindo, a cada dia, mais importância nos meios acadêmico e industrial. Isto ocorre porque existem diversas aplicações que executam tarefas similares, mas os usuários tendem a escolher aquelas que são mais fáceis de usar e, conseqüentemente, mais eficientes e produtivas, além de reduzir as necessidades de treinamento e o tempo de execução.

Nas próximas seções apresentaremos temas mais relacionados à avaliação de qualidade de aplicações Web propriamente dita.

2.6 Dificuldades na Avaliação de Qualidade de Aplicações Web

Em termos gerais, a atividade de avaliar a qualidade de um artefato de software é bastante polêmica. Muitas vezes estão centradas na intuição dos avaliadores, ao invés de estarem baseadas em métricas e modelos quantitativos de qualidade, modelos de estruturação de características e atributos, e métodos quantitativos de pontuação [OLSINA00b]. Há casos em que os avaliadores não detêm conhecimento necessário para julgamentos. Existem, por exemplo, alguns concursos para escolha das melhores aplicações Web mediante votação popular [IBEST04]. Essas decisões são dirigidas pelo juízo do usuário leigo, e não guiadas pelo conhecimento de especialistas.

Uma outra questão diz respeito a fase em que as avaliações de qualidade devem ser executadas. Existem vantagens e desvantagens de aplicar um modelo em cada uma das fases do processo de desenvolvimento. Quanto mais no início é aplicado o modelo de avaliação mais fácil e barato é a correção dos problemas identificados. Porém, quanto mais próximo das fases finais do processo de desenvolvimento, mais meios existem para inferir a qualidade.

Citaremos algumas declarações de pesquisadores e de organizações internacionais reconhecidas no meio acadêmico, acerca das avaliações de qualidade:

- “A avaliação de um sistema é um processo de decisão lógica... a meta básica de um sistema de avaliação é determinar o grau em que um sistema cumpre um conjunto específico de requisitos” [DUJMOVIC82];
- “Definir a qualidade de um software para um sistema é equivalente a definir uma lista de atributos de qualidade de software requeridos por este sistema” [IEEE1061];
- “Para medir atributos de qualidade de software se deve identificar um conjunto apropriado de métricas” [IEEE1061];
- “As métricas de software medem atributos específicos de um produto de software ou de um processo de desenvolvimento de software” [GRADY94]

A qualidade de um produto pode ser expressa em termos de seis características de alto nível de abstração: usabilidade, funcionalidade, confiabilidade, eficiência, portabilidade e

manutenibilidade [ISO9126]. Estas podem ser decompostas, de forma a descrever mais objetivamente a qualidade de um software.

Essas características não são diretamente mensuráveis, por isso devem ser refinadas e subdivididas. Os atributos de qualidade originados a partir destas características podem ser medidos através da utilização de escalas propostas [FENTON97], o que torna a atividade de avaliação mais objetiva. Esta abordagem de decompor as características de qualidade é amparada pela ISO [ISO14598-5], que relata: “A qualidade dos produtos de software podem ser descritas em termos de características de qualidade como as definidas na ISO [ISO9126]. Não obstante, o estado da arte em medição de software é tal que, em geral, não são práticas de medições diretas dessas características. O que é possível é valorar essas características sobre a base de medidas de atributos do produto de menor nível de abstração. E neste contexto o avaliador pode usar sua experiência na Engenharia de Software para realizar a valoração”.

Nesta dissertação consideramos que o padrão de qualidade de produtos poderá ser avaliado através das características, definidas inicialmente em um alto nível de abstração. A partir dessas, subcaracterísticas devem ser propostas, e assim sucessivamente, sendo possível trabalhar com atributos de qualidade e suas respectivas métricas cada vez mais concretas [FENTON97] [LAFUENTE01].

No entanto, vale lembrar que “o uso de métricas de software não elimina a necessidade do juízo humano na avaliação de software” [IEEE1061]. Segundo Olsina [OLSINA00b], para especificar um conjunto de metas a ser alcançado por um processo de avaliação, e a partir destas, especificar requisitos de qualidade, os componentes de subjetividade na tomada de decisão são partes necessárias para iniciar o processo. Cabe ressaltar que esta subjetividade é baseada na experiência humana. “Se a necessária experiência está ausente, as metas e requisitos podem ser ambíguos, ausentes, ou pior, não se pode esperar que o processo de avaliação gere resultados corretos e úteis” [DUJMOVIC82] (por mais que o processo, os modelos e os métodos empregados sejam corretos).

Diante da constatação de que o fator subjetividade infelizmente está presente no contexto de avaliações de qualidade, veremos como podemos minimizá-lo. Na próxima seção falaremos da subjetividade e da objetividade nas avaliações de qualidade.

2.7 Subjetividade e Objetividade nos Sistemas de Avaliação

Um ponto bastante polêmico no contexto das avaliações de qualidade é a presença do fator subjetividade. As metodologias possuem alguns papéis associados no processo de avaliação. Alguns destes papéis estão relacionados com os avaliadores, que muitas vezes diante de situações semelhantes, agem de formas diferentes. E isto influencia o processo e o resultado da avaliação.

Segundo Olsina [OLSINA00b], o processo de avaliação sempre se baseia em um conjunto de requisitos que o artefato deve satisfazer. Estes requisitos geralmente se originam de um conjunto de metas dadas e/ou elicítadas. Obviamente que este conjunto de metas apenas pode ser definido pelos tomadores de decisões, que são seres-humanos. Desta forma encontramos o componente subjetividade.

É verdade que algumas tarefas tendem a ser mais objetivas do que outras [OLSINA00b]. É o caso de observar se uma aplicação Web possui um índice alfabético. Podemos verificar objetivamente se o índice alfabético está ou não presente. Analogamente, é fácil verificar a quantidade de arquivos para download com problemas. Basta listar os endereços dos arquivos. Para isso são utilizados critérios meramente objetivos.

Como vimos, a interface com o usuário, por exemplo, é um componente peculiar da arquitetura da aplicação. Jones, em [JONES03], afirma que esta é uma parte do software totalmente subjetiva, pois o fato de uma tela ser usável ou não depende da opinião dos seus usuários. Algumas características somente podem ser observadas de forma subjetiva, a partir do juízo de especialistas humanos. Ainda bastante subjetiva (mas um pouco menos), podemos citar a tentativa de medir o grau de usabilidade das “tarefas de navegação do site”. Como avaliar? No entanto, este

fator subjetivo pode ceder espaço a outro menos subjetivo ainda (ou até mesmo a um objetivo) cada vez que efetuamos uma espécie de decomposição. Esta é chave da questão. Sempre que decompomos uma característica, nos distanciamos da subjetividade e nos aproximamos da objetividade. E essa decomposição pode ocorrer sucessivas vezes, de forma a buscar cada vez mais objetividade. Com essas decomposições surgem os atributos de qualidade, que são mais objetivamente medidos. Para isso devem ser utilizadas métricas de produto. O seu uso reduz a subjetividade da avaliação de qualidade ao prover uma base quantitativa para tomar decisões acerca da qualidade do software [IEEE1061]. Portanto, para se alcançar o objetivo de avaliar um artefato de forma menos subjetiva, uma das recomendações é a aplicação de métricas [FORTES02].

Deste modo, a fim de reduzir o impacto subjetivo na avaliação de qualidade, a nossa proposta procurou detalhar as características de qualidade e minimizar o universo de possibilidades ao se valorar um atributo de qualidade. Nos próximos capítulos veremos como isso foi possível.

Na Seção 2.8 veremos duas formas de avaliar a qualidade de aplicações Web, a avaliação de qualidade no produto e na fase de requisitos.

2.8 Necessidade de Avaliar para Compreender e Melhorar

Artefatos de qualidade são obtidos por meio de controles, medições e quantificações. Para isso, devemos nos preocupar tanto com a qualidade do produto quanto à do processo. Nosso objetivo é avaliar a qualidade do produto nas fases iniciais do processo de desenvolvimento (qualidade na fase de requisitos). Por isso veremos em detalhes duas dessas vertentes de qualidade, a do produto e a da fase de requisitos.

2.8.1 Avaliar a Qualidade no Produto

Um software deve estar em conformidade com os requisitos definidos pelo usuário. Isto significa dizer que tanto os requisitos funcionais [ALENCAR99] quanto os não-funcionais devem ser satisfeitos pelo produto de software. A qualidade do produto final está associada

ao nível de satisfação desses requisitos funcionais e não-funcionais. Existem algumas características fundamentais do produto de software que devem ser observadas e avaliadas. Algumas normas e modelos têm definido características e subcaracterísticas de produtos de software. Como vimos, a norma ISO [ISO9126] define seis características de qualidade. Essas características são aplicáveis a softwares em geral. Observando com mais atenção, verificamos que as aplicações para Web demandam outras características em particular. Deve ser levado em consideração que aplicações Web evoluem de forma dinâmica e a manutenção destas aplicações também precisa ser mensurável, de forma a permitir avaliar qual o grau de dificuldade ou facilidade encontrado na manutenção de aplicações Web. Como sabemos, não se pode gerenciar adequadamente o que não se pode medir, nem concentrar esforços nas partes deficientes do produto e/ou processo, sem que tenhamos métricas e programas de medição efetivos [PRESSMAN02].

A preocupação da comunidade de Engenharia de Software tem crescido em relação à avaliação de qualidade no produto, tendo em vista a dificuldade e a carência de métricas e principalmente de diretrizes que permitam avaliar de forma viável e efetiva aplicações para Web [OLSINA00b]. Observa-se também a carência de critérios para a avaliação dos artefatos/modelos intermediários no desenvolvimento dessas aplicações.

A avaliação é interessante para que erros anteriores não sejam cometidos em projetos futuros. Ao se levantar eventuais falhas na qualidade de um produto, é preciso “voltar ao passado” e tentar rastrear informações visando descobrir onde e qual erro comprometeu a qualidade de determinado artefato. Uma outra abordagem diz respeito à avaliação de qualidade ainda na fase de requisitos.

2.8.2 Avaliar a Qualidade na Fase de Requisitos

As metodologias de avaliação de qualidade geralmente operam nas fases finais do processo de desenvolvimento, quando os produtos já estão prontos. A nossa proposta antecipa esta avaliação, de forma a efetuar a avaliação na fase inicial do processo de desenvolvimento, enquanto o produto ainda não está finalizado. Como benefício temos a possibilidade de corrigir os

erros identificados durante o processo de avaliação, acarretando a geração de um produto de melhor qualidade.

Nesta dissertação desejamos assegurar mecanismos por meio dos quais possamos construir produtos que cumpram certas características e atributos a partir de um conjunto de requisitos de qualidade. Por meio da avaliação de qualidade, teremos a possibilidade de compreender o grau de cumprimento de um conjunto de características, subcaracterísticas e atributos com relação aos atributos de qualidade estabelecidos [OLSINA00b].

O objetivo da nossa avaliação é melhorar o produto, e não simplesmente apontar falhas. Baseando nosso trabalho em documentos de requisitos, as falhas serão apontadas a tempo de serem corrigidas, isto é, ainda durante as fases iniciais do desenvolvimento.

A nossa avaliação, baseada em requisitos, além de fazer parte do processo de desenvolvimento, pode estimular comparações entre opções de projetos. Ao isolarmos determinada característica de qualidade em um projeto de desenvolvimento, podemos compará-la com uma outra possível solução. Isto se dá através da comparação dos atributos de qualidade daquela característica. Esse estudo também é capaz de sugerir recomendações a fim de que certos pontos inicialmente ausentes sejam inseridos, contribuindo com a qualidade do produto final. Todo o nosso trabalho está centrado na Engenharia de Requisitos. É através dela que iremos especificar uma árvore de características, subcaracterísticas e atributos de qualidade, determinando os critérios de medição.

Apresentaremos a seguir o estado da arte da avaliação de qualidade.

2.9 Estado da Arte da Avaliação de Qualidade

Um processo de avaliação de qualidade deve ser o mais robusto possível, de maneira a produzir resultados repetíveis e reproduzíveis [OLSINA00b]. Idealmente, a avaliação de um produto por um conjunto de avaliadores deve gerar um resultado semelhante a uma outra avaliação desse mesmo produto por outro conjunto de avaliadores.

Apresentaremos agora a situação atual das estratégias de avaliação, principalmente das aplicações Web. Alguns métodos de avaliação não foram originados com o intuito de serem aplicados na Internet, mas podem ser de certa forma adaptados, por isso também falaremos um pouco destes casos.

Pesquisas com relação a este tema surgiram em meados dos anos 60 [GILB69] [GILB76] [MILLER70] [WHITE63]. Apesar do crescente surgimento de métodos de avaliação de qualidade, ainda não existe uma metodologia amplamente consolidada e utilizada na comunidade de Engenharia de Software [OLSINA00b]. Estamos propondo neste trabalho uma visão de qualidade capaz não apenas de avaliar uma aplicação, como também auxiliar na sua construção, uma vez que a avaliação deve ser feita ainda na fase de requisitos do projeto Web.

Algumas metodologias de avaliação de qualidade no campo da Hipermídia surgiram nos anos noventa [GARZOTTO93] [GARZOTTO95]. Neste contexto podemos citar um método de avaliação que se baseava na análise de artefatos de hipermídia com base nos requisitos de usabilidade. As aplicações eram avaliadas nos seguintes quesitos: conteúdo (nível de informação), estrutura (organização do conteúdo), apresentação (forma de expor o conteúdo e as funcionalidades), dinâmica (formas de navegação e interação com a aplicação) e interação (utilização de componentes dinâmicos).

Essas dimensões reforçam que para analisar sistematicamente uma aplicação de hipermídia é necessário um modelo com o qual se possa descrever a aplicação [OLSINA00b]. Outros modelos com objetivos semelhantes surgiram, alguns deles sendo bastante reconhecidos na comunidade de Hipermídia [SCHWABE95] [ISAKOWITZ95] [VASCONCELOS99] [VASCONCELOS02]. As aplicações podem ser então modeladas com o auxílio destas técnicas e uma série de critérios aplicados nestes modelos, conforme relatado em [OLSINA00b]:

- Riqueza, que representa a abundância de itens de informação e a forma de alcançá-los;

- Facilidade, que avalia a acessibilidade da informação e a facilidade de operar e compreender as aplicações;
- Consistência, que avalia se elementos similares são tratados de modo similar e elementos diferentes são tratados diferentemente;
- Auto-evidência, que representa o grau de entendimento do usuário com relação ao propósito do que a ele é apresentado;
- Previsibilidade, relacionada a facilidade de antecipar, identificar previamente o resultado de uma operação;
- Legibilidade, que representa o sentimento de validação de todo o sistema;
- Reuso, que utiliza os mesmos objetos e operações em diferentes contextos e propósitos.

A aplicação desses critérios é conveniente apenas em casos simples. Havendo a necessidade de expressar a aplicação com uma maior quantidade de fatores, esta forma de avaliação vai se tornando continuamente mais difícil, uma vez que não será fácil identificar diferenças mais sutis entre valores similares de atributos comparados [OLSINA00b].

Conforme dissemos, algumas metodologias de avaliação podem ser facilmente adaptadas para o domínio da Web. No entanto, algumas iniciativas focaram exclusivamente este tipo de aplicação, utilizando diretrizes, guias de estilo e de projeto [ROSENFELD98]. A área de qualidade ficou fortalecida com o surgimento de organizações como o *WWW Consortium* [W3C99]. Boas práticas de desenvolvimento foram divulgadas [W3C99] [THURING95], influenciando enormemente as metodologias de avaliação, uma vez que as mesmas passaram a observar se essas práticas estavam sendo utilizadas nos projetos.

A comunidade de Hipermídia e Interface do Usuário freqüentemente realiza pesquisas envolvendo usabilidade e interfaces [BEVAN97] [SCHNEIDERMAN98] [STEPHANIDIS98]. Estudos específicos foram realizados em sites de comércio eletrônico, sendo identificados alguns fatores que influenciavam as suas vendas [LOHSE98] [TILSON98]. De posse destes fatores, foi possível avaliar a usabilidade de aplicações desta

natureza. Técnicas como esta não são consideradas metodologias propriamente ditas de avaliação de qualidade.

Outra abordagem para avaliar a usabilidade das aplicações é estudar a satisfação dos usuários [KIRAKOWSKI94] [KIRAKOWSKI98]. Isso pode ser feito através da utilização de questionários (na internet ou em versão papel) que possibilita uma monitoração desta satisfação. Segundo Olsina [OLSINA00b], dentre os fatores analisados nesses questionários estão: atração, controle, eficiência, ajuda e aprendizado

Contudo, vale observar que os enfoques abordados sobre avaliação de qualidade de artefatos Web não foram centrados na medição da qualidade como uma característica de alto nível que contém e descreve outras características como usabilidade, funcionalidade, eficiência, confiabilidade, manutenibilidade e portabilidade. Geralmente se preocuparam em avaliar a usabilidade e a satisfação dos usuários com relação a um determinado perfil, deixando de ter desta forma uma visão integral de qualidade.

Uma outra linha de pesquisa enfoca na visão de qualidade como uma característica de alto nível, vide os padrões da ISO [ISO9126] e IEEE [IEEE1061]. Estes padrões, a partir dessas seis características gerais descrevem a qualidade de um software, sendo este centrado na Web ou não: “Essas características provêm uma *baseline* para posteriores refinamento e descrições da qualidade de um software” [ISO14598-5].

Toda metodologia de desenvolvimento é dividida em fases. Essas fases geram artefatos que possibilitarão a construção do produto final. Existem algumas metodologias de avaliação que procuram avaliar a qualidade desses artefatos intermediários. O nosso trabalho está intimamente relacionado com metodologias desta natureza que avaliam a qualidade da especificação de requisitos. Neste contexto podemos mencionar o AGORA [KAIYA02], uma metodologia de avaliação de qualidade baseada em objetivos que avalia a qualidade da especificação de requisitos em função da corretude, não ambiguidade, completude, consistência, verificabilidade, modificabilidade, rastreabilidade, grau de importância e estabilidade.

Algumas metodologias de avaliação de qualidade procuram mensurar a qualidade de aplicações Web focando determinado tipo de requisito. Existem, no entanto, iniciativas que atuam na fase operativa levando em consideração tanto a visão dos requisitos funcionais quanto a dos não-funcionais. Neste sentido, a WebFP_QEM (*Web Function Points and Quality Evaluation Method*) [ABRAHÃO02] possibilita a captura dos objetivos da aplicação e do negócio levando em consideração diversos *stakeholders*, especifica modelos navegacionais e informacionais (requisitos funcionais), especifica a qualidade dos modelos e critérios de medição (requisitos não-funcionais), e analisa todos os resultados a fim de contribuir com recomendações e adicionalmente com relatórios de custos. Após a avaliação, algumas atividades relativas à reestruturação da aplicação podem ser executadas. Antes, porém é preciso analisar o impacto destas mudanças. Esta análise deve considerar alguns aspectos das modificações:

- tipo: corretiva, adaptativa, preventiva;
- escopo: tamanho, custo, duração;
- conseqüências: impacto na estrutura navegacional, performance, usabilidade etc.

Várias ferramentas têm sido desenvolvidas de modo a facilitar e automatizar algumas fases de metodologias de avaliações de qualidade [IVORY01] [OLSINA01a] [OLSINA01b]. Esta é uma forma de incentivar o uso desses métodos, tendo em vista que reduzem significativamente o esforço para aplicá-los.

Outro trabalho bastante promissor foi a definição da metodologia (*Web-site QEM*), para a avaliação quantitativa de sites Web, na sua fase operativa [OLSINA00b]. Com base nesta metodologia criamos a REQE (Requirements Engineering Quality Evaluation), que será detalhada no capítulo 4.

2.9.1 Web-site QEM

Apresentaremos agora um panorama com as principais fases, processos e modelos da metodologia *Web-site QEM* [OLSINA00b], sobre o qual se apóia esse trabalho.

O enfoque proposto é essencialmente integral, flexível e robusto, e cobre a maior parte das atividades do processo de avaliação, comparação e seleção de artefatos Web. Essa estratégia denominada *Metodologia de Avaliação de Qualidade de Sites Web* (ou em inglês, *Web-site Quality Evaluation Method*) pretende realizar avaliações sistemáticas, disciplinadas e quantitativas sobre a qualidade de sistemas de informação centrados na Web. O grande desafio é medir, avaliar e potencialmente melhorar a qualidade do sites Web.

Uma estratégia efetiva para encarar esses desafios é o modelo usando enfoques descritivos [OLSINA98]. Particularmente, o modelo de produto nos permite compreender, analisar, otimizar e prever a qualidade dos mesmos. É nessa perspectiva que se pretende utilizar a metodologia em questão. As etapas do estudo estão focadas na fase operativa, ou seja, na avaliação de artefatos Web já existentes ou operativos.

Uma das metas principais da avaliação e comparação da qualidade de artefatos Web consiste em compreender o grau de cumprimento de um conjunto de características e subcaracterísticas com respeito aos requisitos de qualidade estabelecidos. Neste contexto, uma tarefa importante é a definição da árvore de características e atributos de qualidade quantificáveis considerando o domínio das aplicações Web.

Uma fase é uma agregação de processos fortemente relacionados ou coesivos, relacionados em certa ordem. As distintas fases podem exibir comportamentos diferentes. Geralmente, todo modelo de processo para construção de um produto de software possui ou engloba pelo menos três fases: a de exploração, a de desenvolvimento e a operativa (ou manutenção) [PRESSMAN02]. Essa metodologia claramente põe o foco na fase operativa de um sistema de informação centrado na Web.

Vejamos a seguir as fases da metodologia *Web-site QEM*:

- Planificação e Programação da Avaliação de Qualidade;

- Definição e Especificação dos Requisitos de Qualidade;
- Definição e Implementação da Avaliação Elementar;
- Definição e Implementação da Avaliação Global;
- Análises dos Resultados, Conclusão e Documentação;
- Validação de Métricas.

A fase de *Planificação e Programação da Avaliação de Qualidade* contém atividades e procedimentos de suporte, com o fim de determinar objetivos estratégicos, táticos e operacionais. Isto é, permite estabelecer as principais estratégias e metas do processo em um contexto organizacional; permite selecionar um modelo de processo de avaliação, descrever métodos, agentes e recursos para as atividades; e permite planejar certas atividades mesmo após o início do processo de avaliação.

A fase de *Definição e Especificação dos Requisitos de Qualidade* trata de atividades e modelos para a elicitación, determinação, análise e especificação dos requisitos. Partindo de um processo de medição, tem-se o objetivo de avaliar, comparar, analisar e melhorar as características e os atributos de qualidade de artefatos Web. O processo de determinação dos requisitos, realizado em uma mescla de estratégias prescritivas e descritivas, culmina em um documento que hierarquicamente especifica todas as características e atributos quantificáveis que modelam a qualidade segundo as necessidades do usuário.

A fase de *Definição e Implementação da Avaliação Elementar* trata de atividades, modelos, técnicas e ferramentas para determinar métricas e critérios de avaliação para cada atributo quantificável. São considerados todos os tipos de critérios elementares, escalas, preferências, valores críticos, e funcionais para determinar o grau de comprometimento daquele atributo, entre outros assuntos. Uma vez definidos os critérios para medir cada atributo, deve-se efetuar o processo de recolhimento dos dados, computar as métricas e preferências elementares e documentar os resultados.

Na fase de *Definição e Implementação da Avaliação Global* são tratadas as atividades, modelos e ferramentas para determinar os critérios de agregação das preferências de

qualidade elementar, a fim de produzir a preferência global, para cada sistema selecionado. São considerados tipos de funções de agregação para modelar diferentes relações entre atributos e características, a saber: simultaneidade, neutralidade e diferentes níveis de polarização “AND/OR”. Uma vez definidos os critérios, a próxima preocupação é o processo de cálculo e ordenação (*ranking*).

Já a fase de *Análise de Resultados, Conclusões e Documentação* trata de atividades de análise e comparação das preferências de qualidade elementares, parciais e globais, além da justificativa desses resultados. Também são utilizados ferramentas e mecanismos de documentação para facilitar a interpretação dos dados e seu acompanhamento.

Por último temos a *Validação de Métricas*, que é um processo importante na disciplina de avaliação de software. Podemos definí-la como o processo responsável por assegurar que as medidas são uma caracterização numérica do atributo, mostrando que a condição de representação dos atributos em valores numéricos é satisfeita [FENTON97]. Isto é, que a correspondência entre o domínio empírico e o novo domínio numérico preserva a relação funcional.

Dentro das fases, a metodologia *Web-site QEM* está centrada em processos, que são apresentados em detalhe na tese de doutorado de Olsina [OLSINA00b]. Como o nosso trabalho está apoiado na referida metodologia, explicaremos resumidamente cada um desses processos, a saber:

- Definindo o Domínio para a Avaliação de Qualidade;
- Definindo Metas de Avaliação e Selecionando o Perfil do Usuário;
- Especificando Requisitos de Qualidade para Artefatos Web;
- Definindo Critérios Elementares e Implementando Procedimentos de Medição (também chamado de Determinação da Preferência de Qualidade Elementar);
- Definindo as Estruturas de Agregação e Implementando a Avaliação de Qualidade Global;
- Analisando e Comparando os Resultados Parciais e Globais.

No processo *Definindo o Domínio para a Avaliação de Qualidade* os tomadores de decisão devem decidir qual é o domínio da aplicação a avaliar e definir qual o ente a ser avaliado. Três entes gerais podem intervir no processo de avaliação: processo, produtos e recursos. No caso de produtos, por exemplo, para realizar um estudo com o objetivo de valorar, comparar e determinar o estado da arte da qualidade sobre características específicas como usabilidade e funcionalidade, deve-se selecionar dentro de um domínio específico um conjunto de sites típicos. Por outro lado, como as métricas e os critérios utilizados são absolutos, pode-se avaliar atributos e características de apenas um artefato.

No passo *Definindo Metas de Avaliação e Selecionando o Perfil do Usuário*, os avaliadores devem definir e refinar as metas e o alcance do processo de avaliação. Podem avaliar um conjunto de características de parte de um sistema, ou de um sistema completo, ou comparar características de qualidade global dos sistemas. Os resultados podem ser utilizados para compreender, melhorar ou controlar a qualidade dos artefatos Web. Por outro lado, a formulação das metas, e conseqüentemente, a relativa importância das características e atributos de qualidade variam conforme o perfil do usuário selecionado, entre outros fatores. Em termos gerais, e com base no padrão ISO [ISO9126], são considerados três perfis de usuários, em um alto nível de abstração para os domínios Web: visitantes, desenvolvedores, e gerentes. Seguindo um mecanismo de decomposição podemos dividir a categoria de visitantes em classes mais específicas.

Na tarefa *Especificando Requisitos de Qualidade para artefatos Web*, os avaliadores devem especificar as características, subcaracterísticas e atributos de qualidade e agrupá-los em uma árvore. Com relação às características de qualidade de mais alto nível, segue-se a mesma classificação conceitual prescrita no padrão ISO [ISO9126], praticamente idêntica à classificação dada pela IEEE [IEEE1061]. Estas características de alto nível são: usabilidade, funcionalidade, confiabilidade, eficiência, portabilidade e manutenibilidade. Mais uma vez é importante destacar que se pode quantificar a qualidade de um produto de software, por uma apropriada agregação e quantificação de algumas ou todas as características, a partir da medição direta e indireta de atributos. Tudo isso é possível

porque a partir das características se derivam as subcaracterísticas, e a partir destas, seguindo um processo de decomposição recursivo, podem ser especificados os atributos de qualidade. Neste passo, técnicas de elicitação de requisitos como questionário e workshops devem ser utilizadas. Finalmente, considerando o domínio, as metas, e o perfil do usuário (suas necessidades e comportamentos), deve-se especificar a árvore de requisitos de qualidade.

Durante o processo *Definindo Critérios Elementares e Implementando Procedimentos de Medição*, os avaliadores devem definir uma base de critérios para a avaliação elementar e realizar o processo de medição, e pontuação elementar. Um critério de avaliação elementar declara e especifica como medir atributos quantificáveis. O resultado final é uma preferência ou um *ranking* elementar, o qual pode ser interpretado como o grau ou porcentagem dos requisitos elementares satisfeitos. Para cada variável medida X_i , $i = 1, \dots, n$ se define uma função que representa o critério elementar. Esta função é uma correspondência (mapeamento) dos valores computados a partir do domínio empírico no novo domínio numérico, e foi denominada de preferência de qualidade elementar (IE_i). Pode-se entender IE_i como a porcentagem do requisito satisfeito para A_i . Neste sentido, $IE_i = 0\%$ denota uma situação totalmente insatisfatória, ao ponto que $IE_i = 100\%$ representa o outro extremo, sendo a situação totalmente satisfatória. Assim a pontuação elementar cai em um dos três níveis de aceitabilidade, isto é, *insatisfatório* (de 0 a 40%), *marginal* (de 40 a 60%), e *satisfatório* (de 60 a 100%) listados pelo padrão IEEE [IEEE1061]. Para facilitar o entendimento deste processo, imagine, por exemplo, um atributo de qualidade *Suporte a outros idiomas*. Este pode ser medido por meio de um critério elementar multi-variável, contínuo e absoluto. Podemos adotar os seguintes parâmetros:

- $N_a \Rightarrow$ Quantidade de idiomas estrangeiros para o nível de suporte S_i ;
- $S_a \Rightarrow$ Nível de suporte, a saber: *total* (em todo o site); *parcial* (em alguns sub-sites do site); *mínimo* (em algumas página ou documentos). S_a onde $j = (1 \dots n)$, para $n=3$
 - $S_1 = 0,2$ (suporte mínimo);
 - $S_2 = 1$ (suporte parcial);
 - $S_3 = 2$ (suporte total);

- $X = 30 * \sum (S_a * N_a)$;
- $IE_i = X$ (Se $X > 100 \Rightarrow IE_i = 100\%$).

Vejamos alguns cálculos de preferências elementares:

- Site A: ($N_1 = 0$; $N_2 = 0$; $N_3 = 2$ e $S_3 = 2$) $\Rightarrow IE_i = 100\%$
- Site B: ($N_1 = 0$; $N_2 = 1$; $N_3 = 1$ e $S_2 = 1$; $S_3 = 2$) $\Rightarrow IE_i = 90\%$
- Site C: ($N_1 = 0$; $N_2 = 2$; $N_3 = 0$ e $S_2 = 1$) $\Rightarrow IE_i = 60\%$
- Site D: ($N_1 = 5$; $N_2 = 2$; $N_3 = 0$ e $S_1 = 0,2$; $S_2 = 1$) $\Rightarrow IE_i = 90\%$

Note que no caso da avaliação estar sendo feita em apenas um site, seria necessário calcular o critério de preferência de *Suporte a outros idiomas* uma única vez. De forma análoga, ocorreria com os outros atributos de qualidade.

No processo *Definindo as Estruturas de Agregação e Implementando a Qualidade Global*, os avaliadores devem estabelecer estruturas de agregação de preferências elementares para produzir a preferência da qualidade global. Posteriormente, deve-se implementar o processo de avaliação de modo a obter um indicador de qualidade global para cada sistema avaliado. Para n atributos, a correspondente função produz n preferências elementares. No entanto, aplicando um mecanismo de agregação passo a passo, as preferências elementares vão se agrupando convenientemente, produzindo o resultado global. A preferência de qualidade global representa o grau de satisfação de todos os requisitos explícitos e implícitos. Neste ponto pode ser usada uma série de modelos, entre eles o *Logic Scoring of Preference* (LSP) fundamentado na lógica de preferência contínua [ISO14598-5] [DUJMOVIC96].

No passo *Analizando e Comparando os Resultados Parciais e Globais*, os avaliadores analisam, avaliam e comparam resultados parciais e globais considerando as metas e a visão do usuário estabelecida. Este processo culmina com a documentação das conclusões e recomendações. Deste modo, o processo de avaliação, por meio da metodologia *Web-site QEM*, produz informação elementar, parcial e global.

Exemplos da utilização da metodologia *Web-site QEM* podem ser obtidos em [OLSINA99] e [OLSINA00a].

Neste capítulo estudamos os tipos de aplicações Web em geral e identificamos algumas características que as diferenciam dos sistemas tradicionais. Apresentamos algumas classificações capazes de agrupar os diferentes tipos de aplicações Web. Destacamos a interface com o usuário como um ramo bastante importante no domínio Web como um todo.

Discutimos também as dificuldades inerentes à atividade de avaliar a qualidade de aplicações Web, bem como a presença do fator subjetividade em maior ou menor grau nas metodologias de avaliação. Apresentamos o estado da arte na avaliação de qualidade e identificamos que o foco dessas avaliações se concentra em medir a qualidade dos produtos nas fases finais do processo de desenvolvimento. Abordamos com mais detalhes a metodologia *Web-site QEM*, com suas fases e seus processos. Antes de apresentarmos os conceitos da metodologia proposta por esta dissertação precisamos falar sobre requisitos.

3 Requisitos

Neste capítulo mostramos alguns tipos de requisitos dando ênfase aos não-funcionais. Apresentamos também os conceitos básicos da Engenharia do Domínio, que serve de apoio para a metodologia proposta por esta dissertação. Por fim propomos um template para um documento de requisitos.

3.1 Introdução

O estabelecimento dos requisitos é considerado parte essencial para o sucesso de um projeto. Falhas no seu levantamento levam a atrasos no cronograma de projetos e estouros nos seus custos. Quando os requisitos de um sistema são mal entendidos, possivelmente o produto final não será livre de falhas e a versão do sistema entregue ao cliente não corresponderá ao produto encomendado inicialmente. É neste sentido que estamos propondo uma avaliação de qualidade ainda na fase de requisitos. E por isso, temos que estudar esta área tão vasta e importante na construção de softwares de qualidade.

Na Seção 3.2 veremos algumas classificações de requisitos existentes na literatura. Na Seção 3.3 apresentaremos os conceitos básicos da Engenharia de Domínio, uma área de fundamental importância para a avaliação de qualidade proposta por esta dissertação. Em seguida (Seção 3.4) damos ênfase aos requisitos não-funcionais, destacando os RNFs para Web (Seção 3.4.1) e algumas abordagens para modelá-los (Seção 3.4.2). Na Seção 3.4.3 detalhamos cada um dos requisitos não-funcional através de modelos de qualidade. Por fim, falamos sobre o documento de requisitos (Seção 3.5) e propomos uma estrutura para este documento (Seção 3.5.1).

3.2 Classificação dos Requisitos

Numa visão mais tradicional, os requisitos são classificados como requisitos funcionais e não-funcionais, conforme apresentado por Sommerville [SOMMERVILE03]:

- **Requisitos funcionais:** são as declarações das funções que o sistema deve oferecer, como ele se comporta com entradas particulares e como o sistema deve se comportar em situações específicas. O termo “função” é usado no sentido genérico da operação que pode ser realizada pelo sistema, seja por meio de comandos dos usuários, seja pela ocorrência de eventos internos ou externos ao sistema. Em alguns casos, os requisitos funcionais podem também, explicitamente, definir o que o sistema não deve fazer;

- Requisitos não-funcionais (RNFs): são as restrições nas funções oferecidas pelo sistema. Incluem restrições de tempo; restrições no processo de desenvolvimento; padrões; e de qualidades globais de um software, como custo, performance, confiabilidade, manutenibilidade, portabilidade, usabilidade, desempenho, dentre outras.

De forma geral, a diferença entre requisitos funcionais e não-funcionais está no fato de os primeiros descreverem “o que” o sistema deve fazer, enquanto que os outros fixam restrições sobre “como” os requisitos funcionais serão implementados [DAVIS93].

Diversos estudiosos afirmam que a classificação tradicional não mais apresenta respostas à compreensão de todas as facetas que compõem o ambiente das atuais organizações. A preocupação com a observação de aspectos relativos ao ambiente organizacional já vem se manifestando desde o início dos anos 80 [LEITE83] [DOBSON94a] [DOBSON94b]. Uma melhor compreensão da organização requer analisar não somente requisitos funcionais, mas também os aspectos organizacionais [YU98] [CYSNEIROS99] [MYLOPOULOS01] e sociais [SANTOS98] [SUTCLIFFE98].

Por essa razão, a classificação mais moderna de requisitos agrega a conceituação de requisitos organizacionais [LOUCOPOULOS95] [KOTONYA97]:

- Requisitos organizacionais: dizem respeito às metas da empresa, suas políticas estratégicas adotadas, os relacionamentos entre os seus atores junto com seus respectivos objetivos.

A inserção dos requisitos organizacionais amplia a visão tradicional que aborda apenas aspectos funcionais e não-funcionais. Os requisitos na visão moderna não se preocupam apenas com "como" o sistema deve fazer, mas sim com "o que" o sistema deve fazer, associado com "o porque" fazer, compreendendo a intencionalidade dos fatos organizacionais [YU94].

No enfoque dos requisitos organizacionais, estudam-se os objetivos do próprio sistema e o relacionamento desses com os objetivos da organização, bem como outras propriedades dos sistemas, como as relacionadas com o desempenho, segurança e restrições. Também são

abordados problemas e aspectos gerenciais, organizacionais, econômicos, técnicos, sociais e ambientais.

Para Dobson [DOBSON94b], requisitos organizacionais são aqueles que resultam do sistema pretendido, considerando-se o contexto social no qual ele se encontra inserido. Por exemplo, obrigações e responsabilidades, controle e autonomia, valores e éticas que são, normalmente, embutidos na estrutura e na política da empresa, de forma que não são diretamente observados ou facilmente articulados.

Em literatura recente, há distinção entre requisitos iniciais (*early requirements*) e requisitos finais (*late requirements*) [YU97], conforme conceituado a seguir:

- Requisitos iniciais: caracterizados por atividades que tratam de questões organizacionais e não-funcionais, com ênfase na compreensão das motivações e das razões que estão por trás dos requisitos do sistema;
- Requisitos finais: caracterizados por atividades que tratam de questões como completude, consistência e verificação dos requisitos, dando atenção ao refinamento, eliminação de ambigüidades e completude da descrição da alternativa escolhida.

É objetivo da Engenharia de Requisitos sistematizar o processo de definição de todos estes requisitos de modo que seja gerado um compromisso entre os desenvolvedores e usuários/clientes, para que todos participem da geração e definição dos requisitos de um sistema. Existem diversos métodos [KAPLAN00], [SÁNCHEZ02], técnicas e ferramentas que ajudam os engenheiros de requisitos na obtenção de requisitos mais seguros e completos, permitindo compreender o problema, facilitar a obtenção das necessidades do usuário/cliente, validá-las com o usuário/cliente e descrevê-las em especificações de requisitos.

Abordaremos em detalhe nesta dissertação os requisitos não-funcionais, uma vez que estes são os que têm maior relação com o nosso trabalho. Antes porém, introduziremos a

Engenharia do Domínio, que de alguma forma facilita a obtenção dos requisitos de um sistema, além de servir de base para a avaliação de qualidade proposta neste trabalho.

3.3 Engenharia do Domínio

A tarefa de avaliar a qualidade de qualquer ente que seja não é simples. Muitas vezes os resultados são questionados, pois os critérios de avaliação não são aceitos universalmente. Esta foi uma dificuldade encontrada neste trabalho. Por exemplo, ao avaliar a qualidade de sistemas no domínio de aplicações bancárias, o que deveria ser levado em consideração? Uma aplicação “mal” avaliada poderia levar os responsáveis pelo sistema a questionar as características e os atributos de qualidade selecionados para qualificar uma aplicação dessa natureza. A saída que encontramos foi verificar como estes problemas são abordados na Engenharia do Domínio. Utilizando técnicas da Engenharia do Domínio [KANG90] [SIMOS96], a responsabilidade sobre a escolha do que é ou não importante para um determinado domínio de aplicação (principalmente na questão das funcionalidades) de certa forma não recairia sobre a nossa metodologia, que se preocuparia com a avaliação de qualidade a partir de características definidas pela Engenharia do Domínio.

O surgimento dessa disciplina na comunidade de Engenharia de Software se deu a partir do elevado custo de manutenção de softwares. Isso ocorria por causa da introdução de mudanças arbitrárias [ARANGO94a], muitas vezes que não diziam respeito ao domínio da aplicação em questão. Neste contexto surgiram iniciativas para investigar a construção de software baseado na reutilização de domínios [LEITE93a]. Este paradigma de desenvolvimento baseia-se nas experiências iniciais de utilização de um gerador de sistemas baseado em domínios, chamado Draco [NEIGHBORS84]. A motivação original para o seu desenvolvimento foi o de prover uma forma de construção de sistemas de software a partir de componentes preexistentes, ao contrário de construir o sistema através de métodos tradicionais. Um outro ponto de vista é o de construir linguagens de especificação de alto nível para serem utilizadas na construção de sistemas, e por mecanismo de transformações produzirem os programas em linguagens executáveis [SANT'ANNA98].

Neighbors [NEIGHBORS81] definiu o termo *Análise de Domínio* como sendo “Uma tentativa de identificar os objetos, operações e relações entre o que peritos em um determinado domínio percebem como importante”. De fato, Arango [ARANGO94b] fez uma analogia entre os processos da Engenharia de Software e a Engenharia de Domínio, que é considerada uma Engenharia de Software atuando em um meta-nível, ou seja, ao invés de possuir atividades e procedimentos e gerar artefatos necessários à construção de aplicações específicas, os objetos de estudo são famílias de aplicações.

Comparativamente podemos dizer que a *Análise de Requisitos* está para a *Engenharia de Software* como a *Análise de Domínio* está para a *Engenharia de Domínio*. A *Especificação do Sistema* na *Engenharia de Software* está relacionada com a *Especificação da Infra-estrutura* na *Engenharia do Domínio*. Enquanto o *Projeto* e a *Implementação* abordada na *Engenharia de Software* estão relacionados com a *Implementação da Infra-estrutura* da *Engenharia de Domínio*. A ligação entre duas disciplinas não se dá apenas através de uma analogia, existe realmente uma íntima relação, que pode ser vista na Figura 3.1 [GUIZZARDI00].

Veja que a Engenharia de Software poderá utilizar a Engenharia de Domínio. A Engenharia do Domínio realiza o processo de conversão do conhecimento de um determinado domínio em um repositório de componentes, e estes são utilizados pela Engenharia de Software de forma a contemplar o que há de importante no contexto daquele domínio [GUIZZARDI00]. Desta forma, o *Projeto de Domínio* pode ser visto como uma seqüência de duas atividades da Infra-estrutura do domínio: *Especificação* e *Projeto*. Assim é possível modelar o que é importante no contexto do domínio da aplicação. É preciso refinar este modelo com o objetivo de agregar as funcionalidades particulares de cada sistema. Isso é feito no *Projeto da Aplicação Específica*, que engloba as atividades de *Especificação do Sistema* e *Projeto e Implementação*.

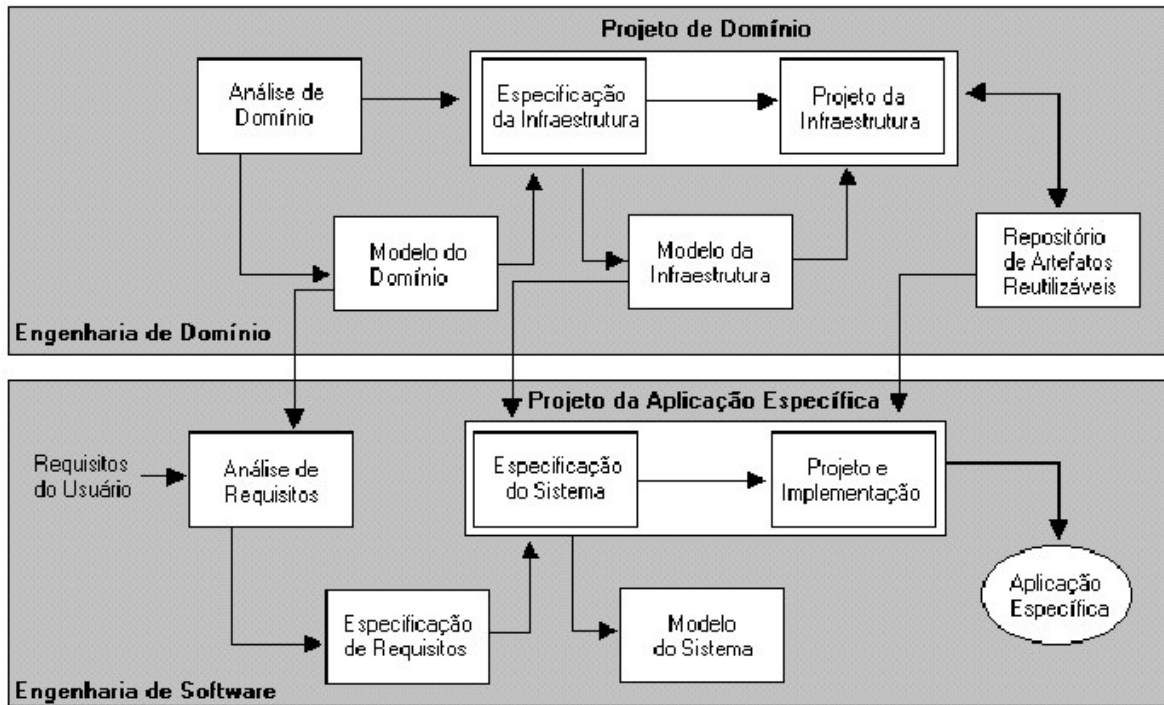


Figura 3.1 – Engenharia do Domínio x Engenharia de Software

O objetivo principal da análise de domínio é identificar, refinar, organizar e apresentar conceitos e o conhecimento sobre algum domínio específico [GUIZZARDI00]. Esta análise é feita em fases, que vai desde a elaboração do modelo do domínio até a especificação da infra-estrutura. As características comuns das aplicações pertencentes a uma determinada família devem ser especificadas de forma abstrata. Deste modo, as aplicações utilizam essas abstrações e realizam adaptações e refinamentos, dependendo das necessidades específicas dos *stakeholders*.

É importante ressaltar que o modelo de domínio não apenas será útil para a elaboração da árvore de características, subcaracterísticas e atributos de qualidade, mas também importante para a fase de análise da aplicação específica, ajudando na compreensão e na comunicação acerca do domínio e, conseqüentemente, contribuindo fortemente para a elicitação dos requisitos.

Note que *Análise de Domínio* pode ser considerado um tipo abstrato de *Análise de Requisitos*, onde os requisitos não dizem respeito a uma aplicação específica, mas são originados das necessidades implícitas e explícitas inerentes à família de aplicações que aqueles sistemas estão inseridos [NEIGHBORS81].

A seguir apresentamos ao menos alguns termos e conceitos básicos utilizados na análise de domínio.

- *Domínio do Problema* ou *Domínio da Aplicação*: expressam a semântica existente em domínios do mundo real, ou classes de problemas. Um de seus objetivos é descrever o mundo real no qual os sistemas de software estão inseridos. Englobam duas perspectivas:
 - Domínios estáveis: têm seus objetivos e suas fronteiras bem definidas de forma a prover uma base de conhecimento suficientemente capaz de produzir soluções;
 - Domínios instáveis: ainda estão sendo investigados e suas características necessitam ser melhor identificadas e isoladas. Grupos diferentes podem ter entendimentos diversos com relação ao mesmo domínio, uma vez que não existe uma espécie de consenso sobre os mesmos.
- *Modelo do Domínio*: é o produto da análise feita sobre o *Domínio do Problema* ou *da Aplicação*. São criados para facilitar uma possível implementação desses domínios, diminuindo a distância existente entre estes e os *Domínios Executáveis*. Os *Domínios da Aplicação* são submetidos a mecanismos transformacionais para atingir código executável, e os modelos do domínio funcionam como etapas intermediárias neste processo. Expressam similaridades e regularidades de aplicações em um determinado domínio, formando uma base comum para as aplicações daquele domínio.

- *Domínios Executáveis*: podem ser diretamente implementados em uma plataforma computacional. São aqueles para quais existe um compilador ou interpretador fora do ambiente Draco [NEIGHBORS84].

No contexto das transformações aplicadas sobre um programa fonte com o objetivo de gerar um novo programa, temos dois tipos:

- Transformações horizontais: ocorrem intra-domínio, quando o resultado da transformação pertence ao mesmo domínio do programa original [LEITE94];
- Transformações verticais: conhecidas como inter-domínios ou refinamentos, quando o resultado da transformação pertence a um novo domínio. Há um mapeamento das estruturas sintáticas de um domínio em estruturas equivalentes em outro domínio [LEITE94].

Conforme consta em [GUIZZARDI00], o estudo de vários métodos de análise de domínio permite especificar um processo composto por três fases:

1. Planejamento: devem ser feitas a análise do negócio e a análise de risco, a fim de identificar a viabilidade da análise do domínio. É preciso verificar a “maturidade” do domínio, para saber se é possível efetuar sua análise, com base nos custos envolvidos. As fronteiras do domínio devem estar bem definidas, para que haja uma definição coerente do seu escopo. Ao final desta fase teremos uma especificação dos requisitos do domínio.
2. Aquisição e seleção dos dados: nesta fase são identificadas as fontes disponíveis para explorar, definir e modelar o domínio. Uma das abordagens utiliza técnicas de engenharia reversa auxiliadas por documentos, código fonte (conceitos de programação) e outros experimentos práticos (conceitos humanos). Não devem ser desprezados livros, revistas científicas, *papers*, registros históricos, entrevistas com especialistas no domínio etc. Cornwell [CORNWELL96] sugeriu que fossem analisados pelo menos três sistemas inseridos no contexto do domínio, com vistas a

um possível reuso do conhecimento modelado. Peritos e especialistas no domínio são considerados as fontes mais ricas de informação, mas são de certa forma recursos escassos e caros. A literatura técnica não deve ser utilizada de forma isolada, evitando situações em que determinados conceitos e descrições estão fora do contexto. As fontes de informações devem complementar-se umas às outras.

3. Análise dos dados e modelagem do domínio: é considerada uma espécie de refinamento e validação das fases anteriores, onde os conceitos agregados são avaliados com relação à consistência, correção e completude.

Outras técnicas usuais nesta etapa são modelos de entidades e relacionamentos, modelos de objetos, pré-condições e invariantes. No entanto, é importante ressaltar que o modelo produzido não deve refletir decisões de implementação. Esta etapa finaliza com a definição da hierarquia, abstração e classificação das entidades. Esta atividade é baseada em fatos como interdependência ou exclusão mútua das entidades selecionadas [GUIZZARDI00].

Além do apoio da Engenharia de Domínio na especificação de árvores de características, subcaracterísticas e atributos de qualidade dentro de um determinado domínio de aplicação, temos também que ter em mente o desenvolvimento baseado no reuso [FREITAS97] [BERGMANN00]. Isso ocorre porque como já mencionamos acima, a Engenharia de Domínio ocorre em um meta-nível, capaz de desenvolver infra-estruturas reutilizáveis que capturam a estrutura e o conhecimento de uma família de aplicações. Uma vez construídas essas infra-estruturas podem ser reutilizadas por atividades de um processo de Engenharia de Software, promovendo assim o desenvolvimento com reuso.

Voltando aos requisitos, estudaremos agora em detalhes, os requisitos não-funcionais.

3.4 Requisitos Não-Funcionais

Primeiramente abordaremos os requisitos não-funcionais de uma forma geral. Em seguida, na Seção 3.4.1 falaremos especificamente dos requisitos não-funcionais para Web. Na Seção 3.4.2 apresentaremos algumas abordagens para modelar RNFs, e em maiores detalhes o *NFR Framework* (Seção 3.4.2.1). A Seção 3.4.3 detalha cada uma das características de qualidade presentes nos padrões ISO [ISO9126] e IEEE [IEEE1061].

Requisitos não-funcionais (RNFs) merecem especial atenção no projeto e desenvolvimento de sistemas. Erros ou omissões nos RNFs são em alguns casos, os mais caros e difíceis de corrigir, pois acarretam grandes mudanças nos orçamentos e cronogramas [DAVIS93]. Consequências vão desde a insatisfação do cliente, até projetos completamente fracassados.

Há diversas definições para RNFs, entre elas: restrições ou limitações de um sistema [BOEHM76], qualidade dos atributos [ROMAN85], qualidade que o software deve exibir, ou requisitos não comportamentais [DAVIS93] em contraste com os comportamentais, que são os requisitos funcionais. De acordo com IEEE [IEEE830], RNFs podem ser considerados em termos de performance, interfaces externas, limitações de projeto, ou qualidade dos atributos. Os requisitos de performance são caracterizados pela interação com os usuários. Os requisitos de interfaces externas estão relacionados com as interfaces propriamente ditas. Requisitos de projetos referem-se às várias limitações que são impostas no projeto dos sistemas, como padrões externos e internos. Finalmente, atributos de qualidade são os requisitos concentrados nas características do sistema, como segurança, manutenibilidade, robustez, etc.

A IEEE [IEEE830] listou 13 RNFs, que devem ser incluídos na seção de requisitos específicos no documento de requisitos. Foram eles:

1. Requisitos de performance;
2. Requisitos de interface;
3. Requisitos operacionais;
4. Requisitos de recursos;

5. Requisitos de verificação;
6. Requisitos de aceitação;
7. Requisitos de documentação;
8. Requisitos de segurança;
9. Requisitos de portabilidade;
10. Requisitos de qualidade;
11. Requisitos integridade;
12. Requisitos manutenibilidade;
13. Requisitos de confiança.

A fim de facilitar o entendimento dos RNFs, Loucopoulos [LOUCOPOULOS95] propôs uma associação entre as necessidades dos usuários (entre parênteses) e os RNFs, expressa na Figura 3.2.

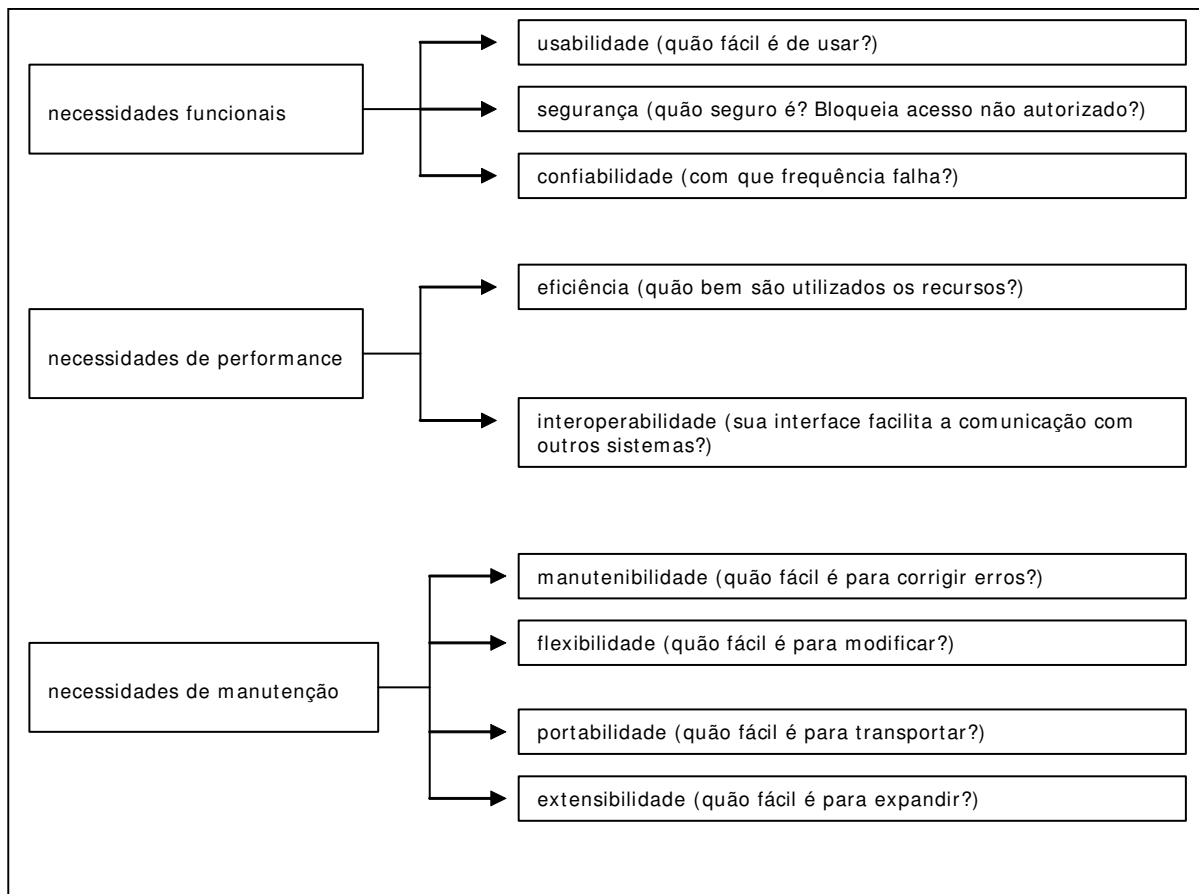


Figura 3.2 – Necessidades dos usuários x RNFs

Já Roman [ROMAN85] identificou seis classes de RNFs:

- r1: limitações de interface;
- r2: limitações de performance;
- r3: limitações operacionais;
- r4: limitações do ciclo de vida;
- r5: limitações econômicas;
- r6: limitações políticas.

Por sua vez Yeh [YEH90] classificou de outra forma, em apenas três classes:

- n1: limitações no objetivo do sistema;
- n2: limitações no desenvolvimento, evolução e manutenção do sistema;
- n3: limitações econômicas do sistema;

É possível encontrar muitas outras classificações que variam devido aos diferentes pontos de vista dos autores. No entanto, existe uma classificação geral bastante aceita proposta por Sommerville [SOMMERVILLE03]:

- Requisitos do processo: fazem menção às limitações do processo de desenvolvimento do sistema. Isso inclui requisitos de um método de desenvolvimento particular, padrões a serem seguidos, ferramentas CASE a serem utilizadas, relatórios de gerenciamento que devem ser gerados, etc. Podem causar limitações reais no projeto de um sistema. Por exemplo, uma organização pode especificar que um determinado conjunto de ferramentas CASE deve ser usado no desenvolvimento do sistema porque os funcionários possuem experiência com a mesma e não há recursos para treinamento. Caso essa ferramenta não possua suporte a desenvolvimento orientado a objeto, isso significa dizer que a arquitetura do sistema não pode ser orientada a objeto;

- Requisitos do produto: especificam as características desejáveis que um sistema deve possuir. Algumas podem ser facilmente medidas ou quantificadas, ao passo que outras são mais subjetivas;
- Requisitos externos: são limitações impostas tanto ao produto quanto ao processo, e originam-se do ambiente no qual o sistema está sendo desenvolvido, seja interno ou externo à empresa. Políticas da empresa a serem seguidas, considerações organizacionais, necessidade de integração com outros sistemas, leis (naturais ou legislativas), etc.

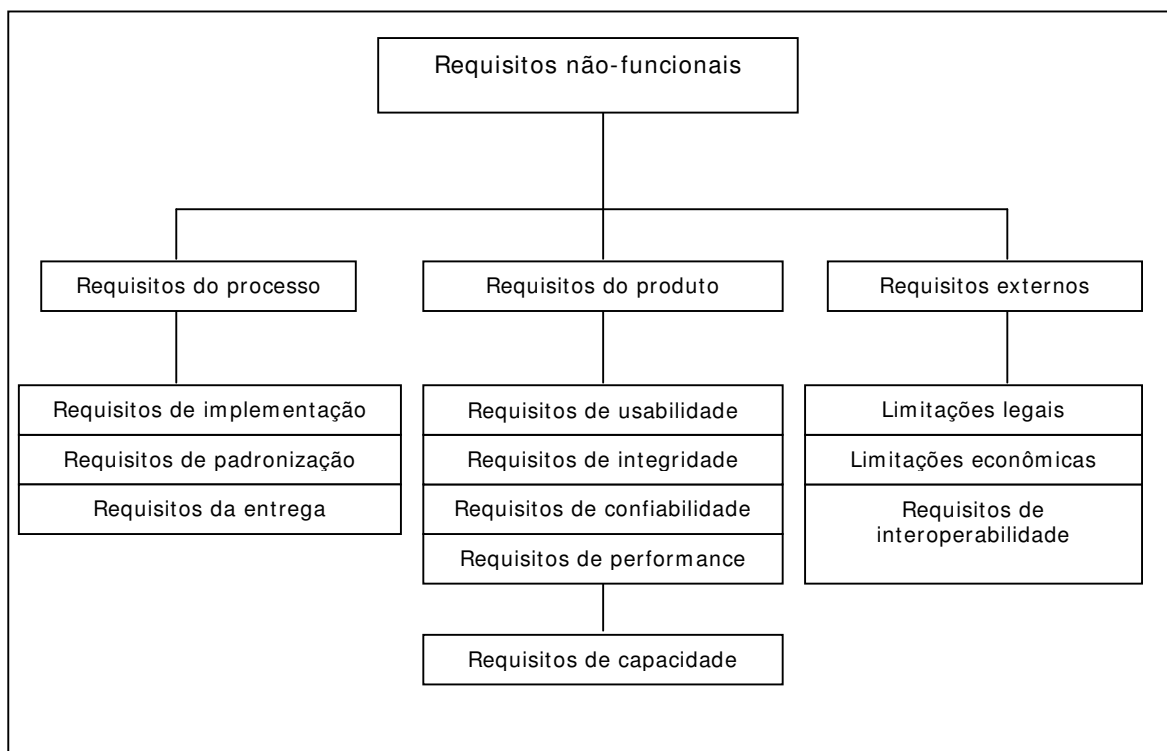


Figura 3.3 – Classificação dos RNFs segundo Sommerville

Cysneiros, em [CYSNEIROS01], propõe uma taxonomia que classifica os RNFs em primários e específicos, conforme apresentado na Figura 3.4. RNFs primários são aqueles mais abstratos que representam propriedades como: *Confiabilidade*, *Rastreabilidade*, *Precisão*, *Segurança*. Já os RNFs específicos são decomposições que se seguem aos RNFs primários e tendem a diminuir o nível de abstração de um determinado RNF, e portanto, atingir um nível de granularidade no tratamento da informação mais detalhado.

Um exemplo desta classificação seria o RNF Primário *Confiabilidade* que pode ser decomposto em *Validação, Autorização e Entrega do software*.

RNFs primários podem ser decompostos em outros RNFs secundários até que se chegue à operacionalização dos RNFs. Uma operacionalização de um RNF é uma ação ou informação que irá garantir a satisfação do RNF.

Existem duas abordagens para o tratamento de requisitos não-funcionais [CHUNG00]:

- Abordagem orientada a produto: avalia o grau que o produto final atende a determinados RNFs. Trata os RNFs do ponto de vista da evolução, isto é, evolução do produto final. Preocupa-se em elaborar definições formais dos RNFs, desta forma o sistema pode ser evoluído para um estágio que possua esses requisitos. Evoluções ocorrem no final do processo de desenvolvimento. A visão do produto é concentrada nas qualidades que o sistema deve ter. Seu propósito é utilizar restrições explícitas expressas pelos componentes do sistema, e testá-los sobre a ótica dos RNFs;
- Abordagem orientada a processo: integra o esforço de descrever RNFs durante o processo de desenvolvimento. Procura desenvolver técnicas para justificar as decisões tomadas durante o processo. Ao invés de evoluir o produto final, tenta racionalizar o processo de desenvolvimento em termos dos RNFs. Decisões de projeto podem afetar positiva ou negativamente cada um dos RNFs. A visão do processo se concentra no modo em que o projetista vai usar os RNFs para tomar decisões, tendo em vista o número de potenciais opções.

Não existe uma abordagem melhor que a outra, elas são complementares e devem ser usadas para obter sistemas que de fato atendam aos requisitos não-funcionais dos *stakeholders*. O tratamento com RNFs também pode ser classificado de forma quantitativa ou qualitativa. A abordagem orientada a produto é quantitativa, uma vez que estuda quantitativamente as métricas, preocupando-se em medir se um sistema de software satisfaz

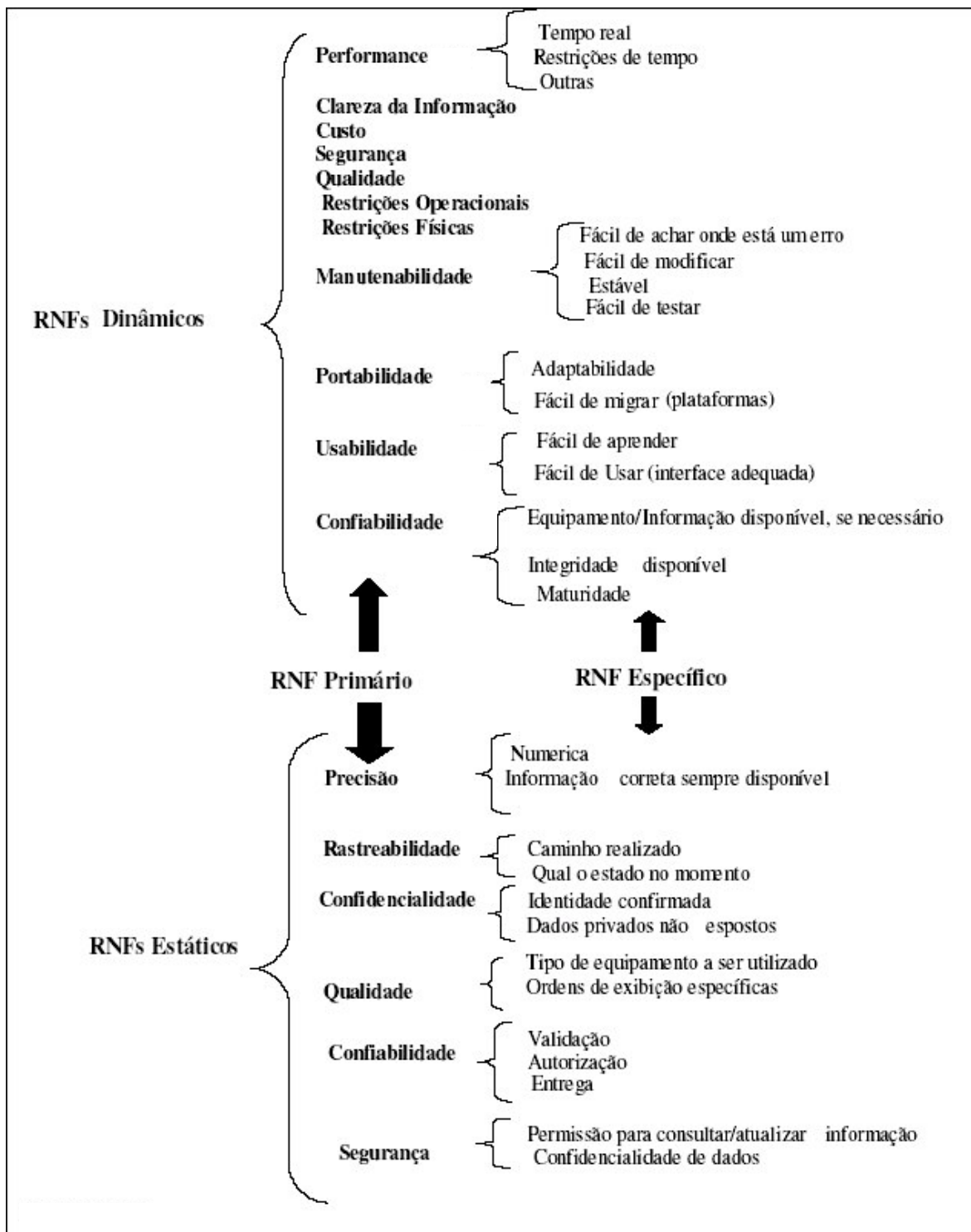


Figura 3.4 - Abordagens para tratamento de RNF segundo [CYSNEIROS01]

a determinados RNFs. Por outro lado, a abordagem orientada a processo é qualitativa, preocupando-se com as decisões, ao invés de medições. Na fase inicial da análise de requisitos a abordagem orientada a processo (com a avaliação qualitativa) é mais utilizada. Já num estágio mais avançado do processo, como na definição dos requisitos, a abordagem orientada a produto (com a avaliação quantitativa) ganha maior ênfase.

Em virtude das divergências de classificações e nomenclaturas dos RNFs, esse ramo da Engenharia de Software tornou-se bastante polêmico. Ainda se não bastasse, existem outros pontos que o torna mais complexo:

- Não existem regras para determinar quando um RNF está otimizado. Qualquer solução pode ser refinada para outra melhor, e assim sucessivamente. Esse efeito pode causar a sensação de que o requisito precisa ser melhorado sempre, não existindo ponto final;
- RNFs têm apenas boas ou más soluções, não certas ou erradas;
- Alguns problemas com RNFs nunca serão definitivamente testados;

Apesar da distinção entre requisitos funcionais e RNFs parecer ser clara, isso nem sempre acontece. Existem alguns casos que os requisitos podem ser expressos tanto em um tipo, quanto em outro. Para ilustrar essa consideração observe o seguinte requisito de segurança de um sistema.

- Abordagem funcional: o sistema deve conter um procedimento de autorização do usuário, no qual usuários serão identificados pelo login e pela senha. Apenas usuários autorizados devem acessar os dados do sistema.
- Abordagem não-funcional: o sistema deve garantir que dados sejam protegidos contra acessos não autorizados.

Esse exemplo nada mais é do que um abstrato RNF sendo decomposto em um requisito funcional mais detalhado, que especifica uma função que deve ser incorporada ao sistema.

Cysneiros [CYSNEIROS01] notou que a distinção entre o que é um requisito funcional e o que é um RNF nem sempre é clara. Parte da razão disto é que os RNFs estão sempre relacionados a requisitos funcionais. Com base na definição acima pode-se dizer que um requisito funcional expressa algum tipo de transformação que tem lugar no software, enquanto um RNF expressa como essa transformação irá se comportar ou que qualidades específicas ela deverá possuir. Em [DIDIER03] encontramos um processo de Engenharia de Requisitos que contempla as necessidades de integração entre requisitos funcionais e não-funcionais.

RNFs abordam importantes aspectos relacionados à qualidade de software, sendo essenciais para que softwares sejam bem sucedidos. O não tratamento destes requisitos resulta em: softwares com inconsistência e de baixa qualidade; clientes e desenvolvedores insatisfeitos; tempo e custo de desenvolvimento além dos previstos devido à necessidade de se consertar softwares que não foram desenvolvidos sob a ótica da utilização destes requisitos.

Apresentaremos na próxima seção os requisitos não-funcionais específicos de aplicações Web, levando em consideração as características particulares desses sistemas.

3.4.1 Requisitos Não-Funcionais para Web

No desenvolvimento de aplicações Web os RNFs desempenham um papel crítico. Problemas na elicitação deste tipo de requisito podem trazer consequências desastrosas, visto que estão entre os mais caros e difíceis de corrigir, uma vez que um sistema tenha sido implementado [DIDIER03].

Esses requisitos também estão presentes nas aplicações tradicionais mas, nas aplicações Web, eles adquirem maior relevância e visibilidade [PINHEIRO03]:

- *Segurança*: a segurança de um sistema Web é de extrema importância, não apenas do ponto de vista técnico, mas do próprio negócio. Falhas de segurança podem abalar a reputação da empresa e trazer consequências

desastrosas para o negócio. A segurança envolve fatores como política de privacidade e segurança nas transações. Quanto à política de privacidade, é necessário deixar claro que as informações fornecidas pelos usuários não serão utilizadas de forma indevida. A segurança das transações é uma questão técnica que tem sido resolvida através do uso de técnicas como criptografia e controle de acesso. O usuário tende a se sentir mais seguro efetuando transações em sites de empresas que ele já conhece. Uma forma de aumentar a confiança que o usuário tem na aplicação é fornecer informações sobre a empresa e alguma garantia de indenização caso algo dê errado;

- *Usabilidade*: em aplicações Web, a usabilidade é um fator de extrema relevância. Uma complicação adicional é a impossibilidade de realizar um treinamento com todos os usuários. Outra dificuldade é a grande diversidade do público alcançado pelas aplicações Web. Sabe-se que usuários se comportam de modo semelhante quando interagindo com certas páginas Web. A questão é como identificar esse comportamento e como identificar os vários grupos (perfis) de usuários e associar um comportamento padrão a cada um deles;
- *Performance*: uma característica presente na maioria dos usuários é a impaciência em relação à velocidade com que a aplicação Web é carregada. A questão da performance está ligada a fatores técnicos como tamanho dos arquivos, tipo de conexão, número de usuários, quantidade de informações manipuladas, etc;
- *Consistência*: uma preocupação em sites que contêm um grande número de informações é que não deve haver informações conflitantes;
- *Objetividade*: a dificuldade em encontrar informações é um dos fatores que podem levar o usuário a desistir de utilizar a aplicação;
- *Escalabilidade*: na história da Internet, não são raros os casos de empresas que atraíram um grande número de usuários para seus sites em um curto espaço de tempo. Como é muito difícil prever o número de

usuários de uma aplicação Web, o sistema deve ser o mais escalável possível e deve-se descobrir, através de simulações, como o sistema deve se comportar no caso de um grande aumento no número de usuários;

- *Disponibilidade*: algumas aplicações Web devem funcionar 24 x 7, isto é, vinte e quatro horas por dia, sete dias por semana. Para alcançar essa meta, faz-se necessário o uso de hardware e software de qualidade [HOLCK01];
- *Manutenibilidade*: a questão da manutenibilidade de sistemas Web está diretamente ligada à disponibilidade, já que há aplicações Web onde a manutenção deve ser feita sem que o serviço fique indisponível [HOLCK01];
- *Atualidade*: as informações presentes em um site devem estar sempre atualizadas, sob pena de causar má impressão aos usuários. Deve-se disponibilizar informações sobre quando a página foi escrita, colocada na Web e atualizada. Deve-se deixar claro quando os dados (gráficos, tabelas, pesquisas) foram obtidos;
- *Utilidade*: uma aplicação Web deve agregar o maior número possível de facilidades para seus usuários. Essas facilidades podem ser desde serviços prestados pela própria aplicação até links para outros sites que forneçam o serviço ou informação. Muitas vezes, entretanto, existe um conflito entre o que o usuário acha que é útil e o que os *stakeholders* do site desejam disponibilizar. Um link para o site de concorrentes pode ser considerado de grande utilidade pelo usuário, mas não é interessante em termos do negócio;
- *Personalização*: a enorme diversidade de usuários de uma aplicação Web faz com que nenhuma solução adotada agrade a todos. Uma solução para resolver esse problema é a incorporação de um sistema de personalização da aplicação. Deve-se identificar as características desejáveis e inaceitáveis para cada tipo de usuário, como ele se comporta quando interagindo com o site, que informações ele gostaria

de receber, que produtos ele costuma comprar etc. A partir dessas informações, a aplicação deve ser configurada de acordo com o usuário que a estiver utilizando;

Veremos agora, como estes e outros RNFs podem ser modelados.

3.4.2 Abordagens para Modelar Requisitos Não-Funcionais

De um modo geral, os RNFs definem qualidades e restrições sobre os sistemas de software [KOTONYA97]. Eles possuem um papel crítico durante o desenvolvimento de software, de modo que alguns pesquisadores têm alertado que eles deveriam receber mais atenção na literatura [CYSNEIROS01] [CHUNG00] [SOUSA03]. Eles são quase sempre descritos informalmente durante a fase de análise de requisitos e isso aumenta o risco de contradições e conflitos na especificação dos RNFs. Alguns aspectos inerentes à sua natureza aumentam a complexidade com que os mesmos devem ser tratados:

- Subjetividade: eles são interpretados e avaliados por diferentes pessoas que têm diferentes perspectivas e necessidades, assim eles podem ter diferentes significados para cada pessoa;
- Relatividade: sua interpretação e importância dependem diretamente de cada sistema em particular, além disso, sua realização é relativa (ex. um RNF pode ser essencial para uma pessoa enquanto para outra não);
- Interatividade: eles interagem entre si, assim a realização de um RNF pode interferir positivamente ou negativamente outros requisitos;
- Algumas limitações como, por exemplo, tempo de resposta, são inerentes ao projeto que ainda não é conhecido na fase de requisitos;
- RNFs tendem a ser refinados para um ou mais requisitos funcionais. Expressar requisitos funcionais e RNFs separadamente torna difícil identificar correspondências entre eles, e deixá-los juntos dificulta a identificação das considerações funcionais das não-funcionais;
- Como já foi citado anteriormente, não existem regras para determinar quando um RNF está otimizado.

Como foi dito, RNFs criam conflitos e contradições entre eles próprios, uma vez que a tentativa de satisfazer um destes requisitos pode prejudicar ou ajudar a satisfazer outros. Como vários RNFs possuem efeitos de natureza global nos softwares, uma solução localizada pode não ser adequada. Como exemplos de conflitos podemos citar:

- Segurança x Flexibilidade: flexibilidade requer estruturas mais gerais. Já segurança precisa utilizar estruturas específicas, que possam garantir certas limitações;
- Segurança x Performance: o processo adicional requerido para controlar o acesso ao sistema faz com que o tempo de resposta seja maior;
- Usabilidade x Performance: Os processos adicionais para facilitar as tarefas dos usuários, geralmente fazem com que os tempos de execução de determinados eventos sejam maiores.

Uma possibilidade de minimizar esse problema é listar os RNFs de acordo com os objetivos mais críticos. Como esses objetivos possuem prioridades diferentes, então o objetivo com maior prioridade deve ser satisfeito.

Por todos esses motivos, quase sempre é bastante difícil e complexo elicitar, expressar e quantificar RNFs [ALVES02]. Apesar do reconhecimento da importância dos RNFs, os métodos para especificar sistemas têm sua abordagem muito mais concentrada na modelagem de requisitos funcionais do que nos RNFs. A representação informal desses requisitos também contribui para isso. Poucos são os métodos de modelagem que provêm qualquer regra ou diretriz para capturar, mostrar ou representar as influências que esses requisitos podem causar no projeto do sistema. Alguns padrões de especificação possuem uma série de definições, mas poucos são os que propõem uma metodologia ou *framework* para tratá-los.

Além destas dificuldades, vale salientar que os RNFs não têm sentido de forma isolada. Alguns fatores influenciam a sua modelagem, como ilustrado na Figura 3.5. Os RNFs dependem, entre outros fatores, dos requisitos funcionais associados. O tempo de resposta

de uma operação, por exemplo, está intimamente ligado com o tipo da operação que está sendo executada. A cultura organizacional representada na Figura 3.5 pela Especificação da empresa também influencia a modelagem de RNFs. Geralmente, as empresas elegem alguma característica própria que é prioridade em todas as suas operações. Por exemplo, agilidade, ou segurança. Estas características pressionam os RNFs, de modo que se comportem como planejado. Além disso, as pessoas envolvidas no processo (especialmente os tomadores de decisão) tendem a interpretar e modelar os RNFs de acordo com suas visões.

Ao expressar os RNFs muitas vezes encontramos objetivos gerais, também conhecidos por objetivos *soft* [CHUNG00], que basicamente são algumas propriedades do sistema. Por exemplo:

- “Um sistema que deve ser fácil de usar e organizado, de forma que erros de usuários sejam minimizados”.

No entanto, é bastante difícil avaliar requisitos como esse, pois não há nada específico que possa ser medido. A única maneira de resolver esse problema é apresentando os RNFs de uma forma objetiva, para que possam ser testados. Um requisito é considerado subjetivo caso ele expresse um desejo, um objetivo, ou uma opinião pessoal. Ele será testável se existir algum processo pelo qual o requisito possa ser testado. Uma outra forma de expressar o requisito funcional mencionado acima seria:

- “O usuário deve estar familiarizado com o sistema em apenas três dias, e não deve cometer mais do que dois erros por dia”.

Para um maior esclarecimento de como RNFs podem ser expressos de forma objetiva e testável, algumas propriedades mensuráveis são mostradas na figura 3.6.

A relação entre diferentes RNFs merece certa atenção. Como já foi dito, algumas vezes são identificados RNFs contraditórios, ou dependências que fazem com que o cumprimento de

certo requisito dependa de outro. É importante que esses conflitos sejam previamente identificados e documentados, para que decisões de projeto possam ser tomadas.

Existem alguns métodos presentes na literatura para assistir a descrição de RNFs [KOTONYA97] [DOBSON91] [LOUCOPOULOS95] [CHUNG00]. Por exemplo, Loucopolos [LOUCOPOULOS95] propôs uma abordagem comparativa para derivar RNFs. Baseia-se na decomposição sucessiva de objetivos em sub-objetivos, e posteriormente em RNFs. Dentre esses trabalhos, o *NFR Framework* proposto por Chung [CHUNG00] é considerado um dos métodos mais completos e bem aceitos para modelar RNFs [SOUSA03].

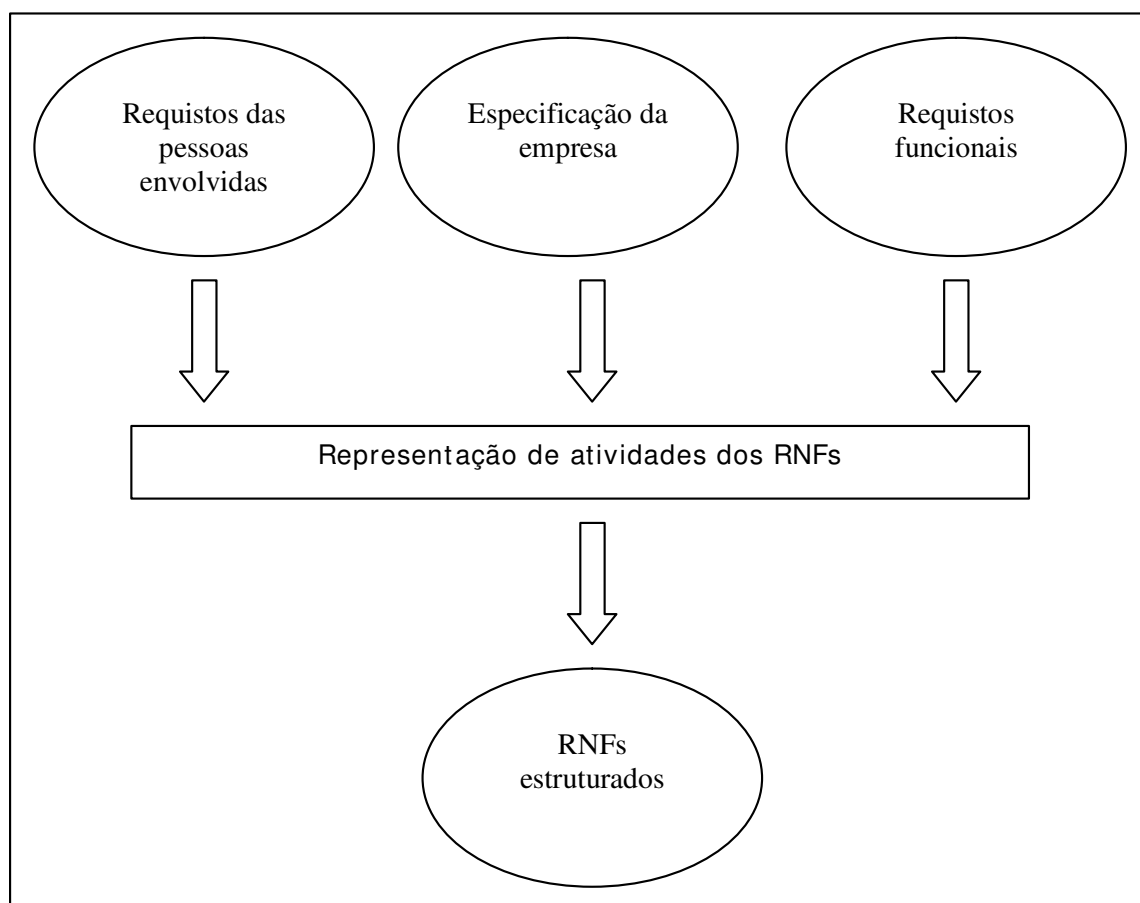


Figura 3.5 – Fatores que influenciam RNFs

3.4.2.1 NFR Framework

O NFR *Framework* é uma abordagem orientada a processo onde os RNFs são explicitamente representados como metas a serem obtidas [CHUNG95a]. Uma importante característica dessas metas é que nem sempre elas podem ser inteiramente satisfeitas. Ao contrário, existem graus de satisfação que podem ser obtidos. Essa abordagem pode tratar melhor os aspectos de subjetividade, relatividade e interatividade inerentes aos RNF. Dessa

Propriedade	Forma de medir
Performance	Tempo de atualização da tela, tempo de resposta de determinado evento, número de transações processadas por segundo.
Tamanho	Tamanho expresso em unidades de armazenamento (Kbytes, GB), quantidade de memória RAM necessária.
Facilidade de uso	Tempo para treinamento (para aprender 80% das facilidades, por exemplo), forma (on-line, manual de instrução) e quantidade de ajudas (<i>help</i>), número de erros produzidos por usuários em determinado período de tempo.
Confiabilidade	Tempo de disponibilidade, probabilidade de indisponibilidade, quantidade de ocorrência de falhas.
Robustez	Tempo de reinício após falhas, porcentagem de eventos que causam falhas, probabilidade de corrupção de dados nas falhas.
Portabilidade	Quantidade de sistemas dependentes.

Figura 3.6 – Propriedades mensuráveis dos RNFs

forma, Chung [CHUNG95b] propõe a noção de *softgoal* para descrever as metas que precisam obter níveis de satisfação. Uma premissa fundamental dessa abordagem é que os *softgoals* possuem a propriedade de interagir entre si, em conflito ou sinergia. Os *softgoals* são decompostos em outros *softgoals* e representados numa estrutura gráfica inspirada nas árvores “AND/OR” para solução de problemas, esses grafos são chamados SIG (*Softgoal Interdependency Graphs*). A aplicação do NFR *Framework* consiste de dois estágios

principais, Aquisição do Conhecimento e Aplicação do *Framework*. A seguir são descritas as atividades que deverão ser executadas em cada fase.

Fase 1 - Aquisição do Conhecimento

- Adquirir conhecimento específico dos RNFs – essa atividade consiste em obter conhecimento sobre os RNFs a partir de experiências práticas ou da literatura disponível. Em seguida, os RNFs são incluídos em catálogos específicos e que serão usados como fonte de informação para as próximas fases;
- Adquirir conhecimento do domínio – durante essa atividade são adquiridas informações sobre o domínio do problema e uma idéia geral dos principais requisitos do sistema.

Fase 2 – Aplicação do NFR Framework

- Identificar conceitos relacionados aos RNFs – nesse momento é feita uma estimativa inicial das prioridades da organização, isso inclui os requisitos críticos. Além disso, são identificados os possíveis métodos ou operacionalizações desses requisitos como também os argumentos para priorizar os requisitos;
- Refinar e inter-relacionar conceitos dos RNFs – essa fase inclui uma série de atividades, são elas:
 - Refinar e relacionar os *softgoals* para esclarecer possíveis ambigüidades;
 - Relacionar métodos para operacionalizar *softgoals*;
 - Identificar *tradeoffs*, interdependências (conflito/sinergia) entre os *softgoals*;
 - Identificar prioridades, assim como partes críticas e dominantes;
 - Fornecer argumentos para tomada de decisão e motivos para escolher determinado método;
 - Avaliar a realização dos *softgoals* (particularmente os prioritários).

relacionamento é negativo, isto é, existe uma relação de conflito entre os requisitos. Em particular, a operacionalização *Autorização[usuário]* interage em conflito com *Simplicidade[sistema]*. Ou seja, os métodos para autorização dos usuários influenciam negativamente a simplicidade do sistema. Em situações como essa é necessário um extenso processo de priorização e negociação desses requisitos, onde a participação dos *stakeholders* assume um papel fundamental.

Tendo explicado resumidamente como funciona o NRF Framework, veremos na próxima seção dois modelos de qualidade propostos por organizações internacionalmente reconhecidas.

3.4.3 Modelos de Qualidade

Conforme a norma ISO [ISO9126] os atributos de qualidade de um software podem ser analisados através de seis características:

- *Funcionalidade*: capacidade do produto de prover as funções definidas nas fases de desenvolvimento, relativas às necessidades explícitas e implícitas dos usuários;
- *Confiabilidade*: capacidade do produto de manter um nível de performance especificado quando usado dentro de condições previstas;
- *Usabilidade*: capacidade do produto de ser entendido, aprendido, usável e atrativo para o usuário, quando utilizado dentro de condições especificadas;
- *Eficiência*: capacidade do produto de prover uma performance apropriada, relativamente ao total de recursos usados, sob condições especificadas;
- *Manutenibilidade*: capacidade do produto de ser modificado. Modificações incluem correções, melhorias ou adaptações;
- *Portabilidade*: capacidade do produto de ser transferido de uma ambiente (organizacional, de hardware ou de software) para outro.

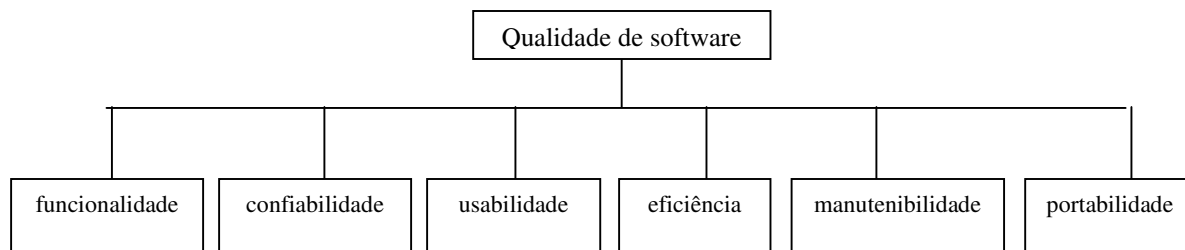


Figura 3.8 – RNFs vistos como características de qualidade

Estas características, por sua vez, podem ser subdivididas. Os padrões ISO [ISO9126] e IEEE [IEEE1061] definem modelos de qualidade de produtos, mostrados nas Figuras 3.9 e 3.10. A metodologia de avaliação de qualidade proposta por esta dissertação utiliza esses modelos como base para representar os requisitos não-funcionais. Com relação às características de qualidade, estes dois modelos coincidem significativamente. Já com respeito às subcaracterísticas, manifestam algumas diferenças. Em algumas situações as subcaracterísticas têm o mesmo nome, mas o significado ligeiramente diferente. Em outras têm nomes diferentes, porém o mesmo significado.

Na próxima seção estudaremos um documento de fundamental importância no processo de desenvolvimento de software, e que serve como entrada para a avaliação proposta por este trabalho.

3.5 Documento de Requisitos

Uma vez identificados e negociados, os requisitos devem ser documentados para que possam servir de base para o restante do processo de desenvolvimento. A linguagem natural é a capaz de ser compreendida por todos os participantes do processo de requisitos e, dessa forma, pode ser utilizada para representá-los. [KOTONYA96]

Contudo, sempre que for necessário, pode-se incluir à documentação dos requisitos diagramas, gráficos, tabelas entre outros que facilitem o entendimento e a elucidação das informações registradas no documento de requisitos. Davis [DAVIS93] destaca que, apesar da vantagem da linguagem natural, que pode ser entendida por todos os participantes do

Característica	Subcaracterística	Pergunta chave para a subcaracterística
Funcionalidade (satisfaz as necessidades?)	Adequação	Propõe-se a fazer o que é apropriado?
	Acurácia	Faz o que foi proposto de forma correta?
	Interoperabilidade	Interage com os sistemas especificados?
	Conformidade	Está de acordo com as Normas, leis, etc.?
	Segurança de acesso	Evita acesso não autorizado aos dados?
Confiabilidade (é imune a falhas?)	Maturidade	Com que frequência apresenta falhas?
	Tolerância a falhas	Ocorrendo falhas, como ele reage?
	Recuperabilidade	É capaz de recuperar dados em caso de falha?
Usabilidade (é fácil de usar?)	Inteligibilidade	É fácil entender o conceito e a aplicação?
	Apreensibilidade	É fácil aprender a usar?
	Operacionalidade	É fácil de operar e controlar?
Eficiência (é rápido e "enxuto"?)	Tempo	Qual é o tempo de resposta, a velocidade de execução?
	Recursos	Quanto recurso usa? Durante quanto tempo?
Manutenibilidade (é fácil de modificar?)	Analísabilidade	É fácil de encontrar uma falha, quando ocorre?
	Modificabilidade	É fácil modificar e adaptar?
	Estabilidade	Há grande risco quando se faz alterações?
	Testabilidade	É fácil testar quando se faz alterações?
Portabilidade (é fácil de usar em outro ambiente?)	Adaptabilidade	É fácil adaptar a outros ambientes?
	Capacidade para ser instalado	É fácil instalar em outros ambientes?
	Conformidade	Está de acordo com padrões de portabilidade?
	Capacidade para substituir	É fácil usar para substituir outro?

Figura 3.9 – Tabela das características e subcaracterísticas definidas pela ISO [ISO9126]

processo de requisitos, ela apresenta um alto grau de ambigüidade, o que favorece o aparecimento de inconsistências.

Segundo ele, isso acontece, pois, quando nos comunicamos verbalmente, parte dos elementos utilizados em nossa expressão oral na linguagem natural - a expressão corporal - é perdida. Dessa forma, conseguimos registrar apenas parte das informações que expressamos. Diversas iniciativas tentaram contornar esse problema [LOPES02]. Leite

Característica	Subcaracterística	Pergunta chave para a subcaracterística
Funcionalidade (as funções e propriedades satisfazem todas as necessidades explícitas e implícitas dos usuários?)	Compleitude	Possui as necessárias e suficientes funções para satisfazer os requisitos dos usuários?
	Corretude	Especifica todas as funções?
	Segurança	Detecta e previne uso ilegal, destruição de recursos e perda de dados?
	Compatibilidade	O novo software pode ser instalado sem trocar o ambiente e modificar as condições?
	Interoperabilidade	Conecta-se e opera facilmente com outros sistemas?
Confiabilidade (é imune a falhas?)	Ausência de defeitos	Possui erros?
	Tolerância a falhas	Ocorrendo um erro, como se comporta com relação a performance especificada? Possui funções de recuperação de falhas?
	Disponibilidade	É capaz de recuperar dados em caso de falha?
Usabilidade (é fácil de usar?)	Compreensibilidade	É fácil entender o conceito e a aplicação?
	Facilidade de Aprender	É fácil aprender a usar?
	Operabilidade	É fácil de operar e controlar?
	Nível de comunicação	O software se projeta conforme as características psicológicas dos usuários?
Eficiência (é rápido e não desperdiça recursos, sob certas condições?)	Economia com relação ao tempo	Realiza funções específicas sob certas condições e em tempo apropriado?
	Economia com relação aos recursos	Realiza funções específicas sob certas condições e em tempo apropriado?
Manutenibilidade (é fácil de modificar e testar?)	Corretabilidade	É fácil corrigir erros e tratar com as demandas dos usuários?
	Expansibilidade	É fácil modificar e adaptar?
	Testabilidade	É fácil de testar?
Portabilidade (é fácil de usar em outro ambiente?)	Independência de hardware	O software depende de outros hardwares específicos?
	Independência de software	O software depende de outros softwares específicos?
	Instalabilidade	É fácil ajustar o software para um novo ambiente?
	Reusabilidade	É fácil reusar o software em outras aplicações?

Figura 3.10 – Tabela das características e subcaracterísticas definidas pela IEEE

[IEEE1061]

introduziu o LEL - Léxico Estendido de Linguagem [LEITE93b]. Seu objetivo era restringir e definir precisamente o vocabulário utilizado nas especificações de requisitos. Fuchs e Schwitter propuseram uma linguagem de especificação denominada ACE – Attempto Controlled English [FUCHS96], na qual padrões de escrita restringem a gramática e o vocabulário, permitindo a documentação de requisitos de forma mais previsível e menos ambígua. ACE é um subconjunto do Inglês para escrever especificações de requisitos. Especificações escritas nessa linguagem dão a impressão de serem informais. São, contudo, passíveis de execução computacional.

Entre os muitos problemas que enfrentamos na documentação de requisitos, certamente, administrar o grande volume de informações gerado pelo processo de requisitos é um dos principais. O uso de uma notação gráfica certamente diminuiria o tamanho do modelo, tornando mais fácil a atividade de administrar a documentação dos requisitos. Isto faz com que se utilizem notações tradicionalmente empregadas na atividade de projeto para modelar requisitos. Apesar de ser fato a necessidade de se utilizar alguma notação para podermos administrar o problema de requisitos, utilizar notações de projeto pode polarizar a solução em favor da implementação [ZAVE97].

Uma boa opção para documentar requisitos é o uso de uma linguagem baseada em objetos, agentes e seus objetivos, desenvolvida exclusivamente para modelar requisitos (ex. KAOS) [DARDENE93]. Ainda na linha de objetivos, porém estendendo o conceito para o contexto organizacional e para o caso particular de sistemas de informação, Yu [YU97] desenvolveu a técnica denominada i^* (i estrela). i^* é utilizada nas fases iniciais do processo de requisitos para raciocinar sobre ambientes organizacionais e seus sistemas de informação. Modela, basicamente, a relação entre vários atores do contexto organizacional (Modelo de Dependência Estratégica) e a descrição de interesses e responsabilidades de cada um deles, caracterizando como os objetivos e interesses de cada um deles podem ser atendidos através das várias configurações de sistema e ambiente (Modelo de Raciocínio Estratégico).

Outros conceitos foram desenvolvidos para a documentação de requisitos não-funcionais, como é o caso de Softgoals, utilizados para caracterizar objetivos que podem ser

parcialmente atendidos [MYLOPOULOS99]. Objetivos exercem influência sobre outros objetivos. Essa influência pode ser favorável a seu atendimento (influência positiva) ou contrária a seu atendimento (influência negativa). Um softgoal é considerado atendido se existem mais influências positivas que negativas sobre ele. Esse conceito é fundamental para tratar requisitos não-funcionais cujo atendimento é fortemente baseado em soluções de compromisso. Por exemplo, mais desempenho aumenta o consumo de memória e pode requerer a construção de algoritmos especiais, o que será mais caro. Assim, a decisão sobre qual o nível de desempenho adequado depende do consumo de memória tolerável, do custo que se está disposto a pagar e do desempenho mínimo necessário para não comprometer a aplicação em desenvolvimento.

Neste trabalho propomos uma estrutura (template) de um documento de requisitos. O documento devidamente preenchido serve como entrada para a metodologia REQE, proposta nesta dissertação, que produz como saída a avaliação do futuro produto, espelho daquele documento.

A elaboração de um documento de requisitos pressupõe a existência da definição do domínio da aplicação, para que possam ser levantadas características importantes que possuem relação direta com o produto a ser desenvolvido. Por exemplo, se estivermos interessados no domínio de sites acadêmicos, alguns atributos como mecanismos de busca por professores e índice de disciplinas acadêmicas devem estar presentes, seguindo a sua análise de domínio. Muitos desses atributos não estariam presentes caso o domínio da aplicação fosse de comércio eletrônico [OLSINA00a].

O documento de requisitos é a declaração oficial do que é desejado do sistema. Descreve seus requisitos e é utilizado para prover comunicação entre usuários, desenvolvedores e gerentes. Possui um conjunto diversificado de usuários, abrangendo desde a alta gerência até usuários finais e engenheiros de software. Por isso, em muitos casos, existem versões distintas do documento, porque comunica informações sobre o sistema a diversos tipos de leitores [SOMMERVILLE03]. Não existe um nome padrão único para representá-lo, em diferentes organizações são utilizados diferentes denominações, como especificação

funcional, especificação dos requisitos do sistema etc. Seguem alguns tópicos que o documento de requisitos poderá abordar:

- Serviços e funções que o sistema deve prover;
- Limitações sobre as quais o sistema deve operar;
- Propriedades gerais do sistema, isto é, comportamento do sistema diante das propriedades emergentes;
- Definições de outros sistemas que irão interagir com o sistema em questão;
- Informações a respeito do domínio de aplicação do sistema;
- Limitações no processo utilizado para desenvolver o sistema;
- Definições sobre o hardware no qual o sistema executará.

Adicionalmente, o documento de requisitos deverá conter uma capítulo introdutório que provê um resumo do sistema, necessidades de negócio suportadas pelo sistema e um glossário que explica a terminologia utilizada [LEITE97]

. Esse glossário é particularmente importante, pois pessoas de diferentes áreas terão contato com o documento de requisitos, facilitando assim sua compreensão.

Seguem exemplos de usuários do documento de requisitos, com seus respectivos papéis:

- Clientes do Sistema: especificam os requisitos e o lêem para avaliar se os mesmos satisfazem suas necessidades;
- Gerentes de Projeto: usam os documentos de requisitos para planejarem uma proposta para o sistema e o processo de desenvolvimento do sistema;
- Engenheiros de Sistema: usam os requisitos para entenderem o sistema em construção;
- Engenheiros de teste do sistema: usam os requisitos para desenvolverem testes de validação do sistema;
- Engenheiros de manutenção do sistema: usam os requisitos para entenderem o sistema.

Muitas organizações propuseram uma estrutura para o documento de requisitos. Entre elas podemos citar o padrão IEEE [IEEE830]

, que sugere uma estrutura para o documento de requisitos, apresentada na Figura 3.10.

É possível omitir partes, e adicionar novas seções, se necessário. Também podemos definir uma lista de partes estáveis e variáveis. Partes estáveis devem aparecer em todos os documentos de requisitos, sejam quais forem os seus leitores. Partes variáveis devem ser incluídas dependendo do grupo alvo. É como se existisse uma versão executiva, mais resumida, e uma versão técnica, como todos os detalhes. Na seção a seguir será proposto um template para um documento de requisitos.

Documento de Requisitos	
1. Introdução	
1.1.	Propósito do documento de requisitos
1.2.	Escopo do produto
1.3.	Definições, acronismos e abreviações
1.4.	Referências
1.5.	Resumo do resto do documento
2. Descrições Gerais	
2.1.	Perspectiva do produto
2.2.	Funções do produto
2.3.	Características do usuário
2.4.	Limitações gerais
2.5.	Suposições e dependências
3. Requisitos específicos	
3.1.	Cobrem requisitos não-funcionais, funcionais e interface
4. Apêndices	
5. Index	

Figura 3.10 – Estrutura de um documento de requisitos segundo IEEE 830.

3.5.1 Template do Documento de Requisitos Proposto

Apresentaremos nesta seção uma estrutura de um documento de requisitos. Uma vez preenchido o documento de requisitos será possível aplicar a metodologia proposta, que

será detalhada no próximo capítulo. Vale salientar que este template pode ser modificado ou parcialmente aproveitado, de acordo com o interesse dos analistas de requisitos responsáveis. No apêndice A é possível visualizar um exemplo não preenchido de um documento de requisitos seguindo o template da Figura 3.11. Cada seção vem acompanhada de uma explicação, a fim de facilitar o seu preenchimento.

No capítulo anterior nos concentramos nas aplicações Web e em algumas metodologias de avaliação de qualidade existentes na literatura. Neste capítulo falamos dos requisitos de uma forma geral, dando ênfase aos requisitos não-funcionais e apresentamos os conceitos da Engenharia do Domínio, capaz de auxiliar a metodologia proposta por esta dissertação por meio de técnicas de análise de domínios. Listamos alguns requisitos não-funcionais críticos para a Web, bem como formas para modelá-los. Apresentamos alguns modelos de qualidade. Mostramos o que deve conter um documento de requisitos e propomos uma estrutura para que seja utilizada no exemplo do capítulo 5. Apresentaremos no capítulo seguinte a metodologia proposta por esta dissertação.

Sumário	5. Requisitos Funcionais do Produto
1. Introdução	5.1 <Funcionalidade 1>
1.1 Finalidade	5.2 <Funcionalidade 2>
1.2 Escopo	...
1.3 Definições, Acrônimos, e Abreviações	5.n <Funcionalidade n>
1.4 Referências	6. Precedência e Prioridade
1.5 Visão geral	7. Requisitos Não-Funcionais do Produto
2. Contextualização do Produto	7.1 Requisitos de Usabilidade
2.1 Oportunidade de Negócio	7.2 Requisitos de Confiabilidade
2.2 Descrição do Problema	7.3 Requisitos de Eficiência
3. Descrição dos Stakeholders	7.4 Requisitos de Manutenibilidade
3.1 Resumo dos Stakeholders	7.5 Requisitos de Portabilidade
3.2 Ambiente do Usuário	7.6 Outros Requisitos
3.3 Necessidades chave dos Stakeholders	8. Interfaces
4. Visão Geral do Produto	8.1 Interfaces de Usuário
4.1 Perspectiva do Produto	8.2 Interfaces de Hardware
4.2 Resumo das características do produto	8.3 Interfaces de Software
4.3 Premissas e Dependências	8.4 Interfaces de Comunicação
	9. Restrições
	10. Recomendações

Figura 3.11 – Template do documento de requisito proposto

4 REQE – Uma Metodologia para Medição da Qualidade de Aplicações Web na Fase de Requisitos

Neste capítulo apresentamos os conceitos da metodologia **REQE** (Requirements Engineering Quality Evaluation), que tem como objetivo avaliar a qualidade de aplicações Web na fase de requisitos.

4.1 Introdução

Uma metodologia de avaliação de qualidade eficiente é um grande passo para atingir produtos de melhor qualidade. Avaliar é medir. Quando medimos estamos ao mesmo tempo compreendendo, controlando. E esta é uma forma de melhorar a qualidade, pois temos a possibilidade de interpretar os resultados e identificar o que pode ser evoluído.

As metodologias de avaliação de qualidade de aplicações Web em geral efetuam as medições dos produtos nas fases finais do processo de desenvolvimento. Estamos propondo nesta dissertação uma metodologia, batizada de **REQE (Requirements Engineering Quality Evaluation)** [REIS02], que visa antecipar esta avaliação. Como esta metodologia se propõe a avaliar a qualidade de aplicações Web numa fase inicial do processo de desenvolvimento, temos como benefício a possibilidade de descobrir erros de uma forma antecipada. E segundo Pressman [PRESSMAN02], o custo de correção de erros na fase de definição de requisitos é cerca de três a seis vezes mais baixo do que na fase de projeto. E este custo aumenta ainda mais quando a correção de erros é realizada em fases mais avançadas do processo de desenvolvimento, como a fase de operação.

À medida que os erros são identificados, tem-se a possibilidade de corrigí-los. Portanto, o objetivo da metodologia proposta é além de avaliar a qualidade de aplicações Web na fase de requisitos, contribuir com a construção de produtos de melhor qualidade.

A aplicação da metodologia REQE consiste em cinco fases:

1. Representação das Características, Subcaracterísticas e Atributos de Qualidade;
2. Especificação Descritiva da Árvore de Características, Subcaracterísticas e Atributos de Qualidade;
3. Associação de Pesos aos Nós;

4. Associação de Escores aos Atributos de Qualidade;

5. Cálculo Geral.

As fases serão detalhadas nas seções a seguir.

4.2 Representação das Características, Subcaracterísticas e Atributos de Qualidade

Nesta primeira fase da metodologia REQE serão elicitados e especificados os requisitos de qualidade, culminando com uma árvore que hierarquicamente lista todas as características, subcaracterísticas e atributos de qualidade, capazes de modelar a qualidade segundo as necessidades do usuário. Cada atributo de qualidade deve ser medido por meio de um escore de conformidade. Este escore verifica se o produto atenderá ou não (com base no documento de requisitos) àquele atributo de qualidade. Na Seção 4.5 apresentaremos os escores propostos. Vale salientar que a metodologia REQE é flexível, permitindo a agregação de novas características, subcaracterísticas e atributos de qualidade de uma forma modular, conforme as necessidades específicas que venham a surgir no decorrer do processo de avaliação.

Já vimos que a norma ISO [ISO9126] define seis características de alto nível capazes de quantificar a qualidade de um produto de software: usabilidade, funcionalidade, confiabilidade, eficiência, portabilidade e manutenibilidade. A partir destas características se derivam subcaracterísticas e a partir destas, seguindo um processo de decomposição recursivo, pode-se especificar atributos de qualidade. Veremos na próxima seção que este processo de decomposição leva em consideração dois pontos: o domínio da aplicação e o perfil do usuário (suas necessidades e comportamentos).

A primeira fase da metodologia REQE possui dois passos. O primeiro passo será apresentado na Seção 4.2.1 e o segundo na Seção 4.2.2. No início de cada seção apresentaremos resumidamente a respectiva fase (ou o passo) em **negrito**, a fim de facilitar o entendimento da evolução da metodologia REQE.

4.2.1 Escolha do Domínio da Aplicação e do Perfil do Interessado.

- Fase 1 - Representação das Características, Subcaracterísticas e Atributos de Qualidade
 - ❖ Passo 1 - Escolha do Domínio da Aplicação e do Perfil do Interessado.
 - ❖ Identificar Domínio da Aplicação;
 - ❖ Selecionar Perfil do Interessado.

Na fase de *Representação das Características, Subcaracterísticas e Atributos de Qualidade*, o primeiro passo consiste em identificar o domínio sob o qual a aplicação está inserida e sob qual ótica do usuário (perfil do interessado) iremos efetuar a avaliação de qualidade.

Antes de confeccionar a árvore de características, subcaracterísticas e atributos de qualidade (nesta dissertação chamada de ACSAQ) temos que identificar qual o domínio da aplicação e o perfil do interessado. Procuramos criar um processo genérico o suficiente para atender qualquer domínio na Web. O domínio da aplicação existe, independente do sistema de avaliação [OLSINA00b]. Os aspectos importantes do domínio serão baseados na Engenharia do Domínio, conforme descrevemos na Seção 3.3, que então definirá o que é (ou não) interessante, útil, importante em determinado domínio. Portanto, o uso da nossa proposta REQE assume como premissa que um adequado tratamento do domínio foi conduzido.

Também é importante a seleção do perfil do interessado, pois a relevância das características de qualidade varia de acordo com o ponto de vista considerado [ISO9126]. De acordo com Olsina [OLSINA00b], por exemplo, a visão de um usuário visitante de uma aplicação Web em geral, consiste no interesse do mesmo em usar os artefatos de software disponíveis, em realizar tarefas de forma ágil (performance, eficiência), em aprender rapidamente e usar com facilidade (usabilidade), entre outras questões. Esses usuários não estão interessados nas características internas do produto ou no desenvolvimento dos artefatos, apesar de sabermos que os atributos do produto contribuem para a qualidade em uso [OLSINA00b]. Estas características são tão transparentes que os usuários comuns não chegam nem mesmo a se importar com elas.

Por outro lado, a visão de qualidade dos desenvolvedores deve considerar não apenas os requisitos do produto à mostra, como também a qualidade dos produtos intermediários resultantes das atividades das fases de desenvolvimento [OLSINA00b]. Neste contexto, a portabilidade merece uma especial atenção dos desenvolvedores, uma vez que a dificuldade em migrar sistemas de um ambiente para outro está intimamente relacionada com esta característica (interna) de qualidade. Outra característica, como a manutenibilidade, também desperta interesse na comunidade dos desenvolvedores, pois estes serão responsáveis por atividades de manutenções, sejam estas corretivas, adaptativas ou preventivas.

Uma outra visão de qualidade é a dos gerentes, consistindo em uma visão integradora que necessita incorporar requisitos de negócios [OLSINA00b]. Um gerente não leva em consideração apenas o resultado prático das atividades, mas também os meios e os esforços necessários para atingi-lo. Nesta visão é importante desenvolver artefatos de qualidade, porém este desenvolvimento está atrelado a custos (que em grande parte precisam ser reduzidos) e deve ser executado com base em um cronograma “apertado”, para que não haja estouro nos prazos de conclusão do projeto.

Nada impede que sejam encontradas classes mais específicas, dentro de determinados domínios. No caso, por exemplo, do domínio de museus acadêmicos [GARZOTTO97a] [GARZOTTO97b] [FURANO97], a classes dos visitantes foi decomposta em visitantes casuais e visitantes intencionais. Segundo Garzotto, um visitante casual se define como uma audiência que ingressa no site por azar (um erro de indexação de uma ferramenta de busca, por exemplo) e permanece provavelmente por um curtíssimo intervalo de tempo. Já um visitante intencional principiante é considerado um usuário que tem ao menos algum conhecimento ou manifesta algum interesse no domínio em questão, e deseja provavelmente coletar alguma informação e aprender mais acerca do mesmo. Sua permanência no site geralmente é maior do que uma audiência casual. Um outro tipo de visitante foi identificado, o intencional especialista, e correspondente à audiência de especialistas no domínio. Sua permanência no site é ainda maior. Este tipo de visitante,

geralmente está interessado em todo o conteúdo (na Web) e em todos os itinerários previstos, tanto na página principal como nas restantes [FURANO97].

Sabemos que cada aplicação Web está inserida em um domínio e que a sua qualidade deve ser medida levando em consideração um perfil de usuário. Veremos agora como, de fato é confeccionada a árvore de características, subcaracterísticas e atributos de qualidade.

4.2.2 Especificação da Árvore de Características, Subcaracterísticas e Atributos de Qualidade

- Fase 1 - Representação das Características, Subcaracterísticas e Atributos de Qualidade
 - Passo 1 - Escolha do Domínio da Aplicação e do Perfil do Interessado.
 - Identificar Domínio da Aplicação;
 - Selecionar Perfil do Interessado.
 - ❖ **Passo 2 - Especificação da Árvore de Características, Subcaracterísticas e Atributos de Qualidade**

Neste momento, os avaliadores e demais participantes do processo devem chegar a um acordo e especificar as características e os atributos de qualidade e agrupá-los em uma árvore, conforme o domínio (lembrar de recorrer à Engenharia do Domínio) e o perfil do interessado selecionado.

Adotaremos um enfoque de decomposição *top-down*. Partimos das seis características definidas pela norma ISO [ISO9126] (usabilidade, funcionalidade, confiabilidade, eficiência, portabilidade e manutenibilidade) que podem ser (ou não) relevantes dependendo do domínio da aplicação e do perfil do interessado. Como essas características não podem ser medidas diretamente, as mesmas são decompostas em subcaracterísticas e estas em atributos de qualidade, até que possam ser quantificadas. Dentre as seis características, devemos selecionar quais devem estar presentes na árvore, levando em consideração principalmente o perfil do interessado.

A especificação da ACSAQ determinará os aspectos a serem medidos. Outros requisitos não-funcionais podem ser agregados por especialistas no domínio.

Algumas dessas características possuem uma dificuldade adicional para serem medidas na ausência de um produto final, outras podem ser mais facilmente avaliadas. Apesar da árvore se basear em métricas de produto e ainda não estar disponível o produto propriamente dito, pois estamos na fase de requisitos do processo de desenvolvimento, já é possível efetuar uma medição, uma vez que o documento de requisitos deve prever e especificar o que será construído. Existem, no entanto, atributos que necessariamente apenas podem ser avaliados na existência do produto final. Estes não devem estar presentes na nossa árvore.

O primeiro nível da árvore é composto por um subconjunto das características de qualidade baseadas na norma ISO [ISO9126]. No nível abaixo das características estão as subcaracterísticas que serão decompostas ou os atributos de qualidade. Por exemplo, na Figura 4.1 definimos as subcaracterísticas hipotéticas de *Usabilidade*: *SubcaracterísticaA1*, *SubcaracterísticaA2* e *SubcaracterísticaA3*. As subcaracterísticas devem ser originadas da decomposição da característica pai (no caso Usabilidade), levando em consideração:

- Os modelos de qualidade, tais como padrões ISO [ISO9126] e IEEE 1061, presentes na Seção 3.4.3;
- O domínio da aplicação, com o auxílio da Engenharia do Domínio;
- O perfil do interessado;
- A opinião de especialistas.

Se existirem subcaracterísticas, estas devem ser decompostas em atributos de qualidade. Por exemplo, na Figura 4.1, a *SubcaracterísticaA1* foi decomposta em três atributos de qualidade (*AtributoA11*, *AtributoA12* e *AtributoA14*) e em uma outra subcaracterística (*SubcaracterísticaA13*). Os atributos de qualidade podem ser diretamente medidos, as subcaracterísticas, por sua vez, necessitam de mais decomposições, como aconteceu no

caso da subcaracterística *SubcatacterísticaA13*, decomposta em *AtributoA131* e *AtributoA132*. Uma subcaracterística deve ser decomposta até encontrarmos atributos de qualidade quantificáveis (de forma objetiva vide Seção 4.5). Qualquer nó da árvore que não possuir filho é um atributo de qualidade, podendo ser medido diretamente, sem a necessidade de ser decomposto.

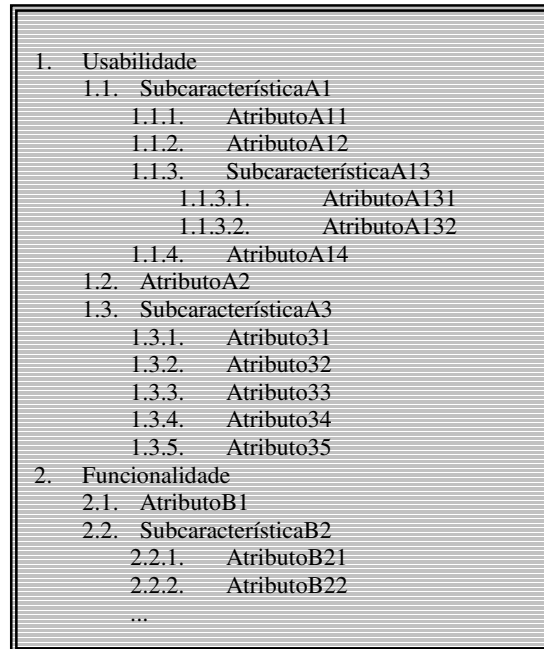


Figura 4.1 – Parte de uma *ACSAQ*

Uma vez definida a árvore de características, subcaracterísticas e atributos de qualidade passaremos para a próxima fase.

4.3 Especificação Descritiva da Árvore de Características, Subcaracterísticas e Atributos de Qualidade

- Fase 1 - Representação das Características, Subcaracterísticas e Atributos de Qualidade
 - Passo 1 - Escolha do Domínio da Aplicação e do Perfil do Interessado.
 - Passo 2 - Especificação da Árvore de Características, Subcaracterísticas e Atributos de Qualidade
- ❖ Fase 2 - Especificação Descritiva da Árvore de Características, Subcaracterísticas e Atributos de Qualidade

Nesta fase, todos os nós da ACSAQ devem ser especificados de forma descritiva. É preciso descrever cada nó detalhadamente, a fim de dirimir eventuais dúvidas a cerca de seus significados. A título de documentação, para cada característica, subcaracterística e atributo de qualidade, deve ser preenchida uma planilha de informação (baseada na planilha em [OLSINA00b]).

Para as características, que estão no primeiro nível da ACSAQ, devem ser especificados os seguintes campos:

- Título: título da característica na ACSAQ;
- Código: código da característica na ACSAQ;
- Modelo de Qualidade: modelo de qualidade que descreve a característica;
- Quantidade de filhos: quantidade de subcaracterísticas e atributos de qualidade filhos na ACSAQ;
- Subcaracterísticas: título e código das subcaracterísticas filhas diretas na ACSAQ. As subcaracterísticas netas (isto é, subcaracterísticas filhas das subcaracterísticas filhas da característica) não devem ser listadas;
- Atributos: título e código dos atributos filhos diretos na ACSAQ. Os atributos filhos das subcaracterísticas filhas da característica não devem ser listados.
- Definição: definição da característica de modo a esclarecer dúvidas com relação ao seu entendimento;
- Peso: todas as características da ACSAQ possuem um peso, que será utilizado no cálculo geral da avaliação de qualidade (de forma objetiva vide Seção 4.4);
- Escore: todas as características da ACSAQ possuem um escore, calculado a partir dos escores das subcaracterísticas e atributos de qualidade filhos (de forma objetiva vide Seção 4.6).

A Figura 4.2. apresenta um exemplo da característica *Usabilidade* presente na Figura 4.1. As subcaracterísticas podem ser entendidas como os nós da ACSAQ que não são características e possuem filhos. Para estas devem ser especificados todos os campos das características, exceto *Modelo de qualidade*.

PLANILHA DE CARACTERÍSTICA
Título: Usabilidade
Código: 1
Modelo de qualidade: ISO/IEC 9126, 1991, International Standard Information technology – Software product evaluation – Quality characteristics and guidelines for their use.
Quantidade de filhos: 3 (três)
Subcaracterísticas: SubcaracterísticaA1 (1.1), SubcaracterísticaA3 (1.3)
Atributos: AtributoA2 (1.2)
Definição: Capacidade do produto de ser entendido, aprendido, usável e atrativo para o usuário, quando utilizado dentro de condições especificadas. É uma característica de qualidade que se preocupa com a facilidade de entendimento, com a facilidade de aprendizado, com a operabilidade, com o nível de comunicação, com a atratividade e a conformidade do produto.
Peso: 4 (quatro)
Escore: 6,5 (calculado através dos filhos)

Figura 4.2 – Planilha de característica

A Figura 4.3 mostra um exemplo de uma especificação descritiva de uma subcaracterística. Este exemplo foi extraído do capítulo 5 desta dissertação. A subcaracterística *Organogramas* foi decomposta em dois atributos de qualidade. O seu escore foi calculado com base nos pesos e escores dos atributos filhos (vide Seção 4.6).

PLANILHA DE SUBCARACTERÍSTICA
Título: Organograma
Código: 2.2.1.3.5
Quantidade de filhos: 2 (dois)
Subcaracterísticas: não há.
Atributos: Organograma global (2.2.1.3.5.1), Organograma departamental (2.2.1.3.5.2)
Definição: Representação gráfica de uma organização indicando os arranjos e as inter-relações de suas unidades constituintes, o limite das atribuições de cada uma delas etc.
Peso: 1 (um)
Escore: 4,5 (calculado através dos filhos)

Figura 4.3 – Planilha de subcaracterística

Já para os atributos de qualidade devem ser especificados os seguintes campos:

- Título: título do atributo de qualidade na ACSAQ;
- Código: código do atributo de qualidade na ACSAQ;
- Definição: definição do atributo de qualidade de modo a esclarecer dúvidas com relação ao seu entendimento;
- Peso: todos os atributos de qualidade da ACSAQ possuem um peso, que será utilizado no cálculo geral da avaliação de qualidade (de forma objetiva vide Seção 4.4);
- Escore: devemos associar um escore de conformidade a todos os atributos de qualidade da ACSAQ (de forma objetiva vide seções 4.5).

Os atributos de qualidade não são decompostos, isto é, não possuem filhos. E são medidos diretamente, ao contrário das características e subcaracterísticas que são medidas de forma indireta (por meio de seus filhos).

A Figura 4.4 apresenta o exemplo da planilha de informação de um atributo de qualidade. Este exemplo também foi extraído do capítulo 5.

PLANILHA DE ATRIBUTO DE QUALIDADE	
Título:	Mapa do site
Código:	1.1.1.1
Definição:	Representação visual do conteúdo de um site. Permite visualizar a hierarquia das páginas.
Peso:	4 (quatro)
Escore:	10 (calculado diretamente)

Figura 4.4 – Planilha de atributo de qualidade

Estas duas primeiras fases da metodologia REQE foram baseadas na fase *Definição e Especificação dos Requisitos de Qualidade* (fase 2) da metodologia *Web-site QEM*. No nosso caso temos que nos preocupar em montar a ACSAQ com atributos de qualidade que

possam ser identificados e avaliados como base apenas no documento de requisitos. Na *Web-site QEM* não há esta preocupação, uma vez que no momento da avaliação tem-se além da aplicação Web operando, todos os artefatos construídos durante as fases do processo de desenvolvimento. Na fase 2 da metodologia *Web-site QEM* são definidas *Planilhas de Referências de Variáveis e Parâmetros* para cada atributo quantificável. A metodologia REQE não trabalha com essas planilhas, definindo um outro domínio de valores para medir os atributos, como veremos na Seção 4.5.

Após termos explicado o que é a *Especificação Descritiva da Árvore de Características, Subcaracterísticas e Atributos de Qualidade*, apresentaremos na próxima seção a terceira fase da metodologia REQE.

4.4 Associação de Pesos aos Nós

- Fase 1 - Representação das Características, Subcaracterísticas e Atributos de Qualidade
 - Passo 1 - Escolha do Domínio da Aplicação e do Perfil do Interessado.
 - Passo 2 - Especificação da Árvore de Características, Subcaracterísticas e Atributos de Qualidade
- Fase 2 - Especificação Descritiva da Árvore de Características, Subcaracterísticas e Atributos de Qualidade
- ❖ **Fase 3 – Associação de Pesos aos Nós**

Esta fase da metodologia REQE tem como objetivo determinar o grau de importância dos nós no contexto do domínio da aplicação. Isto é feito associando pesos aos nós. Primeiramente atribuiremos pesos às características, em seguida às subcaracterísticas e aos atributos de qualidade. Mas esta ordem não precisa ser seguida.

Obviamente que os nós possuem importâncias variadas, onde uns são mais relevantes do que outros, dependendo do domínio da aplicação e do perfil do interessado. Esses pesos devem ser associados pelos avaliadores que estão aplicando a metodologia REQE. Nesta dissertação decidimos adotar o valor 10 (dez) para ser distribuído entre os filhos de cada nó da ACSAQ. No entanto, é possível adotar outro valor. No nosso caso, primeiramente deve-se distribuir 10 (dez) pontos entre as características presentes na árvore. São permitidos apenas números inteiros positivos. Isso vai representar o grau de importância daquela

característica, na qualidade global da aplicação Web. Duas árvores possíveis podem ser vistas na Figuras 4.5. Note que a soma dos pesos de todas as características em qualquer das árvores é 10 (dez). Caso fôssemos distribuir um valor diferente de 10 (dez) a fórmula da Figura 4.12 a ser apresentada na Seção 4.6 teria o número 10 (dez) modificado para o valor escolhido. Isso deve ocorrer, por exemplo, se uma característica possuir mais de 10 (dez) filhos.

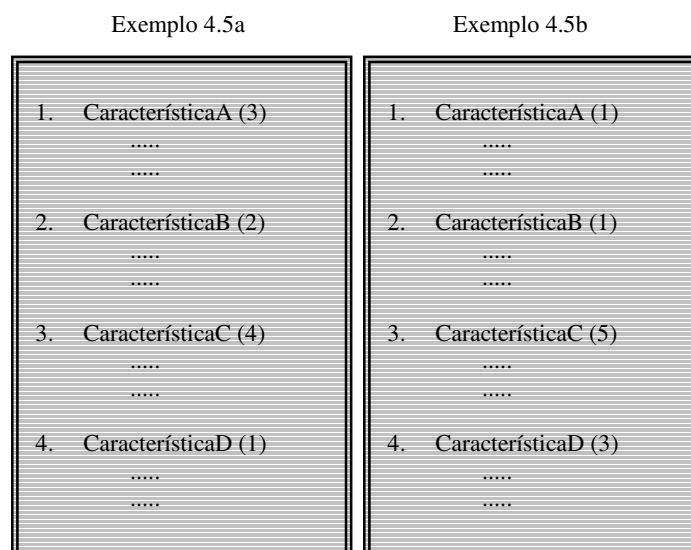


Figura 4.5 – Exemplos de árvores permitidas

Já sabemos que as características são decompostas. O próximo passo é associar pesos aos filhos destas características. Estamos agora identificando quais subcaracterísticas e atributos de qualidade são mais ou menos importantes dentro das suas características (filhos diretos). Se uma característica é decomposta em subcaracterísticas e atributos de qualidade, a soma de todos os pesos dessas subcaracterísticas e atributos deve ser 10 (dez). Lembre-se que estamos nos referindo aos filhos diretos. Vejamos na Figuras 4.6 exemplos de árvores permitidas.

As árvores da Figuras 4.7 não são permitidas pela metodologia REQE. Note que a soma de todos os pesos das subcaracterísticas e atributos de qualidade da *CaracterísticaB* é diferente de 10 (dez) em ambos casos.

Exemplo 4.6a

1. CaracterísticaA (4)
1.1. SubcaracterísticaA1 (3)
1.1.1. AtributoA11
1.1.2. AtributoA12
1.2. SubcaracterísticaA2 (3)
1.2.1. AtributoA21
1.3. SubcaracterísticaA3 (4)
1.3.1. AtributoA31
1.3.2. AtributoA32
1.3.3. AtributoA33
2. CaracterísticaB (2)
2.1. SubcaracterísticaB1 (8)
2.1.1. AtributoB11
2.2. AtributoB2 (2)
3. CaracterísticaC (4)
3.1. AtributoC1 (2)
3.2. AtributoC2 (4)
3.3. SubcaracterísticaC3 (1)
3.3.1. AtributoC31
3.3.2. AtributoC32
3.4. AtributoC4 (3)

Exemplo 4.6b

1. CaracterísticaA (2)
1.1. AtributoA1 (8)
1.2. SubcaracterísticaA2 (1)
1.2.1. AtributoA21
1.2.2. AtributoA22
1.2.3. AtributoA23
1.2.4. AtributoA24
1.3. SubcaracterísticaA3 (1)
1.3.1. AtributoA31
2. CaracterísticaB (2)
2.1. SubcaracterísticaB1 (5)
2.1.1. AtributoB11
2.2. SubcaracterísticaB2 (5)
2.2.1. AtributoB21
3. CaracterísticaC (6)
3.1. SubcaracterísticaC1 (4)
3.1.1. AtributoC11
3.1.2. AtributoC21
3.2. SubcaracterísticaC2 (6)
3.2.1. AtributoC21
3.2.2. AtributoC22
3.2.3. AtributoC23

Figura 4.6 – Exemplos de árvores permitidas

Exemplo 4.7a

1. CaracterísticaA (3)
1.1. AtributoA1 (3)
1.2. SubcaracterísticaA2 (3)
1.2.1. AtributoA21
1.3. SubcaracterísticaA3 (4)
1.3.1. AtributoA31
2. CaracterísticaB (7)
2.1. SubcaracterísticaB1 (2)
2.1.1. AtributoB11
2.1.2. AtributoB12
2.1.3. AtributoB13
2.2. SubcaracterísticaB2 (2)
2.2.1. AtributoB21
2.3. AtributoB3 (4)
2.4. AtributoB4 (1)
2.5. SubcaracterísticaB5(2)
2.5.1. AtributoB51
2.5.2. AtributoB52
2.5.3. AtributoB53

Exemplo 4.7b

1. CaracterísticaA (3)
1.1. AtributoA1 (3)
1.2. SubcaracterísticaA2 (3)
1.2.1. AtributoA21
1.3. SubcaracterísticaA3 (4)
1.3.1. AtributoA31
2. CaracterísticaB (7)
2.1. AtributoB1 (2)
2.2. SubcaracterísticaB2 (2)
2.2.1. AtributoB21
2.2.2. AtributoB22
2.3. AtributoB3 (2)
2.4. AtributoB4 (1)
2.5. AtributoB5(2)

Figura 4.7 – Exemplos de árvores proibidas

Veja que no *Exemplo 4.7a* a soma de todos os pesos dos filhos diretos da *CaracterísticaB* é 11 (onze), enquanto que no *Exemplo 4.7b*, a soma de todos os pesos dos filhos diretos da *CaracterísticaB* é 9 (nove). Nenhum desses casos é permitido pela metodologia REQE.

De forma semelhante iremos trabalhar com os atributos de qualidade. A soma de todos os pesos associados aos atributos de qualidade de uma determinada subcaracterística deve ser igual a 10 (dez). Vejamos exemplos de árvores permitidas e proibidas.

Exemplo 4.8a	Exemplo 4.8b
1. CaracterísticaA (3) 1.1. SubcaracterísticaA1 (3) 1.1.1. AtributoA11 (5) 1.1.2. AtributoA12 (5) 1.2. SubcaracterísticaA2 (3) 1.2.1. AtributoA21 (3) 1.2.2. AtributoA22 (5) 1.2.3. AtributoA23 (2) 1.3. AtributoA3 (4) 2. CaracterísticaB (2) 2.1. SubcaracterísticaB1 (8) 2.1.1. AtributoB11 (10) 2.2. SubcaracterísticaB2 (2) 2.2.1. AtributoB21 (1) 2.2.2. AtributoB22 (2) 2.2.3. AtributoB23 (3) 2.2.4. AtributoB24 (4) 3. CaracterísticaC (4) 3.1. SubcaracterísticaC1 (3) 3.1.1. AtributoC11 (10) 3.2. SubcaracterísticaC2 (7) 3.2.1. AtributoC21 (4) 3.2.2. AtributoC22 (2) 3.2.3. AtributoC23 (4) 4. CaracterísticaD (1) 4.1. SubcaracterísticaD1 (1) 4.1.1. AtributoD11 (10) 4.2. SubcaracterísticaD2 (3) 4.2.1. AtributoD21 (10) 4.3. SubcaracterísticaD3 (2) 4.3.1. AtributoD31 (10) 4.4. SubcaracterísticaD4 (1) 4.4.1. AtributoD41 (5) 4.4.2. AtributoD42 (5) 4.5. SubcaracterísticaD5 (3) 4.5.1. AtributoD51 (3) 4.5.2. AtributoD52 (4) 4.5.3. AtributoD53 (3)	1. CaracterísticaA (3) 1.1. SubcaracterísticaA1 (8) 1.1.1. AtributoA11 (10) 1.2. SubcaracterísticaA2 (1) 1.2.1. AtributoA21 (10) 1.3. SubcaracterísticaA3 (1) 1.3.1. AtributoA31 (2) 1.3.2. AtributoA32 (1) 1.3.3. AtributoA33 (1) 1.3.4. AtributoA34 (2) 1.3.5. AtributoA35 (1) 1.3.6. AtributoA36 (3) 2. CaracterísticaB (2) 2.1. SubcaracterísticaB1 (5) 2.1.1. AtributoB11 (2) 2.1.2. AtributoB12 (8) 2.2. AtributoB2 (5) 3. CaracterísticaC (4) 3.1. SubcaracterísticaC1 (4) 3.1.1. AtributoC11 (5) 3.1.2. AtributoC12 (4) 3.1.3. AtributoC13 (1) 3.2. AtributoC2 (6) 4. CaracterísticaD (1) 4.1. SubcaracterísticaD1 (1) 4.1.1. AtributoD11 (2) 4.1.2. AtributoD12 (2) 4.1.3. AtributoD13 (6) 4.2. SubcaracterísticaD2 (1) 4.2.1. AtributoD21 (5) 4.2.2. AtributoD22 (5) 4.3. SubcaracterísticaD3 (1) 4.3.1. AtributoD31 (3) 4.3.2. AtributoD32 (7) 4.4. SubcaracterísticaD4 (1) 4.4.1. AtributoD41 (10) 4.5. SubcaracterísticaD5 (6) 4.5.1. AtributoD51 (10)

Figura 4.8 – Exemplos de árvores permitidas

Observe que, caso uma subcaracterística seja decomposta em um único atributo de qualidade, o peso deste deve ser obrigatoriamente igual a 10 (dez). Veja que a regra (a soma dos pesos dos atributos filhos de uma subcaracterística deve ser igual a 10) continua sendo válida. Isso ocorreu com os atributos *AtributoB11*, *AtributoC11*, *AtributoD11*, *AtributoD31* no *Exemplo 4.8a* e com os atributos *AtributoA11*, *AtributoA21*, *AtributoD41*, *AtributoD51* no *Exemplo 4.8b*.

Vejamos agora exemplos de árvores proibidas, porque a soma de todos os pesos dos atributos de qualidade de uma subcaracterística não é igual a 10 (dez).

Exemplo 4.9a	Exemplo 4.9b
1. CaracterísticaA (3) 1.1. SubcaracterísticaA1 (3) 1.1.1. AtributoA11 (5) 1.1.2. AtributoA12 (5) 1.2. SubcaracterísticaA2 (3) 1.2.1. AtributoA21 (4) 1.2.2. AtributoA22 (5) 1.2.3. AtributoA23 (2) 1.3. AtributoA3 (4) 2. CaracterísticaB (2) 2.1. SubcaracterísticaB1 (8) 2.1.1. AtributoB11 (10) 2.2. SubcaracterísticaB2 (2) 2.2.1. AtributoB21 (1) 2.2.2. AtributoB22 (2) 2.2.3. AtributoB23 (1) 2.2.4. AtributoB24 (4) 3. CaracterísticaC (4) 3.1. SubcaracterísticaC1 (3) 3.1.1. AtributoC11 (10) 3.2. SubcaracterísticaC2 (7) 3.2.1. AtributoC22 (6) 3.2.2. AtributoC23 (4) 4. CaracterísticaD (1) 4.1. SubcaracterísticaD1 (1) 4.1.1. AtributoD11 (10) 4.2. SubcaracterísticaD2 (3) 4.2.1. AtributoD21 (10) 4.3. SubcaracterísticaD3 (2) 4.3.1. AtributoD31 (10) 4.4. SubcaracterísticaD4 (4) 4.4.1. AtributoD41 (5) 4.4.2. AtributoD43 (2) 4.4.3. AtributoD44 (3)	1. CaracterísticaA (3) 1.1. SubcaracterísticaA1 (8) 1.1.1. AtributoA11 (10) 1.2. SubcaracterísticaA2 (1) 1.2.1. AtributoA21 (9) 1.3. SubcaracterísticaA3 (1) 1.3.1. AtributoA31 (2) 1.3.2. AtributoA32 (1) 1.3.3. AtributoA33 (1) 1.3.4. AtributoA34 (2) 1.3.5. AtributoA35 (1) 1.3.6. AtributoA36 (3) 2. CaracterísticaB (2) 2.1. SubcaracterísticaB1 (5) 2.1.1. AtributoB11 (2) 2.1.2. AtributoB12 (8) 2.2. Subcaracterística B (5) 3. CaracterísticaC (4) 3.1. SubcaracterísticaC1 (4) 3.1.1. AtributoC11 (5) 3.1.2. AtributoC12 (5) 3.2. AtributoC2 (6) 4. CaracterísticaD (1) 4.1. SubcaracterísticaD1 (1) 4.1.1. AtributoD11 (2) 4.1.2. AtributoD13 (4) 4.1.3. AtributoD14 (4) 4.2. SubcaracterísticaD2 (1) 4.2.1. AtributoD21 (5) 4.2.2. AtributoD22 (5) 4.3. SubcaracterísticaD3 (1) 4.3.1. AtributoD31 (3) 4.3.2. AtributoD32 (4) 4.3.3. AtributoD33 (3)

Figura 4.9 – Exemplos de árvores proibidas

Note que no *Exemplo 4.9a*, a soma de todos os pesos dos atributos da *SubcaracterísticaA2* é 11 (onze) e a soma de todos os pesos dos atributos da *SubcaracterísticaB2* é 8 (oito). Enquanto que no *Exemplo 4.9b*, o *AtributoA21*, filho único da *SubcaracterísticaA2* deveria ter peso igual a 10 (dez).

Lembre-se que ao associar a cada característica, subcaracterística ou atributo de qualidade um peso entre 0 (zero) e 10 (dez) estamos quantificando a sua importância. Quanto mais importante for para a qualidade do sistema, maior deverá ser seu peso. Cabe ressaltar que atualmente estamos estudando métodos científicos capazes de associar pesos aos atributos de uma forma objetiva [HUI03] [KAIYA02].

A fase de *Definição e Implementação da Avaliação Global* (fase 4) da metodologia *Web-site QEM* originou as fases 3 e 5 da metodologia REQE. Na *Web-site QEM* não existe uma simples associação de pesos aos atributos. Devem ser utilizadas funções lógicas de agregação para modelar diferentes relações entre atributos e características de qualidade, entre elas simultaneidade, neutralidade, simetrias, e diferentes níveis de polarização “AND/OR”. O cálculo final é efetuado com o auxílio de modelos como o *Logic Scoring of Preference (LSP)* [DUJMOVIC82] [DUJMOVIC96] fundamentado na lógica de preferência contínua, diferentemente da regra criada pela metodologia REQE, a ser apresentada na Seção 4.6.

Concluída esta fase de *Associação de Pesos aos Nós* já podemos passar para a quarta atividade da metodologia REQE.

4.5 Associação de Escores aos Atributos

- Fase 1 - Representação das Características, Subcaracterísticas e Atributos de Qualidade
 - Passo 1 - Escolha do Domínio da Aplicação e do Perfil do Interessado.
 - Passo 2 - Especificação da Árvore de Características, Subcaracterísticas e Atributos de Qualidade
- Fase 2 - Especificação Descritiva da Árvore de Características, Subcaracterísticas e Atributos de Qualidade
- Fase 3 – Associação de Pesos aos Nós
- ❖ **Fase 4 – Associação de Escores aos Atributos**

Ao iniciar esta etapa já contamos com a árvore de características, subcaracterísticas e atributos de qualidade definida e estruturada. Também temos as planilhas de informações preenchidas e os pesos associados a todos os nós da árvore (que identificam a importância relativa de cada item). Nos concentraremos agora em atribuir escores, por meio de uma escala de valores, a cada um dos atributos de qualidade. Este trabalho propõe a seguinte escala de conformidade, baseado em [ALVES01] e [ALVES02]:

<i>Conformidade</i>	<i>Escore</i>
Não atende o atributo de qualidade	0
Atende com sérias restrições	3
Atende parcialmente	6,5
Atende	10

Figura 4.10 – Tabela de conformidade x escore

Não haveria problema em adotar outra escala de valores. Estes escore de conformidade serão associados aos atributos da árvore de características, subcaracterísticas e atributos de qualidade, levando em consideração se o documento de requisitos especifica determinado atributo. Consideramos o número 6,5 por ser um valor intermediário no intervalo [3,10].

Para cada atributo de qualidade da ACSAQ deve ser associado um escore, seguindo a escala da Figura 4.10. Caso o atributo não seja previsto no documento de requisitos, deve receber a conformidade *não atende* (escore 0), pois como o documento não prevê, consideramos que não atende o atributo de qualidade, conseqüentemente, o produto final dificilmente atenderá. Nesta situação, o atributo ausente deverá ser anotado na seção de recomendações do documento de requisitos, para que seja analisado posteriormente a razão da sua omissão. Note que já estamos contribuindo com uma melhoria na qualidade da aplicação no momento em que identificamos prováveis falhas.

O documento de requisitos pode ainda atender determinado atributo de forma bastante precária. Diferencia-se da condição anterior (*não atende*) pelo fato do atributo haver sido ao menos citado ou referenciado no documento de requisitos. Possibilita, então, que o analista

tente esclarecer o requisito que não entendeu. Neste caso deve receber o escore 3 (três), correspondente ao caso de atender com sérias restrições.

Um terceiro tipo ocorre quando o documento de requisitos especifica parcialmente o atributo de qualidade. É como se faltasse alguma informação necessária para uma especificação completa e consistente do atributo. Em virtude desta omissão, possivelmente haverá algum problema na implantação deste requisito no contexto do projeto. O atributo de qualidade foi previsto, mas a maneira como foi exposto atende apenas parcialmente, devendo receber então o escore 6,5 (seis e meio).

Por último temos o caso em que o atributo é perfeitamente previsto e especificado no documento de requisitos, justificando a sua presença quando da existência do produto final. A esses casos deve ser associado o escore 10 (dez).

Essa associação de escores representa o grau de cumprimento dos atributos de qualidade com relação ao documento de requisitos que representa as definições do futuro produto. Quanto melhor especificado e detalhado, maiores as chances de abordar e especificar cada atributo de forma completa e consistente.

As notas das subcaracterísticas serão calculadas através das notas (neste caso, dos escores de conformidade) dos respectivos nós filhos (subcaracterísticas e atributos filhos), associados a seus pesos. De modo análogo serão calculadas as notas das características, por meio das notas dos respectivos nós filhos (subcaracterísticas e atributos filhos), associados a seus pesos. Veremos em detalhes como isso ocorre na próxima seção. Nesta fase, somente serão associados escores aos atributos da ACSAQ, isto é, não serão associados escores às características e subcaracterísticas. Estas terão os seus escores calculados, na próxima fase, por intermédio de seus nós filhos. Na árvore da Figura 4.11, apenas os nós em **negrito** terão escores diretamente associados, os demais serão calculados posteriormente e com base em regras que serão definidas na próxima seção.

Na fase de *Definição e Implementação da Avaliação Elementar* (fase 3) da metodologia *Web-site QEM* são definidos critérios de avaliação e por meio destes os atributos de qualidade são medidos. Para cada variável medida X_i , $i = 1, \dots, n$ se define uma função que representa o critério elementar. Esta função é uma correspondência (mapeamento) dos valores computados a partir do domínio empírico no novo domínio numérico, e foi denominada de preferência de qualidade elementar (IE_i). Pode-se entender IE_i como a porcentagem do requisito satisfeito para A_i . Na metodologia REQE transformamos esta atividade na associação de escores de conformidade aos atributos de qualidade, com base na Figura 4.10. Passaremos então para a última fase da metodologia REQE.

4.6 Cálculo Geral

- Fase 1 - Representação das Características, Subcaracterísticas e Atributos de Qualidade
 - Passo 1 - Escolha do Domínio da Aplicação e do Perfil do Interessado.
 - Passo 2 - Especificação da Árvore de Características, Subcaracterísticas e Atributos de Qualidade
- Fase 2 - Especificação Descritiva da Árvore de Características, Subcaracterísticas e Atributos de Qualidade
- Fase 3 – Associação de Pesos aos Nós
- Fase 4 – Associação de Escores aos Atributos
- ❖ **Fase 5 – Cálculo Geral**

O objetivo desta fase é efetuar o cálculo que consolida todos os escores de conformidade associados, levando em consideração os pesos dos nós. Ao final desta fase teremos a avaliação de qualidade concluída e a possibilidade de corrigir os erros identificados (atributos de qualidade mal avaliados, isto é mal especificados). Neste estágio do processo, todos os atributos estão com seus escores associados. A nota de cada nó-pai é calculada através dos escores e dos pesos dos nós filhos.

$$Nota-Nó-Pai = \left(\sum_{i=1}^n (peso_i * score_i) \right) / 10$$

Onde i varia do primeiro ao último filho.

Figura 4.12 – Fórmula para cálculo da nota de nó pai.

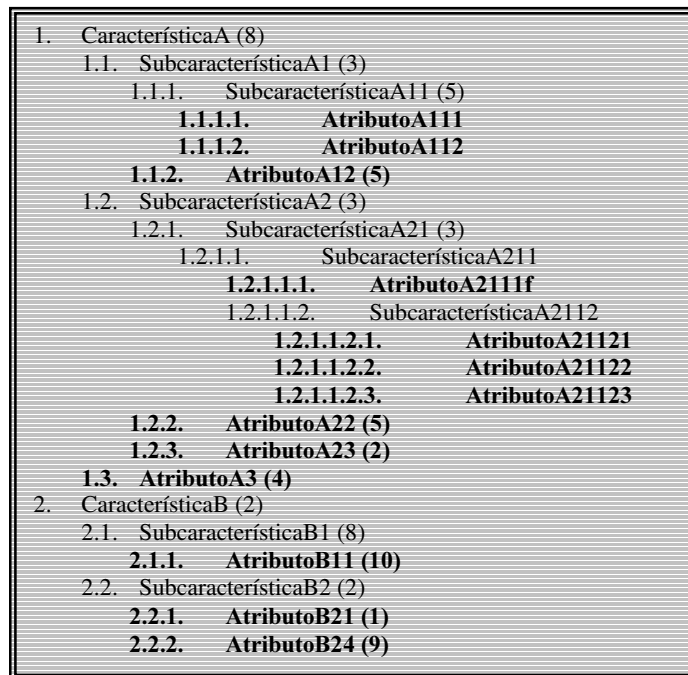


Figura 4.11 – Exemplo de uma ACSAQ com os atributos destacados.

Veja que o peso de cada atributo filho é multiplicado pelo seu escore, todos esses valores são adicionados, e a soma deve ser dividida por 10 (dez), resultando na nota do pai. Esta divisão por 10 (dez) é uma espécie de normalização, uma vez que cada nó teve o valor 10 (dez) dividido entre os pesos de seus nós filhos. Vejamos um exemplo na Figura 4.13.

Note que a nota da subcaracterística SubcaracterísticaA1 foi calculada com base nos pesos e escores de conformidade de seus filhos AtributoA11 e AtributoA12. A nota da subcaracterística SubcaracterísticaB1 também foi calculada através dos pesos e escores de conformidade de seus filhos. Já a nota da característica CaracterísticaB será calculada por meio dos pesos e das notas das subcaracterísticas SubcaracterísticaB1 e SubcaracterísticaB2. E a nota da característica CaracterísticaA será calculada através dos pesos e das notas das subcaracterísticas SubcaracterísticaA1, SubcaracterísticaA2 e do atributo de qualidade AtributoA3.

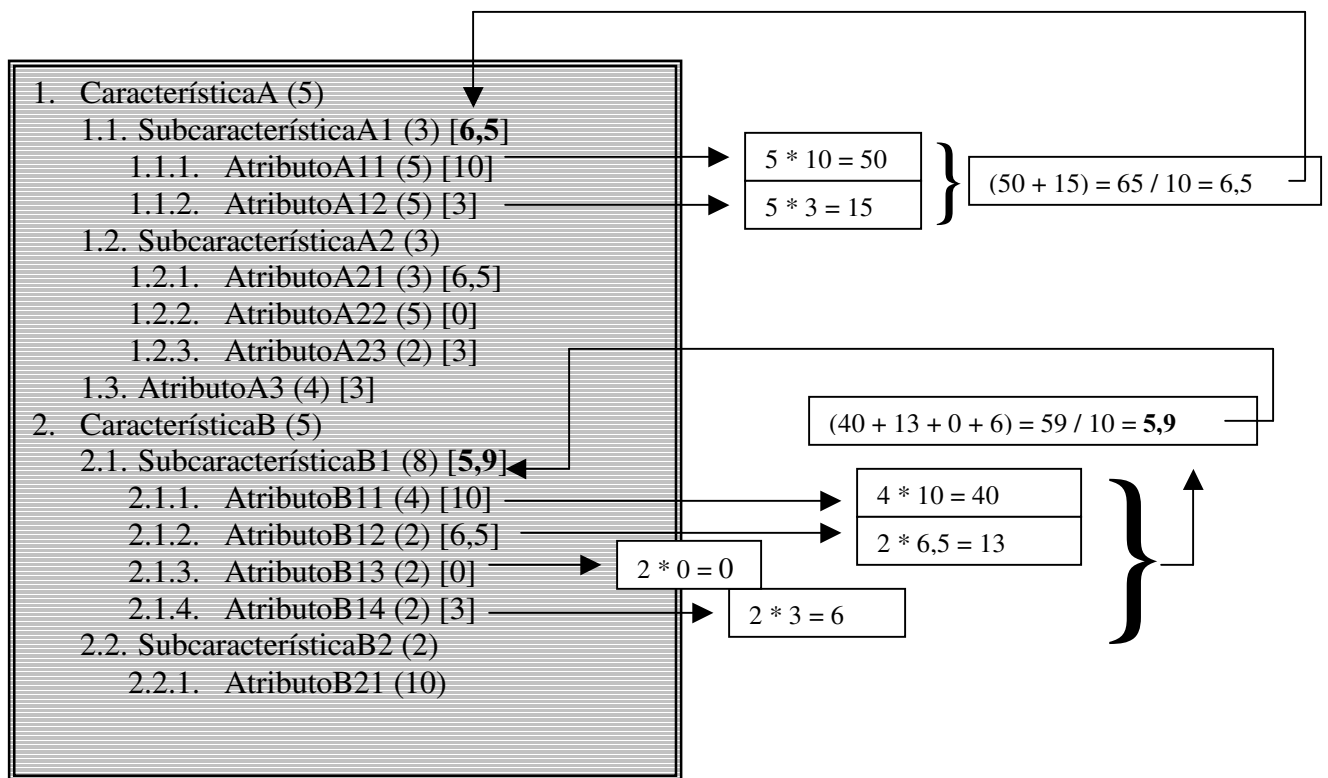


Figura 4.13 – Exemplo de cálculo geral

Sempre que subimos um nível na árvore, para calcularmos a nota de um nó-pai, dividimos a soma das notas dos filhos por 10 (dez). Isso ocorre para que seja mantida a consistência no cálculo da nota do nó avô (nó pai do nó pai). Imagine a seguinte situação:

Como a subcaracterística *SubcaracterísticaA11* teve a sua nota calculada pelos filhos e não houve a divisão por 10 (dez), a subcaracterística *SubcaracterísticaA1* vai ser calculada com base nos seus dois filhos, um com nota 100 e outro com nota 10. Note que todos os atributos medidos abaixo desta subcaracterística, a saber atributos *AtributoA111*, *AtributoA112* e *AtributoA12* foram associados aos escores máximos, logo a sua nota também deveria ser a máxima, isto é 10 (dez). Continuando com o cálculo, chegamos ao absurdo valor de 550. Caso a divisão tivesse sido efetuada teríamos o cálculo correto.

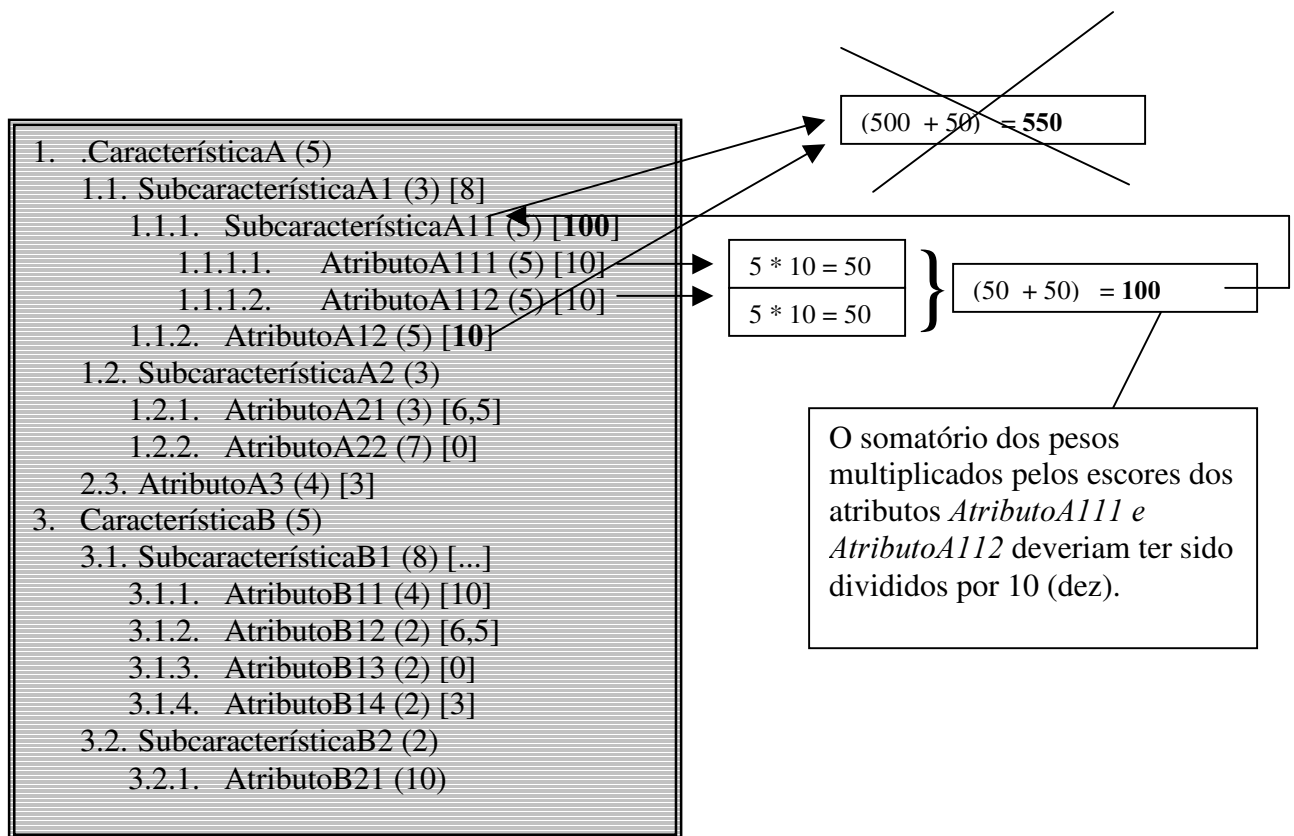


Figura 4.14 – Exemplo incorreto de cálculo geral

Concluindo, sempre que subimos um nível, isto é, sempre que calculamos a nota de um nó através de seus filhos, não podemos esquecer de dividir a soma dos pesos multiplicados pelos os escores (ou notas) dos filhos por 10 (dez).

Cada nota deve possuir apenas duas casas decimais. Usamos o arredondamento para cima se a terceira casa decimal for maior ou igual a cinco e o arredondamento para baixo se for menor do que cinco.

De forma similar devem ser calculadas as demais características e subcaracterísticas (lembre-se que os atributos já estão com os escores associados) da ACSAQ, resultando em uma nota única (calculada através das notas das características de mais alto nível), que é a nota do documento de requisitos que representa o produto final. Vale salientar que possuímos esta nota antes mesmo da existência do produto. É claro que essa avaliação também tem o objetivo de garantir que um produto de melhor qualidade seja desenvolvido.

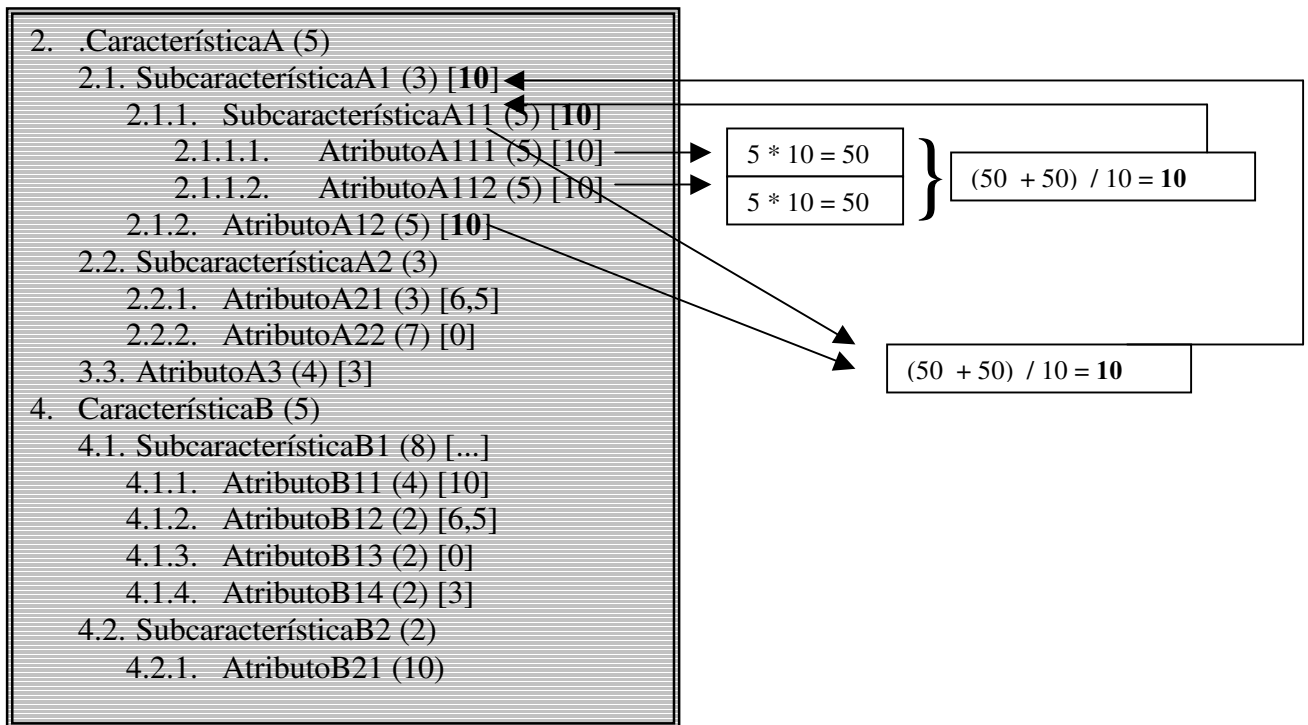


Figura 4.15 – Exemplo de cálculo geral

Acreditamos que com o uso da metodologia REQE podemos incrementar a qualidade de aplicações Web, uma vez que falhas podem ser identificadas numa fase inicial do processo de desenvolvimento. Por outro lado a metodologia REQE é bastante trabalhosa, e estamos estudando meios de automatizar algumas fases ou passos, de forma que não precisem ser feitos manualmente.

Mesmo lembrando que “o uso de métricas de software não elimina a necessidade do juízo humano na avaliação de software” [IEEE1061], sabemos que devemos nos esforçar para atribuir os escores de conformidade aos atributos de uma forma mais objetiva.

Veremos no próximo capítulo um exemplo da aplicação da metodologia REQE.

5 Exemplo

Neste capítulo aplicamos a metodologia REQE para avaliar a qualidade de um site acadêmico.

5.1 Introdução

Este capítulo apresenta um exemplo de utilização da metodologia REQE (Requirements Engineering Quality Evaluation). Avaliaremos a qualidade de um site acadêmico por meio de seu documento de requisitos. Consultando alguns *stakeholders* especificamos o documento de requisitos presente no Apêndice B desta dissertação. É baseado neste documento que iremos efetuar a avaliação de qualidade. Esta avaliação foi realizada seguindo passo a passo as fases da metodologia REQE. As próximas seções descrevem como foram executadas as atividades relativas a cada uma das fases da metodologia.

Antes de aplicarmos a metodologia REQE é importante apresentarmos algumas diretrizes que devem ser levadas em consideração durante o processo de avaliação:

- **Diretriz 1:** a metodologia deve ser aplicada por uma equipe de avaliadores idealmente com diversos perfis (usuário da aplicação, analista de requisitos, desenvolvedor, gerente, especialista no domínio etc);
- **Diretriz 2:** os avaliadores que irão especificar a árvore de características, subcaracterísticas e atributos de qualidade (ACSAQ) não devem ter acesso ao documento de requisitos da aplicação Web a ser avaliada antes de finalizar a especificação da ACSAQ;
- **Diretriz 3:** um representante dos usuários clientes da aplicação Web devem participar da associação de pesos aos nós (fase 3);
- **Diretriz 4:** os avaliadores que irão associar escores de conformidade aos atributos de qualidade (fase 4) devem ter conhecimento da técnica utilizada para modelar os requisitos (por exemplo, LAL, ACE, KAOS, i*). Caso os avaliadores não tenham o conhecimento necessário para entender os requisitos da forma como estão modelados/estruturados no documento, certamente a associação dos escores não será coerente, comprometendo a qualidade da avaliação;
- **Diretriz 5:** ao associar escores de conformidade aos atributos de qualidade (fase 4) é conveniente fazer uma busca no documento de requisitos com as palavras-chave do atributo em questão. Esta heurística tem o objetivo de facilitar a localização da especificação do atributo de qualidade no documento de requisitos.

A título de revisão apresentamos novamente as fases e os passos da metodologia REQE.

- Fase 1 - Representação das Características, Subcaracterísticas e Atributos de Qualidade
 - Passo 1 - Escolha do Domínio da Aplicação e do Perfil do Interessado.
 - Passo 2 - Especificação da Árvore de Características, Subcaracterísticas e Atributos de Qualidade
- Fase 2 - Especificação Descritiva da Árvore de Características, Subcaracterísticas e Atributos de Qualidade
- Fase 3 – Associação de Pesos aos Nós
- Fase 4 – Associação de Escores aos Atributos
- Fase 5 – Cálculo Geral

5.2 Representação das Características, Subcaracterísticas e Atributos de Qualidade

Começaremos com o passo de *Representação das Características, Subcaracterísticas e Atributos de Qualidade*. Já temos o domínio da aplicação, mas precisamos definir qual o perfil do usuário interessado, uma vez que dependendo dessa escolha iremos abordar as características de qualidade de forma diferente.

- Domínio da aplicação: Site acadêmico.
- Perfil do Interessado: Estudante, visitante intencional.

Devemos levar em consideração a visão de um estudante com o perfil de visitante intencional, que observa o site de forma dinâmica, em geral em pouco tempo e em busca de uma determinada informação. Além de utilizar os serviços oferecidos (funcionalidade), este visitante está preocupado com a confiabilidade e a performance (ou eficiência), como também o quão fácil será utilizar os serviços oferecidos (usabilidade). Por meio de questionários orientados a estudantes, discussões entre partes interessadas, e avaliadores ficou definido que características de manutenibilidade e portabilidade não são relevantes para esse tipo de audiência [OLSINA99]. A título de simplificação, neste exemplo, nos concentraremos apenas nas características de usabilidade e funcionalidade.

Com base na consolidação dos Modelos de qualidade dos padrões ISO [ISO9126] e IEEE [IEEE1061], no domínio da aplicação, no perfil do interessado, e no trabalho de [OLSINA99] definimos a árvore de características, subcaracterísticas e atributos de qualidade.

A característica *Usabilidade* foi dividida em quatro subcaracterísticas: *Entendimento global do site*, *Ajuda ao usuário*, *Interface e estética* e *Tarefas Diversas*. Como mostra a Figura 5.1, cada uma dessas subcaracterísticas foi dividida em outras subcaracterísticas e atributos de qualidade. Por exemplo, a subcaracterística *Entendimento global do site* foi dividida nas subcaracterísticas *Esquema de organização global* e *Tarefas de navegação*, e nos atributos de qualidade *Visita guiada para estudantes* e *Mapa de imagens*.

Já a característica *Funcionalidade* foi decomposta em apenas duas subcaracterísticas: *Tarefas de busca* e *Tarefas do domínio relativo ao estudante*. A subcaracterística *Tarefas de busca* foi decomposta em uma única subcaracterísticas: *Mecanismos de busca*. Por sua vez, a subcaracterística *Tarefas do domínio relativo ao estudante* foi decomposta em duas outras subcaracterísticas: *Conteúdo relevante* e *Serviços on-line*. Apresentamos a árvore de características, subcaracterísticas e atributos de qualidade na Figura 5.1.

Tendo identificado o domínio da aplicação, selecionado o perfil do interessado (Fase 1 – Passo 1) e especificado a árvore de características, subcaracterísticas e atributos de qualidade (Fase 1 – Passo 2), passaremos a descrever por meio do preenchimento de planilhas de informações todos os nós da ACSAQ.

1. Usabilidade

- 1.1 Entendimento global do site
 - 1.1.1 Esquema de organização global
 - 1.1.1.1 Mapa do site
 - 1.1.1.2 Tabela de conteúdos
 - 1.1.1.3 Índice alfabético
 - 1.1.1.4 Localizador no mapa
 - 1.1.2 Tarefas de navegação
 - 1.1.2.1 Navegabilidade
 - 1.1.2.1.1 Orientação
 - 1.1.2.1.1.1 Indicador do caminho
 - 1.1.2.1.1.2 Marcador da posição atual
 - 1.1.2.2 Objetos de controle de navegação
 - 1.1.2.2.1 Apresentação dos controles contextuais
 - 1.1.2.2.1.1 Permanência dos controles
 - 1.1.2.2.1.2 Intuitividade dos controles
 - 1.1.2.2.2 Mecanismos de rolagem
 - 1.1.2.2.2.1 Vertical
 - 1.1.2.2.2.2 Horizontal
 - 1.1.2.3 Intuitividade da navegação
 - 1.1.2.3.1 Título do link
 - 1.1.2.3.2 Qualidade da frase do link
 - 1.1.3 Visita guiada para estudantes
 - 1.1.4 Mapa de imagens
- 1.2 Ajuda ao usuário
 - 1.2.1 Qualidade da ajuda
 - 1.2.1.1 Ajuda geral direcionada para o estudante
 - 1.2.1.2 Ajuda para busca
 - 1.2.2 Indicador de atualização
 - 1.2.2.1 Global
 - 1.2.2.2 Por subsite ou página
 - 1.2.3 Acesso aos endereços
 - 1.2.3.1 Endereço eletrônico
 - 1.2.3.2 Telefone/fax
 - 1.2.3.3 Ramais internos
 - 1.2.3.4 Endereço para correspondência
 - 1.2.4 FAQ
 - 1.2.5 Formulários de retorno
 - 1.2.5.1 Questionários
 - 1.2.5.2 Livro de visitas
 - 1.2.5.3 Comentários
- 1.3 Interface e estética
 - 1.3.1 Coesão no agrupamento de objetos de controle

Figura 5.1 – ACSAQ do exemplo de sites acadêmicos, adaptada de [OLSINA99]

- 1.3.2 Apresentação dos controles principais
 - 1.3.2.1 Intuitividade dos controles
 - 1.3.2.2 Permanência dos controles
- 1.3.3 Estilo
 - 1.3.3.1 Uniformidade no estilo de cores dos links
 - 1.3.3.2 Uniformidade no estilo global
 - 1.3.3.3 Guia do estilo global
- 1.4 Tarefas diversas
 - 1.4.1 Suporte a outros idiomas
 - 1.4.2 Mecanismos para destacar novidade
 - 1.4.3 Indicação da resolução do monitor
- 2. Funcionalidade
 - 2.1 Tarefas de busca
 - 2.1.1 Mecanismos de busca
 - 2.1.1.1 Busca por escopo
 - 2.1.1.1.1 Busca de pessoas
 - 2.1.1.1.2 Busca por cursos
 - 2.1.1.1.3 Busca por disciplinas acadêmicas
 - 2.1.1.2 Busca global
 - 2.1.1.2.1 Informações para refinamento
 - 2.2 Tarefas do domínio relativo ao estudante
 - 2.2.1 Conteúdo relevante
 - 2.2.1.1 Informações sobre unidade acadêmicas
 - 2.2.1.1.1 Índice de disciplinas acadêmicas
 - 2.2.1.1.2 Subsite de disciplinas acadêmicas
 - 2.2.1.2 Informações diversas
 - 2.2.1.2.1 Requerimentos
 - 2.2.1.2.1.1 Entrada
 - 2.2.1.2.1.2 Acompanhamento
 - 2.2.1.2.1.3 Download de formulários
 - 2.2.1.3 Qualidade do conteúdo
 - 2.2.1.3.1 Qualidade do índice
 - 2.2.1.3.2 Qualidade do conteúdo
 - 2.2.1.3.3 Qualidade das informações dos cursos
 - 2.2.1.3.3.1 Ementas
 - 2.2.1.3.3.2 Formas de avaliação
 - 2.2.1.3.4 Descrição dos cursos
 - 2.2.1.3.4.1 Comentários
 - 2.2.1.3.4.2 Plano de ensino
 - 2.2.1.3.4.3 Calendário
 - 2.2.1.3.4.3.1 Das aulas
 - 2.2.1.3.4.3.2 Das provas
 - 2.2.1.3.4.3.3 Dos trabalhos
 - 2.2.1.3.4.3.4 Diversos

Continuação da Figura 5.1

- 2.2.1.3.4.4 Cursos de pós-graduação
 - 2.2.1.3.4.4.1 Especialização
 - 2.2.1.3.4.4.2 Mestrado
 - 2.2.1.3.4.4.3 Doutorado
 - 2.2.1.3.5 Organogramas
 - 2.2.1.3.5.1 Organograma global
 - 2.2.1.3.5.2 Organograma departamental
 - 2.2.1.4 Informação dos serviços dos estudantes
 - 2.2.1.4.1 Índice de serviços
 - 2.2.1.4.2 Informações do calendário escolar
 - 2.2.1.4.3 Informações dos alojamentos
 - 2.2.1.4.4 Informações culturais
 - 2.2.1.4.5 Informações esportivas
 - 2.2.1.4.6 Informações sobre congressos
 - 2.2.1.5 Informações da infra-estrutura acadêmica
 - 2.2.1.5.1 Bibliotecas
 - 2.2.1.5.1.1 Busca de exemplares
 - 2.2.1.5.1.2 Reserva de exemplares
 - 2.2.1.5.1.3 Informações gerais
 - 2.2.1.5.2 Laboratórios
 - 2.2.1.5.2.1 Reserva
 - 2.2.1.5.2.2 Informações gerais
 - 2.2.1.5.2.3 Resultados de pesquisa
 - 2.2.2 Serviços on-line
 - 2.2.2.1 Informação da grade curricular
 - 2.2.2.2 Serviços Web
 - 2.2.2.3 Serviços FTP
 - 2.2.2.4 Serviços de newsgroup

Continuação da Figura 5.1

5.3 Especificação Descritiva da Árvore de características, subcaracterísticas e atributos de Qualidade

Neste trabalho não apresentaremos a planilha de todas as subcaracterísticas e atributos de qualidade, devido ao enorme volume dessa documentação. O objetivo deste capítulo é através de exemplos ajudar a entender a metodologia REQE [REIS02], como também validar a proposta. Acreditamos que os exemplos apresentados são suficientes para este entendimento.

Inicialmente descreveremos as duas características identificadas, ou seja, *Usabilidade* e *Funcionalidade*.

PLANILHA DE CARACTERÍSTICA
Título: Usabilidade
Código: 1
Modelo de qualidade: ISO/IEC 9126, 1991, International Standard Information technology – Software product evaluation – Quality characteristics and guidelines for their use.
Quantidade de filhos: 4 (quatro)
Subcaracterísticas: Entendimento global do site (1.1), Ajuda ao usuário (1.2), Interface e estética (1.3), Tarefas diversas (1.4)
Atributos: Não há.
Definição: Capacidade do produto de ser entendido, aprendido, usável e atrativo para o usuário, quando utilizado dentro de condições especificadas. É uma característica de qualidade que se preocupa com a facilidade de entendimento, com a facilidade de aprendizado, com a operabilidade, com o nível de comunicação, com a atratividade e a conformidade do produto.
Peso: 5 (cinco)
Escore: a ser calculado posteriormente

Figura 5.2 – Planilha preenchida da característica *Usabilidade*

Apesar de não ser objeto desta fase (e sim da fase 3), decidimos adiantar que as duas características contempladas no exemplo deste capítulo possuem o mesmo grau de importância na avaliação de qualidade de um site acadêmico. Assim, atribuímos o peso 5 (cinco) a ambas as características.

PLANILHA DE CARACTERÍSTICA
Título: Funcionalidade
Código: 2
Modelo de qualidade: ISO/IEC 9126, 1991, International Standard Information technology – Software product evaluation – Quality characteristics and guidelines for their use.
Quantidade de filhos: 2 (dois)
Subcaracterísticas: Tarefas de busca (2.1), Tarefas do domínio relativo ao estudante (2.2)
Atributos: Não há.
Definição: Capacidade do produto de prover as funções definidas nas fases de desenvolvimento, relativas às necessidades explícitas e implícitas dos usuários.
Peso: 5 (cinco)
Escore: a ser calculado posteriormente

Figura 5.3 – Planilha preenchida da característica *Funcionalidade*

Passaremos então, a descrever as seguintes subcaracterísticas: *Entendimento global do site* (Figura 5.4), *Ajuda ao usuário* (Figura 5.5), *Indicador de atualização* (Figura 5.6) e *Organogramas* (Figura 5.7). Os pesos destas subcaracterísticas serão associados na fase 3 (*Associação de Pesos aos Nós*). Já os escores, calculados com base nos escores dos nós filhos, serão calculados apenas na fase 5 (*Cálculo Geral*).

PLANILHA DE SUBCARACTERÍSTICA
Título: Entendimento global do site
Código: 1.1
Quantidade de filhos: 4 (quatro)
Subcaracterísticas: Esquema de organização global (1.1.1), Tarefas de navegação (1.1.2)
Atributos: Visita guiada para estudantes (1.1.3), Mapa de imagens (1.1.4)
Definição: Retrata a apresentação geral do site com relação às facilidades existentes para o seu uso e manuseio. Quanto maior for o seu entendimento, mais fácil será: achar uma informação procurada, localizar-se dentro da estrutura das páginas, visitar todas as páginas de forma organizada etc
Peso: a ser calculado posteriormente (fase 3)
Escore: a ser calculado posteriormente (fase 5)

Figura 5.4 – Planilha preenchida da subcaracterística *Entendimento global do site*

PLANILHA DE SUBCARACTERÍSTICA
Título: Ajuda ao usuário
Código: 1.2
Quantidade de filhos: 5 (cinco)
Subcaracterísticas: Qualidade da ajuda (1.2.1), Indicador de atualização (1.2.2), Acesso aos endereços (1.2.3) e Formulários de retorno (1.2.5)
Atributos: FAQ (1.2.4)
Definição: formas de auxiliar o usuário a utilizar os serviços da aplicação.
Peso: a ser calculado posteriormente (fase 3)
Escore: a ser calculado posteriormente (fase 5)

Figura 5.5– Planilha preenchida da subcaracterística *Ajuda ao usuário*

PLANILHA DE SUBCARACTERÍSTICA
Título: Indicador de Atualização
Código: 1.2.2
Quantidade de filhos: 2 (dois)
Subcaracterísticas: não há.
Atributos: Global (1.2.2.1), Por subsite ou página (1.2.2.2)
Definição: Indicador da data e hora da última atualização do site.
Peso: a ser calculado posteriormente (fase 3)
Escore: a ser calculado posteriormente (fase 5)

Figura 5.6– Planilha preenchida da subcaracterística *Indicador de atualização*

Para finalizar a fase 2, apresentaremos as planilhas dos seguintes atributos: *Mapa do Site* (Figura 5.8), *FAQ* (Figura 5.9), *Entrada* (Figura 5.10), *Plano de ensino* (Figura 5.11) e *Das provas* (Figura 5.12). Os pesos destes atributos também serão associados na fase 3 (*Associação de Pesos aos Nós*). Já os seus respectivos escores serão associados na fase 4 (*Associação de Escores aos Atributos*).

PLANILHA DE SUBCARACTERÍSTICA
Título: Organograma
Código: 2.2.1.3.5
Quantidade de filhos: 2 (dois)
Subcaracterísticas: não há.
Atributos: Organograma global (2.2.1.3.5.1), Organograma departamental (2.2.1.3.5.2)
Definição: Representação gráfica de uma organização indicando os arranjos e as inter-relações de suas unidades constituintes, o limite das atribuições de cada uma delas etc.
Peso: a ser calculado posteriormente (fase 3)
Escore: a ser calculado posteriormente (fase 5)

Figura 5.7– Planilha preenchida da subcaracterística *Organogramas*

PLANILHA DE ATRIBUTO DE QUALIDADE
Título: Mapa do site
Código: 1.1.1.1
Definição: Representação visual do conteúdo de um site. Permite visualizar a hierarquia das páginas.
Peso: a ser calculado posteriormente (fase 3)
Escore: a ser calculado posteriormente (fase 4)

Figura 5.8 – Planilha preenchida do atributo *Mapa do Site*

PLANILHA DE ATRIBUTO DE QUALIDADE
Título: FAQ
Código: 1.2.4
Definição: Resposta das perguntas freqüentes. Listagem de perguntas acompanhadas das devidas respostas. Evita que o usuário entre em contato com responsáveis para obter determinadas informações bastante comuns, por exemplo, vide http://www.bcb.gov.br/pre/bc_atende/port/crc.asp?idpai=FAQCIDADAO&id=SCRFAQCID
Peso: a ser calculado posteriormente (fase 3)
Escore: a ser calculado posteriormente (fase 4)

Figura 5.9 – Planilha preenchida do atributo *FAQ*

PLANILHA DE ATRIBUTO DE QUALIDADE
Título: Entrada
Código: 2.2.1.2.1.1
Definição: Permite a entrada de requerimentos diversos em todas as secretarias. Os documentos necessários para a entrada de requerimentos devem poder ser entregues de forma eletrônica ou via postal. Como exemplo vide http://www.unicamp.br/cemeq/atendimento.html
Peso: a ser calculado posteriormente (fase 3)
Escore: a ser calculado posteriormente (fase 4)

Figura 5.10 – Planilha preenchida do atributo *Entrada*

PLANILHA DE ATRIBUTO DE QUALIDADE
Título: Plano de ensino
Código: 2.2.1.3.4.2
Definição: Documento contendo os objetivos da disciplina, o conteúdo programático, forma de avaliação, bibliografia etc. Seria interessante que fosse definido um padrão juntamente com orientações para o seu preenchimento. Como exemplo vide http://www.ufrgs.br/farmacia/disciplinas/cbs01103/plano.html
Peso: a ser calculado posteriormente (fase 3)
Escore: a ser calculado posteriormente (fase 4)

Figura 5.11 – Planilha preenchida do atributo *Plano de ensino*

PLANILHA DE ATRIBUTO DE QUALIDADE
Título: Das provas
Código: 2.2.1.3.4.3.3
Definição: Divulga as datas de todas as provas de uma disciplina. Caso deseje visualizar um exemplo acesse a página: http://www.cin.ufpe.br/~if119/calendário.htm
Peso: a ser calculado posteriormente (fase 3)
Escore: a ser calculado posteriormente (fase 4)

Figura 5.12 – Planilha preenchida do atributo *Das provas*

Após especificar descritivamente as características, subcaracterísticas e os atributos de qualidade devemos passar para a outra fase da metodologia REQE.

5.4 Associação de Pesos aos Nós

Passaremos agora para a fase de *Associação de Pesos aos Nós*. Quanto mais importante for a característica, a subcaracterística ou o atributo de qualidade, maior deve ser o seu peso com relação aos nós do seu nível. Esses pesos devem ser associados pelos avaliadores. A fim de comparação de aplicações Web, aplicações do mesmo domínio devem ser submetidas à mesma distribuição de pesos. Diferentemente da fase de *Associação de Escores aos Atributos*, esta fase associa pesos a todos os nós da árvore de características, subcaracterísticas e atributos de qualidade. Vale lembrar que a soma de todos os pesos dos filhos de determinado pai deve ser igual a 10 (dez). Por meio de discussões entre alguns *stakeholders* chegamos à distribuição de pesos constante neste capítulo. Os pesos podem ser vistos entre parênteses ao lado de cada nó nas figuras a seguir.

Definiremos inicialmente o peso das subcaracterísticas de *Usabilidade*. Dentre os seus filhos diretos resolvemos eleger a subcaracterísticas *Ajuda ao Usuário* como a mais importante. Os sites acadêmicos são, muitas vezes, a porta de entrada dos alunos para as universidades. Em um primeiro instante, o sentimento de desejar algum tipo de ajuda e não encontrá-la acaba por desestimular o estudante a prosseguir e explorar o site. Ocorre uma espécie de frustração. Por isso atribuímos a esta subcaracterística o maior peso entre os nós do seu nível, 4 (quatro). Em grau de importância, achamos que a subcaracterística *Entendimento global do site* deveria ficar acima de *Interface e estética*. Assim atribuímos respectivamente os pesos 3 (três) e 2 (dois) a estes nós. As *Tarefas diversas* foram associadas ao peso 1 (um), tendo em vista que são menos relevantes no contexto da *Usabilidade*, em relação as outras subcaracterísticas.

Já em relação aos filhos da característica *Funcionalidade* atribuímos o peso 2 (dois) para as *Tarefas de busca* e o peso 8 (oito) para as *Tarefas do domínio relativo ao estudante*. Pelo fato do universo da subcaracterística *Tarefas do domínio relativo ao estudante* ser bem mais amplo do que o da subcaracterística *Tarefas de busca* resolvemos destacar o seu grau

de importância dentro da característica *Funcionalidade*. A distribuição dos pesos das subcaracterísticas de *Usabilidade* e *Funcionalidade* pode ser vista na Figura 5.13.

- | |
|--|
| <ol style="list-style-type: none"> 1. Usabilidade <ol style="list-style-type: none"> 1.1. Entendimento global do site (3) <ol style="list-style-type: none"> 1.1.1. 1.2. Ajuda ao usuário (4) <ol style="list-style-type: none"> 1.2.1. 1.3. Interface e estética (2) <ol style="list-style-type: none"> 1.3.1. 1.4. Tarefas diversas (1) <ol style="list-style-type: none"> 1.4.1. 2. Funcionalidade <ol style="list-style-type: none"> 2.1. Tarefas de busca (2) <ol style="list-style-type: none"> 2.1.1. 2.2. Tarefas do domínio relativo ao estudante (8) <ol style="list-style-type: none"> 2.2.1. |
|--|

Figura 5.13 – Subcaracterísticas de *Usabilidade* e *Funcionalidade* com pesos

A subcaracterística *Tarefas diversas*, filha da característica Usabilidade, foi decomposta em 3 (três) atributos de qualidade: *Suporte a outros idiomas*, *Mecanismos para destacar novidades* e *Indicação para resolução do monitor*. O primeiro destes atributos de qualidade recebeu o peso 8 (oito) por ser considerado de suma importância, uma vez que um site acadêmico frequentemente é visitado por estudantes de outros países. Os outros dois atributos receberam um importância equivalente, peso 1(um), visto que não são tão fundamentais como o *Suporte a outros idiomas*. Estes pesos podem ser vistos na Figura 5.14.

A subcaracterística *Informações da infra-estrutura acadêmica* foi decomposta em duas outras subcaracterísticas: *Bibliotecas* e *Laboratórios*. Esta primeira subcaracterística recebeu peso 7 (sete) e a segunda 3 (três). Distribuímos o peso desta forma, pois segundo a opinião de alguns *stakeholders* consultados, é mais importante para a qualidade de um site acadêmico conter serviços e informações relativas às bibliotecas do que dos laboratórios. A subcaracterística *Bibliotecas* foi refinada em três atributos de qualidade: *Busca de*

exemplares, *Reserva de exemplares* e *Informações gerais*. O atributo *Busca de exemplares* foi considerado com maior grau de importância, recebendo o peso 7 (sete). Aos outros dois atributos respectivamente foram associados os pesos 2 (dois) e 1 (um). Já a subcaracterística *Laboratórios*, foi decomposta em: *Reserva*, *Informações gerais* e *Resultados de pesquisa*, recebendo respectivamente os pesos 6 (seis), 3 (três) e 1(um). O serviço de reserva dos laboratórios pelo site foi considerado como mais importante entre os atributos filhos da subcaracterística *Laboratórios*, tendo em vista que é bastante desejado pelos estudantes pesquisados. Vejamos na Figura 5.15 como ficou esta distribuição dos pesos.

...
...
1.4 Tarefas diversas
1.4.1 Suporte a outros idiomas (8)
1.4.2 Mecanismos para destacar novidade (1)
1.4.3 Indicação da resolução do monitor (1)
...
...

Figura 5.14 – Atributos de *Tarefas diversas* com pesos

...
...
2.2.1.5 Informações da infra-estrutura acadêmica
2.2.1.5.1 Bibliotecas (7)
2.2.1.5.1.1 Busca de exemplares (7)
2.2.1.5.1.2 Reserva de exemplares (2)
2.2.1.5.1.3 Informações gerais (1)
2.2.1.5.2 Laboratórios (3)
2.2.1.5.2.1 Reservas (6)
2.2.1.5.2.2 Informações gerais (3)
2.2.1.5.2.3 Resultados de pesquisa (1)
...
...

Figura 5.15 – Parte da ACSAQ de *Informações da infra-estrutura acadêmica*

Na Figura 5.16 apresentaremos a árvore completa de características, subcaracterísticas e atributos de qualidade, com os pesos associados a todos os nós.

- 1. Usabilidade (5)
 - 1.1 Entendimento global do site (3)
 - 1.1.1 Esquema de organização global (4)
 - 1.1.1.1 Mapa do site (4)
 - 1.1.1.2 Tabela de conteúdos (3)
 - 1.1.1.3 Índice alfabético (2)
 - 1.1.1.4 Localizador no mapa (1)
 - 1.1.2 Tarefas de navegação (4)
 - 1.1.2.1 Navegabilidade (3)
 - 1.1.2.1.1 Orientação (10)
 - 1.1.2.1.1.1 Indicador do caminho (5)
 - 1.1.2.1.1.2 Marcador da posição atual (5)
 - 1.1.2.2 Objetos de controle de navegação (4)
 - 1.1.2.2.1 Apresentação dos controles contextuais (8)
 - 1.1.2.2.1.1 Permanência dos controles (6)
 - 1.1.2.2.1.2 Intuitividade dos controles (4)
 - 1.1.2.2.2 Mecanismos de rolagem (2)
 - 1.1.2.2.2.1 Vertical (5)
 - 1.1.2.2.2.2 Horizontal (5)
 - 1.1.2.3 Intuitividade da navegação (3)
 - 1.1.2.3.1 Título do link (5)
 - 1.1.2.3.2 Qualidade da frase do link (5)
 - 1.1.3 Visita guiada para estudantes (1)
 - 1.1.4 Mapa de imagens (1)
 - 1.2 Ajuda ao usuário (4)
 - 1.2.1 Qualidade da ajuda (3)
 - 1.2.1.1 Ajuda geral direcionada para o estudante (6)
 - 1.2.1.2 Ajuda para busca (4)
 - 1.2.2 Indicador de atualização (2)
 - 1.2.2.1 Global (6)
 - 1.2.2.2 Por subsite ou página (4)
 - 1.2.3 Acesso aos endereços (2)
 - 1.2.3.1 Endereço eletrônico (5)
 - 1.2.3.2 Telefone/fax (3)
 - 1.2.3.3 Ramais internos (1)
 - 1.2.3.4 Endereço para correspondência (1)
 - 1.2.4 FAQ (2)
 - 1.2.5 Formulários de retorno (1)
 - 1.2.5.1 Questionários (8)
 - 1.2.5.2 Livro de visitas (1)

Figura 5.16 – Árvore de sites acadêmicos com pesos

- 1.2.5.3 Comentários (1)
- 1.3 Interface e estética (2)
 - 1.3.1 Coesão no agrupamento de objetos de controle (2)
 - 1.3.2 Apresentação dos controles principais (4)
 - 1.3.2.1 Intuitividade dos controles (8)
 - 1.3.2.2 Permanência dos controles (2)
 - 1.3.3 Estilo (4)
 - 1.3.3.1 Uniformidade no estilo de cores dos links (5)
 - 1.3.3.2 Uniformidade no estilo global (2)
 - 1.3.3.3 Guia do estilo global (3)
- 1.4 Tarefas diversas (1)
 - 1.4.1 Suporte a outros idiomas (8)
 - 1.4.2 Mecanismos para destacar novidade (1)
 - 1.4.3 Indicação da resolução do monitor (1)
- 2 Funcionalidade (5)
 - 2.1 Tarefas de busca (2)
 - 2.1.1 Mecanismos de busca (10)
 - 2.1.1.1 Busca por escopo (5)
 - 2.1.1.1.1 Busca de pessoas (2)
 - 2.1.1.1.2 Busca por cursos (4)
 - 2.1.1.1.3 Busca por disciplinas acadêmicas (4)
 - 2.1.1.2 Busca global (5)
 - 2.1.1.2.1 Informações para refinamento (10)
 - 2.2 Tarefas do domínio relativo ao estudante (8)
 - 2.2.1 Conteúdo relevante (9)
 - 2.2.1.1 Informações sobre unidade acadêmicas (2)
 - 2.2.1.1.1 Índice de disciplinas acadêmicas (3)
 - 2.2.1.1.2 Subsite de disciplinas acadêmicas (7)
 - 2.2.1.2 Informações diversas (2)
 - 2.2.1.2.1 Requerimentos (10)
 - 2.2.1.2.1.1 Entrada (4)
 - 2.2.1.2.1.2 Acompanhamento (4)
 - 2.2.1.2.1.3 Download de formulários (2)
 - 2.2.1.3 Qualidade do conteúdo (2)
 - 2.2.1.3.1 Qualidade do índice (1)
 - 2.2.1.3.2 Qualidade do conteúdo (1)
 - 2.2.1.3.3 Qualidade das informações dos cursos (2)
 - 2.2.1.3.3.1 Ementas (7)
 - 2.2.1.3.3.2 Formas de avaliação (3)
 - 2.2.1.3.4 Descrição dos cursos (6)
 - 2.2.1.3.4.1 Comentários (1)
 - 2.2.1.3.4.2 Plano de ensino (3)
 - 2.2.1.3.4.3 Calendário (2)
 - 2.2.1.3.4.3.1 Das aulas (2)

Continuação da Figura 5.16

- 2.2.1.3.4.3.2 Das provas (6)
- 2.2.1.3.4.3.3 Dos trabalhos (1)
- 2.2.1.3.4.3.4 Diversos (1)
- 2.2.1.3.4.4 Cursos de pós-graduação (3)
 - 2.2.1.3.4.4.1 Especialização (4)
 - 2.2.1.3.4.4.2 Mestrado (3)
 - 2.2.1.3.4.4.3 Doutorado (3)
- 2.2.1.3.4.5 Organogramas (1)
 - 2.2.1.3.4.5.1 Organograma global (5)
 - 2.2.1.3.4.5.2 Organograma departamental (5)
- 2.2.1.4 Informação dos serviços dos estudantes (2)
 - 2.2.1.4.1 Índice de serviços (2)
 - 2.2.1.4.2 Informações do calendário escolar (2)
 - 2.2.1.4.3 Informações dos alojamentos (2)
 - 2.2.1.4.4 Informações culturais (1)
 - 2.2.1.4.5 Informações esportivas (1)
 - 2.2.1.4.6 Informações sobre congressos (2)
- 2.2.1.5 Informações da infra-estrutura acadêmica (2)
 - 2.2.1.5.1 Bibliotecas (7)
 - 2.2.1.5.1.1 Busca de exemplares (7)
 - 2.2.1.5.1.2 Reserva de exemplares (2)
 - 2.2.1.5.1.3 Informações gerais (1)
 - 2.2.1.5.2 Laboratórios (3)
 - 2.2.1.5.2.1 Reserva (6)
 - 2.2.1.5.2.2 Informações gerais (3)
 - 2.2.1.5.2.3 Resultados de pesquisa (1)
- 2.2.2 Serviços on-line (1)
 - 2.2.2.1 Informação da grade curricular (1)
 - 2.2.2.2 Serviços Web (7)
 - 2.2.2.3 Serviços FTP (1)
 - 2.2.2.4 Serviços de newsgroup (1)

Continuação da Figura 5.16

Ao concluir esta etapa, estamos com a árvore de características, subcaracterísticas e atributos de qualidade definida, especificada item por item por meio das planilhas e com todos os pesos associados aos nós. Precisamos então, na fase *Associação de Escores aos Atributos*, com base no documento de requisitos (constante no Apêndice B) determinar o nível da especificação de cada um dos atributos de qualidade presentes na ACSAQ. Note que algumas seções do documento de requisitos não foram detalhadas por não fazerem

parte do escopo deste exemplo, sem prejudicar, portanto o seu objetivo. Procuramos focar as partes que influenciam diretamente neste relato de experiência.

5.5 Associação de Escores aos Atributos

Adotaremos a tabela de conformidade definida na Figura 4.10, com 4 (quatro) níveis de conformidade: *não atende o atributo de qualidade* (escore 0), *atende com sérias restrições* (escore 3), *atende parcialmente* (escore 6,5) e *atende* (escore 10). Apenas os atributos serão valorados com base nesta tabela. A nota dos outros nós (características e subcaracterísticas) será calculada com base nas regras expostas na Seção 4.6. Apresentaremos novamente a árvore de características, subcaracterísticas e atributos de qualidade, agora além dos pesos teremos os escores de conformidade, estes entre colchetes. Posteriormente explicaremos como os mesmos foram obtidos.

- 1. Usabilidade (5) []
 - 1.1. Entendimento global do site (3) []
 - 1.1.1. Esquema de organização global (4) []
 - 1.1.1.1. Mapa do site (4) [0]
 - 1.1.1.2. Tabela de conteúdos (3) [0]
 - 1.1.1.3. Índice alfabético (2) [0]
 - 1.1.1.4. Localizador no mapa (1) [0]
 - 1.1.2. Tarefas de navegação (4) []
 - 1.1.2.1. Navegabilidade (3) []
 - 1.1.2.1.1. Orientação (10) []
 - 1.1.2.1.1.1. Indicador do caminho (5) [0]
 - 1.1.2.1.1.2. Marcador da posição atual (5) [0]
 - 1.1.2.2. Objetos de controle de navegação (4) []
 - 1.1.2.2.1. Apresentação dos controles contextuais (8) []
 - 1.1.2.2.1.1. Permanência dos controles (6) [0]
 - 1.1.2.2.1.2. Intuitividade dos controles (4) [6,5]
 - 1.1.2.2.2. Mecanismos de rolagem (2) []
 - 1.1.2.2.2.1. Vertical (5) [0]
 - 1.1.2.2.2.2. Horizontal (5) [0]
 - 1.1.2.3. Intuitividade da navegação (3) []
 - 1.1.2.3.1. Título do link (5) [10]
 - 1.1.2.3.2. Qualidade da frase do link (5) [10]
 - 1.1.3. Visita guiada para estudantes (1) [0]

Figura 5.17 – Árvore de sites acadêmicos com pesos e escores

- 1.1.4. Mapa de imagens (1) **[6,5]**
- 1.2. Ajuda ao usuário (4) []
 - 1.2.1. Qualidade da ajuda (3) []
 - 1.2.1.1. Ajuda geral direcionada para o estudante (6) **[3]**
 - 1.2.1.2. Ajuda para busca (4) **[10]**
 - 1.2.2. Indicador de atualização (2) []
 - 1.2.2.1. Global (6) **[0]**
 - 1.2.2.2. Por subsite ou página (4) **[0]**
 - 1.2.3. Acesso aos endereços (2) []
 - 1.2.3.1. Endereço eletrônico (5) **[10]**
 - 1.2.3.2. Telefone/fax (3) **[10]**
 - 1.2.3.3. Ramais internos (1) **[10]**
 - 1.2.3.4. Endereço para correspondência (1) **[10]**
 - 1.2.4. FAQ (2) **[0]**
 - 1.2.5. Formulários de retorno (1) []
 - 1.2.5.1. Questionários (8) **[3]**
 - 1.2.5.2. Livro de visitas (1) **[0]**
 - 1.2.5.3. Comentários (1) **[0]**
- 1.3. Interface e estética (2) []
 - 1.3.1. Coesão no agrupamento de objetos de controle (2) **[6,5]**
 - 1.3.2. Apresentação dos controles principais (4) []
 - 1.3.2.1. Intuitividade dos controles (8) **[6,5]**
 - 1.3.2.2. Permanência dos controles (2) **[6,5]**
 - 1.3.3. Estilo (4) []
 - 1.3.3.1. Uniformidade no estilo de cores dos links (5) **[6,5]**
 - 1.3.3.2. Uniformidade no estilo global (2) **[0]**
 - 1.3.3.3. Guia do estilo global (3) **[0]**
- 1.4. Tarefas diversas (1) []
 - 1.4.1. Suporte a outros idiomas (8) **[0]**
 - 1.4.2. Mecanismos para destacar novidade (1) **[10]**
 - 1.4.3. Indicação da resolução do monitor (1) **[0]**
- 2. Funcionalidade (5) []
 - 2.1. Tarefas de busca (2) []
 - 2.1.1. Mecanismos de busca (10) []
 - 2.1.1.1. Busca por escopo (5) []
 - 2.1.1.1.1. Busca de pessoas (2) **[10]**
 - 2.1.1.1.2. Busca por cursos (4) **[10]**
 - 2.1.1.1.3. Busca por disciplinas acadêmicas (4) **[10]**
 - 2.1.1.2. Busca global (5) []
 - 2.1.1.2.1. Informações para refinamento (10) **[10]**
 - 2.2. Tarefas do domínio relativo ao estudante (8) []
 - 2.2.1. Conteúdo relevante (9) []
 - 2.2.1.1. Informações sobre unidade acadêmicas (2) []
 - 2.2.1.1.1. Índice de disciplinas acadêmicas (3) **[10]**

Figura 5.17 – Continuação

- 2.2.1.1.2. Subsite de disciplinas acadêmicas (7) **[6,5]**
- 2.2.1.2. Informações diversas (2) []
 - 2.2.1.2.1. Requerimentos (10) []
 - 2.2.1.2.1.1. Entrada (4) **[6,5]**
 - 2.2.1.2.1.2. Acompanhamento (4) **[3]**
 - 2.2.1.2.1.3. Download de formulários (2) **[3]**
- 2.2.1.3. Qualidade do conteúdo (2) []
 - 2.2.1.3.1. Qualidade do índice (1) **[3]**
 - 2.2.1.3.2. Qualidade do conteúdo (1) **[10]**
 - 2.2.1.3.3. Qualidade das informações dos cursos (2) []
 - 2.2.1.3.3.1. Ementas (7) **[0]**
 - 2.2.1.3.3.2. Formas de avaliação (3) **[0]**
 - 2.2.1.3.4. Descrição dos cursos (6) []
 - 2.2.1.3.4.1. Comentários (1) **[0]**
 - 2.2.1.3.4.2. Plano de ensino (3) **[0]**
 - 2.2.1.3.4.3. Calendário (2) []
 - 2.2.1.3.4.3.1. Das aulas (2) **[10]**
 - 2.2.1.3.4.3.2. Das provas (6) **[0]**
 - 2.2.1.3.4.3.3. Dos trabalhos (1) **[0]**
 - 2.2.1.3.4.3.4. Diversos (1) **[0]**
 - 2.2.1.3.4.4. Cursos de pós-graduação (3) []
 - 2.2.1.3.4.4.1. Especialização (4) **[6,5]**
 - 2.2.1.3.4.4.2. Mestrado (3) **[6,5]**
 - 2.2.1.3.4.4.3. Doutorado (3) **[6,5]**
 - 2.2.1.3.4.5. Organogramas (1) []
 - 2.2.1.3.4.5.1. Organograma global (5) **[10]**
 - 2.2.1.3.4.5.2. Organograma departamental (5) **[10]**
- 2.2.1.4. Informação dos serviços dos estudantes (2) []
 - 2.2.1.4.1. Índice de serviços (2) **[6,5]**
 - 2.2.1.4.2. Informações do calendário escolar (2) **[6,5]**
 - 2.2.1.4.3. Informações dos alojamentos (2) **[0]**
 - 2.2.1.4.4. Informações culturais (1) **[0]**
 - 2.2.1.4.5. Informações esportivas (1) **[0]**
 - 2.2.1.4.6. Informações sobre congressos (2) **[6,5]**
- 2.2.1.5. Informações da infra-estrutura acadêmica (2) []
 - 2.2.1.5.1. Bibliotecas (7) []
 - 2.2.1.5.1.1. Busca de exemplares (7) **[10]**
 - 2.2.1.5.1.2. Reserva de exemplares (2) **[0]**
 - 2.2.1.5.1.3. Informações gerais (1) **[6,5]**
 - 2.2.1.5.2. Laboratórios (3) []
 - 2.2.1.5.2.1. Reserva (6) **[10]**
 - 2.2.1.5.2.2. Informações gerais (3) **[10]**
 - 2.2.1.5.2.3. Resultados de pesquisa (1) **[0]**
- 2.2.2. Serviços on-line (1) []

Figura 5.17 – Continuação

- 2.2.2.1. Informação da grade curricular (1) [10]
- 2.2.2.2. Serviços Web (7) [6,5]
- 2.2.2.3. Serviços FTP (1) [0]
- 2.2.2.4. Serviços de newsgroup (1) [3]

Figura 5.17 – Continuação

O nós que não são atributos estão com os colchetes vazios, pois terão suas notas calculadas na próxima fase da metodologia.

Explicaremos brevemente como associamos cada escore de conformidade aos atributos. É óbvio que a tarefa de valorar um atributo envolve a interpretação do documento de requisitos (vide Apêndice B). Para algumas pessoas certo item pode estar bem especificado, enquanto para outras necessite de um maior detalhamento. Tudo isso envolve reuniões e discussões a respeito da especificação, conceitos de certa forma subjetivos. Explicaremos resumidamente a idéia de como cada atributo recebeu determinado escore.

Os atributos *Mapa do site*, *Tabela de conteúdos*, *Índice alfabético*, *Localizador no mapa*, *Indicador de caminho*, *Marcador da posição atual*, *Permanência dos controles*, *(Mecanismos de rolagem) Vertical*, *(Mecanismos de rolagem) Horizontal*, *Visita guiada para estudantes*, *(Indicador de atualização) Global*, *(Indicador de atualização) Por subsite ou página*, *FAQ*, *(Formulários de retorno) Livro de visitas*, *(Formulários de retorno) Comentários*, *Uniformidade no estilo global*, *Guia do estilo global*, *Suporte a outros idiomas*, *Indicação da resolução do monitor*, *(Qualidade das informações dos cursos) Formas de avaliação*, *(Descrição dos cursos) Comentários*, *Plano de ensino*, *(Calendário) Das provas*, *(Calendário) Dos trabalhos*, *(Calendário) Diversos*, *Informações dos alojamentos*, *Informações culturais*, *Informações esportivas*, *(Bibliotecas) Reserva de exemplares*, *(Laboratórios) Resultado de pesquisas* e *Serviços FTP* receberam a conformidade “*Não atende o atributo de qualidade*”, que equivale ao escore 0 (zero), pois o documento de requisitos não especifica esses atributos. Por não estarem especificados, obviamente poderão não ser implementados, em consequência o produto final não levaria em consideração tais atributos de qualidade.

O atributo “*Ajuda geral direcionada para o estudante*” (1.2.1.1) recebeu a conformidade “*Atende com sérias restrições*” (escore 3), pois apenas foi citado que o site deveria prover um mecanismo de ajuda. Nenhum detalhe foi abordado, que tipo de ajuda seria essa, como se daria a interação e qual o conteúdo desta ajuda. Vide Seção 6.1 (1º parágrafo) do Documento de Requisitos da Universidade X, conforme Apêndice B.

O atributo “*Questionários*” (1.2.5.1) também recebeu a conformidade “*Atende com sérias restrições*” (escore 3). Apesar de ser citada a importância desse tipo de recurso para o incremento da qualidade, não foi especificado como esse questionário deveria ser apresentado ao usuário, quantas perguntas deveria ter, quais perguntas etc. Vide Seção 6.1 (2º parágrafo) do Documento de Requisitos da Universidade X, conforme Apêndice B.

Os atributos “*Acompanhamento*” (2.2.1.2.1.2) e “*Download de formulários*” (2.2.1.2.1.3) não foram claramente especificados. Apesar do documento de requisitos detalhar a entrada dos requerimentos, apenas cita que um protocolo será gerado para futuras consultas. Como se darão essas consultas? Pela web? O documento especifica que em determinados casos os formulários devem ser entregues manualmente. Uma possível interpretação é que os mesmos estarão disponíveis para download, e posterior preenchimento de modo a serem entregues de forma manual. Mas essa não é a única interpretação possível. Por isso ambos receberam a conformidade “*Atende com sérias restrições*” (escore 3). Vide Seção 5.9 do Documento de Requisitos da Universidade X, conforme Apêndice B.

Com relação a “*Qualidade do Índice*” (2.2.1.3.1) também avaliamos com “*Atende com sérias restrições*” (escore 3), pois apenas foi citado o índice de disciplinas acadêmicas, e uma sugestão para listar os professores, cursos e alunos. Vide Seção 5.14 do Documento de Requisitos da Universidade X, conforme Apêndice B.

O documento cita a existência de newsgroup no site, mas não aborda como os mesmos devem estar estruturados (por exemplo, listas de e-mails, chats, etc). Esse é um erro bastante freqüente. O analista de requisitos acha que determinada informação é tão intuitiva que a omite. Por isso recebeu a conformidade “*Atende com sérias restrições*” (escore 3). Vide Seção 5.11 do Documento de Requisitos da Universidade X, conforme Apêndice B.

A “*Coesão no agrupamento de objetos de controle*” é citada como de fundamental importância na disposição dos mesmos. Este atributo, de certa forma bastante subjetivo, recebe a conformidade “*Atende parcialmente*” (escore 6,5), pela preocupação que o documento se referencia a esse tema. Vale salientar que poderíamos ter refinado este nó em outros mais objetivos. Vide Seção 6.1 (5º parágrafo) do Documento de Requisitos da Universidade X, conforme Apêndice B.

O documento de requisitos não previu quais as cores para os links visitados e não visitados. No entanto, por ter registrado a importância da marcação dos links já visitados na navegabilidade do sistema o atributo “*Uniformidade no estilo de cores dos links*” recebeu a conformidade “*Atende parcialmente*” (escore 6,5). Vide Seção 6.1 (3º parágrafo) do Documento de Requisitos da Universidade X, conforme Apêndice B.

O atributo “*Intuitividade dos controles*” recebeu a conformidade “*Atende parcialmente*” (escore 6,5), pois o documento de requisitos deu bastante ênfase a esse ponto. Vide Seção 6.1 (4º parágrafo) do Documento de Requisitos da Universidade X, conforme Apêndice B.

A funcionalidade “*Mapa Campus*”, nada mais é do que um mapa de imagens. Este foi especificado, porém não o suficiente, por isso recebeu a conformidade “*Atende parcialmente*” (escore 6,5). Vide Seção 5.12 do Documento de Requisitos da Universidade X, conforme Apêndice B.

O atributo “*Subsite de disciplinas acadêmicas*” foi parcialmente atendido (escore 6,5). Não foi contemplada a questão da padronização dos mesmos, bem como que tipos de informações obrigatórias devem estar presentes nesses subsites. Vide Seção 5.15 do Documento de Requisitos da Universidade X, conforme Apêndice B.

O documento especifica a necessidade de disponibilizar informações sobre “*Especialização*”, “*Mestrado*” e “*Doutorado*”. Poderia ser mais específico, então cada um desses atributos recebe a conformidade “*Atende parcialmente*” (escore 6,5). Vide Seção 5.4 do Documento de Requisitos da Universidade X, conforme Apêndice B.

Nas funcionalidades relativas às publicações, as “*Informações do calendário escolar*” e “*Informações sobre congressos*” foram abordadas. Como as especificações desses atributos de qualidade são bastante simples e não demandam maiores detalhes, os colocamos no patamar de “*Atende parcialmente*” (escore 6,5). Não obstante outras informações poderiam ser especificadas como o que deve ser abordado nas notícias sobre os congressos, quantas publicações a referida Universidade (os seus departamentos) teve nos anos anteriores do congresso citado, etc. Vide Seção 5.4 do Documento de Requisitos da Universidade X, conforme Apêndice B.

A “*Busca de exemplares*” das bibliotecas foi suficientemente especificada, recebendo a conformidade “*Atende*” (escore 10). Algumas “*Informações gerais*” foram expostas, atendendo parcialmente o requisito de qualidade (escore 6,5). Vide Seção 5.16 do Documento de Requisitos da Universidade X, conforme Apêndice B.

O “*Índice de serviços*” e “*Serviços Web*” estão intimamente ligados à funcionalidade de manutenção dos requerimentos, o qual foi especificado de forma razoável. Portanto foi avaliado como parcialmente atendido (escore 6,5). Vide Seção 5.19 do Documento de Requisitos da Universidade X, conforme Apêndice B.

A seção do documento de requisitos que trata os requisitos de usabilidade relata algumas regras que garantem a conformidade máxima ao “*Título do link*” e “*Qualidade da frase do*

link”, ou seja, conformidade “*Atende*” (escore 10). Vide Seção 6.1 (4º parágrafo) do Documento de Requisitos da Universidade X, conforme Apêndice B.

O atributo “*Ajuda para busca*” recebeu a conformidade “*Atende*” (escore 10) pelas exigências que o documento de requisitos faz para esse tipo de ajuda. De forma análoga “*Informações para refinamento*” foi em grande parte detalhada no documento, por isso também recebeu escore 10 (dez), tendo sido disponibilizado até mesmo um exemplo de como esse refinamento deve funcionar. Vide Seção 5.13 do Documento de Requisitos da Universidade X, conforme Apêndice B.

“*Endereço eletrônico*”, “*Telefone/fax*”, “*Ramais internos*” e “*Endereço para correspondência*” foram citados como existentes na funcionalidade *Comunicação*. Por esse fato atenderam as expectativas, recebendo a conformidade “*Atende*” (escore 10). Vide Seção 5.18 do Documento de Requisitos da Universidade X, conforme Apêndice B.

A funcionalidade “*Divulgação*” especifica o “*Mecanismo para destacar novidade*”, explicando o funcionamento da publicação de notícias a serem veiculadas no site. Essa funcionalidade pode ser agregada por meio de um componente externo, e recebeu a conformidade “*Atende*” (escore 10). Vide Seção 5.8 do Documento de Requisitos da Universidade X, conforme Apêndice B.

A “*Busca de pessoas*”, “*Busca por cursos*”, “*Busca por disciplina acadêmicas*” foram especificadas como buscas particulares no contexto da busca global. O documento sugeriu uma forma de dispor esses elementos de modo a facilitar a pesquisa de qualquer um deles. Foi proposta uma árvore semelhante ao Windows Explorer, onde seria possível achar determinado elemento sem nem mesmo preencher um critério de busca. A conformidade “*Atende*” foi associada aos referidos atributos, como também ao “*Índice de disciplinas acadêmicas*” (escore 10). Vide Seção 5.14 do Documento de Requisitos da Universidade X, conforme Apêndice B.

O documento de requisitos delega a responsabilidade de revisar todo o conteúdo do site ao assessor de comunicação, de forma a garantir um nível satisfatório do conteúdo publicado. Levamos em consideração aqui a preocupação em se manter um conteúdo de boa qualidade e por isso associamos a conformidade “*Atende*” (escore 10) ao atributo “*Qualidade do conteúdo*”. Vide Seção 5.8 do Documento de Requisitos da Universidade X, conforme Apêndice B.

O documento de requisitos especificou que informações relativas ao calendário das provas deveriam estar disponíveis no site. Assim, atributo “*Das aulas*” recebeu a conformidade “*Atende*” (escore 10). Vide Seção 5.3 e 5.14 do Documento de Requisitos da Universidade X, conforme Apêndice B.

O “*Organograma global*” e o “*Organograma departamental*” foram previstos com detalhes, tendo sido associado a conformidade “*Atende*” (escore 10). Vide Seção 5.1 do Documento de Requisitos da Universidade X, conforme Apêndice B.

A “*Reserva dos laboratórios*” e as “*Informações gerais*” foram especificadas detalhadamente no documento de requisitos, recebendo o escore 10 (dez), indicando o seu atendimento pleno. Vide Seção 5.17 do Documento de Requisitos da Universidade X, conforme Apêndice B.

Para cada curso (de graduação ou pós) também foi detalhada a necessidade de se disponibilizar “*Informação da grade curricular*”, sendo associado a este atributo a conformidade “*Atende*” (escore 10). Vide Seção 5.3 e 5.4 do Documento de Requisitos da Universidade X, conforme Apêndice B.

Estando todos os atributos associados aos seus escores de conformidade, e todos os nós da ACSAQ com seus pesos, podemos passar para a última fase da metodologia REQE.

5.6 Cálculo Geral

Finalmente chegamos a fase do cálculo geral. Por meio deste, todos os outros nós (subcaracterísticas e características) terão uma nota associada. Para facilitar o entendimento mostraremos ao lado das notas como as mesmas foram calculadas. As notas que foram calculadas através da combinação dos nós filhos estão em negrito.

1. Usabilidade (5) **[3,39]** $(3*2,18 + 4*3,98 + 2*5,2 + 1*1)/10 = 3,39$
 - 1.1. Entendimento global do site (3) **[2,18]** $(4*0 + 4*3,83 + 1*0 + 1*6,5)/10 = 2,18$
 - 1.1.1. Esquema de organização global (4) **[0]** $(4*0 + 3*0 + 2*0 + 1*0)/10 = 0$
 - 1.1.1.1. Mapa do site (4) [0]
 - 1.1.1.2. Tabela de conteúdos (3) [0]
 - 1.1.1.3. Índice alfabético (2) [0]
 - 1.1.1.4. Localizador no mapa (1) [0]
 - 1.1.2. Tarefas de navegação (4) **[3,83]** $(3*0 + 4*2,08 + 3*10)/10 = 3,83$
 - 1.1.2.1. Navegabilidade (3) **[0]** $(10*0)/10 = 0$
 - 1.1.2.1.1. Orientação (10) **[0]** $(5*0 + 5*0)/10 = 0$
 - 1.1.2.1.1.1. Indicador do caminho (5) [0]
 - 1.1.2.1.1.2. Marcador da posição atual (5) [0]
 - 1.1.2.2. Objetos de controle de navegação (4) **[2,08]** $(8*2,6 + 2*0)/10 = 2,08$
 - 1.1.2.2.1. Apresentação dos controles contextuais (8) **[2,6]** $(6*0 + 4*6,5)/10 = 2,6$
 - 1.1.2.2.1.1. Permanência dos controles (6) [0]
 - 1.1.2.2.1.2. Intuitividade dos controles (4) [6,5]
 - 1.1.2.2.2. Mecanismos de rolagem (2) **[0]**
 - 1.1.2.2.2.1. Vertical (5) [0]
 - 1.1.2.2.2.2. Horizontal (5) [0]
 - 1.1.2.3. Intuitividade da navegação (3) **[10]** $(5*10 + 5*10)/10 = 10$
 - 1.1.2.3.1. Título do link (5) [10]
 - 1.1.2.3.2. Qualidade da frase do link (5) [10]
 - 1.1.3. Visita guiada para estudantes (1) [0]
 - 1.1.4. Mapa de imagens (1) [6,5]
 - 1.2. Ajuda ao usuário (4) **[3,98]** $(3*5,8 + 2*0 + 2*10 + 2*0 + 1*2,4)/10 = 3,98$
 - 1.2.1. Qualidade da ajuda (3) **[5,8]** $(6*3 + 4*10)/10 = 5,8$
 - 1.2.1.1. Ajuda geral direcionada para o estudante (6) [3]
 - 1.2.1.2. Ajuda para busca (4) [10]
 - 1.2.2. Indicador de atualização (2) **[0]** $(6*0 + 4*0)/10 = 0$
 - 1.2.2.1. Global (6) [0]
 - 1.2.2.2. Por subsite ou página (4) [0]
 - 1.2.3. Acesso aos endereços (2) **[10]** $(5*10 + 3*10 + 1*10 + 1*10)/10 = 10$

Figura 5.18 – ACSAQ de sites acadêmicos com pesos e escores em todos os nós

- 1.2.3.1. Endereço eletrônico (5) [10]
- 1.2.3.2. Telefone/fax (3) [10]
- 1.2.3.3. Ramais internos (1) [10]
- 1.2.3.4. Endereço para correspondência (1) [10]
- 1.2.4. FAQ (2) [0]
- 1.2.5. Formulários de retorno (1) **[2,4]** $(8*3 + 1*0 + 1*0)/10 = 2,4$
 - 1.2.5.1. Questionários (8) [3]
 - 1.2.5.2. Livro de visitas (1) [0]
 - 1.2.5.3. Comentários (1) [0]
- 1.3. Interface e estética (2) **[5,2]** $(2*6,5 + 4*6,5 + 4*3,25)/10 = 5,2$
 - 1.3.1. Coesão no agrupamento de objetos de controle (2) [6,5]
 - 1.3.2. Apresentação dos controles principais (4) **[6,5]** $(8*6,5 + 2*6,5)/10 = 6,5$
 - 1.3.2.1. Intuitividade dos controles (8) [6,5]
 - 1.3.2.2. Permanência dos controles (2) [6,5]
 - 1.3.3. Estilo (4) **[3,25]** $(5*6,5 + 2*0 + 3*0)/10 = 3,25$
 - 1.3.3.1. Uniformidade no estilo de cores dos links (5) [6,5]
 - 1.3.3.2. Uniformidade no estilo global (2) [0]
 - 1.3.3.3. Guia do estilo global (3) [0]
- 1.4. Tarefas diversas (1) **[1]** $(8*0 + 1*10 + 1*0)/10 = 1$
 - 1.4.1. Suporte a outros idiomas (8) [0]
 - 1.4.2. Mecanismos para destacar novidade (1) [10]
 - 1.4.3. Indicação da resolução do monitor (1) [0]
- 2. Funcionalidade (5) **[6,38]** $(2*10 + 8*5,48)/10 = 6,38$
 - 2.1. Tarefas de busca (2) **[10]** $(10*10)/10 = 10$
 - 2.1.1. Mecanismos de busca (10) **[10]** $(5*10 + 5*10)/10 = 10$
 - 2.1.1.1. Busca por escopo (5) **[10]** $(2*10 + 4*10 + 4*10)/10 = 10$
 - 2.1.1.1.1. Busca de pessoas (2) [10]
 - 2.1.1.1.2. Busca por cursos (4) [10]
 - 2.1.1.1.3. Busca por disciplinas acadêmicas (4) [10]
 - 2.1.1.2. Busca global (5) **[10]** $(10*10)/10 = 10$
 - 2.1.1.2.1. Informações para refinamento (10) [10]
 - 2.2. Tarefas do domínio relativo ao estudante (8) **[5,48]** $(9*5,44 + 1*5,85)/10 = 5,48$
 - 2.2.1. Conteúdo relevante (9) **[5,44]** $(2*7,55 + 2*4,4 + 2*3,31 + 2*3,9 + 2*8,06)/10 = 5,44$
 - 2.2.1.1. Informações sobre unidade acadêmicas (2) **[7,55]** $(3*10 + 7*6,5)/10 = 7,55$
 - 2.2.1.1.1. Índice de disciplinas acadêmicas (3) [10]
 - 2.2.1.1.2. Subsite de disciplinas acadêmicas (7) [6,5]
 - 2.2.1.2. Informações diversas (2) **[4,4]** $(10*4,4)/10 = 4,4$
 - 2.2.1.2.1. Requerimentos (10) **[4,4]** $(4*6,5 + 4*3 + 2*3)/10 = 4,4$
 - 2.2.1.2.1.1. Entrada (4) [6,5]
 - 2.2.1.2.1.2. Acompanhamento (4) [3]
 - 2.2.1.2.1.3. Download de formulários (2) [3]
 - 2.2.1.3. Qualidade do conteúdo (2) **[3,31]** $(1*3 + 1*10 + 2*0 + 6*3,35)/10 = 3,31$
 - 2.2.1.3.1. Qualidade do índice (1) [3]
 - 2.2.1.3.2. Qualidade do conteúdo (1) **[10]**

Figura 5.18 – Continuação

- 2.2.1.3.3. Qualidade das informações dos cursos (2) **[0]** $(7*0 + 3*0)/10 = 0$
 - 2.2.1.3.3.1. Ementas (7) [0]
 - 2.2.1.3.3.2. Formas de avaliação (3) [0]
- 2.2.1.3.4. Descrição dos cursos (6) **[3,35]** $(1*0 + 3*0 + 2*2 + 3*6,5 + 1*10)/10 = 3,35$
 - 2.2.1.3.4.1. Comentários (1) [0]
 - 2.2.1.3.4.2. Plano de ensino (3) [0]
 - 2.2.1.3.4.3. Calendário (2) **[2]** $(2*10 + 6*0 + 1*0 + 1*0)/10 = 2$
 - 2.2.1.3.4.3.1. Das aulas (2) [10]
 - 2.2.1.3.4.3.2. Das provas (6) [0]
 - 2.2.1.3.4.3.3. Dos trabalhos (1) [0]
 - 2.2.1.3.4.3.4. Diversos (1) [0]
 - 2.2.1.3.4.4. Cursos de pós-graduação (3) **[6,5]** $(4*6,5 + 3*6,5 + 3*6,5)/10 = 6,5$
 - 2.2.1.3.4.4.1. Especialização (4) [6,5]
 - 2.2.1.3.4.4.2. Mestrado (3) [6,5]
 - 2.2.1.3.4.4.3. Doutorado (3) [6,5]
 - 2.2.1.3.4.5. Organogramas (1) **[10]** $(5*10 + 5*10)/10 = 10$
 - 2.2.1.3.4.5.1. Organograma global (5) [10]
 - 2.2.1.3.4.5.2. Organograma departamental (5) [10]
- 2.2.1.4. Informação dos serviços dos estudantes (2) **[3,9]** $(2*6,5 + 2*6,5 + 2*0 + 1*0 + 1*0 + 2*6,5)/10 = 3,9$
 - 2.2.1.4.1. Índice de serviços (2) **[6,5]**
 - 2.2.1.4.2. Informações do calendário escolar (2) **[6,5]**
 - 2.2.1.4.3. Informações dos alojamentos (2) **[0]**
 - 2.2.1.4.4. Informações culturais (1) **[0]**
 - 2.2.1.4.5. Informações esportivas (1) **[0]**
 - 2.2.1.4.6. Informações sobre congressos (2) **[6,5]**
- 2.2.1.5. Informações da infra-estrutura acadêmica (2) **[8,06]** $(7*7,65 + 3*9)/10 = 8,06$
 - 2.2.1.5.1. Bibliotecas (7) **[7,65]** $(7*10 + 2*0 + 1*6,5)/10 = 7,65$
 - 2.2.1.5.1.1. Busca de exemplares (7) [10]
 - 2.2.1.5.1.2. Reserva de exemplares (2) [0]
 - 2.2.1.5.1.3. Informações gerais (1) [6,5]
 - 2.2.1.5.2. Laboratórios (3) **[9]** $(6*10 + 3*10 + 1*0)/10 = 9$
 - 2.2.1.5.2.1. Reserva (6) [10]
 - 2.2.1.5.2.2. Informações gerais (3) [10]
 - 2.2.1.5.2.3. Resultados de pesquisa (1) [0]
- 2.2.2. Serviços on-line (1) **[5,85]** $(1*10 + 7*6,5 + 1*0 + 1*3)/10 = 5,85$
 - 2.2.2.1. Informação da grade curricular (1) [10]
 - 2.2.2.2. Serviços Web (7) [6,5]
 - 2.2.2.3. Serviços FTP (1) [0]
 - 2.2.2.4. Serviços de newsgroup (1) [3]

Figura 5.18 – Continuação

Para finalizar basta efetuar o cálculo:

Nota Final = $(5 * 3,39 + 5 * 6,38) / 10 = 4,89$

1. Usabilidade (5) [3,39]

2. Funcionalidade (5) [6,38]

Isto significa que, segundo a metodologia REQE o site acadêmico avaliado recebeu a nota 4,89.

A avaliação do uso da metodologia REQE neste exemplo indicou benefícios e dificuldades. A partir da avaliação, é possível identificar uma série de falhas (na verdade requisitos que não foram levados em consideração no documento de requisitos ou mal especificados), que podem ser corrigidas no início do processo de desenvolvimento, acarretando a construção de um produto de melhor qualidade.

No entanto, a metodologia mostrou-se de certa forma trabalhosa, carecendo da automatização de algumas fases ou passos. Temos, também, que melhorar a forma de atribuir pesos e escores de conformidades aos nós, para que estas atividades sejam feitas de maneira mais objetiva.

6 Conclusões e Trabalhos Futuros

Este breve capítulo lista as conclusões deste trabalho e aponta futuras atividades a serem desenvolvidas.

6.1 Principais Contribuições

Este capítulo mostra as conclusões obtidas durante o desenvolvimento desse trabalho, assim como as principais contribuições que ele fornece para a área de avaliação de qualidade. Serão mostrados alguns possíveis trabalhos futuros.

A metodologia REQE (Requirements Engineering Quality Evaluation) descrita neste trabalho além de avaliar a qualidade de um produto na fase de requisitos, pode ser utilizada durante o processo de desenvolvimento com o intuito de incrementar a qualidade do documento de requisitos, o que indubitavelmente levará a uma redução dos erros no produto final.

A proposta aqui apresentada pode ser muito útil se empregada em empresas que desenvolvem sistemas para Web, pelo fato de possibilitar a identificação de erros numa fase inicial do processo de desenvolvimento. É certo que a metodologia REQE apenas se aplica nos casos em que existe um documento de requisitos bem definido. Caso isso não faça parte do processo de desenvolvimento, esse tempo adicional para elaborá-lo será, indiscutivelmente, compensado por uma maior qualidade final do produto, além de uma documentação mais completa do software como um todo.

Com o desenvolvimento da proposta procuramos criar um processo genérico o suficiente para atender diversos domínios e tipos de aplicações Web. Podemos citar como contribuições deste trabalho:

- Reforço da importância da avaliação de qualidade no contexto de aplicações Web, com o objetivo de melhor compreender e assim incrementar a qualidade das mesmas;
- Levantamento do estado da arte da avaliação de qualidade de aplicações Web;

- Reconhecimento da importância da Engenharia do Domínio no processo de melhoria de qualidade dos sistemas;
- Proposta de uma metodologia para medição da qualidade de aplicações Web na fase de requisitos: **REQE** (Requirements Engineering Quality Evaluation);
- Validação da metodologia através do seu uso em um exemplo.

6.2 Trabalhos Futuros

Inicialmente, pretende-se realizar vários estudos de casos mais detalhados onde seja possível acompanhar o uso da metodologia REQE em outras situações reais.

Esse trabalho está centrado nas características de qualidade de artefatos WEB. No entanto, outros indicadores como custo, ou relação custo-benefício podem ser de interesse para avaliação e comparação de sistemas em geral. Como continuação deste trabalho podemos investir na agregação de árvores de custo associadas às árvores de características, subcaracterísticas e atributos de qualidade. Assim, teríamos como identificar o custo da apreciação de cada nó da árvore no produto final. Também poderíamos investigar como tratar de definição de preferências [HUI03].

Um outro trabalho a ser desenvolvido seria analisar a viabilidade da especificação de uma árvore de características, subcaracterísticas e atributos de qualidade mais genérica, que pudesse inclusive ser aplicada a vários domínios de aplicação, bem como para diferentes perfis de usuários. Para a avaliação de qualidade seria necessário estender essa árvore genérica, com atributos que fossem específicos ao domínio em questão.

Um trabalho interessante seria acompanhar todo o desenvolvimento de um projeto Web e aplicar as metodologias REQE e *Web-site QEM*, cada uma na sua respectiva fase, isto é, a

REQE na fase de requisitos e a *Web-site QEM* na fase operativa. Os erros previamente identificados com a nossa metodologia poderiam estar presentes na avaliação de qualidade efetuada pela metodologia Web-site QEM, uma vez que há um “gap” entre as fases de desenvolvimento sobre as quais se aplicam as metodologias. Isto quer dizer que, mesmo partindo de um documento de requisitos bem elaborado e especificado, é necessário investir na qualidade durante as outras fases do processo de desenvolvimento. Havendo este acompanhamento acreditamos que os erros já identificados e corrigidos na fase de requisitos não mais apareceriam na fase operativa. Vale ressaltar que deveria haver uma coerência na especificação da árvore de características e atributos de qualidade utilizada em ambas as metodologias.

Também pretendemos desenvolver ferramentas para apoiar a metodologia REQE. Este trabalho poderia ser integrado a ferramentas como WebQEM_Tool [OLSINA01a] e WebMA [OLSINA01b], capazes de suportar a automatização de métricas Web. Portanto, a meta seria o desenvolvimento de uma aplicação colaborativa, centrada na Web, onde fosse possível integrar o documento de requisitos em um formato pré-determinado com a árvore de características, subcaracterísticas e atributos de qualidade daquele domínio da aplicação em questão. Assim, o resultado da avaliação seria apresentado de forma eficiente e automática.

Referências Bibliográficas

- [ABRAHÃO02] Abrahao S., Olsina L., and Pastor O. *Towards the Quality Evaluation of Functional Aspects of Operative WebApps*. 1st International ER Workshop on Conceptual Modeling Quality (IWCMQ'02), Tampere, Finland, Springer-Verlag. October 2002
- [ALENCAR99] Alencar, F. M. R.; Castro, J. F.B.; "Integrating Early and Late-Phase Requirements: a Factory Case Study". Proceedings of XIII Brazilian Symposium on Software Engineering - SBES99. Florianópolis, SC, Brasil. pp 47-61. Outubro 1999.
- [ALVES01] Alves, C. F.; Castro, J. F. B. CRE: A Systematic Method for COTS Components Selection. SBES02, XV Simpósio Brasileiro de Engenharia de Software, Rio de Janeiro. Outubro, 2001.
- [ALVES02] Alves, C. F. Seleção de Produtos de Software Utilizando Uma Abordagem Baseada em Engenharia de Requisitos. Dissertação de Mestrado, Centro de Informática, Universidade Federal de Pernambuco. 2002.
- [ARANGO94a] Arango, G. Domain Analysis Methods, In: WORKSHOP ON SOFTWARE ARCHITECTURE, 1994, Los Angeles: USC Center for Software Engineering, 1994.
- [ARANGO94b] Arango, G., Prieto-díaz, R. Domain Analysis Concepts and Research Directions. In: WORKSHOP ON SOFTWARE ARCHITECTURE, 1994, USC Center for Software Engineering, Los Angeles, EUA. Los Angeles, 1994.
- [ARAÚJO01] Araújo, A. C. M. "Framework de Análise e Projeto Baseado no RUP para o Desenvolvimento de Aplicações Web". Dissertação de Mestrado, Centro de Informática, Universidade Federal de Pernambuco. 2001.
- [BARESI01] Baresi, L., Garzotto, F., Paolini, P. "Extending UML for Modeling Web Applications", 34 th Hawaii International Conference on System Sciences, USA, 2001.
- [BERGMANN00] Bergmann, U.; Leite, J.C.S.P. From Application Domains to Executable Domains: Achieving Reuse with a Domain Network. Sixth International Conference on Software Reuse 6. Berlin. pp.41-57. 2000.
- [BEVAN97] Bevan, N., "Usability Issues in Web Site Design", Proceed. of Human-Computer Interaction Int'l Conference. San Francisco. Elsevier Press. 1997.
- [BOEHM76] Boehm B., Software engineering. IEEE Transactions on Computer. 1976.
- [CHUNG95a] Chung, L. Nixon, B. e Yu, E. Using Non-Functional Requirements to Systematically Support Change. Proceedings IEEE International Symposium on Requirements Engineering. Mar. 1995.

[CHUNG95b] Chung, L. e Nixon, B. Dealing with Non-Functional Requirements: Three Experimental Studies of a Process-Oriented Approach. 17th International Conference on Software Engineering. Apr. 1995.

[CHUNG00] Chung, L. Nixon, B. Yu, E. e Mylopoulos, J. Non-Functional Requirements in Software Engineering. Kluwer Academic Publisher, 2000.

[CLEOMONS91]. E. K. Clemons and M. C. Row, Information Technology at Rosenbluth Travel: Competitive advantage in a rapidly growing global service company, J. Manage, Info. 8, 2, pág. 53-79. 1991.

[CORNWELL96] Cornwell, P.C. HP Domain Analysis: Producing Useful Models for Reusable Software. Hewlett-Packard Journal, 1996.

[CYSNEIROS99] Cysneiros, L. M. e Leite, J. C. Integrating Non-Functional Requirements into Data Modeling. 4th International Symposium on Requirements Engineering. Limerick, Ireland. Jun. 1999.

[CYSNEIROS01] Cysneiros, L. M. Requisitos Não-Funcionais: da Elicitação ao Modelo Conceitual. Tese de Doutorado, Departamento de Informática, PUC-RJ. Fev. 2001.

[DARDENE93] Dardene, A.; Lamsweerde, A.; Fickas, S. Goal - directed Requirements Acquisition. Science of Computer Programming, v. 20, p. 3-50, 1993.

[DAVIS93] Davis A.M. Software Requirements: Objects, Functions and States. Prentice Hall, Englewood Cliffs, NJ. 1993.

[DIDIER03] Didier A. C. V. B. WRE-Process: Um Processo de Engenharia de Requisitos Baseado no RUP. Dissertação de Mestrado, Centro de Informática, Universidade Federal de Pernambuco. 2003.

[DOBSON91] Dobson, J. A Methodology for Analyzing Human Computer-related Issues in Secure Systems. International Conference on Computer Security and Integrity in our Changing World, 151-170, Espoo, Finland, 1991.

[DOBSON94a] Dobson, J. E and Strems, R. Organizational Requirements Definition for Information Technology. IEEE International Conference on Requirements Engineering. Los Alamitos, California: IEEE Computer Society Press. pp. 158-165. April 1994.

[DOBSON94b] Dobson, J. E.; Blyth, A.J.C., Chudge, J. and Strems, R.. The ORDIT approach to organisational requirements . In M. Jirotko and J. Goguen (Eds.), Requirements Engineering, Academic Press Ltd. pp. 87-106. London, 1994.

[DUJMOVIC82] Dujmovic, J.J.; Elnicki, R., "A DMS Cost/Benefit Decision Model: Mathematical Models for Data Management System Evaluation, Comparison, and Selection ", National Bureau of Standards, Washington D.C. N° GCR 82-374. NTIS N° PB 82-170150. 1982.

[DUJMOVIC96] Dujmovic, J.J., "A Method for Evaluation and Selection of Complex Hardware and Software Systems", The 22nd Int'l Conference for the Resource Management and Performance Evaluation of Enterprise CS. CMG 96 Proceedings, Vol. 1, pp.368-378. 1996.

[DUKIC99] Dukic, L. Non-Functional Requirements for COTS Software Components. Workshop Ensuring Successful COTS Development, Los Angeles, EUA. Mai. 1999.

[ESCALONA03] Escalona, M.J. & Koch; N. Ingeniería de Requisitos en Aplicaciones para la Web: Un Estudio Comparativo. 6º Workshop Ibero-Americano de Engenharia de Requisitos e Ambientes de Software (IDEAS), Asunción, Paraguay. 2003.

[FENTON97] Fenton, N.E., Pfleeger, S.L., "Software Metrics: a Rigorous and Practical Approach", 2nd Ed., PWS Publishing Company. 1997.

[FORTES02] Fortes, R. P. M., Silva, E. A., Paiva, D. M. B. "Utilizando a Norma ISO/IEC 14598-5 na Avaliação da Qualidade de Hiperdocumentos Web", Simpósio Brasileiro de Qualidade de Software, Gramado - RS. 2002.

[FREITAS97] Freitas, F. G. Aplicando Técnicas de Reuso de Software na Construção de Ferramentas de Engenharia Reversa. Dissertação de Mestrado, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro. 1997.

[FUCHS96] Fuchs, N. E.; Schwitter, R. Attempto Controlled English (ACE). In: The First International Workshop on Controlled Language Applications (CLAW 96), Würzburg, Germany. Proceedings... p. 211-218. 1996.

[FURANO97] Furano, F.; Orsini, R.; Celentano, A., "Museum-ondemand: dynamic management of resources in World Wide Web museums", Hypertext and Hypermedia: Products, Tools, Methods (H2PTM'97), Hermes Editorial, Paris, Fr, V. 1, Nº 2-3-4/97 pp. 115,124. 1997.

[GARZOTTO93] Garzotto, F.; Schwabe, D.; Paolini, P., "HDM, a model based approach to Hypermedia Application Design ", ACM Transaction on Information System, Vol. 11, 1, Jan 93, pp. 1-26. 1993.

[GARZOTTO95] Garzotto, F.; Mainetti, L.; Paolini, P., "Hypermedia Design, Analysis, and Evaluation Issues", CACM 38,8 (Ago-95); pp 74-86. 1995.

[GARZOTTO97a] Garzotto, F.; Mainetti, L.; Paolini, P., "Designing Modal Hypermedia Applications", The Eighth ACM Conference on Hypertext, Southampton, England, pp 38-47. 1997.

[GARZOTTO97b] Garzotto, F.; Matera, M., A Systematic Method for Hypermedia Usability Inspection, The New Review of Hypermedia and Multimedia, 3, 1997, pp. 39-65. 1997.

[GELLERSON97] H. W. Gellerson, R. Wicke and M. Gaedke. WebComposition: An Objected-Oriented Support System for the Web Engineering Life Cycle. Sixth International WWW Conference, California, Santa Clara, April, 1997.

[GIBBS94] Gibbs, W., "Software's chronic crisis", Scientific American. 1994.

[GILB69] Gilb, T., "Weighted Ranking by Levels", IAG Journal, Vol 2 (2), pp. 7-22. 1996.

[GILB76] Gilb, T., "Software Metrics", Chartwell-Bratt, Cambridge, MA. 1976.

[GINIGE01] Ginige, A., Murugesan, S., Web Engineering: An Introduction, IEEE Multimedia, V8, N°1, pp.14-18. Jan/Mar, 2001.

[GRADY94] Grady, R.B., "Successfully Applying Software Metrics", IEEE Computer, 27 (9), pp. 18-25. 1994

[GUIZZARDI00] Guizzardi, G. Uma Abordagem Metodológica de Desenvolvimento para e com Reuso, Baseada em Ontologias Formais de Domínio. Dissertação de Mestrado, Universidade Federal do Espírito Santo. 2000.

[HECK98]. E. van Heck and P. Vervest, How Should CIOs Deal With Web-Based Auctions ?, Communications of the ACM, vol. 41, no. 7, pág. 99-100. July, 1998.

[HOLCK01] Holck, J., Clemmensen, T., "What makes web-development different?", 24th. Information Systems Research Seminar in Scandinavia, 2001.

[HUI03] Hui, B.; Liaskos, S., Mylopoulos, J., "Requirements Analysis for Customizable Software: A Goals-Skills-Preferences Framework". 11th IEEE Internacional Requirements Engineering Conference, Monterey Bay, California USA. 2003.

[IBEST04] Prêmio iBest, disponível em <http://premio.ibest.com.br/>. Último acesso em 01/05/2004. 2004.

[IEEE1061] IEEE Std 1061, "IEEE Standard for a Software Quality Metrics Methodology", IEEE Computer Society Press. 1992.

[IEEE830] IEEE Std 830, IEEE Recommended Practice for Software Requirements Specifications. 1998.

[ISAKOWITZ95] Isakowitz, T.; Stohr, E.A.; Balasubramanian; "RMM: A methodology for Structured Hypermedia Design", Communications of the ACM, 38(8): 34-44, 1995.

[ISO9126] ISO/IEC 9126, International Standard Information technology – Software product evaluation – Quality characteristics and guidelines for their use. 1991.

[ISO14598-5] ISO/IEC 14598-5: Information technology – Software 1998. product evaluation -- Part 5: Process for evaluators.

[IVORY01] Ivory, M.Y., Hearst, M.A.: State of the Art in Automating Usability Evaluation of User Interfaces. ACM Computing Surveys, 33,41--47. Acessível em <http://webtango.berkeley.edu/papers/ue-survey/ue-survey.pdf>. Último acesso 01/04/2004. Dezembro 2001.

[JONES03] Jones, W.L., “Distinguish Between Application and Interface: An Alternative Approach to User Interface Development”; 2002; Artigo disponível em The Rational Edge: <http://www.therationaledge.com>, último acesso em 25/04/2004. 2003.

[KAIYA02] Kaiya, H.; Horai, H.; Saeki, M.; “AGORA: Attributed Goal-Oriented Requirements Analysis Method”. 10th IEEE Internacional Requirements Engineering Conference, University of Essen. 2002.

[KANG90] Kang, K., et al. Feature-Oriented Domain Analysis (FODA) Feasibility Study (CMU/SEI-90-TR-21, ADA 235785). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1990.

[KAPLAN00] Kaplan, G.N., Hadad, G.D.S, Doorn, J.H., Leite, J.C.S.P, “ Inspección del Lexico Extendido del Lenguage” , 3th. Workshop de Engenharia de Requisitos, Rio de Janeiro, Brasil, 2000.

[KIRAKOWSKI94] Kirakowski, J., Corbett, M., “SUMI: the Software Usability Measurement Inventory”, Brit J Ed Technology, 23. 1994.

[KIRAKOWSKI98] Kirakowski, J., Cierlik, B., “Measuring the Usability of Web Sites”, Human Factors and Ergonomics Society Annual Conference, Chicago, US. 1998.

[KOTONYA96] Kotonya, G.; Sommerville, I. Requirements Engineering With Viewpoints. Software Engineering Journal, v. 11, n 1, p. 5-11, jan. 1996.

[KOTONYA97] Kotonya, G. e Sommerville, I. Requirements Engineering – Processes and Techniques. John Willy & Sons, 1997.

[LADKIN94] Ladkin, P.B. ;Leue, S. What Do Message Sequence Charts Mean?, in: R.L. Tenney, P.D. Amer, M.U. Uyar (eds.), Formal Description Techniques VI, IFIP Transactions C, Proc. of the 6th International Conference on Formal Description Techniques, North-Holland, p. 301 - 316, 1994

[LAFUENTE01] Lafuente, G.; González Rodríguez, J.; Olsina, L.; Automatizando Métricas Web, Anais 4to Encuentro para la Calidad en las Tecnologías de la Información y las Comunicaciones, (QUATIC), Lisbon, Portugal, pp. 17-24. 2001.

[LANDAUER95] Landauer, T.K. “The trouble with Computers: Usefulness, Usability and Productivity”, MIT Press, 1995.

[LEITE83] Leite, J.C.S.P. O Enfoque Social na Análise de Sistemas. Anais do XVI Congresso Nacional de Informática, São Paulo. pp. 272-279. 1983.

[LEITE93a] Leite, J.C.S.P.; Prado, A. F; Sant'anna M., “Draco-Puc: A Case Study on Software Re-engineering.” Position papers of the 2nd International Workshop on Software Reusability, SWT Memo Nr. 69, Universitat Dortmund, ISSN 093-7725, Pags. 121—124. 1993.

[LEITE93b] Leite J.C.S.P.; Franco, A.P.M. A Strategy for Conceptual Model Acquisition. In: International Symposium on Requirements Engineering, 1., SanDiego, Ca. Proceedings... IEEE Computer Society Press, p. 243-246. 1993.

[LEITE94] Leite, J.C.S.P.; Sant'Anna, M; Gouveia, F. Draco-Puc: A Technology Assembly for Domain Oriented Software Development. Proceedings of the Third International Conference on Software Reuse. IEEE Computer Society Press, Pags. 94-- 100. 1994.

[LEITE97] Leite, J, Gustavo Rossi, Vanesa Maiorana, Federico Balaguer, Gladys Kaplan, Graciela Hadad and Alejandro Oliveros, Enhancing a Requirements Baseline with Scenarios, Requirements Engineering Journal: Vol. 2, N.4, pp. 184 - 198, Springer-Verlag, London. 1997.

[LIMA03] Lima, F.; Schwabe, D., “Modeling Applications for the Semantic Web”. ICWE02, Internacional Conference on Web Engineering Oviedo, Spain, pp. 417-426. 2003.

[LOHSE98] Lohse, G.; Spiller, P., "Electronic Shopping", CACM 41,7 (Jul-98); pp 81-86. 1998.

[LOPES02] Lopes, P. S. N. D. “Uma Taxonomia da Pesquisa na Área de Engenharia de Requisitos”. Dissertação de Mestrado, Universidade de São Paulo. 2002.

[LOUCOPOULOS95] Loucopoulos, P. e Karakostas, V. System Requirements Engineering. London McGraw-Hill, 1995.

[McCALL77] McCall, J.A; Richards, P.K.; Walters, G.F.; “Factors in Software Quality“, RADC TR-77-369. 1977.

[MILLER70] Miller, J.R.; “Professional Decision-Making”, Praeger Publisher. 1970.

[MYLOPOULOS99] Mylopoulos, J.; Chung, L.; Yu, E. From Object-Oriented to Goal-Oriented Requirement Analysis. Communications of the ACM, v. 42, n. 1, Jan. 1999.

[MYLOPOULOS01] Mylopoulos, J.; Kolp, M., and Castro, J. UML for Agent Oriented Software Development: the Tropos Proposal In Proceedings of the Fourth International

Conference on the Unified Modeling Language (<<UML>> 2001), Toronto, Canada, October 2001.

[NEIGHBORS81] Neighbors J. Software Construction Using Components. Tese Doutorado) - Universidade da Califórnia, Irvine, EUA. 1981.

[NEIGHBORS84] Neighbors, J. M.. "The Draco Approach to Constructing Software from Reusable Components". IEEE Transactions on Software Engineering, SE-10(5):564-574, Sep. 1984.

[OLSINA98] Olsina, L., "Toward Improvements in Hypermedia Process Modeling", Proceed. at the Symposium of Software Technology, 27 JAIIO, Bs As., pp. 219-226. 1998.

[OLSINA99] Olsina, L.; Lafuente, G.J.; Godoy, D; Rossi, G.; Assessing the Quality of Academic Websites: a Case Study, New Review of Hypermedia and Multimedia (NRHM) Journal, Taylor Graham Publishers, UK, Vol. 5, pp. 81-103. 1999.

[OLSINA00a] Olsina, L.; Lafuente, G.J.; Rossi, G., E-commerce Site Evaluation: a Case Study, LNCS 1875 of Springer, 1st International Conference on Electronic Commerce and Web Technologies, EC-Web 2000, London, UK, pp. 239-252. 2000.

[OLSINA00b] Olsina, L., Metodologia Quantitativa para a Avaliação e Comparação de Qualidade de Sites Web, tese de doutorado defendida em abril, Facultad de Ciencias Exactas, UNLP, La Plata, Argentina. 2000.

[OLSINA01a] Olsina L., Papa, M.F., Souto, M.E., Rossi, G.; "Providing Automated Support for the Web Quality Evaluation Methodology", Proceed. of the Fourth Workshop on Web Engineering, at the 10th International WWW Conference, Hong Kong, pp. 1-11. 2001.

[OLSINA01b] Olsina L., González J., Lafuente G., Pastor O. *Towards Automated Web Metrics*. 8º Workshop de Qualidade de Software, IME-RJ. Rio de Janeiro, Brasil. Outubro 2001.

[PASTOR03] Pastor, O.; Fons, J.; Pelechano, V., *OOWS: A Method to Develop Web Applications from Web-Oriented Conceptual Models*, pp. 65 - 70, International Workshop on Web Oriented Software Technology (IWWOST), Is International, 3rd, 2003 - July, Oviedo (Spain), 2003.

[PINHEIRO03] Pinheiro, F.A.C., " Técnicas de Análise e Elicitação de Requisitos para a Web", disponível em <http://www.cic.unb.br/~facp/facphome.html>, último acesso em 25/04/2004. 2003.

[PORTER85]. M. Porter and V. Millar, How information gives you competitive advantage. Harvard Bus. Ver., pág. 149-160. July-Aug, 1985.

[POWELL98] Powell, T.; Jones, D; Cutts, D., “Web Site Engineering:Beyond Web Page Design”, Prentice Hall PTR. 1998.

[PRESSMAN02] Pressman, Roger S.; “Engenharia de Software”. McGraw-Hill, 2002.

[REIS02] Reis, T. P. C.; Castro, J. F. B.; Olsina, L.; Medição da Qualidade de um Projeto Web na Fase de Requisitos, SBES02, XVI Simpósio Brasileiro de Engenharia de Software, Gramado, outubro de 2002.

[ROMAN85] Roman, G.C. A Taxonomy of current issues in requirements engineering. IEEE Computer, 18, 14-23. 1985.

[ROSENFELD98] Rosenfeld, L., Morville, P., “Information Architecture for the WWW”, Reilly. 1998.

[ROSSI96] Rossi, G., “Uma metodologia Orientada a Objetos para o projeto de aplicativos Hipermissão”, Tese de Doutorado, PUC-RIO, Rio de Janeiro, Brasil. 1996.

[SÁNCHEZ02] Sánchez, J., “Validación de requisitos de usuario mediante técnicas de transformación de modelos y prototipación automática de interfaces de usuario”, Phd Tese, Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, Valencia, Novembro, 2002.

[SANT'ANNA98] Sant'anna, M.; Leite, J. C. S. P; Prado, A. F., “Draco-PUC: a Workbench for Developing Transformation-Based Software Generators.” Proceedings of the IEEE International Conference on Software Engineering, Volume II, IEEE Computer Society Press, pp. 135—138. 1998.

[SANTOS98] Santos, I. and Carvalho, J. The Structural, Social, Political and Symbolic Dimensions of Computer-Based Systems Requirements. In Proceedings of the 4th International Workshop on Requirements Engineering: Foundations of Software Quality-REFSQ'98, pp. 121-132, Pisa, Italy. June 1998.

[SANTOS00] Santos, M. O Uso da Análise de Requisitos como Estratégia para Apoio ao Projeto de Banco de Dados Distribuídos. Dissertação de Mestrado, Centro de Informática, Universidade Federal de Pernambuco. 2000.

[SCHNEIDERMAN98] Schneiderman, B., Designing the User Interface: Strategies for Effective Human-Computer Interaction, 3rd Edition, Reading, MA: Addison Wesley. 1998.

[SCHWABE95] Schwabe, D.; Rossi, G., “The Object-Oriented Hypermedia Design Model”, Comm. ACM 38,8 (Aug 95) pp. 45-46. 1995.

[SHIPLE03] Shiple, J. “ Information Architecture Tutorial” ; Artigo disponível em http://hotwired.lycos.com/webmonkey/design/site_building/tutorials/tutorial1.html, último acesso em 25/04/2004. 2003.

- [SIMOS96] Simos, M., et al. Software Technology for Adaptable Reliable Systems (STARS) Organization Domain Modeling (ODM) Guidebook Version 2.0 (STARS-VC-A025/001/00). Manassas, VA: Lockheed Martin Tactical Defense Systems, 1996.
- [SOMMERVILLE03] Sommerville, I; “Engenharia de Software”; tradução André Ribeiro; 6a. Edição, Addison Wesley; São Paulo, ISBN 85-88639-07-6, 2003.
- [SOUSA03] Sousa G., Silva I., Castro, J. “Adapting the NFR Framework to Aspect-Oriented Requirements Engineering”. SBES03, XVII Simpósio Brasileiro de Engenharia de Software, Manaus, outubro de 2003.
- [SOUZA02] Souza, R. A. C. “Uma Extensão do Fluxo de Análise e Projeto do RUP para o Desenvolvimento de Aplicações Web”. Dissertação de Mestrado, Centro de Informática, Universidade Federal de Pernambuco. 2002.
- [SOUZA03] Souza, R.; Vasconcelos A. “Uma Extensão do Fluxo de Análise e Projeto do RUP para o Desenvolvimento de Aplicações Web”. SBES03, XVII Simpósio Brasileiro de Engenharia de Software, Manaus, outubro de 2003.
- [STEPHANIDIS98] Stephanidis, C.; Akoumianakis, D.; Sfyraakis, M. and Paramythis, A. Universal accessibility in HCI: Process-oriented design guidelines and tool requirements. In Proceedings of the 4th ERCIM Workshop on User Interfaces for All, Stockholm, Sweden, October 1998.
- [SUTCLIFFE98] Sutcliffe, A. G and Minocha, S. Scenario-based Analysis of Non-Functional Requirements. In Proc. of the fourth International Workshop on Requirements Engineering: Foundations of Software Quality- REFSQ'98, pp. 219-234, Pisa, Italy. June 1998.
- [THURING95] Thüring, M.; Hannemann, J.; Haake, J., “Hypermedia and Cognition: Designing for Comprehension”, Comm. ACM 38, 8 (Aug 95) pp. 57-66. 1995.
- [TILSON98] Tilson, R., Dong, J., Martin, S., Kieke, E., “Factors and Principles Affecting the Usability of Four E-commerce Sites”, 4th Conference on Human Factors & the Web, Baking Ridge, NJ, U S. 1998.
- [TURAU98]. V. Turau, What Practices Are Being Adopted on the Web, IEEE Computer, pág. 106-108. May, 1998.
- [VASCONCELOS99] VASCONCELOS, A. M. L., Neves. A Design Model for Hypermedia Graphic Interfaces In: 8th International Conference on Human-Computer Interaction - HCI'99, 1999, Munique. Anais do HCI'99, 1999.
- [VASCONCELOS02] Vasconcelos, A. M. L., VAZ, M. S. M. G., Souza, Salgado Um Modelo de Metadados na Descrição de Objetos Multimídia In: Conferencia Ibero-Americana en Sistemas, Cibernética e Informática (CISCI2002), 2002, Orlando. Anais do CISCI2002. v.II, 2002.

[VASUDEVAN96]. S. Vasudevan and Y. Wang, Technology adoption in the presence of knowledge barriers: The Case of the World Wide Web. In Proceedings of the 17th International Conference on Information Systems, Dec. 1996.

[W3C99] WWW Consortium, W3C Working Draft, "WAI Accessibility Guidelines: Page Authoring", W3C, W3C Working Draft, "WAI Accessibility Guidelines: Page Authoring", <http://www.w3c.org/TR/WD-WAI-PAGEAUTH/>, último acesso em 25/04/2004. 1999.

[WEST01] Web-Oriented Software Technology.
Disponível em www.dsic.upv.es/~west2001/, último acesso em 01/05/2004. 2001.

[WHITE63] White, D.R.J.; Scott, D.L; Schultz, "POED: a Method of Evaluating System Performance", IEEE TEM, Dec. 1963.

[YEH90] Yeh R. T. and Ng P. A. Software requirements - a management perspective. IEEE Computer Society Press, Los Alamitos, CA. 1990.

[YU94] Yu, E. and Mylopoulos, J. Understanding 'Why' in Software Process Modelling, Analysis and Design. Sixteenth International Conference on Software Engineering. Sorrento, 1994.

[YU97] Yu, E. Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. Proceedings of the 3th IEEE International Symposium on Requirements Engineering - RE97, pp.226-235, Jan. 1997.

[YU98] Yu, E. and Mylopoulos, J. Why Goal-Oriented Requirements Engineering. In Proceedings of the 4th International Workshop on Requirements Engineering: Foundations of Software Quality, REFSQ'98, pp. 15-22, Pisa, Italy. June 1998.

[ZANLORENCI02] Edna Pacheco Zanlorenzi, Robert Carlisle Burnett. "Certificação de Qualidade em Engenharia de Requisitos". Simpósio Brasileiro de Qualidade de Software, Gramado – RS. 2002.

[ZAVE97] Zave, P.; Jackson, M. Four Dark Corners of Requirements Engineering. ACM Transactions on Software Engineering and Methodology, v. 6, n. 1, p. 1-30, 1997.

Apêndice A - Documento de Requisitos Proposto

Esse apêndice apresenta um modelo (template) para criação do documento de requisitos. Todas as suas seções são explicadas, facilitando o seu preenchimento.

<Nome do Projeto> Documento de Requisitos

Versão <1.0.0>

[Este template foi criado para auxiliar no registro das necessidades dos usuários, dos requisitos funcionais e não-funcionais do projeto.]

Os textos que aparecem em itálico e entre colchetes são explicações para ajudar no preenchimento de cada seção e devem ser apagados no documento final.

Preencha nessa página inicial o nome do projeto e a versão do documento. A versão inicial é a 1.0.0 e na medida em que vão sendo feitas revisões e alterações, documentadas na seção Histórico, a versão vai sendo alterada para 1.0.1, 1.0.2, etc.]

Histórico de Revisão

Data	Versão	Descrição	Autor

Sumário

1.	Introdução	.
1.1	Finalidade	.
1.2	Escopo	.
1.3	Definições, Acrônimos, e Abreviações	.
1.4	Referências	.
1.5	Visão geral	.
2.	Contextualização do Produto	.
2.1	Oportunidade de Negócio	.
2.2	Descrição do Problema	.
3.	Descrição dos <i>Stakeholders</i>	.
3.1	Resumo dos <i>Stakeholders</i>	.
3.2	Ambiente do Usuário	.
3.3	Necessidades chave dos <i>Stakeholders</i>	.
4.	Visão Geral do Produto	.
4.1	Perspectiva do Produto	.
4.2	Resumo das características do produto	.
4.3	Premissas e Dependências	.
5.	Requisitos Funcionais do Produto	.
5.1	<Funcionalidade 1>	.
5.2	<Funcionalidade 2>	.
6.	Precedência e Prioridade	.
7.	Requisitos Não-Funcionais do Produto	.
7.1	Requisitos de Usabilidade	.
7.2	Requisitos de Confiabilidade	.
7.3	Requisitos de Eficiência	.
7.4	Requisitos de Manutenibilidade	.
7.5	Requisitos de Portabilidade	.
7.6	Outros Requisitos	.
7.6.1	Padrões Aplicáveis	.
7.6.2	Requisitos de Ambiente	.
7.6.3	Requisitos de Suportabilidade	.
7.6.4	Requisitos de Documentação	.
8.	Interfaces	.
8.1	Interfaces de Usuário	.
8.2	Interfaces de Hardware	.
8.3	Interfaces de Software	.
8.4	Interfaces de Comunicação	.
9.	Restrições	.
10.	Recomendações	.

Documento de Requisitos

1. Introdução

1.1 Finalidade

A finalidade deste documento é definir os requisitos funcionais e não-funcionais do sistema, em termos de suas necessidades. O documento contém uma visão geral dos requisitos mais importantes, sendo a base do acordo com o cliente quanto às características de qualidade do sistema.

1.2 Escopo

[Descrição do escopo do documento, a qual projeto está associado, o que mais é afetado por esse documento.]

1.3 Definições, Acrônimos, e Abreviações

[Listar aqui todos os termos, definições, siglas, abreviações utilizadas no documento, possibilitando sua correta interpretação. Essa informação pode ser substituída por uma referência ao documento de Glossário, se este foi criado para o projeto.]

1.4 Referências

Título	Versão	Data	Onde pode ser obtido

[Incluir a identificação de todos os documentos referenciados no presente documento. Para cada documento indicar o título, versão (caso tenha), data e onde o documento pode ser encontrado para consulta.]

1.5 Visão geral

O documento está estruturado da seguinte forma:

- Seção 2 – Contextualização do Produto: descrição do contexto em que surgiu a idéia/necessidade de se fazer o projeto: oportunidade, para resolver qual problema, etc.
- Seção 3 – Descrição dos *Stakeholders*: descrição dos principais envolvidos/interessados no projeto, com suas responsabilidades e forma de contato. São colocadas aqui também as necessidades desse grupo de pessoas com relação ao projeto.
- Seção 4 – Visão Geral do Produto: descrição geral dos benefícios trazidos pelo produto, como ele se relaciona a outros no ambiente em que será implantado e quais as premissas e dependências desse projeto.
- Seção 5 – Requisitos Funcionais do Produto: listagem e breve descrição dos requisitos funcionais do produto (por exemplo, os casos de uso do sistema).

- Seção 6 – Precedência e Prioridade: priorização dos requisitos funcionais listados anteriormente de acordo com as necessidades do cliente e da precedência lógica existente entre elas (se existir).
- Seção 7 – Requisitos Não-Funcionais do Produto: listagem e breve descrição dos requisitos não-funcionais do produto, tais como usabilidade, confiabilidade, eficiência, manutenibilidade, portabilidade etc.
- Seção 8 – Interfaces: descrição das interfaces do sistema, sua comunicação com o mundo externo, em linhas gerais. São descritas as interfaces de usuário, de software, de hardware e de comunicação.
- Seção 9 – Restrições: descrição das restrições do projeto, restrições externas ou outras dependências. São listadas restrições políticas, econômicas e técnicas, se houver.

2. Contextualização do Produto

2.1 Oportunidade de Negócio

[Descrever a oportunidade de negócio que incentivou ou possibilitou a criação do projeto.]

2.2 Descrição do Problema

[Descrever de forma objetiva o problema que está sendo resolvido pelo projeto. A tabela abaixo é uma sugestão de como organizar essa descrição:]

Problema	<i>[Descrever o problema]</i>
Pessoas Atingidas	<i>[Listar os stakeholders afetados pelo problema]</i>
Impacto	<i>[Descrever os impactos do problema]</i>
Uma solução bem sucedida traria	<i>[Listar os principais benefícios trazidos por uma solução bem sucedida]</i>

3. Descrição dos Stakeholders

[Para fornecer eficazmente produtos e serviços que atendem as necessidades dos stakeholders, é necessário identificar e envolver todos os stakeholders no processo de levantamento dos requisitos. Neste conjunto também se encontram os usuários do sistema. Esta seção deve traçar o perfil dos stakeholders da aplicação e dos problemas chaves que limitam sua produtividade. Esta seção não deve ser utilizada para indicar exigências específicas de um stakeholder e sim para fornecer a fundamentação e a justificativa de porque os requisitos são necessários.]

3.1 Resumo dos Stakeholders

[Nesta seção devem ser listadas as informações sobre os stakeholders, isto é, todos os envolvidos com o sistema, de forma direta ou indireta.

Descrição das colunas:

- *Identificação:* grupo que o stakeholder representa. Exemplo: “Departamento de Recursos Humanos”
- *Responsabilidades:* pelo que este grupo é responsável no contexto deste projeto
- *Stakeholders:* nomes dos contatos para o projeto naquele grupo. Exemplo: José da Silva.
- *Observações:* campo livre para anotar alguma informação que não se encaixe nas outras colunas, se houver.]

Identificação	Responsabilidades	Stakeholders (nomes dos contatos)	Observações

3.2 Ambiente do Usuário

[Detalhe o ambiente de trabalho do usuário alvo. Algumas sugestões:

- *Número de pessoas envolvidas para completar a tarefa?*
- *Qual a duração do ciclo da tarefa? Quantidade de tempo gasto em cada atividade?*
- *Qualquer restrição de ambiente: móvel, externo , etc...*
- *Quais as plataformas de sistemas utilizadas?*
- *Que outras aplicações são usadas? Sua aplicação precisará se integrar com elas?]*

3.3 Necessidades chave dos Stakeholders

[Com base nos principais problemas dos sistemas atuais do ponto de vista dos usuários é possível levantar as necessidades chave com relação ao novo sistema. Esclareça as seguintes questões pra cada problema:

- *Quais as razões para esse problema?*
- *Como isso é resolvido agora?*
- *Que soluções o usuário quer?*

É importante compreender a importância relativa que usuário coloca em resolver cada problema. É preciso identificar os problemas que devem ser resolvidos e as questões que gostariam que fossem resolvidas.

As necessidades são justamente as soluções a serem dadas para os problemas.]

No.	Descrição	Prioridade	Observações
1			

4. Visão Geral do Produto

[Esta seção fornece uma visão de alto nível das potencialidades do produto, das interfaces com outras aplicações e de configurações de sistemas. Esta seção consiste geralmente em três subseções, como segue:

- *Perspectiva do produto;*
- *Funções do produto;*
- *Premissas e dependências.]*

4.1 Perspectiva do Produto

[Esta subseção do documento de requisitos deve pôr o produto em perspectiva a outros produtos relacionados e ao ambiente de usuário. Se o produto for independente e totalmente auto contido, indique-o aqui. Se o produto for um componente de um sistema maior, esta subseção deve relacionar como estes sistemas interagem e identificar as relações relevantes entre os sistemas. Devem ser indicado os componentes principais do sistema maior, das interconexões e das relações externas.]

Essas informações podem ser representadas através de um diagrama de contexto (diagrama de casos de uso dando uma visão geral do sistema).]

4.2 Resumo das características do produto

[Resuma os benefícios principais e características que o produto fornecerá, sem mencionar detalhes de cada uma destas funções.]

Organize as funções de modo que a lista seja compreensível ao cliente ou a qualquer um que leia o documento pela primeira vez. Uma tabela simples que liste os benefícios chave e suas características suportadas deve bastar. Por exemplo:]

Benefícios para o cliente	Funcionalidades correspondentes

4.3 Premissas e Dependências

[Liste cada um dos fatores que afetam as características indicadas no documento. Liste as suposições que, se mudadas, alterarão o Documento de Requisito. Por exemplo, uma suposição pode indicar que um sistema operacional específico estará disponível para o equipamento designado para o produto de software. Se o sistema operacional não estiver disponível, o Documento de Requisitos precisará ser modificado.]

5. Requisitos Funcionais do Produto

[Liste e descreva as funcionalidades do produto. As funcionalidades são as características de alto nível do sistema que são necessárias para entregar benefícios aos usuários. Cada funcionalidade é um serviço externamente desejado que requeira tipicamente uma série de entradas para conseguir o resultado desejado. Por exemplo, poderia ser adotada a técnica de casos de uso. A medida que o modelo de casos de uso toma forma, atualize a descrição da funcionalidade referente ao caso de uso.]

Bastante detalhe deve estar disponível para fornecer à equipe a informação que necessita para criar um modelo de caso de uso.

As funcionalidades fornecem a base fundamental para a definição de produto, a gerência do escopo e a gerência de projeto. Cada funcionalidade será expandida em um detalhe maior no modelo de casos de uso.

Durante toda esta seção, cada funcionalidade deve ser externamente percebida por usuários, por operadores ou por outros sistemas externos. Estas funcionalidades devem incluir uma descrição da função e de todas as questões relevantes para seu uso que devem ser levadas em consideração. Evite o projeto. Mantenha descrições da característica em um nível geral. Focalize nas características necessárias e porque, (não como) devem ser executadas.]

5.1 <Funcionalidade 1>

5.2 <Funcionalidade 2>

....

5.n <Funcionalidade n>

6. Precedência e Prioridade

[Defina a prioridade das diferentes funcionalidades do sistema de acordo com as necessidades dos clientes. As prioridades podem ser classificadas em essencial, importante e desejável. Nada impede que se utilize outra forma de classificação. Indique também a precedência entre as funcionalidades, se existir (Por exemplo, se determinada funcionalidade só pode ser desenvolvida se uma outra existir).]

No.	Funcionalidade	Prioridade	Precedência
1			
2			

7. Requisitos Não-Funcionais do Produto

[Esta seção é dedicada aos requisitos não-funcionais, tais como requisitos de usabilidade, confiabilidade, eficiência, manutenibilidade, portabilidade etc. Deve ser criado um subtítulo para cada tipo de requisitos não-funcionais que for pertinente ao projeto. Esta seção merece especial atenção, uma vez que a árvore de características, subcaracterísticas e atributos de qualidade será preenchida a partir desta, após o estudo da Engenharia do Domínio da aplicação. Segue uma lista dos tipos mais comuns de requisitos não-funcionais para auxiliar no levantamento:]

7.1 Requisitos de Usabilidade

[Esta seção deve incluir todos os requisitos que afetam a usabilidade. Exemplos:

- *Tempo de treinamento exigido para um usuário normal e usuários experientes tornarem-se produtivos em dada operação;*
- *Mensuração de tempo para um tarefa;*
- *Utilizar os requisitos de usabilidade de outros sistemas que os usuários conhecem e gostam;*
- *Grupos de usuários;*
- *Requisitos para obedecer um padrão comum – por exemplo, IBM's CUA standards, ou GUI standards publicados pela Microsoft para o Windows 95.]*

7.2 Requisitos de Confiabilidade

[Listar os requisitos de confiabilidade, se existirem. Sugestões:

- *Disponibilidade – especifica a porcentagem (%) de tempo disponível, horas de uso, modo de degradação das operações etc.*
- *Tempo entre falhas – isso é normalmente especificado em horas, mas pode também ser especificado em termos de dias, meses ou anos.*
- *Tempo de reparo – quanto tempo o sistema pode ficar fora do uso durante uma falha.*
- *Precisão – especifica a precisão que é exigida para o sistema.*

- *Limite aceitável de erros de programação ou defeitos – normalmente expressos em termos de erros de programação por milhares de linhas de código ou erros de programação por ponto de função.*
- *Taxa de erros de programação ou defeitos – categoriza em temos erros pequenos, significantes e críticos.*
- *Política de privacidade – especifica que informações fornecidas pelos usuários não serão utilizadas de forma indevida*
- *Segurança nas transações – quais algoritmos criptográficos estão sendo usados e com se faz o controle de acesso.]*

7.3 Requisitos de Eficiência

[Use esta seção para detalhar exigências de desempenho. As questões de desempenho podem incluir artigos como fatores de carga do usuário, número de usuários, tamanho de arquivos, tipos de conexões, largura de banda ou de capacidade, de exatidão, de confiabilidade ou de tempos de resposta de uma comunicação sob variadas condições de carga.]

7.4 Requisitos de Manutenibilidade

[Detalhe os requisitos de manutenibilidade, explicando como devem ser feitas as manutenções corretivas, evolutivas ou adaptativas. Está diretamente ligada à disponibilidade, já que alguns sistemas precisam ser mantidos sem que fiquem indisponíveis]

7.5 Requisitos de Portabilidade

[Detalhe os requisitos de portabilidade, explicando em quais ambiente os sistema deve operar, quais os requisitos e o esforço necessários para usar o sistema em um ambiente novo etc, defina todas as exigências do sistema necessárias para suportar a aplicação. Estas podem incluir os sistemas operacionais e as plataformas suportadas da rede, configurações, memória, periféricos e software.]

7.6 Outros Requisitos

7.6.1 Padrões Aplicáveis

[Liste todos os padrões com que o produto deve estar de acordo. Estes podem incluir os padrões legais e reguladores, de comunicações, os padrões de plataforma, qualidade e os padrões de segurança.]

7.6.2 Requisitos de Ambiente

[Detalhe exigências ambientais. Para sistemas baseados em hardware, questões ambientais podem incluir a temperatura, choques, a umidade, a radiação, etc.. Para aplicações do software, os fatores ambientais podem incluir condições do uso, ambiente de usuário, disponibilidade de recurso, questões de manutenção, manipulação de erro e recuperação.]

7.6.3 Requisitos de Suportabilidade

[Descreve qualquer requisito que pode aumentar a suportabilidade do sistema, incluindo codificação padrão, convenção de nomes, bibliotecas de classes,etc.]

7.6.4 Requisitos de Documentação

[Descrever os requisitos, se existirem, de documentação de usuário, tais como Manual do usuário, Ajuda on-line, Guia de Instalação e Configuração e Arquivo Leia-me, Rotulagem e Empacotamento]

8. Interfaces

[Esta seção define as interfaces que devem ser suportadas pela aplicação.]

8.1 Interfaces de Usuário

[Descreve as interfaces a serem implementadas pela aplicação.]

8.2 Interfaces de Hardware

[Descreve quaisquer interfaces de hardware que são suportadas pelo software, incluindo estrutura lógica endereço físico, comportamento esperado, etc.]

8.3 Interfaces de Software

[Descreve interfaces de software para outros componentes do sistema. Podem ser componentes comprados, componentes reusados e/ou componentes desenvolvidos para subsistemas fora do escopo mas, que interagem com o sistema.]

8.4 Interfaces de Comunicação

[Descreve quaisquer interfaces de comunicação para outros sistema ou dispositivos, tais como redes locais, dispositivos seriais remotos.]

9. Restrições

[Listar todas as restrições do projeto, restrições externas ou outras dependências. Devem ser listadas restrições políticas, econômicas e técnicas, se houver. Esta seção deve indicar qualquer restrição de projeto(design) do sistema a ser construído. Restrições de projeto representam decisões as quais o projeto(design) do sistema deve se adaptar. Exemplos dessas restrições incluem linguagens de desenvolvimento específicas, componentes adquiridos, bibliotecas de classes, ferramentas a serem utilizadas no desenvolvimento, restrições de arquitetura, etc.]

10. Recomendações.

[Listar os atributos que foram identificados como ausentes na metodologia REQE]

Apêndice B - Documento de Requisitos do Exemplo

Esse apêndice mostra o documento de requisitos utilizado no exemplo presente no capítulo 5, sobre a avaliação de qualidade de um site acadêmico.

Site da universidade X

Documento de Requisitos

Versão 1.0.0

Histórico de Revisão

Data	Versão	Descrição	Autor
--	1.0.0	Versão Inicial	--

Sumário

1.	Introdução	.
1.1	Finalidade	.
1.2	Escopo	.
1.3	Definições, Acrônimos, e Abreviações	.
1.4	Referências	.
1.5	Visão geral	.
2.	Contextualização do Produto	.
2.1	Oportunidade de Negócio	.
2.2	Descrição do Problema	.
3.	Descrição dos <i>Stakeholders</i>	.
3.1	Resumo dos <i>Stakeholders</i>	.
3.2	Ambiente do Usuário	.
3.3	Necessidades chave dos <i>Stakeholders</i>	.
4.	Visão Geral do Produto	.
4.1	Perspectiva do Produto	.
4.2	Resumo das características do produto	.
4.3	Premissas e Dependências	.
5.	Requisitos Funcionais do Produto	.
5.1	<Funcionalidade 1>	.
5.2	<Funcionalidade 2>	.
6.	Precedência e Prioridade	.
7.	Requisitos Não-Funcionais do Produto	.
7.1	Requisitos de Usabilidade	.
7.2	Requisitos de Confiabilidade	.
7.3	Requisitos de Eficiência	.
7.4	Requisitos de Manutenibilidade	.
7.5	Requisitos de Portabilidade	.
7.6	Outros Requisitos	.
7.6.1	Padrões Aplicáveis	.
7.6.2	Requisitos de Ambiente	.
7.6.3	Requisitos de Suportabilidade	.
7.6.4	Requisitos de Documentação	.
8.	Interfaces	.
8.1	Interfaces de Usuário	.
8.2	Interfaces de Hardware	.
8.3	Interfaces de Software	.
8.4	Interfaces de Comunicação	.
9.	Restrições	.
10.	Recomendações	.

Documento de Requisitos

1. Introdução

1.1 Finalidade

A finalidade deste documento é definir os requisitos funcionais e não-funcionais do do site acadêmico da universidade X, em termos de suas necessidades. O documento contém uma visão geral dos requisitos mais importantes, sendo a base do acordo quanto às características de qualidade do sistema.

1.2 Escopo

Esse documento lista as necessidades dos usuários que deverão ser atendidas pelo sistema.

1.3 Visão geral

O documento está estruturado da seguinte forma:

- Seção 2 – Contextualização do Produto: descrição do contexto em que surgiu a idéia/necessidade de se fazer o projeto: oportunidade, para resolver qual problema, etc.
- Seção 3 – Descrição dos *Stakeholders*: descrição dos principais envolvidos/interessados no projeto, com suas responsabilidades e forma de contato. São colocadas aqui também as necessidades desse grupo de pessoas com relação ao projeto.
- Seção 4 – Visão Geral do Produto: descrição geral dos benefícios trazidos pelo produto, como ele se relaciona a outros no ambiente em que será implantado e quais as premissas e dependências desse projeto.
- Seção 5 – Requisitos Funcionais do Produto: listagem e breve descrição dos requisitos funcionais do produto (correspondentes aos casos de uso do sistema, caso seja adotado um processo de desenvolvimento orientado a objetos).
- Seção 6 – Precedência e Prioridade: priorização dos requisitos funcionais listados anteriormente de acordo com as necessidades do cliente e da precedência lógica existente entre elas (se existir).
- Seção 7 – Requisitos Não-Funcionais do Produto: listagem e breve descrição dos requisitos não-funcionais do produto, tais como usabilidade, confiabilidade, eficiência, manutenibilidade, portabilidade etc.
- Seção 8 – Interfaces: descrição das interfaces do sistema, sua comunicação com o mundo externo, em linhas gerais. São descritas as interfaces de usuário, de software, de hardware e de comunicação.

- Seção 9 – Restrições: descrição das restrições do projeto, restrições externas ou outras dependências. São listadas restrições políticas, econômicas e técnicas, se houver.

2. Contextualização do Produto

Oportunidade de Negócio

A criação do site da universidade X tem como objetivo principal a divulgação das ações e notícias relativas ao seu contexto acadêmico. Além das informações da própria universidade, teremos a oportunidade de detalhar o perfil dos colaboradores, com certo destaque aos docentes da instituição; os cursos oferecidos nos mais diversos níveis, desde a graduação até o doutorado; os serviços oferecidos ao público etc. Enfim, o site facilitará a vida daqueles que interagem ou têm o desejo de interagir com esse centro de excelência, tanto a nível nacional quanto internacional.

2.2 Descrição do Problema

Problema	A universidade X precisa ser divulgada e necessita facilitar o contato dos interessados.
Pessoas Atingidas	Atuais e futuros professores, pesquisadores, alunos, funcionários, colaboradores etc
Impacto	Dificulta a agregação de valores intelectuais, uma vez que o não divulgado corresponde ao não conhecido.
Uma boa solução traria os benefícios	Apresentar à sociedade como um todo, em especial à comunidade acadêmica o que faz a universidade X.

3. Descrição dos Stakeholders

3.1 Resumo dos Stakeholders

Identificação	Responsabilidades	Stakeholders (nomes dos contatos)	Observações
Professores	Transmitir conhecimento aos alunos	Jaelson de Castro	
Alunos	Absorver conhecimento repassado pelos professores	Tiago Reis	
Comunidade Acadêmica	Investigar, estudar e evoluir as pesquisas realizadas	Luis Olsina	
Funcionários	Apoiar administrativamente as ações da universidade	Maria Lilia	

3.2 Ambiente do Usuário

O ambiente de trabalho dos usuários é bastante amplo pelo fato do site servir de apoio para qualquer internauta que deseje interagir com a universidade X, independente do local e da máquina de acesso. O site deve estar disponível tanto para o ambiente Windows quanto para o Linux. O tempo de utilização do sistema pelo usuário dependerá do tipo de atividade que este esteja interessado.

3.3 Necessidades chave dos Stakeholders

Atualmente não existe um meio de divulgação eficaz capaz de divulgar o que se passa e se produz na universidade X. Faz-se necessário um real investimento para este fim. Por meio do desenvolvimento deste site será possível suprir essa deficiência.

4. Visão Geral do Produto

O produto é um site acadêmico desenvolvido para a universidade X.

5. Requisitos Funcionais do Produto

5.1 Publicar informações relativas a universidade

Divulgar informações sobre:

- A universidade propriamente dita, histórico, departamentos, cursos, quantidade de professores, de alunos de graduação, de pós-graduação, de funcionários, de salas de aula, de laboratórios, a posição no organograma da universidade, a sua localização física etc;
- Infra-estrutura oferecida;
- A missão institucional e os objetivos da universidade X junto à sociedade, ao país, à região, ao mundo etc;
- As leis ou resoluções que amparam sua criação e existência;

5.2 Publicar informações relativas às pessoas que fazem parte da universidade X

Divulgar informações sobre diretores, coordenadores, docentes, funcionários e alunos. Para todas as pessoas deve estar disponível o endereço de e-mail e a página na internet.

Para o grupo de diretores, coordenadores e docentes deverá ser disponibilizado o currículo e outras informações como telefones, sala, áreas de interesse etc.

5.3 Publicar informações sobre os cursos de graduação

Divulgar informações sobre:

- Histórico do curso;
- Os cursos oferecidos;
- Recursos;

- Matrícula;
- Grade curricular;
- Regimentos;
- Professores;
- Direitos e deveres dos alunos;
- Oportunidades de monitoria e iniciação científica;
- Formas de ingresso nos cursos, horários dos cursos, responsáveis etc;
- Calendário anual das aulas e provas;
- Congressos e eventos em geral.

5.4 Publicar informações sobre os cursos de pós-graduação

Divulgar informações sobre:

- Histórico dos cursos;
- Extensão;
- Especialização;
- Mestrado
- Doutorado;
- Recursos;
- Matrícula;
- Grade curricular;
- Regimentos;
- Professores;
- Direitos e deveres dos alunos;
- Oportunidades de monitoria e iniciação científica;
- Formas de ingresso nos cursos, horários dos cursos, responsáveis etc;
- Calendário anual das aulas e provas;
- Congressos e eventos em geral.
- Linhas de pesquisa;

5.5 Publicar informações sobre pesquisas realizadas na universidade X

Divulgar informações sobre todas as linhas de pesquisa, a saber:

- Banco de dados;
- Ciências Cognitivas e Tecnologia Educacional;
- Engenharia da Computação;
- Inteligência Artificial Simbólica;
- Inteligência Computacional;
- Laboratório de Engenharia de Requisitos;
- Linguagens de Programação e Engenharia de Software;
- Probabilidade Intervalar;
- Processamento Gráfico e Multimídia;
- Redes de Computadores e Sistemas Distribuídos;
- Sistemas Inteligentes e Sistemas Dinâmicos;
- Teoria e Fundamentos Lógicos da Computação.
- Livros, capítulos, artigos, trabalhos em periódicos, trabalhos em anais, relatórios, dissertações e teses. Todo o material deve estar disponível para download quando possível;

5.6 Cadastramento de assinantes

Permite o cadastramento do endereço de e-mail a fim de receber notícias na caixa postal eletrônica. Deve ser possível selecionar as áreas de interesse para receber notícias, como também palavras chave. Essa solução pode ser adquirida por meio de um componente externo.

5.7 Manutenção de assinantes

Permite a modificação do cadastro de assinantes, de forma a atualizar dados.

5.8 Divulgação

Uma das funcionalidades da aplicação é a publicação de notícias. Toda notícia divulgada deve receber um destaque especial, a fim de facilitar a identificação de novas publicações. Deve ser possível fazer o “upload” de imagens, bem como atualizar notícias já divulgadas. Todas as notícias devem ser persistidas no banco de dados, para posteriores consultas. Essa funcionalidade pode ser implementada com o apoio de alguma ferramenta externa, a ser escolhida na fase de elaboração. A exclusão de uma notícia deve funcionar apenas como sua desativação (exclusão lógica), e não fisicamente. No entanto, qualquer notícia excluída não será exibida pelos mecanismos de busca do site. Ao publicar uma nova notícia é preciso preencher alguns campos:

- Título da notícia
- Data da publicação
- Hora da publicação
- Texto da notícia
- Imagens da notícia (as imagens são dispostas no início das notícias)
- Fonte

Com o advento da data-hora da publicação, é possível agendar a publicação de uma notícia.

Qualquer operação deve ser precedida de um logon, onde o usuário entra com o login e a senha. Toda e qualquer publicação no ambiente de produção (disponível para a internet, em contrapartida com o ambiente de teste) deve ser avaliada pelo assessor de comunicação, de forma a garantir a qualidade do que está sendo escrito. É atribuição deste assessor revisar constantemente o conteúdo do site a fim de garantir um certo nível de qualidade no conteúdo do mesmo.

5.9 Envio de sugestões

O site deve oferecer um link para possíveis sugestões, dúvidas ou reclamações. Ao clicar o link, uma janela do programa de correio eletrônico do cliente deve ser aberta, com o endereço de e-mail do destinatário preenchido.

5.10 Buscas

Deve estar disponível um mecanismo de busca de notícias.

5.11 Newsgroup

O site deve prover serviços de newsgroup.

5.12 Mapa Campus

O site deve possuir um mecanismo de informação geográfica do campus da universidade X por meio de um mapa de imagens. Um mapa de imagens é uma imagem em uma página da Web que contém diversos vínculos a outros documentos, páginas, subsites etc. Se você clicar em uma determinada área desta imagem, ela o levará a outra página. Deve ser possível acessar cada um dos departamentos por meio desse mapa do site. Isso vai facilitar não apenas a localização geográfica, como também o acesso às informações dos mesmos.

5.13 Busca

O mecanismo de busca deve permitir uma série de refinamentos:

- Por todas as palavras-chave digitadas;
- Por uma das palavras-chave digitadas;
- Por sub-páginas;
- Por data de publicação/atualização das páginas.

Uma ajuda on-line deve ser disponibilizada de forma a auxiliar o uso do site, informando detalhadamente como as buscas devem ser realizadas, incluindo exemplos. Cada campo existente nas buscas deve ser acompanhado de um “hint-help”, recurso pelo qual aparece uma explicação quando o mouse é posicionado ao lado dos campos. A ajuda on-line deve ser apresentada am um conjunto de usuários que não são familiarizados com o site a fim de receber opiniões de como estão as explicações de auxílio. Os mecanismos de refinamento das buscas deve se basear em http://www.google.com.br/advanced_search?hl=pt-BR.

5.14 Cursos x Disciplinas x Professores x Alunos

No contexto da busca convencional e global deve ser disponibilizada uma forma para pesquisar exclusivamente pessoas, cursos e disciplinas acadêmicas.

Uma tabela de todos os cursos x disciplinas x professores x alunos pode ser um meio de facilitar a busca, sem preencher campos de pesquisa.

Uma árvore hierárquica tendo como nós de primeiro nível os cursos, nós de segundo nível as disciplinas, nós de terceiro nível os professores e nós de quarto nível os alunos seria bastante útil. Por exemplo:

1. Curso1
 - 1.1. Disciplina 1
 - 1.1.1. Professor 1
 - 1.1.1.1. Aluno 1
 - 1.1.1.2. Aluno 2
 - 1.1.1.3. Aluno 3
 - 1.1.1.4. Aluno 4
 - 1.2. Disciplina 2
 - 1.2.1. Professor 1
 - 1.2.1.1. Aluno 1
 - 1.2.1.2. Aluno 3
 - 1.2.1.3. Aluno 4
 - 1.2.2. Professor 2
 - 1.2.2.1. Aluno 2
 - 1.3. Disciplina 3
 - 1.3.1. Professor 3
 - 1.3.1.1. Aluno 2
 - 1.3.1.2. Aluno 4
 - 1.4. Disciplina 4
 - 1.4.1. Professor 4
 - 1.4.1.1. Aluno 1
 - 1.4.1.2. Aluno 3
 - 1.5. Disciplina 5
 - 1.5.1. Professor 3
 - 1.5.1.1. Aluno 3
 - 1.5.1.2. Aluno 4
2. Curso 2
 - 2.1. etc

5.15 Subsite de disciplinas

Todas as disciplinas devem possuir um site, com informações sobre as mesmas. Estes devem ser acessados pelos cursos que as oferecem.

5.16 Biblioteca virtual

O site deve permitir pesquisa no acervo das bibliotecas. É de fundamental importância que todos os livros catalogados possam ser ao menos consultados, e também deve ser informada a sua situação, isto é, se estão disponíveis ou emprestados. As buscas devem ser efetuadas por um ou uma combinação de:

- Nome do autor;
- Nome do título;
- ISBN;
- Palavras chave;
- Disciplinas relacionadas;
- Cursos relacionados.

Regras gerais como multas por atraso, meios para renovação, tempo de empréstimo etc devem estar disponíveis.

5.17 Laboratórios

Os laboratórios devem ser passíveis de reserva por meio do site. Os recursos de cada laboratório devem ser listados (quantidade de máquinas, perfis das máquinas, retro-projetores, canhões de luz etc), bem como seus horários já reservados. Após a reserva um protocolo deve ser gerado. Deve ser possível participar da lista de espera, que também emite um protocolo para acompanhamento futuro.

5.18 Comunicação

Deve ser implementada uma página que informe os dados a fim de facilitar a comunicação com as pessoas. Um requisito fundamental é que a mesma esteja sempre atualizada. Isso inclui o endereço eletrônico de todos os professores, funcionários, alunos, colaboradores etc, os telefones dos departamentos, o ramais internos, fax, endereço para correspondência etc.

5.19 Manutenção de requerimentos

Uma das funcionalidades presentes será a entrada de requerimentos.

Após a entrada no sistema um protocolo será gerado para futuras consultas.

Há quatro estados possíveis para os requerimentos:

- Aguardando análise;
- Deferido;
- Indeferido;
- Cancelado.

Alguns requerimentos exigem a presença do requerente na devida secretaria e apenas podem ser entrados manualmente, após o preenchimento de formulário de entrada.

Um índice de todos os requerimentos e serviços disponíveis deve estar presente.

6. Requisitos Não-Funcionais do Produto

6.1 Requisitos de Usabilidade

Visando facilitar a utilização do site deve ser disponibilizado um mecanismo de ajuda.

A coleta de opiniões por parte dos usuários do site é uma maneira bastante útil para se aumentar a usabilidade, sendo considerado um investimento na qualidade do mesmo. A disponibilização de questionários pode ser uma boa estratégia.

Os links já visitados devem ser marcados, de modo a facilitar a navegabilidade.

Deve existir uma preocupação com a intuitividade dos links e controles. Para isso, é necessário descobrir a palavra-chave ideal para “disparar” qualquer link ou controle. Esta deve ser escolhida pelo principal usuário juntamente com o especialista no domínio, uma vez que têm o conhecimento necessário para avaliar como determinadas operações, expressões, termos etc devem ser chamados. Um site intuitivo é um grande passo para ser fácil de usar. Todos os termos devem estar listados no Documento de Glossário, a fim de esclarecer eventuais dúvidas. Os rótulos dos controles devem passar por revisões, com o objetivo de estar atualizado.

Durante a navegação, os principais controles também devem estar disponíveis nas páginas secundárias. Portanto, o menu principal deve ser acessível de qualquer página dentro do site. Ao clicar determinado item do menu principal, os sub-menus deste também devem ser acessíveis enquanto se está dentro do item pai. Uma outra preocupação é a coesão no agrupamento dos controles. Isso reflete na disposição dos menus e sub-menus. O especialista no domínio deve opinar como estes devem aparecer para o usuário.

O mapa de links deve ser revisado pelo analista de requisitos chefe a fim de identificar eventuais ambigüidades com relação à qualidade dos títulos dos links.

7. Recomendações.

Os atributos que receberam escores de conformidade “*Não Atende*”, devem ser listados nesta seção. As causas destas ausências devem ser avaliadas.

Observação: As outras seções do template (Apêndice A) não foram detalhadas por não fazerem parte do escopo do exemplo do capítulo 5.